Resource Efficient Quality of Service Management for NGN Services in Federated Cloud Environments

vorgelegt von
Dipl-Ing.
Florian Schreiner
geb. in Mussenhausen

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr. Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Axel Küpper

Gutachter/in: 1. Prof. Dr. Thomas Magedanz

2. Prof. Dr. Serge Fdida

3. Prof. Dr. Odej Kao

4. Prof. Dr. Alfonso Ehijo

Tag der wissenschaftlichen Aussprache: 1.10.2014

Berlin 2015

to my mother and my father, who always supported me

Zusammenfassung

Seit dem Aufkommen Internet-basierter Kommunikationsdienste ist die traditionelle Telekommunikationsindustrie einer stets wachsenden Konkurrenz ausgesetzt und steht unter zunehmendem Kostendruck. Die Einführung paketvermittelnder Next Generation Network (NGN) basierter Kommunikationsinfrastrukturen ermöglicht schon heute einer Vielzahl von Netzbetreibern signifikante Kostenersparnisse durch Konsolidierung der Netzinfrastrukturen und durch Konvergenz mobiler, Festnetz-, Kabel- und Satelliten-basierter Zugangsnetze.

Auf der Dienstebene setzen sich zunehmend dienstorientierte Architekturen durch, die Kostenersparnisse durch erhöhte Widerverwendbarkeit von Dienstkomponenten und durch schnellere und einfachere Integration neuer Dienste in Dienstbereitstellungs- und Betriebsunterstützungssysteme ermöglichen.

Auf Grund der ökonomischen Vorteile elastisch skalierender Cloud-basierten Anwendungen und nutzungsbasierter Abrechnungsmodelle werden kommerzielle Cloud Infrastrukturen schon heute von einer Vielzahl von Anbietern Web-basierter Dienste genutzt. Für Anbieter von NGN Diensten stellen mangelnde Mechanismen zur Sicherstellung garantierbarer Dienstgüten und fehlende, elastische Skalierbarkeit von NGN Diensten wesentliche Hinderungsgründe für die Nutzung externer Cloud Infrastrukturen dar.

Für die Überwindung dieser Hindernisse schlägt die vorliegende Arbeit ein System vor, welches Föderationsmechanismen, effiziente Ressourcenallokationstechniken und Mechanismen zur Überwachung und Sicherstellung der Ende-zu-Ende Dienstgüte nutzt um externe Cloudinfrastrukturen für die Erbringung von NGN Diensten nutzbar zu machen. Dabei werden Prinzipen dienstorientierter Architekturen, Politiken-basiertes Management und geschlossene Regelkreise aus dem Bereich des Autonomic Computing ausgenutzt.

Aus der Perspektive von Anbietern NGN Dienste werden die Anforderungen für Ressourcen-effizientes Dienstgütemanagement von, auf föderierten Cloud Infrastrukturen betriebenen Diensten analysiert. Die vorgeschlagenen Mechanismen werden anhand von realen NGN Diensten, realistischen Betriebslasten in emulierten und geographisch verteilten Cloud-Umgebungen validiert und evaluiert. Eine Leistungsbewertung gibt Aufschluss über die Anwendbarkeit, Grenzen und Wirksamkeit des vorgeschlagenen Ansatzes.

Die vorliegende Arbeit wurde im Rahmen der Tätigkeit des Autors als wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Offene Kommunikationssysteme (FOKUS) in Berlin erstellt. Die Ergebnisse wurden im Rahmen nationaler und internationaler Forschungs- und Industrieprojekte erarbeitet und publiziert. Die Ergebnisse flossen auch in offene IEEE Standards für Föderationsmechanismen für Cloud Plattformen, in Industrielösungen und weitere Forschungsprojekte ein.

Schlagwörter: NGN, Cloud, Skalierbarkeit, Dienstgüte, Föderation, Vermittlung, IaaS, Multi-Cloud Brokerage

Abstract

Challenged by an ever-increasing competition of purely internet-based providers of communication services, incumbent NGN service providers are constantly looking for new means that are allowing for lowering capital expenditures and operational costs. To this end the majority of telecommunication operators have lately adopted all-IP based Next Generation Network (NGN) platforms for network infrastructure consolidation and for convergence of fixed, mobile, cable and satellite networks.

For cost-efficiently providing NGN services, service oriented architecture (SOA) based service delivery mechanisms are continuously being utilized, allowing for increased service re-usability and lower service integration and management overheads.

Elastic cloud computing mechanisms as well as attractive pay-per-use cost models of public cloud providers already have attracted a broad range of web service providers. For providers of real-time, conversational, highly reliable NGN services however, the lack of mechanisms for cost-efficient and quality-assured service delivery still represents a significant obstacle for utilization of public clouds.

This work proposes a system which tries to overcome these obstacles, exploiting the cloud provider market through cloud federation mechanisms, efficient cloud resource allocation mechanisms as well as on End-to-End QoS monitoring and assurance mechanisms. To this end, the approach exploits SOA principles, policy-based management, and autonomic computing mechanisms.

Taking the viewpoint of NGN service providers, this work analyses the requirements and critical factors for resource efficient, service quality management of NGN services that are deployed on multiple federated cloud infrastructures. Proposed mechanisms are validated and evaluated in isolated testbeds as well as large-scale multi-cloud facilities, operating standard-based NGN platforms and delivering real NGN services under realistic NGN workloads. The performance evaluations provide insights into applicability, effectiveness and efficiency of the proposed approach.

The author, a senior scientist with Fraunhofer's Institute for Open Communication Systems (FOKUS) in Berlin, leading the Future Internet team at the department for Next Generation Network Infrastructures (NGNI) conducted, this work in the context of several industry, national and European research projects, supervising several diploma and master theses. Results were published in multiple research papers, books and journals, and frequently presented at international tutorials. Aspects of the developed system were used for contributions to the IEEE cloud federation standardization, integrated into industry solutions, testbed infrastructures and used as a basis for further, currently ongoing research projects.

Keywords: NGN, SOA, Cloud, Computing, Elasticity, QoS, Federation, Brokerage, IaaS, Multi-Cloud

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

For almost a decade, the telecommunication industry is at a crossroads. Since the emergence of purely internet-based communication services, so called Over-the-Top (OTT) communication services, traditional telecommunication network operators are witnessing steady erosion of revenues generated by call minutes and messages. As OTT communication service providers do not operate any network infrastructures, they are able to offer their services at significantly lower prices, in many cases free-of-charge. Although the Quality of Service (QoS) of OTT services cannot be assured on an end-to-end basis, for many non-critical communications, best-effort, free-of-charge OTT services can frequently be provided at satisfactory QoS levels.

In order to stay competitive in this rapidly changing market, Telcos are required to maintain their position as providers of reliable, quality-assured NGN services. Furthermore they are required to keep pace with the globally ongoing innovations of NGN services (i.e. minimize services' time-to-market) and to save costs (i.e. minimizing operational as well as capital expenditures).

To this end, Next Generation Network (NGN) principles and architectures are continuously utilized by Telcos for *consolidating* the landscape of telecommunication networks, platforms and services. Horizontal NGN architectures not only foster convergence at the network layer, they also promote harmonization of the NGN service layer. By utilizing harmonized Internet Protocol (IP) – based signaling for NGN service control, operators already foster the *convergence* of fixed, mobile, cable and satellite networks.

Moreover, by exploiting service oriented architecture (SOA) principles, an integration of Information Technology (IT) management with NGN service environments is realized. SOA-based NGN service delivery platforms (SDPs) provide efficient and flexible means for managing the NGN service lifecycle, as well as versatile means for service re-use and composition, thus helping to reduce NGN service's time-to-market. Such SOA-based service compositions, no matter whether being NGN customer-facing services or NGN business processes can be operated in distributed, multi-domain environments.

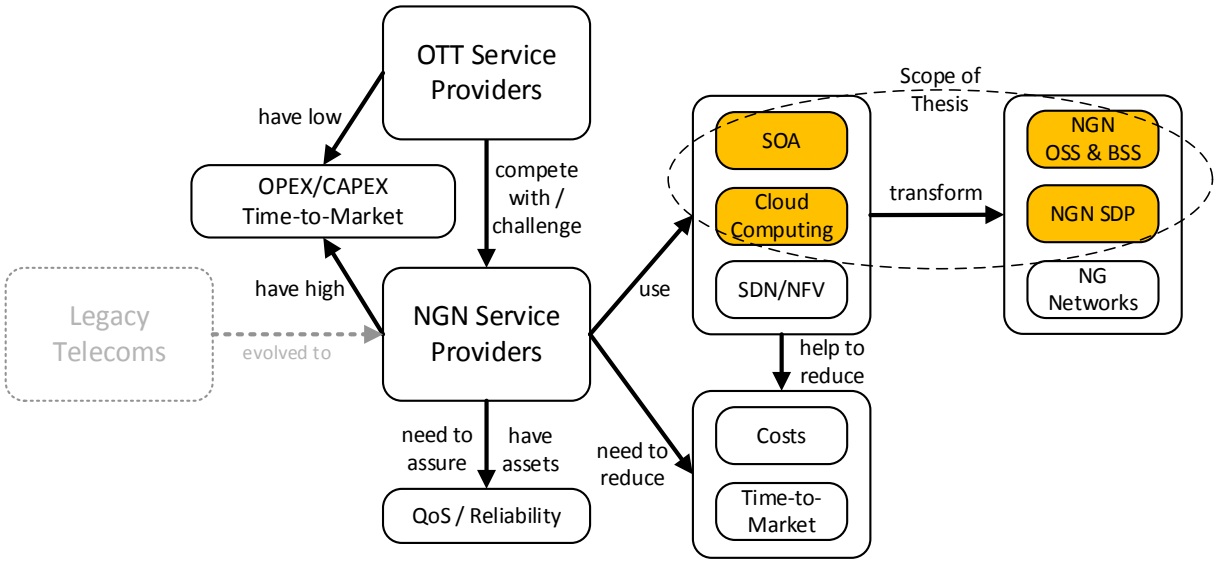

Figure 1: Scene and Scope of Thesis

This is where Cloud infrastructures provided as a service come into play. Virtualization techniques already provide cost-efficient means for IT and NGN service infrastructure consolidation. Cloud infrastructure federation and brokerage, together with elastic resource scaling mechanisms, are promising new opportunities for cost-efficient resource allocation, auto-scaling and on-demand outsourcing. This work, as shown in Figure 1 investigates such mechanisms and analyses how they could be employed by NGN service providers in order to save costs. Whereas one part of the author's publications focus on applying SOA mechanisms to the NGN SDP ([1], [2], [3], [4], [6], [7], [8], [9]), and whereas another part focuses on exploiting SOA mechanisms for enhancing NGN OSS/BSS ([10], [11], [12], [13], [14]. [15], [16], [17]), the author's work integrating Cloud Computing and SOA mechanisms with NGN SDP and NGN OSS/BSS for resource efficiently providing QoS assured NGN services on federated cloud environments is published in [18], [19], [20], [21], [22][1].

1.2 Relevance

Timeliness and general relevance of *elastic* cloud computing mechanisms:

For setting the European cloud computing research agenda and for identifying opportunities and directions of European cloud computing research, the European Commission (EC) convened an expert group (from industry and research) on cloud computing. The outcome, a report [23] on the future of cloud computing, presented in 2010, as number 1 technical R&D topic, recommended to immediately foster the research on *cross-boundary scalability and elasticity*, considering these capabilities to be *essential* for the future of cloud computing. This

[1] Amongst other related work of the author on NGN service control platforms [51], NGN services [52], and federation of ICT infrastructures [5]

came as no surprise, as the de-facto standard for defining cloud computing from NIST [24], already listed *"rapid elasticity"* to be one of *the* five essential characteristics of clouds. The report further on states that uptake of cloud computing technologies was unattractive to both, cloud providers as well as cloud users as long as resources are still "*wasted unnecessarily*". It states that lack of elastic cloud computing capabilities, "*in particular in large scale situations*", may lead to "*undesired resource consumption*" and that the effective usage of cloud-based applications cannot be predicted "*so as to cater for timely and efficient adaptations*". Needless to say that NGN services, typically provided to millions of subscribers *are* consumed on a large-scale basis.

Timeliness and general relevance of *cloud brokering* mechanisms:

In the following three years, a broad range of Cloud Computing research, focusing on many different research areas (SLAs, cloud security, cloud infrastructure management, cloud service management, cloud federation, etc.) was carried out under the EC's FP7 funding programme (outlined in section 2.11). In order to fix the European roadmap of cloud computing research for the next funding programme (FP8/H2020) , the cloud expert group, in December 2012 released their recommendations [25]. The report makes clear that "*Brokering: Reselling and mediating CLOUD infrastructures and services is a growing market as it helps users in making educated choices for selecting the right CLOUD for specific tasks.*", in line with their previous report [26] where the expert group states that multi-cloud brokering algorithms "*are needed to find the best services given the user requirements and the resource provision*". The report [25], again, emphasizes the importance of research on cloud federation, where, in order to offer "*quality compliant services*", mechanisms are needed that provide "*incorporation of resources and services independent of their location or infrastructure*". In the context of cloud federation and interoperability, the report further states that "*new methods for interoperation of applications (as services) across CLOUDs offerings and federation of multiple heterogeneous CLOUD platforms to appear to the application environment as one uniform platform"* are demanded.

Timeliness and relevance of multi-cloud computing for the telecommunication industry:

In 2012, the International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), in their report on *cloud computing benefits from the telecommunication and ICT perspectives* [27], state that procedures and interfaces are required for *discovering, reserving, leasing, and releasing cloud resources among multiple clouds*. The report further on also states that "*real-time monitoring and control for load allocation to minimize power consumption"* were introduced to ITU-T's cloud focus group, however with limited investigation at that given point in time. Most of the above mentioned, required procedures are reconfirmed in ITU-T Cloud FG's Cloud resource management gap analysis [28], highlighting the importance of multi-cloud resource management mechanisms, where additionally it is stated that "*it should be possible to reserve available resources based on different priorities*", of which "*required quality guarantees*" is mentioned explicitly.

Finally, in their recommendation for a cloud computing framework for end-to-end resource management, released in 2013 [29] the ITU-T Cloud FG outline their "*vision for adoption of cloud resource management in a telecommunication-rich environment*" and "*multi-cloud, end-to-end resource management for cloud services*". The recommendation lists the following functional requirements for a cloud computing platform that enables end-to-end cloud resource management, stating that the platform should:

1) offer the deployment choice and workload portability across multiple Cloud Service Providers (CSPs)

2) provide the ability to support hybrid cloud applications

3) support workload portability and related management capabilities (e.g. control, operation and monitoring) amongst cloud service providers, supporting various cloud deployment models, in a cost-effective way.

Benefits of integrating Cloud Computing with NGNs:

The telecommunication industry already at earlier stages (motivated by Grid Computing technologies) knew about the opportunities that an integration into NGNs of Grid and Cloud Computing technologies provides, as outlined in a study supported by the EC and ETSI TC GRID [30]. Amongst other benefits such as, resource pooling, flexible service composition, decoupling of applications from physical resources, the study states that Cloud/NGN integration allows "*a network operator to virtualize various NGN subsystems, thus providing dynamic scalability, load-balancing, and fault tolerance*". The study furthermore highlights the importance of SOAs (incl. the adherence of SOA principles) for realizing Cloud/NGN integration, as well the benefits (e.g. consolidation, convergence) and the unique selling point of NGN-based QoS-enabled Grid/Cloud applications (i.e. being able to offer QoS guarantees/assurance, as opposed to OTT players). Also in [31] it is stated that the main opportunity that "*Telco clouds*" and "*Virtual Telcos*" (NGN-based IaaS/PaaS) provide to NGN operators, was the difference in value proposition. As opposed to "OTT Clouds", NGN operators would be able to offer a "*service that combines reliable, guaranteed network connectivity together with elastic compute and storage resources*." [31]

Above mentioned viewpoints, at that time, however predominantly focused on the value add of operating local Cloud infrastructures, as opposed to outsourcing / distributing applications across multiple clouds. This aspect is taken care about in the initial analysis of required standards for cloud services by ETSI's FG Cloud [32], where it is stated that regarding applications and services, standards are needed since *"Components of a single application could be deployed across multiple cloud infrastructure providers and possibly reconfigured while running, or with limited interruption, to respond to changes in usage patterns or resource availability, for example."*[32], particularly mentioning the scaling and migration of computational resources as an example for such required multi-cloud re-configurations. ETSI in [32] further state that standards for such multi-cloud-based service deployments would be required as it was "*desirable that the stakeholders (e.g. cloud service*

providers, application provider)" … "*can make informed decisions to achieve good resource utilization and assure application quality.*"

Timeliness and relevance of elastic cloud brokering mechanisms for NGNs:

Already many service providers are migrating their internal IT resources and service infrastructures to cloud-based infrastructures. Some network operators are starting to provide cloud services themselves. This strategy however is minimizing infrastructure investments only to a certain extent and only an option for some telecommunication operators, where others seek to reduce infrastructure costs through means of on-demand, cloud-based infrastructure outsourcing.

Efficient, on-demand infrastructure outsourcing, and efficient resource allocation / renting which strives to benefit from pay-as-you-go cloud service cost models, is achieved by elastic, resource scaling, "cloud elasticity" mechanisms. Continuously adapting the current, rented resource capacity, for resource-efficiently serving the daily load promises infrastructure cost savings. On the more time-constraint other end of cloud-based resource outsourcing mechanisms are so called "Cloud bursting" mechanisms. Cloud bursting mechanisms provide means for rapid and on-demand outsourcing of computing, storage and networking capacities for efficiently (in terms of resources consumption) coping with peak-load situations. While being able to mitigate service degradation during time of peak-load, cloud-bursting mechanisms promise to significantly reduce the amount of usually required, over-provisioned infrastructure capacities. In principle, however, both, elastic scaling as well as cloud bursting services make use of similar resource allocation mechanisms.

Public Infrastructure as a Service (IaaS) cloud service providers are offering on-demand computing, networking and storage resources "as a service" over the internet. For several, standard, usually Web-based services, IaaS providers are even offering means for elastic scalability. For several over the top players, especially web service providers, outsourcing of service infrastructures to public cloud platform providers already provides efficient means to reduce infrastructure costs [33].

In research, the cloud computing paradigm is conceived to be an intermediate step, between Distributed/Grid Computing, SOA and Utility Computing. The concept of Utility Computing, mapped to this work encompasses the notion of a static set of available cloud platforms and cloud resource and static/resource costs. Efficient and economical multi-cloud resource allocation / consumption strive to exploit the cloud infrastructure/platform and service market and its player's competition to its fullest. Already cloud resources are offered on a long-term rental basis and on an on-demand basis. Cloud resource prices are either fixed, over a long term rental contract, fixed for on demand usage or variable, offered on a spot-cloud basis.

For NGN service providers, outsourcing to public, external platforms is only an option if reliable, quality assured service delivery and service level agreements (SLAs) can be assured in order to retain their customers. Otherwise, they would lose their only, unique selling point

in the competition with OTT Players and OTT Services. However, reliable, quality assured conversational, real-time NGN service delivery opposes significantly higher challenges to platform performance and (inter-domain) network reliability and performance as compared to simple web service delivery.

Benefits of integrating elastic, Multi-Cloud Brokering into NGNs:

In a broader sense, not exclusively focusing on NGN platforms and NGN services, the question is how service quality can be assured in federated and hybrid cloud/utility computing scenarios, i.e. on-demand computing, by distributing services across several platforms by finding mitigation and coping strategies for both, platform and network unreliability.

The importance to come up with appropriate solutions for overcoming this challenge is currently well understood in research. Already there is a broad range of research projects conducted which try to tackle the issue from different angles. From a single cloud platform centric view, the research focuses on efficient resource allocation mechanisms. From a technical service provider point of view, research focuses on QoS-related, resource scalability, service availability of multi-cloud services. From service provider's business point of view research focuses on multi-cloud SLA-assurance and efficient infrastructure cost management.

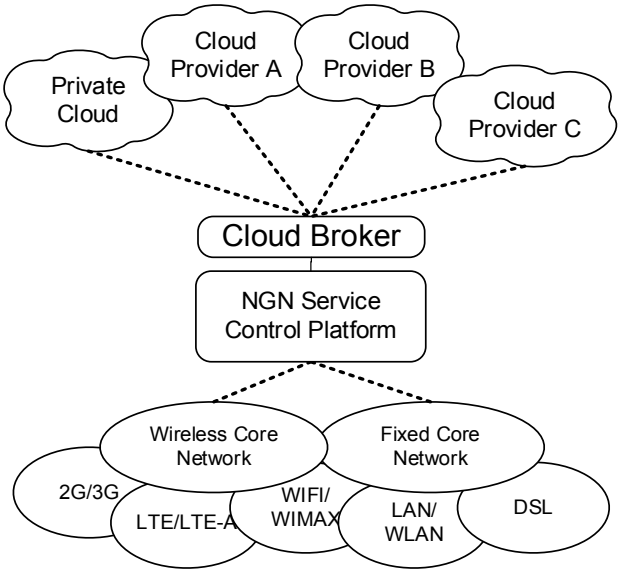

Figure 2: Cloud Broker Integration into NGN Service Platforms

Technically, the notion of cloud brokerage needs to be introduced here. Literature differentiates three modes of cloud brokering: Cloud Service Intermediation, Aggregation and Cloud Service Arbitrage. The focus of this work is on the latter, Cloud Service Arbitrage, i.e. mechanisms that supply flexibility and "opportunistic choices" – and foster competition between clouds. In this regard cloud brokering provides a unified interface for interacting with multiple clouds. Although operating outside of clouds the cloud broker controls and monitors those clouds, detects cloud failures and reacts by moving infrastructure elements from one cloud to another.

Table 1 Benefits of Public Clouds compared to traditional IT infrastructures and managed IT infrastructures

	Traditional private IT Infrastructure	Managed IT Infrastructure Outsourcing	Public Cloud Infrastructure Outsourcing
Scalability	Slow and Manual	Gradual and Manual	Instant and flexible, provided that mechanisms for rapid elasticity are available
Delay of Scaling	Weeks to Months	Days to Weeks	Minutes
Capital Investments	High upfront investments for ordering and deployment of IT infrastructure	No upfront investments, but usually mid-/long term contracts	No upfront investments, no long-term contracts required, pay-per-use pricing
Efficiency of Resource Utilization	Significant overprovisioning for coping with peak loads required	Significant overprovisioning for coping with peak loads required	Minimal overprovisioning, provided that efficient elastic scaling mechanism are available Cloud bursting for coping with peak-loads
Resource Usage	Once purchased, resources are either used or idle	Leasing of fixed / static capacities	On-demand renting and usage
Risk of false investments	High	Medium	Low

Elastic, multi-cloud brokering mechanisms provide all the benefits of public on-demand and pay-per-use cloud infrastructure outsourcing, *rapid elastic scalability* and additionally allow for the dynamic *selection of optimal cloud platforms and resources*.

This work takes the viewpoint of NGN service providers intending to utilize elastic and cost-efficient, multi-cloud brokering mechanisms for outsourcing their service infrastructure. As such, this work takes into account the specificities of NGN infrastructures, services and workloads. It provides answers to NGN service providers, asking:

1) How, and to which extend, can public cloud resources cost-efficiently be utilized for NGN service provision?
2) How, and to which extend can the QoS of NGN services be assured in multi-cloud environments?

Provided that above questions are answered to the satisfaction of NGN service providers, the subsequent questions to be answered would be:

1) How should multi-cloud brokering mechanisms be integrated with NGN SDPs?
2) How should multi-cloud brokering mechanisms be integrated with NGN OSS & BSS?

1.3 Problem Statement

Providing Web services over best-effort, public internet, as such already imposes certain risks regarding QoS and service reliability. NGN services 1) have specific QoS requirements, 2) are

particularly sensitive to network performance degradations and 3) require NGN service control platform interworking. Providing NGN services over public internet connections, imposes significant challenges on the NGN service management and cloud brokering systems.

Aiming for cost-efficient resource allocation, i.e. striving for optimal capacity savings (under varying workload conditions), furthermore imposes severe risks of QoS degradations, caused by potentially under-provisioned cloud resource capacities. Here, not only NGN service's sensitivity to over-load situations has to be mastered, but also the (at times) high volatility of typical NGN usage demand / peak-loads have to be mastered efficiently.

Furthermore, integration of cloud brokering systems into NGN service environments and NGN OSS and BSS platforms requires interworking with several components. Not only need NGN service control mechanisms to be provisioned dynamically, but also service lifecycle management mechanisms need to be extended for interworking with cloud-based NGN services, dynamically migrated across different cloud platforms.

1.4 Objectives

This section introduces most fundamental terms used in this thesis, provides a taxonomy of related areas and terms, introduces the fundamental hypotheses and research questions, defines and classifies the Key Performance Indicators (KPIs) against which the final approach is evaluated.

1.4.1 Definitions

In order to narrow down the field of research this thesis is about, the definitions for "*resource*" "*efficient*" "*NGN service*" "*quality management*" on "*federated*" "*cloud infrastructures*" are provided. Followed by a taxonomy which further narrows down topics addressed in this work in the field of "*cloud infrastructure federation*", this section introduces the most fundamental terms and areas of concern.

Definition of *Resource:*

> This work makes very specific use of the term "resource". In contrast to broad notions of "resources", which, could range from technology, business to human resources, this work focuses on virtualize-able resources, which can be used and consumed over network connections, particularly the internet. In this regard, the definition of resources in the context of this work is limited to, what the NIST definition of cloud computing [24] refers to, when speaking about
>
> *"[…] a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) […]"*[24].
>
> In the scope of this work, resources are primarily computing resources, which provide computation, storage and networking capacities. As the principles of efficient resource

allocation, analysed in this work, mainly are sensitive to resources' computation capacity, for initial ease of understanding, the reader is even advised to primarily focus on *computation capacity*. In this work, networking and storage capacities of a given resource can indeed become constraints for satisfactory service delivery and need to be monitored and allocated appropriately. Mechanisms for computation centric resource allocation mechanisms, discussed in this thesis, can, with some adaptations be also applied to dynamic scaling of storage and networking resources. This (efficient storage and networking resource allocation), and particularly the combination of all three (efficient computation, storage and networking capacity resource allocation mechanisms) is analysed only in medium depth, whereas the primary focus is on resources that provide computation capacity.

Definition of "*Service*":

From a high-level perspective the term "service", as defined by the Merriam Webster dictionary refers to "*1 a: the occupation or function of serving* […]" [34]. A more business oriented perspective [35], which, for the this work is more useful, particularly for modeling also the stakeholders (service consumer and service provider) describes services as "[…] *activities* […] *of a more or less intangible nature that normally* […] *take place in interactions between the customer and service employees and/or physical resources or goods and/or systems of the service provider, which are provided as solutions to customer problems*" [35].

In the following, two, more concrete definitions of a "service" of the IT service management and telecommunication / NGN service management world are provided. The first definition, from the IT Service Management Forum (isSMF), who are promoting the IT Infrastructure Library (ITIL) best practices, describe a "service" in ITIL Version 3 as "[…] *a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks*" [36]. The second service definition from the Telemanagement Forum's (TMForum) perspective, describe services in the enhanced Telecom Operations Map (eTOM) [37] as being "[…] *developed by a Service Provider for sale within Products. The same service may be included in multiple products, packaged differently, with different pricing, etc.* […]" [37] and further, in the TMForum Service Level Agreement (SLA) Management Handbook [38] "*A telecommunication service is a set of independent functions that are an integral part of one or more business processes. This functional set consists of the hardware and software components as well as the underlying communications medium. The Customer sees all of these components as an amalgamated unit.*" [38].

For the scope of this work, the TMForum's definition of a "service" is considered to be most valuable, as it is highlighting the hardware, software and underlying communications medium components of a given service. In this work, the hardware is provided in a virtualized form by the cloud provider, the software is the application provided by the telecommunication / NGN service provider and the communications medium is provided by a carrier. In the following, readers are advices to differentiate

between the actual *NGN service*, which can be deployed at the premises of the choice of the NGN service provider, and the *cloud service*, which in this work primarily relates to the cloud provider's offering of cloud infrastructure resources "as a service".

Definition of "*Efficiency*":

An "efficient operation" as the Merriam Webster dictionary defines it is "*b (1) :* […] *measured by a comparison of production with cost (as in energy, time, and money)* […]" [34].

For the scope of this work, the focus is put on costs, i.e. money spent for the usage of infrastructure resources. Being "efficient" with regards to allocated and consumed infrastructure resources, from the telecommunication / NGN service provider's perspective directly relates to cost savings. From a different angle however, saving resources has also a direct relationship to saving energy. This relationship however, is not investigated in depth in this work.

Definition of "Quality of Service"

The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) defines Quality of Service (QoS) as the "*Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.*" [39].

ITU-T recommendations regarding QoS initially mainly concentrated on *network performance levels,* where service quality related objectives are organized in *network QoS classes* [40]. The ITU-T therefore differentiates [41] *network performance related QoS criteria* from *non-network related QoS criteria*, both impacting the Service Quality Agreement (SQA), the service quality related sub-set of an SLA. For measuring the network related QoS classes, the ITU-T recommendations to a broad extend make use of the network performance metrics (e.g. one-way delay, round-trip delay, packet re-ordering, delay variation, etc.) standardized in the context of multiple Requests for Comments (RFCs) of the IP Performance Metrics (IPPM) Working Group of the Internet Engineering Task Force (IETF).

In the scope of this work, network performance parameters, (cloud-) platform and (cloud-) resource performance parameters and application performance parameters are differentiated, all contributing to the overall End-to-End quality of voice, video, data and call control / signalling telecommunication / NGN services.

Definition of "Service Quality Management":

TMForum positions *Service Quality Management* operations as part of the Assurance Operations vertical, defining the following seven operations, to be *part of Service Quality Management Operations*:

I. Monitor Service Quality
II. Create Service Performance Degradation Report
III. Analyse Service Quality
IV. Track and Manage Service Quality Performance Resolution
V. Improve Service Quality
VI. Close Service Performance Degradation Report
VII. Report Service Quality Performance

For the scope of this work, *Service Quality Management* involves *monitoring, analyzing, improving and reporting services performance*. The eTOM perspective, in the context of this work, is primarily useful for understanding the operations for realizing Service Quality related aspects of SLM, its relationship to Resource Performance Management, and SLA Management.

Definition of "Cloud Infrastructure":

When it comes to Cloud-related definitions, the National Institute of Standards and Technology (NIST) definition of cloud computing [24] is commonly widely used. It is also used as a basis and referred to by telecommunication standard development organizations such as the ITU-T, the Third Generation Partnership Program (3GPP), the European Telecommunications Standards Institute (ETSI) as well as the TMForum. NIST defines a *Cloud Infrastructure* to be a *"[…] collection of hardware and software that enables the five essential characteristics of cloud computing* […]" [24]. NIST further defines a Cloud Infrastructure to be comprised of a *physical layer* usually comprised of server, storage and network resources, and an *abstraction layer* consisting of software enabling the aforementioned five characteristics (I On-demand self-service, II Broad network access, III Resource pooling, IV Rapid elasticity, V Measured service, later described in more detail) [24]:

For the scope of this work, the most important characteristic of a Cloud Infrastructure is I. *on-demand self-service*, which NIST further defines as a capability through which a "[…] *consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.*" [24]

Whereas the essential cloud characteristics II and III, for this work, are considered to be a pre-requisite of a given Cloud Infrastructure, not further amplified, whereas characteristics IV. and V. are taken care of externally (by mechanisms and systems proposed in this work), meaning, *not required* by a Cloud Infrastructure to comply to and be part of the proposed approach.

Definition of "*Federation*":

Wikipedia defines federation in Information Technology (IT) to "[...] *multiple computing and/or network providers agreeing upon standards of operation in a collective fashion. The term may be used when describing the inter-operation of two distinct, formally disconnected, telecommunications networks that may have different internal structures* [...]" [42]

For the scope of this work, in line with the Wikipedia definition, multiple cloud infrastructure providers are agreeing upon standard interfaces enabling rapid cloud resource provisioning. By doing so, federations of multiple cloud infrastructures can dynamically be created and released and multiple services can be distributed across different cloud infrastructures. In order to prevent any ambiguity, this work uses the term "*federation*" in a broader sense compared to some related, lately published work [43], where the term "*federation*" is only used to define "*inter-cloud*" communication and the term "*multi-cloud*" is used for point-to-multi-point cloud computing mechanisms. In this work, the term "federation" is used for inter-cloud computing as well as multi-cloud computing mechanisms.

Finally also the difference between cloud infrastructure *federation* mechanisms and cloud *brokering* mechanisms needs to be clarified, as both cloud federation and cloud brokering make use of many identical mechanisms. According to standard cloud computing concepts, e.g. the NIST cloud computing reference architecture [44], cloud brokers provide services in the following three modes:

- *"Service Intermediation"*: Enhancement of available services, by improving specific capabilities
- *"Service Aggregation"*: Combination and Integration of multiple services into new services
- *"Service Arbitrage"*: Ability to dynamically select cloud services from multiple cloud service providers

If the term cloud brokerage is used to identify IaaS *service arbitrage* mechanisms, the overlap of mechanisms employed by both, *cloud federation* and *cloud brokering* systems is so ample, that the terms could almost be used in exchange. This work defines the following: Cloud Federation comprises the main mechanisms of Cloud Brokers for IaaS service arbitrage, plus mechanisms for cross platform (including NGN service control platform) service orchestration. The remainder of this thesis uses the term *Cloud Brokerage*, which encompasses both, the cross-domain federation (incl. NGN provisioning and orchestration) aspects as well as the brokering aspects of the mechanisms investigated in this work.

1.4.2 Taxonomy

Areas of concern, related aspects, involved actors, and KPIs related to Multi-Cloud Brokerage for NGN Services which will frequently re-occur in the course of this thesis are depicted in Figure 3.

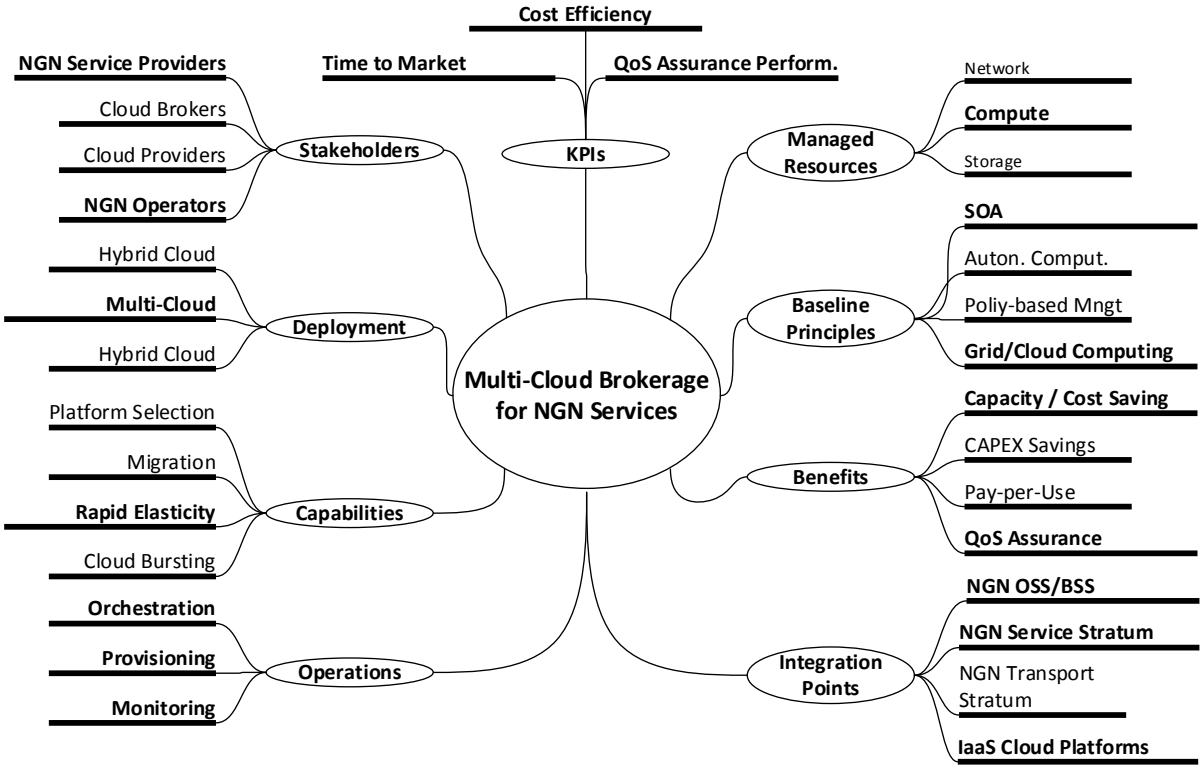

Figure 3: Cloud Infrastructure Federation Taxonomy

1.4.3 Hypothesis and Research Questions

This section provides the fundamental *hypotheses*, which this thesis tries to verify 1) by providing a short condensed statement and 2) by also outlining the rationale used for verification of the hypotheses. Subsequently the main *research questions* are given, which this work tries to answer.

Research hypothesis – short sentence

[By knowing a specific NGN service's resource capacity requirements, its QoS requirements and the relationship between these requirements]

a cost-efficient system,

brokering and scaling resources across multiple cloud infrastructures can be designed, build and integrated into standard-compliant NGN service and management environments,

that assures the specific QoS requirements of NGN services.

Research hypothesis – long sentence

In order to efficiently allocate distributed cloud infrastructure resources for quality assured NGN service delivery,

I. a *framework for cost-efficient* Multi-Cloud brokerage can be designed that allows defining *models and methodologies,* based on which

II. an according *system solution can be specified and instantiated*

III. which validates the overall concept and confirms usefulness and applicability and serves to *evaluate the performance* of such approach in emulated and real world scenarios providing credible data on

i. cost efficiency,

ii. quality assurance performance,

for realistic NGN workloads,

iii. by which potential cost savings can be related to QoS risks, existing approaches can be evaluated, and standards in the field can be improved.

Research Questions:

I. What are the requirements for resource efficient NGN QoS management in federated multi-cloud environments?

Here, the main contributions of this work are:

- an in-depth requirements analysis identifying requirements of the different actors involved, identifying technological needs, categorizing functional and non-functional requirements

a) How can cloud resources efficiently be allocated in federated, multi-cloud environments, satisfying realistic workload demands?

b) How can NGN QoS levels be assured in federated, multi-cloud environments?

Synthesis of a) and b):

II. How can cloud resources dynamically and efficiently be allocated in federated, multi-cloud environments, assuring QoS levels under realistic workload conditions?

Here the main contribution of this work is:

- the design of models for involved entities, their relationships and use-case scenarios
- the design of a *framework for QoS-aware Multi-Cloud Brokering for NGN Services (QOSMUC)*, based on state-of-the-art NGN management frameworks

- the design of information models and methodologies

III. How feasible is such an approach and what are the benefits, constraints and limitations of such an approach regarding *cloud resource allocation efficiency / capacity saving performance* under different QoS constraints under emulated and real-world conditions?

Here the main contribution of this work is:

- the instantiation of the QOSMUC framework, i.e the specification, implementation and deployment of *a Cloud Brokering system for NGN services* and its emulated and real-world scenarios
- validation of the overall concept, performance evaluation of the Cloud Brokering system through observational experimentation in emulated and real-world, multi-cloud scenarios under realistic workload conditions
- specification of performance evaluation criteria, definition of parameters for measuring Cloud Brokering performance and quantification of potential/ideal and achieved results

1.4.4 Classification of Key Performance Indicators

The performance of the Multi-Cloud Brokering system for NGN services is evaluated by defining KPIs, which provide quantifiable answers to the following two questions:

- How cost-efficiently is such a system able to allocate cloud resources for NGN services?
- How well is the system assuring the quality of NGN services?

For the following KPIs, metrics should be identified, against which the instantiated solution is evaluated, in emulated and real-world scenarios and under realistic workload conditions:

Resource Allocation Efficiency KPI:

I. A metric, capable of quantifying the Resource Allocation Efficiency will be defined for evaluating the performance of the approach.

QoS Assurance Performance KPI:

II. Standardized QoS classes for telecommunication / NGN services, typically used by the NGN service providing industry will be used for evaluating QoS assurance performance.

Synthesis: Resource Allocation Performance under QoS Constraints

> As there is a *direct relationship between Resource Allocation Efficiency and QoS Assurance Performance*, only the combined quotation of both KPIs provides meaningful information of the performance of the overall system. Additional *environment variable*s, such as workload characteristics, NGN service characteristics and Cloud resource characteristics are specified for providing the complete picture.

1.5 Scope of Thesis

This thesis combines NGN management principles and mechanisms with Cloud management mechanisms. It takes the viewpoint of NGN service providers outsourcing NGN service resources to public Cloud infrastructures.

Resource Allocation/Provisioning and QoS Management/Assurance processes find their place in both domains; in the NGN Management (ITU-T, NGN Management Principles [45]) as well as in the Cloud Service Management (NIST, Cloud Reference Architecture [44]) domain (as depicted in Figure 4). The approach taken in this thesis *brings Cloud Service Management processes to the realm of NGN resource and service management*, integrating Cloud Provisioning and Configuration processes into the NGN management plane.

As an intermediary, a Cloud Broker, according to NIST [44] providing "*Service Arbitrage*", i.e. "*the flexibility to choose services from multiple agencies*" handles the interworking of NGN management processes with *multiple* cloud infrastructures and resources.

Resulting Cloud-based NGN Service and Cloud-based NGN Resource management processes, in turn *need to integrate into the overall NGN Service Lifecycle Management*, which (lately with the help of SOA) became part of the NGN service stratum. Management of the NGN Service Lifecycle involves processes for 1) service creation, 2) resource provisioning, 3) service deployment, 4) service provisioning, 5) service operation and finally 6) the processes for service decommissioning.

Figure 4 depicts the overall scope of this thesis. In scope are the necessary functions for *Multi-Cloud Service Management*, i.e. multi-cloud NGN service monitoring functions, platform selection functions, resource allocation functions and service orchestration functions (i.e. NGN service control, cloud and service provisioning functions).

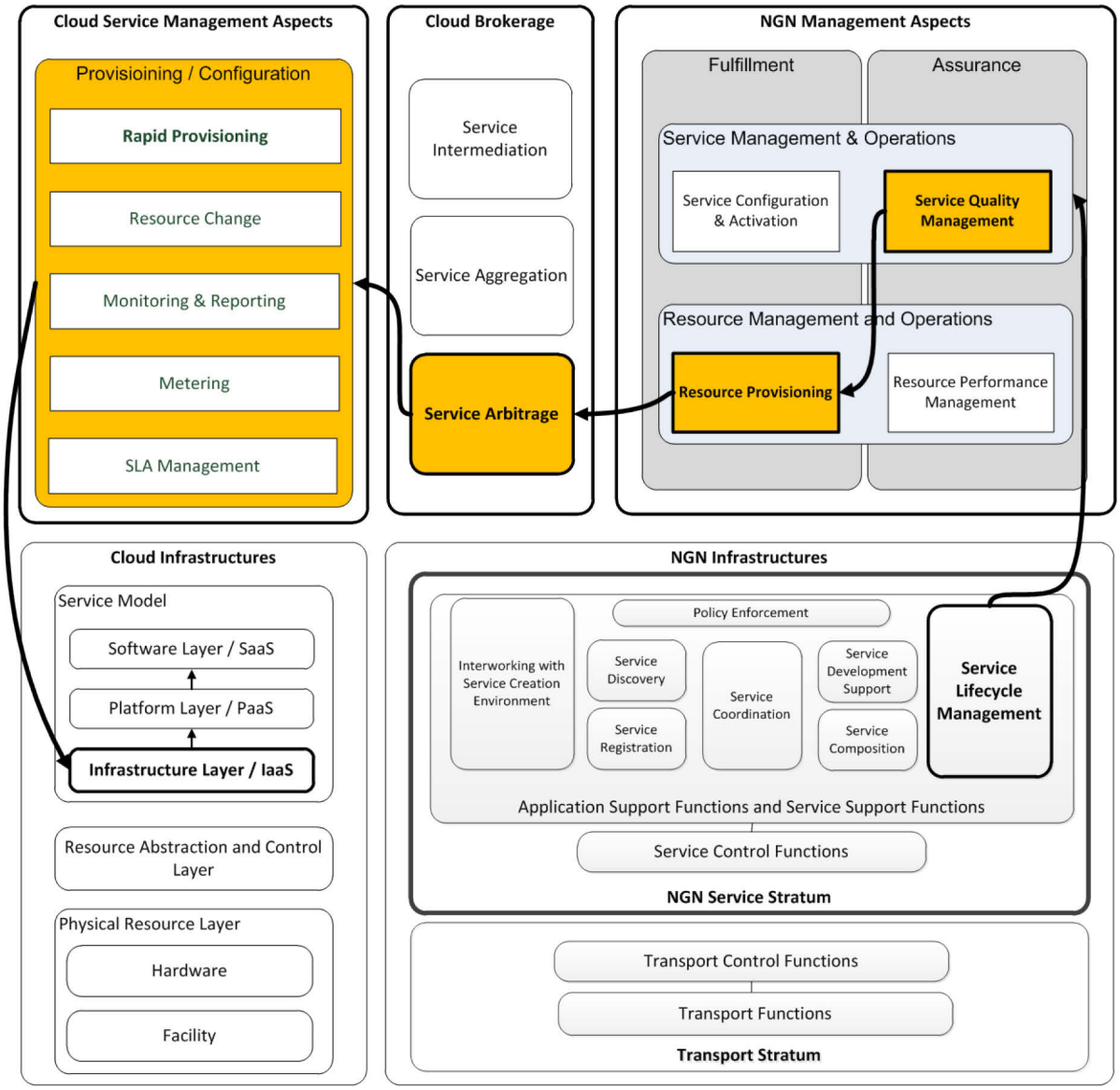

Figure 4: Overall Research Framework, Terminology: NIST Cloud [44], ITU-T NGN [46], TMF Mgmt [47]

1.6 Research Methodology

This section first classifies the scientific approach taken in this work, and subsequently describes the actual workflow of this thesis.

Scientific Classification

The scientific approach, taken in this work, falls in the field of *information system design-science research*. The workflow follows the so-called *regulative cycle* [48], which, applied to engineering research is also called *engineering cycle*, the general structure of a rational problem solving process [49], which typically starts with the investigation of a real-world problem, specifies solution designs, implements a selected design, evaluates the outcome of the implementation, which can be the start of a new iteration of the regulative cycle.

Information System Research framework

The fundamental research methodology applied in this thesis is based on Hevner's [50] Information Systems Research Framework. Shown in Figure 5, are environments, knowledge basis and the Information System research conducted in the scope of this thesis.

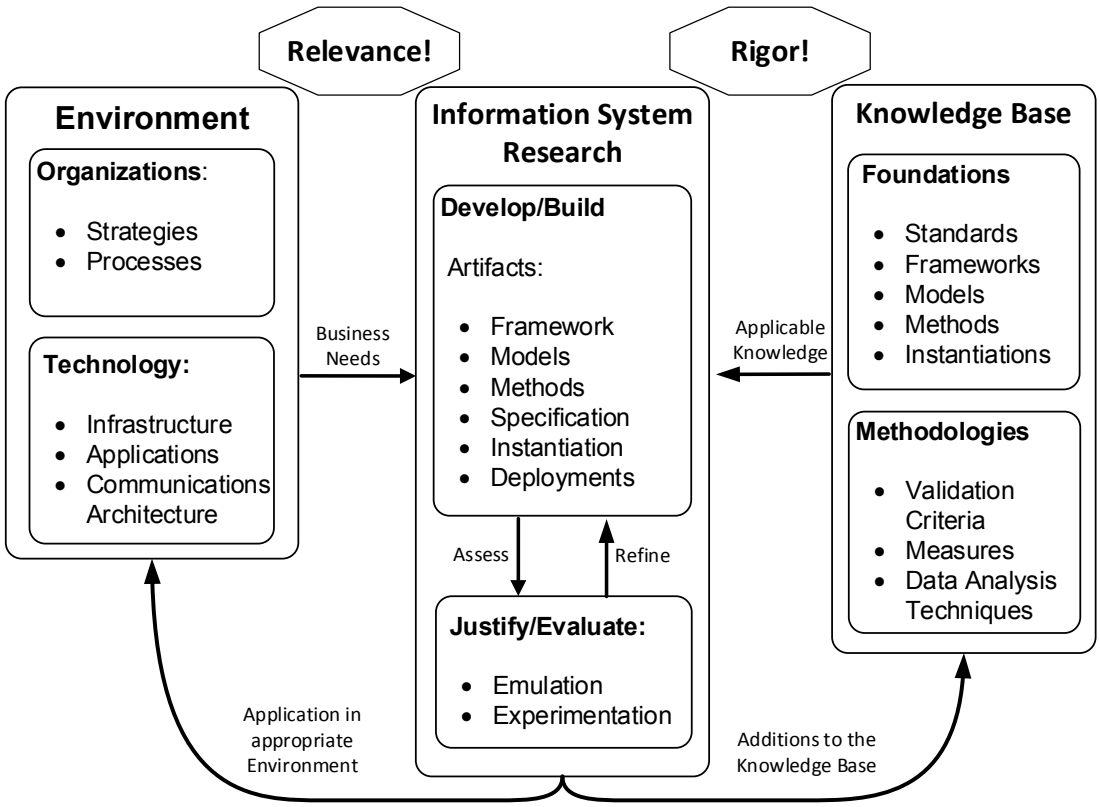

Figure 5: This thesis mapped to Hevner's [50] Information systems Research Framework mapped

Whereas the overall workflow of this thesis, also referring to this research methodology is explained in 1.8, Table 2 briefly outlines how Hevner's research framework maps to this thesis.

Table 2 Application of Research Framework to the Areas of Concern of this thesis

Environment	
Organizations:	Focus: NGN Service Providers Related: NGN Network Operators, Cloud Providers
Strategies	NGN Service Provider's Strategies
Processes	NGN Service Provider's Service Environment Processes
Technology:	Focus: NGN Service Environment and NGN Service Management Related: NGN Control Platforms, Cloud Platforms
Infrastructure	NGNs, IaaS Clouds
Applications	NGN Services
Communications Architecture	NGN Service Environment Architectures, NGN Management Architectures, Cloud Architectures

Information System Research	
Development of Artifacts	
Framework	QoS-aware Multi-Cloud Brokering for NGN Services (QOSMUC) Framework
Models	NGN Services, Cloud Resources, Policies
Methods	Orchestration of NGN Service Lifecycle Management Resource Allocation federated Cloud Environments
Specification	NGN Service Cloud Broker
Instantiation	NGN Service Cloud Broker, Elasticity Engine
Deployments	Multi-Cloud Testbed Environments (local, pan-European), IMS, EPC, M2M Testbeds
Justify / Evaluate	
Emulation	Local Cloud Infrastructure Testing and Evaluation
Experimentation	Real-World Multi-Cloud Testing and Evaluation

Knowledge Base	
Foundations	
Standards	Technological Paradigms, NGN, Cloud Standards
Frameworks	NGN Management Frameworks (TeleManagement Forum's NGOSS/SES, SID eTOM)
Models	Resources, Services, Processes (SID, eTOM)
Methods	Service Lifecycle Management Orchestration Autonomic Control Loops
Instantiations	Basis: NGN Service Broker Components: NGN Service Control Platform, Monitoring System, Provisioning System, Service Orchestration Engine, NGN Service Environment, Policy Engine, Cloud Infrastructure Management System
Methodologies	
Validation Criteria	Resource Allocation Efficiency QoS Assurance Performance
Measures	Capacity Saving Efficiency (CSE), Overprovisioning Factor (OPF) NGN QoS Service Classes (ITU-T)
Data Analysis Techniques	Linear time-series analysis and Models, Regression Analysis

Relevance	NGN Service Industry's needs for Multi-Cloud NGN Service Management and Brokerage
Rigor	Realism of Workload, Environments, Systems and Services

1.7 Major Contributions

The author's work on NGN management principles and mechanisms profoundly shaped the design and instantiation of the NGN Cloud Brokering system described in this thesis. In designing NGN management systems, the author systematically made use of advanced mechanisms and design principles fostering heightened degrees of automation, configurability and manageability, for identical reasons governing the design of the NGN Cloud Broker, i.e. shorter time-to-market of NGN services / shorter service management lifecycles, reduction of operational overhead of NGN management systems.

Throughout the author's work on NGN service management principles and platforms, high emphasis is put on SOA principles allowing for seamless integration of NGN services into converged NGN service delivery environments and NGN management environments and for dynamic and adaptive NGN service composition, orchestration and brokerage.

Of similar importance to the majority of the author's contributions are policy-based management mechanisms allowing for increased adaption of management systems to changes (at initialization time / for re-configuration, as well as during operations runtime).

The use of autonomic management principles (like the MAPE approach taken in this thesis), similarly shaped the author's work, where NGN monitoring mechanisms are integrated with dynamic SOA-based service orchestration mechanisms (e.g. service provisioning, dynamic service re-composition) exploiting the benefits of policy-based management approaches.

For supporting the goal of increased NGN service compose-ability the author, studied the benefits of Model Driven Engineering (MDE), semantic service annotation and automated service composition for contributing to the Autonomic Communication Forum ACF [6].

Throughout his work on NGN service management mechanisms, the author maintained and published the current state-of-the-art of NGN service control platforms [51], NGN services [52] and service delivery platforms [7].

In the context of NGN management frameworks, focusing on NGN Operation Support Systems, the author developed SOA-based NGN management mechanisms applying them for automating service provisioning and fault management mechanisms for IMS [10], [11] and SDP [12], NGN fault localization [13], automated management of NGN service compositions [14]. The author's main proposition of bringing SOA principles to the domain of NGN service management [15] and SOA-based integration of NGN management services into the SOA-based service delivery environment [16], relates to the enablement of higher degrees of automation of the service management lifecycle (i.e. plug and play principles allowing automated composition, deployment and management of composite NGN services, as shown in [17]).

In the context of SOA-based NGN service orchestration, the author contributed with innovative mechanisms for NGN service brokerage [1] and NGN service delivery [2], [3], modular NGN service exposure [4], including cross-domain service orchestration mechanisms through infrastructure federation mechanisms [2], [4], [5].

Through application and integration of aforementioned SOA-based NGN service management principles, the author led the design and implementation of the SOA- and policy-based NGN management system OMACO (OSIMS Management Console) [13], [12], [14], [15] and significantly contributed to the NGN/OMA-standard based design and development of SOA-based NGN service orchestration and brokering mechanisms [3], [8], [2], [9], [4], [1] leading to the implementation and evaluation of NGN service brokering systems such as the

eXtended POlicy based Semantically enabled sErvice bRoker (XPOSER), later called FOKUS Broker. OMACO and XPOSER can be regarded as the predecessors of the NGN Cloud Broker described in this work.

Aforementioned works of the author were subsequently exploited for integrating cloud computing mechanisms into NGN service management and service delivery frameworks. With a focus on multi-cloud service brokering mechanisms, allowing flexible outsourcing of NGN service resources to external cloud platforms, the author investigates management mechanism for multi-cloud, IMS-based NGN QoS assurance [18], highlighting the importance of network performance awareness and dynamic cloud platform selection [19] (nominated for best paper award IEEE CLOUDNET'12), combining QoS assurance with elastic multi-cloud resource provisioning / auto-scaling mechanisms for NGN services [20], [21], with resource allocation efficiency / capacity savings evaluated in [22].

Aforementioned work and publications are reflected in this thesis, in the context of the state-of-the-art (NGN management, NGN service brokerage), the design of the QOSMUC framework, the instantiation and especially the evaluation of the NGN Cloud Broker.

Artifacts, designed, specified and instantiated based on the QOSMUC framework (Chapter 6) and the knowledge obtained by instantiating (Chapter 8.3) and evaluating (Chapter 9) the NGN Cloud Broker was exploited in several dimensions, ranging from sustained deployments in operational testbed environments, integrations in commercial solutions, utilizations and further enhancements in research and industry projects, for standardization and for dissemination and educational purposes. Utilization of QOSMUC artifacts in appropriate Environments, Sustainability, Exploitation and Integrations are summarized in Table 3, extensive details are provided in *Appendix I: Author's Dissemination*.

Table 3 QOSMUC artifacts deployments, integrations, project exploitations, dissemination and standardization

Deployments, Federation and Integration	• Cloud Federation facility: Pan-European multi-cloud facility for Internet of Service experimentation "BonFIRE" • European Testbed Federations: Pan-European commercial Testbed-as-a-Service facility "FanTaaStic"– as part of the FUSECO Playground, Pan-European Future Internet facility for Research and experimentation "Fed4FIRE" – as part of the FUSECO Playground • Research and Development Laboratories: FUSECO Playground, Open SOA Telco Playground
Exploitation in Projects	• Research Projects: NUBOMEDIA, BonFIRE, Fed4FIRE, FanTaaStic, MAMSplus • Industry Projects: NTT, Deutsche Telekom
Integration into Commercial Platforms	• NGN Service Broker: FOKUS Broker • NFV/SDN Platform: OpenSDNCore • Machine-type Communication Platform: OpenMTC
Additions to the knowledge base, dissemination of QOSMUC artifacts	• Books (1) • Journals / Book Chapters (7) • Publications (18) • Tutorials (4) • Master Theses Supervision (4) • Workshop/Conference Organization (26), i.e. Chair (2), TPC-Co Chair (2), TPC Member (6), Reviewer (16) • Conference Presentations (many)
Standardization	• IEEE P2303 InterCloud Testbed

1.8 Structure and Workflow of Thesis

The overall workflow and structure of this thesis is depicted in Figure 6.

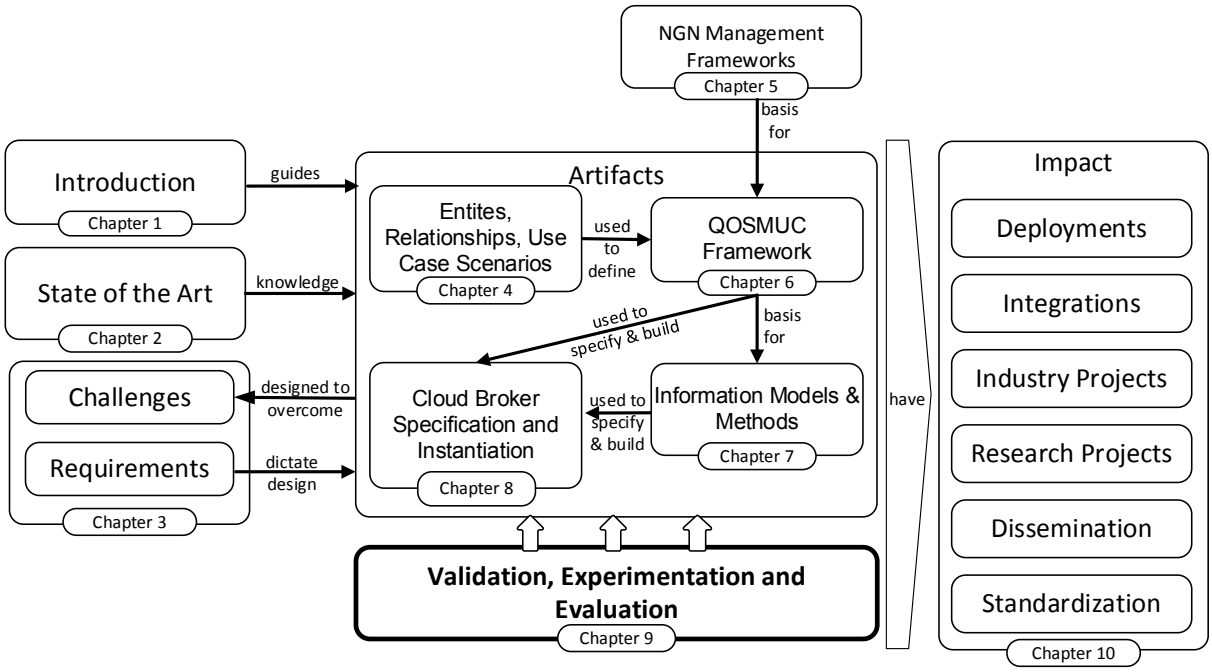

Figure 6: Structure and Workflow of Thesis

In *Chapter 1*, following standard motivation, relevance, hypothesis, research questions, important definitions, major contributions, as well as the overall scientific approach following standard design science guidelines [50] are introduced.

In *Chapter 2*, an in-depth, state-of-the-art analysis addresses fundamental principles followed in this thesis, current NGN and Cloud technologies, providing the fundamental knowledge for designing the artifacts produced in this work.

In *Chapter 3*, the main challenges and requirements for overcoming those challenges differentiating the viewpoints of the different actors/administrative domains are analyzed.

In *Chapter 4*, actors/entities, their roles, relationships, modes of action and use-case scenarios are explained.

In *Chapter 5*, de-facto standard frameworks (processes and information models) for telecommunication enterprise management (Products, services, resources, suppliers) widely used in the telecommunication industry are introduced and relevant areas identified.

In *Chapter 6*, a **Framework for QoS-aware Multi-Cloud Brokering for NGN Services (QOSMUC)** is established, combining 1) a framework for managing the lifecycle of NGN

services in federated cloud environments and 2) a framework for resource allocation control for the actual service operation phase.

In *Chapter 7*, these frameworks are subsequently used for modeling the information of NGN services, Cloud resources and policies as well as for establishing methodologies for 1) the orchestration of the processes required for the management of the service lifecycle as well as 2) for the allocation of resources in federated cloud environments.

In *Chapter 8*, based on the QOSMUC framework (chapter 6) and information models and methodologies (chapter 7), the actual **specification and instantiation of the Cloud Broker for NGN services** is provided. Whereas the architecture of the NGN Cloud Broker is provided in the initial sections, the chapter proceeds in providing the actual implementation of the core cloud brokering and monitoring system and the integration into NGN service platforms and Cloud management platforms.

In *Chapter 9*, the overall system is validated through deployment and experimentation in multiple scenarios. An **in-depth performance evaluation of the Cloud Brokering system** in emulated and real-world, multi-cloud environments under realistic workload conditions is conducted. Measures and metrics for assessing the capacity saving performance of the Cloud Brokering system are established, against which the Cloud Brokering system is evaluated. As such, the performance evaluation represents **the most substantial contribution of this thesis**, where not only the performance of the Cloud Broker is evaluated, but also its design aspects are evaluated, performance impacting factors are evaluated and a comparative evaluation against related approaches is carried out. Based on the findings provided in this chapter, the reader will be able to understand the potential gains, the limitations and constraints of resource efficient QoS management for NGN services in federated cloud environments and enabled for making educated decisions between different technological options.

In *Chapter 10*, the core findings of this work are summarized, the produces impact of this work, in terms of sustainability of results (deployments, integrations), exploitation (in industry and research projects), as well as dissemination and standardization activities are described.

Chapter 2

State of the Art

The evolution of information and communication technologies continuously paves the way for convergence of networks, infrastructures, platforms and services. Through fixed-mobile convergence and NGN technologies, SOA principles, virtualization technologies and cloud computing mechanisms, convergence across all layers is becoming reality.

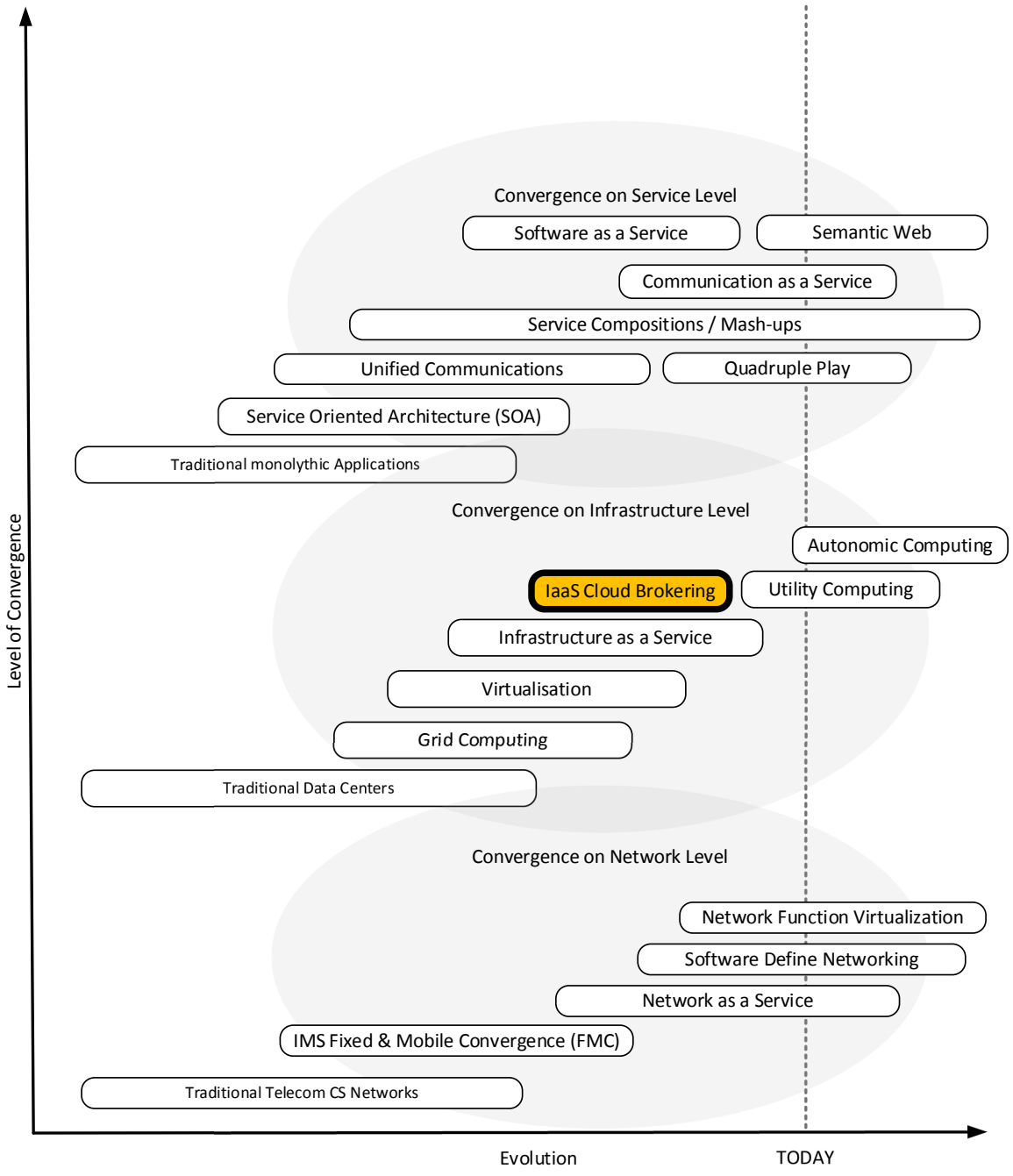

Figure 7: Evolution of Network, Infrastructure and Service Convergence

As shown in Figure 7, service orientation applied to network, infrastructure and service levels more and more allows provision and consumption of network, infrastructure and software resources as a service. All-IP-based NGNs paved the way for full convergences at access network level (mobile, fixed, cable, satellite) and provisioning of a new spectrum of network-agnostic feature-rich communication services as for instance Internet Protocol Multimedia Subsystem (IMS) - based Rich Communication Services (RCS) or IMS-based Internet Protocol Television (IPTV). NGN service platforms, by applying SOA principles allow not only for feature-richness (e.g. “quadruple play service”, a combination of fixed, mobile telephony, and television services), but also for high degrees of service re-usability through utilization of re-usable service enablers and ever shorter time-to-market through SOA-based service composition mechanisms. With the break-through of virtualization technologies (including Network Function Virtualization (NFV)) and the subsequent offspring of cloud-based infrastructures, platforms and services, full service orientation at all levels is at the brink of becoming full reality. Unprecedented levels of flexibility and compose-ability will soon allow for fully distributed provision of multi-domain services and compute, storage and networking (functions) provided and consumed dynamically and on-demand (i.e. as a utility).

From this high level perspective, this work focuses on the exploitation of cloud-based infrastructures provided as a service (i.e. IaaS clouds) for cost-efficient provision of NGN services. Thus, generally speaking, this work intends to contribute to the convergence of Cloud and NGN technologies from a Cloud infrastructure and NGN service perspective. Taking the viewpoint of NGNs, this approach investigates how NGN management frameworks can be applied to the management of NGN services that are provided on multiple cloud infrastructures. This chapter, as shown in Figure 8, introduces NGN principles, the basics of NGN service control, NGN service delivery and NGN management, cloud computing principles, cloud service management and cloud brokerage. The initial sections of this chapter, however, introduce the fundamental technological paradigms and principles needed for understanding the actual design choices taken in course of this work, but to some extend also for understanding the state of the art of NGNs and Clouds.

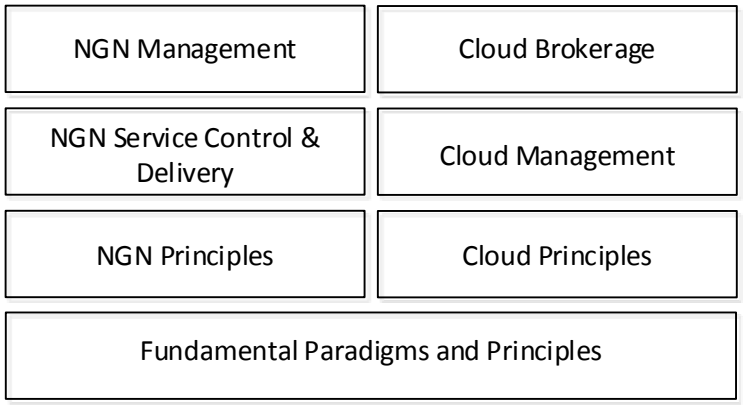

Figure 8: Technological Areas of Concern

2.1 Fundamental Paradigms and Principles

This section outlines the most fundamental paradigms and principles, which not only heavily influence the current state-of-the-art of NGNs, NGN management and Cloud Computing (described subsequently), but also the design of this work's frameworks, methodologies and instantiations.

2.1.1 Service oriented Architectures

According to the Organization for the Advancement of Structured Information Standards (OASIS) Reference Model for Service Oriented Architectures [53], SOA is a "*paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.*"

Above mentioned "*uniform means*" are typically realized by common, SOA-based interfaces and protocols, and methods which allow for standard service offering / publishing, service discovery, and service invocation and consumption, as depicted in Figure 9. In the case of unitary service utilization, this allows for flexible and dynamic service discovery and consumption. Done sequentially, SOAs allow for flexible and versatile service chaining, rapid composition of (business) processes, across multiple administrative domains, increasing the re-use of a single service component/building-block (which can be offered and utilized as an element of multiple service compositions).

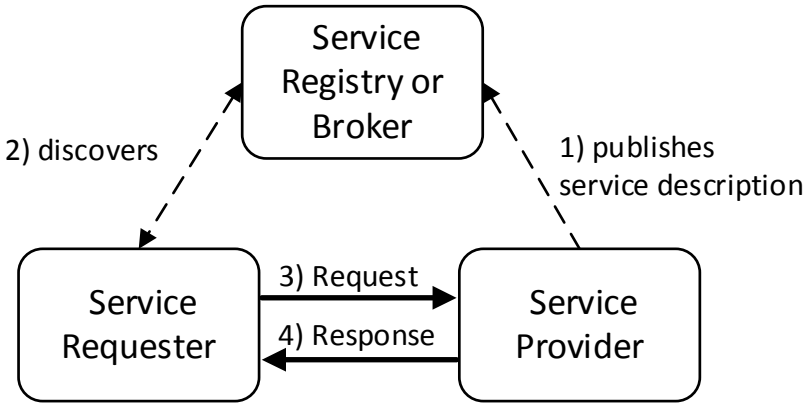

Figure 9: SOA-based, Service Publishing, Discovery, Invocation and Consumption (bsd. on [54])

Typically, for realizing a SOA, the following service-orientation design principles are adhered to (summarized from [55]):

- *Standardized Service Contracts*

  A service's public technical interface should be designed in a way that expresses the service's functionality, data types, data models and how policies are asserted.

- *Service Loose Coupling*

  The dependencies of a service or service composition on its service elements, its surrounding environment and its consumers should be kept as low as possible, promoting the independent design and evolution of its logic and implementation.

- *Service Abstraction*

  The principle of service abstraction promotes hiding of a service's underlying details, and finding the right level of granularity so as to serve multiple consumers or compositions.

- *Service Reusability*

  The service-oriented design should maximize the reusability of service's logic for enabling its shared utilization in multiple contexts and for multiple purposes.

- *Service Autonomy*

  Through mechanisms of isolation, interferences with / influences of other services (utilizing the same, share service environment and resources) a high level of autonomy should be achieved in order to increase a service's behavioural predictability.

- *Service Statelessness*

  A high degree of statelessness should be sought, in order to increase scalability potential, as well as availability.

- *Service Discoverability*

  In order to foster a service's utilization / re-use, supplemented meta-data of a service, should support effective discoverability.

- *Service Compose-ability*

  Also in order to increase the re-use of a particular service, i.e. its utilization in other service compositions should be foreseen at design time, allowing for later interoperability with other service building blocks (with compositions of services).

SOA-based Service Orchestration and Brokerage

This brings us to a core aspect of SOAs, which has strong influences on NGN service environment/platforms, NGN management platforms and Cloud platforms, where the *orchestration* of multiple services, their *compose-ability* within various workflows, and their *brokerage* within and *across multiple domains* are of fundamental importance.

Versatile SOA-based service orchestration mechanisms 1) significantly reduce effort and time for deployment and integration into service environments, 2) allow for dynamic integration into service management (OSS, BSS) environments, 3) enable value-added compositions of workflows across domains, 4) enable dynamic integration of external service building blocks into service compositions / and business process workflows.

As shown in Figure 10, with an example of a Web Service (SOAP, REST) realization of a SOA, and a BPEL realization for service orchestration, a service, consumed via a single SOA-based Web Service interface, is comprised of multiple service components/elements orchestrated/composed by a service orchestration function potentially utilizing service elements from different administrative domains.

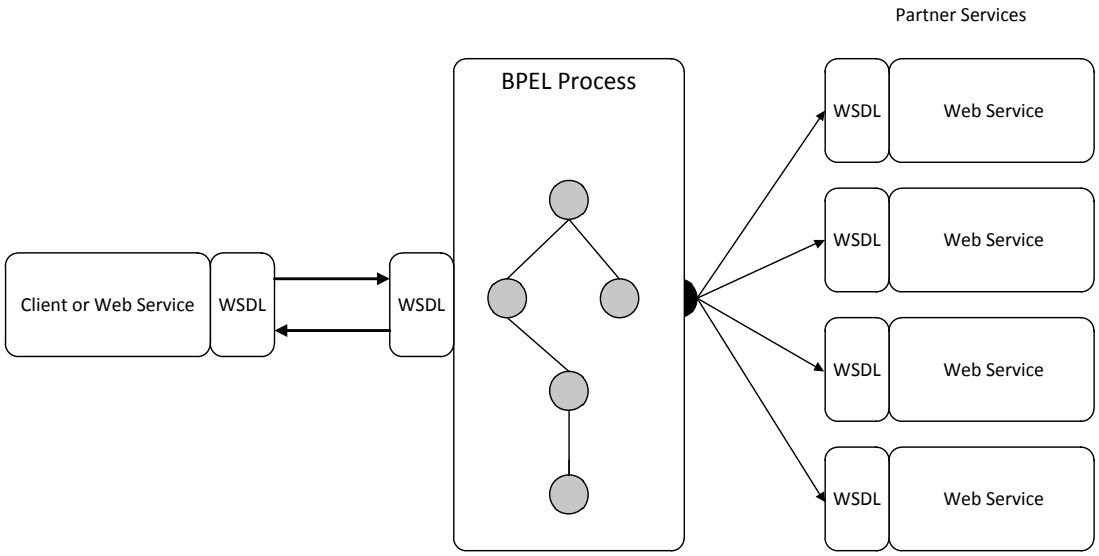

Figure 10: SOA-based, Web Service Composition/Orchestration

There is a broad spectrum of technologies, which have evolved by utilizing SOA principles, for realizing agile service delivery, management and composition. A huge research field is concerned with improving service compose-ability by means of automated service discovery and service composition. Usually by utilization of semantic registries/service descriptions/meta-data, reasoning algorithms and engines, policy/rule engines, and orchestration engines, services compositions are dynamically assembled, re-composed and exposed and consumed.

Several of the author's initial works apply SOA principles, and policy-/SOA-based service orchestration mechanisms, to the field of NGN service platforms (SDPs) (exploiting SOA principles for NGN service brokerage [1], NGN service delivery [2], [3], modular NGN service exposure [4], cross-domain service orchestration and infrastructure federation mechanisms [2], [4], [5]).

Similarly, the author applied SOA mechanisms to the field of NGN service management (OSS, BSS) fostering the SOA-based automation of service provisioning and fault management mechanisms for IMS [10], [11] and SDP [12], utilizing SOA principles for NGN fault localization [13], for NGN service management [15], for the management of NGN service compositions [14], and for SOA-based integration of NGN management operations into the SOA-based service delivery environment [16]. Using SOA principles, the author developed means for automating the NGN service management lifecycle (i.e. plug and play principles allowing automated composition, deployment and management of composite NGN services, as published in [17]

The author particularly advances in the field SOA-based NGN service brokering, and NGN service management, which lay the fundamental basis and provide fundamental components for the work described in this thesis.

2.1.2 Policy-based Management

Initially introduced mainly for the management of networking resources and service [56], policy-based management principles and approaches were soon also applied to the application layer / service environments and platforms [57]. Policy-based architectures typically comprise the elements, depicted in Figure 11. Typically, a tool for managing policies allows system administrators and/or external applications to provision and update system-wide policies in a policy registry. Typically during runtime, current system-wide monitoring data is aggregated and made available at policy decision point (PDP). By retrieving policy rules from the policy registry and by analyzing current system states (frequently realized by a policy evaluation engine), the PDP commands the Policy Enforcement Point to apply and execute specific policies. The PDP and PEP can be realized as two separate entities [58] or, as in [59], combinedly called "*policy consumer*". Finally the PEP applies policies and executes commands towards one or several "*policy targets*", i.e. systems to be provisioned.

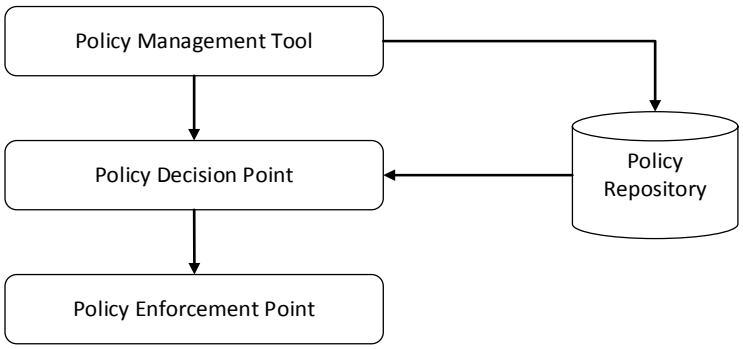

Figure 11: IETF/DMTF Policy Framework, based on [58]

A broad range of policy models and policy languages, generic as well as domain-specific, network oriented, as well as application oriented (e.g. the IETF policy core information model [60], an extension of DMTF's Common Information Model CIM [61]), PONDER [62], Policy Description Languages([63]), Policy Management for semantic Web Services KAoS [64], the W3C Web Service Policy Framework [65].

For the *telecommunication management domain* (OSS, BSS), also policy-extensions for TMForum's SID have been produced [66], section 2.4.

For the *NGN service layer*, the OMA specified as a core component of the OMA Service Environment (OSE), the Policy Evaluation, Enforcement Management (PEEM), which defines its policy expression language in [67] (utilizing the IETF model as policy rule set and BPEL-based policy expression language for business processes [68]). This will become relevant in section 2.3.3 NGN service environments. Especially in combination with SOA principles, the author exploited policy-based management mechanisms for autonomic NGN service composability [6]

2.1.3 Autonomic Computing

The initially postulated vision of autonomic computing, provided already in 2003 by IBM [69], since then influenced a broad range of IT and ICT application areas, particularly areas related to network, resource and service management. By exploitation of autonomic computing principles, typically known as the "Self-*" mechanisms manage-ability of complex systems (e.g. IT enterprise infrastructures and services, telecommunication infrastructures and services) is aimed to be improved. By applying Self-Configuration/-Optimization,-Healing, and –Protection mechanisms [69] systems are aimed to be made more efficient, robust against failures, and manage-able. Between basic, manual management/administration IT/ICT systems to fully autonomic management, [70] defines "*managed*", "*predictive*" and "*adaptive*" as intermediary levels of a system's "autonomicity", stating that with each higher level of "autonomicity", the *time to fix problems* is reduced, a *system's availability* is increased, *service level agreements* can be attained, *customer satisfaction* is increased/assured, *business goals* can be achieved/assured, and finally the *business responsiveness* to market-changes/changes in demand can be improved.

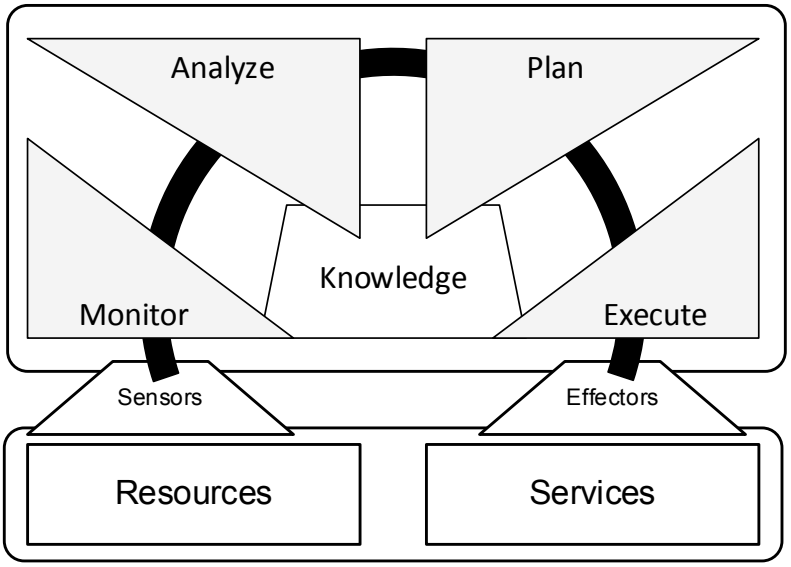

Figure 12: Monitor, Analyze, Plan, Execute MAPE - IBM view on autonomic computing [71]

Close Feedback loops are typical cornerstones for realizing Self-* autonomic computing mechanisms. Derivatives of the Monitor-Analyze-Plan-Execute (MAPE) approach (introduced by IBM [71], shown in Figure 12), such as ITIL's [72] closed loop for managing resource and service capacities, shown in Figure 13. mentions the benefit of closed-loop feedback systems referring to the ISO/IEC 20000 [73] standard for IT service management, which highlights the Plan–Do–Check–Act (PDCA) management method.

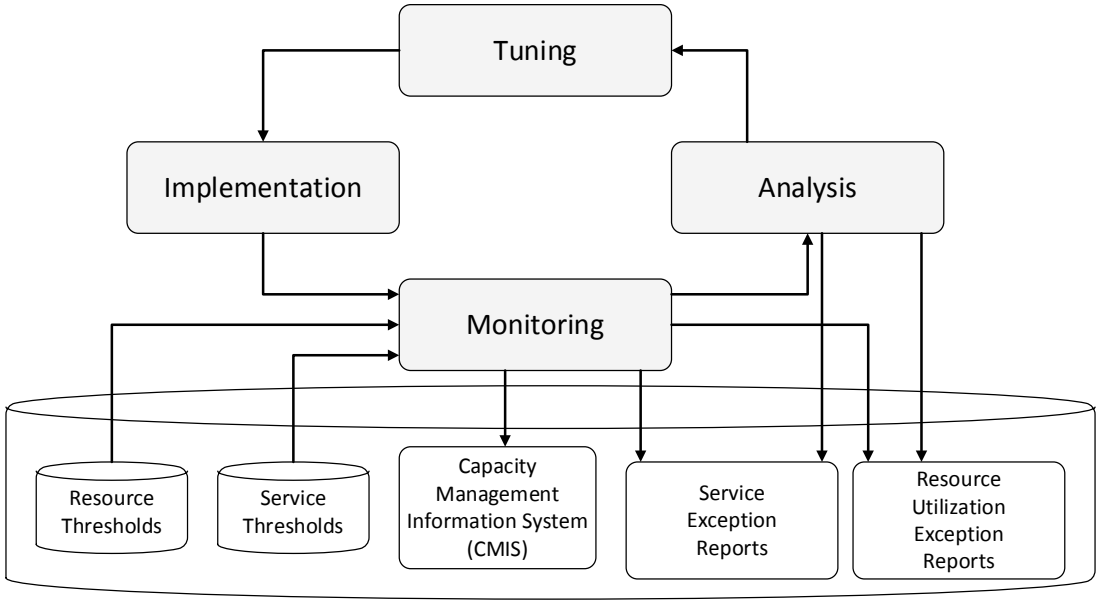

Figure 13: Resource, Service Capacity Management Cycle in ITIL [72]

Already, autonomic computing mechanisms are used, not only for managing IT infrastructures [72] and services, but also for managing cloud infrastructures [74] and cloud-based services, where several challenges [75] need to be overcome. Already architectures for autonomic service management in cloud-based systems [76] and autonomic mechanisms for detecting SLA violations of cloud-based services [77] are being proposed.

This work is focusing on autonomic mechanisms for cloud platform selection and cloud resource allocation for NGN services. This work designs, instantiates and evaluates a Multi-Cloud Broker, which is comprised of a multi-cloud resource allocation controller, selecting optimal platforms and scaling cloud resources according to pre-defined policies for serving variable workloads. Autonomic principles, combine with Model-Driven Engineering (MDE) and policy-based management pave the way for this thesis and allow for high levels of service integration, composition and operation [6].

2.2 NGN Principles

Telecommunication operators are currently migrating their infrastructures and services from legacy circuit switched networks and intelligent network-based services towards NGNs. Although there are many more aspects, which are typically associated with the term "Next Generation Network", key for understanding this work are NGN's "All-IPness", i.e. packet-based signaling between systems and services and the decoupling of transport and service layer. The ITU-T's general overview of NGN [78] defines an NGN as *"[...] a packet-based network able to provide Telecommunication Services to users and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent of the underlying transport-related technologies.*"

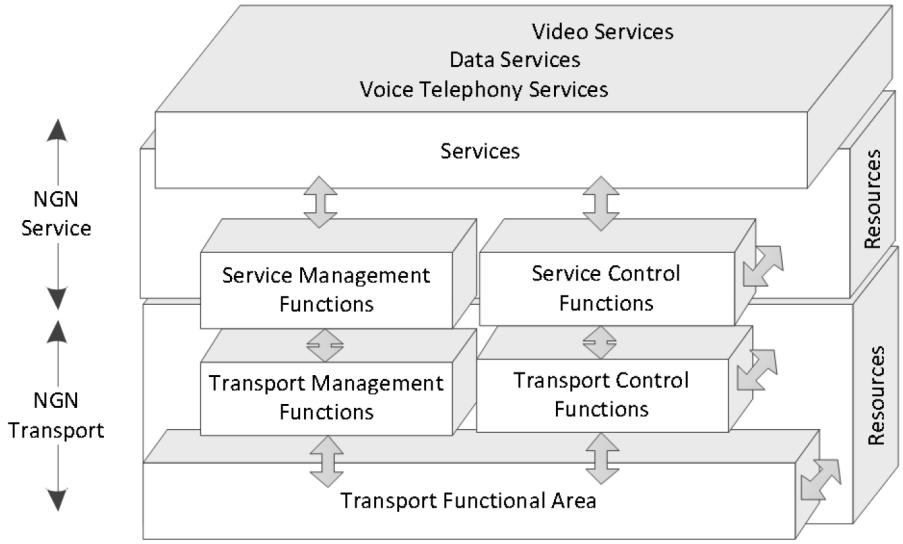

Figure 14: ITU-T General Functional NGN Model [79]

This clear separation between transport and service technologies, as depicted in Figure 14, allows different constellations between telecommunication transport infrastructures and service infrastructures. Whereas a tight coupling and co-located deployment of transport and service technologies was and still is widely common, with increased performance and reliability of networks, a more loosely coupled constellation became feasible.

What is usually called the "core network" or "network core", which comprises (apart from authentication, charging functions), important service control (incl. service invocation)

functions, is usually still located at the network operator premises, close to the transport network, mainly due to significantly higher constraints regarding signaling latencies. This however can be regarded as a transient phase, as there is significant work currently being carried out, to further decouple service control functions from the transport layer. In contrast to the service control functions, the application functions, including all functions typically associated with a Service Delivery Platform (SDP) can in principle already be deployed in a loosely coupled, spatially distributed fashion, as long as certain, mainly QoS-related conditions are being met.

Therefore, although the service stratum comprises both, the service control as well as the application functions, this work starts from the premise that service *control* functions are located in close vicinity to the transport network, whereas application functions can be deployed in a distributed fashion.

Another important aspect depicted in Figure 14 is related to the decoupling of functions and resources, which, for the context of this work is also of utmost importance. As the entire service stratum, including the service control functions are software-only systems, full virtualization, i.e. full decoupling of function and resource became possible.

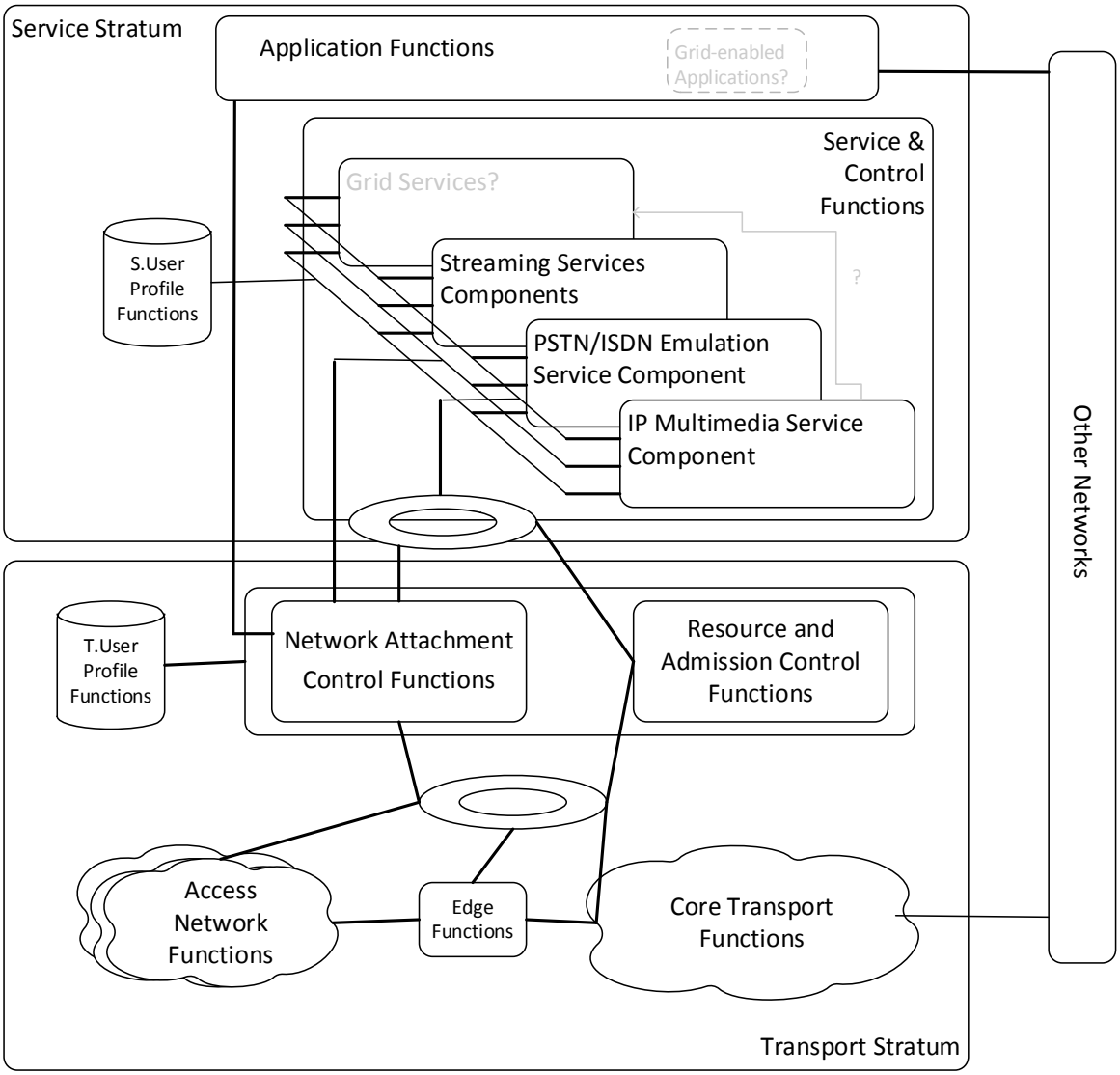

Figure 15: ETSI NGN Architecture bsd on [80], extended bsd on [81]

The NGN Architecture described in NGN Release 1 by ETSI TISPAN [80], complies to this strict separation of transport and service stratum, as shown in Figure 15.

Over the time not only the IMS session/service control functions entered the service stratum, but also IPTV, PSTN emulation control functions, and (under discussion [81]) also Grid/Cloud control functions.

2.3 NGN Service Control and Service Delivery

Whereas during the initial emergence of NGNs, only little focus was put on the NGN service delivery environment, with increasing numbers of NGN services the need for flexible service environments became immanent. A more detailed view of NGN layers and functions, particularly highlighting the NGN service stratum, as provided by ITU-T [46] is depicted in Figure 16.

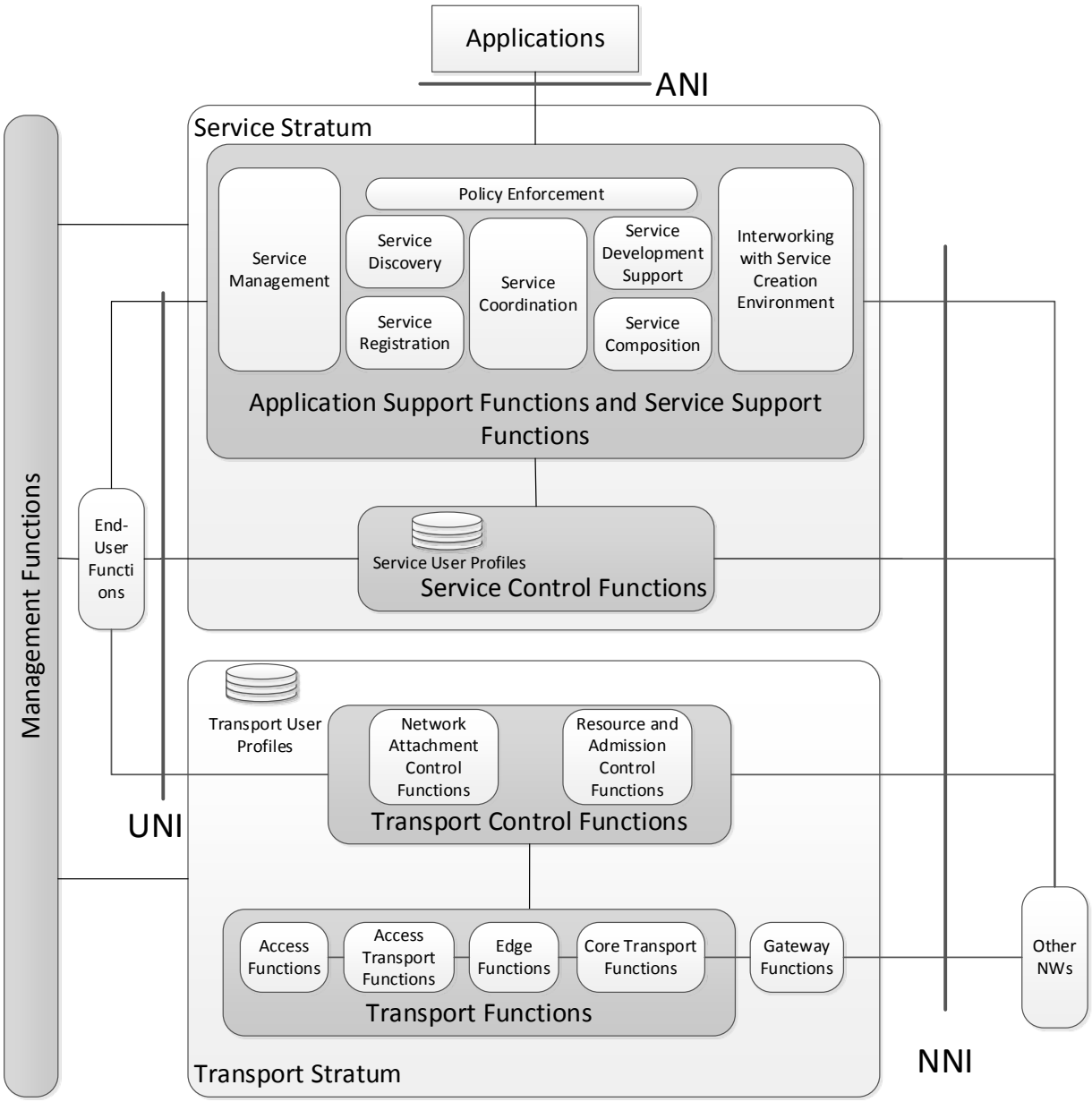

Figure 16: ITU-T NGN Architecture with Open Service Environment Capabilities [46]

The service stratum is significantly enhanced by several application and service support functions. Whereas a policy enforcement function controls external access to the open NGN service stratum, a service coordination function coordinates invocation of and communication between services. Dedicated service creation, development support and service composition functions are used to maximize re-use of existing services and to significantly speed up new service's time to market. Dedicated service management functions are employed to complement already existing cross-cutting management functions. As will be explained later in more detail, these functions glued together with flexible service oriented communication mechanisms represent the core requisites of a so called Service Delivery Platform (SDP).

Also shown in Figure 16 are the NGN's transport control functions, the network attachment control and resource and admission control functions. Whereas for the context of this work, network attachment control functions, providing IP-layer based authentication, user-profile –based authorization and access network configuration functionalities are not of significant importance, the resource and admission control functions provide important functionalities for controlling network QoS, based on user-profiles, service-profiles or real-time invocation mechanisms.

2.3.1 NGN Transport

The RAC Functions, as shown in Figure 17 are the actual QoS enabling functions in NGNs. It is the architectural place where currently significant work is being carried out, especially for controlling inter-domain and flow-based QoS. It is here where Software Defined Networking mechanisms, which are currently broadly investigated, hook up. Being able to reliably and flexibly control inter-domain traffic QoS and network performance is promising to enable a new range of inter-domain service interworking, service composition and multi-tenant, multi-provider service provisioning.

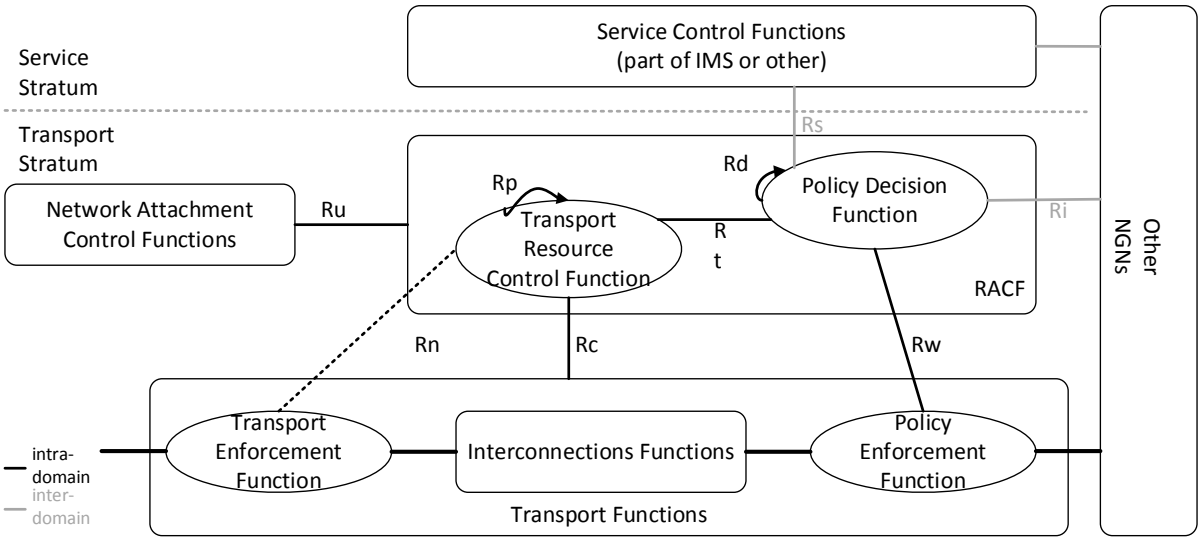

Figure 17: ITU-T NGN Resource and Admission Control Functions [82]

At the point of writing this thesis, already promising approaches for inter-domain bandwidth on-demand, traffic prioritization and QoS-enabled flow control are investigated. This work shows the potentials of future network as a service (NaaS) approaches, but starts from the premise that in contrast to local intra-domain, QoS-enabled, reliable network conditions, the actual inter-domain networking conditions are as unreliable as the standard, unmanaged, best-effort internet currently typically is. Therefore, the current only way to provide QoS assured telecommunication services in a multi-domain environment, with unreliable inter-domain networking conditions, remains in mitigation of network performance degradations through dynamic identification and selection of alternative, service providing domains which at a given point in time satisfy QoS and network performance requirements.

2.3.2 NGN Service Control

As shown in Figure 15, the NGN service/session control layer may be comprised of different, specialized service control functions, depending on the specific type of service. For controlling service invocations and sessions of traditional telephony services, PSTN/ISDN Emulation service control components might be employed. For supporting multimedia streaming applications, particularly of importance for Internet Protocol Television (IPTV) applications, a dedicated multimedia service control component might be employed. Nevertheless the de-facto, standard NGN multimedia service control component already broadly utilized in telecommunication environments remains the IP Multimedia Subsystem (IMS) as standardized by the 3GPP [83], as in many cases the IMS is also used for controlling PSTN/ISDN emulation services as well as IPTV services (IMS-based IPTV). Nevertheless, especially for controlling services of the Internet of Things (IoT), it is worth noting that additional, different service control components for controlling machine-type communication are currently under investigation which are expected to soon, further enrich the NGN service control landscape.

The work described in this thesis assumes that an IMS-like service control system is in place for coordinating service invocations and session control, in an access-network agnostic fashion (for converging multiple access networks) based on user- and service profiles. Typically, as shown in Figure 18 an IMS Core, apart from specific gateways for access network convergence, is comprised of different Call Session Control Functions (CSCFs) and databases hosting user and service related profiles, the Home Subscriber Servers (HSSs).

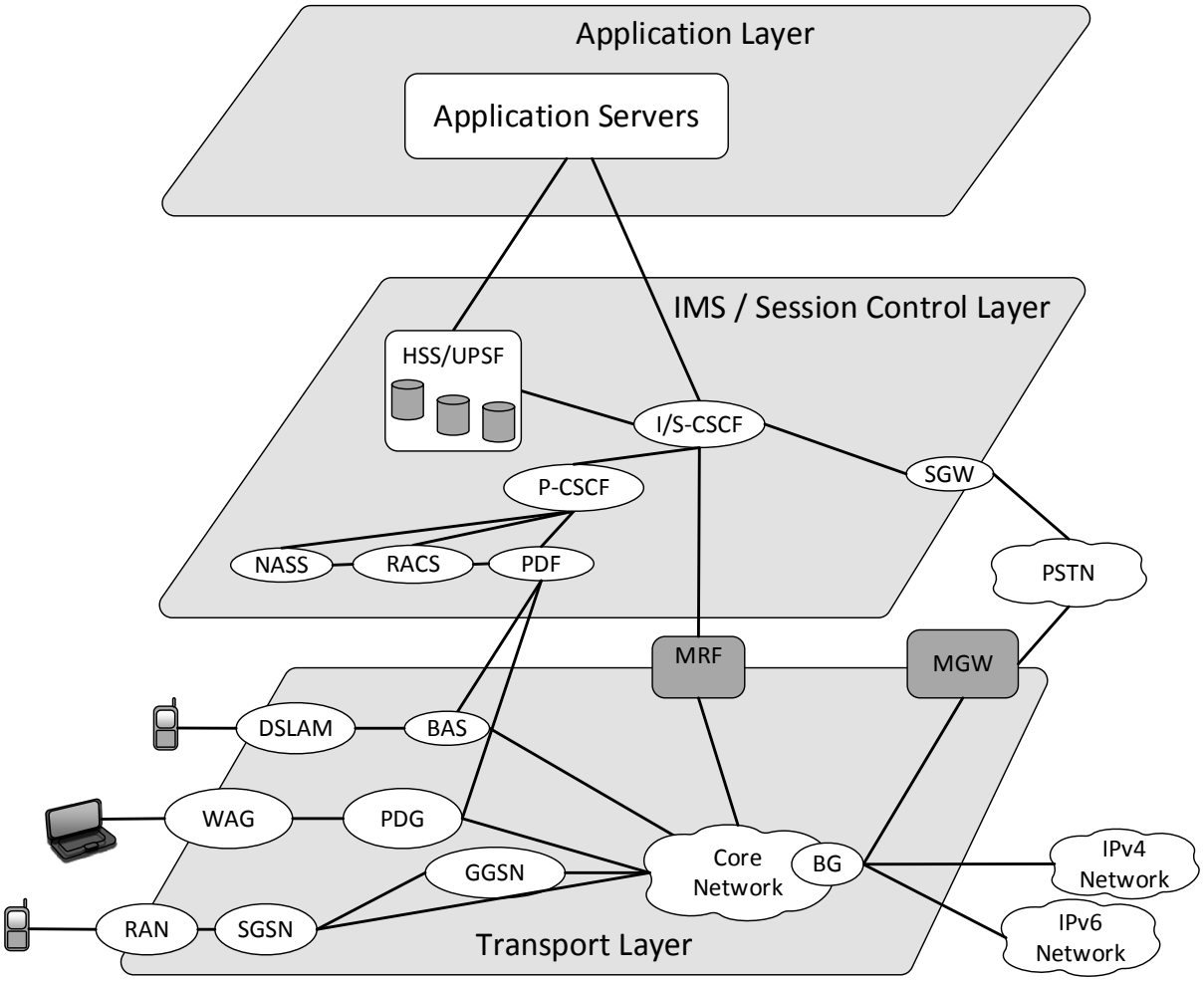

Figure 18: IMS Core Functions (simplification bsd. on 3GPP and ETSI TISPAN IMS spec)

Of core importance for the context of this work are the IMS functions that control the invocation and control of different, distributed NGN services or NGN service enabling functions, whose location can dynamically change over time. Each time a user registers to an NGN, user profiles and service profiles are being downloaded from the HSS to the CSCFs, based on which the actual routing and service control are coordinated. NGN service control predominantly makes use of signaling protocols such as the Service Initiation Protocol (SIP) [84] and Diameter [85]. Many different SIP methods for invocation and real-time control of NGN services are being used for controlling state-less service invocations (such as plain, single request messaging services), but also for complex, state-full service invocations (such as telephony sessions, presence services and multi-party voice, video and messaging sessions).

There are different coping strategies for controlling services in a dynamically changing service environment (where application servicers dynamically change their location) depending on the type of NGN service, whether being a state-less or state-full session-based service. Where it is relatively easy to dynamically change the location of a state-less application server / service, significantly more complex mechanisms have to be put in place to dynamically change the location of a state-full service, assuring continuity of currently active sessions.

Being well aware of this fact this work, for the sake of simplicity starts by analyzing dynamic resource allocation and service deployment mechanisms for state-less NGN services. Depending on the complexity of state-full telecommunication services different, service-specific solutions have to be employed, in order to maintain session states, where different service logics might be used which are out of the scope of this particular work.

Whereas the core principles of NGN service control are published in the author's work on IMS principles [51], the author contributed with state-of-the-art overviews works on NGN services [52] and service delivery platforms [7].

2.3.3 NGN Service Environments

As introduced earlier, the second most important value-add of NGN principles and architectures, apart from access network convergence, relates to the decoupling of transport and service layer, which significantly enables a far richer spectrum of services than traditional telephony networks typically provided, with significantly shorter service creation and service roll-out times. In a world of rapidly changing and evolving IP-based telecommunication services, key to successful NGN service creation, deployment and delivery are versatile mechanisms which improve the entire service creation process and the overall service management lifecycle.

For enhancing service creation processes, the concept of service enabling functions became very popular. The NGN service landscape initially evolved with basic, intrinsic, soon standardized (by the OMA) IMS-based service elements like presence [86], group and list management [87], and simple instant messaging [88] service enabling functions, the most basic service building blocks of many NGN services. Already at an early stage service compositions, making use of those basic service enabling functions showed the benefit and power of service oriented composition and de-composition mechanisms. One example of such an early combinational service, which, by itself only provided a small layer of intelligence on top of already existing service building blocks was the Push-to-Talk service [89]; a service which enables walky-talky-like communication amongst multiple IMS clients / users.

This, however were purely SIP-based applications, which are still relevant for currently emerging Rich Communication Suite (RCS) [90] services. Other, more flexible, more easily compose-able service composition mechanisms utilizing service oriented web services came into play. But let us first understand the space of purely SIP-based services. The RCS initiative is a global initiative which unites telecommunication service providers, mainly telecommunication network (fixed and mobile) operators delivering legacy as well as All-IP-/SIP-based telecommunication services. RCS evolution comprises several RCS versions. Early RCS versions provide simple, complementary IP/SIP-based services like presence or instant messaging. Later versions provide fully IP-based replacements of traditional telecommunication services, such as Voice over LTE (VoLTE) [91]. Whereas currently, commercially available RCS services only provide complimentary IP-based services, soon fully IP-based telephony services will be rolled out, entirely replacing the "old", legacy Circuit-Switched world (incl. infrastructures and services). This however accounts for the

mobile telecommunication world. The fixed telecommunication world already moved towards SIP-based telephony services, already replacing legacy PSTN services to a significant amount (e.g. bundled Direct Subscriber Line (DSL) internet/data and SIP-based PSTN emulation telephony services). Therefore it must be understood that purely SIP-based telecommunication services are and certainly will be telecommunication network and service operator's means for delivering QoS-assured basic, but also feature-rich NGN services across telecommunication operator's domains, countries in a standardized and interoperable, cross-domain fashion (similar to the Global System for Mobile Communication (GSM) [92] "roaming" mechanisms).

This work, in contrast to multiple works on purely cloud-based Web services focuses on NGN services, which are controlled and coordinated through SIP communication mechanisms (in contrast to HTTP-based SOAP and REST-only Web-Service invocation and control mechanisms).

Nevertheless, there is another, more service-oriented world of telecommunication service invocation, composition and control than pure SIP-based service composition, interworking and control. Talking here about SOA–based SDPs for telecommunication services, open telecommunication service Application Programming interfaces (APIs) and Web 2.0 / Telco - converged services. Open APIs in this regard means accessible and remotely useable (accountable, and chargeable) by external third parties, "as is" or in combination / integration with additional Web-based services.

This SOA-based domain, in parallel to purely SIP-based service domains, lately gained significant momentum. Browser-based clients (in contrast to SIP/IMS clients) more and more become capable of utilizing a similar, broad-range (and in many cases even broader ranges) of multimedia telecommunication services (e.g. entire browser / app-based services for messaging, call-setup and control, presence, file-sharing, integrated into social media applications, facilitated by geo-mapping functions. On the one hand the evolution of standardized, end device-base APIs and functions (e.g. accessing device-specific camera, microphone, speaker, as well as location, and system-based presence states) tightly integrated into web browser environments contribute to that evaluation. On the other hand the evolution of HTTP [93] more and more enabling media control functions at the HTTP Layer (e.g. HTTP v5) contribute to that evolution. Fully HTTP v5 enabled, browser-based, thin clients and network-based counter parts (systems coordinating web-services-based services; service with SIP-based service back-ends) have the potential to fully make SIP-based communication / signaling at the User/CPE-to-network interface redundant.

Therefore, both domains the pure SIP-based service domain as well as the SOA-/SDP-based domain need to be taken into account for the telecommunication/NGN-based service specific solution proposed in this work. Whereas it becomes implicitly clear (through the approaches proposed in this work, but surely also through multiple other approaches) that entire SDP functionalities (including all aforementioned SOA-based functions like service creation, composition, coordination and management) can be deployed and operated externally (hosted on cloud and non-cloud based infrastructures) this work only focuses on

single NGN-based telecommunication services (and here only state-less telecommunication services) dynamically and elastically scalable hosted on external clouds, connected to the NGN operator's core network (access networks, transport and session control functions) through standard, best-effort internet.

In this context, the current state-of-play, the current de-facto standard for NGN-based service environments needs to be shortly introduced. Back in the early days of the onset of the NGN evolution, the ITU-T in [78] already defined one important and fundamental characteristic of NGNs related to NGN's functionalities for *"[...] decoupling of service provision from transport, and provision of open interfaces [...]"*[78]. The much later standardized service environment for NGN-based telecommunication services [94], as shown in Figure 19, gives tribute to this, much-anticipated capability (open and secured interfaces, access control for remotely accessing telecommunication service capabilities / telecommunication service enabling functions and service building blocks).

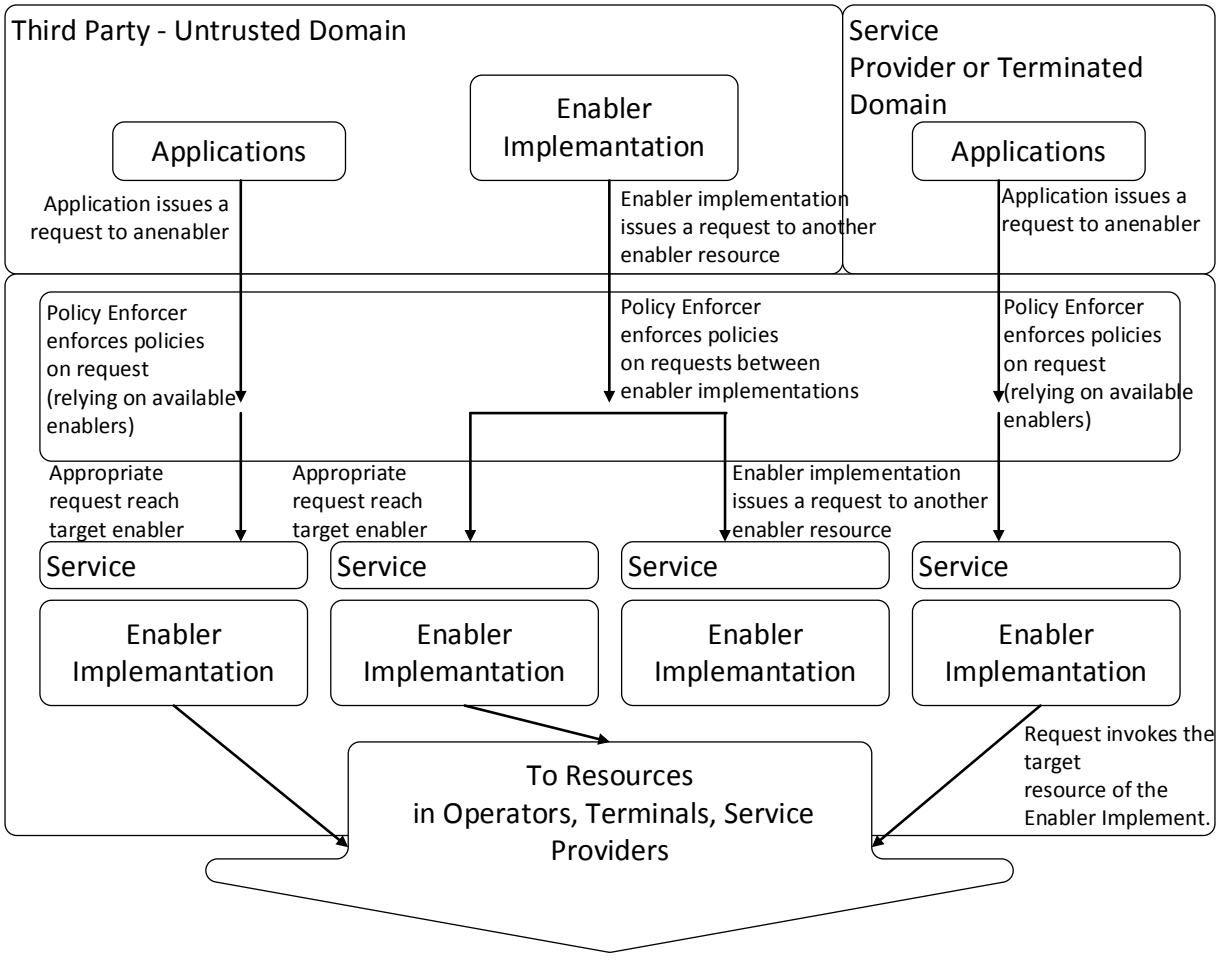

Figure 19: OMA Service Environment Architecture [94]

The main insights to be gained from aforementioned NGN service delivery mechanism descriptions, first is, that both, the purely SIP-based, NGN session control layer based as well as the SDP, Open API-based service control have both concurrently their right to exist. A typical, modern NGN-based telecommunication service provides both, a SIP-based service

control interface as well as a web-based / SOA-based interface (such as Parlay [95], SOAP-based Parlay-X [96] or REST-based GSMA OneAPI [97] interfaces). If the latter interface (web-service based / service oriented) is not provided, in many cases SDP-inert or specific service abstraction mechanisms and service mediation mechanisms are being employed to provide Web-based, service-oriented access to telecommunication service (enabling) functions (translating web-service requests to SIP-requests and vice versa).

The NGN service environment, depicted in Figure 19 initially standardized by the OMA [94] and later adopted by the ITU-T [98], only comprises the policy-based access control and service orchestration functions of the NGN-based SDP. Although being the entry point for external, third party service providers and their users (controlling access and coordinating between services) and therefore of crucial importance, one important aspect of SOA-based SDPs relates to their tight integration of service management functions via service oriented interfaces, integrated into SOA-based SDPs as explained in [16].

Figure 16 shows the management entity as a cross-cutting NGN domain-wide system for the management of NGN systems and services. These SOA-based service management functionalities, which allow for seamless (at least effort-less) integration into the service delivery life-cycle, are introduced in the upcoming sections.

In the context of SOA-based NGN service management and orchestration, the author contributed with innovative mechanisms for NGN service brokerage [1] and NGN service delivery [2], [3], modular NGN service exposure [4], including cross-domain service orchestration mechanisms through infrastructure federation mechanisms [2], [4], [5]. The author contributed to the NGN/OMA-standard based design and development of SOA-based NGN service orchestration and brokering mechanisms [3], [8], [2], [9], [4], [1] leading to the implementation and evaluation of NGN service brokering systems such as the eXtended POlicy based Semantically enabled sErvice bRoker (XPOSER), later called FOKUS Broker.

Domains and Roles NGN-based Telecommunication Service Delivery

As shown in Figure 20, with the emergence of NGN infrastructures (networks, platforms and services), network operators, service platform providers and service providers cloud be different players, each specializing on his specific domain of excellence or, as in traditional telecoms, where all of these roles were united under one umbrella, one single company, where the network operator IS the only service provider. With the help of standardized service control platforms (such as the IMS), it became possible for external, third parties to operate an NGN-based service platform (Service Companies, providing platforms / telecommunication PaaS) on top of a telecommunication network provided by one or several network operators / companies as a service (NaaS). With the help of open service environments (such the OMA service environment [99]), it became possible to offer NGN functions (e.g. multi access network connectivity, QoS, security) and NGN-based service enablers (e.g. presence, group & list management, location, messaging) to third party service providers, who are the actual

customer-facing providers of final, composed service (Sales Companies, or the SaaS providers in Cloud terminology).

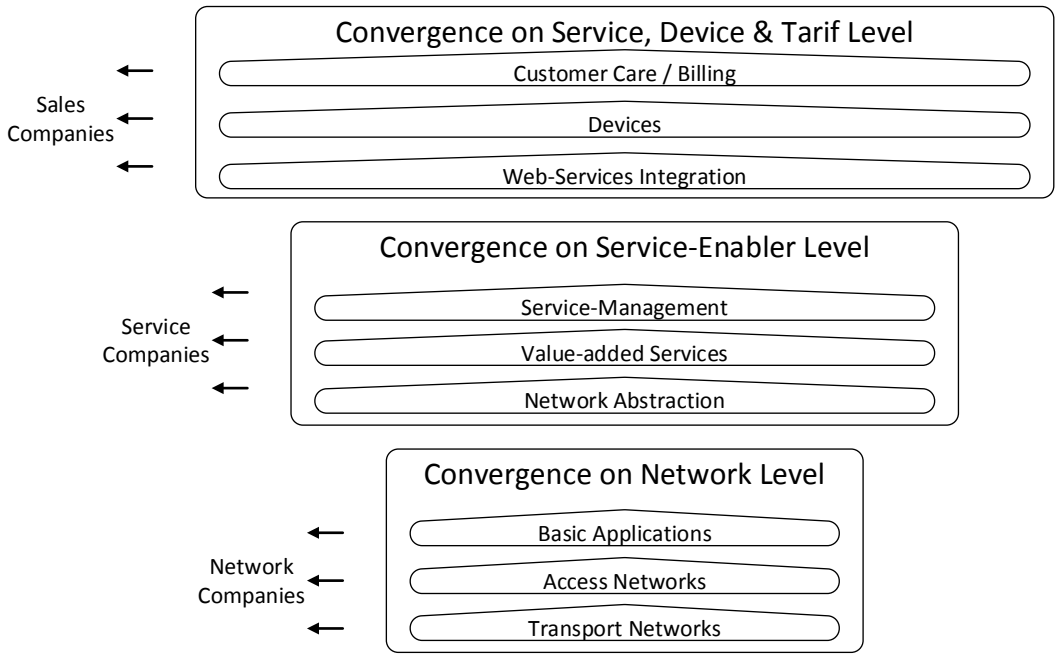

Figure 20: Domains and Roles in NGN Service Delivery

2.4 NGN Service Management

This section introduces the current state-of-the-art NGN management, but basically telecommunication enterprise generic (not exclusively focusing on NGN telecommunication networks and service) management. The benefit of exploiting service oriented interfaces and communication mechanisms for harmonizing service management processes and operations and for easy integration of management mechanisms into ever changing service environments and services (to be managed), already became clear at an early stage. Whereas early approaches for service oriented service management systems inter-process communication exploited Common Object Request Broker Architecture (CORBA) - based service-oriented inter-process communication / service invocation mechanisms, SOAP-based and later in REST-based web service communication based mechanisms became popular.

The following sections introduce NGN Management frameworks and principles as they have evolved from basic IT-Management principles, to enterprise-wide business-oriented management, to cross-domain, SLA-aware, multi-provider frameworks.

2.4.1 NGN Management Principles

With the emergence of NGNs, growing service portfolios, ever shorter service life-cycles and service time to market, management frameworks needed to evolve in order to cope with these new requirements. From vertical, service specific resource and service management mechanisms, which require dedicated integration of management functions for each and every

new service to be rolled-out, management mechanisms needed to evolve towards service-oriented, "plug-and-play"-like, dynamic integration of management functionalities already being prepared for, and integrated into the service logic at service creation time. This is realized through Model-driven-Engineering (MDE) principles and practices, which require standardized interfaces, information and data models as well as standardized process frameworks.

SOA-principles further aided this evolution, through loosely-coupled service interworking, standardized service discovery and composition mechanisms. These mechanisms enable "plug-and-play" functionalities, which are of utmost importance for shortening service delivery life-cycles, not only by enabling easier integration of new service functionalities into service delivery infrastructures, but also by facilitating and improving inter-management system and process communication. "Everything as a Service" principles thus are not only useful and utilized for customer / consumer-facing service building blocks and service compositions, but also for resource and management system facing service integration mechanisms.

But let us start at the point, where telecommunication networks, infrastructures and enterprises required enhanced management functionalities as compared to (typically in-house) IT-management functions. Early IT-management standards, i.e. standards for managing IT infrastructures, like the Open Systems Interconnection (OSI) Systems Management Overview (SMO) defined five cornerstones of management domains, which are 1) fault, 2) configuration, 3) accounting, 4) performance, and 5) security management (FCAPS).

Although FCAPS was adopted by the telecommunication domain / the ITU-T (as part of ITU-T's Telecommunications Management Network (TMN) standards), the need for additional telecommunication management functions became apparent, as shown in . Particularly of importance for this work are the additional QoS management functions.

Table 4 3GPP [100] Extensions to ITU management tasks

ITU's *traditional* management tasks (FCAPS)	3GPP's *additional* management tasks
Fault management	Roaming management
Configuration management	Fraud management
Accounting management	Software management
Performance management	User and Equipment management
Security management	Quality of Service management
	Subscriber and Equipment trace management

Later on, the TMForum defined the FAB, the Fulfilment, Assurance and Billing model, which can be roughly mapped to FCAPS, mapping Configuration and Security Management to Fulfilment (F) processes, Fault and Performance management to Assurance processes (A) and Accounting management to Billing processes (B). TMForum's FAB (particularly the

enhance telecom operations map (eTOM)[47]) can be seen as the successor of the FCAPS model.

When it comes to the actual specification of telecommunication infrastructure and service management processes, there are two widely used standards: eTOM specifically focusing on the telecommunications domain and the more generic Information Technology Infrastructure Library (ITIL), which at a first glimpse might be seen as competing approaches and standards, but as will be seen later are actually providing complimentary viewpoints on Telco/IT management functions and processes.

Table 5 eTOM vs. ITIL (bsd. on [101])

	eTOM	ITIL
Context	Prescriptive Catalogue of Process elements Total Telco Enterprise process framework	Non-prescriptive guidelines IT Service Management
Objectives	Provides business process blueprint for Telco service providers Provides vocabulary and enables effective communication	Provides requirements for IT services Improves IT service's quality Reduces service provisioning costs
Scope	Top-down view, hierarchical view of business processes across whole Telco enterprise Does not by itself address how these processes should be realized (automated or human action) Focus on service delivery to external customers	Specifies flows in a number of key operational areas, with orientation towards how processes map to IT support environments Focus on internal IT customers
Adoption	eTOM has been adopted by the ITU as an international standard for Telcos	The ITIL set of best practices is used by thousands of companies worldwide
Implementation	eTOM is a framework; implementation differs from company to company Implementation is supported by other TMF/NGOSS specs, including SID, NGOSS Lifecycle & Methodology.	ITIL is a framework; implementation differs from company to company Only lately, ITIL provides implementation guidelines
Compliance	Certification is on tools, not on organizations or processes.	ITIL is not a standard, nor a set of regulations, therefore neither people, processes or tools can be deemed "ITIL compliant"

As a result both approaches, the eTOM framework and ITIL best practices, applied together as explained in [101], provide eTOM compliant processes that deliver ITIL compliant services, which can be regarded as the best-of-breed telecommunication service management models combined with the best-of-breed IT management best practices.

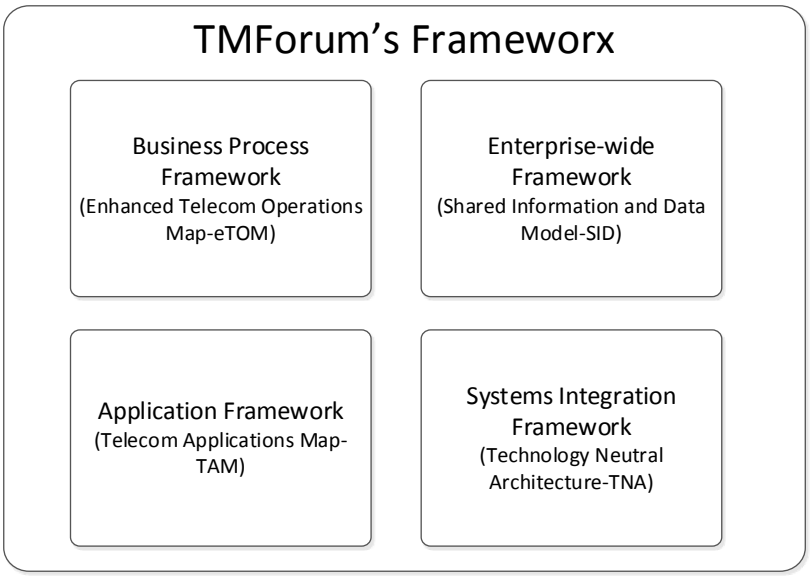

Figure 21: TMForum's NGOSS [102], now TMForum "Frameworx"

The former New Generation Operations Systems and Software (NGOSS) Framework [102], lately termed TMForum's "Frameworx", as shown in Figure 21, is comprised of four complimentary frameworks:

- the *Process Framework* – the enhanced Telecom Operations Map (eTOM) [47] providing a framework for enterprise-wide supplier, resource, service, product, customer management processes
- the *Information Framework* – the Shared Information and Data Model (SID) [103] a common information model which describes all involved entities (customers, resources, services) and the information that flows through the enterprise
- the *Application Framework* – the Technology Application Map (TAM) [104] providing logical groupings of applications, a systems map capturing how processes are implemented in deployable applications
- and the *Integration Framework* [105], formerly known as the Technology Neutral Architecture (TNA), provides practices, tools and standard interfaces for the enablement of open, interoperable virtualized digital services focusing on maximizing re-use, agility of operations and scalability

Together, the TeleManagement Forum Frameworx provide a framework of enterprise-wide processes (the eTOM), a functional, information (the SID) and physical architecture blueprint (Integration Framework), methodologies for developing, modelling and orchestrating solutions, based on contract-driven SOA which integrates business, service, and resource management.

Following a model driven architecture (MDA) approach, the TMForum frameworks, provide processes, information, as well as specifications of common interfaces for enabling SOA-based (i.e. re-usable, open, plug-n-play), and policy-based (i.e. flexibly configurable, increased automation) OSSs and BSSs for telecommunication service providing enterprises.

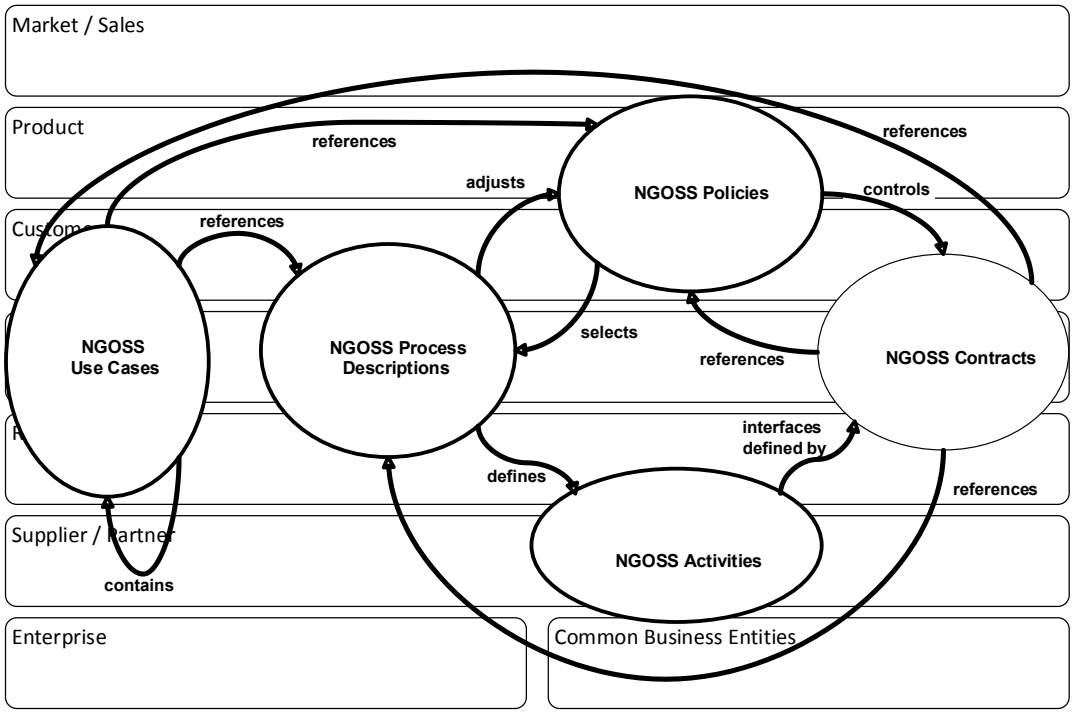

Figure 22: Relationship of NGOSS Artifacts [102]

Shown in Figure 22 are the relationships of NGOSS artifacts. In the background of Figure 22, the SID is shown, i.e. the enterprise-/process-wide information framework. NGOSS Use Cases, on a high-level define actors, goals and policies and processes used to reach those goals. Processes referenced in those Use Cases are refined and concretized, based on the eTOM framework, as part of the NGOSS process descriptions. These define the NGOSS activities (detailed, fine-grained eTOM processes). On another leg, the NGOSS Use Cases refer to NGOSS policies [66], part of the SID, enabling policy based management, used to manage and control the behavior of the activities. The NGOSS, SOA-based Contract specifications drive the implementable and deployable solution, offering a high-degree of re-use and by doing so shorten the overall problem solving lifecycle [102].

Shortening the lifecycle of developing and deploying management solutions is one of the primary goals of TMForum's Frameworx utilization. With the break-through of SOA technologies, TMForum's Service Delivery Framework SDF [106] (now part of TMForum's Software enabled Service (SES) Management Solution), further expanded on mechanisms and interfaces for automated, plug-and-play integrations of management processes and services into the overall OSS, BSS platform. The SDF reference model, shown in

Figure 23 shows the scope of interfaces and services defined by SDF. Through strong adherence to SOA principles, the SDF provides a Framework through which management services, as well as infrastructure support services can seamlessly be plugged together and into their respective platforms. With the final goal of automating the entire service lifecycle process, SDF utilization enables rapid and automated (i.e. "plug and play") integration of SDF-enabled services into OSS, BSS (management platforms/service). By exposing service-oriented service management interfaces (SMIs), SDF-enabled services can easily be plugged

into the management environment of an enterprise. Service chaining is also enabled as an SDF-enabled service can issue its dependencies on other service capabilities, including required capabilities and capacities of the underlying network and service infrastructure.

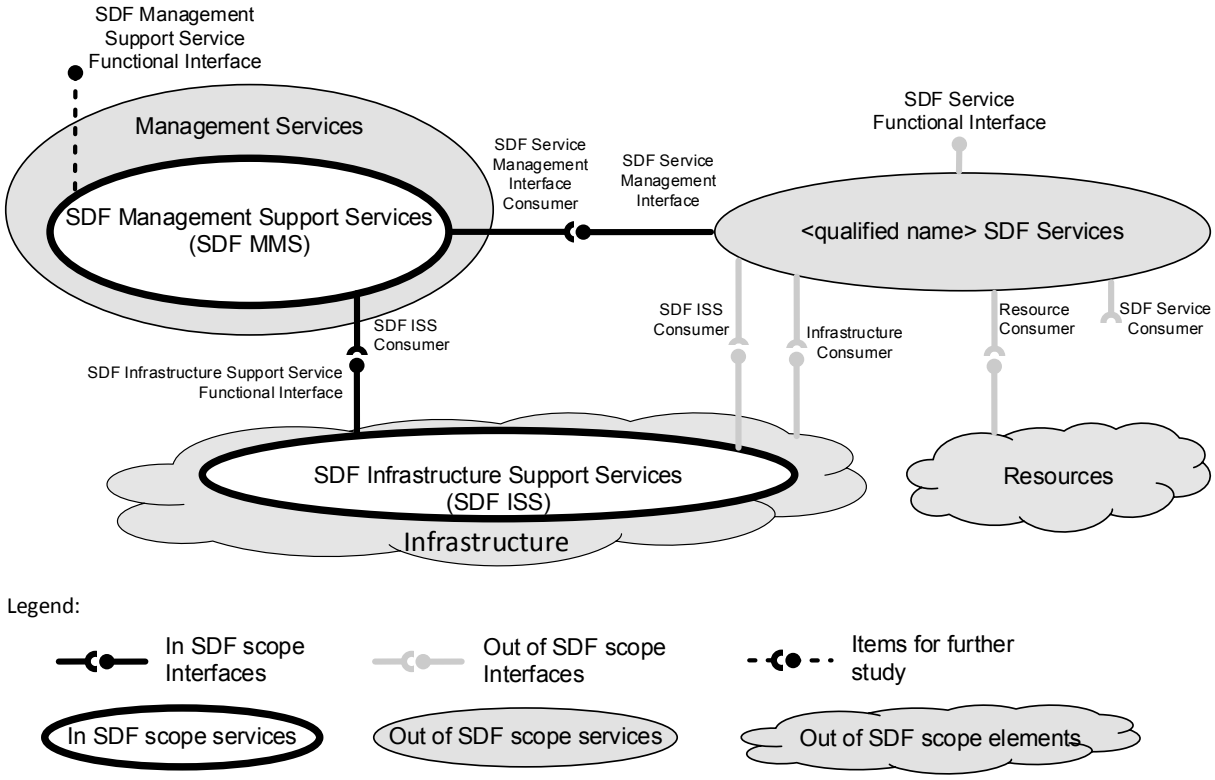

Figure 23: TMForum SDF Reference Model [106]

This work exploits TMForum's Frameworx in chapter (for modeling policies, resources, and services) and the SDF concepts and models for coming up with an SDF-oriented cloud-based NGN service lifecycle management framework.

In the context of NGN management frameworks, focusing on NGN Operation Support Systems, the author developed SOA-based NGN management mechanisms applying them for automating service provisioning and fault management mechanisms for IMS [10], [11] and SDP [12], NGN fault localization [13], automated management of NGN service compositions [14]. The author's main proposition of bringing SOA principles to the domain of NGN service management [15] and SOA-based integration of NGN management services into the SOA-based service delivery environment [16], relates to the enablement of higher degrees of automation of the service management lifecycle (i.e. plug and play principles allowing automated composition, deployment and management of composite NGN services, as shown in [17]). Through application and integration of aforementioned SOA-based NGN service management principles, the author led the design and implementation of the SOA- and policy-based NGN management system OMACO (OSIMS Management Console) [13], [12], [14], [15].

2.4.2 Inter-domain Telecommunication Service Management

With IPSphere [107], the TMForum fosters inter-domain and inter-provider service management. Exploiting SOA mechanisms across administrative domains, requires tightly interworking Operations Support Systems and Business Support Systems (OSS / BSS) of multiple enterprises. By providing cross-domain OSS/BSS interworking, the long envisioned goal of cross-domain service compositions, as shown in Figure 24, can be realized. Cross-domain billing mechanisms, allow involved parties providing service components to the overall composite service to be dynamically and flexibly integrated into the value chain, where revenue is dynamically shared across contributing service component providers. Similarly, and equally important, in such multi-stakeholder scenarios, cross-domain SLA management and fault management can be realized through IPSphere's approach to SOA-based, cross-domain OSS interworking mechanisms.

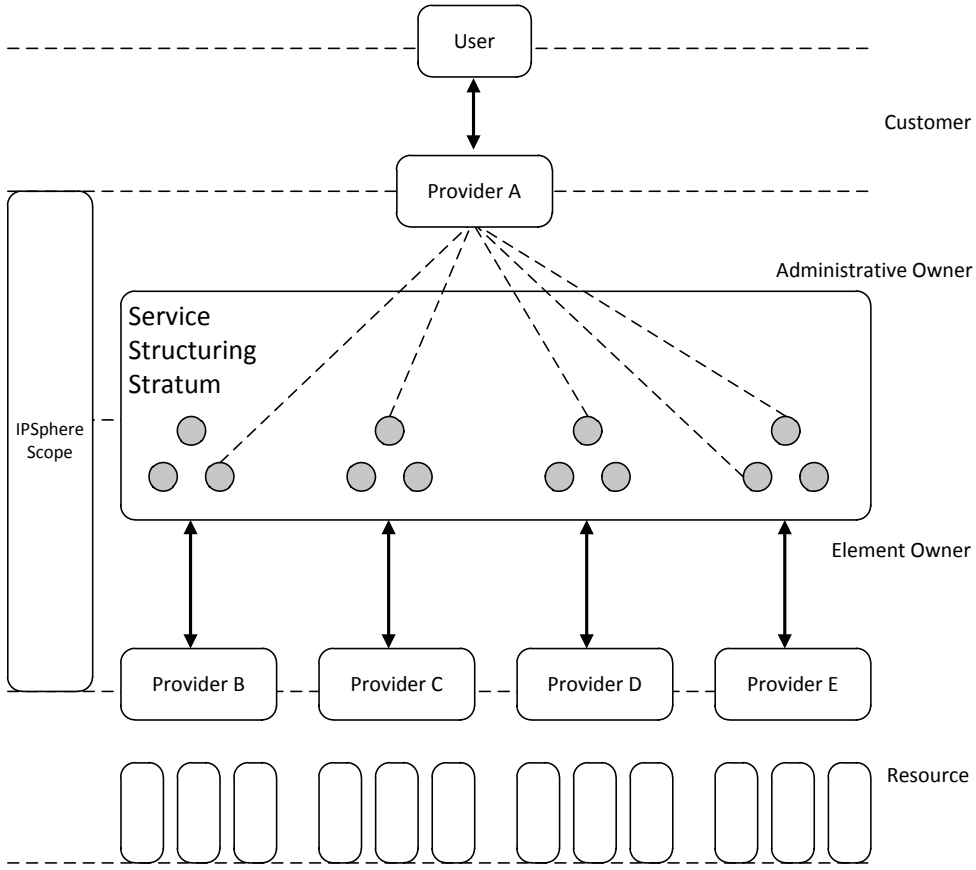

Figure 24: IPSphere [107] Scope

2.4.3 NGN Resource and Service Management

Having understood that the TMForum's frameworks for telecommunication enterprises and services management, in-fact can be seen as the de-facto standard for the telecommunication industry in the management domain, further focus is put on management domains and tasks of core importance for this work, which are Resource and Service Management domain.

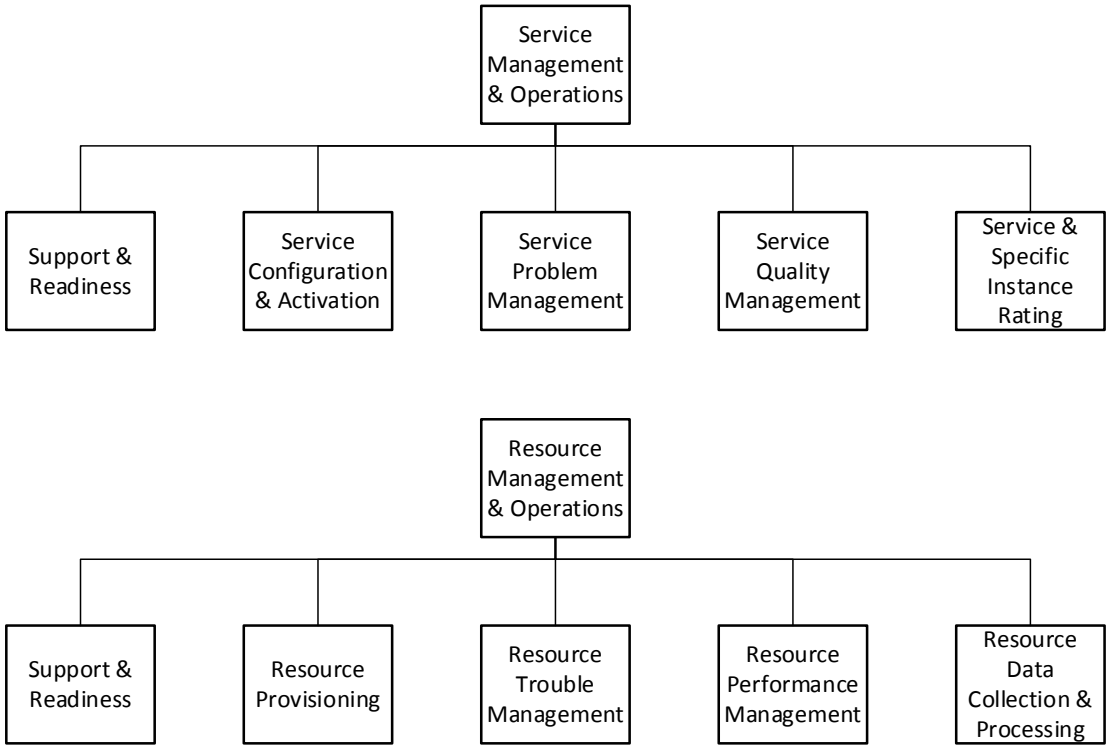

Figure 25: eTOM [47] resource and service operation management processes

Looking at the main Service Management and Resource Management processes as defined by TMForum's eTOM [47] framework allows identifying *Service Quality Management* (SQM) in the Service Domain and *Resource Provisioning* in the Resource Management domain to be of core importance to this work with implications to *Service Configuration* and *Resource Performance Management* processes.

Refining the focused eTOM resource and service management processes by taking the ITIL view [108] on these areas into account brings about the following two, relevant ITIL disciplines / best practices: *Capacity Management* (closely related to eTOM resource provisioning) and *Service Level Management* (SLM) (closely related to eTOM's SQM). While in ITIL Capacity Management differentiates between Business Capacity Management (BCM), Service Capacity Management (SCM) and Resource Capacity Management (RCM), this work primarily focuses on RCM. In ITIL v3, the term Resource Capacity Management changed to Component Capacity Management (CCM).

2.5 Cloud Computing Principles

Cloud Computing infrastructures and services are primarily enabled through compute, storage physical resource / hardware and network virtualization mechanisms, which provide an abstraction from the actual physical resources. This enables significantly more flexible allocation of capacities for applications and services compared to static application hosting mechanisms. However, using virtualization mechanisms alone, from the perspective of this work, does not suffice for a virtualized data-center or managed IT service infrastructure to be

called a cloud infrastructure, which often was and is a source of confusion. From the viewpoint of this work, central for an infrastructure to legitimately be called a cloud infrastructure are its elastic, demand-based resource scaling capabilities. To expedite on the essential characteristics of cloud computing infrastructure and services, the de-facto standard (already accepted by the majority of the research community [109]) for defining cloud related mechanisms is introduced. The NIST definition of cloud computing [24] defines 5 essential characteristics, 4 deployment models and 3 service models, as depicted in Figure 26.

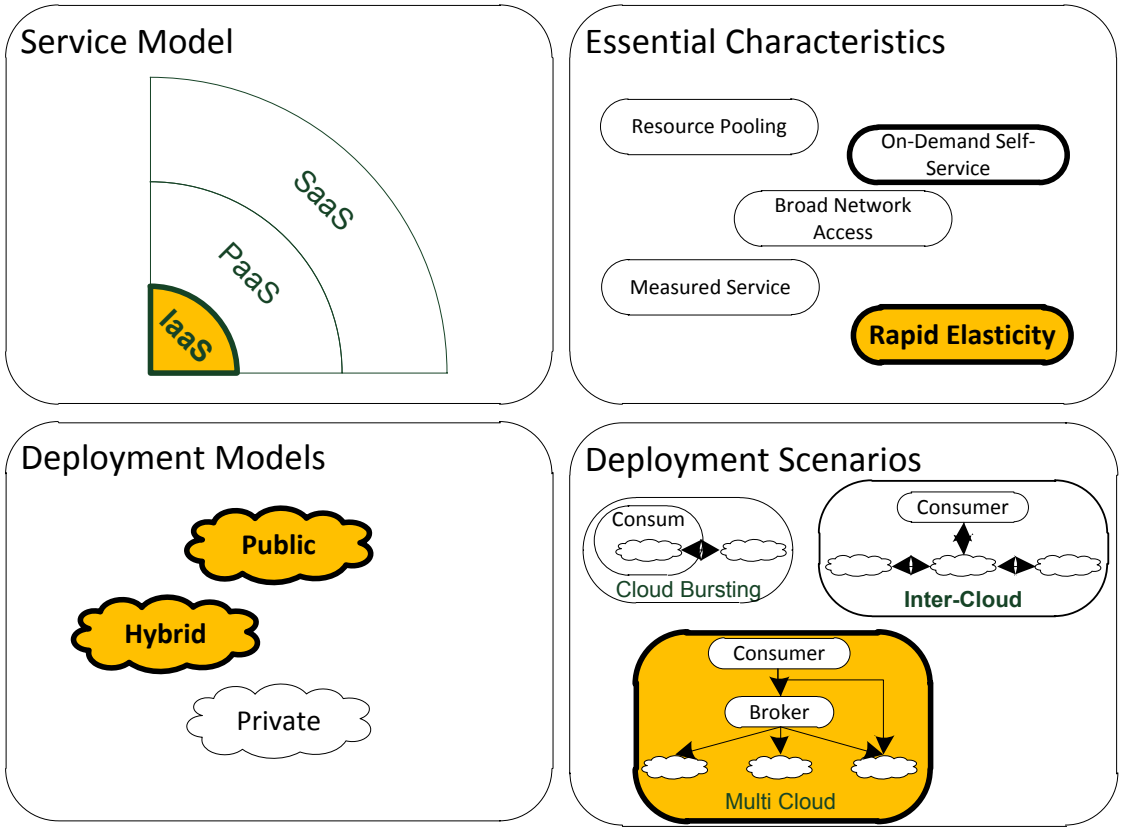

Figure 26: Essential characteristics, service and deployment models, bsd. on NIST [24]

Here it becomes clear that *resource pooling* mechanisms are only one of the *five essential cloud computing characteristics*, whereas *rapid elasticity*, either achieved through cloud-internal resource provisioning / de-provisioning and load-balancing mechanisms or through remote, *on-demand self-service* resource provisioning requests is at least of similar importance. Therefore, public cloud infrastructures like their most famous representative, the Amazon EC2, only after introducing and providing automated scaling functionalities, were (according to the NIST definition and the viewpoint taken in this work) entitled to fully call themselves a cloud infrastructure. This however, happened only years after the EC2 was introduced and was already widely recognized as a cloud infrastructure, although from a critical perspective providing only little more than traditional, managed IT infrastructure hosting providers. The fourth essential *measured service* characteristic as defined by NIST, is of equal importance for providing internal scaling mechanisms as for providing the ultimate added-value of cloud infrastructure usage from an economic/business perspective - "pay-per-use" pricing models. On the one hand, only by being able to measure current resource

utilization / consumption, adequate resource scaling mechanisms can be applied, on the other hand, only by tracking and reporting utilized resource capacities at each given point in time, transparent (for providers as well as for consumers) accounting mechanisms can be provided. Finally, and perhaps most critical for the full break-through of cloud computing services is the essential *broad network access* characteristic. The usability of cloud infrastructure services like cloud storage services, as well as cloud-based office services stands and falls with the capabilities / performance of the network connecting end-users / cloud clients with the actual cloud service.

This brings us to the three different *cloud service models* as shown in Figure 26 and defined by NIST. Whereas this work exclusively focuses on *Infrastructures as a Service* (IaaS), providing computing/processing, storage and networking resources as-a-services, two more cloud service models are defined, *Platforms-as-a-Service* (PaaS) and Software-as-a-Service (SaaS). In contrast to providing bare computing, storage and networking resources, PaaS services provide environments for cloud-based application developers to develop and deploy scalable, multi-tenant cloud applications exploiting dedicated PaaS APIs, providing access to common, frequently used application building blocks and service enabling functions. This significantly simplifies the overall application development process, as well as the need to develop and integrate mechanisms for application scalability. It should be noted however, that typically, PaaS-based applications and service, cannot simply be migrated from one PaaS to another PaaS platforms, as each PaaS platform typically exposes its own specific APIs and has its own platform specificities. This fact has strong implications regarding platform vendor / provider lock-ins. Finally, *Software-as-a-Service* (SaaS) cloud services models typically provide Web-based applications and services that are usually utilized and consumed directly by the final service end-users. For the actual SaaS user underlying PaaS and IaaS cloud computing technologies are usually fully transparent. SaaS users might not even know whether a service is a SaaS-based service or a traditional Web-based service running on legacy managed IT infrastructure hosting platforms. However, with growing underlying capabilities and capacities of SaaS-based services, ever more application logic and computation is carried out on the network, i.e. on the SaaS providing cloud platform, which steadily reduces the need for capabilities and capacities that were traditionally required by the end-devices. This, together with the so called multi-tenancy of SaaS-based services allows for utilization of complex, computation-heavy applications (including enterprise applications) by multiple end-users in parallel, using only light-weighted, “thin” clients (e.g. tablets or smart-phones). Apart from several advantages through central software and security updates and patches and central service management, SaaS-based services not only revolutionize the business-to-consumer (B2C) market, but also the B2B and B2B2C market. Here, SaaS-based services readily make use of SOA technologies, open APIs and secured inter-domain / inter-provider service composition and communication mechanisms.

Conceptually, five different roles or five potentially different administrative domains are commonly distinguished, the domain of the cloud consumer, cloud provider, cloud broker, cloud auditor, and cloud carrier as depicted in Figure 27 (based on NIST [24]).

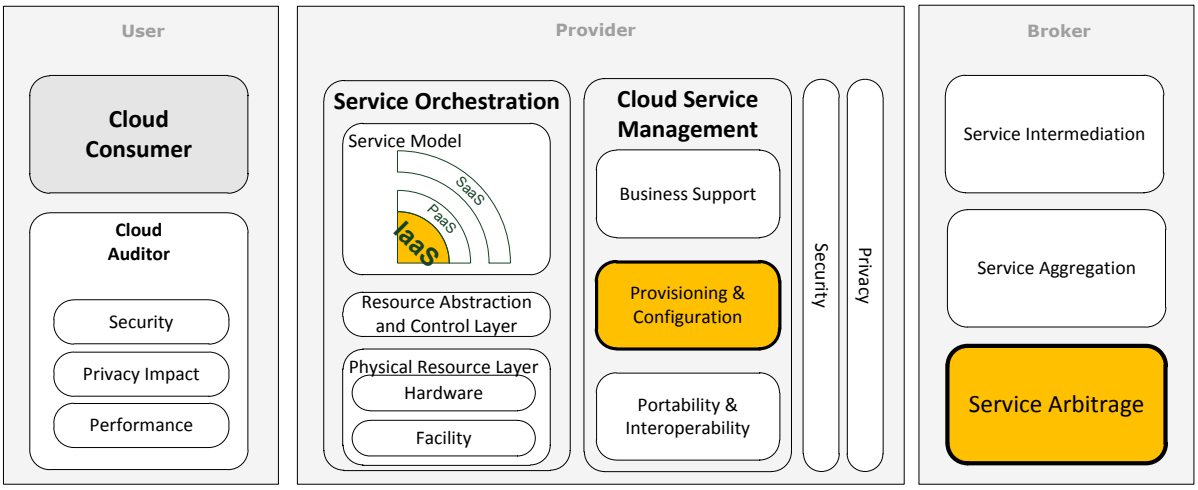

Figure 27: Combined conceptual Cloud reference diagram bsd. on NIST [24]

- In short, these five roles are typically associated with the following attributes:
- *Cloud Carriers* provide the connectivity, i.e. the networking facilities between consumer and cloud provider
- *Cloud Providers* provide the actual cloud IaaS/Paas/SaaS cloud services to the actual consumers according to aforementioned cloud models, on top of physical resources, which are virtualized, abstracted and controlled through specific cloud infrastructure management mechanisms. Cloud Service Management, Service Deployment, Service Orchestration, Security Management, and Privacy Management account for cloud providers' main activities and responsibilities.
- *Cloud Brokers* negotiating between consumers and cloud providers, i.e. providing a specific added-value to the cloud services that are provided are of core importance for the subject of this work. This added value might either be the provision of a specific service-enhancing or service facilitating functionality, capability or feature (*service intermediation*) or the creation of new services though combination and/or integration of multiple services (*service aggregation*), or in contrast to creation of new services from a fixed set of services as in service aggregation, to dynamically and flexibly identify new, alternative, potentially more optimal services for delivery to the end-user (*service arbitrage*).*
- *Cloud Auditors* asses the capabilities and functionalities of a cloud service, e.g. provided security mechanisms, privacy or reliability and usually provide the results of such audits to potential cloud users.
- *Cloud Users* not necessarily are the final service consumers / end-users. In fact, in the case of this work as well as in the case of most IaaS and PaaS cloud service users, cloud users are service providers themselves, using cloud services to create and/ or deploy (manage) their applications.

*Note: It is worth mentioning that most of these service brokering functions (intermediation, aggregation and arbitrage) have a strong affiliation with SOA-based service composition and orchestration mechanisms. In fact, for as long as Web service based cloud service

communication means and interfaces are being used, there is significant overleap between both technological areas. This is why it is worth to further narrow down the "cloud brokering" approach, followed in this work. This work focuses on IaaS cloud brokering mechanisms, predominantly providing value-add through mechanisms for IaaS Cloud service arbitrage.

2.6 Cloud Infrastructures

Any cloud infrastructure is comprised of underlying physical hardware resources (i.e. compute, storage and networking resources). Hardware virtualization mechanisms, whether through full, partial or para-virtualization techniques and hypervisor technologies allow for slicing of the physical resource capacity based on which multiple virtual instances / virtual machines can be operated. These virtual instances typically provide different combinations of computing, storage and networking capacities and are used for operating different operating systems serving different kinds of applications / services. For the actual applications / service virtualization is typically fully transparent. Also the type of virtualization technology used, the utilized hypervisor technologies are typically invisible to the hosted platforms and applications. For managing such virtual infrastructures, starting, suspending, stopping virtual instances, cloud infrastructure management systems / platforms are used, described in the subsequent section.

2.7 Cloud Infrastructure Management

The essential component of cloud infrastructure management systems are virtual infrastructure managers (VIMs), responsible for the actual abstraction of physical resources, i.e. for the provisioning and un-provisioning (starting, stopping, suspending and re-starting) of virtual machines (VMs). VIMs can work with single types of hypervisor / virtualization technologies or with several, heterogeneous hypervisors in parallel. Some VIMs can even manage external cloud infrastructure resources through interworking with their (e.g. public cloud providers') APIs. Crucial for IaaS resource management are externally accessible APIs, in Figure 28 termed "Virtual Platform Interface", through which IaaS users can communicate with VIMs. Typically users first need to discover available resource instances (e.g. micro, small, large instances VM instances) through API requests, and subsequently issue virtual resource provisioning requests. In the case of virtual compute resources, users either have to select from a list of virtual images (containing pre-configured OSs, compatible with the selected instance type) or provide the location / URL to a VM image which has been pre-configured by themselves. Based on users' requests VIMs manage the deployment / management processes of VMs, including the provisioning of storage and networking related elements.

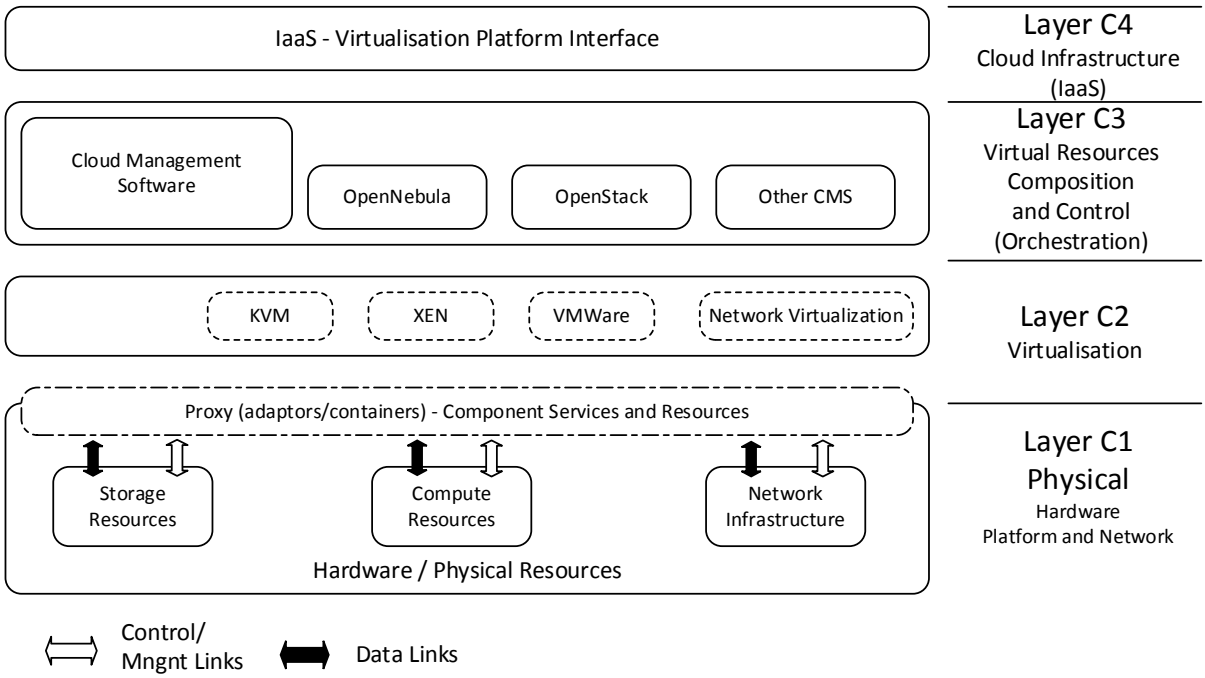

Figure 28: IaaS Layers, simplification of [110]

2.7.1 Cloud Management APIs and Virtualization Formats

The typical methods exposed by IaaS Cloud Management APIs are methods for discovering the resources of an IaaS infrastructure as well as methods for provisioning cloud resources. Typical cloud management APIs allow for retrieving information about available cloud instances, hardware profiles, realms, and available VM images. Furthermore, typically the management of virtual instances, i.e. creation, destruction, starting, stopping and re-start is supported through cloud management APIs.

The Open Cloud Computing Interface (OCCI) [111] standardized by the Open Grid Forum (OGF) is a Cloud management API which has evolved over the past years. The OCCI allows management of the entire Cloud resources, is compatible with Open Virtualization Format (OVF) [112] and the Cloud Data Management Interface (CDMI). Whereas the OVF, standardized by the Distributed Management Task Force (DMTF) provides an open format for packaging software to be run in virtual machine, the CDMI, standardized by the Storage Networking Industry Association (SNIA) provides an interface for managing Cloud storage. Now that the OCCI has been also adopted by OpenStack it can be regarded as the de-facto, open Cloud management API standard (though acknoledging that still the Amazon Web Service API is dominating the global landscape of Cloud APIs).

For IaaS Cloud Brokers, it is of supreme importance to interface and interwork with as many IaaS clouds as possible (increasing the choice of available Cloud resources). Cloud API mediation functions significantly aid in overcoming these platform-specific API-specificities, allowing for multiple heterogeneous IaaS Cloud APIs to be controlled in a unified fashion.

API mediators like DeltaCloud[2], jClouds[3], Apache LibCloud[4], Dasein Cloud[5], Cloudloop, LunaCloud or Simple Cloud provide Cloud API mediation functionalities, abstracting from different Cloud APIs and providing a single API for the management of several clouds exposing heterogeneous APIs, as shown in Figure 29. For multi-cloud brokering systems the employment of such cloud API mediation mechanisms is imperative.

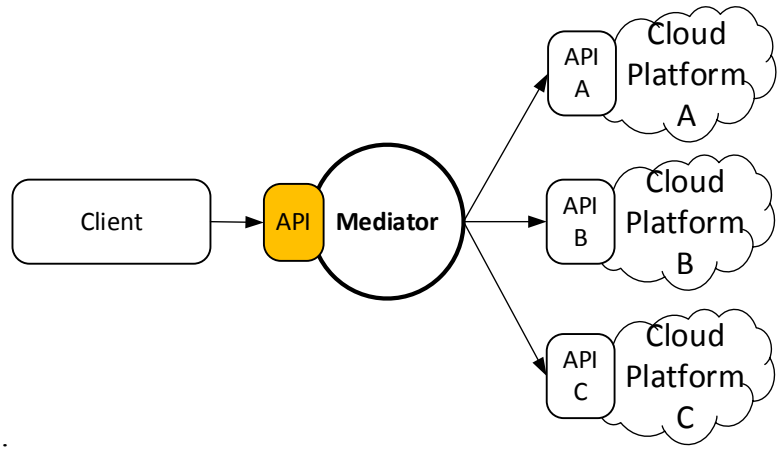

Figure 29: Cloud API mediation and abstraction

2.7.2 Cloud Resource Provisioning and Scheduling Mechanisms

In contrast to on-demand, ad-hoc cloud resource allocation, typically administered and invoked by cloud infrastructure users directly, dynamic cloud service management involves the allocation of resources based on different policies, triggers, events and thresholds. The most typical policy which usually governs single cloud internal automatic application scaling processes is based on current resource utilization metrics. As soon as monitored resource utilization metrics (e.g. average CPU, memory or network utilization) exceeds or deceeds certain pre-configured thresholds resource allocation mechanisms for application scaling are being triggered. Nevertheless, apart from resource utilization-based metrics, QoS/SLA-related metrics, but also cost-related metrics might be used. In a federated cloud scenarios, involving multiple cloud infrastructures (and their resources), additional resource allocation policies, not necessarily related to elastic scaling mechanisms are governing the resource allocation process. Here, additional cloud infrastructure specific policies and metrics, like infrastructure management performance, connectivity / network performance, resource costs, reliability, trustworthiness and reputation might be used for controlling the resource allocation process.

[2] https://deltacloud.apache.org

[3] https:// jclouds.apache.org

[4] https://libcloud.apache.org

[5] http://dasein-cloud.sourceforge.net

Typically resource allocation mechanisms have the objective i) to ensure capacities are sufficient for serving application load / service requests, ii) to ensure that allocated resources are sufficient for assuring specific QoS / SLA levels iii) to minimize resource allocation overhead / over- provisioned resources, usually involving unnecessary costs and energy iv) to ensure that other functional and non-functional (e.g. security / privacy) parameters are assured.

Typically, there are two, sometimes three different modes of IaaS cloud resource booking and provisioning mechanisms. As depicted in Figure 30 long-term resource reservation and provisioning typically involves renting of IaaS cloud resource capacities on a monthly/yearly basis. Short-term, on-demand cloud resource provisioning can typically be conducted within seconds to minutes. Nevertheless, it is common to have a minimal lease time for cloud resources, typically in the order of minutes / hours. This in some cases needs to be taken into account when calculating the benefit vs. the cost of moving / migrating cloud resources / service from one provider platform to another. Finally, the IaaS cloud resource spot-market offers a third option; cloud resource reservation and provisioning based on variable cloud resource prices. Resources are only provisioned if resource costs are within pre-defined margins. The latter is particularly of interest to cloud resource users, who are able to flexibly choose the time when cloud resources are needed (e.g. for computationally heavy, single-shot tasks). IaaS cloud federation brokers primarily make use of IaaS cloud platform provider's on-demand resource provisioning mechanisms. Nevertheless, by repeated updating of internally maintained cloud resource cost tables, cloud federation brokers also provide spot-market-based resource provisioning.

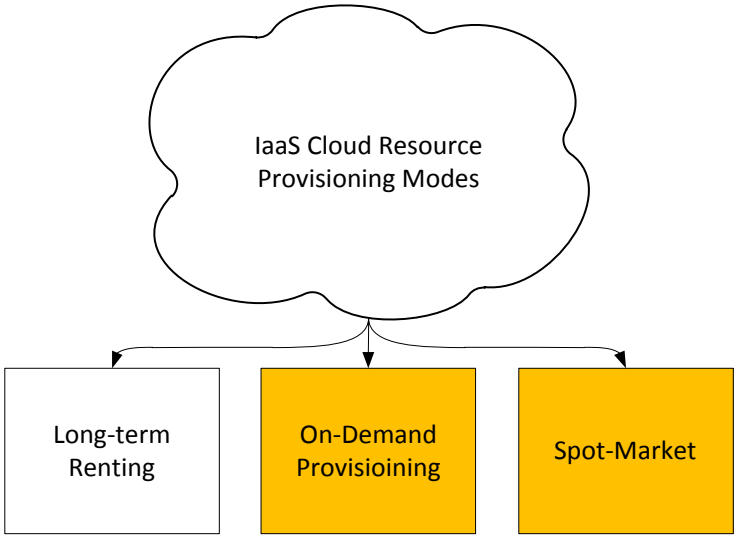

Figure 30: IaaS Cloud Resource Provisioning Modes

2.8 Cloud Service Management

Whereas VIMs are typically agnostic to the actual service configurations, dependencies and interworking service elements, cloud service management, as defined by NIST, shown in ,

provide scaling / rapid services provisioning services, monitoring, reporting metering, and SLA management services as well as various business support services and services for portability and interoperability.

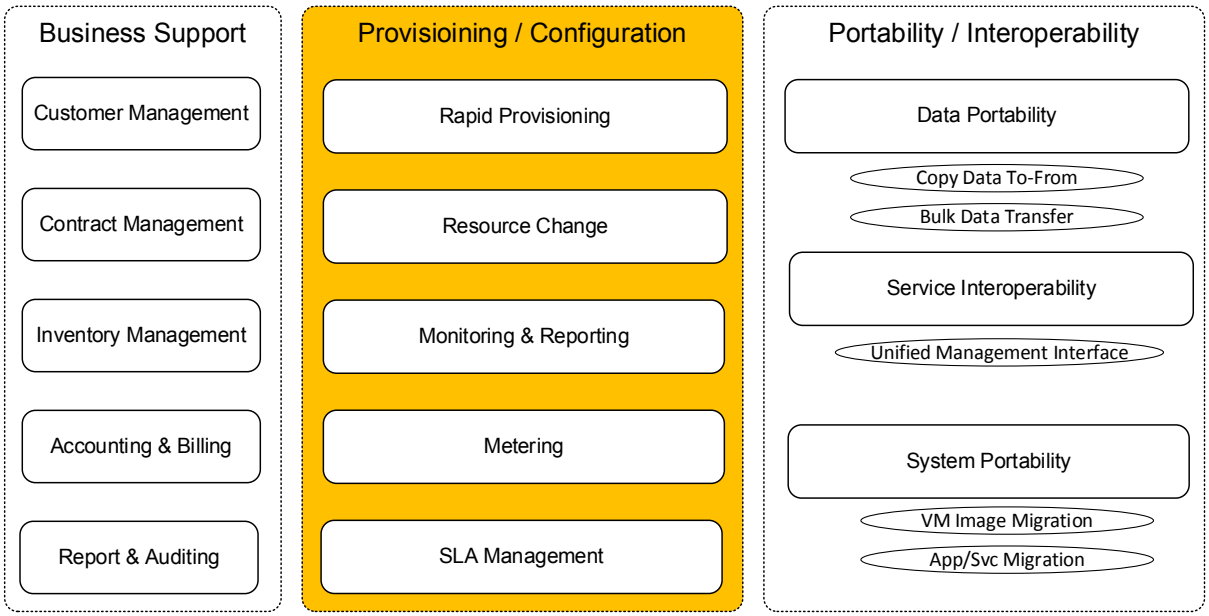

Figure 31: NIST Cloud Service Management Reference [44]

Resource and Application Scalability Mechanisms

There is a broad range of mechanisms which can typically be utilized to dynamically scale an applications' capacity for serving varying user loads. Enabling scalable mechanisms for cloud-based applications, in fact typically accounts for one of the most important and in many cases also time-consuming endeavors in order to "cloudify" legacy applications, i.e. enable legacy applications for deployment on cloud infrastructures. As a rule of thumb, the complexity of solutions for enabling scalable applications, heavily depend on the actual application complexity, whether the application logic involves several components, requires short-term or long-term state-full, session and/or service awareness. This is why in many cases it is easier to re-build an application making use of already existing powerful PaaS service enabling functions than providing application-wrapping functionalities for enabling scalability.

Typical three-tiered (presentation, logic and data tier) Web applications for instance, can usually be comparatively easily be scaled at the presentation and logic tier, whereas it becomes significantly more difficult to scale the data tier, which is responsible for maintaining state-full session and application data. In fact, the data tier where particularly databases are needed to be made scalable for long represented the true bottleneck for applications to be made fully scalable on cloud infrastructures and platforms. In the meantime several distributed database frameworks (e.g. Hadoop, Mapreduce, Cassandra) are bringing more scalability even across sites to the cloud-based data tier.

In general, two cloud infrastructure *resource scaling mechanisms* are typically differentiated; vertical scalability and horizontal scalability. Whereas *horizontal scalability* accounts for mechanisms which rely on replication of virtual instances / resource, *vertical scalability* relates to live re-sizing mechanisms of a single virtual instance / resource. Being able to dynamically and flexibly scale resource capacities without service down-times, without involving complex load-balancing application logics is a big advantage of vertical scaling mechanisms compared to horizontal scaling mechanisms. However, in contrast to horizontal scaling mechanisms, vertical scaling mechanisms are typically limited by the maximum capacity of a single virtual node / instance. Horizontal scaling mechanisms in contrast, virtually provide un-limited scalability, usually only limited by the total capacities of cloud infrastructures. Horizontal scaling mechanisms are the pre-dominantly utilized mechanisms for cloud elasticity / scalable cloud-based service architectures involving single or hierarchical load-balancing mechanisms.

Load-balancing mechanisms can be applied on different layers and levels for enabling application scalability. At the networking layer, DNS-load-balancing mechanisms are commonly used for dynamically routing service requests between users and varying numbers of slaves, i.e. application serving instances. UDP or TCP load balancers coordinate requests on the Transport layer. SIP or HTTP load balancers provide protocol specific load balancing mechanisms. Whereas for simple small-scale cloud-based web applications, it is typical to utilize single virtualized HTTP load balancing instances, for complex, potentially distributed large-scale applications, different combinations of network, transport or application level load balancing mechanisms and hierarchical load-balancing topologies might be required.

Furthermore, state-full and state-less load-balancing mechanisms are commonly differentiated. Whereas lower layered, e.g. DNS-based load balancing mechanisms are typically agnostic to application level session states, some SIP and HTTP load balancers provide session or at least transactional awareness important for assuring service continuity in dynamically changing application node topologies.

Also of importance in this context are the load-balancing algorithms that a particular load-balancing solution supports. Simple round-robin load balancing algorithms balance the incoming load by sequentially forwarding incoming requests to one after another slave node in a circular fashion. More complex load balancing algorithms for instance allow for weighted load balancing, e.g. weighted round-robin load balancing, which forwards incoming requests based on pre-defined or continuously updated weighting policies. This allows for balancing incoming load between different resource capacities of application slave's resources. Also network-aware load-balancing algorithms, where the distribution of load is conducted based on network performance metrics such as latency, or persistent algorithms are commonly used.

2.9 Multi-Cloud Service Management and Orchestration

Multi-Cloud service management, as well as Inter-Cloud service management requires new levels of cloud service interoperability and portability. Although still an area where several

standards and frameworks are being developed, it seems obvious that the future of Cloud computing will allow for highly distributed provisioning of Cloud services, where on the one hand a single user can seamlessly utilize cloud resources and services from multiple clouds (the focus of this work), where on the other hand Cloud providers join forces, establish new levels of interoperability and dynamically share resources and services in fashion that is fully transparent for the actual end-user.

Whereas standards and frameworks for end-to-end SLA management in multi-cloud scenarios as lately released by the TMForum [113], as well as end-to-end resource management frameworks for multi-cloud scenarios as lately released by the ITU-T [29] account for the bi-lateral or multi-lateral management of Cloud services (including interworking of inter-domain OSS/BSS, similar to the IPSphere initiative introduced in section). Inter-Cloud standards on the other hand, such as the IEEE's Cloud Computing Standards Committee's technical standards for cloud-to-cloud interoperability [114] and federation [115] provide means for Cloud Portability and Interoperability Profiles (CPIP) and Standards for Intercloud Interoperability and Federation (SIIF) [116].

On the way to globally accepted standards for cloud-based application portability, the Organization for the Advancement of Structured Information Standards (OASIS) standard for Topology and Orchestration Specification for Cloud Applications (TOSCA) [117], depicted in , is continuously adopted by other SDO (e.g. TMForum), as well as by de-facto standard open-source cloud management plaforms, such as OpenStack. TOSCA provides a standard for deploying and managing of applications across clouds. TOSCA supports the management of the cloud service lifecycle and enhances cloud-based application portability. By 1) providing interoperable descriptions of cloud-based application and cloud infrastructure 2) relationships between service components and 3) the operational behaviour of cloud-based services, TOSCA provides portable deployment of cloud-based applications to any cloud capable of orchestrating TOSCA-based service templates.

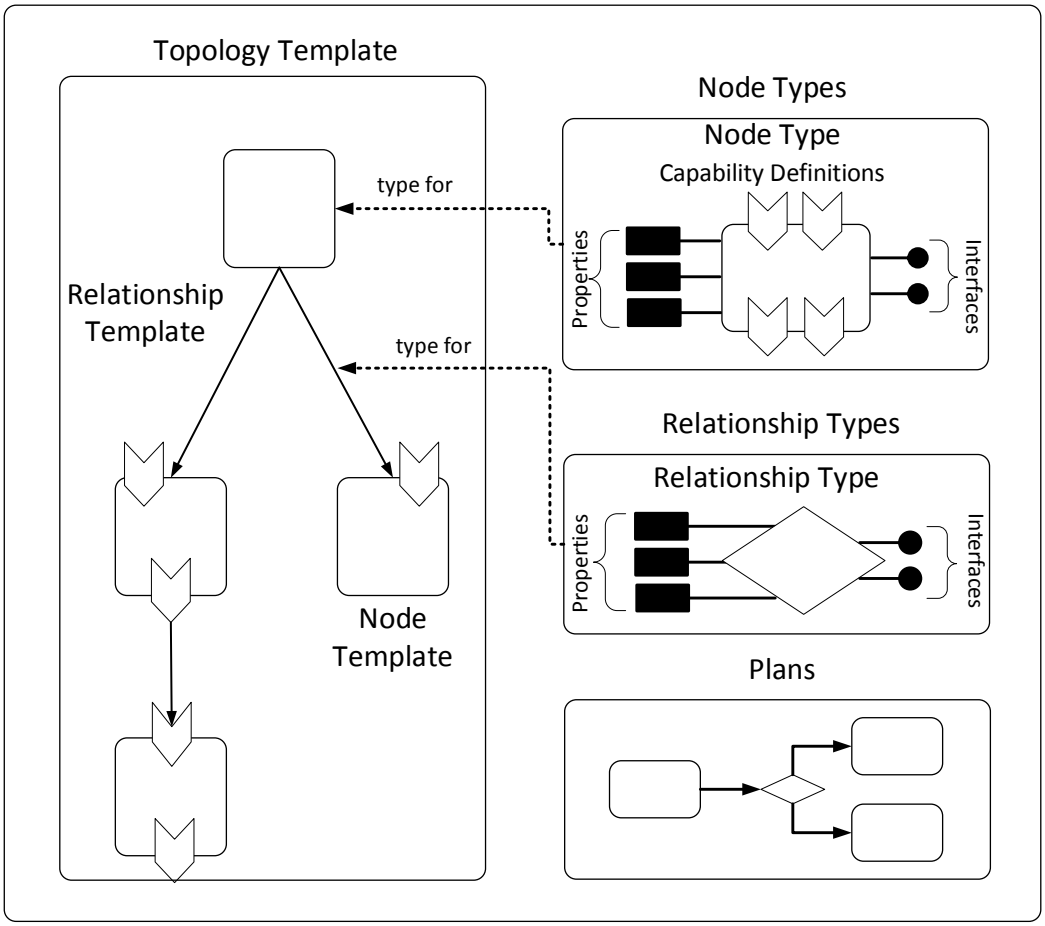

Figure 32: Topology and Orchestration Specification for Cloud Applications (TOSCA) [117]

Multi-Cloud service management is at the core of this thesis. The author investigated multi-cloud based NGN QoS assurance mechanisms in [18], highlighting the importance of network performance awareness. Dynamic cloud platform selection for multi-cloud based NGN services is investigated in [19] (nominated for best paper award IEEE CLOUDNET'12). The combination of QoS assurance with elastic multi-cloud resource provisioning / auto-scaling mechanisms for NGN services is investigated in [20], [21]. Resource allocation efficiency / capacity savings is evaluated in [22].

Cloud Platform Selection Mechanisms

The problem faced when having to find the optimal cloud platform 1) for a given NGN service having specific QoS requirements, 2) under specific cost constraints and 3) under a selection of several additional constraints (e.g. reliability, recommendation, security, location, ranking) falls into the field of multi-factorial or multi-attribute optimization or decision making mechanisms, commonly subordinated by the a sub-discipline of operations research termed multi-criteria decision analysis (MCDA) / multi-criteria decision making (MCDM). Common selection approaches can be differentiated as in [118] into evolutionary and non-evolutionary optimization approaches, as well as Multi-Attribute Utility Theory (MAUT),

Analytic Hierarchy Process (AHP) decision making processes as well as ELECTRE or PROMETEE.

2.10 Cloud Standardization Survey

This section summarizes the standardization efforts related to the different areas of cloud computing. Clearly when it comes to cloud standards, the cloud computing definitions and cloud reference architectures defined by NIST have been widely adopted, but also NIST's work on cloud security and privacy in the cloud are highly relevant in the field. Also important for the field of overarching cloud computing architecture standardization activities are the DMTF standards for cloud management architectures, including use cases for managing clouds. Particularly of relevance are DTMF's cloud infrastructure management interface (CIMI), as well as the Open Virtualization Format (OVF) standards. Relevant for bridging the gap between telecommunications and Grid/Cloud computing technologies are the standardization efforts by ETSI's TC Cloud around Gird/Cloud and ICT interoperability and SLAs for cloud services, as well as ITU-T's SG13 Cloud Focus Group's standardization on cloud computing requirements and frameworks for end-to-end multi-cloud resource management, inter-clouds for networks and infrastructures.

Similar to DMTF's CIMI, the OGF standardized the Open Cloud Computing Interface (OCCI), which in contrast to the OpenStack API, Amazon's AWS API and Google's Cloud API represents a vendor independent, extensible cloud management interface, which has been adopted by various open source cloud management platforms including OpenStack, OpenNebula and Eucalyptus. Whereas with OVF DTMF focuses on compute resources, SNIA focuses mainly on storage resources, amongst others standardizing the Cloud Data Management Interface (CDMI), and network functions virtualization is mainly driven by ETSI, with protocols standardized by the IETF.

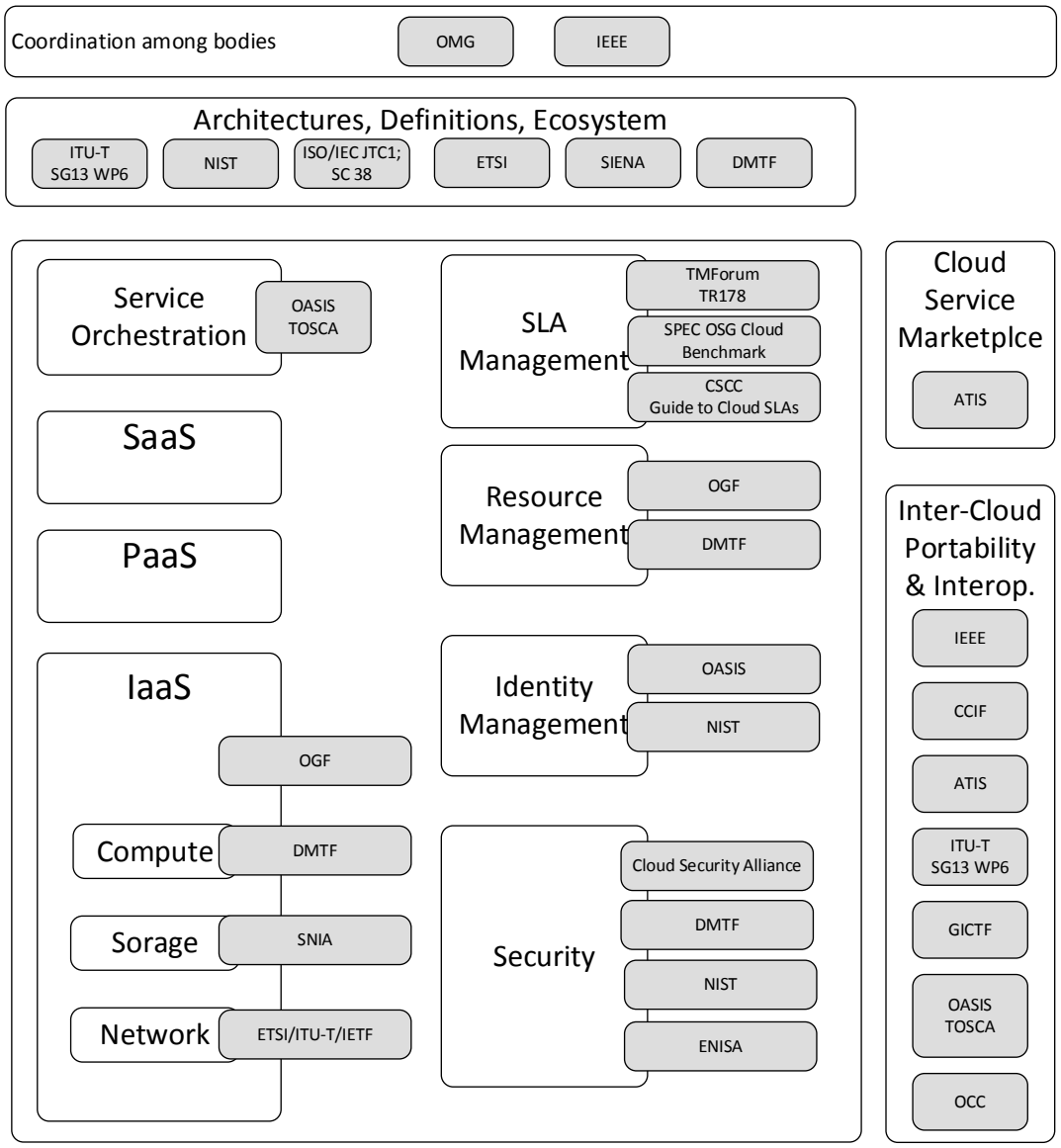

Figure 33: Cloud Standard Definition Organizations and related standardization areas

In the area of SLA management, the TMForum has published several technical reports focusing on multi-cloud service management and on management of cloud service SLAs. The practical guide to cloud SLAs released by CSCC provides useful summary of factors to take care about when deploying services on clouds for assuring SLAs. The OSG Cloud SPEC Benchmark is developing benchmarks for measuring IaaS services, including agility and elasticity.

Regarding cloud identity management, the OASIS CloudTC has produced several use-cases for identity in the cloud and NIST provided architectures and mechanisms for identity management as well as requirements for identity management in cloud computing.

In the area of security, NIST is working on cybersecurity, SOA security architectures, functional requirements for SaaS and cloud-based telecommunication services, as well as

security guidelines for cloud computing in the telecommunication area. CSA identified the top security threats to cloud computing, critical areas, security cloud audit. DMTF specified cloud management interface requirements on Security. ENISA developed a framework for cloud computing information assurance and specified risks and recommendations for information security in cloud computing.

Regarding Inter-Cloud portability and interoperability, the IEEE is working on standards for Intercloud interoperability and federation, cloud portability and interoperability profiles. ATIS works on data models for service enablers driving the Intercloud demand. The GICTF released interface specifications for inter cloud computing. The Open Cloud Consortium (OCC) works on standards and frameworks for interoperating between clouds, develops benchmarks and supports reference implementations. With TOSCA, OASIS enables Intercloud portability of applications, multi-cloud service provisioning, including multi-cloud resource scaling. By providing semantic cloud abstractions, the CCIF supports cloud interoperability.

2.11 Related Research Projects

In the past couple of years a broad range of Cloud-related European-wide and national research projects have been carried out. Whereas early projects, such as EU FP7 RESERVOIR [119] and StratusLab [120] developed IaaS Cloud management platforms and tools with only thin service management layers, subsequent projects such as OPTIMIS [121] and Contrail [122] significantly put more emphasis on Cloud federation, Cloud SLA management and Cloud brokering and Cloud service lifecycle management. Of particular relevance for the scope of this work are the RESERVOIR project, providing the IaaS management basis, the OPTIMIS project, providing a solid framework for Multi-Cloud, SLA-aware Cloud service brokerage and the BonFIRE [123] project providing a large scale, pan-European multi-site Cloud environment, which was extensively made use of for carrying out real-world evaluations of the resource efficient Cloud brokering system for NGN service developed in the context of this work.

More details on this work's similarities with aforementioned related research projects can be found in Related Research Projects.

2.12 Open Source Cloud Management Solutions

At the onset of this work, the cloud infrastructure management Eucalyptus [124] and OpenNebula [125] were the de-facto standard cloud infrastructure management toolkits predominantly used by the research community. OpenNebula, is used as a virtualization tool managing virtual infrastructures providing cloud interfaces for virtual machine, storage and network management. OpenNebula being the first open-source cloud management platform adopting the OCCI [111] standards was utilized in the majority of EU research projects, such as RESERVOIR [119], OPTIMIS [121] and BonFIRE [123] (see section 2.11). The OCCI

service is provided on top of the OpenNebula Cloud API (OCA) layer which exposes the capabilities of an OpenNebula-based cloud, including the Sinatra web framework.

However, with the onset of the huge and ever since growing OpenStack development community (starting with the OpenStack "Austin" release in October 2010), OpenStack [126] rapidly became the de-facto open source cloud management toolkit not only taken up by the research community, but increasingly also used in several commercial deployments. Whereas the OpenStack "Cactus" release [127] (simplified architecture shown in Figure 35), comprised the core elements of a typical virtual element execution environment, "Glance" providing a catalogue and repository of virtual images, "Swift" providing virtual object storage and "Nova" providing the actual VM provisioning service, later releases such as the OpenStack Folsom release provide complex network management capabilities (incl. OpenFlow control) through the "Quantum" tool, block storage management through a tool called "Cinder", identity management through a tool called "Keystone" and a telemetry service provided by a tool called "Ceilometer".

Of particular relevance for the scope of this work is the just lately released OpenStack "Heat" service, an orchestration tool multiple composite cloud applications, which also includes auto-scaling capabilities. It is expected that with the evolution of "Heat", future OpenStack-based cloud management solutions provide many cloud service orchestration and scaling capabilities investigated in this work.

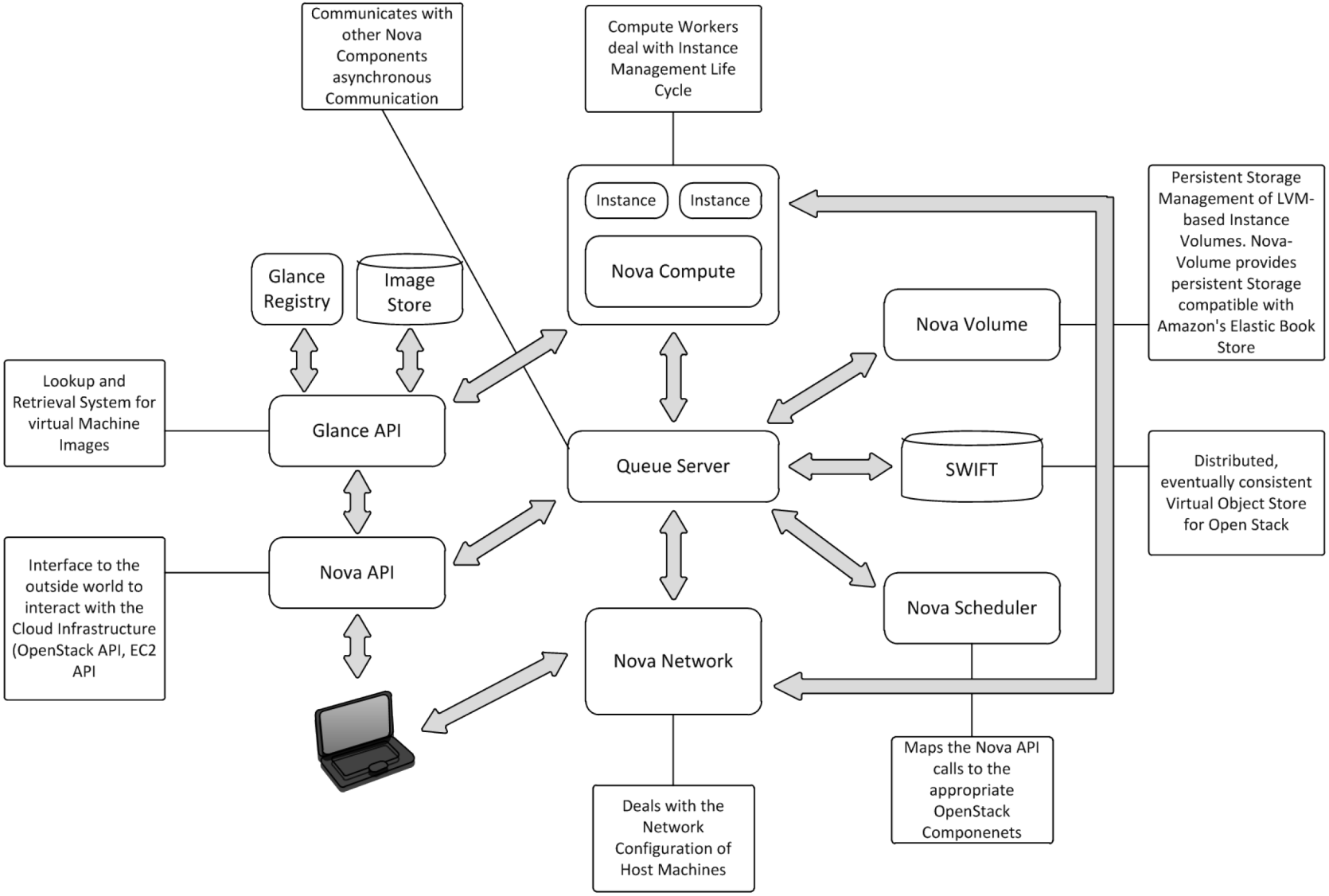

Figure 34: Simplified OpenStack “Cactus” architecture bsd. on [127]

Chapter 3

Challenges and Requirements

This section first explores the general challenges for Resource Efficient QoS Management of NGN Services in Federated Cloud Infrastructures, and subsequently identifies the requirements from the perspective of Cloud Infrastructure Provider, NGN Service Provider, NGN Operator, Cloud Broker and Cloud-based NGN service User.

3.1 Overview of Challenges

For NGN service providers, the overall challenge is to provide NGN services at traditional QoS levels deployed on multiple cloud platforms at minimal costs. For NGN service users it should be fully transparent, whether the service is deployed and operated on local infrastructures or on federated cloud infrastructures.

This however imposes a number of challenges to the cloud federation and brokering providing system, whether it is maintained locally at the premise of the NGN operator, the NGN service provider or provided as a service by a third party.

3.2 Properties of Key Interest of Cloud Federation Brokerage

There are mainly two properties of key interest, being the core measures for evaluating the performance of "Resource Efficient NGN Service Quality Management", namely:

I. Resource Allocation Efficiency Performance
II. QoS Assurance Performance

For I, measurable metrics could be 1) costs (measured in units of currencies), or energy (measured in units of Watts) or consumed resource capacity, measured in units of compute/CPUs, storage/GByte, or network/GBit multiplied with the duration of their utilization in units of time. Aforementioned compute, storage and network substrates are offered as so called cloud instances in a "packaged" form, which could be measured in compute units e.g. Amazon's *EC2 Compute Unit* (*ECU*) [128]. However the ECU is neither globally standardized, nor, as will be shown, sufficient for comparing cloud instance capacities. *In this work resource allocation performance is determined by measuring the utilized capacities (in units of cloud instances) multiplied by their duration of use (unit of time) and a rough mapping to costs using current market prices is provided.*

For II, measureable metrics/units are service availability (measured in percentage of successful service requests per unit of time), service execution time (measured in units of

time). In the domain of NGN services, there are additional service-specific QoS metrics, as for instance Voice over IP (VoIP) quality, measured in units of Mean Opinion Score (MOS) [129] scales or Perceptual Evaluation of Speech Quality (PESQ) [130] scales, or video quality, measured in units of Perceptual Evaluation of Video Quality (PEVQ) scales. *In this work focus is put on voice quality metrics, network performance metrics (i.e. packet loss, jitter and packet delay) and service availability.*

3.3 Categorization of Requirements

For cloud brokering mechanisms to efficiently allocate cloud resources and assure service qualities, there are a number of requirements to be fulfilled by the service providing entity, the network providing entity, and the cloud infrastructure providing entity shown in Figure 35.

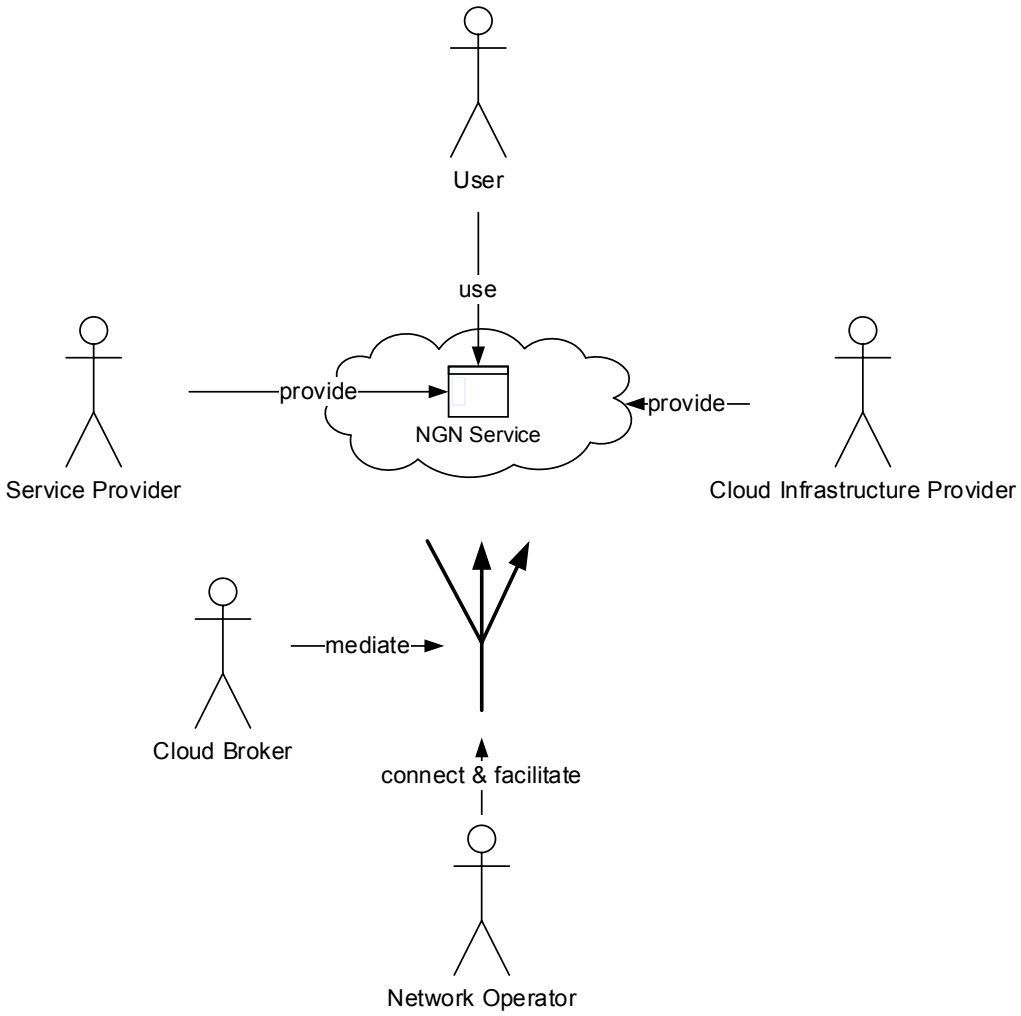

Figure 35: High-level view of entities involved in NGN-based cloud brokering scenarios

The following sections provide an overview of high-level requirements that need *to be fulfilled from* aforementioned stakeholders. Requirements for satisfying the *needs of the stakeholders* are provided in the section of cloud federation broker perspective.

3.3.1 Cloud Infrastructure Provider Perspective

Cloud IaaS Providers eligible to participate in a Federation are required to provide certain functionalities and information about their infrastructure resources, some of which are absolutely mandatory whereas others can be mitigated. Those requirements, defined in Table 6, that are absolutely needed, i.e. critical requirements are specified with a MUST, those that can be mitigated are marked with a SHALL.

Table 6 Requirements – Cloud Provider Perspective

Requirement	Description
Resource Provisioning Interface	Cloud Infrastructures MUST provide open interfaces for remote provisioning of cloud resources (compute storage, networking)
Resource Description	Reserve-able Cloud Infrastructure Resource Instances Types MUST be specified by the Cloud Infrastructure Provider
Instance Prices	Instance Prices MUST be specified by the Cloud Infrastructure Provider
Minimal Instance Lease Time	Minimal Instance Lease Times MUST be specified by the Cloud Infrastructure Provider
Limits of available Numbers of Instances	Limits of available numbers of Instances SHOULD be specified by the Cloud Infrastructure Provider
OS Types, VM Image Import	Supported Operation System Types and / or VM Image import mechanisms MUST be specified by the Cloud Infrastructure Provider
Service Level Agreements	Service Level Agreements, particularly instances SHOULD be specified by the Cloud Infrastructure Provider
Provisioning Delay	For each instance type, time needed for provisioning SHOULD be specified by the Cloud Infrastructure Provider

3.3.2 NGN Service Provider Perspective

NGN Service Providers who intend to deploy their applications on federated Cloud Infrastructures are required to provide certain functionalities and information about their applications, some of which are absolutely mandatory, whereas others can be mitigated. Those requirements, as defined in Table 7, that are absolutely needed, i.e. critical requirements are specified with a MUST, those that can be mitigated are marked with a SHALL.

Table 7 Requirements - NGN Service Provider Perspective

Requirement	Description
Service Description	NGN Service Control Protocols MUST be specified by the NGN Service Provider NGN Service Control Ports MUST be specified by the NGN Service Provider NGN Service Trigger / Invocation Methods MUST be specified by the NGN Service Provider
Service QoS Requirements	Service QoS Requirements MUST be specified by the NGN Service Provider
Resource Requirements	Service Resource Requirements MUST be specified by the NGN Service Provider

3.3.3 NGN Operator Perspective

NGN Operators not only provide basic connectivity to NGN service users, but also the necessary NGN service control mechanisms by which NGN services can dynamically be accessed and the service sessions can be controlled. Furthermore, in advanced deployment modes, NGN Operators are able to utilize QoS control mechanisms through which service can communicate their requirements to NGN networks by which several network functions can dynamically be enabled, e.g. QoS and security functions. The following Table 8 lists NGN functionalities that are absolutely mandatory, i.e. critical requirements, specified with a MUST, and those that can be mitigated, marked with a SHOULD.

Table 8 Requirements - NGN Operator Perspective

Requirement	Description
User Authentication and Authorization	Mechanisms for authenticating Users and User Endpoints MUST be provided by the NGN Operator Mechanisms for authorizing Users to access specific services MUST be provided by the NGN Operator
Session and Service Control	Service control mechanisms through which NGN users are dynamically connected to changing service end-points, i.e. dynamically changing cloud locations of NGN services MUST be provided by the NGN Operator.
NGN Provisioning Interfaces	NGN Provisioning Interfaces, through which dynamically changing service end-points can rapidly be configured at the NGN service control level MUST be provided by the NGN Operator
QoS Control	NGNs QoS management interfaces, through which the network QoS can dynamically be adapted to services needs SHOULD be provided by the NGN Operator

3.3.4 NGN Service User Perspective

To telecommunication service users, it should be *fully transparent* whether services are hosted locally, at the premises of NGN service providers or remotely at distributed cloud infrastructures. For utilization of NGN services, standard fixed-line authentication or mobile / subscriber-identity-module (SIM) / device-based network registration mechanisms apply.

Table 9 Requirements - NGN Service User Perspective

Requirement	Description
Network Subscription	NGN service users are required to subscribe to NGN service usage in the traditional way of making contacts with fixed or mobile NGN service providers
Network Registration	Users, by traditional means of SIM / device-based network registration are required to connect to and authenticate against fixed or mobile NGN networks

3.3.5 Cloud Broker Perspective

Cloud Brokers, whether acting as independent entities or need to provide a certain set of mandatory functionalities. Those functionalities, as defined in Table 9 that are absolutely needed, i.e. critical requirements are specified with a MUST, those that can be mitigated are marked with a SHALL.

Table 10 Requirements - Cloud Broker Perspective

Requirement	Description
Core Functions:	
Platform Selection	Cloud Brokers MUST provide means allowing for dynamic selection of Cloud Platforms and Cloud Resources according to pre-defined rules, evaluated against real-time monitoring data and enforced repeatedly.
Resource Allocation	Cloud Brokers MUST provide means allowing for the dynamic allocation of Cloud resources across multiple cloud platforms according to pre-defined rules, evaluated against real-time monitoring data and enforced repeatedly.
Support Functions:	
Cloud Infrastructure Catalogue	Cloud Brokers MUST maintain a Cloud Infrastructure Catalogue which lists available Cloud Infrastructures, potentially eligible for telecommunication service to be deployed on.
Service Registry and Repository	Cloud Brokers MUST maintain a Service Registry, which stores the necessary information describing the service's QoS requirements, describing service control protocols, ports, invocation methods. A service repository must be in place which is either maintained locally, or can be accessed remotely, which stores the application software.
Monitoring Functions	Cloud Brokers MUST comprise versatile Monitoring *Aggregation Functions*, aggregating Monitoring information from distributed NGN Services, Cloud Infrastructures, Cloud Infrastructure Resources and Networks that interconnect NGNs and Cloud Platforms. Cloud Brokers MUST provide active and passive Monitoring Functions through which active and passive measurements of Network, Infrastructure, Resource and Service Performance can be conducted. Cloud Brokers SHOULD provide Cost Monitoring Functions through which they can monitor dynamically changing prices of e.g. Spot Cloud Resources.
Provisioning Mechanisms	Cloud Brokers MUST provide the means for the dynamic provisioning of Cloud Infrastructure Resources, Service Elements (e.g. required for Service Scalability such as Load Balancers) and NGN Service control Elements (e.g. IMS HSS).
Pre-Requisites:	
Interconnection	Connectivity for interconnecting Cloud Brokers with NGN platforms and Cloud Platforms MUST be provided. Network performance between aforementioned interconnections, SHOULD be of "good" quality.

3.4 Evaluation Criteria

After having defined the functional requirements, i.e. what the cloud brokering system is supposed to do, for the evaluation of the proposed cloud brokering system a set of *performance evaluation criteria* as well as a set of architectural, *non-functional evaluation criteria* are defined in this section.

Performance Criteria / KPIs against which the proposed cloud brokering system will be evaluated are listed in Table 11.

Table 11 Performance Evaluation Criteria

Performance Evaluation Criteria	Description
Capacity Saving Efficiency	The Cloud Brokering system should enable *High Capacity Saving Efficiency*, low Overprovisioning – High Cost/Energy Saving, compared to non-elastic and ideal capacity saving / usage.
QoS Assurance Performance	The Cloud Brokering system should enable *High QoS Assurance Performance Levels* – matching standardized QoS service classes (ITU-T) and common SLAs (service availability) and NGN service / voice quality levels.

Non-functional, architectural criteria, i.e. properties of the system, defining how the cloud brokering system is supposed to be, against which the proposed cloud brokering system will be evaluated are listed in Table 12.

Table 12 Non-functional Evaluation Criteria

Non-functional Evaluation Criteria	Description
Execution Qualities	
NGN QoS Awareness	The cloud brokering system should not only be aware of resource utilization but also ware of the end-to-end service quality of managed NGN services.
Network Performance Awareness	The cloud brokering system should not only be aware of resource utilization but also ware of network performance between NGN and Cloud Platforms.
Fault Tolerance	The nature of the cloud brokering system's distributed operations, across domains and platforms, requires a high degree of fault tolerance, as failures in such federated environments are commonplace, especially if rather unreliable platforms participate.
Reliability	Failure of the cloud brokering system has a direct, negative impact on NGN service's availability and QoS. Therefore the cloud brokering system should provide high reliability.
Evolution Qualities	
Maintainability / Configurability	Required efforts for system configuration changes needed to adapt the behavior of the cloud brokering system for an existing service or for configurations needed for the management / brokering of new services should be kept as low as possible.
Platform Independence	The Cloud Brokering system should aim at supporting interworking with as many cloud platforms as possible, in order to benefit from a rich set of alternative cloud resource offerings, provided at different prices and at different QoS levels.
Scalability	With growing numbers of services to be brokered across cloud infrastructures, as well as with growing user demand and workloads, scalability of the Cloud Brokering system becomes critical. Thus, the cloud brokering system should be designed in an intrinsically scalable fashion, allowing on-demand scalability.
Service Agnosticism / Versatility	Core mechanisms of the cloud brokering system should be designed in a highly service agnostic manner, allowing versatile utilization of the system for a broad range of services.
Portability	Core mechanisms of the cloud brokering system should be deployable in different contexts and environments. To this end, the system should be designed in a highly portable fashion.
3rd Party Dependency	The cloud brokering system should be designed in a highly independent fashion, allowing for operation inside of a particular stakeholder domain, but also outside, provided as a services to multiple stakeholders / NGN service providers.
Extensibility	Core Mechanisms of the cloud brokering system should be extensible enabling its application in other contexts and other environments or environments requiring more complex capabilities.

Chapter 4

Entities, Relationships and Scenarios

This chapter introduces the conceptual model for brokerage of cloud infrastructure and resources for NGN service provisioning in federated cloud environments. It introduces the basic *model entities* and *domains* that are later used in the methodology chapter. After introducing the model entities, this chapter provides models for *actors and their roles* related to cloud infrastructure federation brokerage, federation brokerage *modes* and federation brokerage *scenarios*.

4.1 Entities

For the sake of better understanding the upcoming chapters, the parts of the initial hypothesis of this thesis, relevant to the cloud federation brokerage model are recapitulated here:

I. In order to efficiently allocate distributed cloud infrastructure resources for quality assured telecommunication service delivery

II. a cloud infrastructure federation brokerage model can be designed that allows to define a cloud resource federation brokerage methodology and framework based on which an according system solution can be instantiated

From a general conceptual perspective, as shown in Figure 36, three different domains are differentiated:

- Network Domain: An NGN-based core network, capable of converging multiple access networks (e.g. fixed, mobile, cable, satellite)
- Service Domain: An NGN-based service control and service delivery environment
- Cloud Domain: Private or public cloud infrastructures

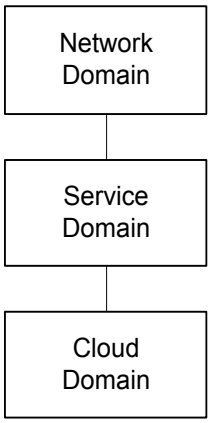

Figure 36: Network, Service and Cloud Domain

As shown in Figure 37, the following conceptual entities are introduced:

- S – NGN Service Entity:

  An application provided to NGN users / subscribers as an on-demand consumable, chargeable service, typically a voice/video/messaging telecommunication service

- SC – NGN Service Control Entity:

  A platform dynamically connecting NGN users / subscribers with the requested services and controlling the invoked session between one, two or multiple parties

- SD – NGN Service Delivery Entity:

  A platform for the delivery of NGN services, proving access control and intermediation between service requesters and services

- CM – Cloud Management Entity:

  A platform for the management of Cloud-based services, allowing for the dynamic provisioning of Cloud resources through remotely accessible cloud management interfaces

- R – Cloud Resource Entity:

  A virtualized element / instance of the pool of resources a cloud platform offers to its users

- BRO – Cloud Resource Brokering Entity:

  A platform dynamically allocating cloud resources for NGN applications and orchestrating NGN and NGN service functions

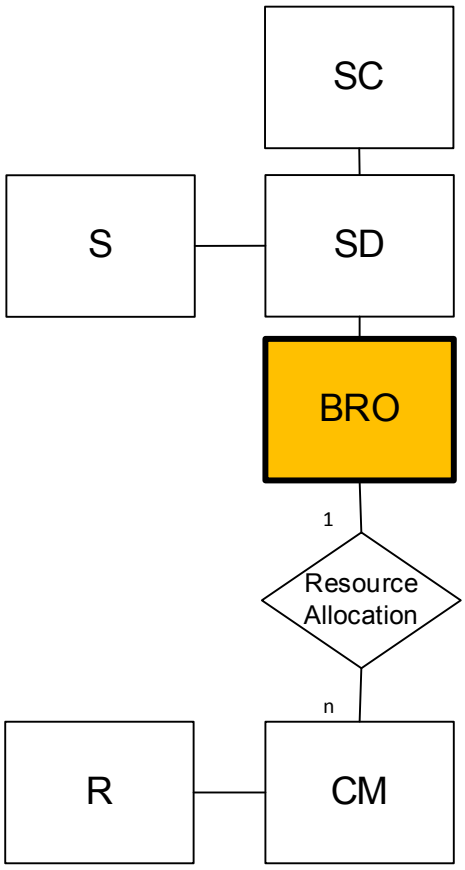

Figure 37: Overview of entities in NGN-based cloud infrastructure and resource federation brokerage

4.2 Actors and Roles

Similar to the NIST differentiation of actors in Cloud computing environments, as described in section 2.5, but differentiating between the final user of the NGN service and the user of Cloud resources, i.e. the NGN service provider the following actors and roles as shown in Figure 38 are identified:

- NGN Network Operator:

  The carrier / NGN network provider interconnecting NGN service users with NGN services

- NGN Service User:

  The final consumer of NGN services, typically subscribed to an NGN operator's network services, against which he authenticates with the help of fixed or mobile end-devices

- Cloud Infrastructure Provider:

  The private or public provider of Cloud infrastructure resources, i.e. an IaaS provider

- NGN Service Provider:

  The provider of NGN Services, which not necessarily needs to be identical with the NGN operator, thus could be acting as a third party, deploying NGN services on third party cloud resources and offering latter to users

- Cloud Infrastructure Provider:

  The private or public provider of Cloud infrastructure resources, i.e. an IaaS provider, using the carrier's network for offering cloud resources to third parties, which in the case of this work, are not the final end-users, but NGN service providers.

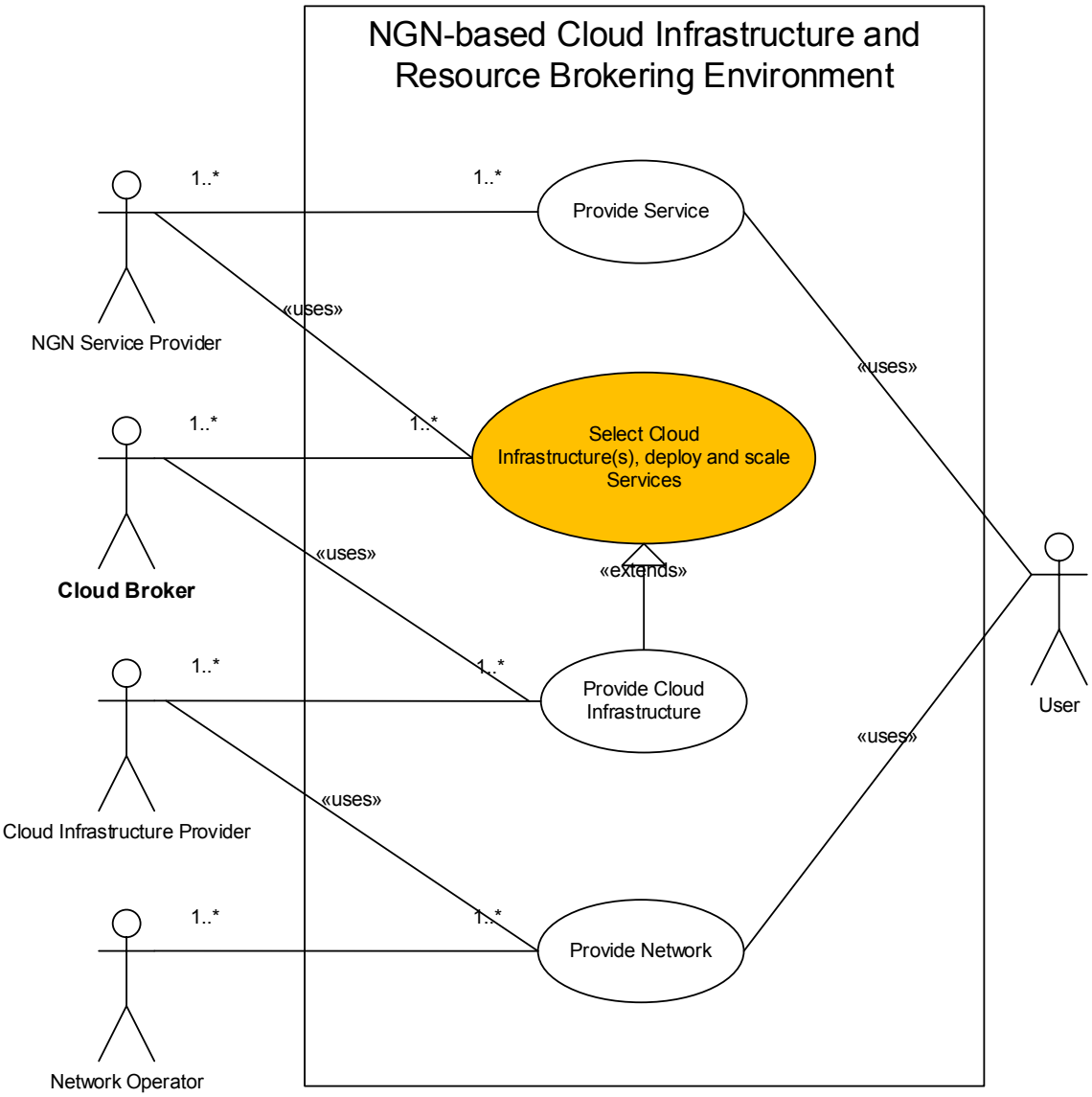

Figure 38 Actors and Roles in NGN-based Cloud Infrastructure and Resource Brokering Environment

4.3 Modes

4.3.1 Private Cloud - Elastic Cloud Resource Management

In a scenario, where network, service and cloud domain reside within a single private infrastructure of an NGN operator, the Cloud Broker is utilized in its simplest mode of operation. Here the Cloud Broker's task is limited to private cloud resource scaling and the according service orchestration in order to serve current workloads.

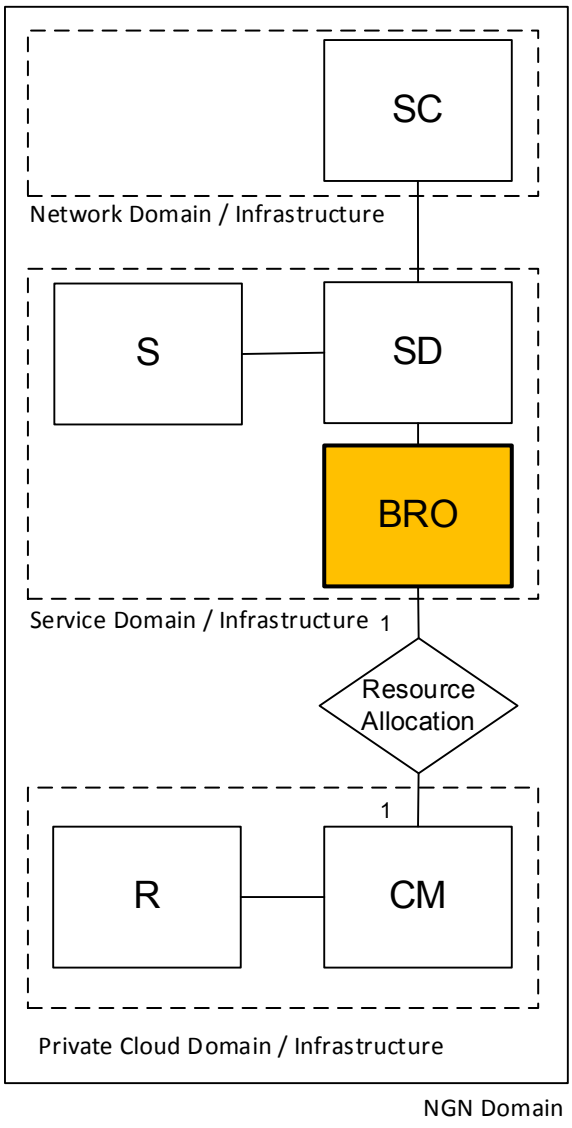

Figure 39: Private Cloud Infrastructure – Elastic Cloud Resource Allocation Mode

4.3.2 Public Cloud – Elastic Cloud Resource Management

Elastic Cloud resource management mechanisms of the Cloud Brokering Entity allow for dynamic scaling of public cloud infrastructure resources according to current workload requirements. In contrast to surrendering the control of resource allocation to a public cloud infrastructure / IaaS provider, the Cloud Brokering Entity, as part of the NGN service domain,

controls the dynamic and elastic allocation of required cloud resources. However, in this scenario, a single, statically utilized public Cloud infrastructure is being used.

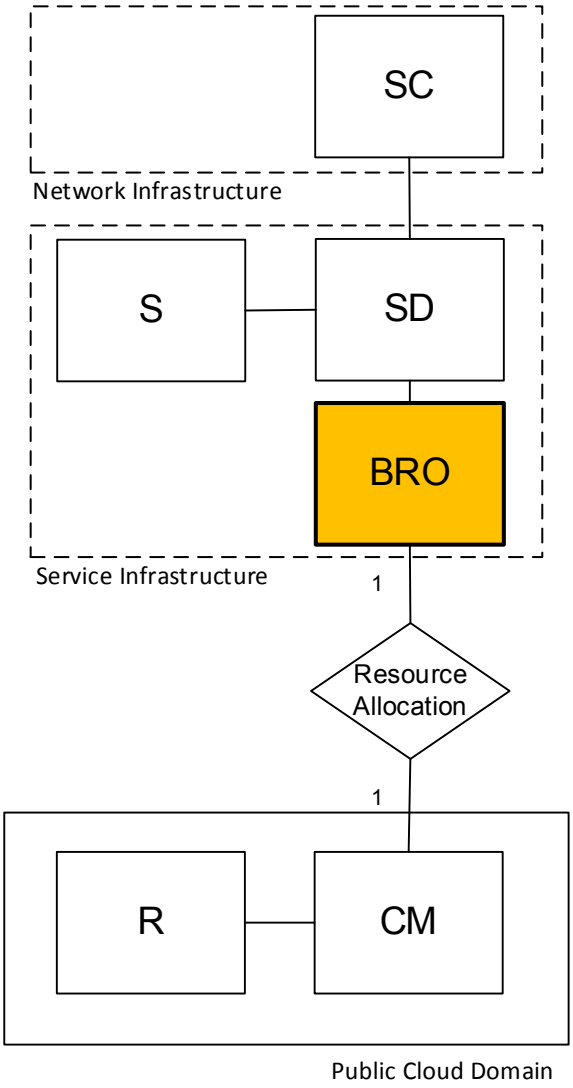

Figure 40: Single Public Cloud – Elastic Cloud Resource Allocation Mode

4.3.3 Hybrid Cloud – Elastic Infrastructure Resource Management and Cloud Bursting

Elastic Cloud resource management mechanisms of the Cloud Brokering Entity allow for dynamic scaling of a combination of private and public cloud resources according to current workload requirements. In such a hybrid cloud scenario, as depicted in Figure 41, the Cloud Brokering Entity controls the dynamic and elastic allocation of required cloud infrastructure resources between a private and a public cloud infrastructure. Typically this mode is chosen to outsource required capacities on-demand. Whereas during typical, moderate workload situations, mainly local cloud resources are being allocated and utilized for serving current NGN service workloads, outsourcing takes place during peak load situations, during which local capacities are insufficient to cope with accruing workloads. In this mode, the Cloud Brokering entity supports so called “cloud bursting”.

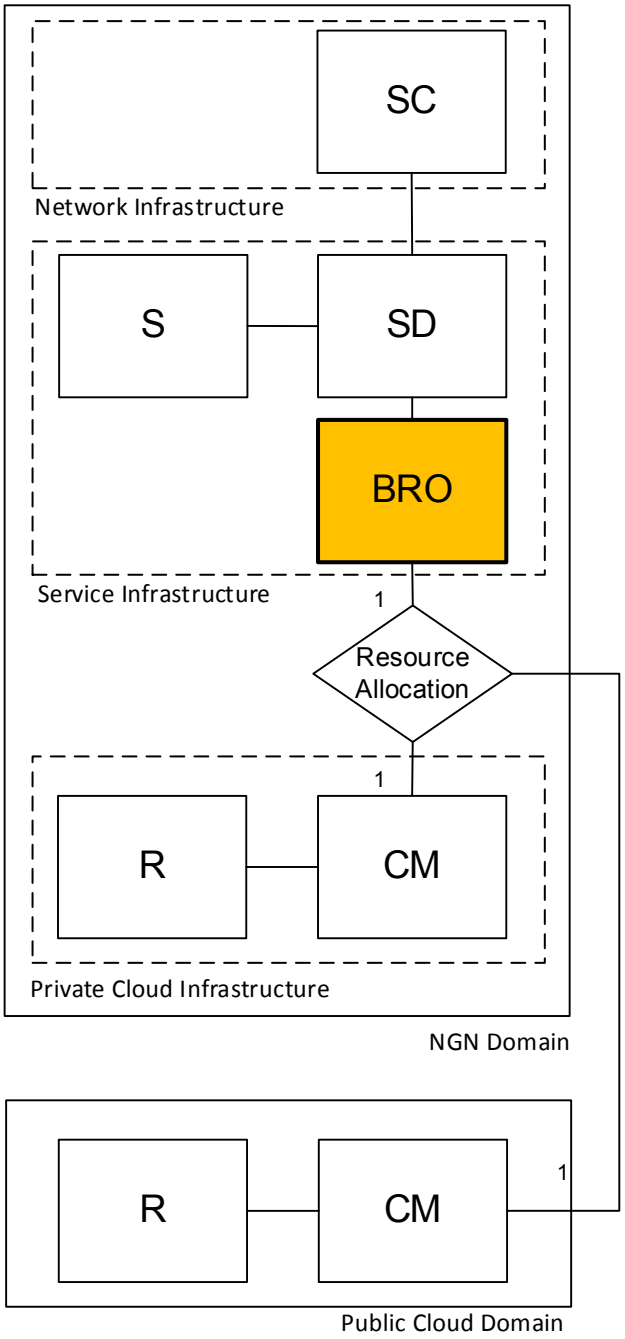

Figure 41: Hybrid Cloud Elastic Cloud Resource Allocation - Cloud Bursting Mode

4.3.4 Cloud Federation – Elastic Cloud Resource Brokerage

Elastic Cloud resource allocation mechanisms of the Cloud Brokering entity allow for dynamic scaling cloud resources across multiple public Cloud infrastructures according to current NGN service workload demands. In this mode, as depicted in Figure 42, the Cloud Brokering entity also *dynamically selects Cloud infrastructures* and subsequently selects and allocates cloud resources for serving current workload demands. This mode includes dynamic service migration across multiple public Cloud infrastructures during service delivery run-time, allowing for cost and performance optimization and fault management.

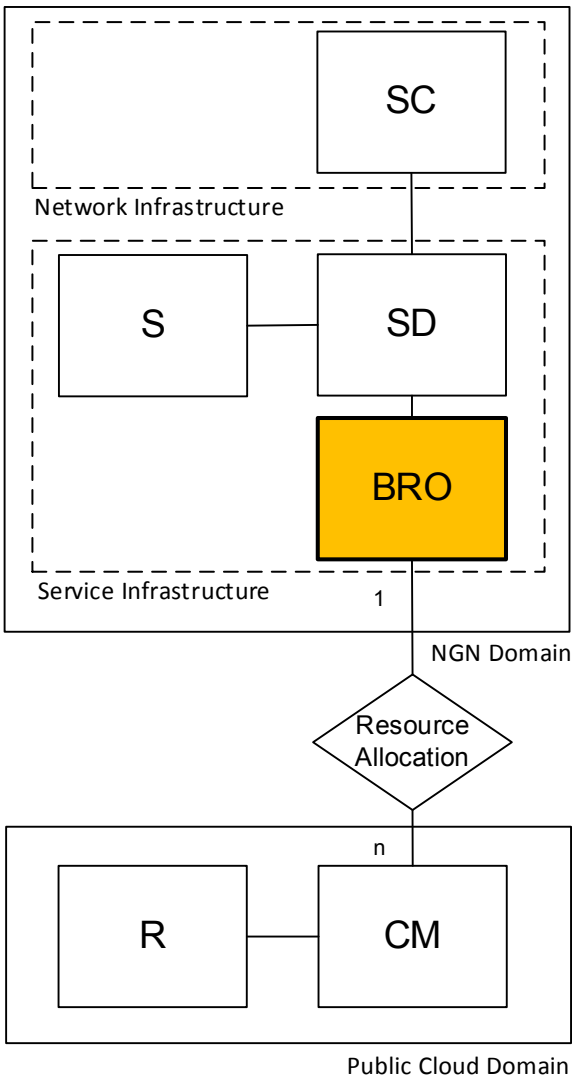

Figure 42: Cloud Federation – Elastic Infrastructure Resource Brokerage Mode

4.4 Scenarios

The following Cloud Brokering entity deployment scenarios are differentiated:

1) NGN Deployment Scenario
2) External Service Domain Deployment Scenario
3) Autonomous Deployment Scenario

4.4.1 NGN Deployment Scenario

In the NGN-centric deployment scenario, as depicted in Figure 43, the Cloud Brokering entity is deployed and operated at the premises of the network operator. The network operator, who also operates an NGN service environment, operates and controls the Cloud Brokering entity. In this scenario, the converged, hybrid resource management of local, private cloud resources as well as public cloud resources is enabled by the Cloud Brokering entity. This scenario

significantly reduces complexity and increases reliability of overall signaling and communication mechanisms, as most of the control messages for provisioning the BRO and SC entities can be controlled and assured within the domain of the NGN Operator.

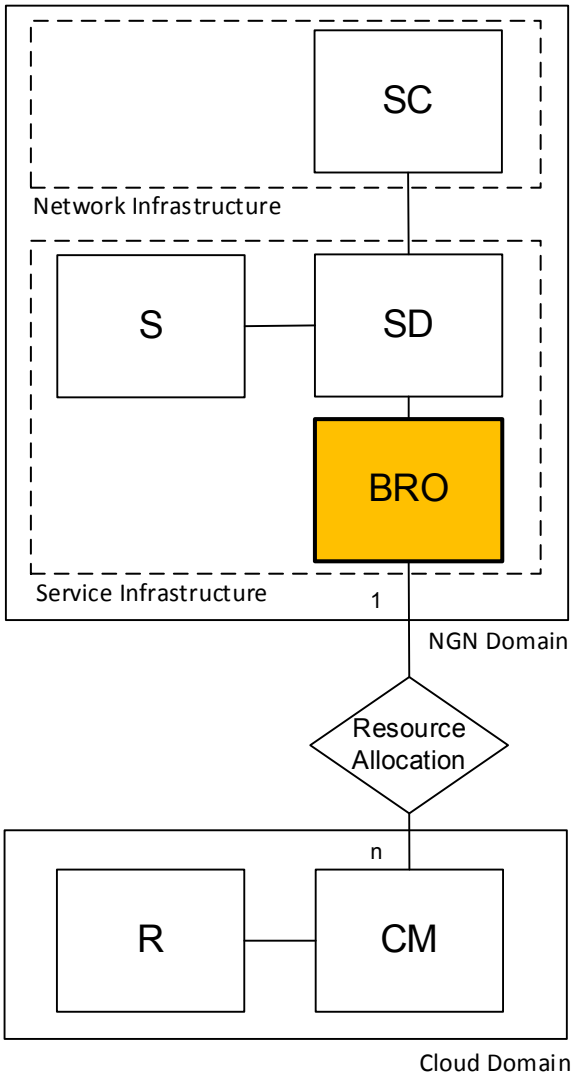

Figure 43: NGN-centric Deployment Scenario

4.4.2 Service Domain Deployment Scenario

There are two potential service domain deployment scenarios for the Cloud Brokering entity, where the NGN domain is separated from service domain.

In the first scenario, as depicted in Figure 44, the NGN operators are operating NGN service control platforms independently interworking with the BRO deployed on the premises of the Service Provider. This allows the Service Provider to act as an entity independent from the NGN Operator, allowing the provision of Cloud-based NGN services to the users / subscribers of multiple Network Operators.

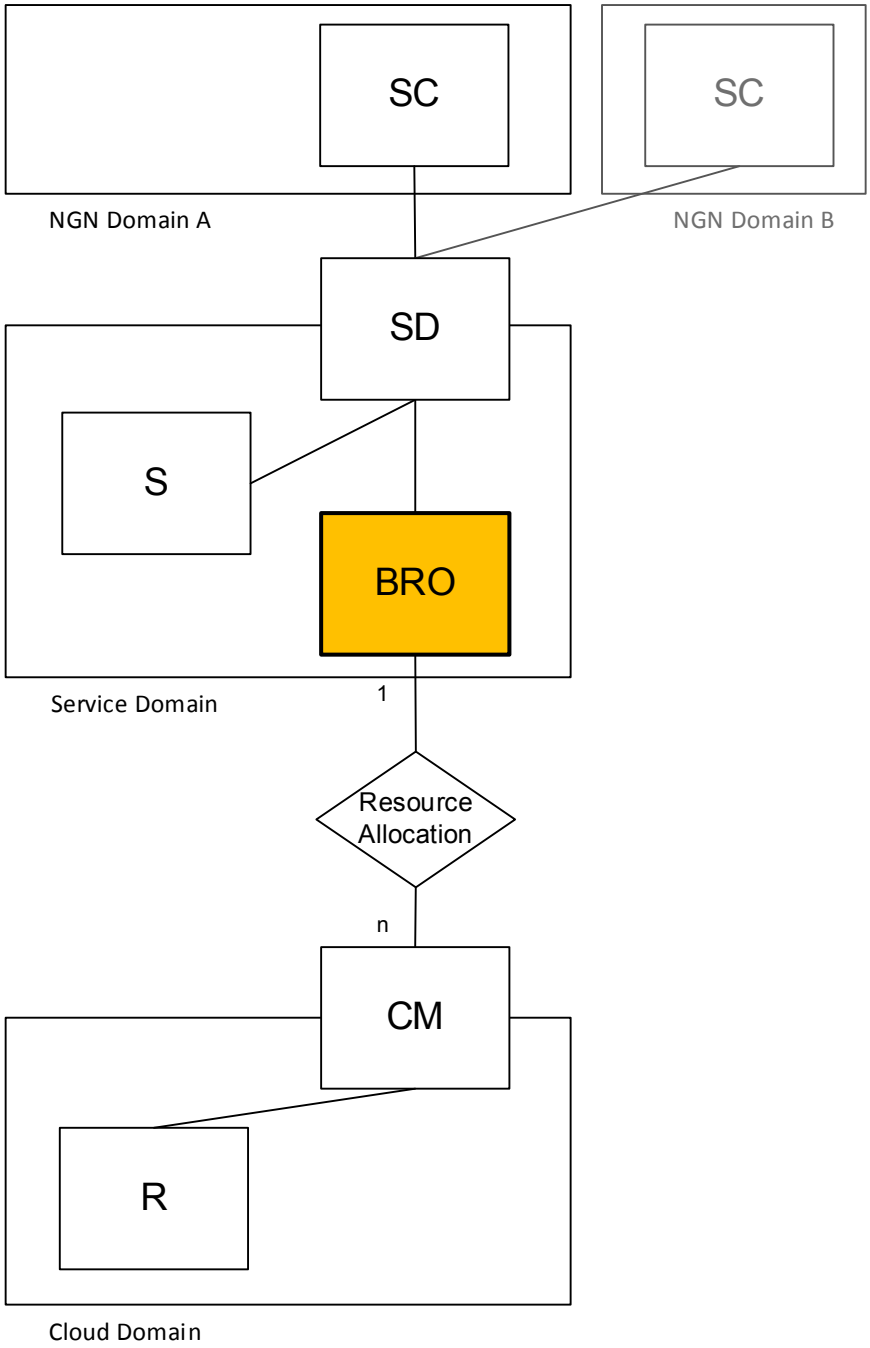

Figure 44: Service Domain Deployment Scenario

One special case, in this scenario, depicted in Figure 47, is the full outsourcing of the SC entity (traditionally residing at the premises of NGN network operators), to the domain of the NGN service provider. This scenario requires full surrendering of service control from the NGN operator to the NGN service provider domain.

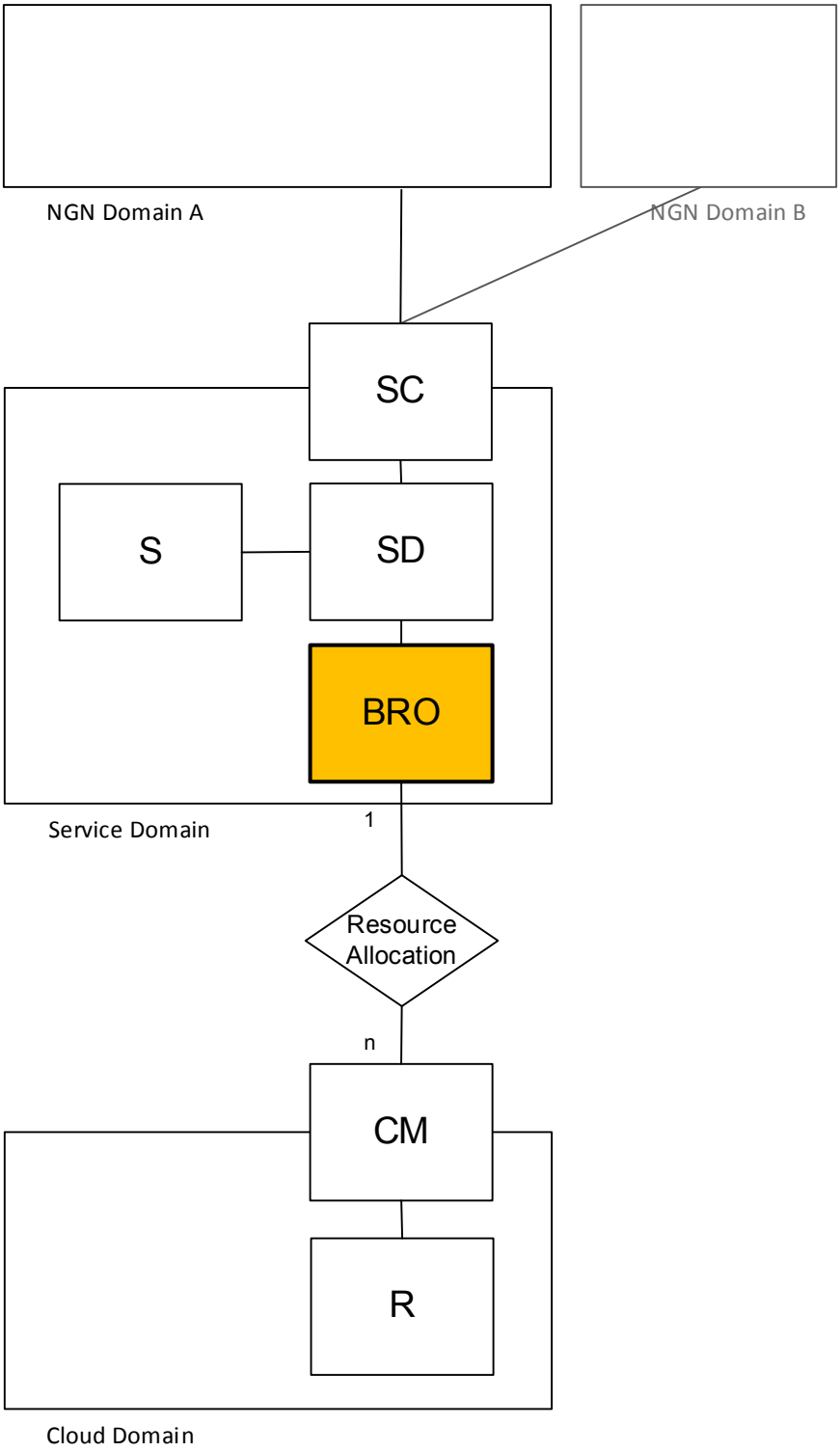

Figure 45: Service Domain Deployment Scenario - NGN Service Control Entity Outsourcing

4.4.3 Autonomous Deployment Scenario

In the autonomous deployment scenario, depicted in Figure 46, the BRO entity is operated and provided as a service by a third party, independent from NGN Operator, the NGN Service Provider or Cloud Provider domains.

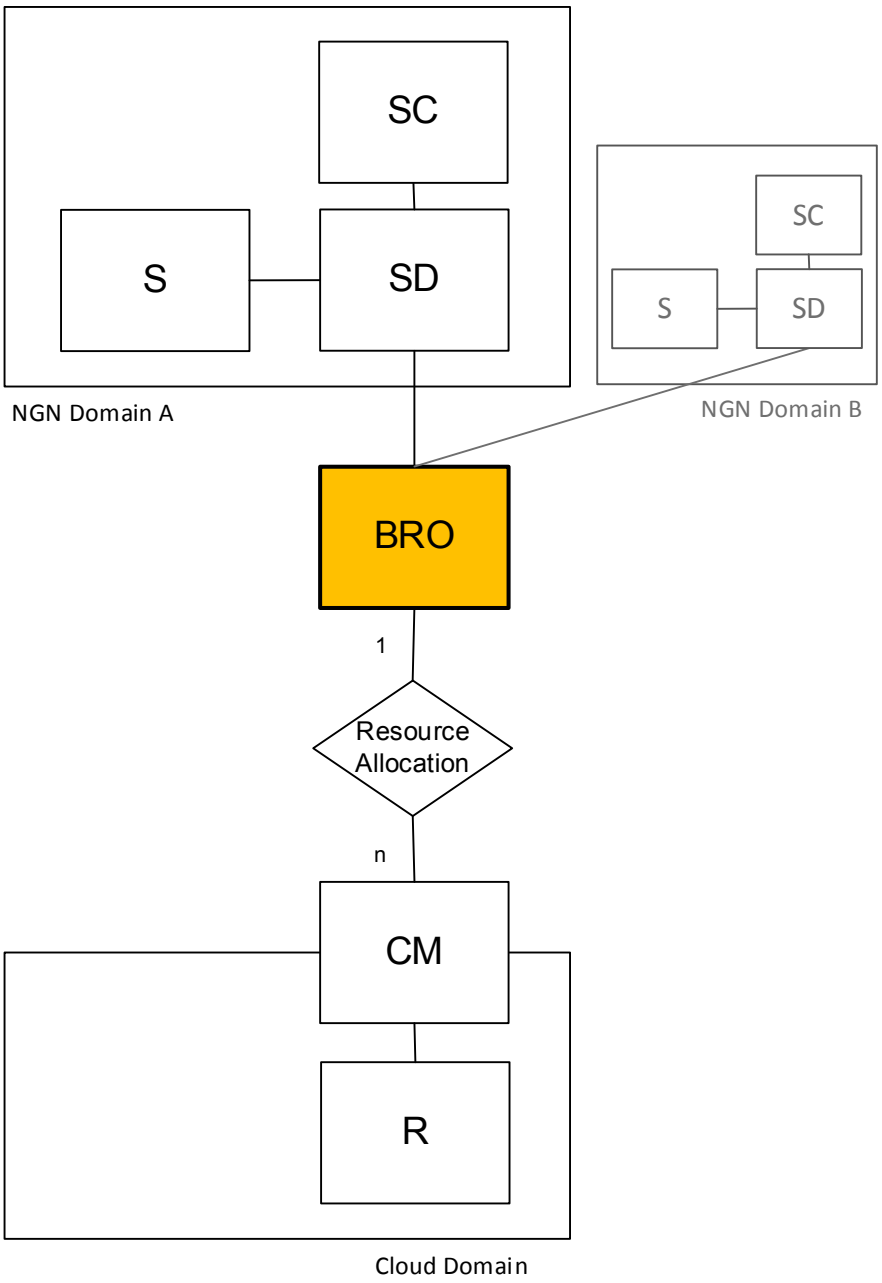

Figure 46: Autonomous Deployment Scenario

Chapter 5

NGN Management Frameworks

5.1 Introduction

For appropriately modeling the required resource and service related information and resource and service related management processes, as depicted in Figure 47, the viewpoint of an NGN service provider is being taken.

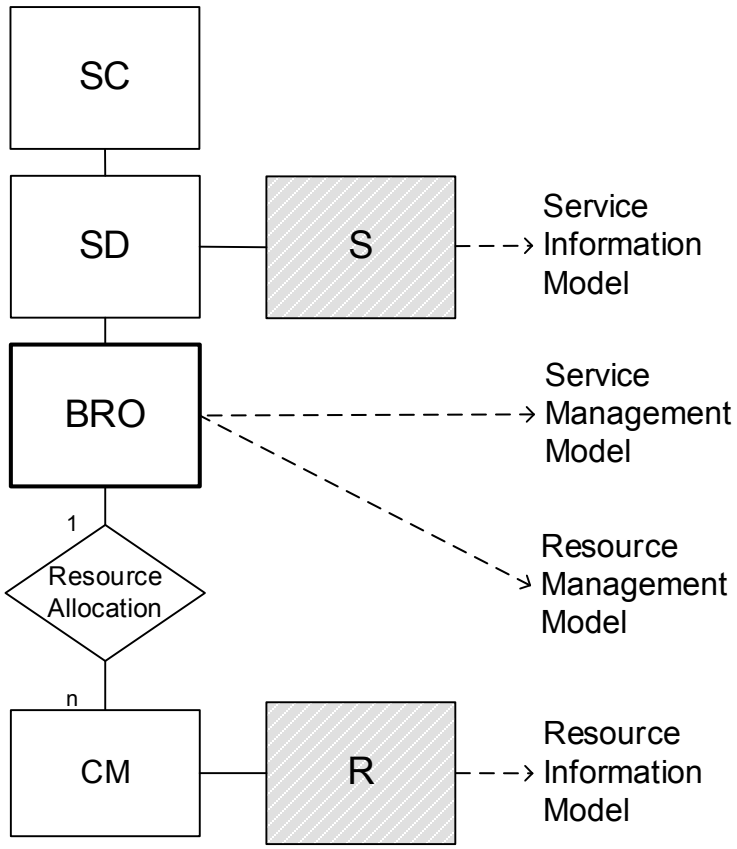

Figure 47: Resource and Service Management processes and Information Models

As explained in section 2.4, comprehensive frameworks for modeling information and processes for the management of entire telecommunication enterprises (including involved resources and services) already exist. Therefore it was avoided to come up with entirely new frameworks for the resource efficient QoS management of NGN services in federated cloud environments, but rather adopt existing frameworks and adapt/complement the latter where necessary. Another benefit of choosing existing frameworks is related to advantages that utilization of existing frameworks provide when integrating described mechanisms into existing, standard-compliant management frameworks. Many inter-management-process interfaces not only have already been specified, but SOA-based solutions TMForum standard-

based like OSS/J [131] or MTOSI [132] are readily available supporting apposite integration into SOA-based management environments.

5.2 Baseline Frameworks for Telecommunication Resource and Service Management

Focusing on the perspective of NGN service providing enterprises for designing a framework for "Resource Efficient Quality of Service Management of NGN Service in Federated Cloud Environments" it is investigated how existing telecommunication enterprise architecture frameworks can be utilized, as the focused Cloud Brokering mechanisms need to find their place in these very frameworks for being integrated as future component of NGN service providing environments.

> From Wikipedia: *"An enterprise architecture framework (EA framework) defines how to organize the structure and views and objects associated with an enterprise architecture. An architecture framework serves as guiding principles to establish a common practice for creating, interpreting, analyzing and using architecture descriptions within a particular domain and/or layers. The framework structures the practitioner's way of thinking in the specific area with supporting maps, matrices and models*." [133]

There are several enterprise architecture frameworks, with different scopes and views on an enterprise already available. Further focusing on the telecommunication service providing enterprises, the TMForum frameworx, as introduced in section 2.4.1, aligned with The Open Group Architecture Framework (TOGAF) [134], standardize business process frameworks (i.e. the TMForum eTOM [47]), information model frameworks (i.e the TMForum SID [66]) and applications (i.e. the TMForum TNA [135]), which provide a comprehensive basis for modeling the information and processes involved in the scope of this work, i.e. resource provisioning and QoS management for NGN services.

5.3 Framework for Resource and Service Information Model

The SID, already from a high level perspective, shown in Figure 48, models the relationship between resources and services. The SID differentiates between logical or physical resources. In the case of this work, cloud resources are logical resource required by cloud-based NGN services. Also the differentiation between customer- and resource-facing services is considered useful for modeling the Cloud Brokering system's information, since customer-facing descriptions of services are less of importance to this work as the particular resource-facing descriptions of a service. With benefits brought about by virtualization techniques and evermore through cloud computing IaaS options, there is no need to describe physical resources and their interdependencies with the services they host anymore. In the TMForum terminology logical resources that implement resource facing services could be cloud instances hosing application servers, databases, control platforms, etc.

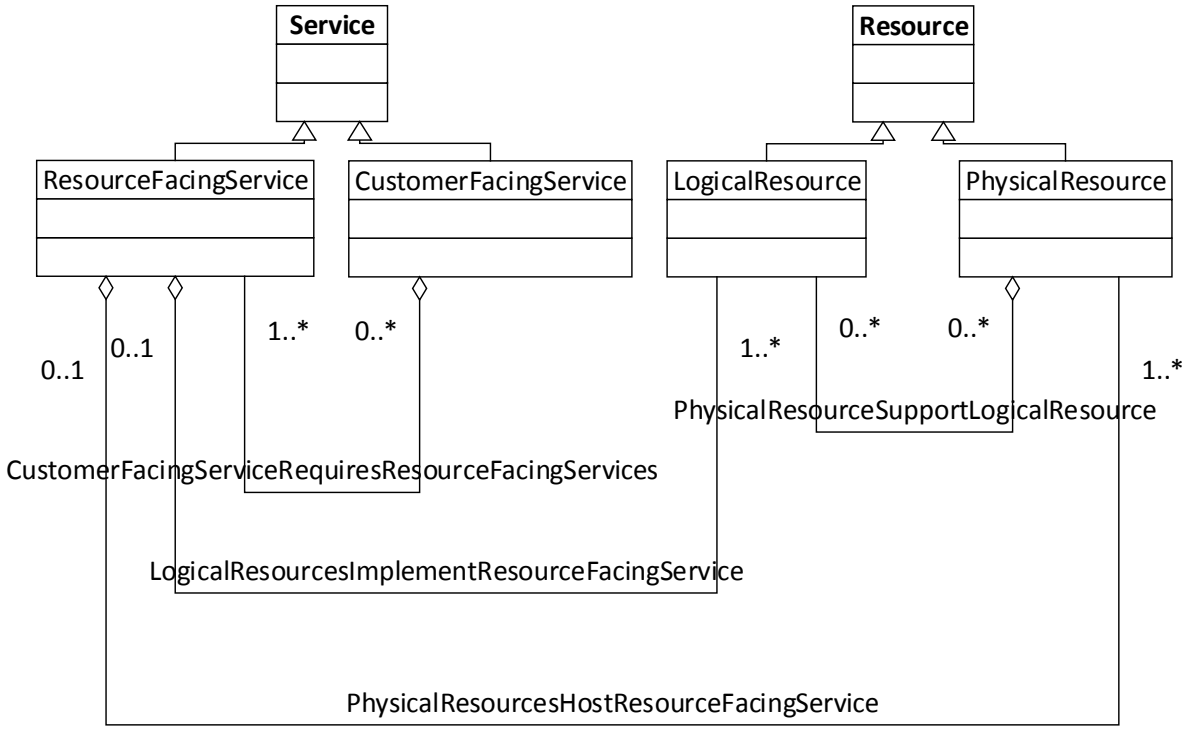

Figure 48: SID high level Resource and Service Model [103]

In Figure 48 the SID for a telecommunication service is depicted. Of core importance for modeling the required service-specific information, the following relevant information is identified 1) Service Level Specification (it needs to be known how QoS is determined/measured for a particular service), 2) Service Capacity Demand (as the capacities that are demanded by a particular service need to be known), 3) Service Definition (as the components that need to be provisioned for activating a particular service need to be known) and 4) Service Access Point (as this parameter dynamically changes, as soon as services are deployed on different cloud infrastructures).

Figure 49: SID Service Information Model, areas of concern highlighted

The SID [66] for resources within a telecommunication service providing enterprise is depicted Figure 50: SID Resource Information Model, areas of concern highlighted. Of core importance for modeling the required resource-specific information are 1) Logical Resource Specification (as this being the specification of a particular cloud resource instance), 2) Resource Capacity (as this should provide information about the service-specific capacity of a particular resource), 3) Performance Specification (as this information serves for judging whether a particular resource provides sufficient performance for a given service) and 4) Resource Usage Specification (as this information will be needed for analysing costs).

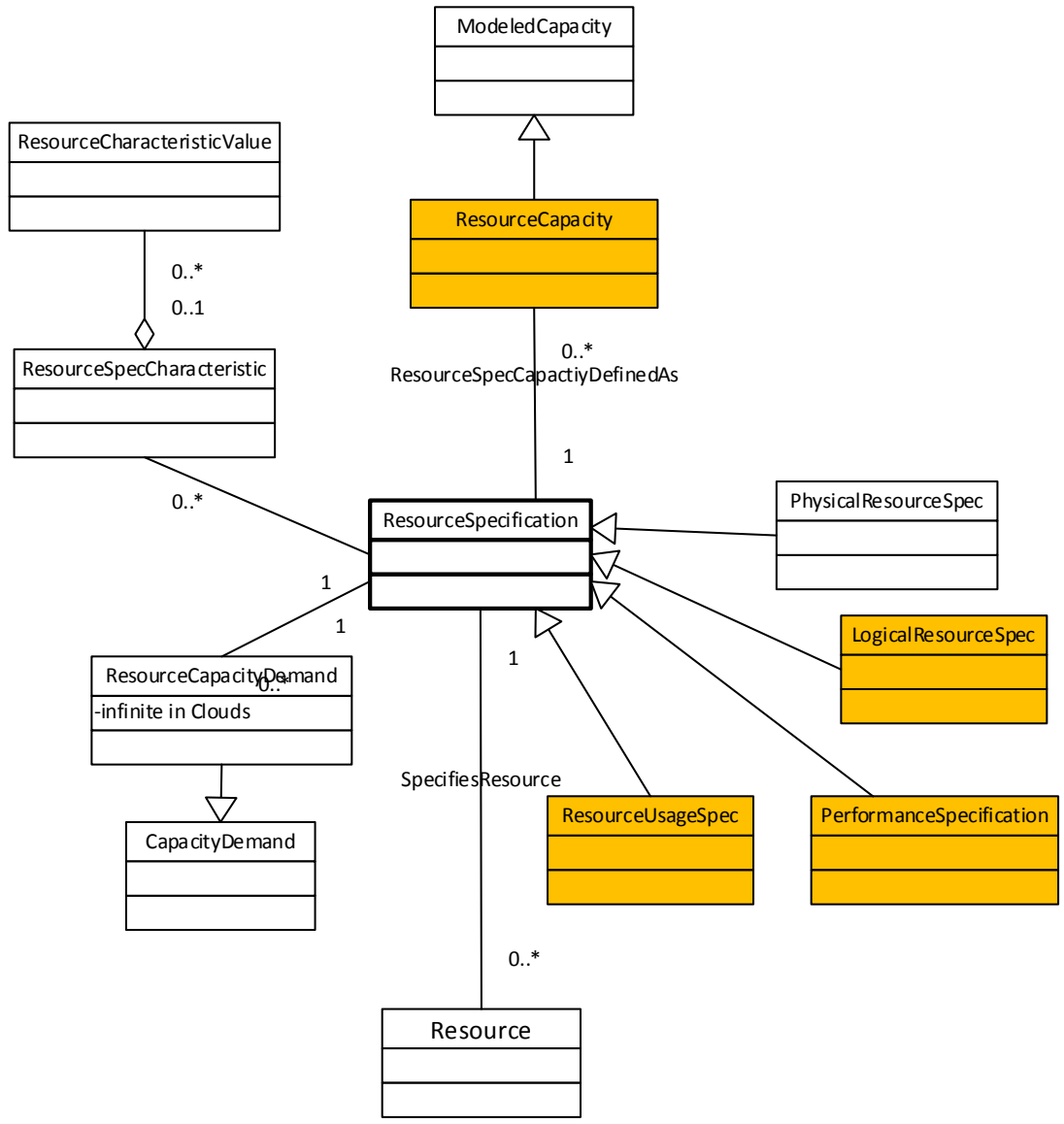

Figure 50: SID Resource Information Model, areas of concern highlighted

As summarized in Figure 51, relevant for this work are not only the specific service specifications on service levels (and how they are measured), the demanded capacity and information on interdependencies required for orchestration and provisioning. Also the exclusively resource related information such as capacity provided by each resource instance (later cloud instance), specification of logical resources (later cloud resources), its performance specification, and usage specification are not sufficient for modeling the information required in this work.

Of particular importance for this work is also the information where concrete relations between services, their QoS dependencies on resource capacities, their dynamically changing end-points (in SID, the ServiceAccessPoints) are described.

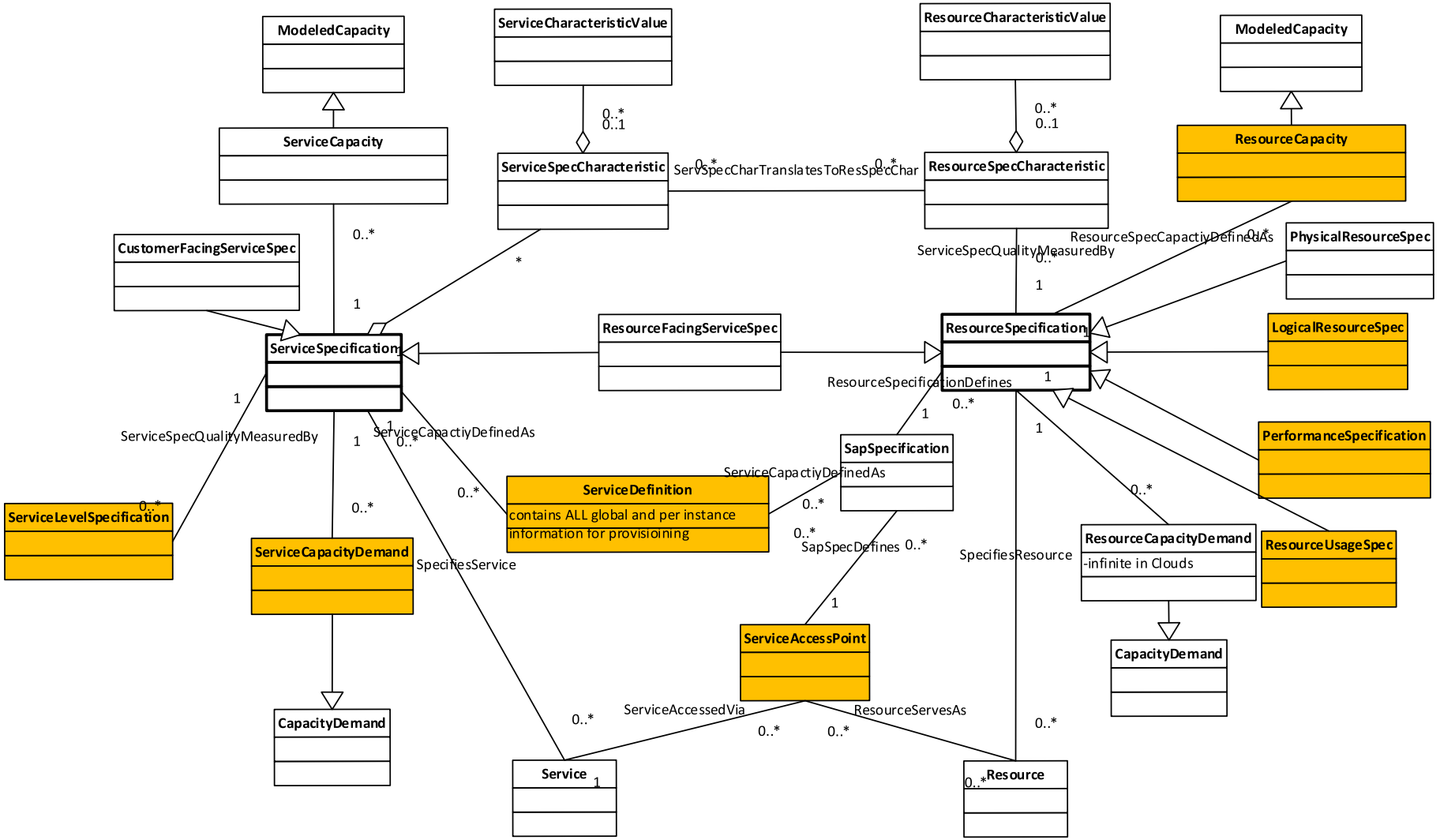

Figure 51: Relevant SID entities for Telecommunication Services and Resources

5.4 Framework for Resource and Service Management Processes

The eTOM framework's resource and service management processes for the telecommunication operations domain provide a blueprint for modeling related processes regarding resource provisioning and QoS management. As shown in Figure 52, this work investigates the interdependency between resource provisioning (resource allocation) mechanisms and service quality management mechanisms. In the context of this work, service quality management processes are only investigated in their relationship with insufficiently allocated resources, or cloud infrastructure related shortcomings.

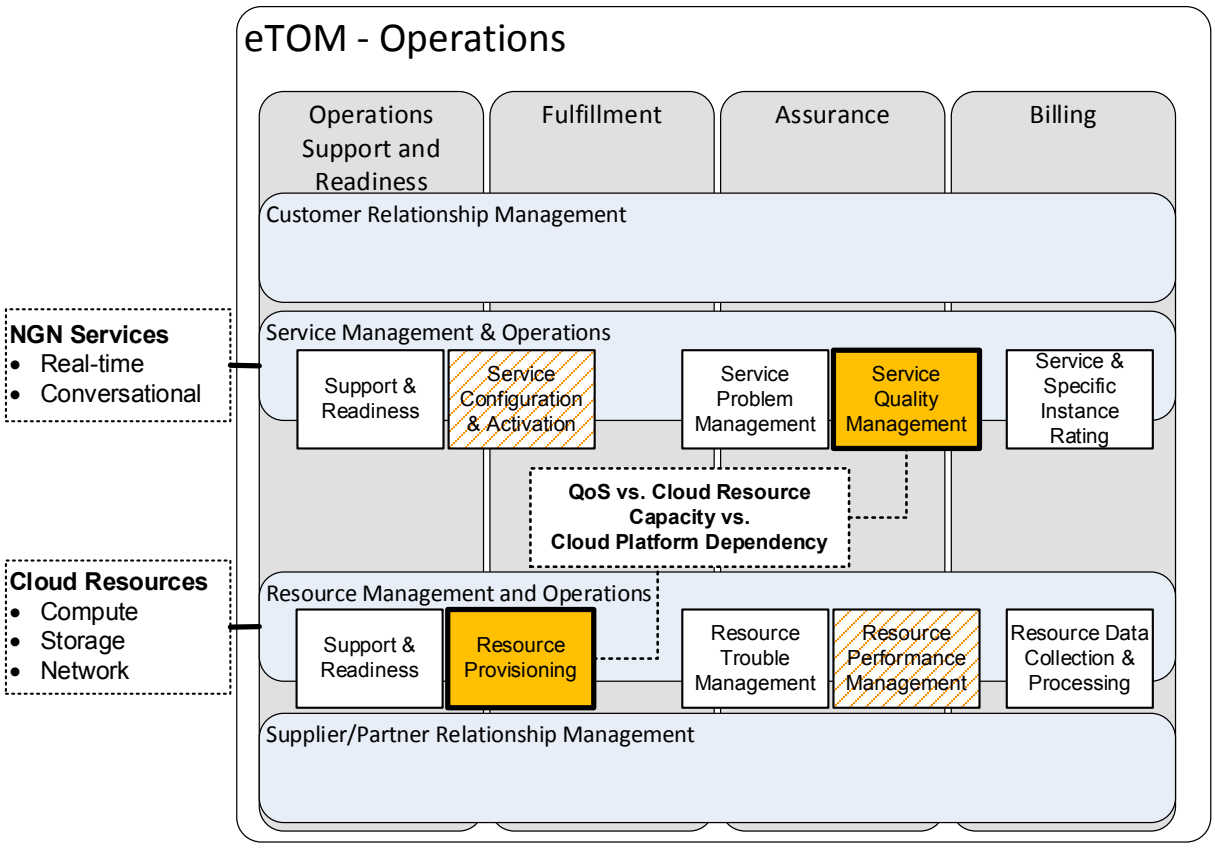

Figure 52: eTOM, Resource Management relations to Service Quality Management

Figure 52 shows the eTOM where the 4 different horizontal layers (customer relationship management, service management and operations, resource management and operations, supplier/partner relationship management) of a telecommunication service provider enterprise, and the 4 vertical operations (operation support & readiness, fulfillment, assurance and billing) are shown. At the core of this thesis are the dependencies between (cloud-) resource provisioning, resource performance management on *Service Quality Management*.

5.4.1 Resource Management Processes

The resource management processes defined by eTOM are depicted in Figure 53. For the context of this work, management processes related to the *provisioning of resources* (cloud resources in the context of this work) are relevant. As highlighted in Figure 53, the following eTOM processes are considered relevant for this work:

- Support and Readiness Processes:

  1) Enablement of Resource Provisioning

     The Cloud Broker needs to be enabled for provisioning Cloud resources of multiple-Clouds infrastructures.

  2) Enablement of Data Collection and Processing

     The Cloud Broker needs to be able to collect and process resource related data, from multiple Cloud infrastructures.

  3) Management of Resource Inventory

     The Cloud Broker needs to maintain an inventory of cloud resources, including the cloud infrastructure offering the latter. For each new cloud infrastructure joining the federation of brokered cloud resources, the Cloud Broker needs to update the inventory.

- Resource Provisioning Processes

  1) Allocation and Delivery of Resources

     The allocation of Cloud resources is the CORE TASK of the Cloud Broker. For doing so, Cloud infrastructures need to be selected from the pool of available Cloud infrastructures as well as Cloud instances / resources offered by them.

  2) Configuration and Activation of Resources

     After Cloud resources have been allocated, the Cloud Broker needs to configure and activate the latter.

  3) Collect, Update & Report Resource Configuration Data

     The Cloud Broker has to maintain, update and report resource configuration data, not only for being constantly aware of currently allocated and activated Cloud resources, but also for billing mechanisms, which need to be aware of the allocation of Cloud resources at each given point in time.

- Resource Performance Management

4) Monitor Resource Performance

The Cloud Broker needs to be aware of the performance of each allocated Cloud resource, as performance degradations impact the delivered QoS.

5) Analyze Resource Performance

In order to plan required resource capacities, the Cloud Broker needs to analyze and maintain data related to the performance / capacity of each Cloud instance in the resource inventory.

6) Control Resource Performance

Whenever the performance of Cloud resources deteriorates or higher performance is required, the Cloud Broker needs to provide mechanisms by which the resource performance can be controlled. While the focus of this thesis is not on fault management processes (e.g. replacing malicious Cloud resources against healthy ones), the main resource performance controlling mechanism of the Cloud Broker are 1) the selection of an alternative cloud infrastructure in case of performance degradations and 2) the control of resource capacities, e.g. up-scaling mechanisms.

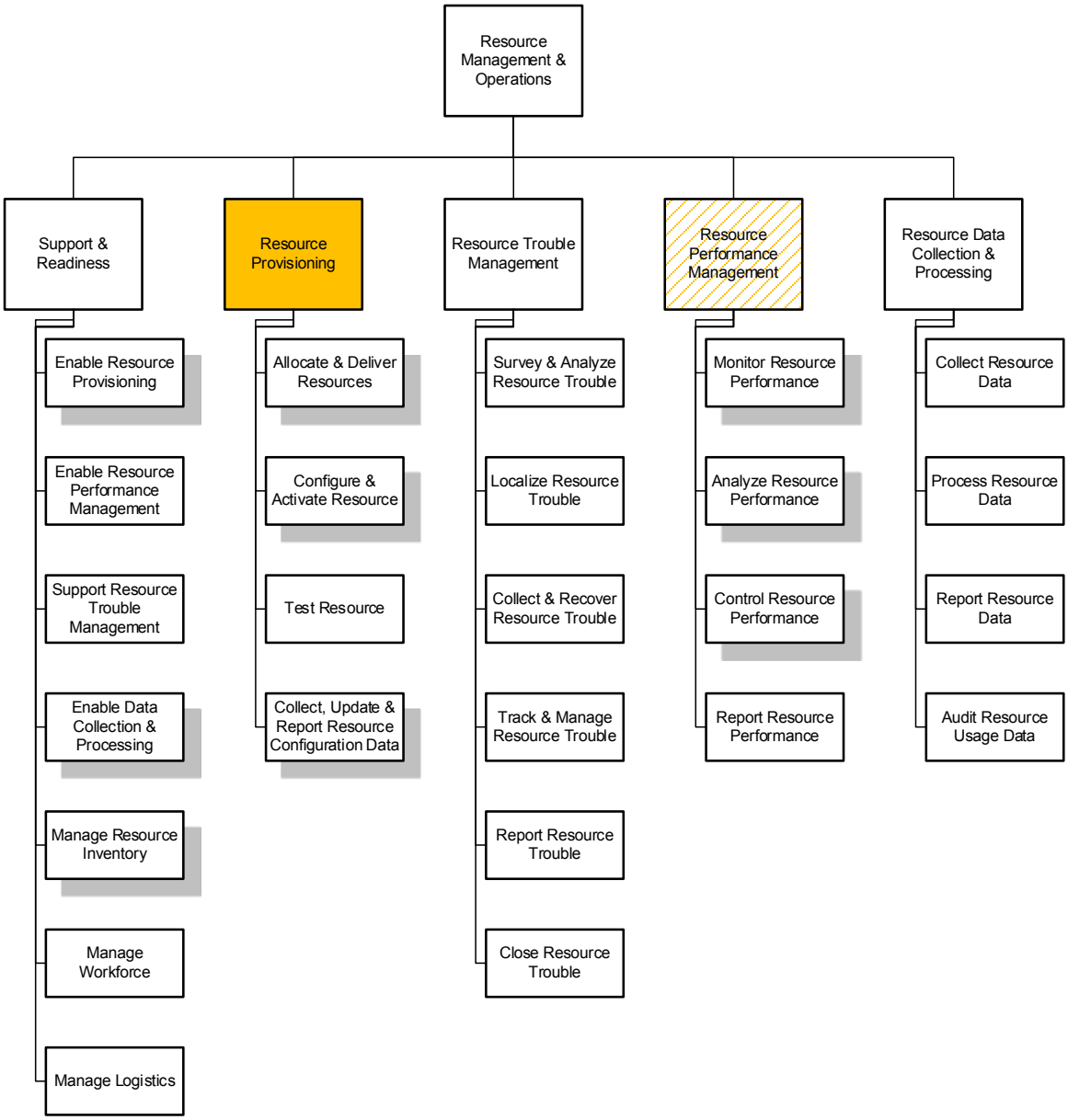

Figure 53: Related eTOM Resource Management and Operations

5.4.2 Service Management Processes

The service management processes defined by eTOM are depicted in Figure 54. For the context of this work, management processes related to the management of service quality (NGN service quality in the context of this work) are relevant. As highlighted in Figure 54, the following eTOM processes are considered relevant for this work:

- Support and Readiness Processes

  1) Enablement of Service Configuration and Activation

     The Cloud Broker needs to be able to configure and activate NGN services. For dynamically being able to deploy NGN services on multiple cloud platforms, the Cloud Broker needs to configure and activate NGN service dynamically.

7) Enablement Service Quality Management

The Cloud Broker needs to be aware of the Service Quality of NGN services and be able to manage the Service Quality / QoS.

- Service Configuration and Activation

1) Allocation of Specific Resource to Services

For each specific NGN service it manages, the Cloud Broker needs to be able to allocate specific and appropriate resources dynamically, for serving the demand of varying NGN service workloads.

2) Implement and Configure Service

The Cloud Broker needs to be able to implement and configure NGN service on federated cloud environments.

3) Activate Service

The Cloud Broker, after having allocated resources for NGN services, implemented and configured the service on federated clouds, needs to activate the service at the level of the NGN service control and service delivery platform.

- Service Quality Management

1) Monitor Service Quality

The Cloud Broker needs to be able to monitor the service quality of each service instance deployed on federated cloud environments.

2) Analyze Service Quality

The Cloud Broker needs to be able to analyze the service quality of each service instance deployed on federated cloud environments.

3) Improve Service Quality

The Cloud Broker needs to be able to improve the service quality of NGN services it manages across multiple domains.

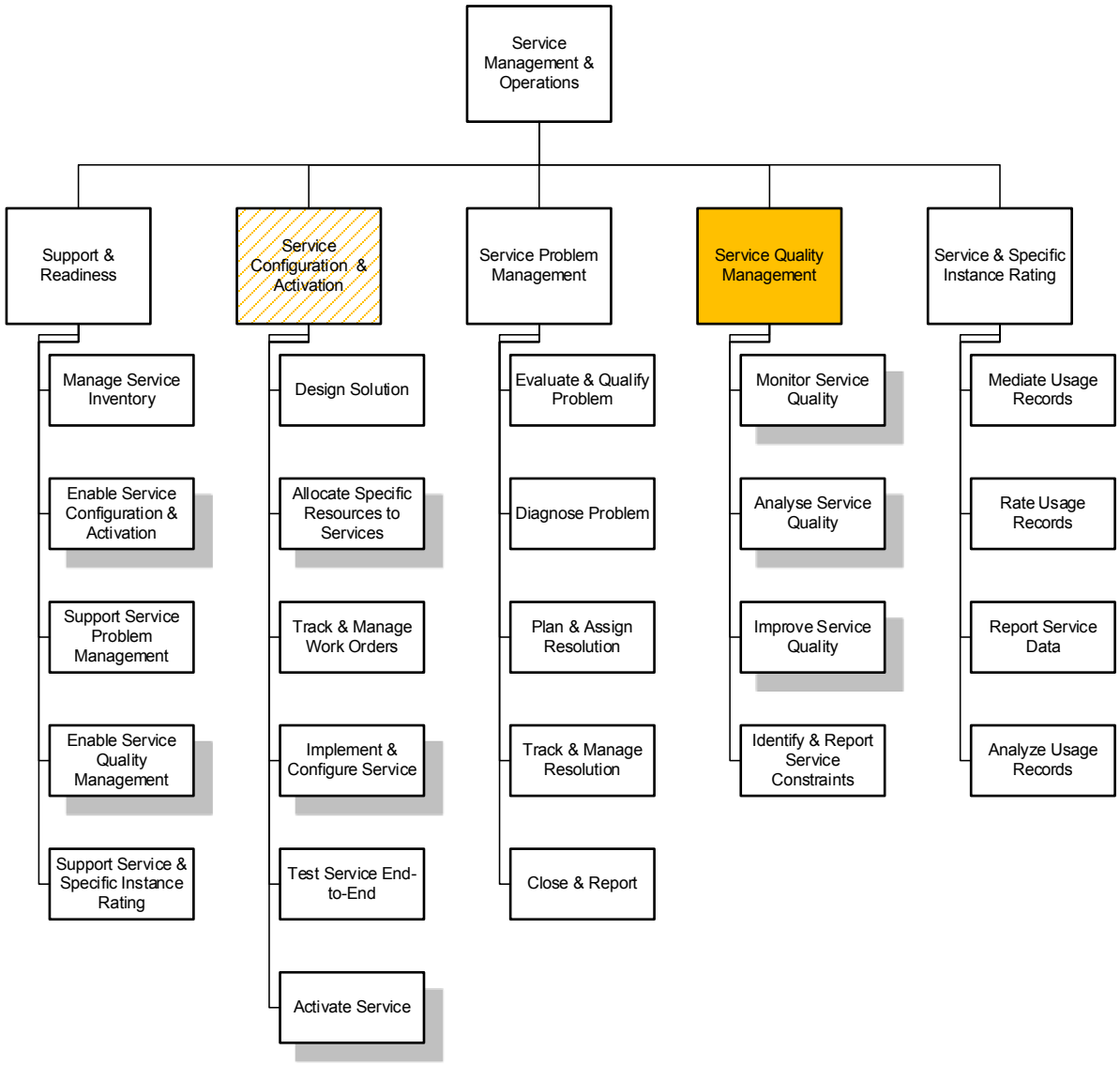

Figure 54: Related eTOM Service Management and Operations Processes [37]

Having identified the relevant Service Quality Management processes, for designing resource efficient QoS management systems for NGN services in federated Cloud environments, the TMForum TAM, as shown in Figure 55, defines the applications required for Service Quality Management. Obviously, before the actual Service Quality Management processes can operate, a Service Quality Model needs to be established. The TMForum TAM [104], defines "*Service Quality Model Establishment applications provide the necessary functionality to establish what will be monitored and how it will be monitored in terms of service quality*"; applications that provide the following functionality: 1) *Definition of the service quality model and its dependencies*, 2) *Establishment of KQIs and SLOs*, 3) *Accepting input from customer contracts or service definitions* and 4) *Establishment of data sources for monitoring of the above*. For the scope of this work, the establishment of NGN service quality models, including all functionalities 1) - 4) are considered relevant, particularly, because the relationship between Cloud resource performance vs. NGN service QoS, as well as Cloud platform performance vs. NGN service QoS, needs to be modeled and monitored.

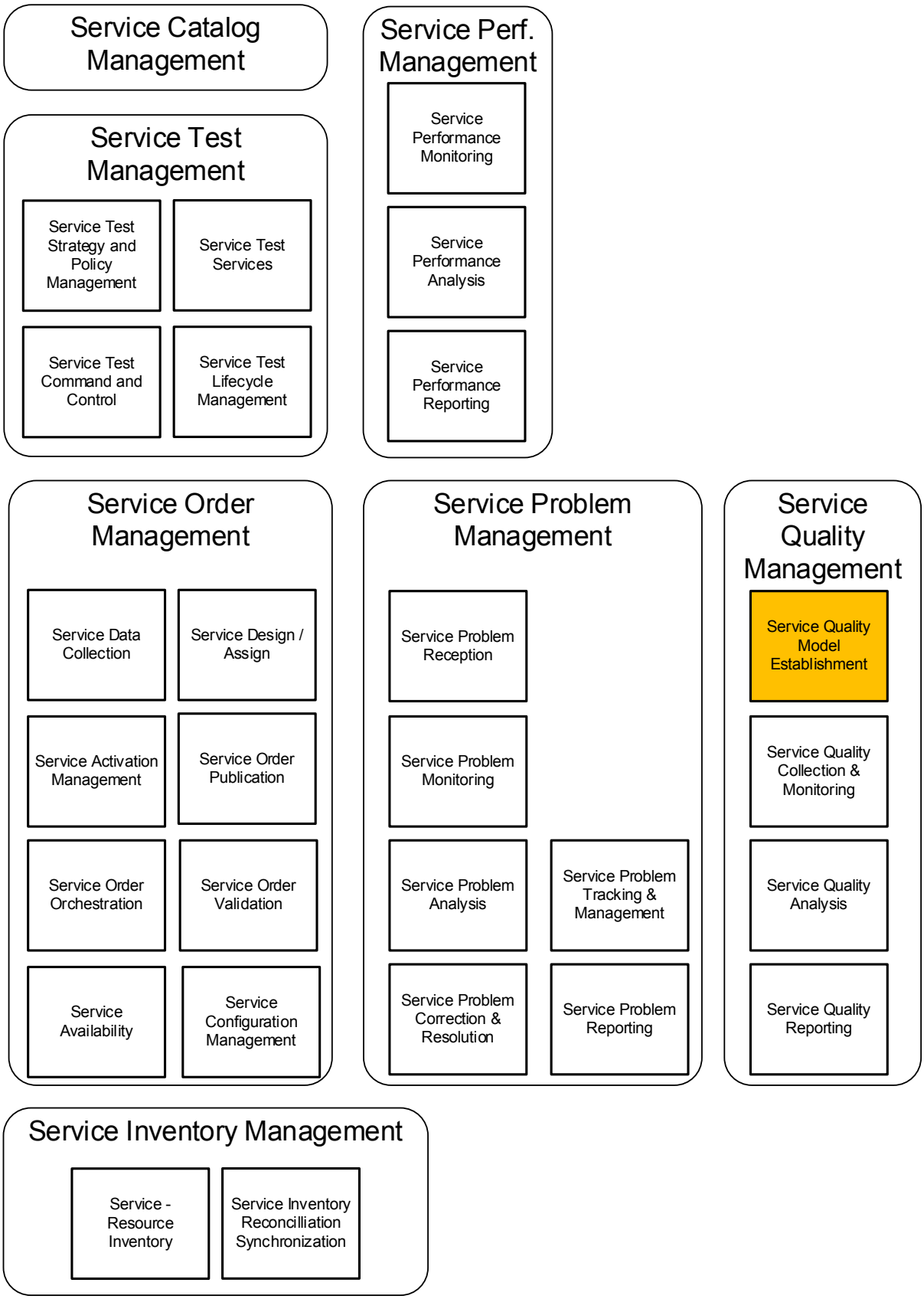

Figure 55: TMForum TAM [104]

5.5 Framework for the Management of the overall Service Lifecycle

The TMForum Service Delivery Framework (SDF), as introduced in section 2.4 provides a framework for the SOA-based management of the telecommunication service lifecycle, in a workflow-oriented fashion. The full SDF service lifecycle is depicted in Figure 56. Based on the full eTOM model, the full service lifecycle workflow consists of the following five phases:

1) *Service Strategy*, dealing with business strategies, product development from a business perspective.
2) *Service Creation*, dealing with the actual service development processes, encompassing required resources, integration into the overall service delivery and management environment.
3) *Service Deployment,* the actual process of integrating the service into the operative environment, configuring related resources and services (i.e. fulfillment), enabling subsequent activation.
4) *Service Execution,* the activation of the service at the level of the NGN service delivery layer / platform and the start of provisioning of the service to users
5) *Service Operations,* the assurance of adequate (at user satisfactory levels) delivery of the service, incl. QoS assurance, fault management, monitoring, etc.

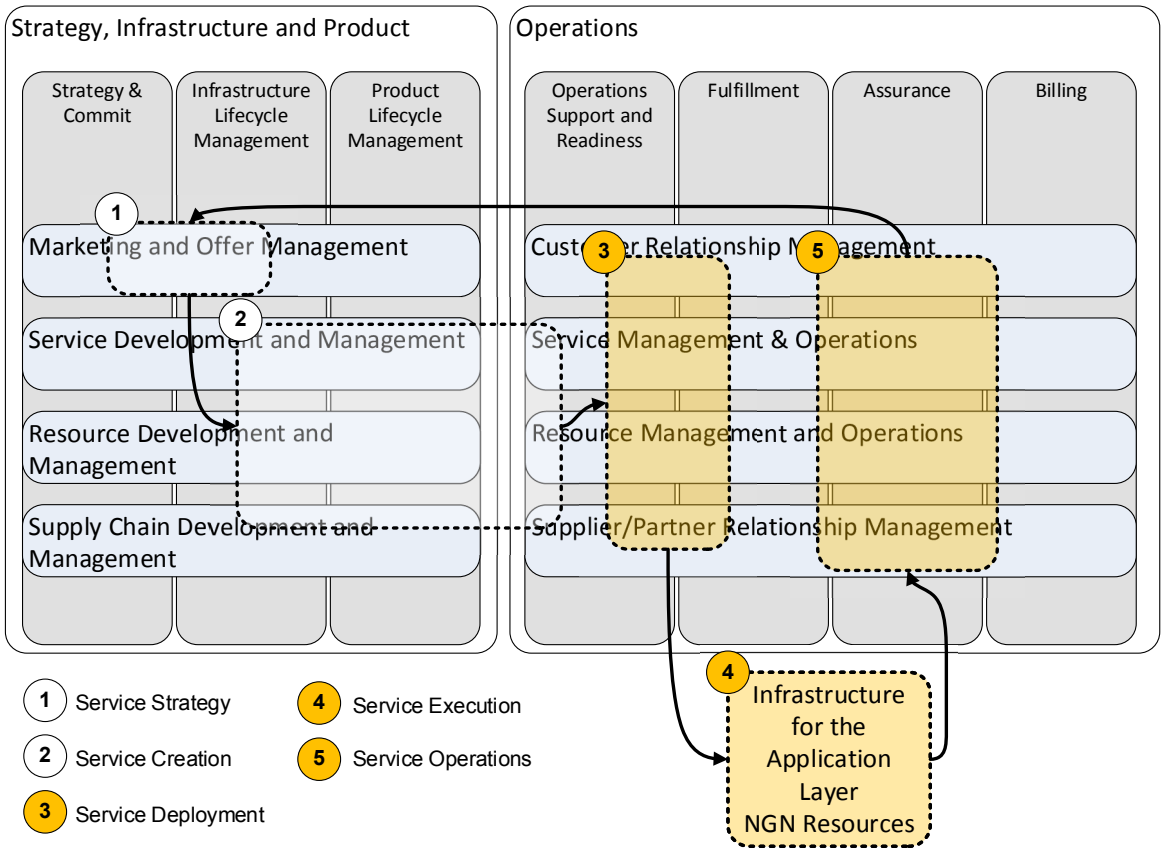

Figure 56: TMForum Service Delivery Framework Lifecycle Operations (based on [136])

Of specific relevance for the scope of this work (not focusing on business, strategic or specific service creation aspects) are Workflow Step 3 “Service Deployment”, i.e. where optimal cloud infrastructures and resources are selected, allocated, provisioned and the NGN service gets deployed (on multiple cloud infrastructures), Workflow Step 4, where orchestrated, SOA-based service provisioning steps, also configuring service dependent components lead to the actual activation of the NGN service and Workflow Step 5, where the health and quality of the NGN service is assured, by means of monitoring and controlling actions. Shown in Figure 57, the main focus is put on automating the following processes with the help of SOA-based service orchestration workflows.

3) Service Deployment:

   - Resource Support and Readiness, enabling Resource Management
   - *Resource Provisioning*; in the context of this work: provisioning of Cloud Resources
   - Service Support and Readiness, enabling Resource Management
   - *Service Configuration*; in the context of this work: NGN service and dependent elements for achieving scalability, control-ability, monitor-ability

4) Service Execution:

   - *Provisioning of NGN Applications*; in the context of this work: NGN Service Delivery Platform element provisioning, and Service Control / IMS provisioning for activating the service
   - Provisioning of NGN Resources; in the context of this work: NGN transport layer elements (not relevant for the scope of this work)

5) Service Operations:

   - *Resource Performance Management*; in the context of this work: monitoring and dynamically scaling cloud resource capacities, thus increasing and de-creasing overall resource performance
   - *Service Quality Management*; in the context of this work: QoS monitoring and execution of actions (such as migration or scaling) for improving QoS

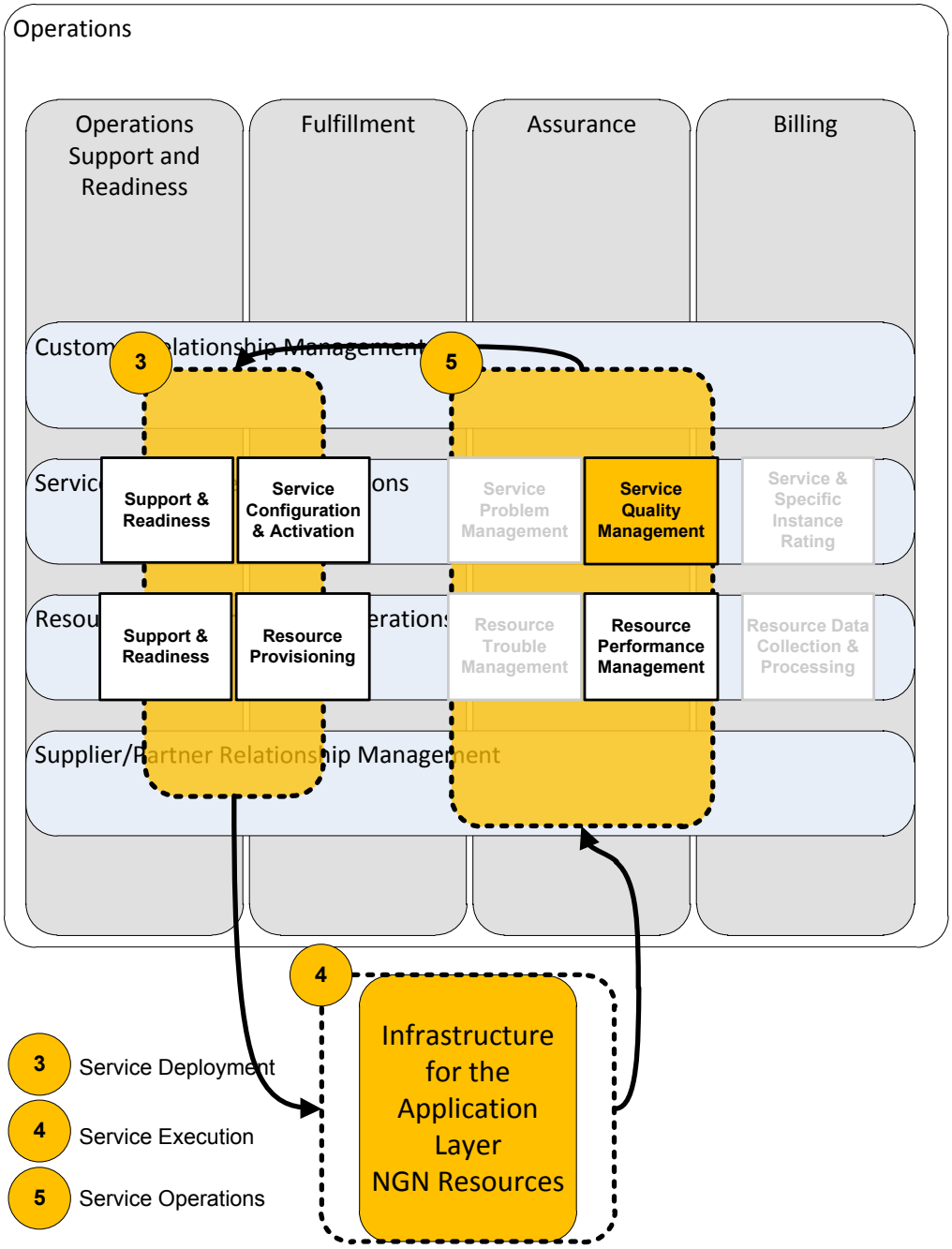

Figure 57: TMForum SDF service deployment, exec. and operations eTOM Processes (bsd. on [136])

Capturing the dynamic aspects of the relevant processes within step 3,4 and 5 (highlighted in *italics* above) the *resource provisioning*, *service provisioning (*subsuming here also the NGN layer provisioning processes), *resource performance management* and *service quality management* workflow is modeled as shown in Figure 58.

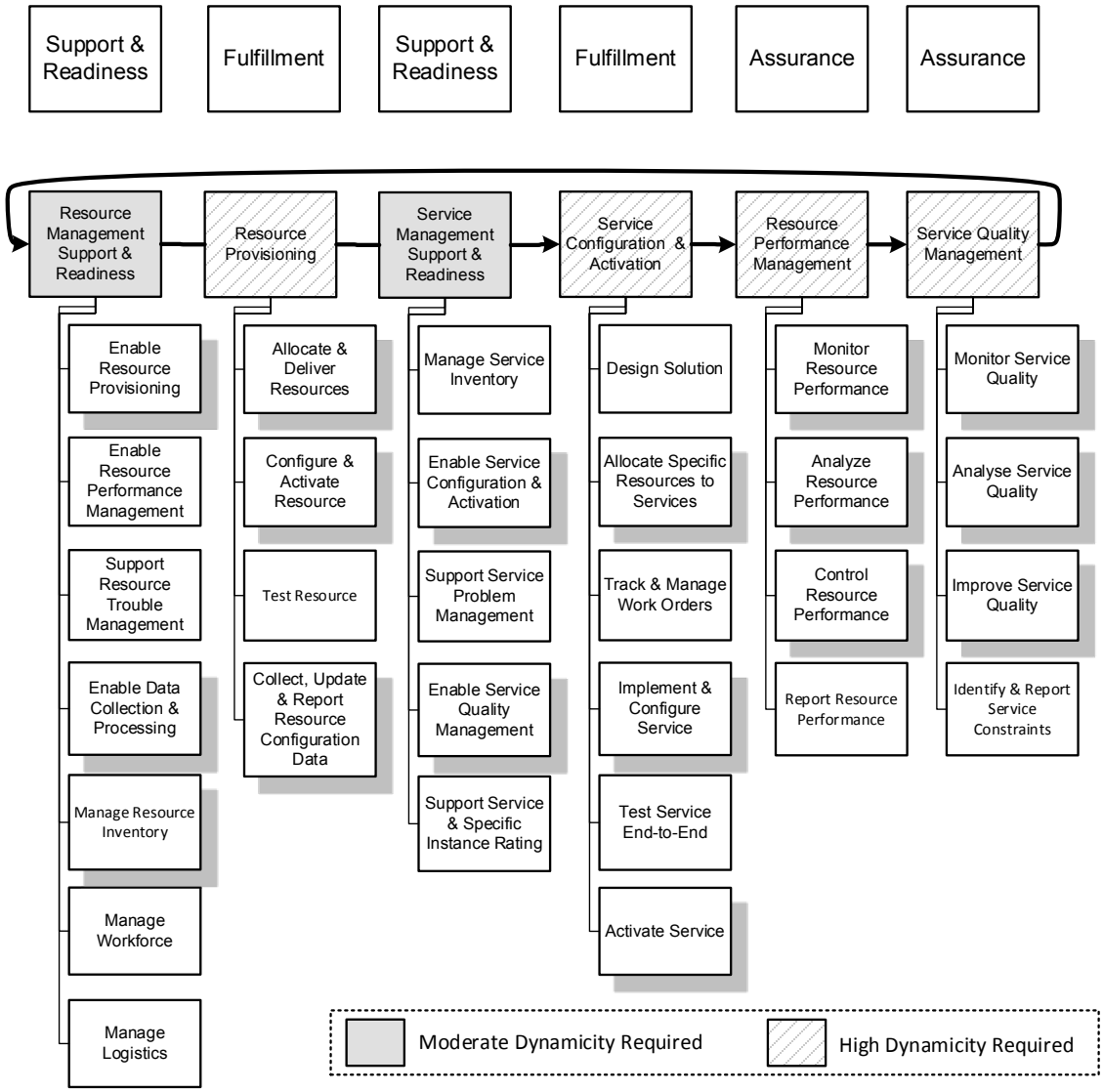

Figure 58: Highly dynamic aspects and moderately dynamic aspects of the Service Management Lifecycle

The highly dynamic aspects of the service management lifecycle, i.e. in this work: the dynamically changing topology of active cloud resources are identified to be dynamic *resource provisioning* mechanisms, *service configuration*, *resource performance management* and *service quality management* processes. Comparably moderate dynamicity is required for processes enabling new services (NGN services in the context of this work) to be managed, as well as new resources (cloud resources in the context of this work) to be introduced to the overall management system.

5.6 Framework for the Autonomic Resource Allocation and QoS Control

For the highly dynamic aspects of the service management lifecycle as described in above section 5.5, an autonomous control theoretic modeling approach is utilized, i.e. the highly dynamic aspects of the eTOM fulfilment and assurance processes are mapped to the Monitor,

Analyze, Plan, and Execute (MAPE) autonomic control loop framework, as introduced in section 2.1.3.

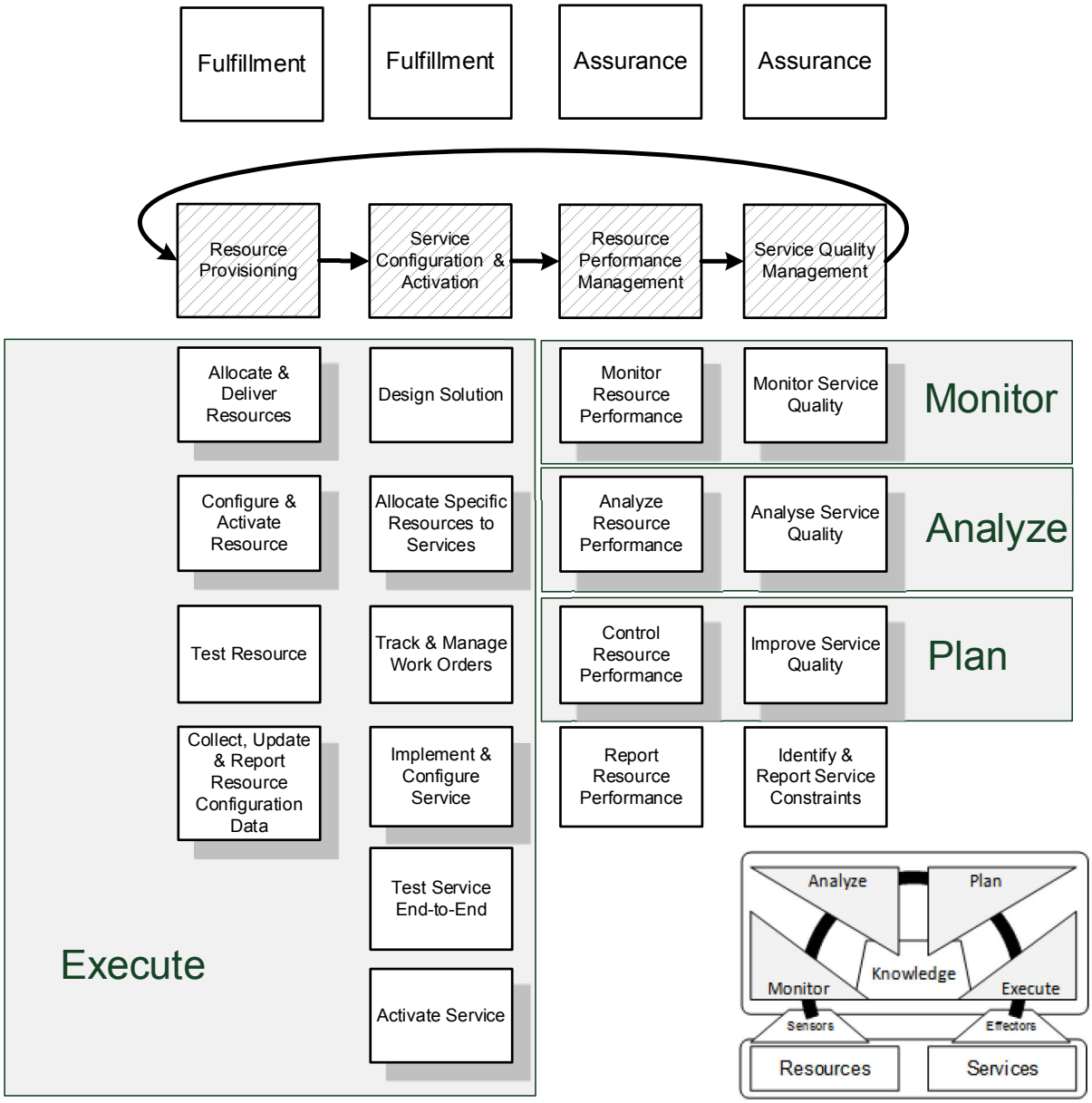

Figure 59: Mapping highly dynamic SDF processes to MAPE Framework

Chapter 6

Framework for QoS-aware Multi-Cloud Brokering for NGN Services

The following sections describe the design and the specification of the QoS-aware Multi-Cloud Brokering Framework for NGN Services (QOSMUC). The first section outlines the major design aspects, where critical criteria that are used as a guideline for designing QOSMUC are contemplated. The second section introduces the actual QOSMUC framework, describing the main architectural design functions and building blocks. In the subsequent and final section of this chapter the core mechanisms of QOSMUC are described in detail.

Based on the TMForum eTOM framework introduced in Chapter 5 (Section 5.4, Figure 54), the following eTOM operation domains are identified to be of core relevance for the *resource management processes of the QOSMUC framework*, shown in Figure 60:

- Resource Provisioning (as described and detailed in section 5.4.1), for dynamically provisioning cloud resources through virtual resource management mechanisms within the Cloud Infrastructure Domain

- Resource Performance Management (as described and detailed in section 5.4.1), by dynamically monitoring Cloud Resource performance within the Cloud Infrastructure Domain

Based on the TMForum eTOM framework introduced in Chapter 5 (Section 5.4, Figure 54), the following eTOM operation domains are identified to be of core relevance for the *service management processes of the QOSMUC framework, shown in* Figure 60:

- Service Configuration and Activation (as described and detailed in section 5.4.2), by dynamically configuring and activating NGN services within the NGN service environment, by interworking with NGN service lifecycle management as well as NGN service orchestration systems

- Service Quality Management (as described and detailed in section 5.4.2), by dynamically monitoring qualities of NGN services of the NGN service environment

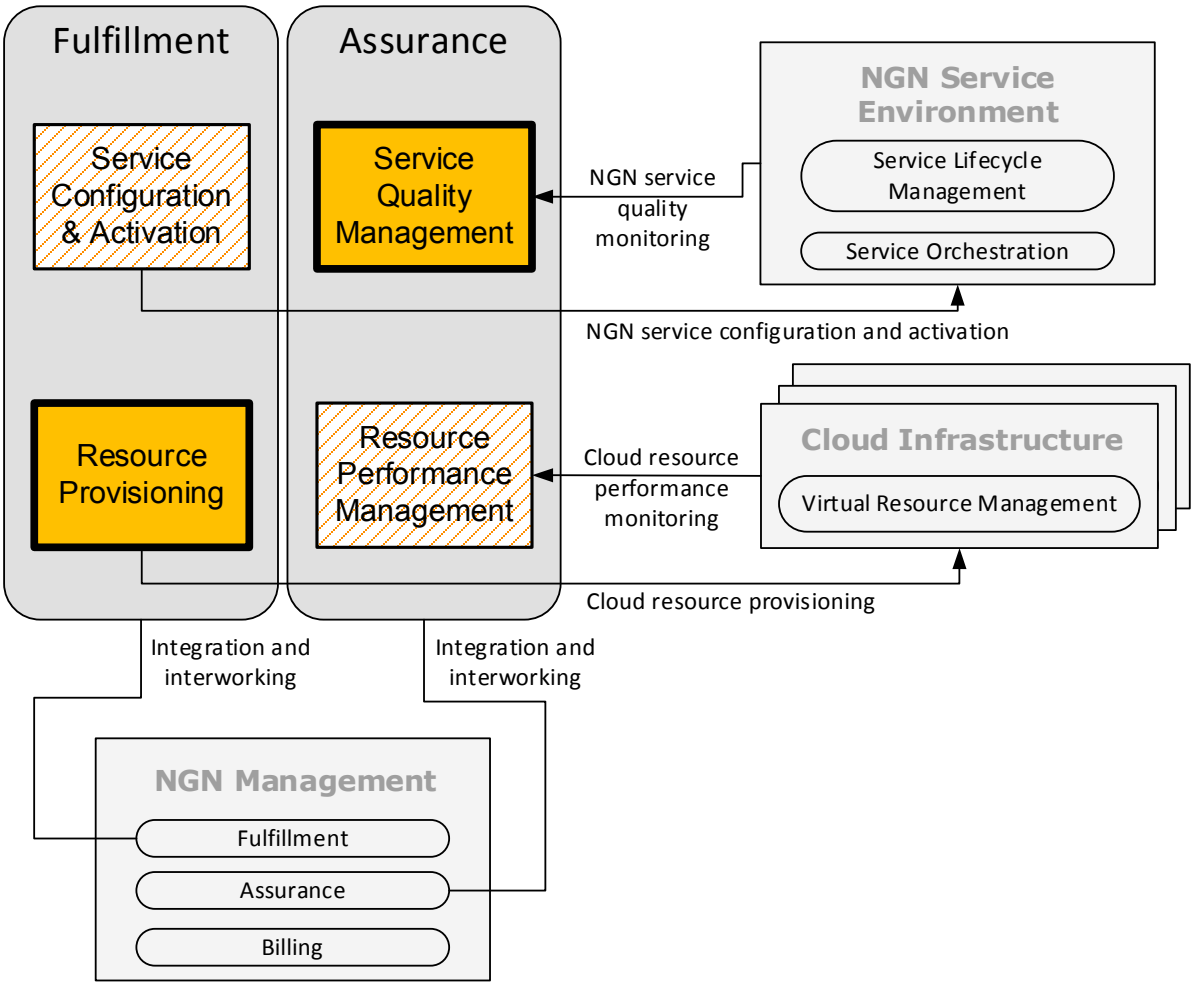

Figure 60: eTOM Operations required and Relationship to Management, Service and Cloud Domain

Furthermore, the establishment of an NGN service-specific Quality Model as described and detailed in section 5.4.2, is of core importance for efficient QoS management of NGN services in federated cloud environments as it enables the service quality management, taking into account cloud resource performance dependencies, as shown in Figure 61.

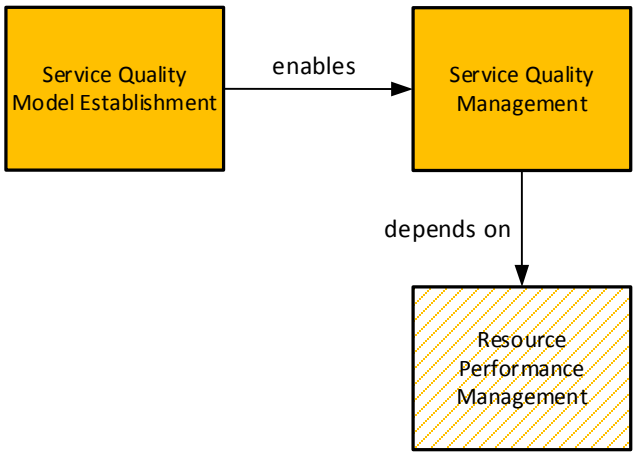

Figure 61: Service Quality Model Establishment enabling NGN Service Quality Management

Finally representing the core aspects of the later designed NGN Cloud Brokering system are capabilities for dynamic cloud infrastructure selection and cloud resource selection and provisioning. These capabilities determine the overall performance of the overall Cloud Brokering system and are evaluated in-depth in Chapter 9. By intelligently and dynamically selecting optimal Cloud Platforms, the Quality of NGN services (hosted on each particular Cloud) can be managed and optimized. By intelligently and dynamically allocating Cloud Resources, first, Cloud resource provisioning mechanisms are being controlled and secondly, the performance of the overall capacity of Cloud Resources can be managed.

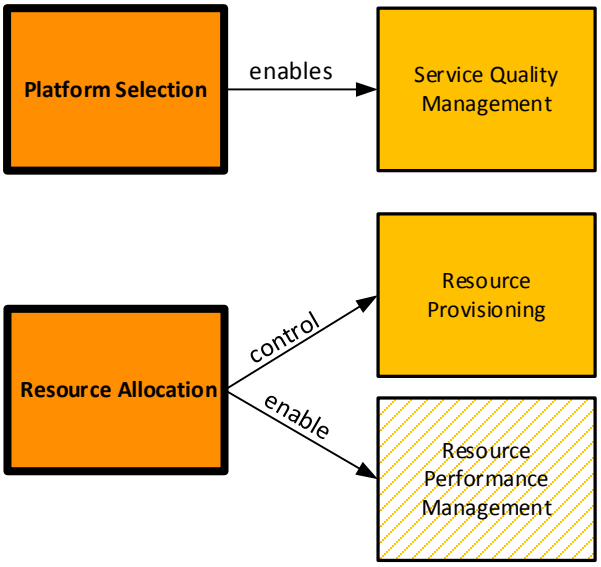

Figure 62 Platform Selection and Resource Allocation relationships to SQM, RP, RPM

The above mentioned processes and applications make up for the core of the operations that are required for resource efficient QoS management of NGN services in federated cloud environments, thus being the core of the QOSMUC framework as shown in Figure 63. Also shown in Figure 63 are the affiliated domains, i.e. the *NGN management domain*, where interdependencies of fulfillment, assurance and billing processes must be adhered, the *NGN service environment domain*, where interdependencies of service lifecycle management operations as well as service orchestration and control processes must be also the *cloud infrastructure domain,* where interdependencies with virtual / cloud resource management processes must be adhered.

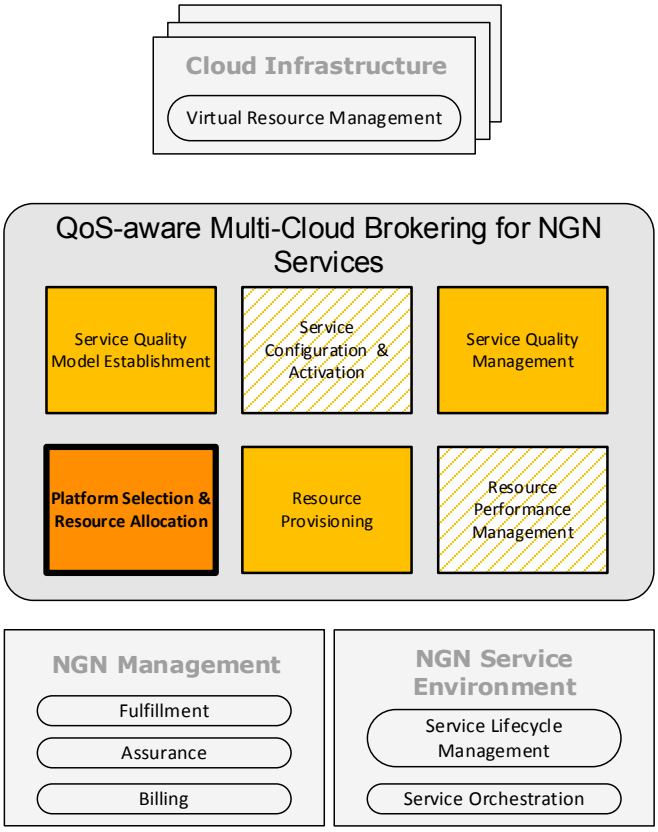

Figure 63: The QOSMUC Framework

As shown in Figure 64 and as introduced in section 5.5 and section 5.6, two interdependent cycles are differentiated, i.e. 1) the overall service management lifecycle and 2) the resource allocation and QoS management control loop. The following sections detail the scope of each cycle / loop in the context of this work.

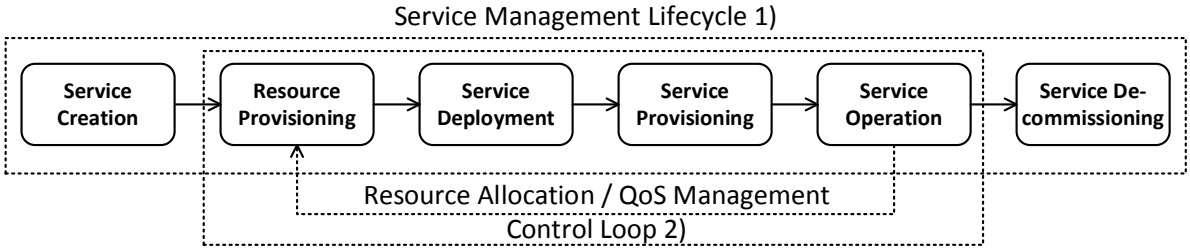

Figure 64: Service Management Lifecycle vs. Resource Allocation and QoS Management Control Loop

1) Service Management Lifecycle

The service management lifecycle starts with a) new services being configured and prepared for activation, policies and user preferences being configured, and b) proceeds with the actual operation phase (i.e. the resource allocation and QoS management control loop defined in 2)), and c) ends with the final decommissioning of services.

2) Resource Allocation and QoS Management Control Loop

The control loop always starts by a) selecting appropriate cloud infrastructure(s), and cloud resources, continues with b) provisioning cloud resources so as to meet the capacity demand of current NGN service workloads, proceeds with c) where NGN services are deployed on the new cloud instances and d) where the service, including dependent service components are provisioned and ends with e) where the new topology is activated and put into operation.

Whereas the QOSMUC framework for the service management lifecycle is described in the following section 6.1, the framework for QOSMUC's resource allocation and QoS management control loop is described in section 6.2.

6.1 QOSMUC Service Management Lifecycle Framework

The service management lifecycle of QOSMUC is depicted in Figure 65. The Service Creation / Preparation processes are mainly those eTOM processes related to the eTOM *resource and service support and readiness* domains, as well as TMForum TAM's *Service Quality Model Establishment.* Service de-commissioning includes processes for releasing services, resources and the accompanied data (i.e. policies, configurations and measurement data).

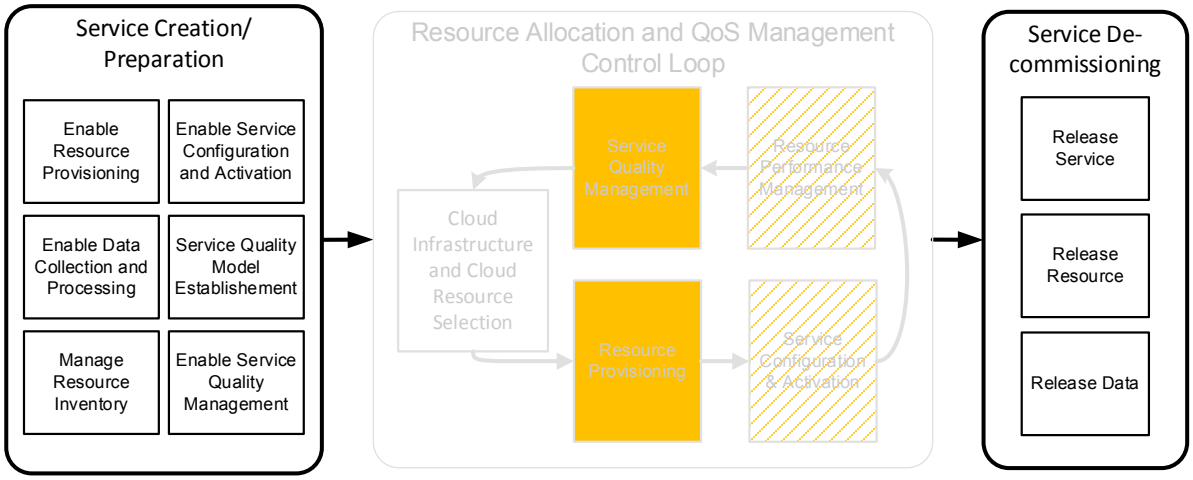

Figure 65: QOSMUC's Service Lifecycle Management Framework

Mechanisms for initial resource allocation and service deployment

The initial resource allocation and service deployment phase of the resource and service management lifecycle involves:

- initial selection of cloud resource providing platform(s)

which is typically only able to take static information into account in the context of the platform selection process.

- initial allocation of cloud resources

which requires the a-priori availability of information about resource capacity requirements of a given service, as opposed to self-learning, platform probing mechanisms, which have not been analyzed in detail in this work.

- initial resource, service and service depending instance provisioning and orchestration

which represents the initial execution of the provisioning workflows generated in the preceding, preparation phase.

It should be noted however, that the basic mechanisms conducted during this initial provisioning and orchestration phase are identical to the mechanisms conducted during each cycle of QOSMUC's resource allocation and quality management control loop, only differing by a limited set of available, dynamic information like QoS related or cost related information.

The successful execution of the initial resource allocation and service deployment phase marks the point where QOSMUC is ready to be subjected to incoming user load. This is where the QOSMUC's resource allocation and quality management control loop is being entered, which is described in the subsequent section 6.2.

Mechanisms for resource and service de-commissioning

During the final phase of QOSMUC's resource and service management lifecycle

- services and service depending instances are being de-provisioned
- resources are being de-provisioned
- polices are being deleted

Whereas the de-provisioning workflow, is modeled as a simple roll-back of QOSMUC's provisioning and orchestration workflow, deletion of the remaining artifacts, like policies and further service related information involves additional provisioning steps.

6.2 QOSMUC Resource Allocation and Quality Management Control Framework

QOSMUC's core resource allocation and quality management control framework is modeled according to relevant TMForum eTOM [47] processes for 1) resource provisioning, 2) service configuration and activation, 3) resource performance management, 4) service quality

management and mapped to the core autonomic computing Monitor, Analyse, Plan, and Execute (MAPE) phases of a closed control loop. This mapping is shown in Figure 65, where on the right the eTOM processes are being shown, whereas on the left the functions for realizing the related eTOM processes mapped to a MAPE loop are shown.

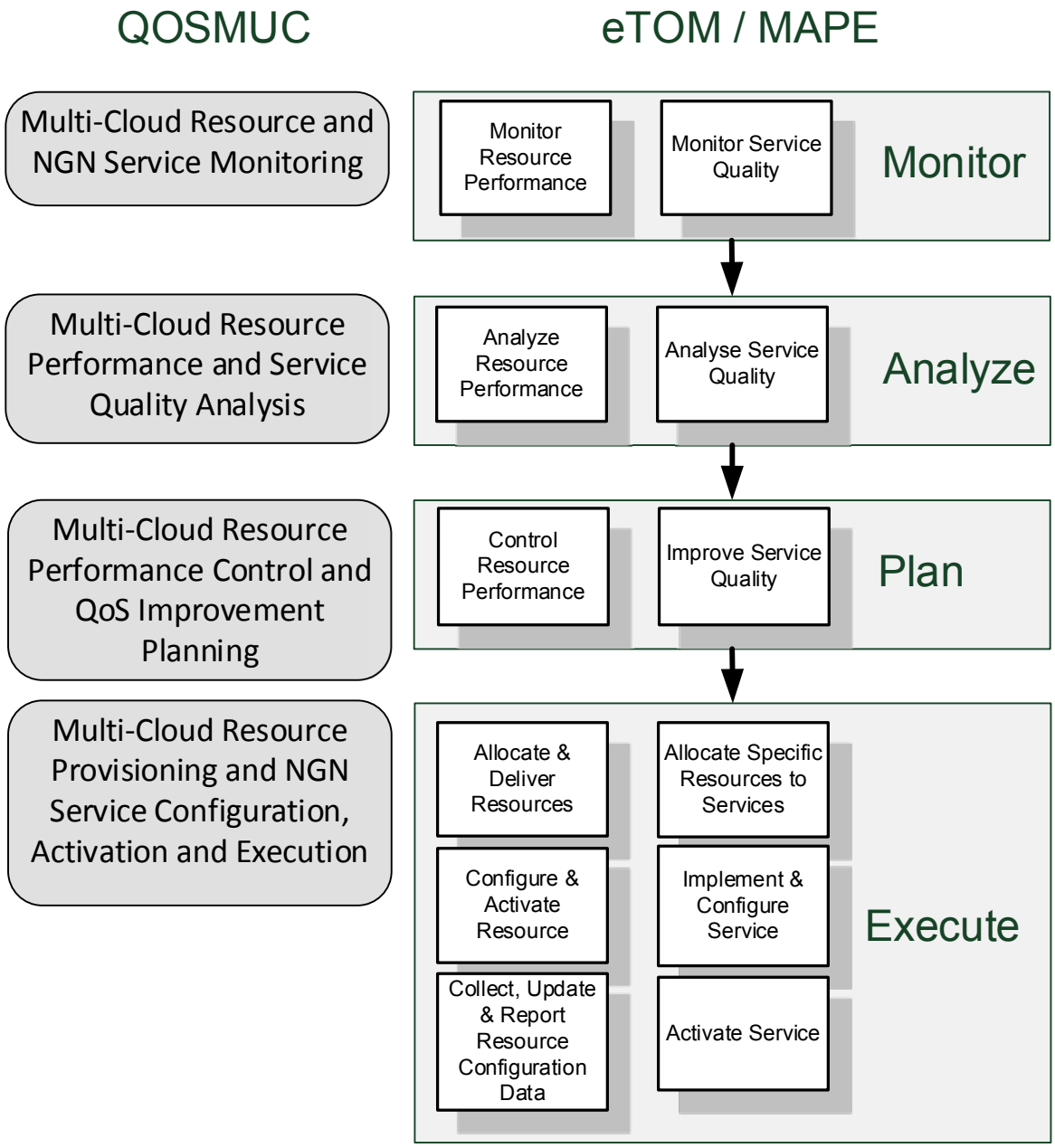

Figure 66: QOSMUC Control Loop Framework

6.3 QOSMUC Information Framework

Based on the TMForum SID Information framework for telecommunication / NGN resources and services, the required information for modeling QOSMUC's resources and services is determined.

QOSMUC Service Information Model

As shown in Figure 67, for modeling QOSMUC's NGN services, the following SID models relevant for QOSMUC's service information model are identified: SID Service Level Specification (QoS requirements, as part of QOSMUC's service QoS model), SID Service Capacity Demand (QOSMUC's Service's Resource Requirements), SID Service Definition (QOSMUC's Service Type and Interdependencies) and SID Service Access Point Specification (QOSMUC's Service Interface).

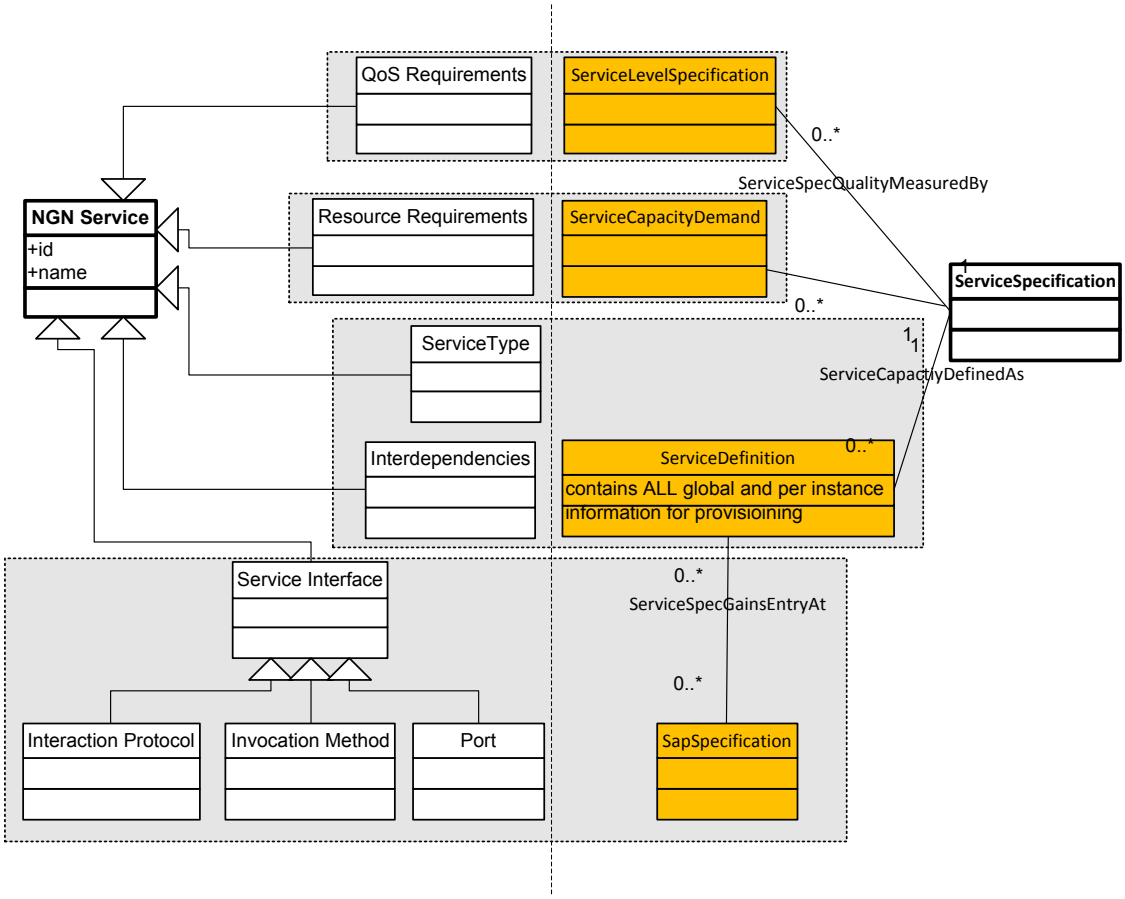

Figure 67: QOSMUC vs. SID Service Information Model

QOSMUC Resource Information Model

As shown in Figure 68, for modeling QOSMUC's NGN resources, the following SID models relevant for QOSMUC's resource information model are identified: SID Resource Capacity (as part of QOSMUC's Cloud Infrastructure Resource Model, i.e. Cloud Instance and Cloud Platform), SID Logical Resource Specification (as part of QOSMUC's Cloud Infrastructure Resource Model, i.e. VM Image), SID Performance Specification (as part of QOSMUC's Cloud Platform Model and Cloud Infrastructure Resource Model).

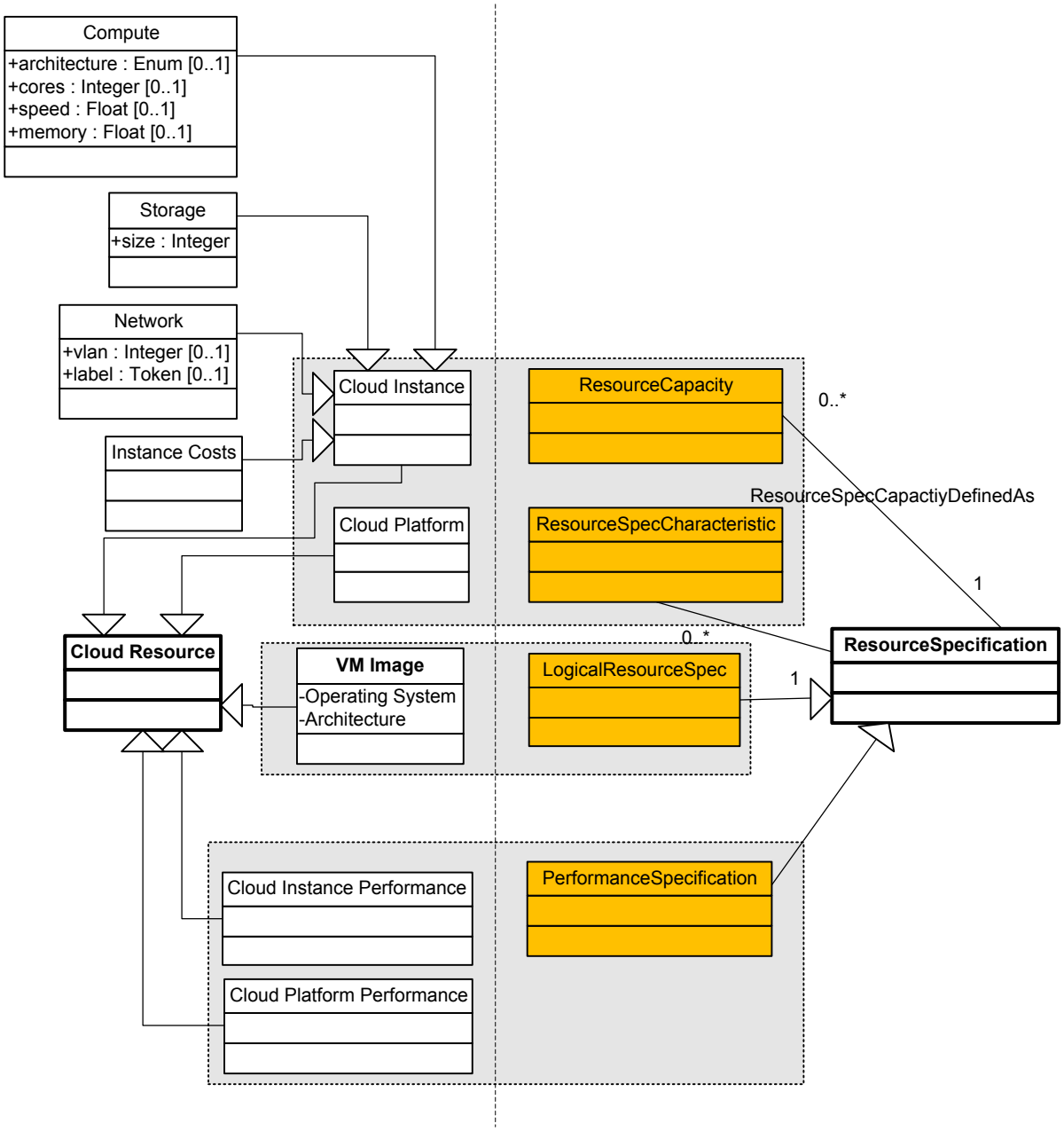

Figure 68: QOSMUC vs. SID Resource Information Model

Chapter 7

Information Models and Methods

For the Cloud Federation Broker to efficiently allocate cloud infrastructure resources across multiple, federated cloud infrastructures, assuring telecommunication service specific QoS requirements, information about the service, it's QoS requirements, it's interdependencies with NGN systems and service components, the network, the cloud infrastructure performance, the current load, as well as information about the available cloud infrastructure resources, including their cost, need to be available in a structured form. Furthermore the Cloud Federation Broker needs to access Cloud Infrastructure Management (CM) interfaces, for provisioning Cloud infrastructure resources, addressing them in an abstracted, well-defined fashion.

The following sections describe the main methods for designing the Cloud Federation Broker, which are 1) telecommunication service description model 2) QoS model 3) service interdependency model 4) resource requirements model 5) cloud infrastructure resource abstraction 6) cloud infrastructure resource provisioning 7) Cloud Resource Description Model 8) Cloud Resource Cost Model.

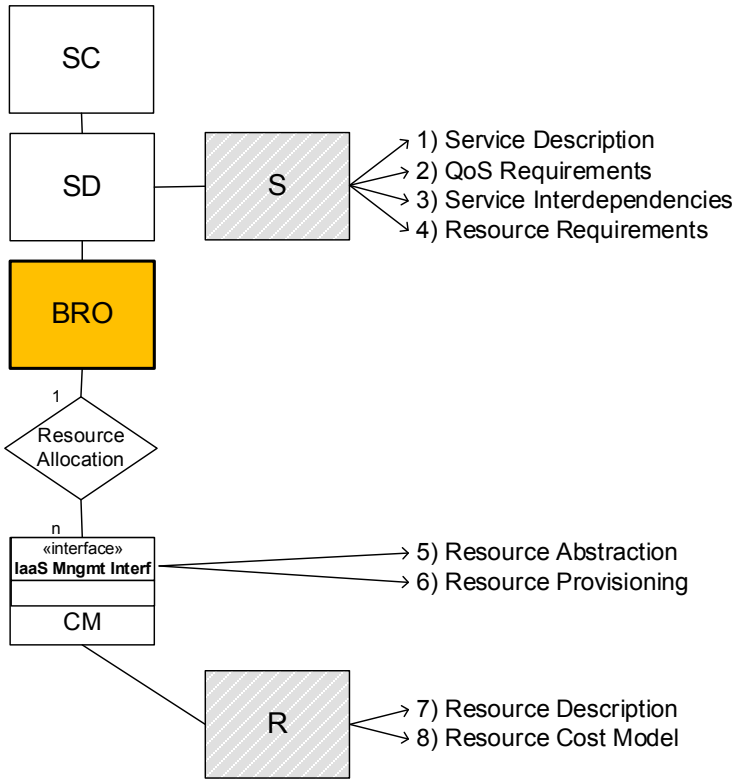

Figure 69: Key Information Models and Methods

7.1 Resource, Service, Infrastructure and Policy Information Models

7.1.1 NGN Service Model

In order for the Cloud Federation Broker to dynamically and efficiently deploy and provision a specific Telecommunication Service on multiple Cloud Infrastructures a set of Service describing information has to be provided. The main information to be provided is shown in the following Figure 70.

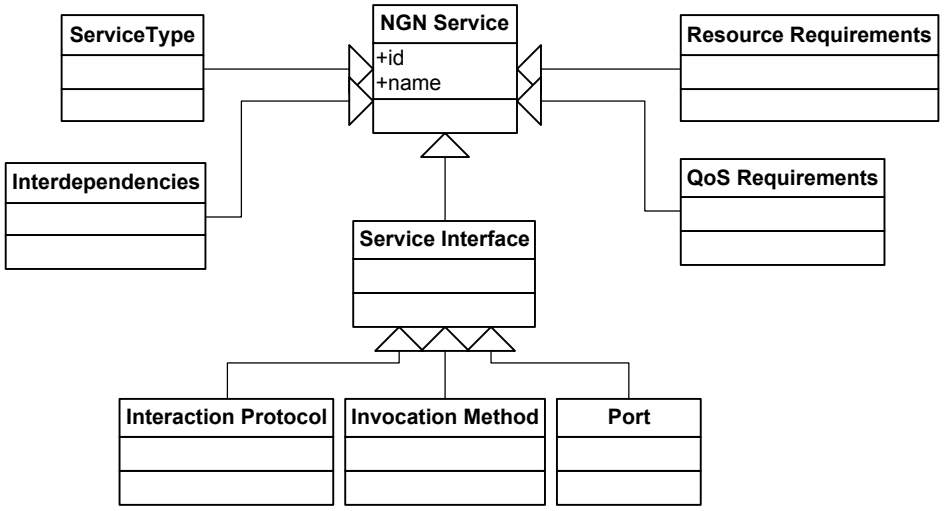

Figure 70: NGN Service Model

7.1.2 NGN Service QoS Model

Each NGN service has its specific QoS requirements for network, cloud infrastructure and application performance. In order for the Cloud Federation Broker to assure the QoS of a specific NGN service, QoS factors that influence signalling, media and data communication quality need to be known.

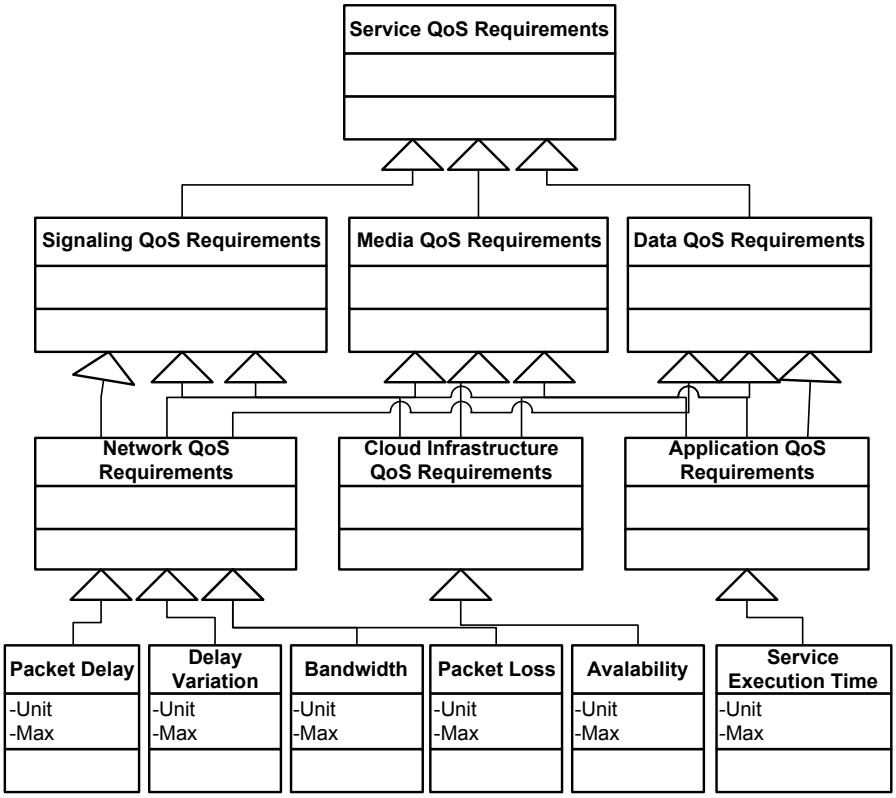

Figure 71: NGN Service QoS Model

7.1.3 NGN Service Interdependency Model

For the Cloud Federation Broker to resolve service interdependencies upon deployment of a specific Telecommunication Service its interdependencies with the NGN service control platform, the NGN OSS and BSS as well as its interdependencies with other service components need to be known.

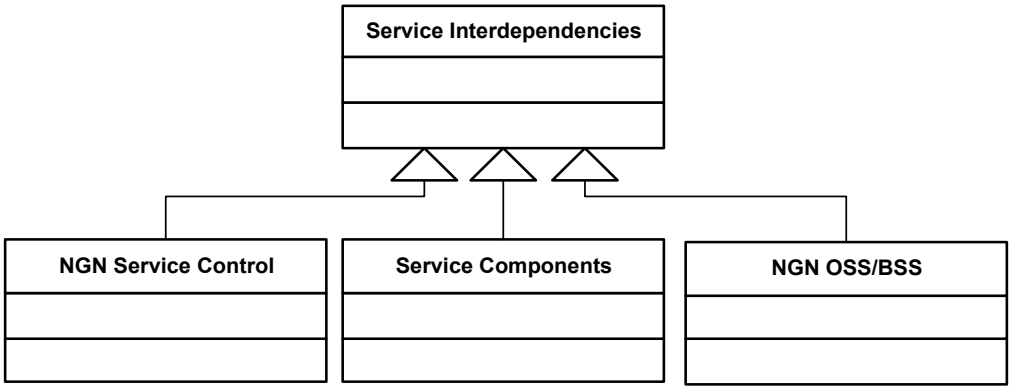

Figure 72: Service Interdependencies Model

7.1.4 NGN Service Resource Requirements Model

Telecommunication Services need to be deployed on specific Cloud Infrastructure Resources, with a specific set of Compute, Storage and Networking capacities. Typically Services are already pre-packaged in the form of VM Images, which can be retrieved from a specific location. Upon deployment, system-internal settings have to be provided for the service to interconnect with affiliated, dependent systems and services. Furthermore, specific Elasticity rules can be issued at deployment time, which specify certain thresholds for triggering resource scaling mechanisms. Finally, external, dependent systems and services that need to be provisioned upon deployment have to be specified.

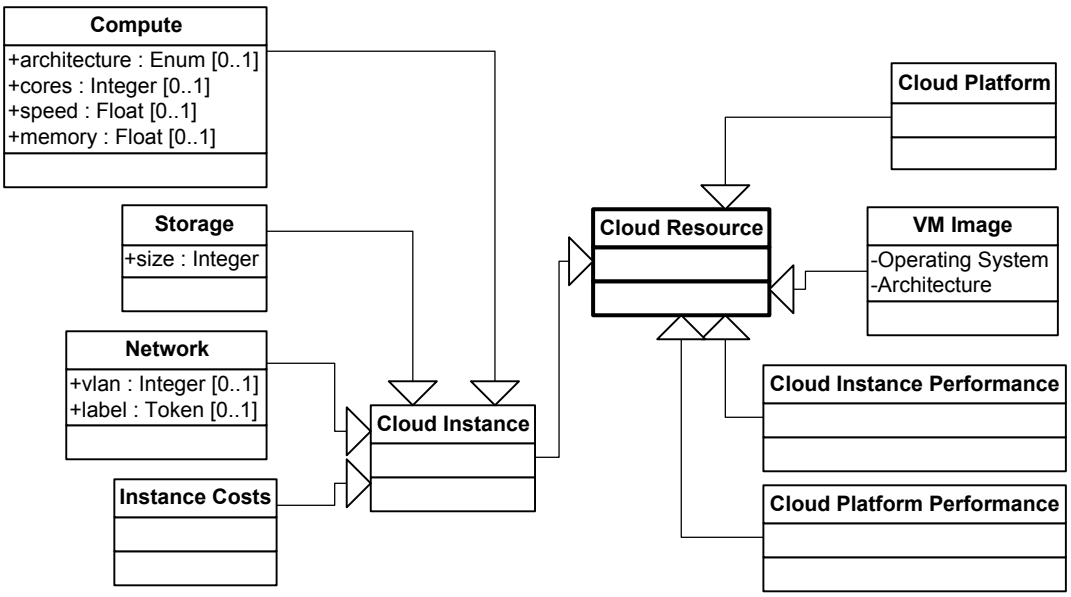

Figure 73: Cloud Resource Model

7.1.5 Cloud Platform Model

Cloud platforms relevant for this work, are IaaS providing cloud platforms. Models for the actual provided resources and their costs are provided in subsequent sections. The cloud infrastructure model as shown in Figure 74 only covers those aspects, which are of relevance to users apart from the provided resources and their costs.

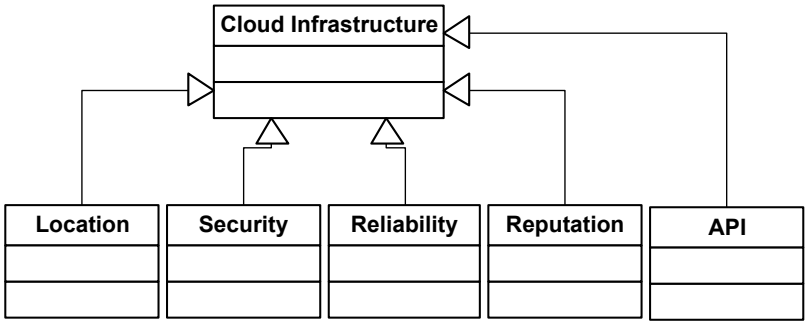

Figure 74: Cloud Infrastructure Model

7.1.6 Cloud Infrastructure Resource Model

Typically Cloud Infrastructure Resources cannot be reserved with fully adjustable Compute, Storage and Networking capacities. Rather than that, Cloud Infrastructure Resource Providers offer differently sized instances of Cloud Resources, i.e. packages with small, medium or large, fixed Compute, Cloud and Networking capacities. For the Cloud Federation Broker there must be way for comparing the capacities of an instance from one cloud provider with an instance from another cloud provider. This would ideally be done before service deployment, either based on benchmarking, or based on a *standard cloud computing unit*. Alternatively however, a rough inter-provider mapping of instance capacities can be conducted during service run-time.

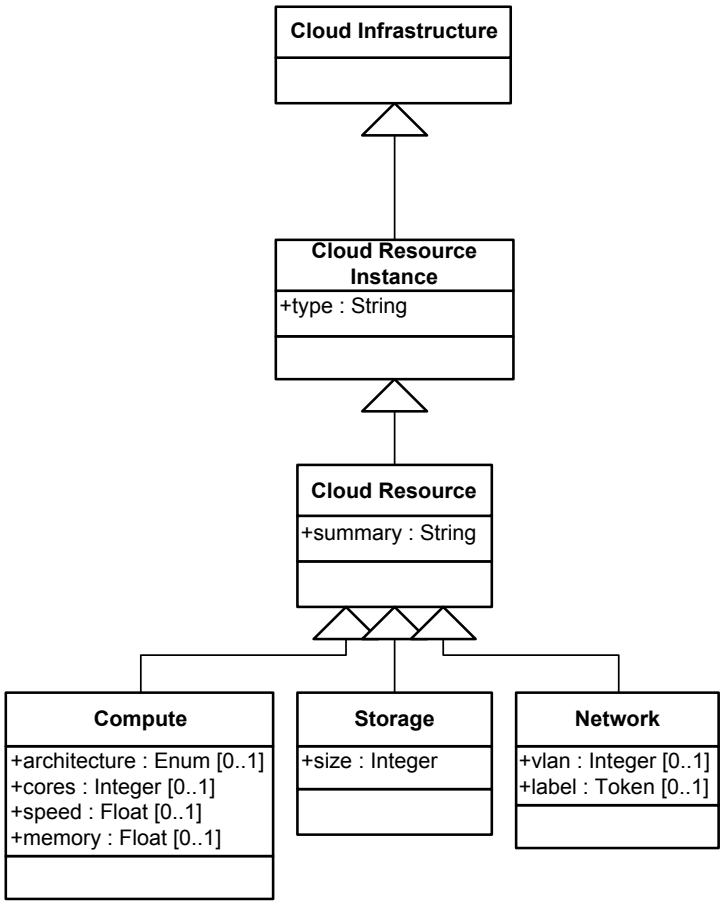

Figure 75: Cloud Infrastructure Resource Abstraction (bsd. on [111])

7.1.7 Cloud Infrastructure Resource Cost Model

Public Cloud Infrastructure Resources are provided at either static or flexible (spot-on prices) costs. On-demand allocation and provisioning of Cloud Infrastructure Resources usually involves a minimal lease time, which is the minimal time resources are being charged for, even if not utilized for the entire time. This impacts the strategy of cost-efficiently provisioning resources, since short-term, on-demand usage comes at a cost which can be significantly higher than linear (e.g. per minute) pricing models would let assume. Therefore this factor needs to be modelled into the Cloud Infrastructure Cost Model as shown in Figure 76.

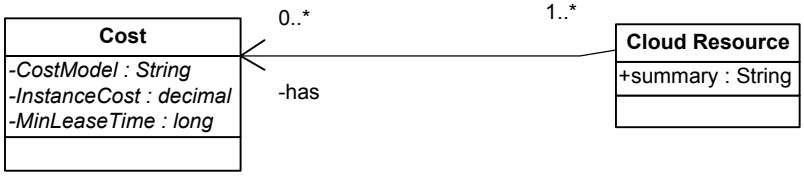

Figure 76: Cloud Infrastructure Resource Cost Model

7.1.8 Policy Information Model

Policies are governing the overall resource allocation and platform selection process. Many different types of policies need to be defined. As shown in Figure 77 no matter how complex compositions of policies might get, policy sets are made up of policy rules and policy groups. Each rule is made up of one or several conditions and one or several actions to be carried out upon conditions have been met.

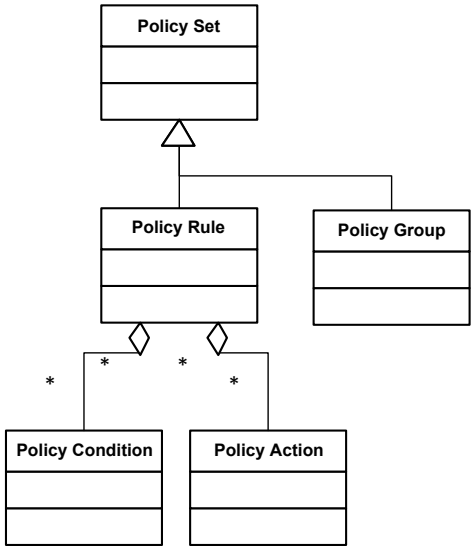

Figure 77: Policy Information Model

7.2 QoS-aware Multi-Cloud Brokering Operations

Based on the QOSMUC lifecycle management framework, as described in section 6.1 and the QOSMUC resource allocation and QoS management framework described in 6.2, the following sections detail the involved operations for:

1) Multi-Cloud NGN Service Lifecycle Management Initialization
2) Multi-Cloud Resource and NGN Service Monitoring
3) Multi-Cloud Resource Performance and Service Quality Analysis
4) Multi-Cloud Resource Performance Control and QoS Improvement Planning
5) Multi-Cloud Resource Provisioning and Service Configuration and Activation Execution
6) Service Decommissioning

As explained in chapter 6, whereas the full service management lifecyle comprises all operations from 1) - 6), the highly dynamic resource allocation and QoS management control loop revolves around operations 2) – 5), i.e. the MAPE control loop.

7.2.1 Multi-Cloud NGN Service Lifecycle Management Initialization

As summarized in Figure 76, for preparing the management of the NGN service lifecycle, several procedures for 1) enabling the provisioning of Cloud resources, 2) for enabling the configuration and activation of NGN services, 3) for enabling the collection and processing of data, 4) for managing the resource inventory and 5) for enabling the management of QoS have to be conducted.

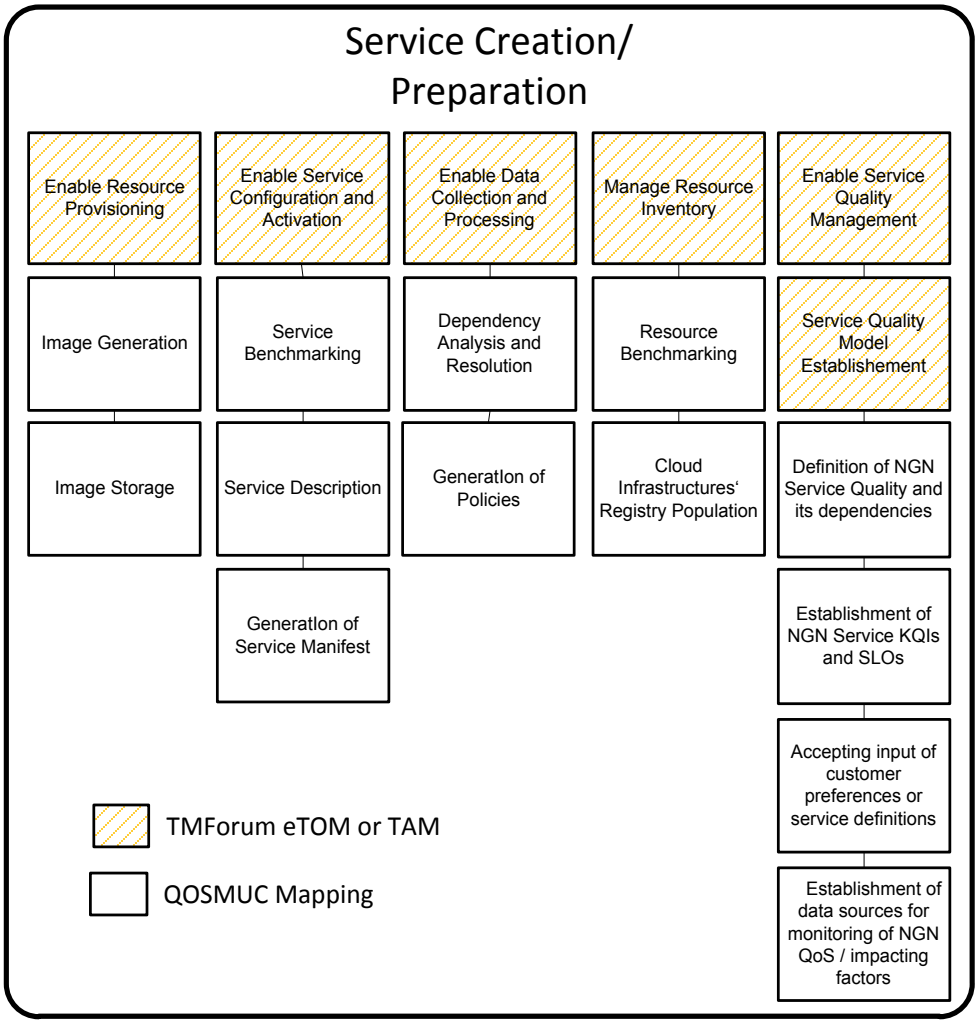

Figure 78: Service Creation / Preparation

Enabling the provisioning of Cloud resources

Image Generation: Services to be dynamically deployed on multiple Cloud infrastructures need to be packaged within VM Images. Starting of a particular service instance on a particular Cloud infrastructure involves the specification of the pre-packaged location (which can be stored locally, at the premises of the particular Cloud infrastructure, or remotely retrievable via a publicly accessible URL) as part of the Cloud instance provisioning request.

Image Storage: The pre-packaged VM image needs to be stored either at the premises of the particular Cloud infrastructure (improves start-up / deployment time), or at a remotely location publicly accessible and addressable via a URL.

Enabling the configuration and activation of NGN services

Service Description: The following information about the particular NGN service has to be provided to the system:

- Information about the *service* to be deployed on cloud resources, including QoS requirements, interdependencies, service type and service interface according to the NGN service model introduced in 7.1.1.

- Information about the *resource and resource capacity requirements* of a particular NGN service, including cloud instance requirements, VM Image / operating system, cloud instance performance, according to the service resource requirement model introduced in 7.1.4.

- Information about service management requirements, such as scaling rules,

- Information about the *services' interdependencies*, including dependent service components, required facilitating components, NGN service control dependencies and NGN OSS/BSS dependencies, according to the telecommunication service interdependency model introduced in 7.1.3.

Service Benchmarking: Whereas some of the required information is easily obtainable, like for instance the information about the available resources, including their costs of a given cloud resource provider, other information as for instance the capacity requirements of a given service is typically not easily available, or not available at all. In order to come up with some of the hardly available or sometime unknown information, mechanisms like benchmarking of cloud-based NGN service's capacity requirements, QoS-related requirements can be used in order to acquire the required information.

Generation of Service Manifest: The generation of the service manifest, involves packaging, structuring and provisioning of aforementioned service-related information in a machine-readable fashion for the configuration of orchestration and provisioning mechanisms.

Enabling the collection and processing of data

Dependency Analysis and Resolution: Based on the service descriptions / the service manifest generated in the context of service configuration and activation enablement procedures (see above) and based on resource descriptions provided by resource inventory management processes (see below), the dependency analysis extracts the information required for managing the NGN service on federated cloud infrastructures.

Generation of Policies: Based on aforementioned extracted information, the policy generation process involves the description and storage of atomic policies and composed policies specified according to the policy information model as described in section 7.1.8, such as:

- *Policies* that govern the behavior of the resource allocation and quality control processes
- *Provisioning workflow policies* that are utilized by orchestration mechanisms of service lifecycle management processes

Management of resource inventory

Cloud Infrastructure Registry Population: Information about the Cloud resources available and provided by multiple cloud platforms is aggregated and made available for later use. The following cloud resource related information is captured:

- Information about available *resources* that are provided by a particular cloud platform, according to the cloud infrastructure resource model introduced in 7.1.6.

- Information about the *costs and the pricing* of a particular cloud resource, according to the cloud infrastructure resource cost model introduced in 7.1.7.

Resource Benchmarking: Whereas some of the required information is *easily obtainable*, like for instance the information about the available resources, including their costs of a given cloud resource provider, other information as for instance the actual resource capacity (as this needs to be specified in terms of capacity for serving a particular service's amount of workload) is typically not openly available. In order to come up with some of the hardly available or sometime unknown information, mechanisms like *benchmarking* of cloud resource capacities, as well as cloud platform performance can be used in order to acquire the required information.

Enabling the NGN service quality management

QoS requirements and QoS impacting factors (and their impact on the overall QoS) of a given telecommunication service are a special case of information, which is sometime difficult to obtain. Whereas for simple web services some simple metrics like service execution time might be sufficient for defining a metric and a level for the required QoS, telecommunication services frequently several more QoS impacting factors. Therefore it is crucial to establish a model, which sufficiently describes a particular telecommunication service's QoS requirements. According to the TMForum's Technology Application Map [104], QoS model establishment, involves the following operations.

Definition of an NGN service's QoS dependencies: QoS dependencies, such as the impact of network performance, resource performance on the overall end-to-end QoE/QoS can either be measured or sometimes be looked up in respective telecommunication standards, as for instance the impact of network impairments on Voice or Video over IP quality is standardized

in ITU's quality of telecommunication services models, objectives and dependability plan [39]. However, the impact of resource performance on a particular service's quality can typically not be looked up as this is highly depending on the particular resource, as well as on the particularities / behavior of the service. Such dependencies need to be analyzed by benchmarking techniques, where QoS is determined under different resource performance conditions, and needs to be determined for each and every utilized resource.

Establishment of Key Quality Indicators (KQIs) and Service Level Objectives (SLOs): KQIs can be deducted from the measurements or standards obtained by the QoS dependency analysis as described before. SLOs however, are determined individually, i.e. based on specific business models / product strategies and SLAs, either high quality provisioning of a particular service can be demanded, or best-effort provisioning might be sufficient. In telecommunications, clear standards (such as [39]) or different service classes differentiating high service quality from best-effort service quality exist.

Accepting input from customer contracts or service definitions: A combination of both, allowing the user to specify the QoS objectives, and further refining those requirements either based on measurements, or based on above mentioned standards and the application their of.

Establishment of data sources for monitoring of the NGN QoS and QoS impacting factors: Establishment of data sources for QoS monitoring in federated Cloud environments, involves the dynamic configuration, provisioning and deployment of monitoring agents and the dynamic aggregation of QoS related monitoring data. As NGN service instances, with changing resource capacities dynamically change, and as service instances are dynamically deployed on multiple cloud platforms, the lifecycle of these "monitoring data sources" involves autonomous deployment, activation, auto-registration at monitoring aggregation components as well as the flexible release of the latter.

The information about *QoS requirements and QoS impacting factors* of multi-cloud-based NGN services, needs to be provided to the system according to the telecommunication service QoS model introduced in 7.1.2.

7.2.2 Multi-Cloud Resource Performance and NGN Service Quality Monitoring

Multi-Cloud resource and NGN service monitoring, represents an intrinsic cornerstone of the MAPE-based resource allocation and quality management control loop (i.e. the M in MAPE). Resource performance monitoring and service quality monitoring processes, as explained in section 5.5 account for the initial processes of eTOM's resource performance management (eTOM) and service quality management (eTOM) operations domains. Mapped to multi-cloud resource and QoS management of NGN services, as shown in Figure 83, the following required mechanisms are identified.

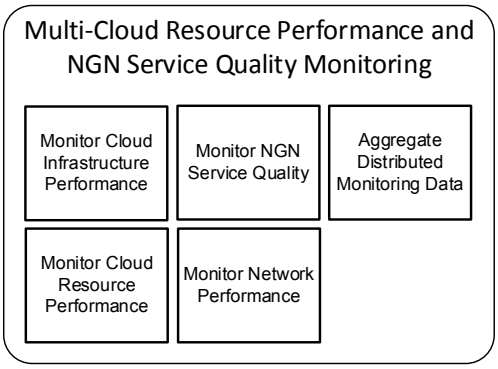

Figure 79: Multi-Cloud Resource and NGN Service Monitoring Processes

Resource performance monitoring processes (eTOM):

In the case of federated cloud environments, comprised of multiple cloud platforms and the cloud resources each of those platforms are offering, resource performance monitoring involves processes for monitoring of cloud infrastructure performance, cloud resource performance, network performance (both, as a resource as well as a QoS impacting factor, as described in the service quality section below). The relevant monitoring data is shown in Figure 80:

Cloud Infrastructure Performance Monitoring: Monitoring of the performance of cloud infrastructures involves monitoring of the performance of its cloud management platform, i.e. mainly the cloud instance provisioning performance of the platform.

Cloud Resource Performance Monitoring: Monitoring of cloud resources performance involves monitoring of memory, computing capacity / utilization or storage capacity / utilization.

Network Performance Monitoring: Monitoring of network performance involves monitoring of network performance parameters such as packet delay, jitter, packet loss and bandwidth.

Multi-Cloud Monitoring Data Source Lifecycle Management: As introduced in section 7.2.1, the establishment of data sources for QoS monitoring in federated Cloud environments, involves the dynamic configuration, provisioning and deployment of monitoring agents and the dynamic aggregation of QoS related monitoring data. Since NGN service instances, with changing resource capacities, are dynamically changing in numbers and in their location, and since service instances are dynamically deployed on multiple cloud platforms, the lifecycle of these “monitoring data sources” involves autonomous deployment, activation, auto-registration at monitoring aggregation components as well as the flexible release of the latter.

Service quality monitoring processes (eTOM):

In the case of cloud-based NGN services, the monitoring of NGN service quality and service quality impacting factors, involves the application of monitoring processes at the application layer, but also monitoring of the networking layer, as final End-to-End QoS of a cloud-based telecommunication service requires network related service quality monitoring also.

NGN Service Quality Performance Monitoring:

The overall quality of an NGN service typically does not depend on a single parameter. In contrast to typical Web services, where service quality in many cases is determined by service execution performance (measured in units such as service execution delay), typical NGN service have 1) a signaling component, and 2) a voice/video streaming component. Both, signaling and multimedia quality strongly depend on network performance, but surely also on the resource performance. Therefore monitoring of NGN service quality and service quality impacting factors, requires monitoring of application layer parameters related to the NGN signaling performance (e.g. retransmissions, service execution time, signaling delay), NGN voice / video quality (e.g. VoIP quality measured in PESQ), network performance monitoring (as described above). As there is no single metric defining the QoS of NGN services the NGN Cloud Broker's monitoring solution needed to allow for flexible, self-defined provisioning of monitoring metrics (including the integration of monitoring agents allowing monitoring of the latter).

Summary: Multi-Cloud resource performance and service quality monitoring involves the dynamic deployment of different sources of monitoring data and the dynamic aggregation of monitored data as depicted in Figure 80. The monitoring knowledge base needs to be sufficiently expressive for performance and QoS analysis and planning operations to be able to draw accurate conclusions regarding cloud platform performance, cloud resource performance and NGN service quality and to accurately plan optimal platform selection and resource allocation.

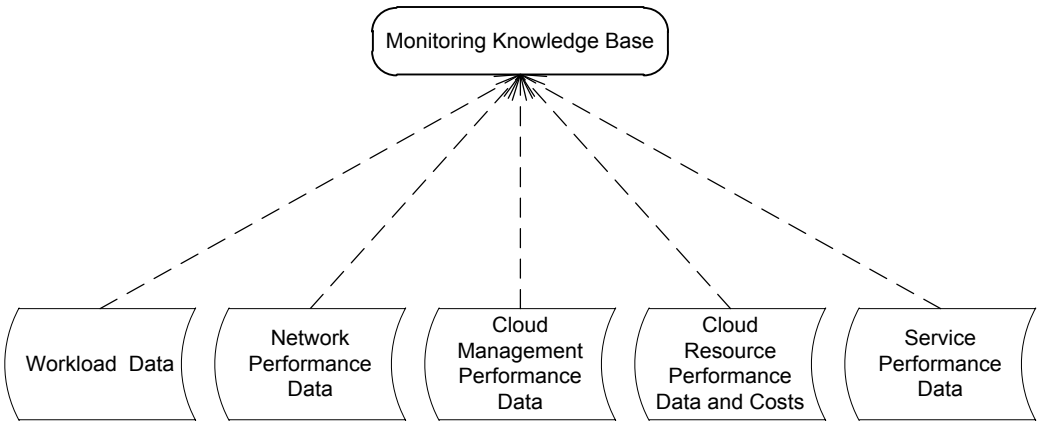

Figure 80: Monitoring Data aggregated by Cloud Federation Broker

7.2.3 Multi-Cloud Resource Performance and Service Quality Analysis

Resource performance and service quality analysis represents an intrinsic cornerstone of the MAPE-based resource allocation and quality management control loop (i.e. the "A" in MAPE). Resource performance and service quality analysis processes, as explained in section 5.6 account for the subsequent processes of eTOM's resource performance management (eTOM) and service quality management (eTOM) operations domains. Mapped to multi-cloud resource and QoS management of NGN services, as shown in Figure 81, the following required mechanisms are identified.

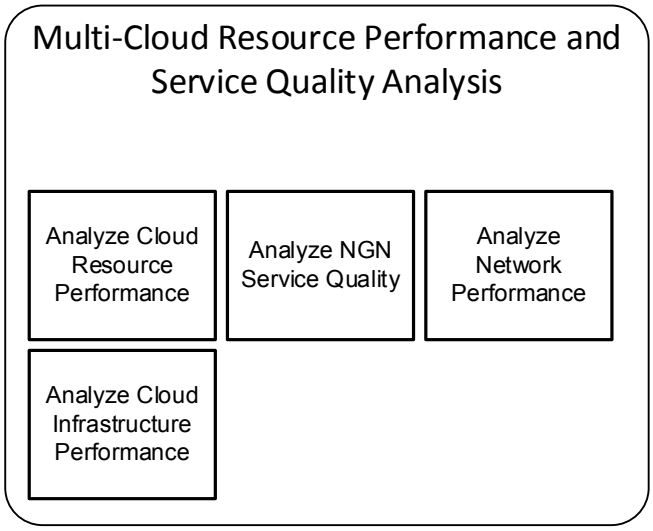

Figure 81: Multi-Cloud Resource Performance and Service Quality Analysis

Resource performance analysis processes (eTOM):

In the case of federated cloud environments, comprised of multiple cloud platforms and cloud resources that each of those platforms are offering, resource performance analysis involves platform, cloud resource and network performance analysis. Analysis involves comparison of current resource performance parameters with policies / thresholds, as well as forecasting mechanisms (e.g. prediction / trending), again comparing the forecasted values with policies / thresholds.

Cloud resource performance analysis:

The analysis of cloud resource performance involves analysis of compute, storage and network performance related data.

Cloud infrastructure performance analysis:

The analysis of cloud infrastructure performance involves analysis of cloud infrastructure management performance related data.

Network performance analysis:

The analysis of network performance involves analysis of connectivity / IP performance (i.e. packet delay, jitter, packet loss and bandwidth) related data.

Service quality analysis processes (eTOM):

In the case of cloud-based NGN services, the analysis of NGN service quality and service quality impacting factors, involves the analysis of application layer and analysis of the networking layer. The analysis involves both, comparison of actual service quality values with thresholds, as well as comparison of QoS impacting factors with pre-defined thresholds.

7.2.4 Multi-Cloud Resource Performance Control and QoS Improvement Planning

Resource performance and service quality improvement planning represents an intrinsic cornerstone of the MAPE-based resource allocation and quality management control loop (i.e. the "P" in MAPE). Resource performance and service quality improvement planning, as explained in section 5.6 account for the subsequent processes of eTOM's resource performance management (eTOM) and service quality management (eTOM) operations domains. Mapped to multi-cloud resource and QoS management of NGN services, as shown in Figure 82, the following required mechanisms are identified.

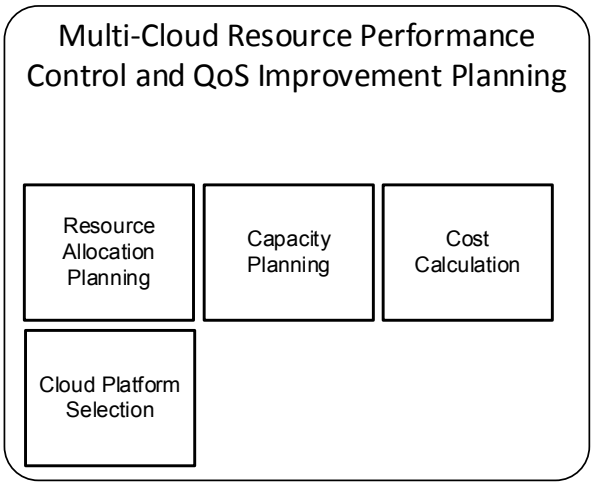

Figure 82: Multi-Cloud Resource Performance Control and QoS Improvement Planning

Resource performance controlling processes (eTOM):

In the case of this work, resource performance controlling mechanisms are realized through elastic resource allocation mechanisms, which control the overall resource performance by *planning and dynamically scaling resource capacities* according to the current service demand / workload.

Cloud Resource Capacity Planning:

Cloud resource capacity planning involves processes for workload prediction and capacity forecasting. Resource performance control is achieved by allocating sufficient resource capacities for serving NGN service workloads.

Resource Allocation Planning (including cost calculations):

Cloud resource allocation planning involves mapping of required resource capacities to available cloud resources. As different cloud resource with different capacities are provided from cloud platform to cloud platform, as well as different cloud resources are offered by each single cloud platform, the resource allocation process involves determination of optimal location, as well as optimal resource types / cloud instance types. “Optimality” of cloud resource allocation processes in the context of this work is determined by levels of resource efficiency, which can directly be mapped to cost-efficiency.

Service quality improvement processes (eTOM)

In the case of this work, service quality improvement relates to cloud platform selection processes, which identify the currently optimal cloud platform for NGN service deployment, which offer *“improved” service qualities*.

Cloud Platform Selection:

Apart from assuring that sufficient resource capacities are allocated for a given service, cloud platform selection mechanism represent the core mechanisms for QoS improvement from the perspective of the cloud platform and resource brokering entity. Migration of service instances from one cloud platform to an alternative cloud platform, especially in the case of network degradation or platform outages / performance degradation represents a core strategy for QoS improvement / assurance in federated cloud environments.

The overall resource allocation planning also involves *resource cost optimization mechanisms*, which, however, are only applied after performance controlling as well as quality improvement processes have been applied.

7.2.5 Multi-Cloud Resource Provisioning, Service Configuration and Activation Execution

Execution of resource provisioning, service configuration and activation represents an intrinsic cornerstone of the MAPE-based resource allocation and quality management control loop (i.e. the “E” in MAPE). Resource provisioning, service configuration and activation, as explained in section 5.6, account for the processes of eTOM’s fulfillment domain of eTOM’s

resource and service management operations. Mapped to multi-cloud resource and QoS management of NGN services, as shown in Figure 83 the following required mechanisms are identified.

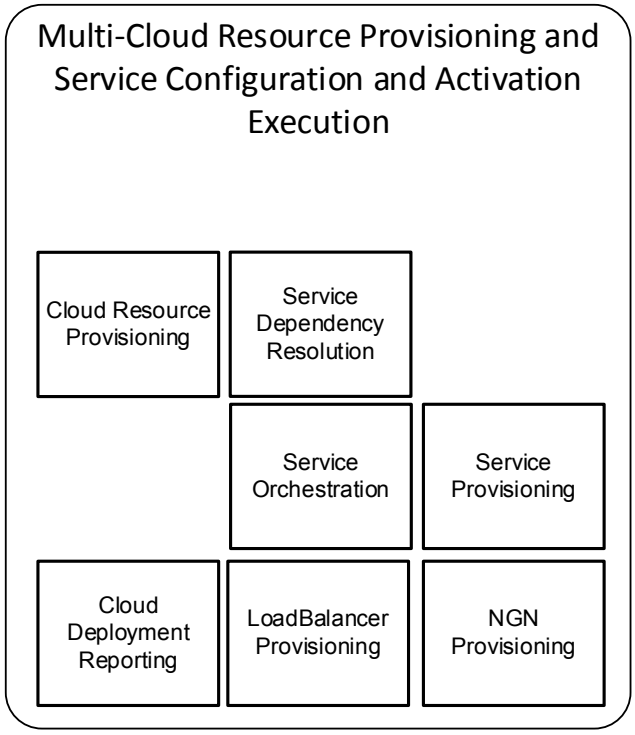

Figure 83: Multi-Cloud Resource Provisioning, NGN Service Configuration and Activation

Resource Provisioning and Activation (eTOM):

- Resource Allocation and Delivery (eTOM)

Cloud resource provisioning, Reservation: Resource Allocation and Delivery in the case of this work, relates to the actual process of reserving and provisioning of cloud resources (across multiple cloud platforms).

- Resource Configuration and Activation (eTOM)

Cloud resource provisioning: Resource configuration and activation in the case of this work, is tightly coupled with the preceding process of reserving cloud resources, where “resource configuration and activation” relates to the process of providing the actual configuration data and details of the virtual images to be started up on the particular allocated cloud resource

Service Configuration and Activation

- Allocation of specific Resources to Services (eTOM)

Service Provisioning: Allocation of specific Resources to Services in the case of this work relates to the process of adding or removing a service instance to/from the pool

of contributing instances, i.e. adding new service instance end-points to the service environment (e.g. load-balancing, data-base, service control and delivery integration).

- Service Implementation and Configuration (eTOM), Service / NGN Provisioning

Service provisioning: Service implementation and configuration in the case of this work, relates to the workflow of orchestration, i.e. the executing of provisioning tasks, configuring several service related elements, at the cloud layer (e.g. dependent service elements like load balancing or database or monitoring elements) as well as at the NGN layer (e.g. service control and service delivery elements).

- Service Activation (eTOM)

Service Provisioning: Service Activation in the case of this work, represents the final finishing of a control loop, after which the (new) service configuration / topology is set ready to be utilized, i.e. the service is ready to be consumed.

The end of each execution phase marks the finish of each resource allocation and quality management control loop. Important for the overall business process management of a telecommunication service providing enterprise is the appropriate reporting of:

- utilized resources / resource costs (QOSMUC KPI), compare Figure 53 “Report Resource Data”, which represents a vital parameter for the overall billing processes

- QoS / SLA assurance performance (QOSMUC KPI), compare Figure 54 “Report Service Data” which represents a vital parameter for the overall operations as well as billing processes (in case of incurred penalties due to SLA violations)

7.2.6 Multi-Cloud Resource, Service and Data Decommissioning

As summarized in Figure 84, the end of each service management lifecycle involves procedures for 1) releasing the service from the pool of managed / brokered service, 2) releasing resources from the pool of activated / utilized resources, 3) release of related data (i.e. configurations, VM Images, measurements, etc.).

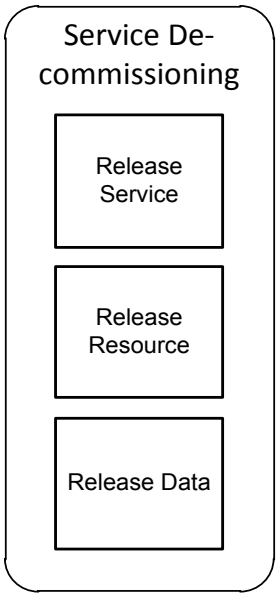

Figure 84: Service Decommissioning

Chapter 8

Specification and Instantiation of the NGN Cloud Broker

8.1 Key Assumptions and Design Aspects

The following sections provide some boundaries and guidelines that are used for the design of the NGN Cloud Broker, on the one hand by postulating some key assumptions about involved administrative domains and pre-established relationships between them and on the other hand by describing several key design aspects that should be adhered to. Finally several design options are contemplated and evaluated and an explanation for the specific choice is provided.

8.1.1 Key Assumptions

For the design of the NGN Cloud Broker the following key assumptions on the involved administrative domains and the pre-established relationships between them are made.

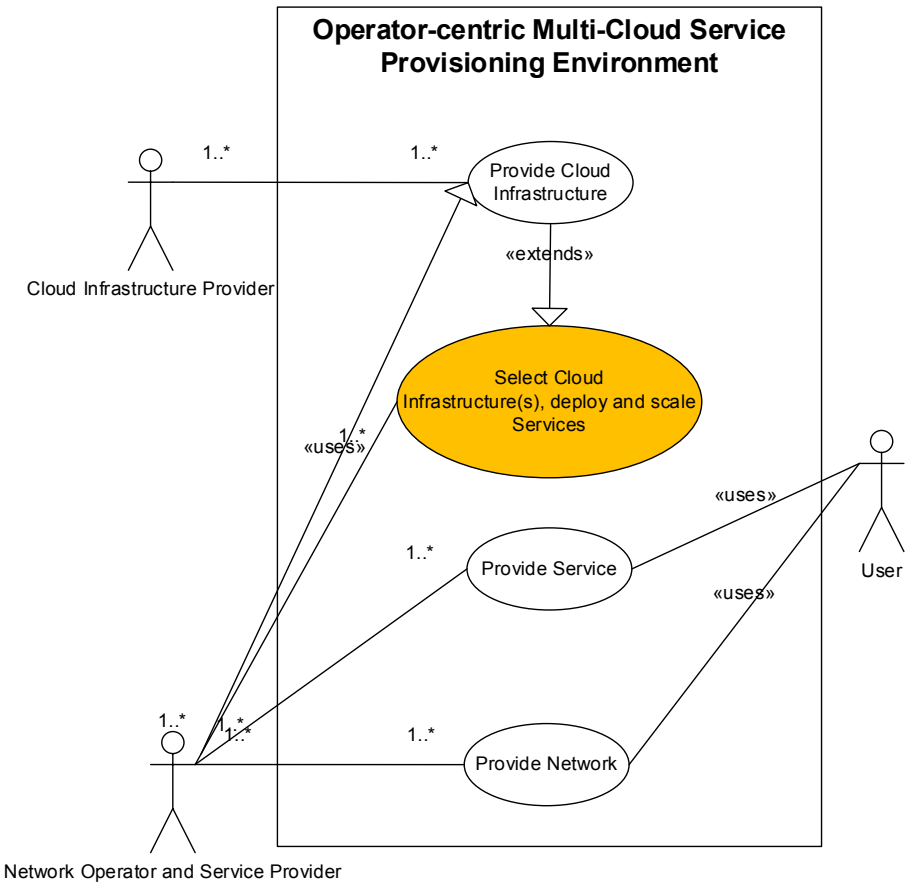

Figure 85: Merged NGN Operator and Service Provider Perspective

1) The NGN network provider is either an NGN service provider itself, or trusted relationships, legally as wells as from a technological interworking perspective exist between NGN operator and service provider. A trusted infrastructural relationship in this regard assumes secured, but accessible interfaces between each domain, the NGN network and the NGN service domain. Systems of the NGN domain have full access to the systems in the Service domain, and vice versa the Service domain, including the NGN Cloud Broker has access to provisioning interfaces of the NGN domain. Without restricting the capabilities and the applicability of the NGN Cloud Broker, for the remainder of this work the entity relationship is simplified, by assuming that the NGN Operator itself is an NGN service provider as shown in the following Figure 85.
2) It is assumed that the NGN service provider already has established the contractual conditions allowing him to reserve and utilize cloud infrastructure resources from multiple cloud providers on an on-demand basis.
3) It is assumed that cloud infrastructure resource providers expose open resource provisioning interfaces which allow the basic resource management operations required for allocating and dismissing cloud resources as defined earlier.
4) NGN Services have already been preconfigured, packaged and are provided as an image accessible through public internet.

8.1.2 Critical Design Aspects

Guiding the design of NGN Cloud Broker, several quality criterions and design principles need to be taken into account. From an NGN infrastructure provider perspective, offering NGN services to its customers, there are several requirements to be fulfilled by a cloud brokering system such as the NGN Cloud Broker, as this component resides at a highly critical position from a business perspective. System failures would lead to full service outages, which could easily have significant negative impact on its business and customer retention. The following quality factors, as shown in Figure 86, are taken into account for designing the NGN Cloud Broker.

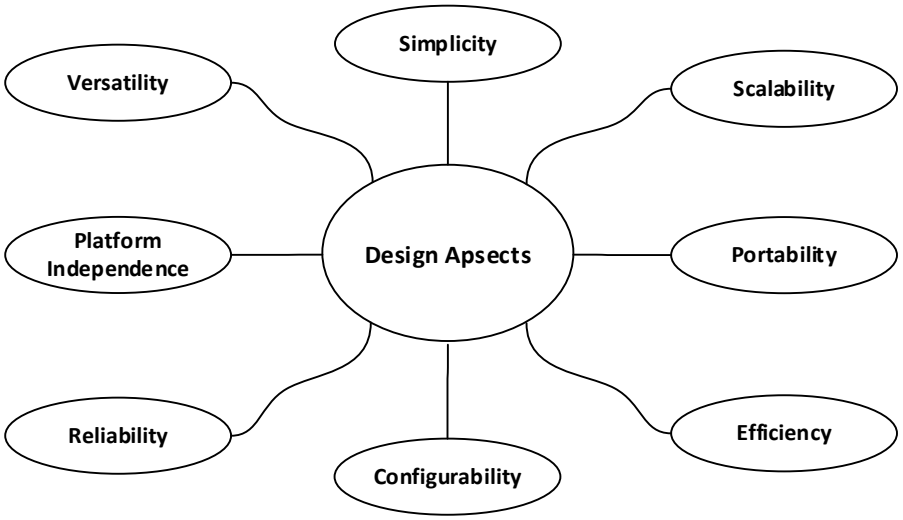

Figure 86: Design Aspects

The following Table 13 provides an explanation for the design aspects taken into account.

Table 13 Key Design Aspects

Scalability The system has to be designed in a way that allows inherent scalability. With growing demand a system like the NGN Cloud Broker should be designed in a seamlessly scalable fashion. Scalability is required for both, a growing number of services to be brokered across cloud infrastructure federations as well as a growing number of user demand, utilizing such services.
Reliability Reliability is a core aspect to be taken into account for designing such a critical component. Not only should mechanisms be incorporated which allow for monitoring the health of such a component, but also for allowing incorporation of redundancy means, by which an system failure or outage can quickly be mitigated.
Simplicity Simplicity is a principle which should govern the design of such a system as much as possible. The more complexity and system logic is incorporated into such systems, the more prone to failure such systems get. Outsourcing of core functionalities to existing systems, which have proven maturity and reliability, is in many cases a valid strategy to be preferred to integrating multiple functionalities into one single component.
Efficiency Wherever algorithms, computation and communication intensive mechanisms need to be employed, a critical analysis of the added value that these mechanisms provide to the overall system performance should be carried out. If the actual gain is rather insignificant, efficiency should rule out performance gains.
Versatility For the NGN Cloud Broker, especially for enabling cloud federation brokerage for a multitude of different NGN services, versatility is a key aspect. The more agnostic the Cloud Broker is designed to service specificities, the more services can be supported.
Configurability As several parameters, constraints and policies determine the behavior of the NGN Cloud Broker and as for each and every managed service these parameters have to be configured in a service-specific fashion, it is important that the NGN Cloud Broker allows easy, but also versatile configurability.

Platform Independence The NGN Cloud Broker needs to interwork with / support as many Cloud Platforms as possible, as with growing choice of available cloud resources, chances of finding more cost-efficient and/or better performing cloud resources increase.
Portability As already explained in section 4.4, there are several deployment scenarios relevant for the deployment of the NGN Cloud Broker. Therefore a high degree of portability, allowing the NGN Cloud Broker to operate in different deployment scenarios as well as modes should be supported.

8.1.3 Evaluated Design Options

Three basic, but important a-priori design decisions are made, related to the actual positioning of the Cloud Federation Broker system, basic means for component interaction and the utilized event processing mechanisms.

Component Placement

The NGN Cloud Broker core needs to interwork with three functional systems, monitoring aggregation systems, network and service provisioning system, and service load balancing system. Whereas it would be possible to integrate the NCB Core with all aforementioned systems, it was decided to position these components at three architecturally different locations, for the following reasons.

The monitoring aggregation system is placed at the NGN OSS level since typical OSSs already provide monitoring aggregation functionalities, which can either be enriched, re-used, or provide additional NGN related monitoring data to the overall monitoring knowledge base required by the NGN Cloud Broker.

The service load balancing system(s) are placed together with the service nodes on the particular cloud platform infrastructures as one or several additional virtual node(s). Whereas the actual load balancing could be executed from the NGN level as well as from the Service level, it would involve extra networking overhead and reduce robustness of the overall signaling architecture. Nevertheless the load-balancing elements need to be flexibly and dynamically provisioned through open interfaces accessible by the NCB.

The NGN and service provisioning system(s) are split into three parts, one part residing at the Cloud Federation Broker level, which issues basic provisioning requests, one part residing at the NGN platform level, exposing north-bound provisioning interfaces to the NCB

controlling service control provisioning functions and a final cloud-based provisioning component also exposing open provisioning interfaces dynamically accessible by the NCB.

Finally the NCB core itself could principally be place anywhere in the overall architecture, but placing the component at the Service layer bears several advantages. First, the telecommunication service provider needs frequent access the NCB for configuration and initialization of new services to be deployed. Second, the NCB needs to provide the service provider with continuous data on resource usage and QoS assurance performance. Third, an interaction with service level components and service enabling elements might be beneficial. Fourth, exploiting the trusted relationship between service layer and network resources, particularly OSS and BSS resources reduces the overhead of securing those connections.

Component Interaction

The NCB core needs secured connectivity to NGN infrastructure resources like OSS and BSS for provisioning and monitoring. Furthermore connectivity to cloud infrastructure resource management interfaces must be guaranteed. Finally, also the direct access to particular cloud resources, e.g. load-balancing systems, needs to be assured.

Event Processing

The centrally kept monitoring knowledge base is continuously updated by monitoring agents and checked for policy and threshold violations, upon which specific alarms are being triggered. With each resource allocation cycle, the NCB core pulls relevant monitoring data from the knowledge based for analysis, planning and execution of appropriate resource allocation actions. This allows for both, re-active, alarm-based fault management actions as well as pro-active ahead-looking capacity planning.

The actual execution of resource allocation actions is conducted in a sequential fashion, through which availability of elements in the orchestration cycle is assured and a step-wise, dependency resolving provisioning of networking and service elements conducted. This allows for failure identification, NCB state-fullness, and potential roll-back and counter measures.

Discussion

Realizing a combination of re-active event-based management and pro-active prediction-based control loops requires de-composition of certain elements and loosely-coupled component interaction. For the re-active event-management, the analysis and planning phase of the autonomic control loop is bypassed for quicker problem resolution. In order to achieve this, the policy evaluation function needs to be de-composed from the overall resource allocation logic and the enforcement function, i.e. the execution of resource allocation, network and service orchestration function needs to expose well defined interfaces.

8.2 NGN Cloud Broker Architecture

For the NGN Cloud Broker being a cross-platform, multi-domain cloud federation, and brokering system there are at least three administrative domains with which the NGN Cloud Broker has to interwork:

1) The NGN OSS and BSS Layer comprising monitoring, service fulfillment / provisioning and service monitoring functions for managing
2) the NGN core layer comprising NGN service control functions
3) the NGN Service Layer comprising service elements, service management, and orchestration functions
4) The Cloud Platform Layer comprising cloud infrastructure resource management functions

From a high-level perspective, as shown in Figure 87, the NGN Cloud Broker's main Input, Operations and Output are:

- *Input:*

  1) Service Description and QoS and Cost Constraints

- *Operations:*

  2) Resource Allocation, including Cloud Platform Selection

  3) NGN Platform, NGN Service and Cloud Platform Provisioning

- *Output:*

  4) Report of Costs and Service Quality

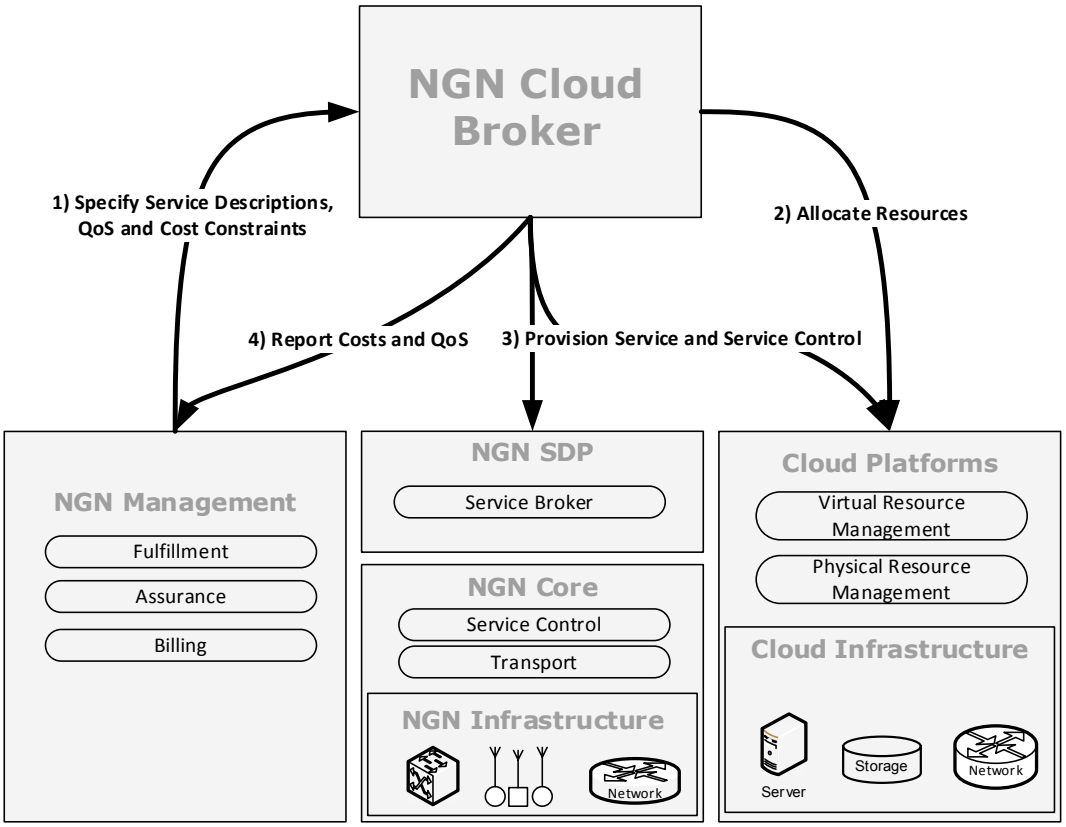

Figure 87: NGN Cloud Broker's main interactions with NGN and Cloud Systems

Based on the generic QOSMUC Framework (as introduced in Chapter 6) and based on the QOSMUC Resource Allocation and Quality Management Control Framework (as developed in section 6.2), the core of the NGN Cloud Broker Architecture is designed as an engine realizing a *MAPE control loop* (compare the QOSMUC Control Loop Framework shown in Figure 66), the "Cloud Federation Brokering Engine".

As shown in Figure 88, the Cloud Federation Brokering Engine realizes the 4 distinc procedural MAPE steps, which are exeuted within each cycle of the control loop.

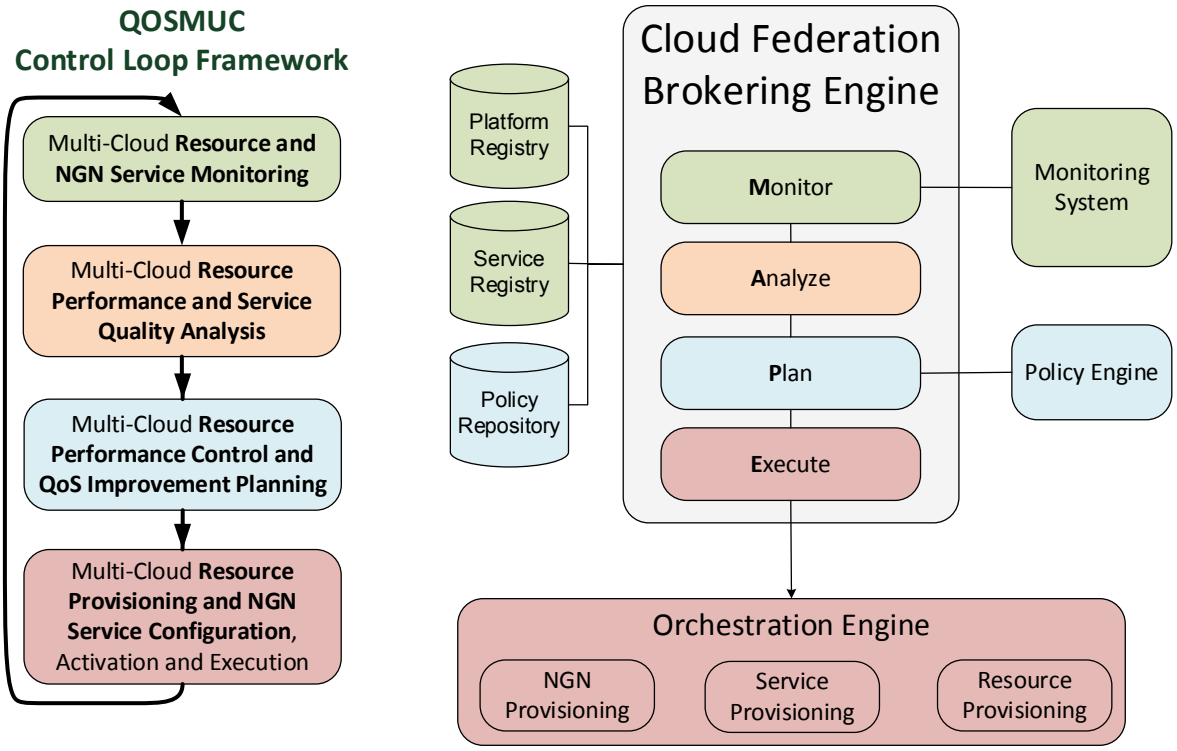

Figure 88 The Cloud Federation Brokering Engine's relation to the QOSMUC Control Loop Framework

For realizing the *Monitoring Part* of the Cloud Federation Brokering Engine, monitoring functions, need to be in place, capable of distributedly monitoring the quality of NGN services, the connectivity / networking parameters between NGNs and Clouds as well as Cloud resource performance parameters. Detailed specification of the monitoring functions of the NGN Cloud Broker / its Cloud Federation Brokering Engine are provided in AII.3.2 NGN Cloud Broker - Monitoring Functions

For realizing the *Analysis Part* of the Cloud Federation Brokering Engine, Monitoring Aggregation and Analysis Functions need to be in place analyzing NGN service quality, network performance, as well as Cloud resource performance. Details on the analysis part are provided in AII.3.5 NGN Cloud Broker - Cloud Brokering Functions.

For realizing the *Planning Part* of the Cloud Federation Brokering Engine, first Policy Evaluation Functions as specified in AII.3.3 NGN Cloud Broker - Policy Evaluation Functions need to be in place. Based on aggregated and analyzed measurement of the previous step and based on the previously evaluated policies, the actual core logic of the NGN Cloud Broker / its Cloud Federation Brokering Engine needs to provide Cloud Platform Selection functions as specified in AII.3.5.1 NGN Cloud Broker - Cloud Platform Selection Functions and Capacity Forecasting and Resource Allocation Functions as specified in AII.3.5.2 NGN Cloud Broker - Capacity Forecasting and Resource Allocation Functions.

For realizing the *Execution Part* of the Cloud Federation Brokering Engine, several provisioning steps have to be orchestrated, including the provisioning of Cloud resources

within previously selected Cloud platforms according to previously determined Cloud resource allocation, the provisioning of NGN core platform and service environment functions as well as NGN service specific functions (e.g. NGN service load-balancing functions). The orchestration functions of the NGN Cloud Broker / its Cloud Federation Brokering engine and interworking orchestration engine part are specified in AII.3.4 NGN Cloud Broker - Service Orchestration Functions.

The aforementioned core functions of the NGN Cloud Broker's Cloud Federation Brokering Engine, together with the interworking functional elements at the NGN management layer, the NGN service layer (core network as well as service environment) as well as Cloud platform layer are depicted in the NGN Cloud Broker's functional Architecture in Figure 89.

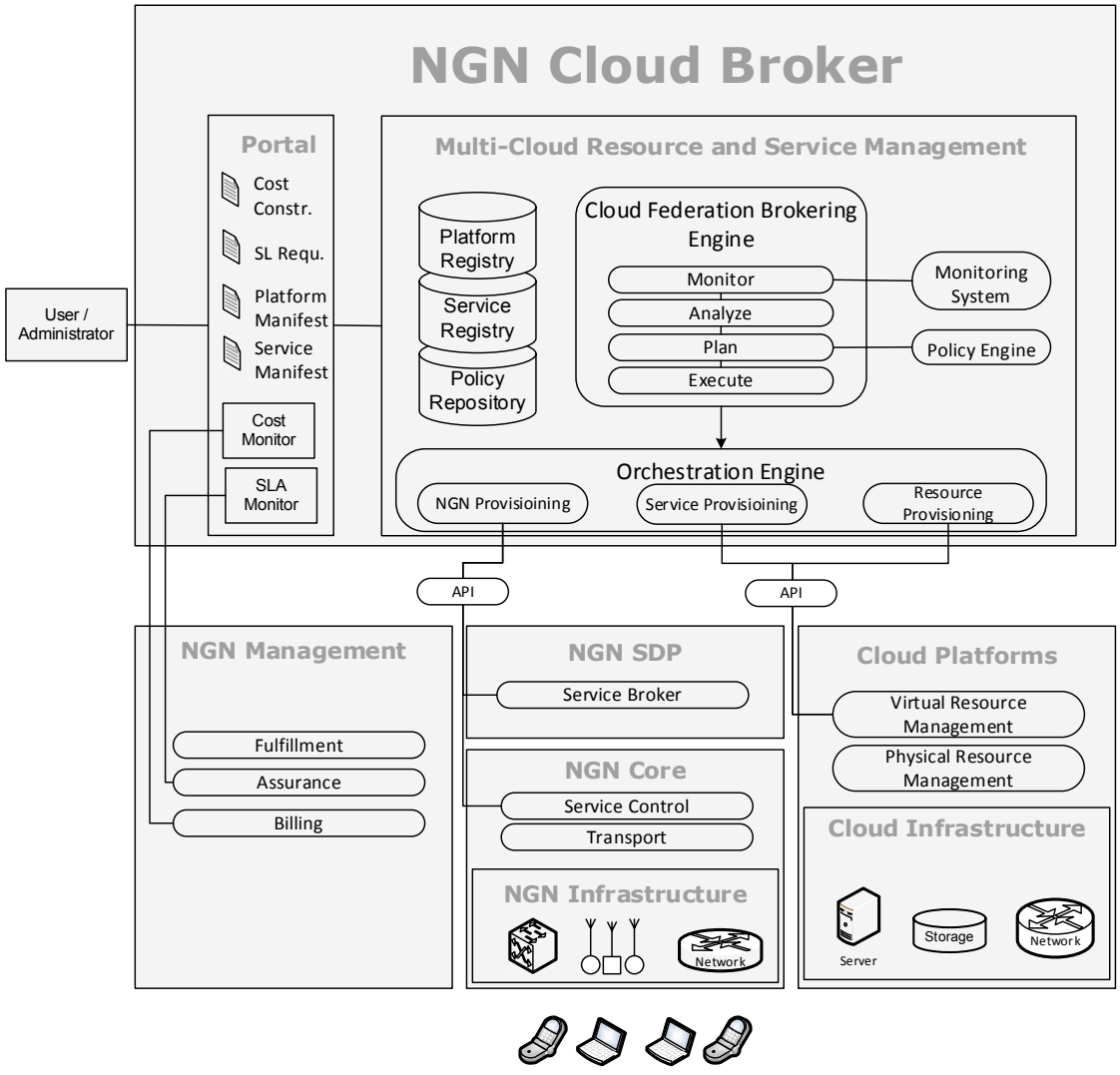

Figure 89: NGN Cloud Broker Functional Architecture

Apart from the previously described core elements of the NGN Cloud Broker architecture, i.e. the Cloud Federation brokering engine with its affiliated systems, Figure 89 also shows the NGN Cloud Broker's Portal, which is mainly used for initialization of the overall system, as well as for displaying / monitoring current costs and SLAs. For initializing the NGN Cloud

Broker, cost constraints (setting upper limits to cloud resource costs), service level / service quality constraints (setting lower levels to the minimum service quality to be achieved), cloud platform manifests (defining cloud resources, instance types of available cloud platforms) as well as NGN service manifests (detailing the service quality model including its dependencies on cloud resource performances and network performance).

As shown in Figure 90 after 1) initialization of the NGN Cloud Broker (involving above mentioned input parameters), 2) cloud resources are allocated and 3) the service is deployed and provisioned. Subsequently, 4) further service orchestration steps are carried out, initializing service control elements at the NGN control platform and service environment level, as well as initializing the required monitoring mechanisms.

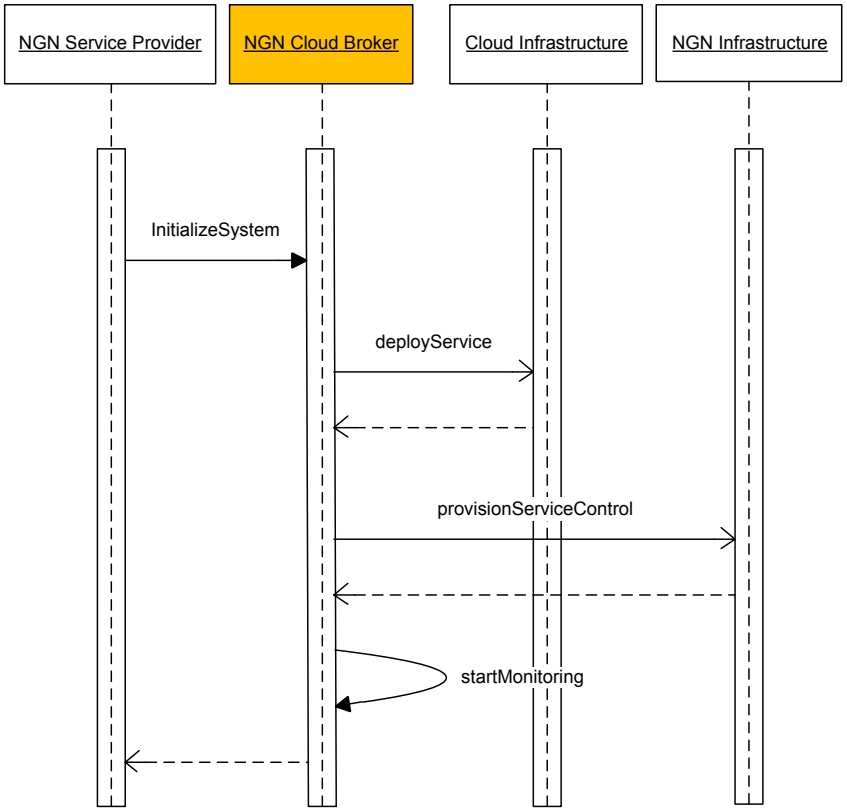

Figure 90: Cloud Infrastructure Resource Provisioning and Service Orchestration

Finally, also shown in the NGN Cloud Broker architecture in Figure 89, are again the interworking domains (NGN management domain, NGN service control and service environment domain, Cloud platform domain), already introduced *Chapter 6 the "QOSMUC Framework"*. Whereas the NGN Cloud Broker itself is taking over several aspects of the NGN management domain (as shown in Figure 60), such as aspects from the NGN management fulfillment domain (through its service and resource provisioning capabilities) and NGN management assurance domain (through its service quality and resource performance management capabilities), the NGN Cloud Broker needs to interwork and

dynamically provision elements of the NGN control and service environment domain as well as provision resources through interworking with APIs at the Cloud platform domain (virtual resource management APIs). The detailed specification of the NGN Cloud Broker is provided in *Appendix II: Detailed Specification of the NGN Cloud Broker* where specification details are provided for:

- AII.1 Interfacing NGN and Cloud Platform Functions

  In this section, the functions needed to interwork with above mentioned domains are specified, which are:

  - AII.1.1 NGN Service Control Platform Functions
  - AII.1.2 NGN Service Delivery Platform Functions
  - AII.1.3 NGN Management Platform Functions
  - AII.1.4 Cloud Service Provisioning Functions

- AII.2 NGN Service and Service Scalability Functions

  In this section NGN service specificities are introduced, including different NGN service invocation protocols and methods as well as mechanisms to scale NGN services horizontally, provided in:

  - AII.2.1 NGN Service Functions
  - AII.2.2 NGN Service Scalability Functions

- AII.3 NGN Cloud Broker Functions

  In this section, the main functions of the NGN Cloud Broker are specified including functions for initializing the NGN Cloud Broker as well as the Cloud Federation Brokering Engine functions (as already introduced in Figure 88), as follows:

  - AII.3.1 NGN Cloud Broker - Initialization Functions
  - AII.3.2 NGN Cloud Broker - Monitoring Functions
  - AII.3.3 NGN Cloud Broker - Policy Evaluation Functions
  - AII.3.4 NGN Cloud Broker - Service Orchestration Functions
  - AII.3.5 NGN Cloud Broker - Cloud Brokering Functions

8.3 Instantiation of the NGN Cloud Broker

The following section outlines the instantiation of the NGN Cloud Broker. Full details on the instantiation of the NGN Cloud Broker are provided in *Appendix III: Detailed Instantiation of the NGN Cloud Broker*. As shown in Figure 91, and as previously mentioned the Core of the NGN Cloud Broker requires interworking with several interfacing systems, located at the NGN management, service control, and service delivery layer as well as located at the Cloud platform layer.

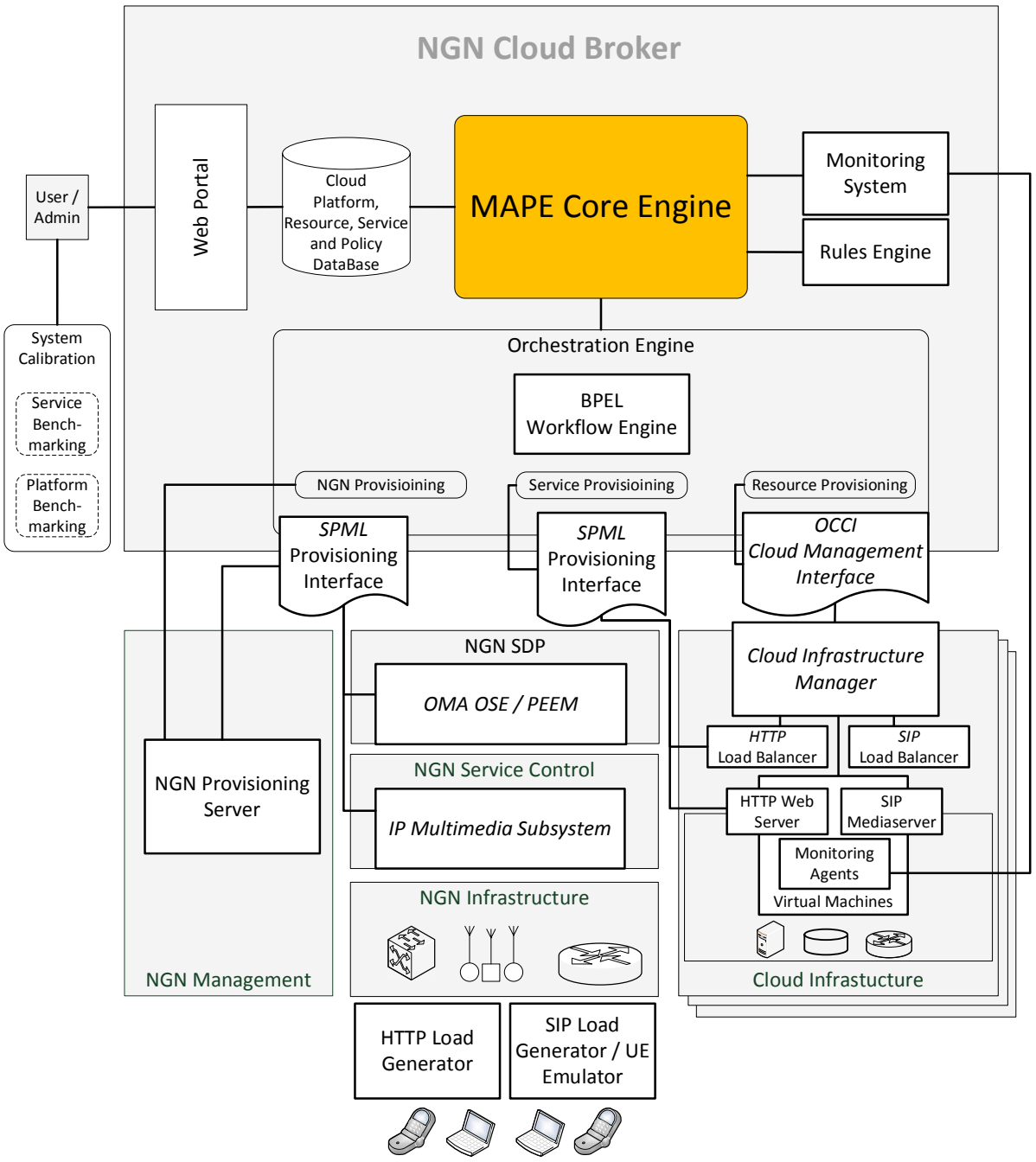

Figure 91: NGN Cloud Broker instantiation and interworking baseline system

For the instantiation of the NGN Cloud Broker high emphasis was put on realism and standard conformant reference implementations and interfaces. As shown in Figure 91 and as explained in detail in *Appendix III: Detailed Instantiation of the NGN Cloud Broker* the instantiation of the NGN Cloud Broker involved selection and integration of elements at the:

- AIII.1 Cloud Layer

  where, as detailed in this section, emphasis was put on selection and integration of a Cloud infrastructure management system which provides open and standardized cloud management APIs (for which the OCCI cloud management interface was identified for reasons provided in this section)

- AIII.2 NGN Layer

  where, as detailed in this section, emphasis was put on realistic and standard compliant reference implementations for the:

  - AIII.2.1 NGN Service Control Platform

    for which a a reference implementation of the 3GPP IP Multimedia Subsystem is selected and integrated into the NGN Cloud Broker as detailed in this section

  - AIII.2.2 NGN Service Delivery Platform

    for which a reference implementation of an OMA compliant NGN Service Environment is selected and integrated into the NGN Cloud Broker as detailed in this section

  - AIII.2.3 NGN Management Platform

    for which a SOA-based provisioning system (according to the requirements of the TMForum for New Generation Operation Support Systems) is integrated into the NGN Cloud Broker as detailed in this section

  - AIII.2.4 NGN Client Layer

    for which SIP-conformant NGN clients as well as load-generators are selected as detailed in this section

- AIII.3 NGN Service Layer

  where again emphasis was put on high levels of realism driving the selection and integration of:

  - AIII.3.1 NGN Application Servers

for which SIP-compliant NGN Mediaservers, providing a typical NGN service (a typical VoIP-based call announcement service) are selected and integrated as detailed in this section

- AIII.3.2 Components for Application Scalability

  for which SIP- as well as HTTP load-balancing systems are selected and integrated as detailed in this section

- AIII.4 NGN Cloud Broker

  where less emphasis was put standard compliance (as there are no standards for such systems yet), however enabling standard-compliant interworking / interfacing with above mentioned external systems. Details on the instantiation of the components of the NGN Cloud Broker (apart from the Core Engine) are provided in:

  - AIII.4.1 NGN Cloud Broker - Monitoring System

    for which details for the instantiated distributed monitoring system, including its monitoring agents, which are distributed across Cloud and NGN service elements are provided in this section

  - AIII.4.2 NGN Cloud Broker - Resource and Platform Registry

    for which details on the instantiation of the registries and repositories for resource and platform registries are provided in this section

  - AIII.4.3 NGN Cloud Broker - Policy Engine

    for which details on the integrated policy/rules engine are provided in this section

  - AIII.4.4 NGN Cloud Broker - Orchestration Engine

    for which details on the integration of the SOA-based orchestration engine are provided in this section

Chapter 9

Validation and Evaluation

The validation and evaluation of the NGN Cloud Broker represents one of the core achievements of this work. Strong focus was not only put on validating the principle approach, but also on evaluating the boundaries within which such cloud brokering system is capable of saving capacity while still assuring NGN service quality. To this end a broad range of experiments and tests under emulated (ideal testbed conditions) and realistic (large-scale, multi-cloud testbed) conditions were carried out.

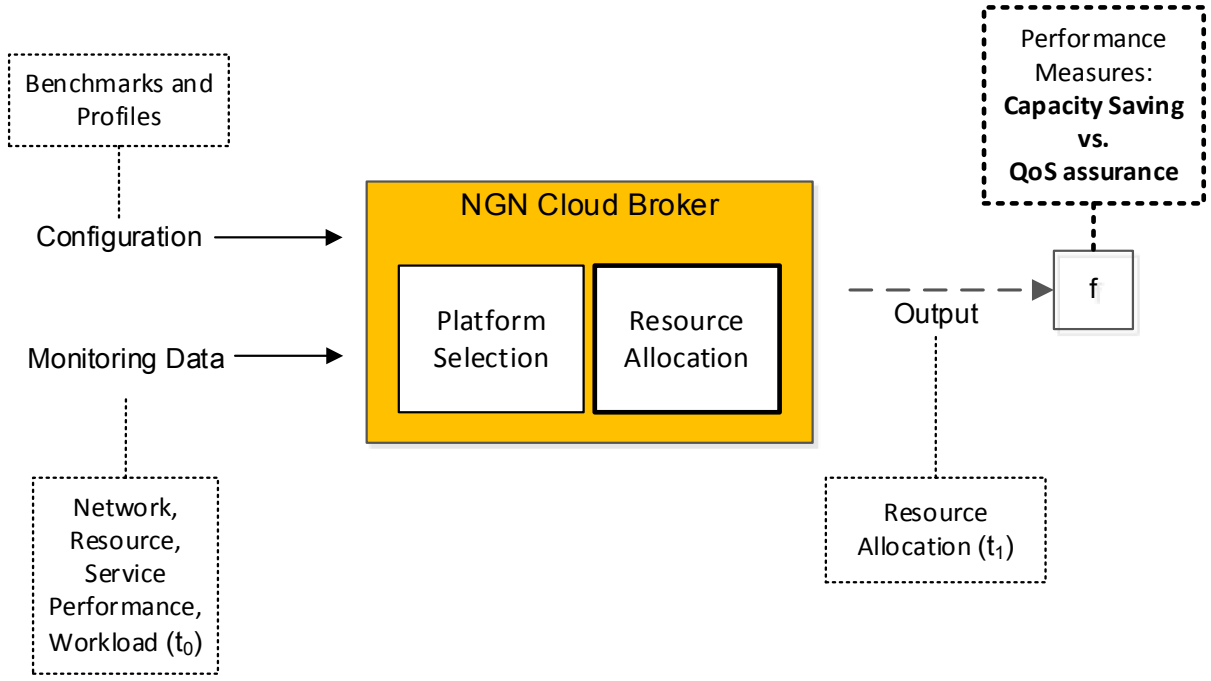

Figure 92: Scope of NGN Cloud Broker Evaluation

As depicted in Figure 92, the actual "output" of the NGN Cloud Broker is quite simple, being an updated allocation of cloud resources / cloud instances across multiple clouds. The scope is further being limited by addressing the parallel usage of cloud resources only from a high level. This means that after the currently optimal cloud platform is identified and selected, only cloud resources of the selected platform are elastically scaled, assuming that for a specific service, a cloud platform's number of resources / capacity can be assumed to be "infinite", at least, large enough for serving the maximum workload at any given point in time.

For configuring the NGN Cloud Broker, significant effort was put on benchmarking real NGN services, cloud resources and network performances, analyzing their interdependencies. An effort, which the performance evaluation revealed was well spent.

The dynamically changing input of the NGN Cloud Broker is the real-time monitoring data of current (t_0) network, resource, service performance and workload, based on which platform selection and resource allocation algorithms try to optimize the current resource allocation, saving capacities where possible, assuring QoS where needed.

The core performance measures of the NGN Cloud Broker are its capacity saving performance, while assuring standard QoS levels.

9.1 Evaluation Plan

The evaluation provided in the following sections will first give a definition of measures for the two KPIs, 1) the capacity saving performance in section 9.2 as well as the 2) QoS assurance performance in section 9.3.

- Subsequently, in section 9.4, the two environments are introduced that were used for carrying out the evaluation of the NGN Cloud Broker, i.e. 1) the FUSECO Playground, providing a NGN testbed for conducting isolated tests, and 2) the BonFIRE pan-European Multi-Cloud testbed.
- Profiling of the interdependencies between NGN service quality, network performance, cloud resource capacities and analysis of real NGN workload distribution is described in section 9.5.
- Ideal and worst-case benchmarks, which will later allow for assessing the performance of the NGN Cloud Broker's resource allocation efficiency, i.e. capacity saving performance, are described in section 9.6
- The actual performance evaluation in both, the isolated as well as the real multi-cloud environment is documented in section 9.7.
- Based on the results of the performance evaluation, the main KPIs, i.e. capacity saving performance and QoS assurance performance, are evaluated in section 9.8.
- Subsequently, non-functional evaluation criteria of the NGN Cloud Broker are evaluated in section 9.9.
- A factorial impact analysis, provided in section 9.10, assesses the impact of the different factors on the overall cost-efficiency and QoS assurance capability of the NGN Cloud Broker.
- The overall hypothesis of this thesis is verified in section 9.11.
- The approach taken in this thesis is compared to other, similar approaches in section 9.12.

9.2 Capacity Saving KPI Measures

The core intention of the NGN Cloud Broker is to save costs, while preserving service quality. In order to do so, the Broker selects a cloud platform, that is 1) capable of delivering the required service quality, and 2) offers the best value for money, i.e. the best performance of

resources / cloud instances (value) at the lowest costs (money). After platform and resources are selected, the actual resource allocation process takes places, which continuously seeks to optimize the current allocated capacity (costs), against the required workload. As shown in Figure 93, in an ideal world, where 1) resource capacities can be chosen completely freely, at any granularity (i.e. cloud instances of any capacity are available), and 2) change of capacity requirements can be accommodated for in no time, at any given point in time only as much capacity is required as needed to serve the current workload.

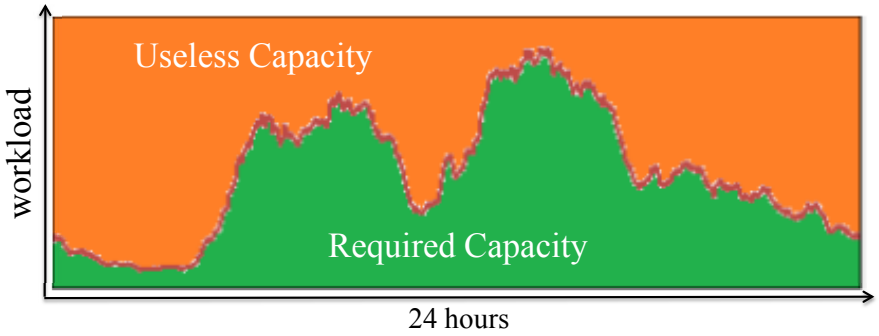

Figure 93: Required Capacity vs. Useless Capacity, Realistic daily NGN workload

Focusing on a single daily workload distribution, as shown in Figure 93, fully overprovisioned, static allocation of capacities (i.e. useless plus required capacity in Figure 93) is determined as the "worst-case", the "non-elastic" capacity required for serving a daily workload. This measure is being used as a reference, against which capacity saving efficiency can be calculated.

Further, a minute-granularity as a reasonable level of discretization of the continuum of time-based workload values / capacities is identified. For assessing upper and lower boundary levels of ideal and worst-case capacity-savings, the following assumptions (similar to [27], published in [22]) are made:

- n := number of VMs = {vm1, vm2, …,vmn}
- m := number of cloud platforms = {cp1, cp2,…,cpm}
- l := number of instance types = {IT1, IT2,…,ITl}
- CCITj, where $1 \leq j \leq l$:= Compute Capacity of Instant Type l
- Pjk := Hourly price for leasing instant CCITj in cloud platform cpk

For a single cloud platform and a single instance type (m=1, l=1), the ***Total Infrastructure Capacity*** TIC and ***Total Infrastructure Price*** TIP of cpk are calculated:

$$\boldsymbol{Total\ Infrastructure\ Capacity} - \boldsymbol{TIC_k} = \sum_{i=1}^{n} CCIT_j$$

$$\textbf{\textit{Total Infrastructure Price}} - \textbf{\textit{TIP}}_{\boldsymbol{k}} = \sum_{i=1}^{n} P_j$$

Based on the TIC and TIP, the following resource allocation performance metrics are established:

$$\textbf{\textit{Capacity Saving Efficiency}}\ (\textbf{\textit{CSE}}) := \frac{Static\ TIC_k}{TIC_k}$$

and

$$\textbf{\textit{Overprovisioning Factor}}\ (\textbf{\textit{OPF}}) := \frac{TIC_k}{Ideal\ TIC_k}$$

The typical ranges for the CSE and the OPF are as follows:

1) the typical range of a cloud broker's CSE to be within [1(worst) ≤ CSE ≤ ideal CSE], where a value of 1 represents no capacity is saved at all / static provisioning and the ideal CSE represents maximal capacity saving
2) the typical range of a cloud broker's OPF to be within [1 (ideal) ≤ OPF ≤ ∞], where 1 denotes for no overprovisioning (ideal) and higher values denote for wasted capacities

The goal of the cloud broker is to maximize the CSE and minimize the OPF, while satisfying QoS constraints.

Specification of CSE/OPF values of a cloud brokering system alone, does not provide full insight into the performance of the system, since optimal capacity saving performance values are meaningless for as long QoS cannot be assured. Therefore the main capacity saving KPI needs always to be stated along with the QoS assurance performance KPI, explained in the next section.

9.3 QoS Assurance KPI Measures

For evaluating the QoS assurance performance, three different QoS performance parameters were taken into account: 1) network performance parameters, 2) service-specific quality parameters and 3) SLA-related service availability parameters.

For 1) ***network performance parameters*** packet loss, jitter and packet delay were measured and the results were evaluated against the ITU-T [40] standard for VoIP QoS classes, where the following QoS classes relevant for NGN services were identified to be relevant:

- *QoS Class 0* "*Real-time, jitter sensitive, high interaction (VoIP, VTC)*": IP packet Transfer Delay: <100ms, IP packet Delay Variation < 50ms, IP packet Loss Ratio < 0,001

- *QoS Class 1* "*Real-time, jitter sensitive, interactive (VoIP, VTC)*": IP packet Transfer Delay: <400ms, IP packet Delay Variation < 50ms, IP packet Loss Ratio < 0,001

For 2) ***NGN signaling performance*** and ***NGN service voice quality*** was measured and the results were evaluated using an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, the Perceptual Evaluation of Speech Quality (PESQ), as recommended by the ITU-T [130].

For 3) the results were evaluated against telecommunication industry standards of 99,999% ***carrier grade service availability*** and 99,99% ***high service availability***.

9.4 Test Environments

Validation and evaluations took place in a local, isolated testbed environment, the FUSECO Playground at Fraunhofer institute FOKUS, as well in a large-scale experimental facility, the Pan-European, multi-cloud facility of BonFIRE.

Whereas several experiments, studying the effects single, network and service related parameters were carried out in the isolated testbed environment (used to benchmark and profile network and services), the large-scale experimental facility was used to validate and evaluate the overall performance of the NGN Cloud Broker.

9.4.1 Isolated Testbed

The testbed setup within the laboratory of the FUSECO Playground, depicted in Figure 94, is comprised of an IMS setup, two OpenNebula-based Cloud setups, IMS load balancers, network performance emulation components and an IMS load generator / user-endpoint emulator. This setup allows testing of the NGN Cloud Broker's performance under different workload and network conditions. Additionally, the testbed allows service-specific benchmarking and profiling of workload vs. resource utilization, QoS vs. resource utilization, end-to-end QoS vs. resource utilization, network vs. QoS under controlled conditions.

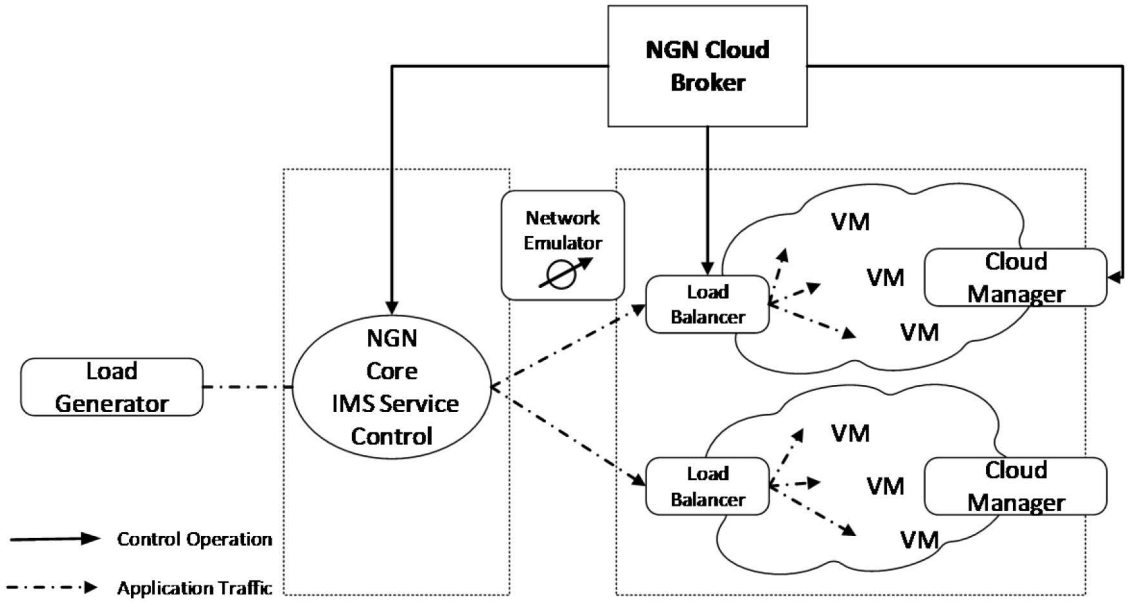

Figure 94: Testbed Setup – Workload Generation, Load Balancing and Cloud Management

9.4.2 Large Scale Multi-Cloud Test Facility

The evaluation of the NGN Cloud Broker in a real world, multi-cloud scenario, was carried out in pan-European BonFIRE multi-cloud facility. BonFIRE provides a unique testing facility for cloud-based services and systems. The facility is comprised of multiple cloud sites, located in different European countries. Shown in Figure 95, are BonFIRE's sites in Germany, France, UK and Poland.

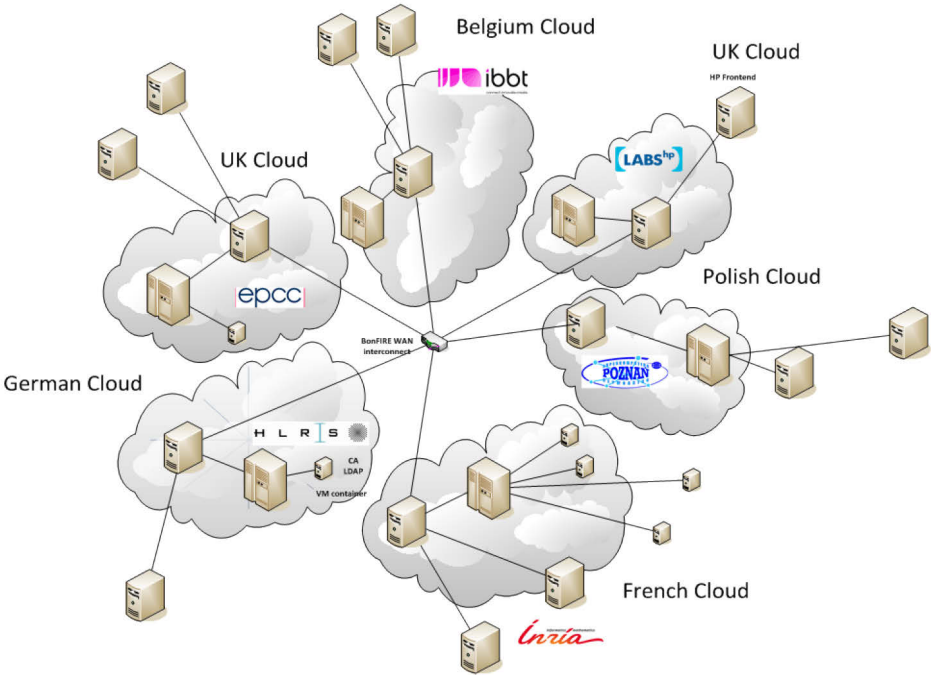

Figure 95: BonFIRE Pan-European Multi-Cloud Testing Facility

The BonFIRE setup consists of an NGN platform in Germany, comprising an IMS Core (including Proxy-, Interrogating-and Serving Call State Control Functions P-/I-/S-CSCF and a Home Subscriber Server HSS) and Kamailio-based IMS/SIP load balancers and NGN application servers deployed on, German (HLRS), French (INRIA) and the UK (EPCC) clouds. IMS Bench SIPp is used as ETSI-compliant SIP/IMS load generator, generating constant load and workload variations. For avoiding NAT traversal problems an RTP-proxy integrated into the P-CSCF node is utilized. Figure 96 shows a simplification of the test setup, depicting the control signaling of the NGN Cloud Broker, the IMS/SIP signaling as well as the media path (RTP).

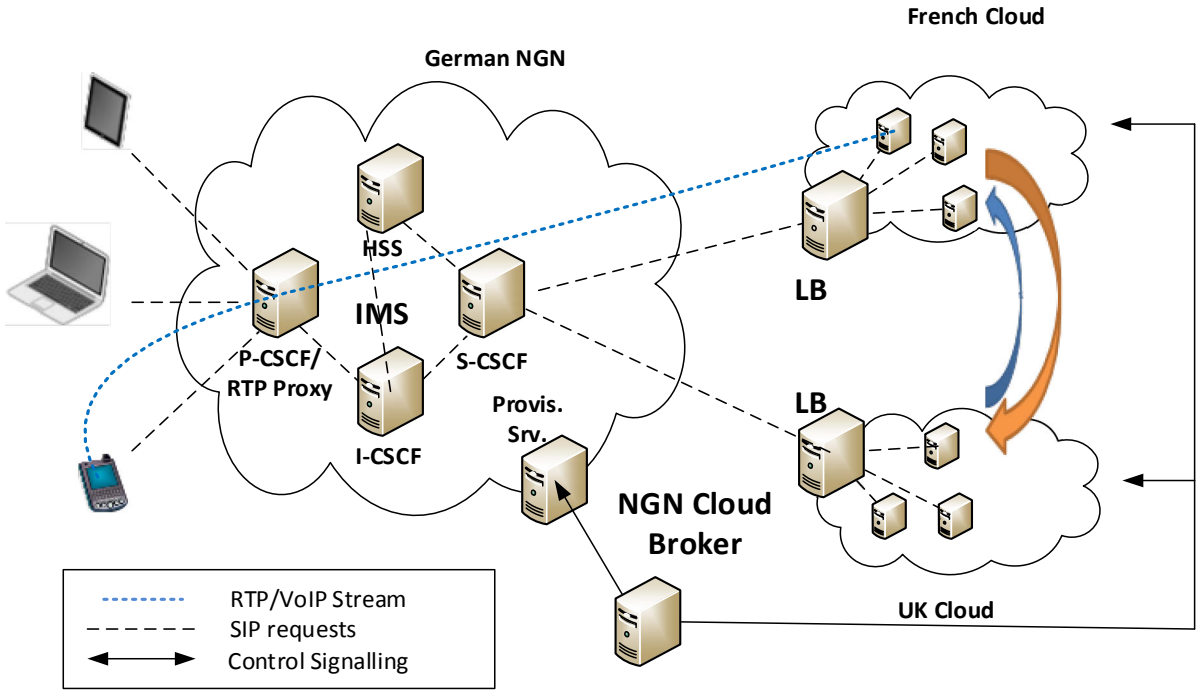

Figure 96: Large Scale Multi Cloud Test Setup on BonFIRE, cmp. [22]

9.5 Resource, Service and Workload Profiling

One part of the benchmarking conducted aimed at assuring that involved components, particularly the IMS, cloud managers and the load balancers do not impose limitations / performance limits to or affect the overall cloud brokering system's performance. By doing so, it was assured that within the range of workload and signaling involved in the evaluation, neither the OpenIMScore, nor the Kamailio load-balancers, nor the OpenNebula cloud management system represented sources of error/imposed effects of overload.

The other part of benchmarking and profiling, however, was conducted in order to configure the NGN Cloud Broker appropriately. First and foremost, a clear understanding of the capacity of a particular cloud instance (i.e. small, medium, large instance of cloud X) was needed. Second, the actual NGN service's QoS needed to be benchmarked under different

workload and network conditions. Third, the realistic workload was analyzed in order to assess the boundaries within which, capacity saving can be realized through the NGN Cloud Broker, as well as forecasting mechanisms can be applied.

Workload Profile

For testing with realistic workload, a call data record from an early Spanish IMS/VoIP service provider was utilized and the data record was scaled by a factor of 4.5 to match with the capacity of a mid-size incumbent NGN operator (2.5 Mio calls per day). Figure 97 shows the daily distribution of initiated VoIP calls per minute. Based on this workload, artificial workload was generated using the SIPp [25] load generator.

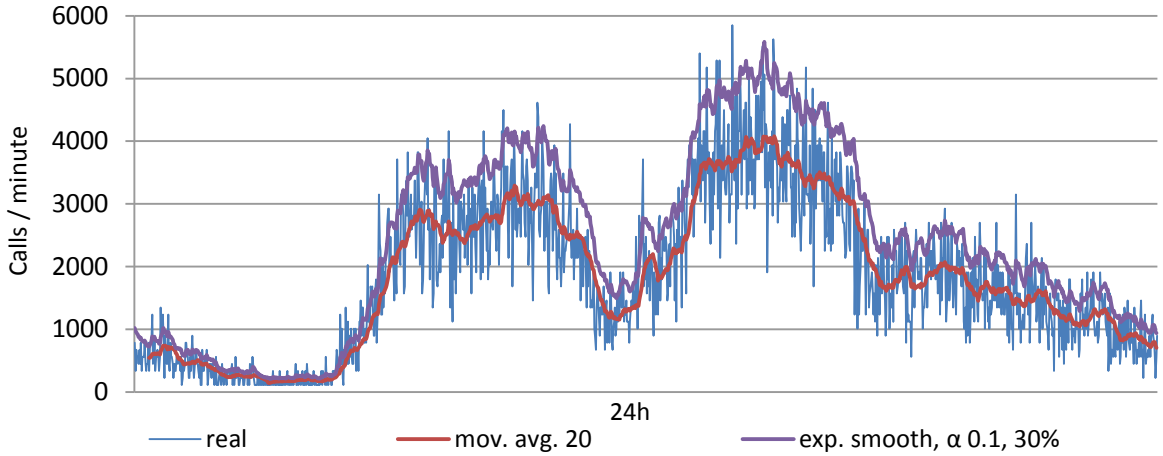

Figure 97: Realistic IMS/VoIP Workload, mov. avg. and exp. smoothing filters applied, publ. in [22]

Several estimators, and filters, including power weighted moving average and exponential smoothing filters are applied for workload prediction, signal smoothing and system calibration. The main purpose of finding an appropriate filtering and prediction mechanism is to find an optimal balance, of overprovisioning of resources (required for accommodating for the NGN Broker's overall inertia) and capacity saving (without encountering overload-based QoS degradations).

Based on the workload profile, and based on capacity profiles of selected cloud instances, ideal (fully elastic) and statically overprovisioned capacities ($TIC_{ideal}/TIC_{max)}$ are calculated subsequently. The workload profile shown in Figure 97, has a maximum of 5850 calls per minute (0 calls per minute minimum), 1844 average calls per minute, standard deviation of 1229. Compared to the filtered workload time series, the moving average and exponential smoothing expose significantly lower volatility, as shown in Table 14.

Table 14 Workload Regression Analysis, real, exponential smoothing, moving average

Regression Statistics			
	real	***exp***	***ma***
Multiple R	0,789781	0,396263	0,342367
R Square	0,623754	0,157024	0,117215
Adjusted R Square	0,623012	0,156398	0,11656

Standard Error	1360,513	1025,145	1155,728
ANNOVA F	2233	250	178

Workload versus QoS Profile

Initial benchmarking of the selected, NGN service, a SIP media-server SEMS [137] a call answering services, was conducted in FUSECO Playground laboratory [19]. Shown in Figure 98, over a period of 342 seconds, by generating an increasing number of calls per second (requests/s), the impact of workload and CPU utilization on the resulting voice quality (PESQ) was analyzed. For the small cloud instances used in this test, a maximum of 77% CPU utilization was measured, corresponding to 23 calls per second above which the PESQ significantly deteriorates.

The PESQ is measured by comparing the original voice quality with the quality received at the IMS/NGN fixed/mobile end-device. The overall PESQ significantly depends on the utilized Codec (G.711 was used). PESQ values well above 4 can be reached (5 being the maximum value) with high bandwidth/quality Codecs. Using narrowband codecs (such as the G.711 codec), however, PESQ values better than 3 could not be achieved, even without any inferences. PESQ values above 2 were subjectively perceived to be still acceptable (sufficiently clear voice quality).

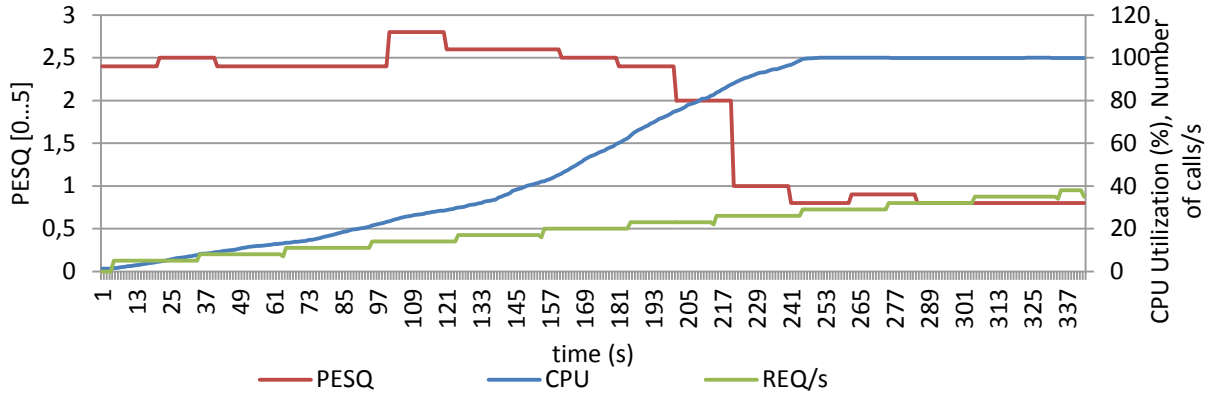

Figure 98: NGN QoS (Voice quality) vs. Resource Utilization vs. workload, isolated cloud testbed [19]

Subsequently, the NGN service was deployed on the multi-cloud testbed of BonFIRE where three cloud platforms were selected [22]. Within each platform the instance type “small” was selected, i.e. a single core virtual machine, with 1 GB Memory (used physical hardware for the CPUs are AMD Opteron, 2,3 GHz at EPCC/UK, Intel XEON 2,3 GHz at INRIA/France, Intel XEON 2,6 GHz at HLRS/Germany). Different workload levels are generated in a step-wise fashion from 0 to 300 requests / calls per second. By doing so it was possible to identify the performance of each particular cloud platform’s small instance involved as shown in Figure 99. Superior physical hardware, obviously, results not necessarily in better workload robustness against QoS deteriorations. The conducted measurements show that with increasing workload resulting PESQ levels differ by a factor

greater than 2. This has *significant impact on required capacity planning* and would result in a significant resource-cost difference (if similar costs would be attached to the same instance types).

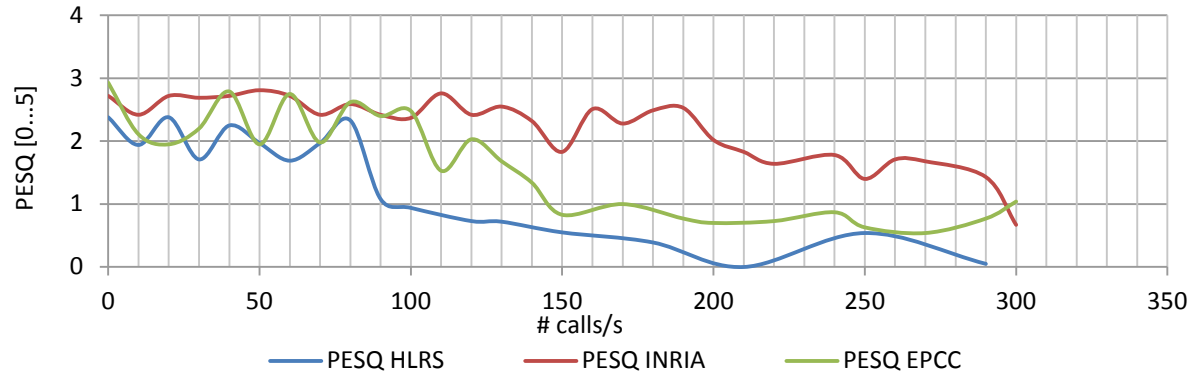

Figure 99: Workload vs. QoS (Voice Quality), small inst. types, German, French, UK Cloud, publ. in [22]

Of particular relevance for assessing NGN service quality, not only is the quality of the media part, but also the quality of the IMS/SIP signaling part. In fact, for many NGN services, where media does not play a role, such as messaging or exchange of presence information, the signaling quality / performance the important, service quality determining factor. Therefore, the SIP signaling performance of the three NGN service instances residing within the three different cloud platforms was determined. The conducted measurements reveal that, whereas with increasing workload, the three NGN service deployments expose significant differences in voice quality deterioration, they only *marginally differ in call attempt / SIP signaling performance* as shown in Figure 100. This significant difference underlines the importance of service-specific QoS versus workload benchmarking.

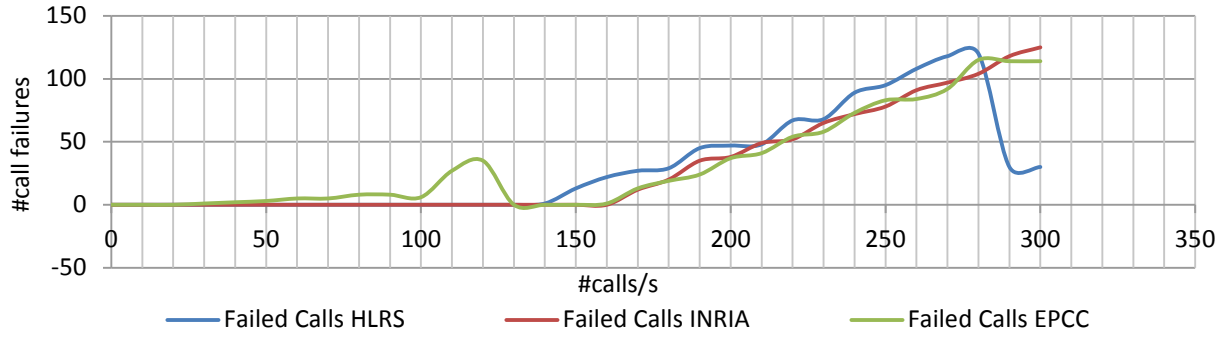

Figure 100: Workload vs. QoS (Call Failures), publ. in [22]

Workload versus Capacity consumption Profile

The corresponding workload versus capacity consumption profile for the three small instances of the three cloud platforms running the SEMS IMS/NGN media server is shown in Figure 101. Comparing the French/INRIA and the German/HLRS profile of CPU consumption versus capacity/resource consumption, with the corresponding PESQ measurements (shown

in Figure 99), in the case of the German/HLRS cloud, CPU utilizations of above 70% (~30% CPU idle time) already show a significant, unacceptable negative impact on the resulting PESQ. Apparently the French/INRIA cloud instance significantly outperforms the other instances, particularly the German/HLRS instance, where CPU utilization of above 70% occurs at much higher (~ factor 1.7) workloads, with corresponding QoS/PESQ deteriorations happening at even higher workloads. These measurements reveal that 1) CPU load not equally correlates with QoS degradations across different cloud instances and 2) significant performance differences exist even between the same types (small in the measurements shown) with similar underlying hardware configurations.

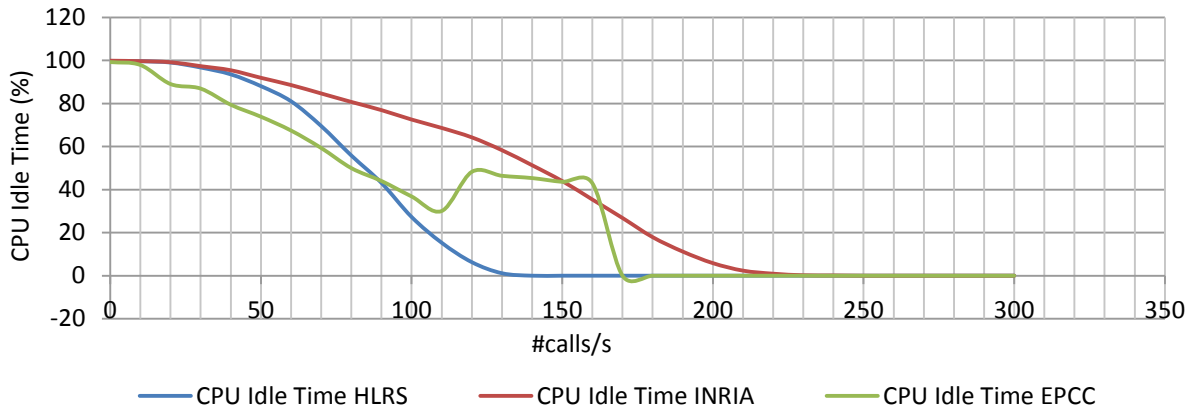

Figure 101: Workload vs. Idle CPU Time (= 100% - CPU Utilization), publ. in [22]

Network performance versus QoS Profile

Resource overload as shown in the measurements above are not the only factor resulting in QoS degradations. Network performance parameters such as Jitter and Packet Loss of the network link between the NGN and the interworking cloud-based services need to be equally taken into account as the resulting end-to-end QoS might be impacted by both factors.

Measurement correlating network performance parameters with the resulting QoS were carried out in the isolated, local testbed described in section 9.4.1. Shown in Figure 102 is the impact of Jitter on the QoS of the cloud-based NGN service. The conducted measurements show that Jitter levels of 30-50 milliseconds in some cases already led to a QoS/PESQ level degradation of below 2 (i.e. unacceptable voice quality).

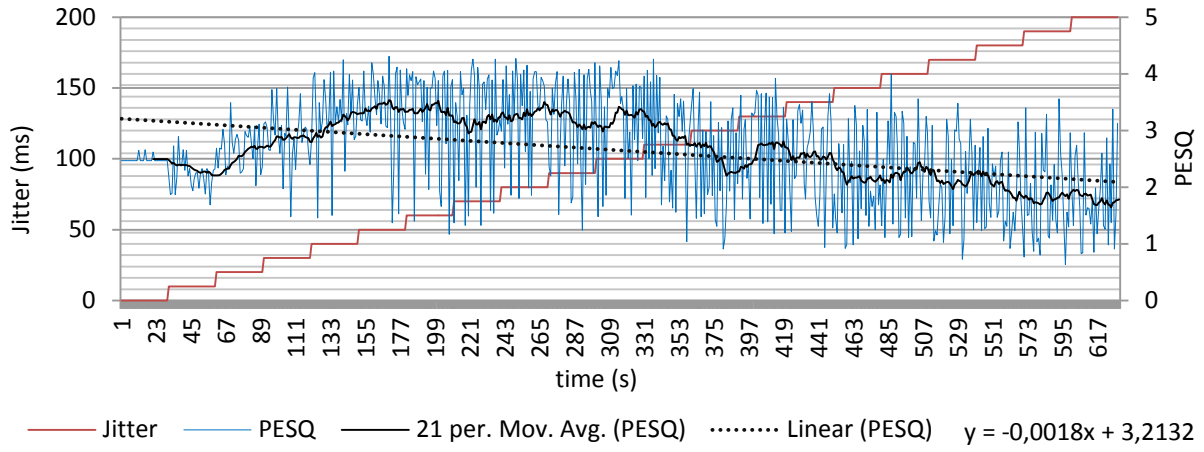

Figure 102: Impact of Jitter on PESQ (analysis bsd. on [19], publ. in [22])

Similarly, the impact of packet loss on the end-to-end QoS/PESQ level was measured as depicted in Figure 103, showing that unacceptable PESQ/voice quality (PESQ below 2) might already result from packet loss values above 3%.

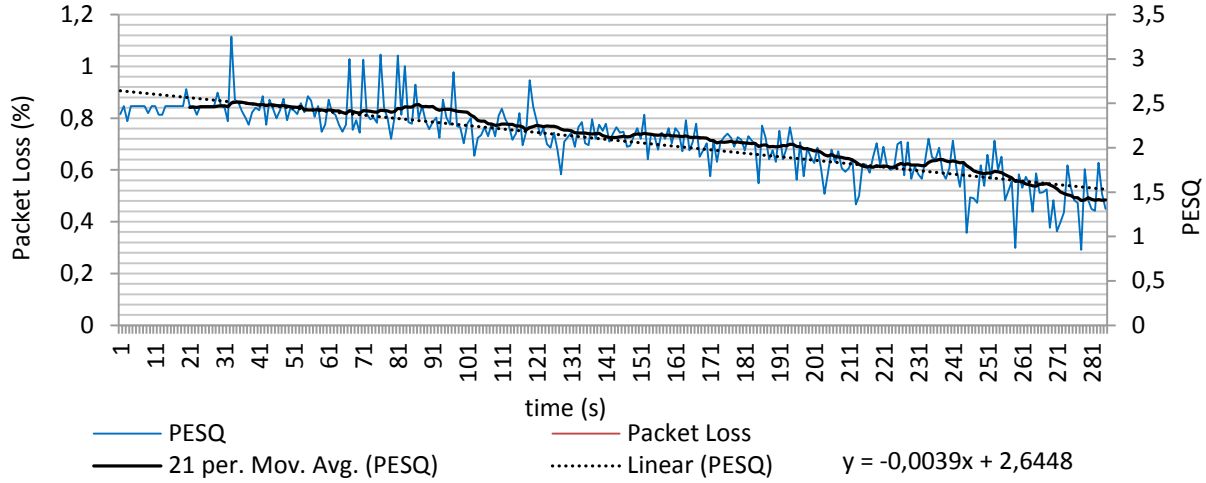

Figure 103: Impact of Packet Loss on QoS (analysis bsd. on [19]), publ. in [22]

For analyzing the likelihood of such network-performance induced degradations of the QoS of cloud-based NGN services, network performance measurements between the NGN in Germany and the involved cloud platforms were conducted. Shown in Figure 104 are Jitter measurements between the NGN and the French/INRIA and UK/EPCC cloud platforms that were hosting the NGN service. For most the time during the 17 hours and 30 minutes of the conducted measurements, the Jitter levels between the NGN in Germany and each cloud site hosting the media/announcement service kept at negligible levels (not having significant impact on the QoS/PESQ). However, during 30 minutes a significant increase of Jitter of up to 25ms occurred between the German NGN and the UK cloud platform, which impacted the finally perceived QoS/PESQ. These findings are well corresponding to the ITU-T recommendation on “Network performance objectives for IP-based services” [40] stating that Jitter values of high quality (QoS class 0 and 1) VoIP services need to be below 50ms.

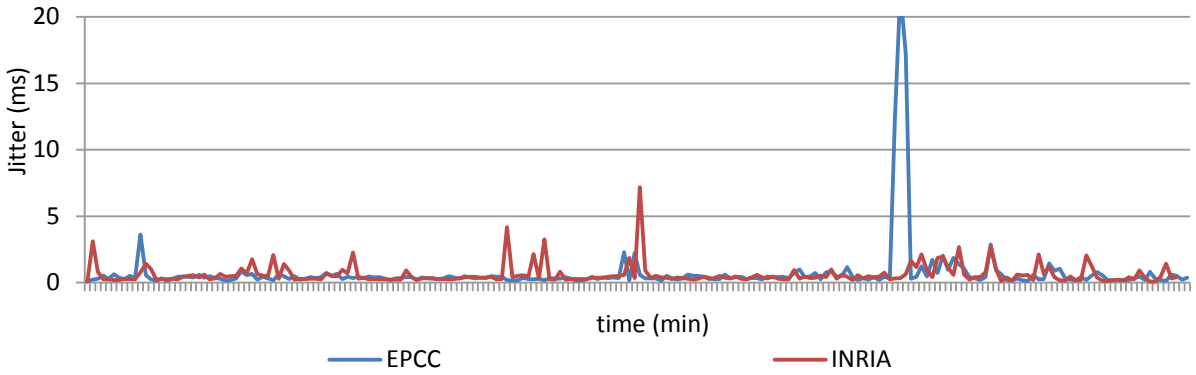

Figure 104: Multi-Cloud Network Performance Measurements publ. in [19]

The conducted measurements revealed that indeed, for providing QoS-assured NGN services in federated cloud environments, network performances need to be taken into account and that QoS degradations resulting from poor network performances are not unlikely. Only if both, platform/resource specific as well as network specific factors are well monitored and mechanisms for scaling and migration are in place, can those effects be compensated.

Summary

In summary, above findings show that

1) there are *significant differences between cloud instance capacities* even of the same type and similar underlying physical hardware, for cost-efficient selection,
2) the capacity of a cloud instance *needs to be determined in a service-specific way* as there are significant differences in capacity limits depending on the type of service,
3) resource utilization only roughly correlates with QoS deterioration due to resource overload and therefore *needs to be determined for each cloud instance and each particular service*,
4) *network performance represents a significant factor* impacting the finally perceived end-to-end NGN QoS and therefore needs to be taken into account while selecting a particular cloud platform as well as during service operations.

A detailed discussion of the impact of the above described findings, which also includes the system-specific impact of the Cloud Broker (section 9.7) is provided in section 9.10.

9.6 Capacity Saving and QoS Assurance Performance Benchmarks

Findings made in the course of above described benchmarking results clearly show that on the one hand, a *generic* metric (one and for all services), like the Amazon Elastic Compute Unit

(ECU, Amazon's metric for specifying the capacities of their compute instances) is not sufficient for determining the capacity of a cloud instance for providing any type of services, and needs to be specified for each particular service / hosted application in order to become meaningful. On the other hand, the conducted benchmarking measurements show that maximum capacities of a particular cloud instance needs to be mapped to the maximum workload (not the resource utilization) at which a particular service can be still be provided at sufficient QoS levels.

By analyzing profiles and benchmarks of section 9.5, it was possible to determine the maximum capacities of the three cloud platform's small instances, as summarized in Table 15. Obviously, a small instance of the French/INRIA cloud platform is capable of serving up to twice as many calls per second as the compared small instance of the German/HLRS cloud platform. Also summarized in Table 15 are the encounter delays of provisioning small cloud resource instances per cloud platform. Obviously, involved cloud platforms exposed significant differences (up to factor 4) in cloud provisioning delays ranging from 50 seconds to up to 200 seconds. Finally, as an indicative measure, it was assumed that each of the small instances of each involved cloud platform would be priced in the same way as the current Amazon small instances (at the point of writing 0.065$ per hour).

Table 15 Instance Capacities for NGN service (SEMS), platform provisioning delays, publ. in [22]

cp_k	**Single Small Instance Capacity, Hourly Price**
	$CCIT_i, P_{ik}$
INRIA/French Cloud	160 calls/s, 0.065$/h, 50-65s provisioning delay
HLRS/ German Cloud	82 calls/s, 0.065$/h, 60-200s provisioning delay
EPCC/UK Cloud	105 calls/s, 0.065$/h, 100-120s provisioning delay

Based on the determined NGN service-specific capacities of small instances of each involved cloud platform, the worst-case, statically full provisioned $\boldsymbol{TIC_k}$ and $\boldsymbol{TIP_k}$ as well as the ideal, fully elastic $\boldsymbol{TIC_k}$ and $\boldsymbol{TIP_k}$ (minute granularity) as summarized in Table 16 is calculated. Whereas for serving a maximum of 5850 calls/s of the workload shown in Figure 97 a fixed capacity of 864 Virtual Machine hours (VMh) would be needed if the French Cloud's small instances would be used, the German Cloud would require 1728 VMh, and the UK Cloud 1344 VMh respectively. For the same daily workload, the ideal capacity $\boldsymbol{TIC_k}$ is calculated, based on the assumption that an ideal, fully elastic scaling mechanism would be able to dynamically and rapidly provide the exact amount of capacities (limited by the size of VM instances only) within each given minute of the day.

Table 16 Worst Case/Static and Ideal TICs and TIPs for daily, NGN workload, publ. in [22]

cp_k	**Required Capacity / Costs for daily NGN service Workload**	
	Static $\boldsymbol{TIC_k}$ **,** $\boldsymbol{TIP_k}$	***Ideal Elastic*** $\boldsymbol{TIC_k}$ **,** $\boldsymbol{TIP_k}$
INRIA/French Cloud	864 VMh, (~56$/day)	270 VMh, (~18$/day)
HLRS/ German Cloud	1728 VMh, (~112$/day)	517 VMh, (~34$/day)
EPCC/UK Cloud	1344 VMh, (~87$/day)	406 VMh, (~26$/day)

The calculated ideal and worst-case total infrastructure capacities can subsequently be mapped to the ideal and worst-case workload-distribution-dependent (fully static workloads do not allow for any CSE improvements) CSEs. For the NGN workload shown in Figure 97, an *optimal/ideal CSE of 3,2* is determined, i.e. *69% capacity saving*. The corresponding OPF of the worst-case, static capacity compared to the ideal elastic is 3,2.

The performance of the NGN Cloud Broker's Resource Allocation mechanisms is determined by its Capacity Saving Efficiency ranging from 1 (no efficiency) to 3,2 (ideal efficiency in the case of the selected, realistic NGN workload shown in Figure 97).

Note: Commercial cloud offerings, in many cases impose a *minimal lease time of cloud resources* (usually between 10 minutes, 1 hour as with Amazon or even a full day). It should be noted that this limitation represents a critical factor for achieving optimal CSE. For the NGN workload shown in Figure 97, a minimal lease time of 1 hour, would correspond to a maximum achievable optimal CSE of 1,92 (i.e. a maximum of 48% capacity saving), corresponding to minimal OPF of 1,7, solely caused by the cloud provider's pricing policies. For the scope of this evaluation, it was assumed that no limitations regarding minimal lease times are imposed.

9.7 Performance Evaluation

The performance evaluation of the NGN Cloud Broker consists of an evaluation of 1) the evaluation of the NGN Cloud Broker's Platform and Instance Selection Performance and 2) the evaluation of the NGN Cloud Broker's Resource Allocation Performance.

Clearly the emphasis of this work and particularly the emphasis of the performance evaluation is on 2) capacity saving and QoS assurance capabilities of the NGN Cloud Broker's Resource Allocation mechanisms. Compared to the resource allocation mechanisms, the Cloud Broker's platform selection mechanisms are rather trivial. Therefore, validation of the selection process is only briefly provided and subsequently, full *focus is put on the resource allocation performance evaluation*.

NGN Cloud Broker – Platform and Instance Selection Validation

The evaluation of the NGN Cloud Broker's Platform Selection Performance was carried out in the BonFIRE multi-site cloud facility. In terms of selection criteria, the space of evaluated parameters is limited to 1) resource costs and 2) network performance, as these are the core factors for 1) saving costs and 2) assuring QoS. Surely, the selection process could encompass several more factors (such as security, trustworthiness, location of a cloud platform).

The validation was carried out utilizing the multi-cloud facility of BonFIRE, particularly the Cloud platforms in the UK as well as in France, together with the NGN/IMS testbed setup

as described 9.4.2. By applying the NGN Cloud Broker's platform selection mechanisms as described in AII.3.5.1, the NGN Cloud Broker dynamically selected the current best cloud platform, based on perpetual evaluation of resource cost and network performance.

For doing so cloud instance price variations (based on current commercial pricings) were simulated on an hourly basis as shown in Figure 105 for the duration of the experiment and packet loss and jitter network performance parameters were measured, as shown in Figure 104.

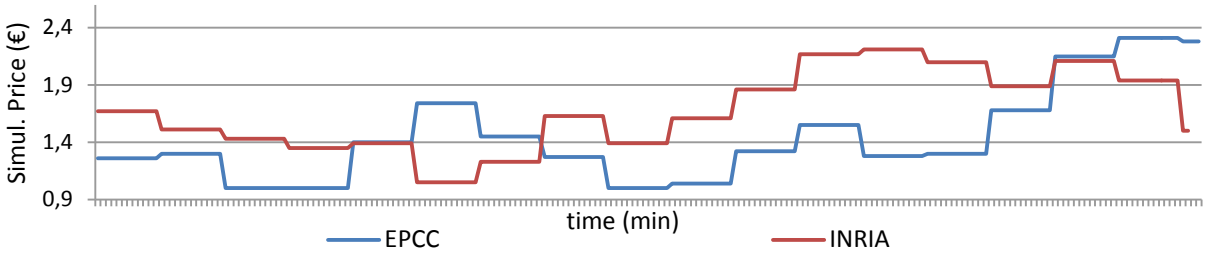

Figure 105: Simulated Cloud Instances Prices of UK and French Cloud Platforms, 17,5 hours [19]

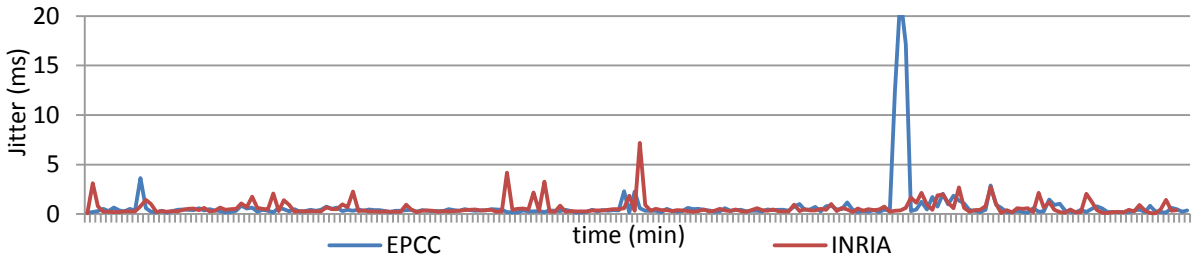

Figure 106: Network Performance of UK and French Cloud Platform, 17,5 hours [19]

During the duration of the 17,5 hour experiment, the NGN Cloud Broker evaluated current resource prices and network performance parameters and computed a score for each platform as shown in Figure 107.

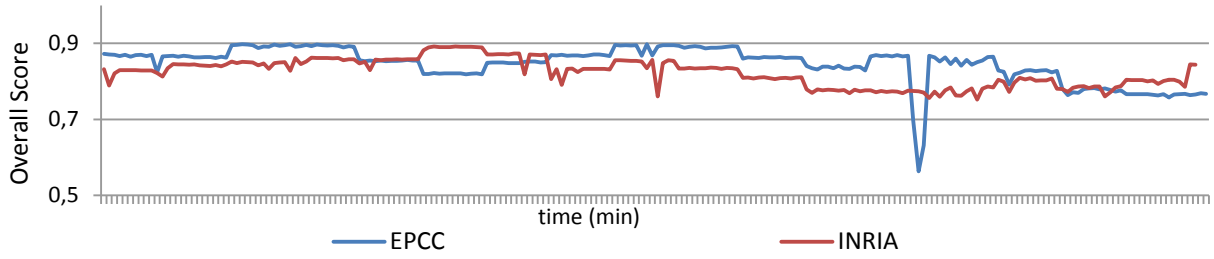

Figure 107: Platform Scoring, 17,5 hours [19]

During the course of the experiment, the NGN Cloud Broker selected the cloud platform with the current best score and activated the NGN service of the particular platform, as shown in Figure 108.

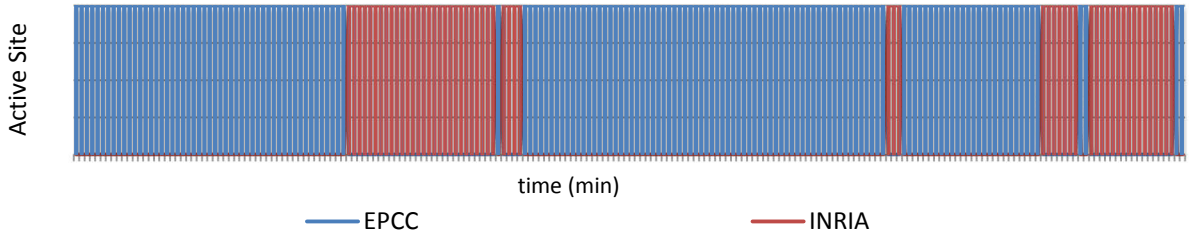

Figure 108: Cloud Platform Selection,17,5 hours [19]

By incorporating a short (10 seconds) grace period (the duration of the announcements played be the cloud-based media server was 8 seconds), it was assured that during the transition / migration period, currently served sessions were not aborted. Thereby continuous availability of the service without service failures was assured.

The evaluation of the cloud platform selection performance shows that indeed, service quality can be assured, by dynamically selecting and utilizing alternative cloud platforms, in case of cloud platform / associated network performance deteriorates. The proof of concept also shows that indeed cloud selection mechanisms can serve to optimize costs, by dynamically selecting and utilizing current best offerings.

The results show, that for the duration of the experiment, the cost factor was predominantly influencing the platform selecting process. Only during 30 minutes of significant network performance degradation, i.e. significant increase of Jitter of the network connection between the German NGN and the cloud platform in the UK (hosting the NGN service), network performance parameters were responsible for triggering platform re-selection / migration.

Discussion: There are several factors, which were simplified or not sufficiently taken into account for the evaluation. First, focus was mainly put on the actual platform selection and activation processes, thus *cold service migration* mechanisms were not conducted, as cloud instances were already in operation at both sites (UK and France). On the one hand, this allowed for rapid switching between both sites and quick activation of services, simply by administering / provisioning new service endpoints at the NGN service control layer (IMS), which took less than 50 milliseconds. Doing so, however, would only be an appropriate measure for achieving high availability, while neglecting costs. A more realistic scenario, where costs do matter, would require cold migration, i.e. full initial deployment and activation of cloud services, which, for the focused NGN service could take up to 100 seconds as measured in [21] (depending on the particular cloud platform management / instance provisioning performance, as well as the particular network performance sometimes even higher durations can be expected, see Table 15). Second, the selection process does not take into account the *costs of migration*. As already stated in section 9.5, cloud platform providers usually impose a minimal lease time for leasing cloud resources. Therefore, before freely switching between cloud platforms, based on minor price differences, both factors should be taken into account for optimization of the overall platform selection performance. In summary however, aforementioned constraints, can easily be described in terms of policies, and made available for trimming the NGN Cloud Broker's platform selection processes. As such, the

general cost-/QoS-optimal cloud platform selection mechanism of the NGN Cloud Broker is a straight-forward filtering and weighting process, which can readily be applied to real-world scenarios. Whereas indeed, optimal platform selection also allows for cost as well as QoS optimization (depending on current offerings of the IaaS market), this work and particularly this evaluation rather focuses on the cost savings and QoS assurance mechanisms enabled by elastic cloud resource allocation mechanisms, evaluated in more depth in the subsequent chapters.

NGN Cloud Broker – Resource Allocation Performance Evaluation

The evaluation of the NGN Cloud Broker's Resource Allocation Performance was carried out using small VM instance types, first locally in the isolated cloud testbed, described in section 9.4.1, using artificial workloads and subsequently in the BonFIRE multi-site cloud testbed, using the realistic NGN workload as introduced in section 9.5.

For the evaluation of the NGN Cloud Broker in the isolated cloud testbed, in-/ and de-creasing ramps of IMS calls per second were generated, with a step size of 5 calls per second, with a duration of 50 seconds for each step and a maximum of 270 calls per second as shown in Figure 109. The maximum CPU utilization threshold is set to 70% as shown in Figure 110, reaching a maximum of 265 handled calls per second using 9 load-balanced VMs at the peak.

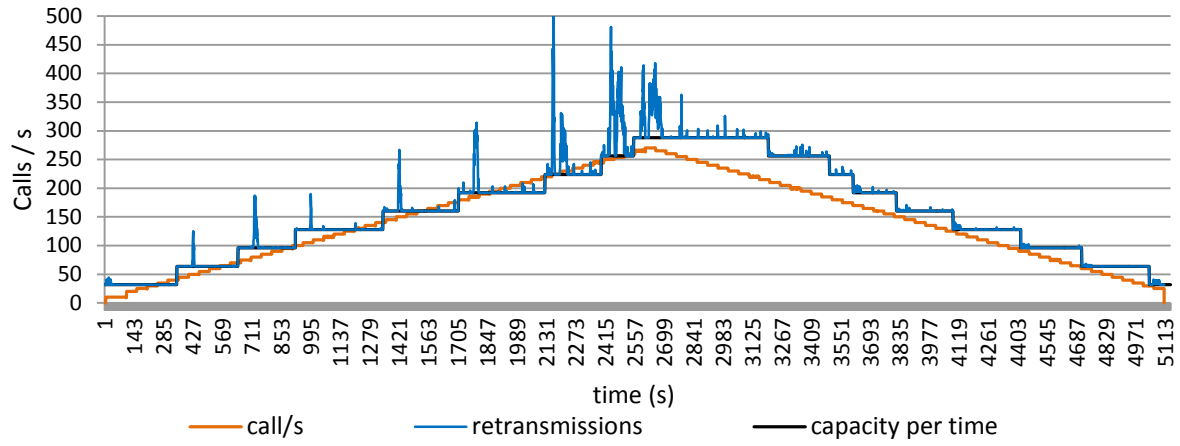

Figure 109: Up- and Down-Scaling - Calls/s vs. #VMs vs. Retransmissions, publ. in [22]

In this scenario the Cloud Broker's Resource Allocation performance parameters are: **Capacity Saving Efficiency = 1,80; Overprovisioning Factor = 1,12**

Although an OPF of 1,12 (12% of over-provisioned resources) was realized in this scenario (which is very close to an ideal, elastic scaling performance of an OPF of 1), retransmissions shown in Figure 109, indicate that elastic resource allocation in some cases was conducted too late in time. As shown, especially during the transmission phases, the overall system capacity was insufficient (under-provisioned resource capacities) for handling the amount of workload. This led to times of CPU overload (also shown in Figure 110), causing undesired retransmissions as shown in Figure 109.

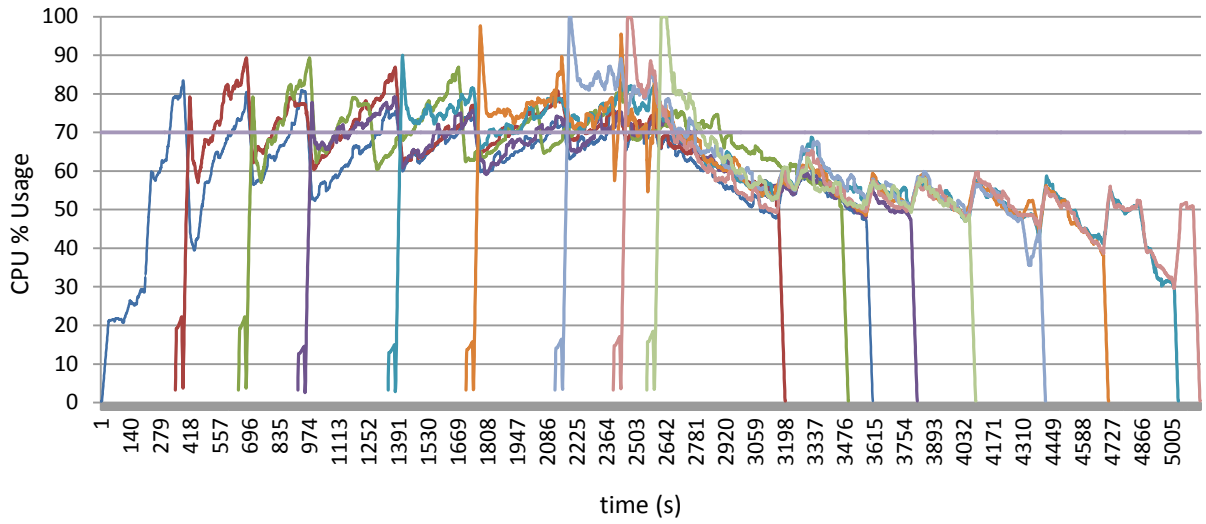

Figure 110: Up- and Down-Scaling - Single Node's CPU util., isolated testbed, workload ramp [19]

Although in very few cases retransmissions led to complete call failures, acceptable SIP signalling performance alone, as explained in section 9.5, is only one part of the NGN service quality. During each period, where CPU utilization exceeds levels of 80%, PESQ / voice quality levels become unacceptable as shown in Figure 98. In this context, the NGN Cloud Broker's inertia, i.e. the delay between reaching a threshold to the time of updated resource allocations becomes relevant.

Also observable in Figure 110, the NGN Cloud Broker's inertia (mainly determined by the time required to instantiate new cloud instances) amounts to 70-90 seconds. This is the minimal look-ahead step for any workload forecasting mechanism to be applied, however varying from cloud platform to cloud platform, as measurements in Table 16 show. Alternatively, of course, lower CPU utilization threshold could be chosen for addressing the inertia, which would directly lead to an increase of capacity overprovisioning (increased OPF, decreased CSE).

Figure 110 also reveals that individual CPU levels were not always aligned with each other, a clear sign of sub-optimal load-balancing (some machines were exposed to higher load than others, although identical VMs were used). It seems that especially during these periods, re-transmissions were encountered, which is quite natural since the NGN Cloud Broker's scaling algorithm was sensitive to the overall system load (average load of all VMs) and not sensitive to a particular VM's CPU utilization. Utilization of weighted, round-robin load-balancing mechanisms, which are dynamically adjusting the forwarded load based on the feedback (current CPU load) received by each participating node/VM could help resolving this effect.

Further analysis revealed that during elastic scaling of resources a single VM's capacity for serving requests increased with growing numbers of participating instances. As shown in Figure 111 with every additional instance, the average number of calls handled by a single instance increased from an average of 27 requests per machine (while only one machine was

participating) to an average of 32 handled requests per machine (during times where 6 machines participated). A logarithmic approximation for modeling this effect was chosen since with growing numbers of nodes, a saturation of the effect was identified. As an explanation for this effect, it was reckoned, that with each additional node, the average utilization of the overall system increases. This leads to an overall system mode of operation, which (with each additional participating node) more closely approximates the pre-defined threshold of maximum, system-wide resource utilization, leveling out under-utilized capacities in favor of fully saturated utilization of resources.

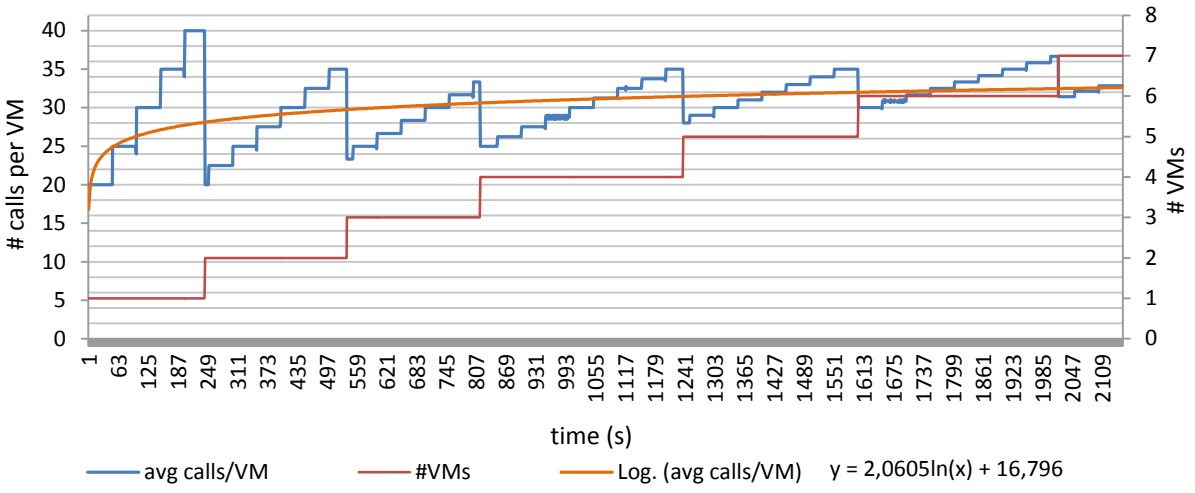

Figure 111: Impact of Scaling on VM Capacity (avg Calls handled per VM), publ. in [22]

NGN Cloud Broker Resource Allocation Performance in Federated Cloud Environments

Using the realistic daily NGN workload (shown in Figure 97) and by using small instances of the French Cloud / INRIA, which performed best in terms of performance (summarized in Table 15, shown in Figure 101) and robustness of QoS levels (see Figure 99), the NGN Cloud Broker was tested on the multi-cloud environment of BonFIRE, with the NGN deployed in Germany. Testing PESQ / QoS assurance performance with different CPU utilization thresholds revealed that the CPU threshold needed to be lowered (first to 50%, then increased to 60%) in order not to encounter PESQ failures. Due to constraints in capacity the maximum workload had to be decreased from 5850 to 1170 max calls / second, thus the workload distributions were downscaled by a factor of 5. Results are shown in Figure 112.

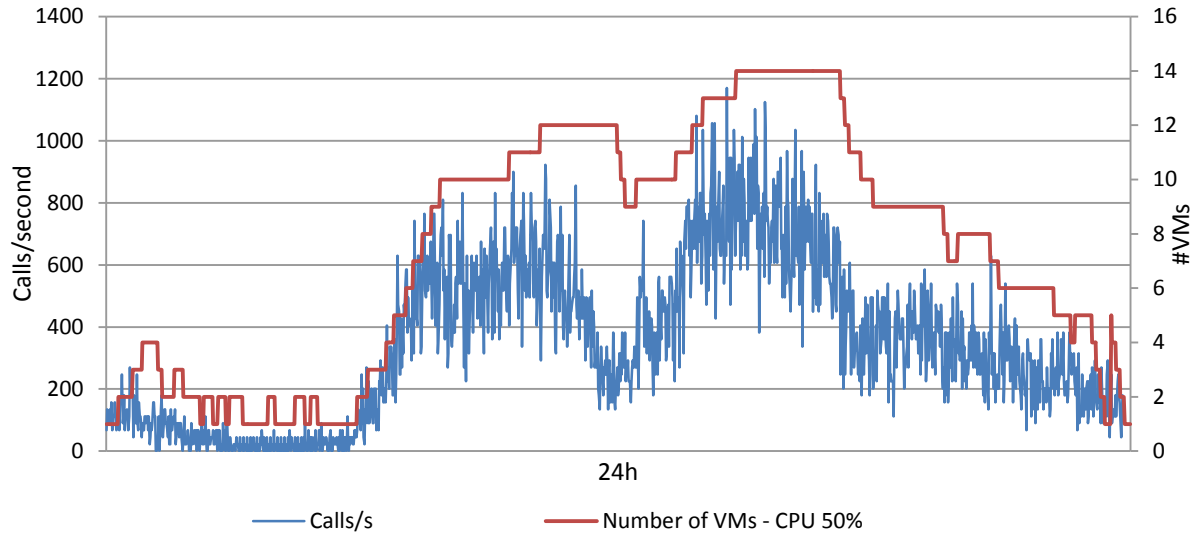

Figure 112: Full Day Auto-Scaling, 50% CPU threshold, Resource Consumption, publ. in [22]

The capacities that are required for serving the realistic NGN intra-day workload are shown in Figure 112. The resulting CSE is 1,88 and the OPF is 1,8. The Capacity Saving Performance of the NGN Cloud under real workload conditions and within a real multi-cloud environment is lower than the resource allocation / capacity saving performance in the isolated testbed under artificial load.

As shown in Figure 113, the attained Voice Quality of the NGN Service only in a view cases, dropped below the “poor quality” threshold of PESQ values below 2. With an average of a PESQ value of 2,58 and 0,7% of PESQ values below 2.

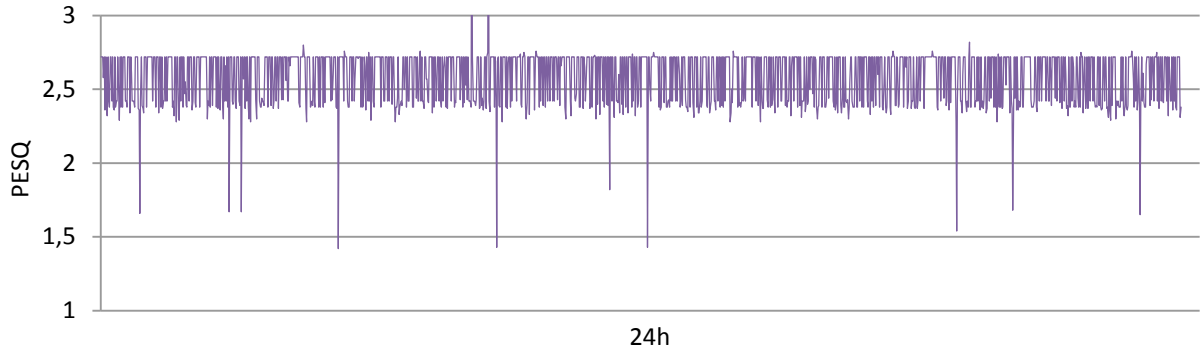

Figure 113: Full Day Auto-Scaling, 50% CPU threshold, E2E NGN Service Voice Quality (PESQ) [22]

As shown in Figure 114, also with 60% CPU threshold the attained Voice Quality of the NGN Service only in a view cases, dropped below the “poor quality” threshold of PESQ values below 2, whereas a CPU threshold of 80% led to significant, unacceptable voice quality deteriorations, throughout the entire duration of the test.

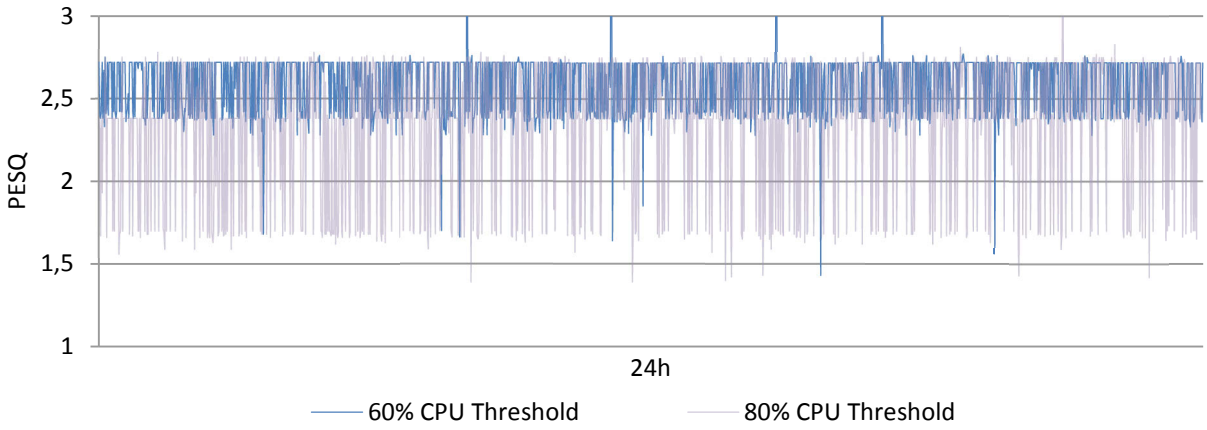

Figure 114: Full Day Auto-Scaling, 60% and 80% CPU threshold, E2E NGN Service Voice Quality (PESQ)

9.8 Evaluation of KPIs

The Key Performance Evaluation Criteria, defined in section 3.4, for convenience are recapitulated in Table 17.

Table 17 Performance Evaluation Criteria

Performance Evaluation Criteria	**Description**
Capacity Saving Efficiency / *Resource Allocation Performance*	The Cloud Brokering system should enable *High Capacity Saving Efficiency*, low Overprovisioning – High Cost/Energy Saving, compared to non-elastic and ideal capacity saving / usage (as well as compared to commercial systems)
QoS Assurance Performance	The Cloud Brokering system should enable *High QoS Assurance Performance Levels* – matching standardized QoS service classes (ITU-T) and common SLAs (service availability) and NGN service / voice quality levels

Evaluation of *Resource Allocation Performance*

Being the core performance evaluation criteria of this work, the resource allocation performance needs to be evaluated in depth.

In order to come up with metrics for quantification of the resource allocation performance of the cloud brokering system, the following metrics, as defined in section 9.2 used:

$$\boldsymbol{Capacity\ Saving\ Efficiency\ (CSE)} := \frac{Static\ TIC_k}{TIC_k}$$

and

$$\boldsymbol{Overprovisioning\ Factor\ (OPF)} := \frac{TIC_k}{Ideal\ TIC_k}$$

Both metrics *depend on the actual workload distribution*. Without workload variations, the ideal total infrastructure capacity would be identical to the static total infrastructure capacity. Elastic resource allocation would not be needed, static capacity allocation would be sufficient and although the cloud broker's overprovisioning factor would be close to 1 (perfect), no capacity / costs could be saved.

The NGN Cloud Broker resource allocation performance as evaluated in the two scenarios, 1) the artificial workload (Figure 109) generated for the tests in the isolated, local cloud environment and 2) the realisitc workload (Figure 112) generated for the tests in the multi-cloud environment of BonFIRE (as described in section 9.7) is depicted in Figure 115.

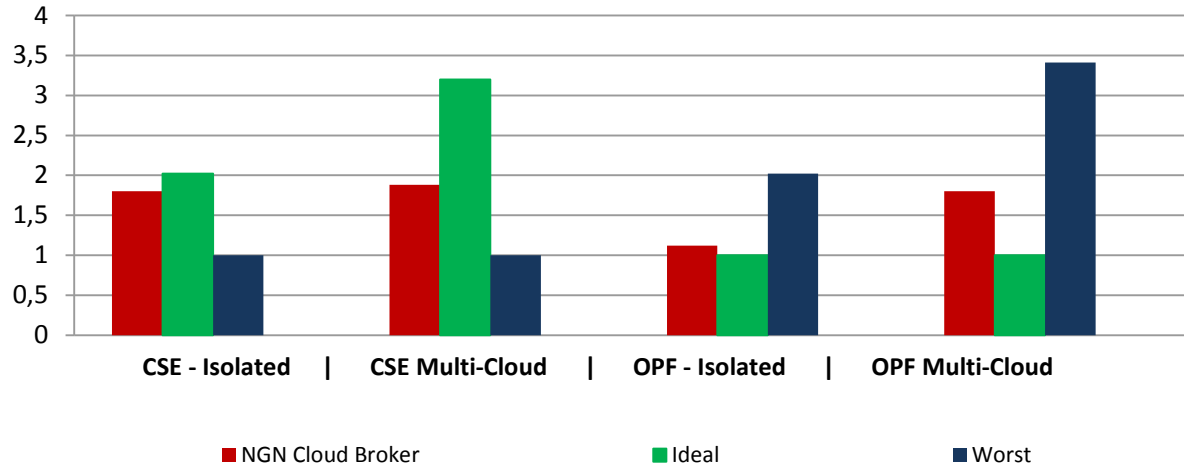

Figure 115: Capacity Saving Performance Evaluation

The results obtained through testing the NGN Cloud Broker's resource allocation performance in the isolated testbed, the FUSECO Playground (described in section 9.4.1), show that NGN QoS levels can be assured (within acceptable ranges) while allowing for a significant degree of capacity saving (55%) only requiring a small overprovisioning of resource (12%), however under artificial workload conditions.

The multi-cloud tests on BonFIRE, using realistic NGN workload, reveal lesser capacity saving performance, and require higher levels of overprovisioning. However, compared to a fully static allocation of resources, the auto-scaling mechanisms allowed for saving up to 47% of cloud capacity/costs, while still being able to assure NGN QoS levels.

Evaluation of *QoS Assurance Performance* against industry standards

As stated in the introduction of this work, in section 1.4.4 Classification of Key Performance Indicators, current telecommunication QoS standards should be used to evaluate NGN Cloud Broker 's performance for assuring *QoS classes* and SLA standards for *service accessibility / availability*.

QoS Classes:

The pan-European multi-cloud experiments and measurements have shown that whereas delay and packet loss values required for high quality delivery of NGN services have always been met, measured delay variations might indeed at times have significant impact on the overall NGN service quality. Whereas full assessment of provided network performances would have required long-term studies of network performance effects, already the limited measurements (~24hours) have shown that delay variations of up to 30ms are likely to occur.

In summary, the conducted measurements show that NGN service quality can indeed be provided by cloud-based NGN service deployments. Mitigation of network outages through migration of services from one public cloud infrastructure to another, involves up to 100 seconds of delay, during which QoS deteriorations (not necessarily leading to SLA violations) might occur. In the worst case, however, network or platform outages lead to entire service unavailability, which is being discussed below.

Service Availability / SLAs:

Carrier-grade networks are supposed to provide 99,999% ("five nines") availability of their services throughout the year. This allows for a maximum of 5 min 15s outages / service unavailability. The results of the conducted measurements of 100 seconds required for mitigating network / platform outages through service migration across platforms, would allow for 3 migrations per year. Commercial cloud offerings only start to provide compensations for SLA violations at significant lower levels (e.g. Amazon's EC2 starts defining SLA violations below 99,95% availability). Typically, also telecommunication operators/carriers do not provide five nine SLAs, providing 99,99% availability (a maximum of 53 minutes outage/unavailability per year) for their best (e.g. platin/gold) service classes. Taking again 100ms as a rough estimation of maximum migration delay, this would allow for up to 31 cloud platform migrations per year.

In summary, achieving carrier grade service availability through multi-cloud service management is very difficult to achieve. Neither cloud platforms, nor telecommunication operators guarantee true carrier grade SLAs. However, "four nines" are still conceived as high availability and the typical basis of telecom SLAs. With 31 possible migrations per year for mitigating network/cloud platform outages, long term experiments / tests could indeed proof that "four nines" SLAs could be satisfied.

Lifecycle management performance:

1) Monitoring delay < 100ms (retrieve latest data from monitoring DB)
2) Analysis / Plan Delay (< 50ms), monitoring data analysis, workload prediction/capacity planning, platform selection
3) *Resource provisioning delay* *50s-200s* (cloud platform dependent)
4) Service provisioning delay (NGN + dependent services/load balancer) < 100ms

Critical factors for improving cloud brokering performance:

It was shown that 1) differences of cloud platform virtual resource management performance (i.e. cloud resource provisioning delay), 2) differences in resource performance (even among identical resource types), 3) minimal lease time, and 4) the actual workload distribution are the *predominant factors influencing the level of achievable capacity saving performance*.

1) Main determining factor for the overall inertia of the cloud brokering system to (re-) allocate resources (up to 95% of the overall delay for provisioning resources and activating)
2) Main determining factor for required resources / capacity (up to 3 times as much resources are required when low performance cloud instances, of the same instance type, are chosen (provided by different cloud platforms)
3) Differences in minimal lease time / usage of a cloud resource have a significant impact on the achievable resource capacity saving (up to 70% redundant capacity of hourly lease times compared to minute-based leasing).

Given that 1), 2) are optimized, i.e. the optimal platform and optimal cloud instance type(s) is/are chosen, and 3) given that respective cloud platform providers do not impose limitations/restrictions in terms of minimal lease times (i.e. minute-based accounting, as opposed to hourly/daily), the actual performance limiting factors of the cloud brokering system are:

1) *system inertia* (i.e. time needed to 1) retrieve latest monitoring data, 2) analyze data, 3) plan allocation of resources for the next interval/step, 4) execution of provisioning requests) [<1 minute]

2) *workload prediction / capacity forecasting accuracy* (i.e. determining the required incorporation of over-provisioned capacity in order to cope with uncertain workload changes, that might occur during the next step/interval) [up to 40% increase of required capacity]

Overall Performance Evaluation

The overall performance of the cloud brokering system (the cloud broker including Cloud Platform, NGN platform and dependent applications) predominantly depends on the performance and pricing models of selected cloud platforms.

For as long as the performance of cloud resource instances are not specified in a way sufficiently expressive for allowing calculable capacity planning, for each particular service (to be deployed on a cloud platform) dimensioning, *profiling, and benchmarking is needed*. If not done with sufficient rigor, either overprovisioning or under-provisioning of resources is the consequence. Given that cloud resources have sufficiently been benchmarked, so that for each cloud resource instance its capacity (requests/sec, calls/sec, transactions/sec) for providing a specific service can be specified, resource efficient resource allocation can be realized. The importance of benchmarking cloud resource performance against service workload cannot be emphasized sufficiently. The conducted measurements show that up to three times as much resources (of the same instance type) would be required if this measure is neglected (benchmarking omitted).

Now, given that service specific resource capacities can be specified (e.g. a small instance is able to serve 100 requests per second of service x), the overall capacity saving performance of the overall system mainly depends on the overall inertia to (re-)allocate cloud resources. Again, this inertia is mainly determined by selected cloud platform`s inertia to instantiate or shut-down cloud resources. Each cloud platform needs to specify its capability for "rapid elasticity" for each instance type (i.e. micro, small, medium, large,..), or again must be measured/benchmarked.

Given that resource performances as well as cloud platform provisioning performance are benchmarked/determined sufficiently, the final *performance of the cloud brokering system* to elastically and efficiently allocate resources depends on its capability to 1) match capacity with required demand and 2) timely allocation of capacities.

The actual capacity saving performance, in turn, heavily depends on the actual workload distribution. Whereas slowly changing workload changes, can rather easily be managed, quickly changing workloads, i.e bursts, require either rapid adaptation of capacities or incorporation of sufficient overprovisioned capacities (to be omitted). Therefore, capacity saving performance of an elastic cloud brokering system should always be specified along with the actual workload's burstiness (ideally short term workload variations, as well as long-term/daily variations). Still, other factors as for instance the granularity / types of cloud resource instances have an impact on the overall achievable capacity saving performance, for an ideal capacity matching would require availability of cloud resource types of any size. Working with discrete resource types (small, medium, large) however opposes limitations in finding optimal capacities for specific workloads. The impact of this factor, however, decreases with the magnitude of workloads, e.g. where hundreds of instances are used to serve a particular workload demand.

9.9 Requirements Validation

Functional Requirements

All mandatory functional requirements, as specified in section 3.3 with a "MUST" have been satisfied by the NGN Cloud Broker. The optional functions, regarding the capability of monitoring variable costs of cloud instances, specified in section 3.3 with a "SHOULD" has not been implemented, for the following reasons. For the evaluation of the NGN Cloud Broker, a multi-cloud research infrastructure is used, which does neither have prices attached to its cloud instances, nor provides interfaces to read out the current prices. Therefore, the automatic retrieval of cloud instance prices is not implemented in the NGN Cloud Broker. For the evaluation, cloud instance prices are manually configured to the system and artificially emulated for evaluating the Cloud Platform Selection performance of the NGN Cloud Broker (as shown in Figure 105).

Table 18 Requirements - Cloud Broker Perspective

Requirement	Description	Fulfill-ment
Core Functions:		
Platform Selection	Cloud Brokers MUST provide means allowing for dynamic selection of Cloud Platforms and Cloud Resources according to pre-defined rules, evaluated against real-time monitoring data and enforced repeatedly.	yes
Resource Allocation	Cloud Brokers MUST provide means allowing for the dynamic allocation of Cloud resources across multiple cloud platforms according to pre-defined rules, evaluated against real-time monitoring data and enforced repeatedly.	yes
Support Functions:		
Cloud Infrastructur e Catalogue	Cloud Brokers MUST maintain a Cloud Infrastructure Catalogue which lists available Cloud Infrastructures, potentially eligible for telecommunication service to be deployed on.	yes
Service Registry and Repository	Cloud Brokers MUST maintain a Service Registry, which stores the necessary information describing the service's QoS requirements, describing service control protocols, ports, invocation methods. A service repository must be in place which is either maintained locally, or can be accessed remotely, which stores the application software.	yes
Monitoring Functions	Cloud Brokers MUST comprise versatile Monitoring *Aggregation Functions*, aggregating Monitoring information from distributed NGN Services, Cloud Infrastructures, Cloud Infrastructure Resources and Networks that interconnect NGNs and Cloud Platforms. Cloud Brokers MUST provide active and passive Monitoring Functions through which active and passive measurements of Network, Infrastructure, Resource and Service Performance can be conducted. Cloud Brokers SHOULD provide Cost Monitoring Functions through which they can monitor dynamically changing prices of e.g. Spot Cloud Resources. Explanation: For the evaluation of the system, artificially simulated prices for cloud instances were sufficient. For the real-word application of the NGN Cloud Broker, however, this functionality would be needed, although it is not common practice, for Cloud Providers to publish resource costs on an API basis. Therefore manual provisioning of cloud resource costs might be required in many cases.	yes yes no
Provisioning Mechanisms	Cloud Brokers MUST provide the means for the dynamic provisioning of Cloud Infrastructure Resources, Service Elements (e.g. required for Service Scalability such as Load Balancers) and NGN Service control Elements (e.g. IMS HSS)	yes
Pre-Requisites:		
Interconne ction:	Connectivity for interconnecting Cloud Brokers with NGN platforms and Cloud Platforms MUST be provided. Network performance between aforementioned interconnections, SHOULD be of "good" quality.	yes yes

Non-Functional Requirements

Non-functional, execution qualities, against which the proposed cloud brokering system is evaluated against are listed in Table 12 and recapitulated in Table 19.

Table 19 Non-functional Evaluation Criteria – Execution Qualities

Non-functional Evaluation Criteria	Description
NGN Service Quality Awareness	The cloud brokering system should not only be aware of resource utilization, but also be aware of the end-to-end service quality of managed NGN services
Network Performance Awareness	The cloud brokering system should not only be aware of resource utilization, but also be aware of network performance between NGN and Cloud Platforms
Fault Tolerance	The nature of the cloud brokering system's distributed operations, across domains and platforms, requires a high degree of fault tolerance, as failures in such federated environments are commonplace, especially if rather unreliable platforms participate.
Reliability	Failure of the cloud brokering system has a direct, negative impact on NGN service's availability and QoS. Therefore the cloud brokering system should provide high reliability.

The normative evaluation as shown in Figure 116, is justified in the following sections, where each of the non-functional execution qualities of the NGN Cloud Broker is evaluated against single Cloud auto-scalers (e.g. Amazon's auto-scaling) and multi-cloud auto-scalers (e.g. Scalr).

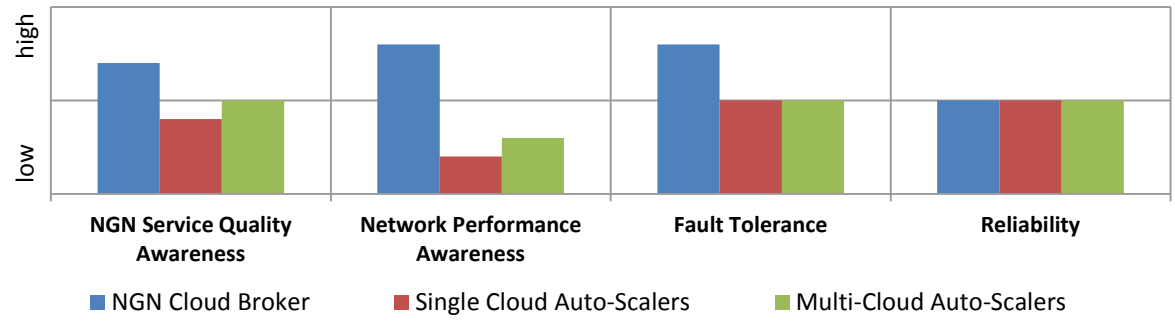

Figure 116: Evaluation of non-functional Execution Qualities

NGN Service Quality Awareness

Typically, commercial solutions, as well as service-agnostic academic approaches 1) do not support end-to-end service quality monitoring or at least 2) do not support NGN service quality monitoring (Scalr[6] for instance supports URL response time monitoring / scaling triggers based thereon). Thus, as the NGN Cloud Broker provides end-to-end NGN service quality monitoring mechanisms (i.e. end-to-end Voice quality is implemented and evaluated, but the service specific end-to-end quality monitoring could be extended in various dimensions), it provided enhanced end-to-end QoS awareness.

Network Performance Awareness

One of the core advantages of the NGN Cloud Broker compared to single cloud auto-scaling solutions (commercial or academic) relates to its capability to monitor NGN to Cloud network

[6] Scalr, Enterprise Cloud Management Platform, online: http://www.scalr.com, accessed 21st May 2014

performance and to trigger migration of services from one cloud platform (with insufficient connectivity / network performance) to an alternative cloud platform (with better connectivity / network performance). As has been shown, QoS deterioration might be resulting from various factors. It was shown that end-to-end QoS deteriorates in situations of cloud resource overload, but also in situations of network performance deteriorations (which cannot be resolved by increasing cloud resource capacities). Therefore, network performance awareness is highly critical for providing QoS-assured services on federated cloud environments.

Fault Tolerance

The NGN Cloud Broker compared to single cloud auto-scalers or current multi-cloud brokers (such as Scalr), provides enhanced fault tolerance since it is able to swap/migrate resources from one cloud platform to another. Whereas single outages or failures of cloud instances can be overcome by replacement / additional scaling mechanisms with current approaches, entire cloud platform outages (including network failures) can currently not be resolved with commercial solutions.

Reliability

Although long-term tests have shown that the NGN Cloud Broker is operating in a stable and reliable fashion over days and weeks, a comparative evaluation against the reliability of single cloud auto-scalers (e.g. Amazon auto-scaling) or multi-cloud auto-scalers (e.g. Sclar) has not been conducted. Therefore the reliability of the NGN Cloud Broker could not be evaluated to be neither better nor worse than approaches it is compared with.

Non-functional, evolution qualities, against which the proposed NGN Cloud Brokering system is being evaluated, are listed in Table 19.

Table 20 Non-functional Evaluation Criteria – Evolution Qualities

Non-functional Evaluation Criteria	**Description**
Maintainability / Configurability	Required efforts for system configuration changes needed to adapt the behavior of the cloud brokering system, for an existing service, or for configurations needed for the management / brokering of new services, should be kept as low as possible.
Platform Independence	The Cloud Brokering system should aim at supporting interworking with as many cloud platforms as possible, in order to benefit from a rich set of alternative cloud resource offerings, provided at different prices and at different QoS levels.
Scalability	With growing numbers of services to be brokered across cloud infrastructures, as well as with growing user demand and workloads, scalability of the Cloud Brokering system becomes critical. Thus, the cloud brokering system should be designed in an intrinsically scalable fashion, allowing on-demand scalability.
Service Agnosticism / Versatility	Core mechanisms of the cloud brokering system should be designed in a highly service agnostic manner, allowing versatile utilization of the system for a broad range of services.
Portability	Core mechanisms of the cloud brokering system should be deployable in different contexts and environments. To this end, the system should be designed in a highly portable fashion.
3rd Party Dependency	The cloud brokering system should be designed in a highly independent fashion,

	allowing for operation inside of a particular stakeholder domain, but also outside, provided as a services to multiple stakeholders / NGN service providers.
Extensibility	Core Mechanisms of the cloud brokering system should be extensible enabling its application in other contexts and other environments or environments requiring more complex capabilities.

The normative evaluation as shown in Figure 117, is justified in the following sections, where each of the non-functional evoluation qualities of the NGN Cloud Broker is evaluated against single Cloud auto-scalers (e.g. Amazon's auto-scaling) and multi-cloud auto-scalers (e.g. Scalr).

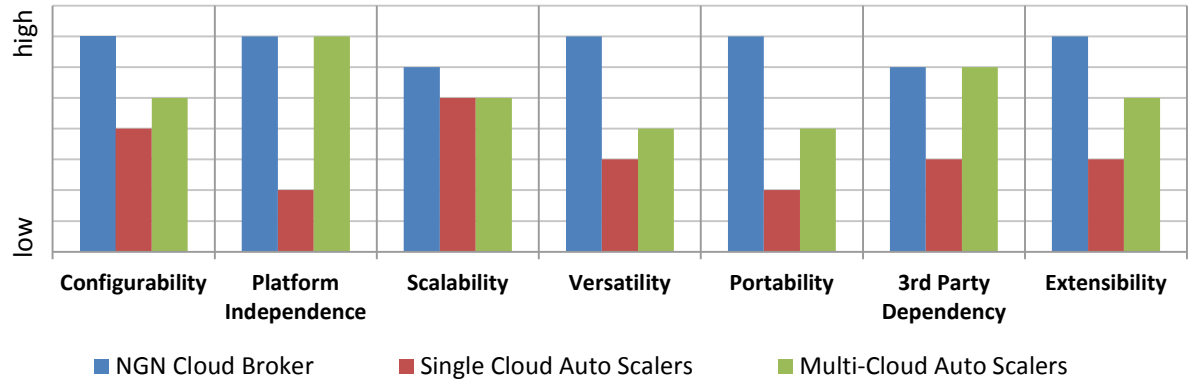

Figure 117: Evaluation of non-functional Evolution Qualities

Configurability

In contrast to Single-Cloud Scalers like Amazon's elastic auto-scaling functions, or Multi-Cloud Auto-Scalers like Scalr's auto-scaling functions, the NGN Cloud Broker provides powerful ways of configuring new policies and a significantly richer (and extensible) set of metrics (including monitoring tools) through which real-time network, platform, resource and application performance data (incl. QoS) can be monitored and scaling, as well as platform selection actions be defined.

Platform Independence

Through integration of multi-platform mediation functions of CompatibleOne, the NGN Cloud Broker is capable of interworking with numerous cloud management platforms (OpenNebula, OpenStack, VMware vCloud), and cloud management APIs of multiple vendors (Amazon, Azure, CloudSigma, Rackspace…). Furthermore, since the NGN Cloud Broker is scaling cloud resources externally (i.e. not requiring platform-internal scaling tools), a high degree of platform independence is achieved. Whereas Single Cloud Scalers like Amazon's auto-scaling functions are tailored to the Amazon EC2 environment, Multi-Cloud Scalers like Scalr are supporting a rich set of cloud management APIs of multiple cloud management systems (i.e. OpenStack, Cloudstack, Eucalyptus, Nimbula), and commercial cloud platform providers (i.e. Amazon EC2, Rackspace, Google Compute Engine). However, systems like Scalr only provide a-priori, static selection of cloud platforms and do not support dynamic switching/migration between cloud platforms during runtime.

Scalability

The modular system of the NGN Cloud Broker provides a high degree of scalability of its monitoring components (which provide hierarchical scalability), a high degree of scalability at the service orchestration/load-balancing level (since the IMS can be configured to forward service requests to multiple load-balancing system) and is intrinsically scalable at the core system level, which can be virtualized and deployed on clouds and scaled in the same way as the service instances it controls. A meaningful comparison with single-cloud auto-scalers such as Amazon's auto-scaling tools, or multi-cloud scaling tools, such as Sclar is difficult, as 1) details on whether scalability is incorporated at the load-balancing level and 2) for evaluation of high levels of user load would be needed to be generated. Based on the available documentation, it seems that the utilized load-balancing solutions do not provide integrated, dynamic scalability. However, scalability could certainly be achieved manually, by clustering service instances and utilizing multiple load-balancing systems in parallel, and incorporation of additional DNS-load-balancing solutions.

Versatility

With the NGN Cloud Broker higher levels of versatility are achieved, by also integrating SIP load-balancing functions. Thus, in contrast to single cloud auto-scalers like Amazon's tools, and multi-cloud scalers like Scalr, the NGN Cloud Broke not only supports Web applications, but also SIP-based NGN applications.

Portability

The NGN Cloud Broker is highly portable, as it can be deployed in various environments, internal, hybrid or multi-cloud contexts and portable to merely any Linux-based operating environment.

3rd Party Dependency

By providing all required systems and tools for platform selection and resource allocation / auto-scaling within an externally deployable system, the NGN Cloud Broker does not depend on cloud platform systems such as single-cloud auto-scaling systems provided by Amazon. However, systems such as Sclar, also provide all necessary components on a stand-alone, freely deploy-able fashion.

Extensibility

Due to its modular design, the openness of the integrated monitoring solution for supporting various additional monitoring tools and metrics, the openness and configurability of the rules engine, the NGN Cloud Broker can be extended in various fashions. In section 10.2, several deployments, integrations and extensions of the NGN Cloud Broker are illustrated, of which surely its extension towards SDN scalability and its integration into service brokering systems represent its most noteworthy extensions.

9.10 Factorial Impact Evaluation

It was shown that several factors have an impact on the QoS assurance performance and cost efficiency of elastic cloud brokering mechanisms scaling NGN services across multiple clouds.

For *optimal brokering* multiple cloud platforms:

1) *Network Performance* need to be monitored (round trip delay < 300 ms, jitter < 30 ms, packet loss < 2%)
2) *Cloud Resource Provisioning Delay* needs to be known, in the utilized multi-cloud testing environment varying between 60s (French cloud) and 200s (German cloud)
3) *Instance Capacities* need to be known, in the utilized multi-cloud testing environment ranging from 82 calls/s per German Cloud small instance type, to 160 calls/s per French Cloud small instance type
4) *Instance Prices and Instance Pricing* needs to be known, in order to translate instance capacities into value, but also for estimating overhead imposed by minimal instance lease times (often the case with commercial cloud providers, e.g. 1h min lease time)

For *optimizing resource/cost efficiency*:

- *Cloud Resource Provisioning Delays, Instance Capacities, Instance Prices and Instance Pricing* needs to be known (see above)
- *Impact of Resource Utilization on QoS* needs to be known, to avoid situations of overload
- *Impact of Workload on QoS* needs to be known for adjusting *Instance Capacity,* particularly in the case of NGN where the overall QoS relates to more than a single metric (i.e. signalling as well as media streaming)

The different factors impacting the NGN Cloud Broker's capability to save capacity and to assure QoS are summarized in the following Table 21.

Table 21 Evaluation of Factors impacting KPIs

	Impact				
KPIs	***Network***	***NGN Platform / Service Dependencies***	***Cloud Platform / Resources***	***Cloud Broker***	***Workload***
Capacity Saving Efficiency	**Low** small impact of network delay on cloud broker inertia	**Low** small impact of NGN platform provisioning delay on cloud broker inertia (~50-100ms) small impact of Load-Balancing provisioning delay (50ms)	**High** Resource: significant differences of **cloud instance performance** (up to factor 3 for similar instance types) Platform: significant differences of **cloud resource provisioning delay** – core factor determining cloud broker inertia (up to factor 4 in differences) significant impact of Platform: minimal resource lease time (commercial offerings vary from day, hour and 10 minutes minimal usage of an instance), up to factor 3 impact	**Medium** **cloud broker inertia** mainly determined by cloud platform performances. Network, NGN platform, monitoring delay, system inertia determined time to re-act comparably small factor. Inertia core factor for determining look-ahead step of workload prediction / capacity forecasting (45s – 3min) medium impact of workload prediction performance on required overprovisioning (~factor 0.5)	**High** main determining factor determining **maximum attainable capacity saving** workload variance/volatility / predictability determines required **overprovisioning factor** f (~volatility, ~cloud broker inertia)

	Impact				
	Network	***NGN Platform, Service Dependencies***	***Cloud Platform / Resources***	***Cloud Broker***	***Workload***
QoS Assurance Performance	**Medium** Overall QoS significantly depends on NGN – Cloud network performance. The conducted real-world measurements show that significant deteriorations of QoS caused by network performance deteriorations are rather rare (inter-European, best effort internet performance). If network outages occur however, the time to recover of at least 100s would be needed (strong dependency on cloud platform performance)	**Medium** small impact of NGN platform provisioning delay on cloud broker inertia (~50-100ms) small impact of Load-Balancing provisioning delay (50ms) medium impact of Load-Balancing algorithm / performance on single instance outages. weighted round robin algorithms recommended.	**High** Platform: Cloud instance provisioning delay, is the major factor determining the time to react (on overload, as well as on platform/network outages) Resource: Significant differences in QoS vs. workload behavior, even between cloud instances of the same type.	**High** QoS deteriorations caused by insufficient resource capacity mainly represents a failure of the Cloud Broker (capacity planning and/or inertia/slowness) QoS deterioration caused by platform/network failure mitigated by timely migration of resources from one platform (malicious) to an alternative platform (~100s+ time to recover / re-deploy)	**High** High workload volatility challenges the capabilities of the overall system to react on / predict rapidly increased demand/workload peaks of the cloud broker. Depending on the inertia of the overall system, as well as the incorporated overprovisioning factor, workload peaks can either be mitigated or result in temporary QoS deteriorations (static workload distributions, if capacity is sufficiently allocated, however have limited impact on the overall QoS assurance performance

9.11 Hypothesis Verification

Coming back to the initial hypothesis of this work:

"*By knowing a specific NGN service's resource capacity requirements, the End-to-End QoS requirements and the precise relationship between these requirements, a cost-/resource-efficient system, which federates multiple cloud infrastructures can be built that assures the specific QoS requirements of today's telecommunication industry*."

This work shows that by utilizing SOA principles, policy-based management mechanisms, and autonomic computing principles, NGN service platforms can indeed, under certain constraints and, within certain limits, benefit from federated cloud computing and cloud brokering mechanisms in terms of cost-efficient resource usage, while still being capable of assuring QoS requirements.

9.12 Comparison with related work

Different mechanisms, originating from different research fields contribute to the proposed approach for cloud brokering and elastic resource scaling, i.e.

1) multi-criteria platform and instance selection mechanisms,
2) capacity forecasting mechanisms,
3) auto-scaling mechanisms,
4) QoS management mechanisms,

The evolution and optimization of dynamic cloud brokering and elastic resource scaling mechanisms proceeds from *re-active*, to *pro-active*, to *predictive*, to *autonomous/self-manageable* [138] systems, where all of above mentioned areas and techniques are employed and iteratively tuned for optimizing resource consumption and for assuring QoS levels.

The evaluated cloud brokering system uses *workload-filtering mechanisms* (for capacity forecasting) and *auto-scaling* mechanisms (dynamically in-/decreasing resource capacities). Auto-scaling and cloud instance selection (described in [19]) is realized through round-robin load-balancing, selection of homogeneous types of VMs and policy-based service/cloud platform selection. Incorporation of feedback control mechanisms is left to future improvements.

Whereas the following sections provide an overview of how existing work relates to the concepts and approach of the NGN Cloud Broker, Table 22 summarizes the comparison of the NGN Cloud Broker with academic as well as commercial approaches.

Platform and Instance Selection

In principle, the problem of dynamically selecting the most appropriate / optimal cloud platform, as well as selecting a particular type cloud resources / instances, as made plausible also in [139] relates to the art of Multi Criteria Decision Making (MCDM) [140], where methods such as Analytical Hierarchy Processes (AHP), Multi-attribute Utility Theory (MAUT) or outranking methods like PROMETHEE or ELECTRE are used during the selection process. Similar to the approach taken for the NGN Cloud Broker, using an MCDN approach, i.e. following the typical process of decomposing the decision making problem into a system of hierarchies as described in [141], the work in [142] formalizes the multi-cloud service selection problem. However, no evaluation is provided, nor a concrete set of criteria, which are evaluated during the decision process. The multi-cloud service selection algorithm proposed in [143], using outranking mechanisms similar to the MCDM approach taken in this work. Also using MCDM techniques, the approach taken in [144] takes into account historic data on QoS performance, which are incorporated in the MCDM/AHP weighting process. However, QoS is measured in units of service response time, which is not predominantly relevant for NGN services, and furthermore and most importantly, the end-to-end service response time, is not measured separately from the network performance / delay. For the decision making process of the NGN Cloud Broker, this separation is vital, as an increase of end-to-end service response could be caused, either by an overload of capacities or by network congestions. Also similar to the approach taken in this work, the algorithm tries to minimize costs and to maximize gains. However, first the work stays vague in defining such gains, not taking into account clear exclusion criteria. Also in contrast to the approach taken in this work, [143] does not provide any evaluation of the approach. Similar to the approach taken in this work, [145] defines network QoS as an important criteria for selecting an optimal cloud platform. Similar to the NGN Cloud Broker, several QoS parameters (network as well as service related) are taken into account during the ranking process. However, first the algorithm does not provide a clear exclusion criterion (i.e. services are still taken into account during the ranking process, although the network is down), secondly costs are not taken into account and third non-service-specific hardware benchmarks are used for the scoring process.

Capacity Forecasting

Workload prediction mechanisms have been studied extensively in the past and already are applied in many fields of Grid and Cloud computing. For the dynamic allocation of cloud capacities workload prediction alone, due to non-linear application-specific effects provides only insufficient results. Therefore capacity forecasting involves application-specific profiling of demands vs. workloads. Work in [146] integrated an application-specific technique for modeling the dynamics of applications in the cloud allowing for feedback on the efficiency of capacity provisioning actions on resulting QoS and SLAs. Application-specific, empirical cloud capacity forecasting mechanisms make use of learning phases, where for instance neural networks are trained and sliding window techniques are applied as in [147], or regression models as in [148] or autoregressive moving average (ARMA) or auto-regressive integrated moving average (ARIMA) models as in [149] are used. In contrast to history-based approaches, it was intentionally refrained from approaches requiring training and in contrast to complex machine-learning approaches, it was intended to keep the computational overhead

limited (no pattern matching), thus apply simple moving average and exponential smoothing mechanisms for forecasting capacities.

Multi-Cloud elastic Auto-Scaling

Auto-scaling mechanisms allow for dynamic up-scaling and down-scaling of capacities based on resource utilization thresholds. The majority of approaches aim for optimizing dynamic scaling processes within a single cloud vertically as in [150], but predominantly horizontally. Here, apart from QoS/SLA assurance [151], optimizing the utilization of physical resources within a cloud-based data center [152] is a key objective.

Multi-Cloud resource allocation has been studied by authors of [153], who focus on scaling mechanisms for business-processes [154] and who focus on cost optimization and resource availability (as done also in this thesis), however, only in a static fashion and only providing simulations. Closest to the approach taken in this work, are Tordsson et al.'s work on models for dynamic scheduling [155] and cloud brokering mechanism for optimizing multi-cloud resource placements [156], which allow for budget, placement, load-balancing constraints, trying to optimize costs. In contrast to their very promising simulations, in this thesis real-world testing and evaluation is conducted, based on real workload, real cloud platforms and real NGN services.

QoS Management for Cloud-based NGN Services

In contrast to typical Web applications, where primarily metrics like service execution time determine the QoS, there are many different metrics relevant for determining the quality of NGN services, such as voice/video and signaling quality. Voice/video (over IP) quality can either be measured directly, e.g. by using the Perceptual Evaluation of Speech Quality (PESQ) parameter (i.e. ITU-T standard for end-to-end speech quality assessment [130]), but can also be inferred by measuring IP performance metrics (such as packet delay, jitter, packet loss). Therefor both means are utilized in this work.

QoS management of cloud-based series mainly deals with monitoring QoS and finding means to improve QoS. In the case of focused mechanisms of this work, these means are limited to either 1) moving services to alternative cloud platforms or 2) increasing resource capacities within a particular cloud platform. Whereas significant amount of work has been conducted in the context of cloud-based (multi-tier) Web service QoS management [157], apart from works published by the author on QoS-aware NGN service management [18] work on QoS management for cloud-based SIP/IMS/NGN applications and services is limited.

Table 22 Comparison with academic and commercial approaches

Cost Awareness	*QoS Awareness*	*Network Awareness*	*Single Cloud/Prov.*	*Multi-Cloud*	*Platform Selection*	*Workload Forecast*	*Auto-Scaling*	*Web Service*	*NGN Service*	*Benchmarking*	*Architecture*	***Main Area***	***Approach***	***Description***
X	(x)		X	X	X		X	X			X	Multi-Cloud Brokering	Academic: EU FP7 OPTIMIS[7] [158][156]	Definitely the closest match in terms of similarity with the QOSMUC framework and the instantiation of the NGN Cloud Broker are the approaches and developments of the EU FP7 Project OPTIMIS. The published work of OPTIMIS related to cloud brokerage and elastic multi-cloud resource allocation, provides algorithms, which allow for optimized resource allocation under budget, placement and load-balancing constraints. Simulations only are provided, which solve the optimization problem by means of binary integer programming. However, the work on cloud brokering of the OPTIMIS project, 1) stays agnostic to services, e.g. requirements of NGN/multimedia services are not reflected, 2) agnostic to real workload distributions (critical for evaluating the applicability of and performance of the cloud brokering/resource allocation function), 3) lacks real-world evaluations, 4) seems unaware of connectivity/network performance. As a consequence the performance of the OPTIMIS cloud brokering system, capacity savings, as well as e2e QoS assurance performance, could not be compared to this work.
X			X	X			X	X			X	Cloud Resource Allocation	Academic, EU FP7 RESERVOIR[8] / Claudia Platform [119][159] [160][161]	The RESERVOIR project developed a set of tools for cloud monitoring, policy models, and policy engines for service management, enhancements to the cloud OpenNebula management solution. Conceptually, the RESERVOIR service management platform “Claudia”, shared many similarities with the NGN Cloud Broker, receiving service-related policies, orchestrating elastic resource allocation requests. However, the software was only shortly publicly accessible, poorly documented and incomplete. Performance evaluations, apart from high-level simulations can hardly be found.
			X				X	X				Cloud	Commerci	Clearly the most famous IaaS Cloud Offering is the Elastic Compute Cloud (EC2) of Amazon.

[7]EU FP7 OPTIMIS Project, online: http://www.optimis-project.eu, accessed 21st May 2014

[8]EU FP7 RESERVOIR Project, online: http://www.reservoir-fp7.eu,accessed 21st May 2014

Cost Awareness	*QoS Awareness*	*Network Awareness*	*Single Cloud/Prov.*	*Multi-Cloud*	*Platform Selection*	*Workload Forecast*	*Auto-Scaling*	*Web Service*	*NGN Service*	*Benchmarking*	*Architecture*	***Main Area***	***Approach***	***Description***
												Load-Balancing, Auto-Scaling	al, Amazon EC2 Elastic Load-Balancing[9], Auto-Scaling[10]	Similarly to the components of the NGN Cloud Broker, Amazon provides PaaS tools for 1) cloud monitoring, 2) load-balancing,, 3) auto-scaling. In combination, time- or event-based scaling for EC2 instances is provided. However, 1) still lacking SIP-load-balancing mechanisms, therefore not applicable to NGN services, 2) QoS agnostic, thus appropriate scaling policies (mainly CPU-based) need to be determined by users. As with many other single-cloud examples such as Rackspace' [162] "*Cloud Loadbalancers*", auto-scaling systems provided for use within ElasticHosts' [163] cloud platforms, 1) can currently not be used for NGN services, 2) are by nature unaware of network performances, 3) not capable of selecting alternative platforms 4) do not provide cost-aware brokering support.
	x		X	X			X	X				Multi-Cloud Auto-Scaling	Commercial: SCALR[11]	The SCALR multi-cloud system represents a commercial solution, which shares most communality with the NGN Cloud Broker. Similar to all other approaches, due to the lack of SIP Load-Balancing mechanisms, NGN service auto-scaling is not supported. However, the SCALR toolkit supports multi-cloud resource allocation and multi-policy-based auto-scaling. Several commercial cloud platforms, such as Amazon's AWS, Rackspace and Google Compute Engine as well as several cloud management platforms such as OpenStack, CloudStack and Eucalyptus are supported. An Orchestration Engine supports policy-based execution of actions on different nodes, grouped together as a farm. Similar to the approach of this work, based on local agents "Scalarizr Agents", and a monitoring system, auto-scaling is conducted based on real-time monitoring data. Cloud Platform selection is not supported, thus dynamic migration of services across cloud platforms is not supported. Cloud platforms are initially selected and kept. Neither cost, nor network performance monitoring/awareness is currently supported. QoS-based scaling is supported for URL Response time only.

[9] Amazon Web Services, Elastic Loadbalancing, online: http://aws.amazon.com/elasticloadbalancing, accessed 21st May 2014

[10] Amazon Web Services, Auto-Scaling, online: http://aws.amazon.com/autoscaling, accessed 21st May 2014

[11] Scalr, Enterprise Cloud Management Platform, online: http://www.scalr.com, accessed 21st May 2014

Chapter 10

Summary & Outlook

10.1 Summary

This thesis shows that significant amounts of costs (up to 47% for typical daily NGN workloads with the NGN Cloud Broker in its current version) can be saved by elastically brokering cloud resources across multiple cloud platforms without violating SLAs or deteriorating NGN service qualities.

Full carrier-grade (99,999% availability) provision of NGN services on public clouds, however, would 1) require cloud platforms to offer carrier grade SLAs, or alternatively a reduction of the duration of service outages caused by inter-cloud service migration delays and 2) increased reliability of NGN to Cloud network performance (soon to be expected by emerging SDN solutions).

This thesis also shows that without knowledge of dependencies between application-specific QoS, cloud-platform-specific capacities and provisioning performances and cloud-broker-specific performance, cost-optimal and QoS assured multi-cloud brokering is not possible. It was found out that significant differences (by factor 2 and greater) of application performances on different, seemingly similar cloud resources, have a significant impact on the overall costs and required capacities.

Currently, there are no standard metrics, nor methods available, which allow for a-priori specification of expected costs and QoS assurance levels of cloud brokering / auto-scaling mechanisms. Amazon's Elastic Compute Unit (ECU) metric, for instance, being agnostic to specific applications, is not sufficient for assessing compute capacities of cloud instances. Application-specific benchmarks for cloud compute capacities are needed in order to 1) be able to a-priori asses costs, 2) assure efficient and reliable cloud brokering without the required manual benchmarking and profiling.

For this purpose, the author's team joined the IEEE P2302 InterCloud Testbed initiative [34] (details in section 11.2.5), where the knowledge gained in the scope of the presented work is further exploited for coming up with global standards for cloud interoperability and cloud federation.

10.2 Contributions and Impact

The full workflow of this thesis, which finally leads to the production of several distinct artifacts and impacts, which will be described in the following sections, is shown in Figure 118.

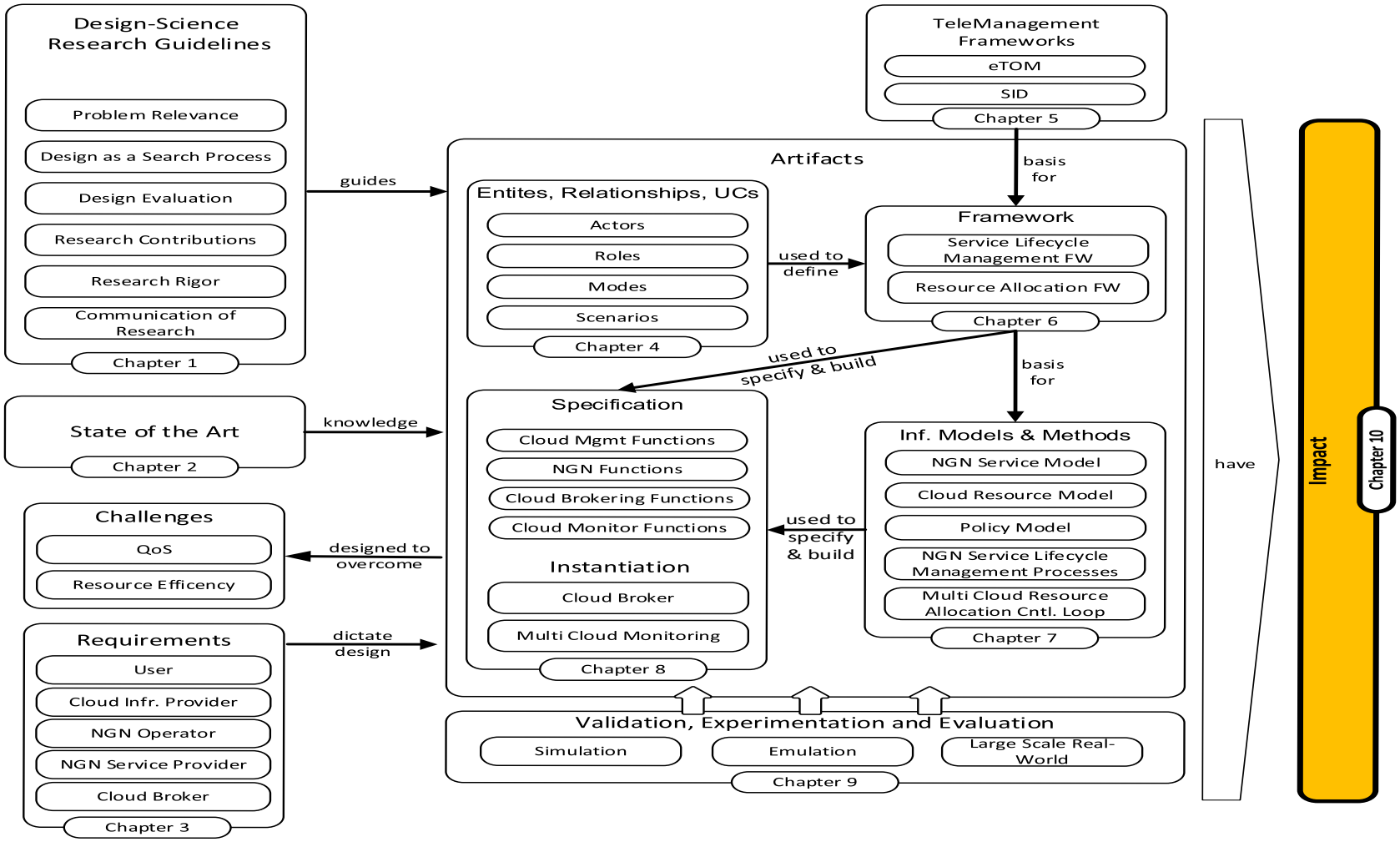

Figure 118: Thesis Workflow, Artifacts and Impact

In chapter 1, following standard design science guidelines [50], relevance, hypothesis, research questions, as well as the overall scientific approach, are briefly explained.

In chapter 2, an in-depth state-of-the-art analysis of current NGN and Cloud technologies provide the fundamental knowledge for designing the artifacts produced in this work.

Challenges and a requirements analysis for overcoming those challenges of cloud-based, cross-platform NGN service delivery are provided in chapter 3, differentiating the viewpoints of the different parties involved.

This allowed identifying potentially applicable approaches from the pool of approaches outlined in the previous chapter, narrowed down the search space and helped identifying and shaping the actors/entities, their roles, relationships and modes of action and scenarios/use-cases explained in chapter 4.

Together with well accepted frameworks (processes and information models) from the telecommunication industry, introduced in chapter 5, these previously defined entity relationship models are subsequently used in chapter 6 to define domain-specific frameworks 1) for the management of the lifecycle of NGN services in federated cloud environments (i.e. the workflow from service deployment, provisioning to operation and termination) as well as 2) for the resource allocation control cycles for the actual service operation phase (i.e. where cloud resources are dynamically scaled and allocated across multiple cloud platforms based on the current workload, taking into account QoS performance parameters).

These frameworks are subsequently used in chapter 7 for modeling the information of NGN services (their QoS requirements, interdependencies with dependent service building blocks, and required resources), Cloud resources (including a cost model) and policies, as well as for establishing a methodology for the orchestration of the processes required for the management of the service lifecycle, and for the allocation of resources in federated cloud environments.

Based on the QOSMUC framework and information models and methodologies, chapter 8 provides the actual specification and instantiation of involved Cloud management, NGN service control, cloud brokering and monitoring functions, based on which the actual implementation of the core cloud brokering and monitoring system and the integration into NGN service platforms and Cloud management platforms is described subsequently.

The overall system is subsequently validated through deployment and experimentation in multiple scenarios and several core aspects are evaluated, as described in chapter 9.

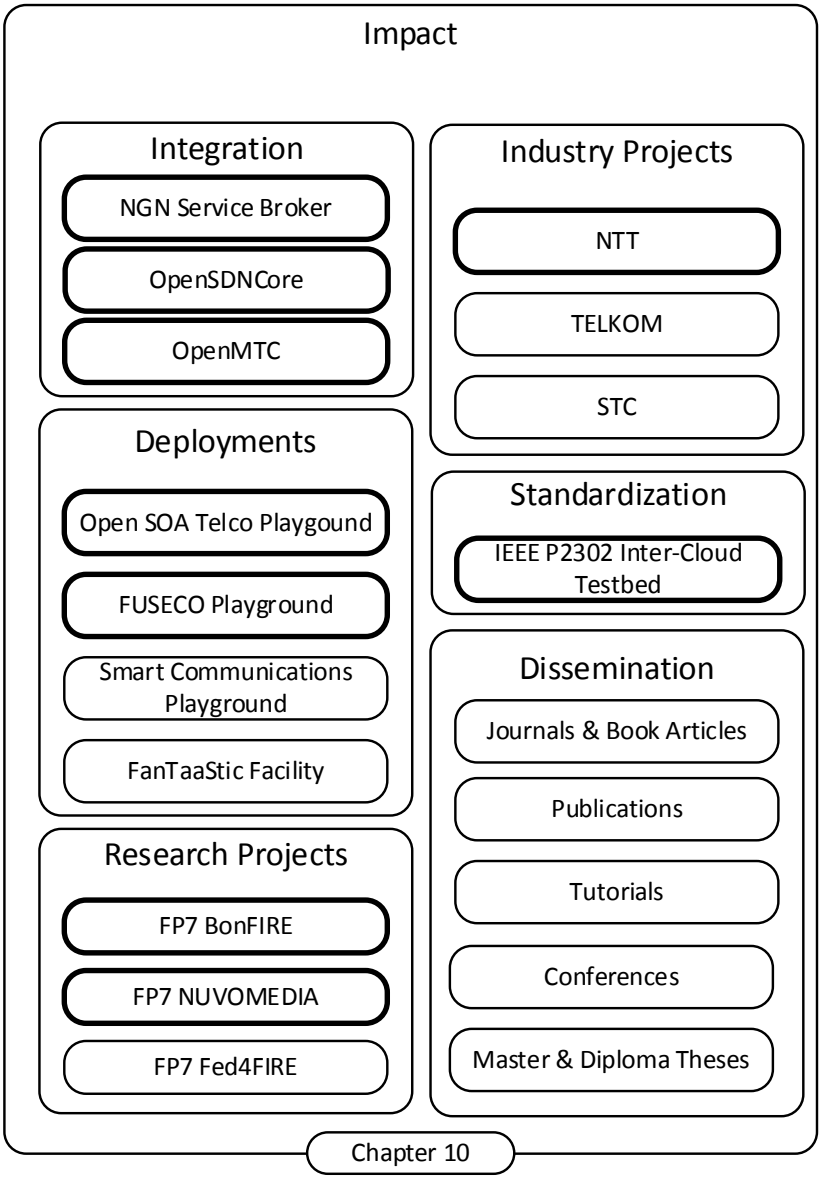

Figure 119: Impact of Thesis

The artefacts produced in this work and the knowledge obtained, is exploited in several dimensions as shown in Figure 119; ranging from 1) sustained deployments in operational testbed environments, 2) integrations into commercial solutions, 4) utilizations and further enhancements in research and industry projects, and 5) for standardization and for dissemination and educational purposes.

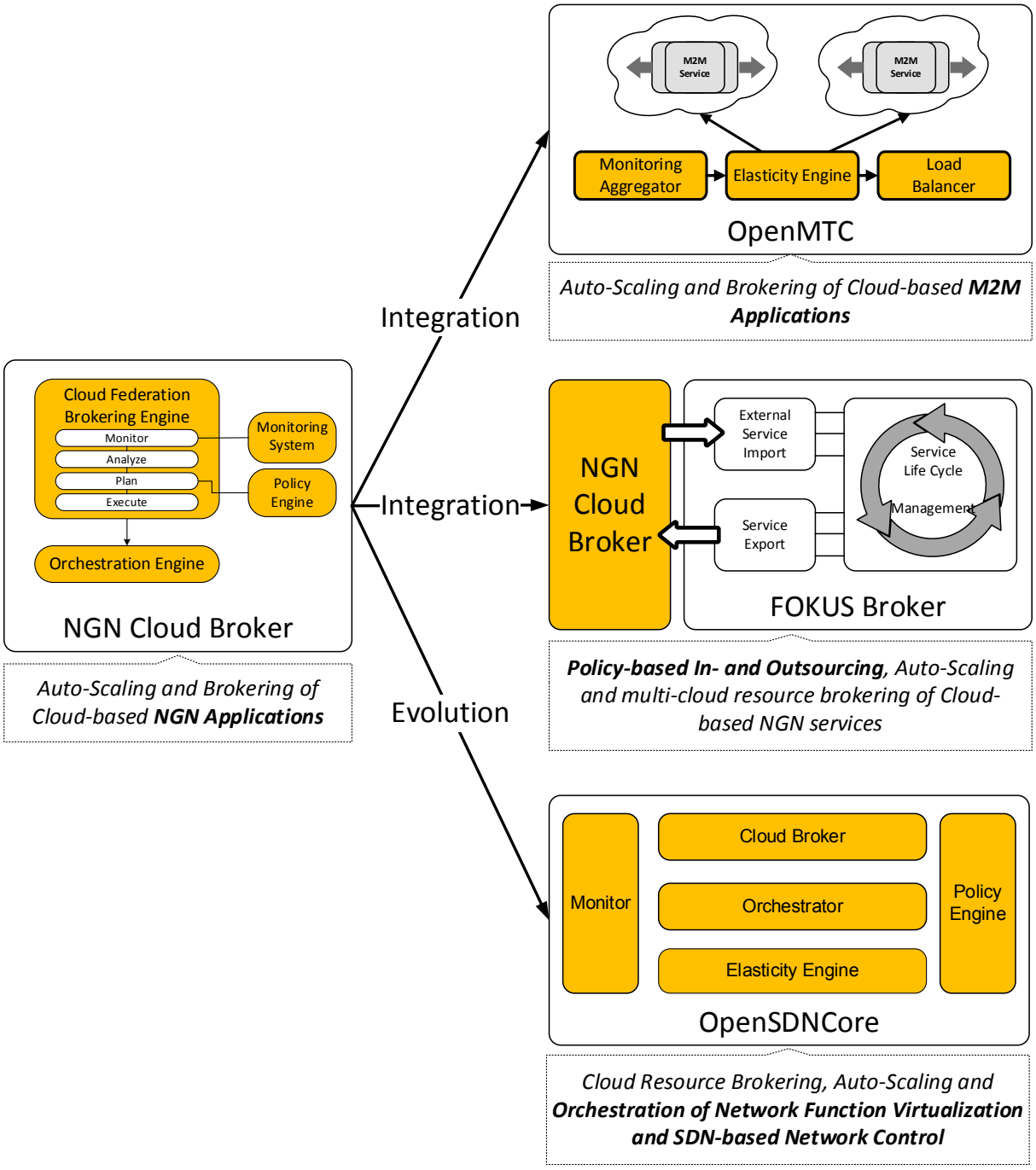

Figure 120 Integration and evolution of the NGN Cloud Broker

Of particular importance for the impact achieved by the developments carried out in the context of this thesis, are the utilization and integration of the NGN Cloud Broker for enhancing M2M, NGN service brokering and NFV/SDN platforms.

As shown in Figure 120, the following sections will explain 1) the integration of NGN Cloud Broker components into the Machine-to-Machine Platform OpenMTC for achieving Auto-Scaling and Brokering of Cloud-based M2M applications, 2) the integration of the NGN Cloud Broker into the OMA standard based SDP/service brokering platform, allowing for policy-based in-and out-sourcing of NGN services, most importantly 3) *the evolution of the*

NGN Cloud Broker towards converged Cloud, Network Function Virtualization and Software Defined Networking orchestration, realized by the OpenSDNCore.

10.2.1 Integration into NGN Platforms

Service Broker – Cloud Broker Integration

The FOKUS Broker, represents the heart of the Open SOA Telco Playground (see sections below, Figure 126), a SOA-based, NGN service environment, based on OMA standards. Clearly the work on the FOKUS Broker as documented in the author's publications in [8], [9], [7], and [3] marks the beginning of the developments of the NGN Cloud Broker. Based on the OMA specification of the OMA service environment [94], a SOA-based service delivery platform, was developed, which allows for 1) policy-based access control, 2) NGN/Web 2.0 service compositions, 3) integrated service management [13], [14], [16] 4) service discovery, and 5) versatile service import and export.

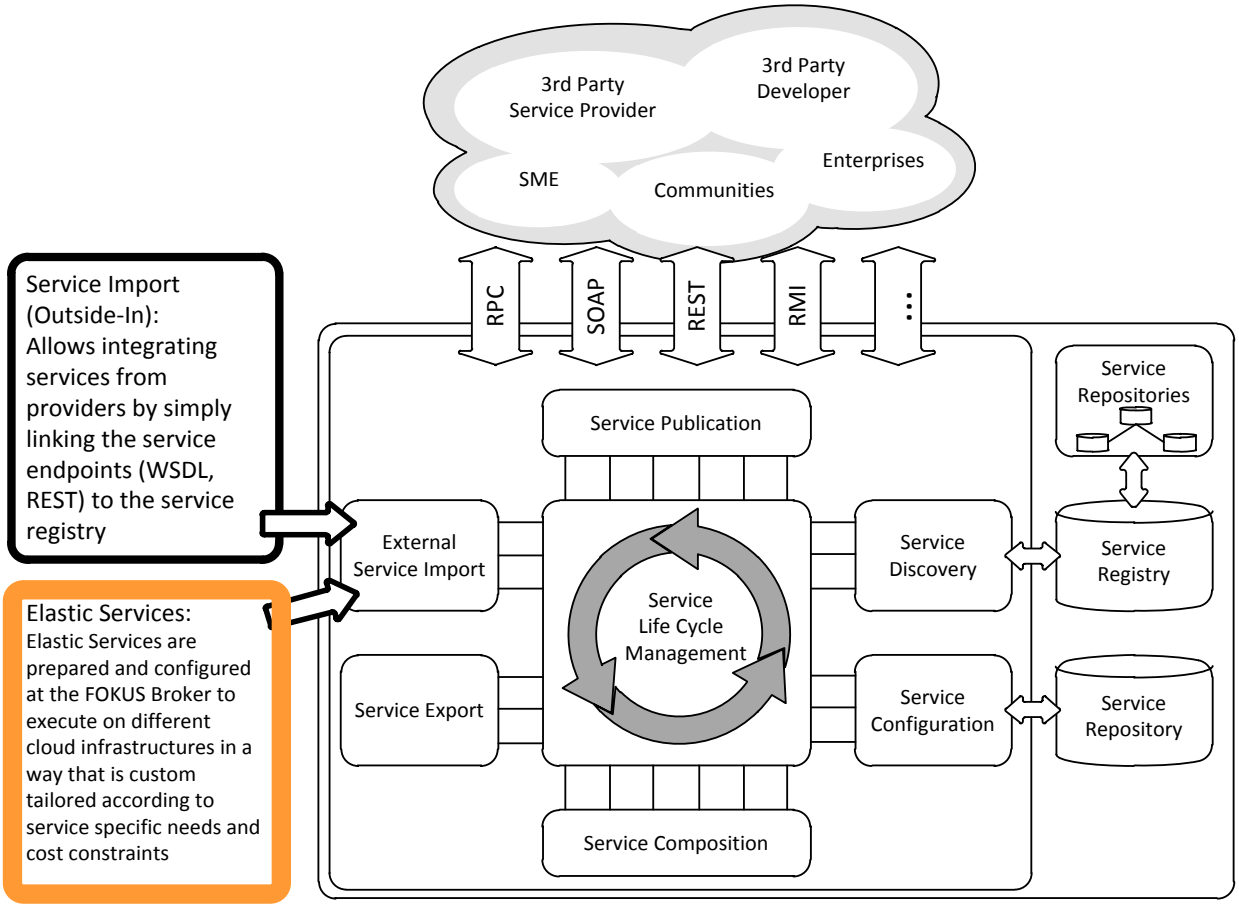

Figure 121: Integration of the NGN Cloud Broker's Elasticity into the FOKUS Broker

As documented in [21], the integration of cloud brokering and elastic scaling mechanisms into the FOKUS Broker, enhances its capabilities in several dimensions. Based on service specific needs, user and service-specific policies and cost constraints, the FOKUS Broker is since capable of executing services deployed and scaled on multiple cloud infrastructures.

This provides several benefits to NGN service providers, being able to control which specific service's cloud deployment is provided to which specific user. Identical services can since be provided on different cloud platforms, allowing for user/subscriber-specific provision of cloud-based services, where quality differentiations can be made, based on customer segments. Whereas premium customers are provided with highly reliable NGN services hosted on highly reliable cloud infrastructures (e.g. local clouds on the premises of the NGN operator), budget customers are provided with services hosted on public, best-effort cloud infrastructures, at lower SLA levels.

Integration of Elasticity into M2M Platform

The architecture of the OpenMTC[12] platform is shown in Figure 122. The OpenMTC, developed at Fraunhofer FOKUS, is an ETSI machine-to-machine (M2M) compliant middleware solution, connecting arbitrary numbers of devices (i.e. sensors and actuators) with arbitrary numbers of applications. The Network Service Capability Layer (NSCL), collecting sensor data from various sources (home gateways, attached sensors, etc.), and exposing sensor data to Network Applications (NA), the consumers of Smart Metering data represents the heart of the OpenMTC platform.

[12] www.open-mtc.org

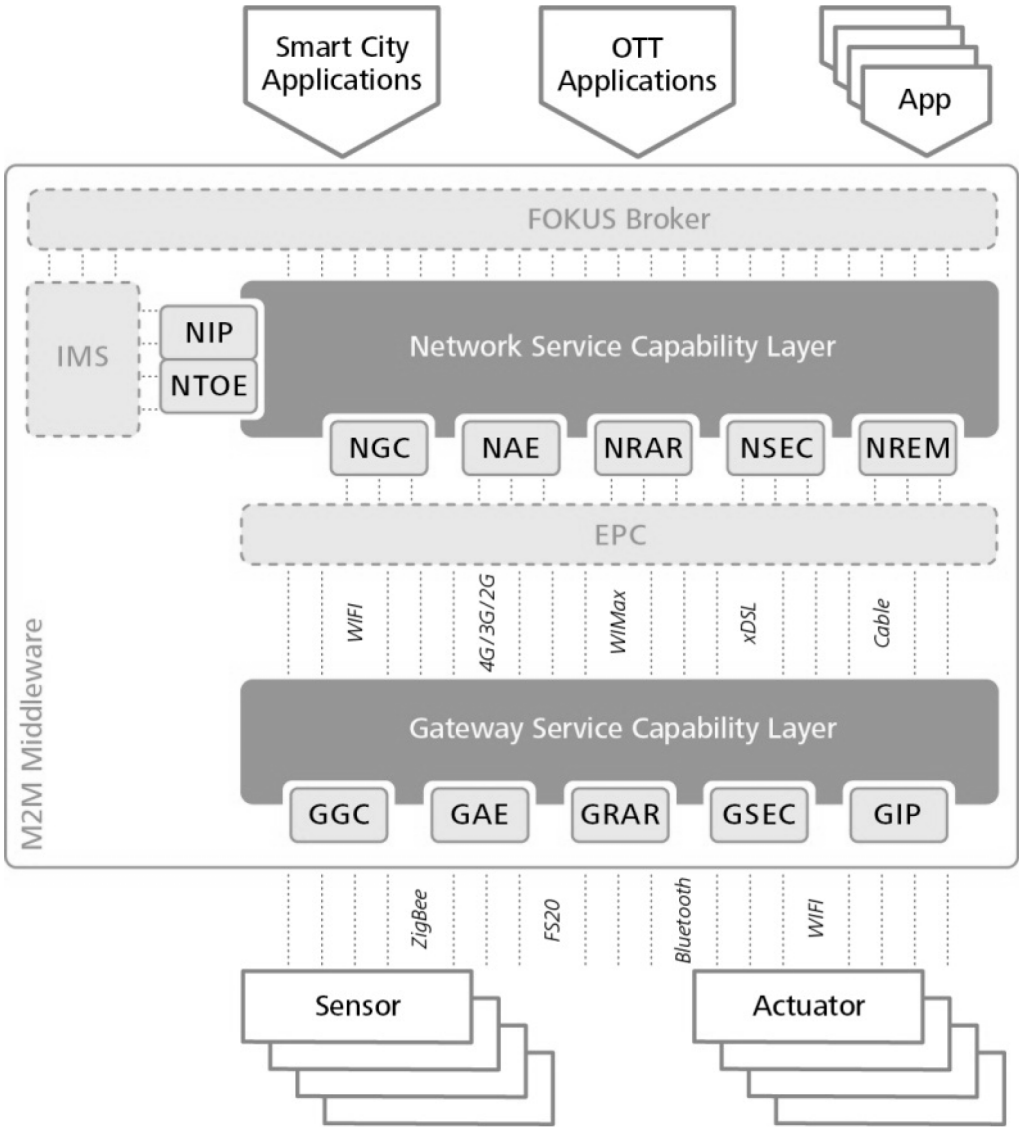

Figure 122: OpenMTC Architecture

With growing amounts of sensor data, as well as with growing numbers of network applications making use of the aggregated data, the workload of the NSCL increases. By deploying the OpenMTC platform on multiple cloud infrastructures and by integrating the NGN Cloud Broker's Elasticity Engine, load-balancing and cloud monitoring components as shown by its integration into BonFIRE in Figure 123, elastic scalability for the OpenMTC was realized.

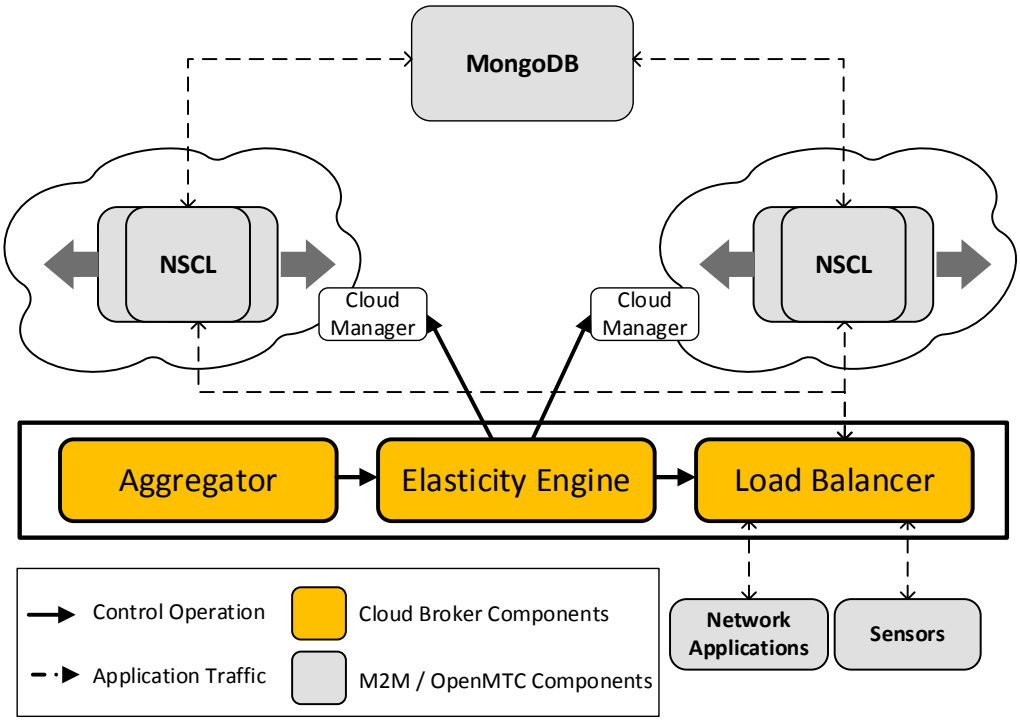

Figure 123: Cloud Broker's Elasticity integration into the OpenMTC's NSCL

Elastic auto-scaling of the OpenMTC platform was successfully demonstrated to the European Commission, the BonFIRE consortium and several reviewers at the BonFIRE project review in May 2012. It was shown that with in- and de-creasing workloads of the NSCL, the elasticity engine of the NGN Cloud Broker is capable of dynamically scaling NSCL instances across multiple clouds.

Evolution of towards converged Cloud/NFV/SDN Platform

The OpenSDNCore[13], currently developed at Fraunhofer FOKUS and available in its first release, realizes SDN and SDN concepts based on ETSI NFV MANO, ONF OpenFlow standards. Self-adaptation functionalities are realized through network orchestration and management, integrated into OpenStack, and adaptable control platform and programmable OpenFlow switches.

[13] The OpenSDNCore, SDN and NFV Management and Orchestration Toolkit, online: www.opensdncore.org, accessed 21st May 2014

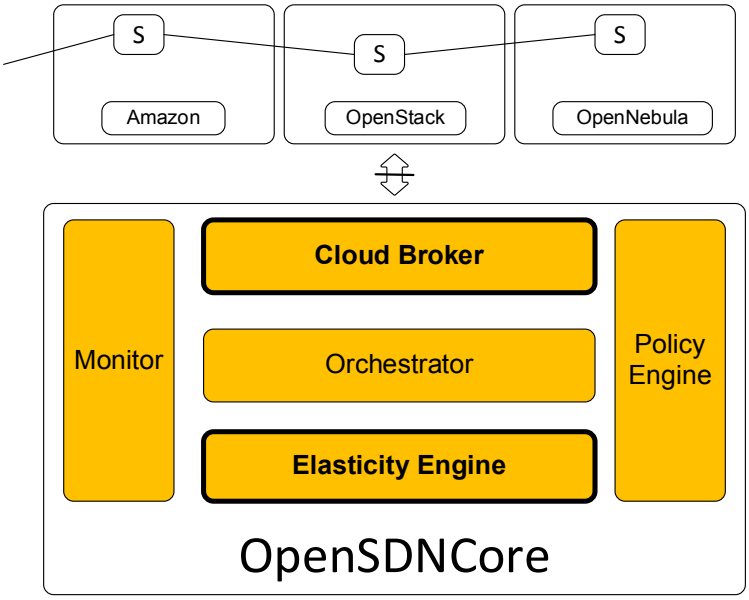

Figure 124: Integration of the NGN Cloud Broker and Elasticity Engine into the OpenSDNCore

Shown in Figure 124 is the current integration of the NGN Cloud Broker and its Elasticity Engine into the OpenSDNCore. The basic unit, orchestrated and elastically scaled through the upcoming release 2 of the OpenSDNCore is "the atomic resource", a virtual machine, which serves as a container for a service. It can be provisioned via an API and placed in multiple regions / countries. A service is comprised by a set of units, i.e. a composition of units that are exposed to the final end-users or to other services. A topology is comprised by a set of services, combined, e.g. a complete IMS, ot a complete EPC.

The orchestrator is managing the lifecycle of those topologies on top of multiple clouds. During the instantiation phase the orchestrator requests some resources in form of a unit from the cloud broker. The cloud broker decides where to place the units, and returns the information of the created unit to the orchestrator, which then manages the lifecycle of the services running on top.

The elasticity engine controls the behaviour of the instantiated topology. So in case of capacity bottlenecks of a service, the elasticity engine sends a request to the orchestrator for increasing/decreasing the number of units of that particular service.

10.2.2 Deployments in Testbed environments

The FUSECO Playground - Federated IMS/EPC + Cloud Testbed

The FUSECO Playground[14] at Fraunhofer Insitute FOKUS is a unique mobile broadband testbed, which, based on the OpenEPC[15], and the OpenIMSCore[16] provides a complete

[14] FUSECO Playground, www.fuseco-playground.org

mobile core network for testing and experimentation. The wireless lab, depicted in Figure 125, provides full 2G/3G/LTE and WiFi connectivity to regular phones, SIP and IMS clients for both circuit switched (CS) and packet switched (PS) services. Together with the cloud testbed, which provides OpenNebula[17], as well as OpenStack[18] based cloud management capabilities; a broad range of mobile cloud scenarios can rapidly be provisioned and set up, for various cross-cutting tests and experiments. Flexible and rapid resource and service provisioning, reservation and control is realized through FITeagle[19], a unique Future Internet testbed experimentation and management framework, the core of FUSECO Playground's testbed control center. It moreover allows federation of the FUSECO Playground testbed including its IMS, EPC, Cloud resources and services, with external testbeds and provides remote access to external users. Through its capabilities for federation with other testbeds, the FUSECO Playground became one of the first testbeds part of the community of the pan-European federation of Future Internet testbeds, currently governed by the Fed4FIRE project [5].

[15] Open Evolved Packet Core, OpenEPC, online: www.openepc.net, accessed 21st May 2014

[16] Open IMS Core, online: www.openimscore.org, accessed 21st May 2014

[17] OpenNebula Virtual Infrastructure Management Solution, online: www.opennebula.org, accessed 21st May 2014

[18] OpenStack,Virtual Infrastructure Management Solution, www.openstack.org, accessed 21st May 2014

[19] FITeagle, Future Internet Testbed Experimentation and Management Framework, www.fiteagle.org, accessed 21st May 2014

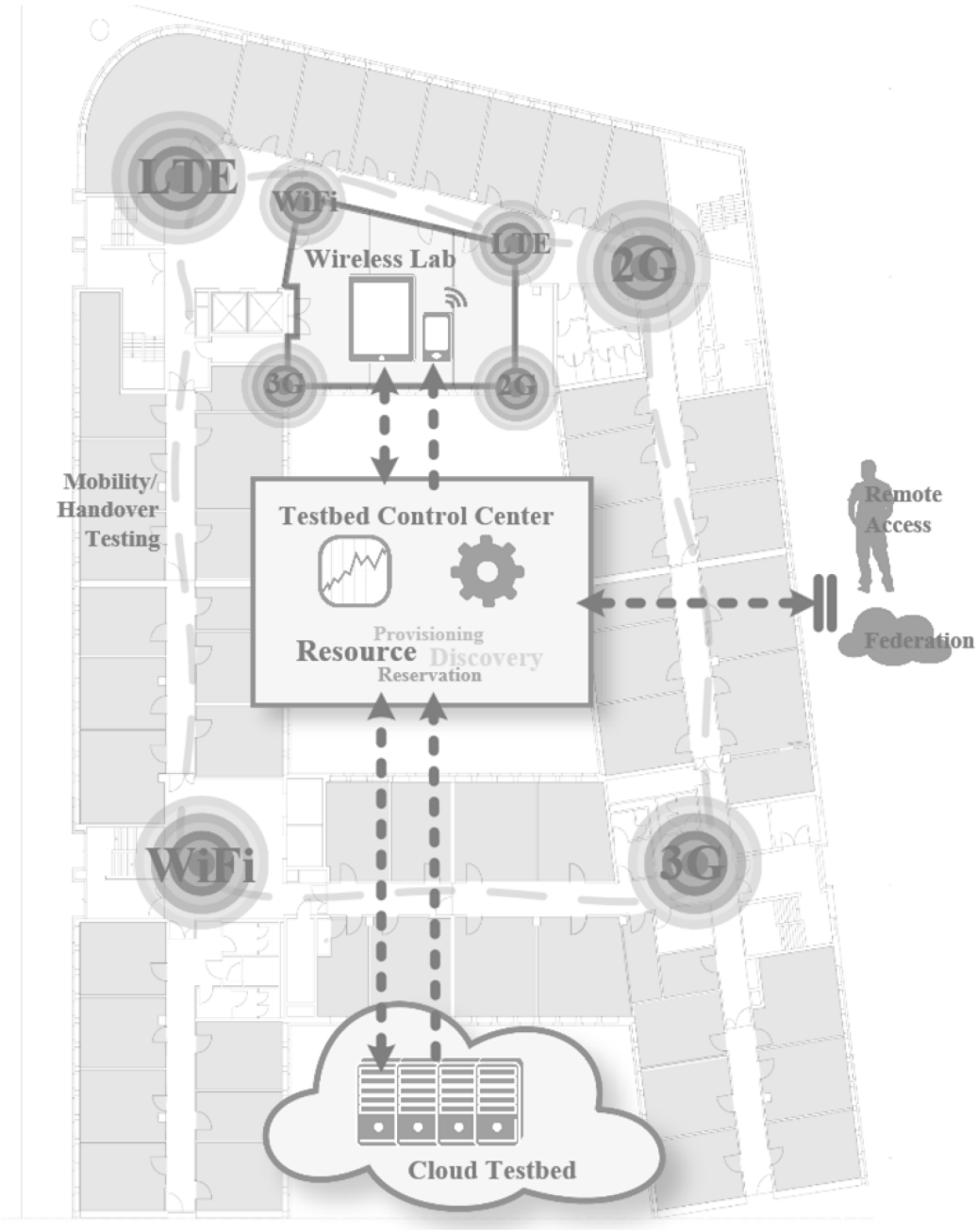

Figure 125: FUSECO Playground @ Fraunhofer FOKUS

The converged IMS/EPC management and cloud brokering and monitoring solution, developed in this work is tightly integrated into the FUSECO Playground. Initially allowing for OpenNebula-based elastic resource control, subsequently also for OpenStack-based control, the cloud testbed component of the FUSECO playground was invited to join the IEEE 2032 Intercloud testbed [116]. Several in-house tests and experiments conducted in the course of this work exploited the unique capabilities of the FUSECO Playground. Showcasing of thereby achieved dynamic, cross-cutting (i.e. application, compute and network) control

capabilities at various events[20] underpinned the international recognition of this unique testbed infrastructure.

The Open SOA Telco Playground

The Open SOA Telco Playground[21] [14] represents Fraunhofer FOKUS' NGN service platform, i.e. the application layer of the NGN testbeds at Fraunhofer FOKUS. Based on state-of-the-art Service-Oriented Architecture (SOA) principles, it provides a generic service platform, communication and connectivity interfaces for various service verticals. Figure 126 provides a high-level overview of the architectural components and enabled usage areas.

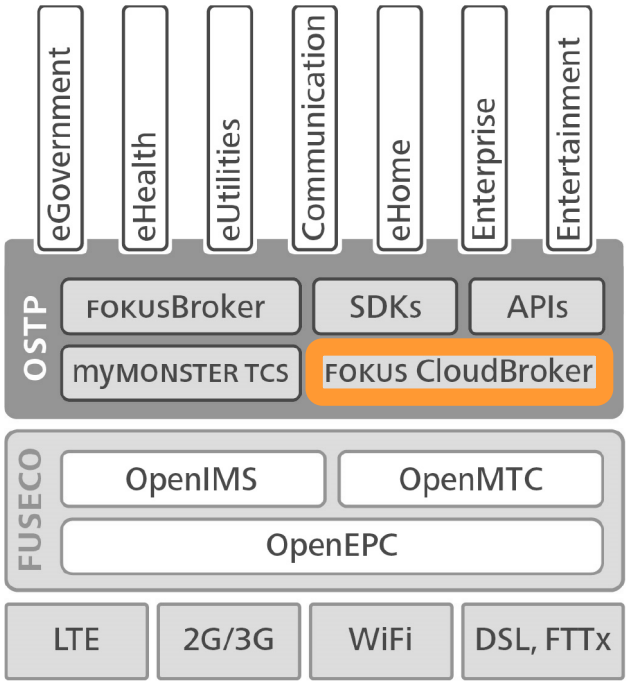

Figure 126: Open SOA Telco Playground High-Level

By tightly integrating the cloud brokering and monitoring solutions, developed in the context of this work into the Open SOA Telco Playground, particularly through tight integration with the policy-based FOKUS Broker, as described in [21], new levels of flexibility for NGN service delivery are achieved. By doing so, not only new levels of scalability are achieved, but also in- and out-sourcing of compute capabilities, including hybrid scenarios are realized. Based on the policy control and service execution capabilities of the FOKUS Broker, as shown in Figure 127, it is not only possible to dynamically decide WHO, can use WHICH service enabler of the SDP, but also WHERE (on which cloud

[20] FUSECO Playground Demonstration at the FUSECO Forum 2013, online: http://www.fokus.fraunhofer.de/en/fokus_events/ngni/fuseco_forum_2013/Demos/The_FUSECO_Playground/index.html, accessed 21st May 2014

[21] The Open SOA Telco Playground, online:www.opensoaplayground.org, accessed 21st May 2014

infrastructure) service enablers are being executed, allowing for increased flexibility and also providing economic benefits.

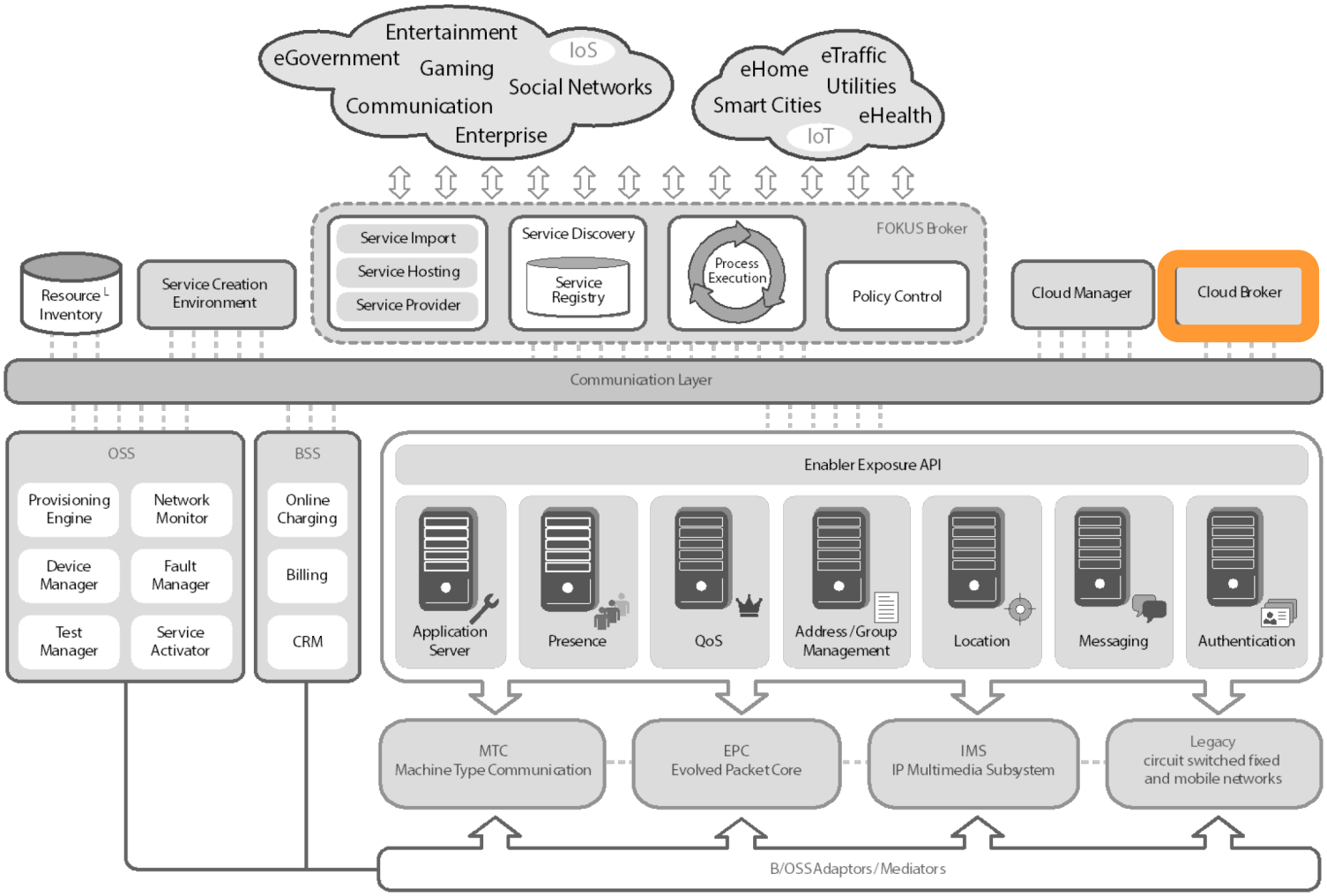

Figure 127: Open SOA Telco Playground in Detail

Apart from the SOA-based integration of the Cloud Broker and Cloud Manager, Figure 127 also shows the other building blocks on which the overall cloud brokering system depends. For coming up with the overall system, described in this work, significant use was made of the FOKUS Broker's policy control components (for realization of the Cloud Broker's policy-based management parts) and process execution components (for realization of the Cloud Broker's SOA-based service lifecycle management and service orchestration). Also the provisioning engine, as depicted as part of the SDP's OSS, plays an important role for provisioning the NGN session control parts. The cross-layer and cross-domain cloud monitoring part, developed in the course of this work, heavily made use of the network monitoring components (also depicted as part of the SDP's OSS).

10.2.3 Research Projects

EU FP7 BonFIRE - Multi-Cloud Experimental Facility

The BonFIRE (Building service testbeds for Future Internet Research and Experimentation) Project [164] designed, built, and operated a large-scale multi-site cloud facility for experimentation and testing purposes of the European Future Internet community. Throughout the lifetime of BonFIRE, and even beyond, researchers and testers from the industry were and still are given access to an experimental facility, which enables large scale experimentation of cloud-based systems and applications, the evaluation of cross-layer effects of converged services and network infrastructures.

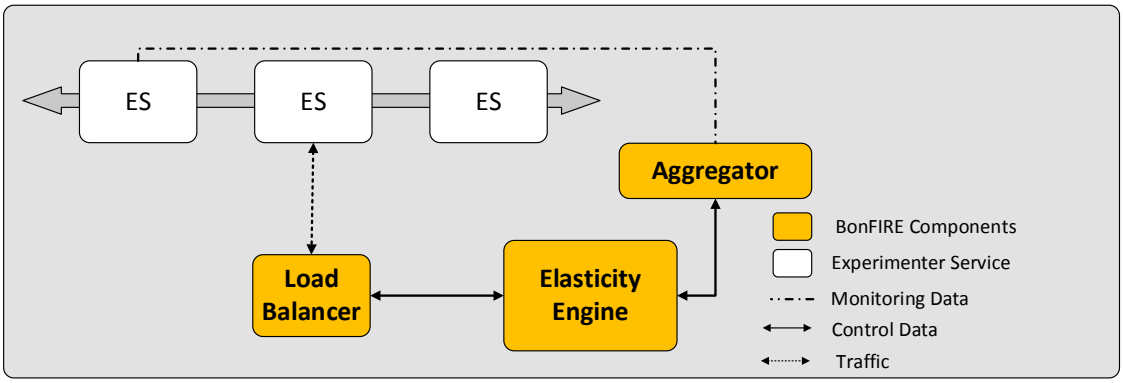

Figure 128: Elasticity in BonFIRE [20]

By utilizing the NGN Cloud Broker's cloud brokering and monitoring mechanisms, as shown in Figure 128 a complex, but easy-to-use Elasticity Engine [20] was implemented and integrated into BonFIRE's multi-cloud facility, which was since providing numerous experimenters and testers with Elasticity as a Service capabilities. Both, by administering BonFIRE's web portal, or through utilization of BonFIRE's API, dynamic and elastic resource scaling can be enabled, trigger-points / thresholds defined and load-balancing mechanisms selected [165].

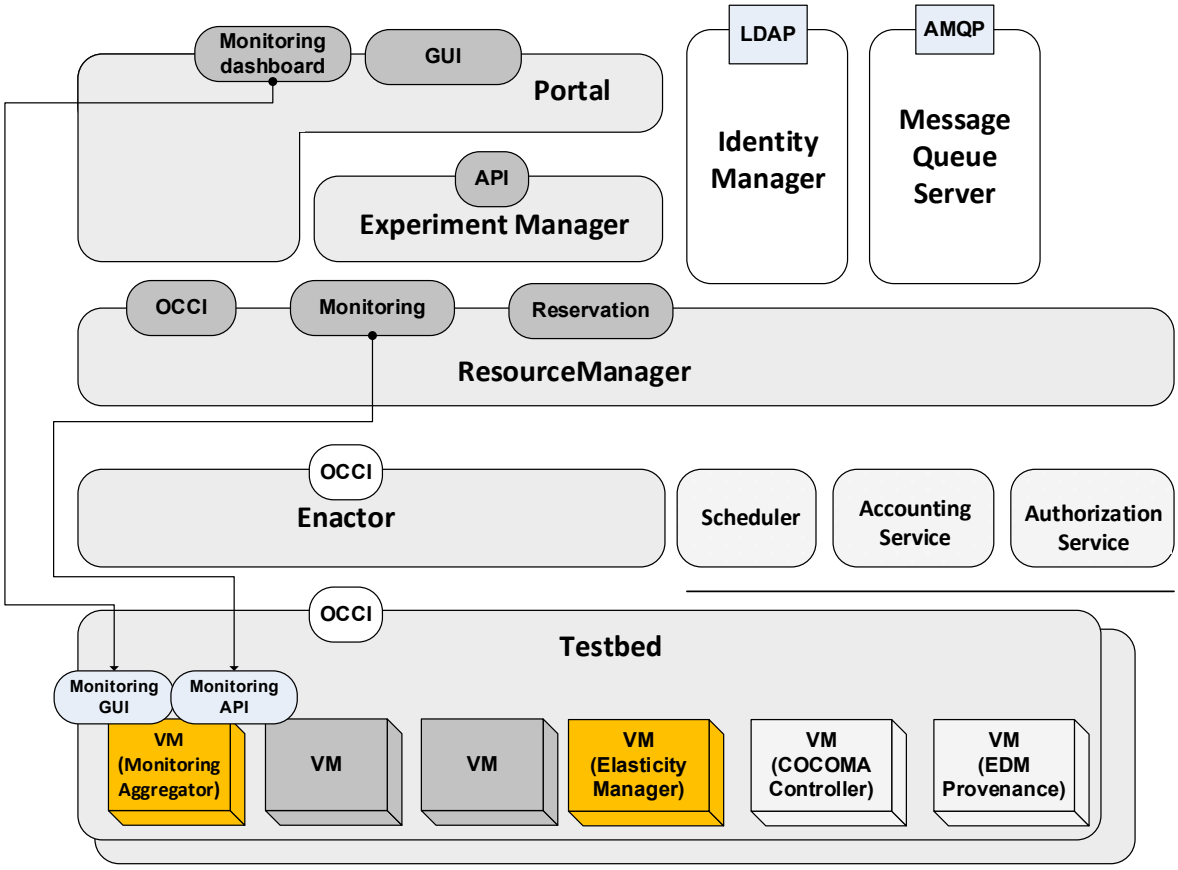

Figure 129: BonFIRE Rel. 4.0.5 Architecture, simplification of [164][166], documented in [165]

As shown in Figure 129, the Elasticity Manager is tightly integrated at the testbed level. Together with the Monitoring Aggregator, elastic resource management / scaling is achieved by instantiating specialized VMs (i.e. elasticity manager VM and monitoring aggregator VM, plus load-balancing VM) within each testbed, and by contextualizing all service-containing VMs so as to send monitoring data to the aggregator for the elasticity engine to balance load, and make informed decisions for elastically scaling resources.

EU FP7 Fed4FIRE - Federation of Multiple, Heterogeneous Facilities

The EU FP7 Federation for Future Internet Research and Experimentation (Fed4FIRE) project is addressing the work programme topic Future Internet Research and Experimentation (FIRE). It develops a common federation framework which is continuously adopted by different experimentation facilities in the area of Future Internet Research. The framework supports powerful experiments and cross-platform/-domain experiment life-cycle management (federated identity management, federated resource discovery, provisioning, control and monitoring) project.

The author is not only working on the realization of the Fed4FIRE architecture and systems for federating multiple heterogeneous facilities (including IMS/EPC-based mobile broadband as well as cloud infrastructures), as describe in [5], but also integrates and federates the cloud-based Future Internet facility, the FUSECO Playground (section 10.2.1)

with the federation of Fed4FIRE testbeds. In Fed4FIRE mechanisms for federating, brokering and monitoring resources across multiple infrastructures are extensively being utilized and exploited and the federation, monitoring and cross-domain experiment/resource control capabilities of the FUSECO Playground are continuously being enhanced.

EU FP7 NUBOMEDIA - Elastic Cloud Platform for Multimedia

The EU FP7 project "*Elastic Platform as a Service (PaaS) Cloud for Interactive Social Multimedia*" (NUBOMEDIA) develops the first cloud platform explicitly designed for hosting interactive multimedia services. The NUBOMEDIA architecture utilized media pipelines, i.e. chains of elements that are providing media capabilities (e.g. encryption, transcoding, augmented reality or video content analysis), which allow building arbitrarily complex media processing for applications. The NUBOMEDIA cloud infrastructure acts like a single virtual super-computer incorporating the available, underlying physical network resources in a cross-cutting fashion. By doing so, NUBOMEDIA applications can elastically be scaled and adapted to the required workload where QoS and SLAs are assured.

In NUBOMEDIA the results of this work are extensively being exploited, for both, the realization of NUBOMEDIA's elastic cloud scaling and monitoring mechanisms, as well as for realization of versatile service orchestration, composition and delivery mechanisms as depicted in Figure 127.

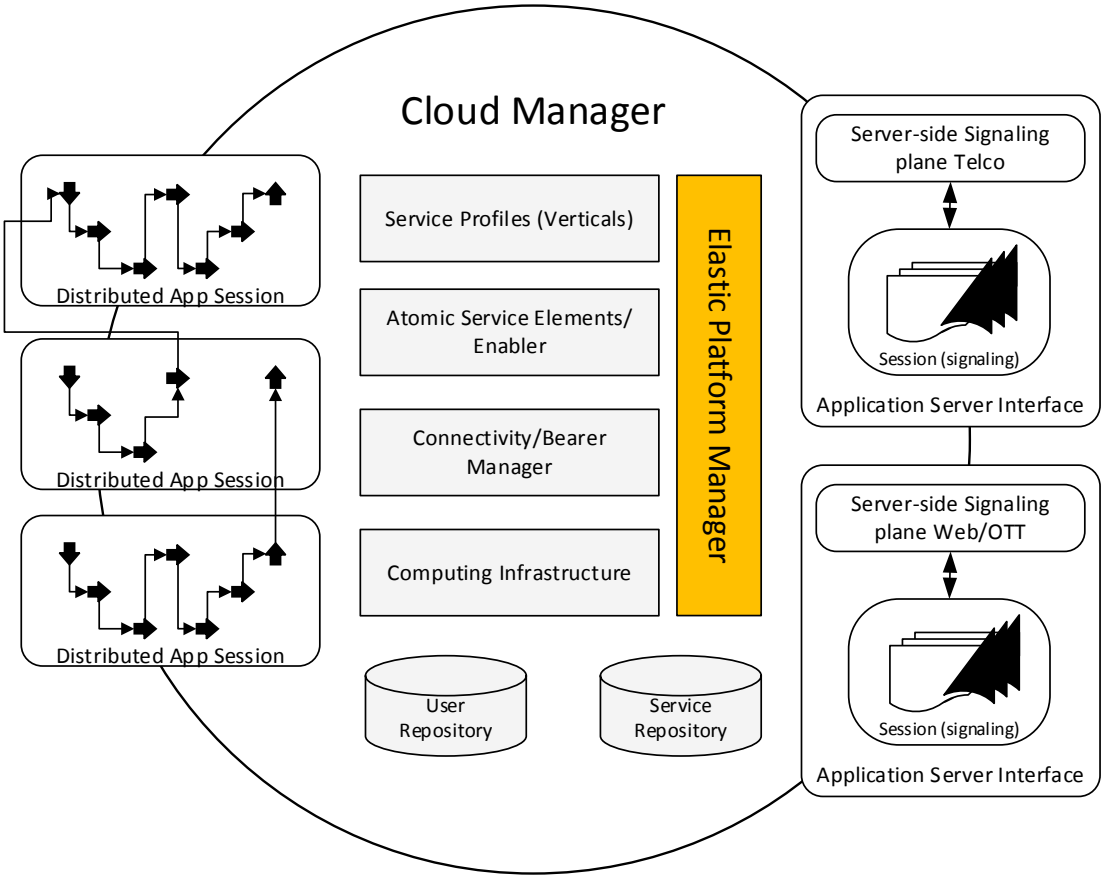

Figure 130: Nubomedia Elastic Cloud Platform

10.2.4 Industry Projects

Nippon Telegraph & Telephone Corporation (NTT) - Elastic Cloud Platform for Feature-based, Real-time Communication Convergence

Based on the results of this thesis, together with partners from NTT, Japan, an elastic service platform for real-time communication services providing mechanisms for dynamic convergence with additional, optional service features was implemented. The platform allows supplementing communication services with images, overlays, maps or textual interactions. Those enriched, interactive services are provided via IMS-based signaling and clients as well as via web-/browser-based WebRTC clients.

A broad range of mechanisms and solutions developed in the context of this work have been exploited in this project for the realization of the targeted elastic service platform. As shown in Figure 131, the architecture of the elastic RTC control comprises, an Elasticity Engine, load balancing functions as well as cloud management functions for the dynamically scaling (in-/decreasing) of RTC resources according to different signaling and media processing workloads. By doing so the entire cloud-based RTC control system is being made horizontally scalable.

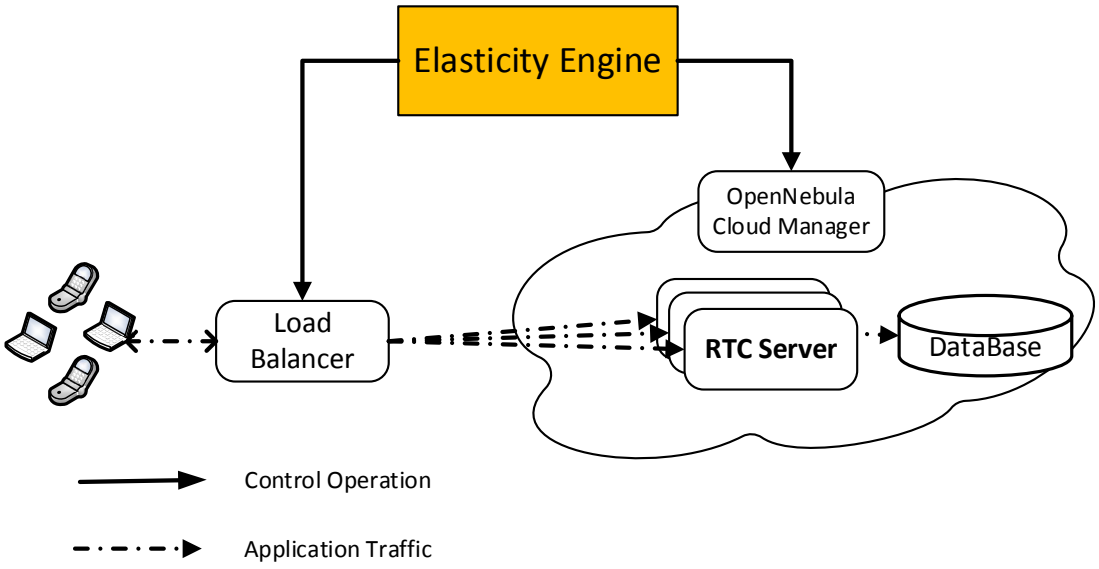

Figure 131: Elasticity Engine in NGN Service Broker Project with NTT

10.2.5 Contributions to Standardization

IEEE P2302 Intercloud Testbed

The IEEE Cloud Computing Standards Committee[22] is fostering the development of technical standards for cloud to cloud interoperability, mainly through the P2301 working group, developing the Guide for Cloud Portability and Interoperability Profiles (CPIP) and the P2302 working group, developing Standards for Intercloud Interoperability and Federation (SIIF). In the context of P2302, the IEEE Intercloud Testbed[23] project was established in order to create a global lab to prove and improve the Intercloud technologies, depicted in Figure 132.

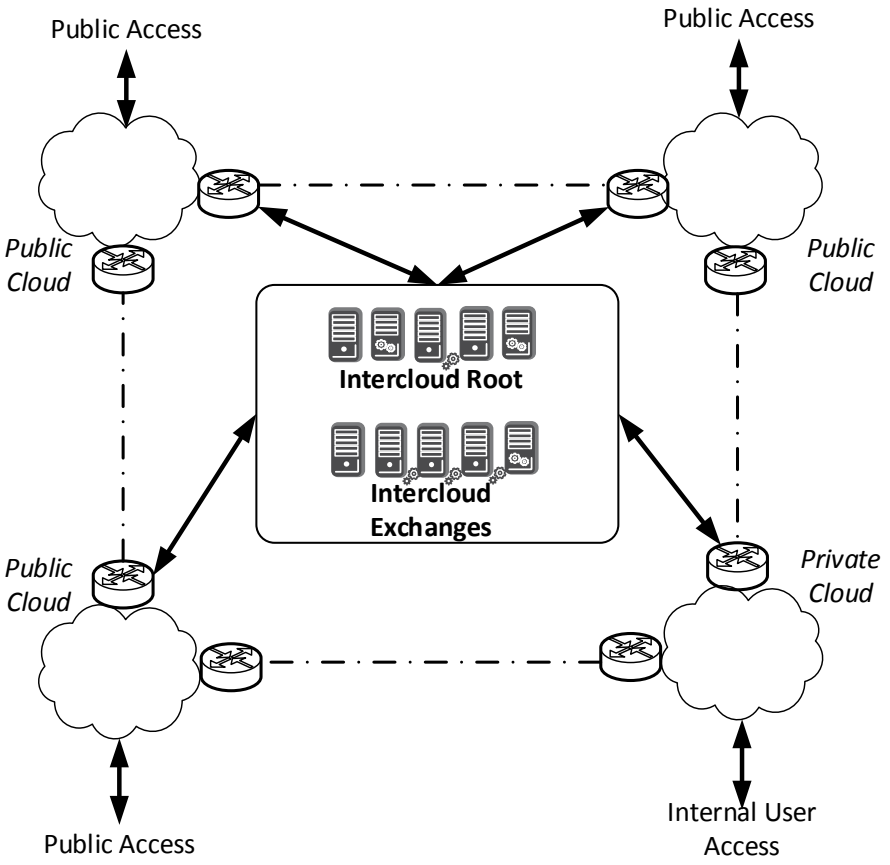

Figure 132: IEEE P2302 Intercloud Reference Network Intercloud Topology [167]

For establishing the global IEEE Intercloud testbed, the IEEE has partnered with several universities and research centers world-wide, already operating and maintaining cloud testbeds and cloud resources. By joining the Intercloud testbed, with FOKUS' cloud testbed resources and systems (see FUSECO Playground and Open SOA Telco Playground) active contributions are being made to the evolution of cloud federation and interoperability. Through the developments and real-world experiments conducted in the context of this work,

[22] IEEE Cloud Computing Web Portal, online: www.cloudcomputing.ieee.org, accessed 21st May 2014

[23] The IEEE Intercloud Testbed, online: www.intercloudtestbed.org, accessed 21st May 2014

contributions to the IEEE Intercloud testbed are made by exploiting the gained expertise in the field of cloud brokerage, elastic multi-cloud resource management and particularly by contributing with the specific focus on cloud-based NGN/telecommunication services and platforms. By contributing to the evolution and by utilizing Intercloud standards (protocols and formats) [43],[114] for federating and interoperating with other Intercloud testbeds contributions are not only made through standard definitions and specifications, but also by developing standard-based open source tools for the community which are tested in the Intercloud real-word, global testbed.

10.2.6 Dissemination

The full list of the author's publications, including books (1), journals / book chapters (7), publications (17), tutorials (4) supervision of Master Theses (4), workshop/conference organization (26), i.e. chair (2), TPC-Co chair (2), TPC Member (6), Reviewer (16) as well as conference presentations (many) can be found in Appendix I:

10.3 Outlook and Future Directions

Since the onset of this work, a broad spectrum of technologies contributed to the rapid evolution of telecommunication infrastructures and services. Under the umbrella term of NGNs, telecommunication core networks, as well as service platforms evolved in multiple dimensions. IT technologies such as IP-based communication technologies, session / service control platforms (IMS, EPC), SOAs, cloud computing, autonomic computing/communications, SDNs, M2M were and are continuously reshaping telecommunication landscape. All these efforts helped increasing the efficiency, scalability and versatility of telecommunication infrastructures and gave birth to a broad spectrum of entirely new communication services. With an ever decreasing time to market and a yet unforeseen range of possible service combinations this evolution is rapidly fertilizing a broad range of usage areas, from eHealth, to eTransport, to eMobiliy, to eLearning, etc.

The contributions of this work, fit into the currently ongoing ICT research, vaguely subsumed under the umbrella term "*Future Internet*" or the "Internet of Everything", the "Autonomic Internet", where *Virtualization* technologies are continuously applied at all levels and layers. The evolution of the Future Internet evolution is frequently defined by its three pillars, the *Network of the Future*, the *Internet of Things* and the *Internet of Services*.

Through *Software Defined Networking* (SDN) technologies, *Network Function Virtualization* (NFV) the cornerstone technologies for the *Network of the Future* networks are increasingly becoming smarter, versatile, self-organized, autonomic and robust, and most importantly adaptive to the needs of services.

The *Internet of Things* continuously connects ever more devices, enriching in-house, city-wide, nation-wide environments with ever more intelligent sensors, actuators and robots,

generating exponentially increasing amounts of data, aggregated and provided to multiple tenants through data clouds.

The *Internet of Services*, together with the "*Semantic Internet*", continuously enables rapid cross-cutting (XaaS) and cross-domain compositions of services and business processes. Networks, infrastructures, platforms and software are ever more flexibly, freely compose-ably and dynamically orchestrate-ably provided *as a service*. Complex business processes, and service compositions, spanning across several administrative domains, composed of multiple service building blocks / service enablers, which are provided by multiple service providers, are continuously enriching a broad range of application domains (public, governmental, private) in all different kinds of usage areas like eHealth, eTransport, eMobility, eGovernment or eTourism enabling ever smarter cities (enterprises, homes, regions, countries) and augmented realities.

The core mechanisms of *Resource Efficient Quality of Service Management in Federated Cloud Environments,* although having emerged out of the Internet of Services research field, more specifically the Grid and Infrastructure as a Service research fields find their applications in all of the above mentioned Future Internet areas. And this is not only because cost-efficient computation as a utility will always be needed in all sectors and areas. With virtualization techniques being applied to radio/spectrum resources, network resources, infrastructure resources, etc. mechanisms that are providing adaptive "*Resource Efficiency*" complying with specific constraints will be needed.

On the one hand, more and more historically physical infrastructure-dependent applications and service platforms, e.g. the IMS, the EPC, M2M platforms (as the OpenMTC), are "cloudified", i.e. fully virtualized and enabled for cloud deployment. This "cloudification", if done correctly, will always involve elastic resource allocation tailored to the needs (i.e. capacity and QoS) of each particular service platform and application. IMS-as-a-service, EPC-as-a-service, M2M-as-a-service providers (to stay within these telecommunication/NGN centric examples), simply by economics of scale (best value for money) will more and more seek for vendor-independent, multi-cloud platform solutions. Competition between cloud infrastructure providers is rising; buying and operating own resources will continuously become less attractive to individual service providers (at least to those, not entering the cloud infrastructure market themselves).

On the other hand, mechanisms for network virtualization such as SDN and NFV are quickly evolving. They are not only allowing network operators and network providers to provide their resources more efficiently or users to use resources more efficiently. Networks and networking functions (network QoS-, security-, reliability-related) provided as-a-service, are allowing for the dynamic selection and composition of network functions based on the particular needs of specific services. Whereas to a lesser extend relevant for traditional best-effort Web services, mission critical services, services requiring high degrees of security or service in industry automation, health, surveillance or transport requiring real-time responsiveness will increasingly exploit NFV/SDN capabilities. NaaS combined with elastic

and QoS assured IaaS solutions will be able to seamlessly satisfy the needs of future applications, proving functions, capacities and scalability on-demand.

To these ends, the next evolutionary steps clearly lead to:

- Full integration of adaptive feedback control into the NGN Cloud Broker
- Full integration and interworking of IaaS with NaaS (multi-domain SDN and NFV) technologies
- Full interworking and dynamic composition of NaaS, IaaS, PaaS, SaaS (XaaS)
- Globally available inter-cloud technologies, the emergence of a cloud platform provider ecosystem, where providers dynamically and on-demand exchange and interwork with cloud resources (the "cloud of clouds")

This work has shown the cumbersome efforts that were needed for benchmarking and profiling, not only cloud platforms and their resources, but also NGN services. The evolution of the NGN Cloud Broker is clearly driven by the need for increased automation. To this end, the author is currently investigating and developing further control theoretic approaches for further enhancing the NGN Cloud Broker. Self-adaptive and self-learning mechanisms, which provide enhanced feedback about the current QoS, resource utilization and cloud platform performance, are being incorporated into the NGN Cloud Broker. The author expects significant reduction of benchmarking efforts and an increase of capacity saving as well as QoS assurance capabilities, as well as improvements in terms of robustness and fault tolerance. By doing so also the lifecycle/time-to-market will be shortened, and a broader spectrum of services will be supported.

At Fraunhofer FOKUS, the author is currently furthermore concentrating on prototyping elastic scaling mechanisms for SDNs (as outlined in section 10.2.1) which will soon be released as part of the OpenSDNCore's second release. By applying the QOSMUC framework, developed in the context of this work, and by utilization of the developments of the NGN Cloud Broker, the OpenSDNCore Rel. 2 will soon be capable of dynamically scaling network resources, in a fully policy-based fashion (where resource allocation can be triggered based on user-, service-, platform-profiles and related policies).

The author's Future Internet team at Fraunhofer FOKUS, together with the Integrated Service Architecture team is continuously enhancing NGN service platforms with elastic resource allocation and cloud monitoring functions. NaaS capabilities soon provided by the OpenSDNCore, together with the developed RTC-as-a-service, IMS-as-a-service, EPC-as-a-service, and M2M-as-a-service capabilities are continuously exploited by Fraunhofer FOKUS's service-platform-oriented teams. The cloud management, cloud federation and brokering, as well as the cloud monitoring expertise and developed solutions are continuously utilized in industry projects (e.g. those mentioned in section 10.2.4 and beyond) as well as in international research projects (e.g. those mentioned in section 10.2.3 and more under the Horizon 2020 umbrella of the European Commission), giving birth, not only to innovative

telecommunication services, but also to eHealth, eMobility, IT4Energy, and eGovernment service prototypes and solutions.

Internationally regarded as telecommunication cloud experts, the authorÄs Future Internet team engages into a broad range of further contributions in the field of cloud federation and cloud interoperability standardization for communication services (e.g. those mentioned in section 10.2.5, but soon also the TMForum and ETSI).

Acronyms

AHP – Analytical Hierarchy Process
API – Application Programming Interface
ARIMA - Auto Regressive Integrated Moving Average
ARMA – Autoregressive Moving Average
ATIS - Alliance for Telecommunications Industry Solutions
AWS – Amazon Web Service
BCM – Business Capacity Management
BPEL – Business Process Execution Language
BPF – TM Forum Business Process Framework
BSS – Business Support Systems
CCIF - Cloud Computing Interoperability Forum
CCM – Component Capacity Management
CDMI – Cloud Data Management Interface
CIM – Common Information Model (DMTF)
CIMI - Cloud Infrastructure Management Interface (DMTF)
CMDBf – Configuration Management Database (federated)
CMWG –DMTF Cloud Management Working Group
CORBA – Common Object Request Broker Architecture
CPE – Customer Premises Equipment
CPIP - Cloud Portability and Interoperability Profiles
CPU – Central Processing Unit
CRUD – Create, Delete, Update and Delete operations
CSA – Cloud Security Alliance
CSCC - Cloud Standards Customer Council
CSCF – Call State Control Function
CSE – Capacity Saving Efficiency
DMD – Deployment Model Descriptor
DMTF – Distributed Management Task Force
DNS – Domain Name Service
DSL – Direct Subscriber Line
DSML - Domain-Specific Modeling Languages
DTMF – Desktop Management Task Force
ECLC –TM Forum Enterprise Cloud Leadership Council
ECU – Elastic Compute Units (Amazon's measure of CPU resources)
ED – Experiment Descriptor
EE – Elasticity Engine
ELECTRE - ELimination Et Choix Traduisant la REalité
EMS – Elasticity Management System
ENISA - European Network and Information Security Agency
EPC – Evolved Packet Core
ETSI - European Telecommunications Standards Institute
FAB – Fulfillment, Assurance and Billing management areas

FCAPS - Fault, Configuration, Accounting, Performance, Security management areas
FIRE – Future Internet Research and Experimentation
FMC – Fixed-Mobile Convergence
FOKUS – Fraunhofer Institute for Open Communication Systems
FUSECO – Future Seamless Communication
GICTF - Global Inter-Cloud Technology Forum
GSM - Global System for Mobile Communications
GSMA – GSM Association
HSS – Home Subscriber Server
HTTP – Hypertext Transfer Protocol
IaaS – Infrastructure as a Service
IBM - International Business Machines Corporation
ICT - Information and Communications Technologies
IEC - International Electrotechnical Commission
IEEE - Institute of Electrical and Electronics Engineers
IETF – Internet Engineering Task Force
IF – TM Forum Information Framework
IMS – IP Multimedia Subsystem
IPPM – IP Performance Metrics
IPTV – Internet Protocol Television
ISDN - Integrated Services Digital Network
ISO – International Organization for Standardization
ITIL – Information Technology Infrastructure Library
ITU - International Telecommunication Union
ITU-T – International Telecommunication Union, Telecommunication Standardization
JOSIF – Joint Open Source Interface Framework
JSON –JavaScript Object Notation
KPI – Key Performance Indicator
KQI – Key Quality Indicator
KVM - Kernel-based Virtual Machine
LTE – Long-Term Evolution
MANO – NFV Management and Orchestration Architecture (ETSI)
MAPE – Monitor-Analyze-Plan-Execute
MAUT – Multi Attribute Utility Theory
MCDA – Multi Criteria Decision Analysis
MCDM – Multi Criteria Decision Making
MDA – Model Driven Architecture
MDE – Model Driven Engineering
MOS – Mean Opinion Score
MTNM – Multi-Technology Network Management
MTOSI –Multi-Technology Operations Systems Interface
NaaS – Network as a Service
NAT – Network Address Translation
NCB – NGN Cloud Broker
NFV – Network Function Virtualization

NGN – Next Generation Network
NGNI – Next Generation Network Infrastructures
NGOSS – New Generation Operations Systems and Software
NGSDP – Next Generation Service Delivery Platform
NIST –National Institute of Standards and Technology
NSCL - Network Service Capability Layer
NTT – Nippon Telegraph and Telephone
OASIS - Organization for the Advancement of Structured Information Standards
OCA – Open Component Architecture
OCC – Open Cloud Consortium
OCCI –Open Cloud Computing Interface
ODCA –Open Data Center Alliance
OGF – Open Grid Forum
OMA – Open Mobile Alliance
OMACO – Open IMS Management Console
OMG – Object Management Group
OPEX – Operational Expenditure
OPF – Overprovisioning Factor
OSE – OMA Service Environment
OSG – Open Systems Group
OSI – Open Systems Interconnection
OSIMS – Open Source IMS Core
OSS – Operations Support Systems
OTT – Over-the-Top
OVF – Open Virtualization Format
PaaS – Platform as a Service
PDCA – Plan-Do-Check-Act Cycle
PDP – Policy Decision Point
PEEM – Policy Evaluation Enforcement Management
PEP – Policy Evaluation Point
PESQ - Perceptual Evaluation of Speech Quality
PEVQ - Perceptual Evaluation of Video Quality
PPP – Public Private Partnership
PROMETHEE - Preference Ranking Organisation Method for Enrichment Evaluations
PSO – Provisioning Service Objects (SPML)
PSP – Provisioning Service Provider (SPML)
PST – Provisioning Service Target (SPML)
PSTN – Public Switched Telephone Network
QoE – Quality of Experience
QoS – Quality of Service
QOSMUC – QoS-aware Multi-Cloud Brokering Framework for NGN Services
RAC – Resource and Admission Control
RAM – Random-Accesss Memory
RCM – Resource Capacity Management
RCS – Rich Communication Service

REST –Representation State Transfer
ROM – Read only Memory
RPC – Remote Procedure Call
RRD – Round-Robin Database
RTC – Real-Time Communication
RTP – Real-Time Transport Protocol
SaaS – Software as a Service
SAJACC – NIST group on Standards Acceleration to Jumpstart Adoption of Cloud
SCM – Service Capacity Management
SDF – Service Delivery Framework
SDN – Software Defined Networking
SDO –Standards Development Organization
SDP – Service Delivery Platform
SEMS - SIP Express Media Server
SENAI - Serviço Nacional de Aprendizagem Industrial
SES – Software Enabled Services team (TM Forum)
SID – Shared Information and Data Model (TM Forum)
SIGEVO – Special Interest Group on Genetic and Evolutionary Computation (ACM)
SIGMETRICS – Special Interest Group on Measurement and Evaluation (ACM)
SIIF - Standards for Intercloud Interoperability and Federation (IEEE)
SIM - Subscriber Identity Module
SIMPLE - SIP for Instant Messaging and Presence Leveraging Extensions
SIP – Session Initiation Protocol
SLA –Service Level Agreement
SLM – Service Level Management (ITIL)
SLO – Service Level Objective
SMO - Systems Management Overview
SNIA - Storage Networking Industry Association
SNIA –Storage Networking Industry Association
SNMP – Simple Network Management Protocol
SOA –Service Oriented Architecture
SOAP – Simple Object Access Protocol
SPLC – TM Forum Service Provider Leadership Council
SPML – Service Provisioning Markup Language
SQA – Service Quality Analysis
SQM – Service Quality Management
TAM – Telecom Application Map
TCP – Transmission Control Protocol
TIC – Total Infrastructure Costs
TIP – TM Forum Integration Program
TISPAN - TIPHON (Telecommunications and Internet Protocol Harmonization over Networks) and SPAN (Services and Protocols for Advanced Networks)
TMF – TeleManagement Forum / TM Forum
TMN – Telecommunications Management Network
TNA – Technology Neutral Architecture

TOGAF - The Open Group Architecture Framework
TOSCA – Topology and Orchestration Specification for Cloud Applications
TPC – Technical Program Committee
UDDI - Universal Description, Discovery and Integration
UDP – User Datagram Protocol
URI – Uniform Resource Identifier
URL - Uniform Resource Locator
VA – Virtual Appliance
VEEM – Virtual Execution Environment Manager
VM – Virtual Machine
VMI – Virtual Machine Image
VPN – Virtual Private Network
XML –Extensible Markup Language
XSD – XML Schema Definition

Bibliography

[1] N. Blum, T. Magedanz, and F. Schreiner, "Services, enablers and architectures: Definition of a connected web 2.0/telco service broker to enable new flexible service exposure models," *Proc. of International Conference on Intelligence in Networks (ICIN), Bordeaux, France*. Bordeaux, France, pp. 20–23, 2008, ISBN 978-1-60702-868-0.

[2] N. Blum, T. Magedanz, F. Schreiner, and S. Wahle, "Service oriented testbed infrastructures: a cross-layer approach for NGNs," in *Mobile Networks and Applications*, 2010, vol. 15, no. 3, pp. 413–424, ISSN 1383-469X (Print) 1572-8153.

[3] N. Blum, T. Magedanz, F. Schreiner, and S. Wahle, "A research infrastructure for SOA-based Service Delivery Frameworks," *5th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops*. Washington, DC, USA, 2009, ISBN: 978-1-4244-2847-2, IEEE CN: CFP09364.

[4] F. Schreiner, S. Wahle, N. Blum, and T. Magedanz, "Modular exposure of next generation network services to enterprises and testbed federations," *Second International Conference on Communications and Electronics*. Hoi An City, Vietnam, pp. 98–103, 2008, ISBN 978-1-4244-2425-2, IEEE CN CFP0816B-PRT.

[5] W. Vandenberghe, B. Vermeulen, P. Demeester, A. Willner, S. Papavassiliou, A. Gavras, A. Quereilhac, Y. Al-hazmi, F. Lobillo, C. Velayos, A. Vico-oton, and G. Androulidakis, "Architecture for the Heterogeneous Federation of Future Internet Experimentation Facilities," *IEEE Future Network and Mobile Summit (FutureNetworkSummit)*. IEEE, Lisbon, Portugal, 2013, ISBN: 978-1-905824-37-3.

[6] T. Magedanz, J. a. Lozano, F. Schreiner, F. Gouveia, and J. M. Gonzalez, "Towards Autonomic Communication Mechanisms for Service Composability Management," *Fifth IEEE Workshop on Engineering of Autonomic and Autonomous Systems. EASE 2008.*, pp. 197–203, Mar. 2008, ISBN: 978-0-7695-3140-3.

[7] N. Blum, J. Müller, F. Schreiner, and T. Magedanz, "Telecom Applications, APIs and Service Platforms," *Evolution of Telecommunication Services*. Springer, pp. 25–46, 2013, ISBN 978-3-642-41568-5.

[8] N. Blum, T. Magedanz, and F. Schreier, "The role of service brokers for composed services in an open service environment," *Tele-Kommunikation aktuell*, vol. 1, no. November 2008. Erlangen, Germany, 2008.

[9] N. Blum, T. Magedanz, and F. Schreiner, "Definition of a service delivery platform for service exposure and service orchestration in next generation networks," *Ubiquitous Computing and Communication (UbiCC) Journal*, vol. 3, no. 3. Sherbrooke QC, Canada, pp. 102–111, 2008, ISSN: 1994-4608.

[10] N. Blum, P. Jacak, F. Schreiner, D. Vingarzan, and P. Weik, "Towards Standardized and Automated Fault Management and Service Provisioning for NGNs," *Journal of*

Network and Systems Management, vol. 16, no. 1. Springer, pp. 63–91, 22-Dec-2008, ISSN: 1064-7570 (Print) 1573-7705 (Online).

[11] F. Dinu, P. Jacak, N. Blum, F. Schreiner, and T. Magedanz, "Automated service provisioning and fault management for OSIMS based NGNs," *Proceedings of the 14th Annual Workshop of HP Software University Association*. Stuttgard, Germany, pp. 271–276, 2007, ISBN-13: 978-3-00-021690-9.

[12] B. V. Harjoc, T. Magedanz, and F. Schreiner, "Automated Root Cause Analysis for IMS and SDP," *2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE, Cardiff, UK, pp. 27–32, Sep-2009, ISBN: 978-0-7695-3786-3.

[13] B. Harjoc, F. Schreiner, D. Vingarzan, and T. Magedanz, "Automated fault localization based on unified Web service and NGN benchmarking," *2009 IEEE Symposium on Computers and Communications*. Sousse, Tunisia, pp. 77–82, 2009, ISSN: 1530-1346, ISBN: 978-1-4244-4672-8.

[14] N. Blum, T. Magedanz, and F. Schreiner, "Management of SOA based NGN service exposure, service discovery and service composition," *2009 IFIP/IEEE International Symposium on Integrated Network Management*. Long Island, NY, pp. 430–437, 2009, ISBN 978-1-4244-3487-9.

[15] T. Magedanz, F. Schreiner, and S. Wahle, "From NGN to Future Internet Testbed Management – Collaborative Testbeds as Enabler Layer and Cross-Domain Communication and Network," *Tele-Kommunikation aktuell*, vol. 5–6, no. TKA 62 (2008). Erlangen, Germany, pp. pp.1–20, 2008, ISSN 1619-2036.

[16] N. Blum, T. Magedanz, F. Schreiner, and S. Wahle, "From IMS Management to SOA Based NGN Management," *Journal of Network and Systems Management*, Issue 17(1-2), June 2009, pp. 33-52, ISSN: 1064-7570 (Print) 1573-7705

[17] T. Magedanz, F. Schreiner, and S. Wahle, "Service-oriented testbed infrastructures and cross-domain federation for Future Internet research," *2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops*. Long Island, NY, pp. 101–106, 2009, ISBN: 978-1-4244-3923-2.

[18] P. Bellavista, G. Carella, L. Foschini, T. Magedanz, F. Schreiner, and K. Campowsky, "QoS-aware elastic cloud brokering for IMS infrastructures," *IEEE Symposium on Computers and Communications (ISCC)*. IEEE, Cappadocia, Turkey, pp. 157–160, 2012, ISBN: 9781467327121, ISSN: 15301346.

[19] G. Carella, T. Magedanz, K. Campowsky, and F. Schreiner, "Network-aware Cloud Brokerage for telecommunication services," *IEEE 1st International Conference on Cloud Networking (CLOUDNET)*. Paris, France, pp. 131–136, 2012, ISBN: 978-1-4673-2798-5.

[20] G. Carella, F. Schreiner, K. Campowsky, and T. Magedanz, "Elasticity as a Service for Federated Cloud Testbeds," *IEEE International Conference on Communications (ICC), Second Workshop on Clouds, Networks and Data Centers*. Budapest, Hungary, pp. 256 – 260, 2013, ISBN: 978-1-4673-5753-1.

[21] K. Campowsky, G. Carella, T. Magedanz, and F. Schreiner, “Optimization of Elastic Cloud Brokerage Mechanisms for Future Telecommunication Service Environments,” *Praxis der Informationsverarbeitung und Kommunikation*, vol. 35, no. 3, pp. 131–136, 2012, ISSN (Online) 1865-8342, ISSN (Print) 0930-5157.

[22] F. Schreiner and T. Magedanz, “QoS-aware Multi-Cloud Brokering for NGN Services,” *The 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE 2014)*. Da Nang, Vietnam, 2014, pp. 157-160, ISBN: 978-1-4799-5049-2.

[23] L. Schubert, K. Jeffery, and B. Neidecker-Lutz, “The Future of Cloud Computing: Opportunities for European Cloud Computing Beyond 2010, European Commission Expert Group Report,” *The Future of Cloud Computing*, pp. 1–71, 2010.

[24] P. Mell and T. Grance, “The NIST Definition of Cloud Computing-Recommendations of the National Institute of Standards and Technology. NIST,” *NIST Special Publication*. National Institute of Standards and Technology, 2011.

[25] L. Schubert, K. Jeffery, and B. Neidecker-Lutz, “A Roadmap for Advanced Cloud Technologies under H2020,” *Recommendations by the Cloud Expert Group*, no. December. Publications Office of the European Union, Luxembourg,, 2012.

[26] L. Schubert and K. Jeffery, “Advances in Clouds Research in Future Cloud Computing,” *Expert Group Report, Public version*, vol. 1.0. Publications Office of the European Union, Luxembourg, 2012.

[27] ITU-T FG Cloud TR_2012, “Technical Report: Part 7: Cloud computing benefits from telecommunication and ICT perspectives,” International Telecommunications Union, 2012.

[28] ITU-T FG Cloud TR_2012, “Technical Report: Part 4: Cloud Resource Management Gap Analysis,” International Telecommunications Union, 2012.

[29] ITU-T, “Recommendation Y.3520 Cloud computing framework for end-to-end resource management.” International Telecommunications Union, 2013.

[30] T. Rings, G. Caryer, J. Gallop, J. Grabowski, T. Kovacikova, S. Schulz, and I. Stokes-Rees, “Grid and Cloud Computing: Opportunities for Integration with the Next Generation Network,” *Journal of Grid Computing*, vol. 7, no. 3. Springer, pp. 375–393, 28-Aug-2009.

[31] P. Bosch, A. Duminuco, F. Pianese, and T. L. Wood, “Telco clouds and Virtual Telco: Consolidation, convergence, and beyond,” *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. Ieee, Dublin, Ireland, pp. 982–988, 2011.

[32] ETSI FC Cloud-Technical Report, “Initial analysis of standardization requirements for Cloud services,” vol. 1. European Telecommunications Standards Institute, pp. 1–13, 2010.

[33] E. Federico, “The Economic Impact of Cloud Computing on Business Creation, Employment and Output in Europe, An application of the Endogenous Market Structures,” *Review of Business and Economics (2009)*, vol. 54. Leuven, Belgium, pp. 179–208, 2009.

[34] Merriam-Webster, *Merriam-Webster’s Collegiate Dictionary, 11th Edition thumb-notched with Win/Mac CD-ROM and Online Subscription*. Merriam-Webster, 2003.

[35] C. Grönroos, “Service management and marketing: a customer relationship management approach.” John Wiley & Sons Incorporated, Hoboken, NJ, 2000.

[36] M. Iqbal and M. Nieves, “Service strategy : ITIL,” *TSO, London*, vol. 1191. TSO (The Stationery Office), Norwich, pp. 3252–3252, 2007.

[37] TMF GB921, “Business Process Framwork (eTOM) Enhanced Telecom Operations Map Concepts and Principies/Version 8. 1.” TeleManagement Forum, 2008.

[38] TMF GB917, “SLA Management Handbook,” vol. 4. TeleManagement Forum, 2005.

[39] ITU-T, “Recommendation E.800. Quality of Telecommunication Services: Concepts, Models, Objectives and Dependability Planning. Terms and Definitions Related to the Quality of Telecommunication Services,” International Telecommunication Union, 2008.

[40] ITU-T, “Recommendation Y.1541. Network Performance Objectives for IP-Based Services.” International Telecommunication Union, 2003.

[41] ITU-T, *Recommendation I.350: Integrated Services Digital Network (ISDN) Overall Network Aspects and Functions : General Aspects of Quality of Service and Network Performance in Digital Networks, Including ISDNs*. International Telecommunication Union, 1993.

[42] Wikipedia, “Federation (information technology) --- Wikipedia, The Free Encyclopedia,” 2013. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Federation_(information_technology)&oldid=545524193. [Accessed: 11-Feb-2014].

[43] N. Grozev and R. Buyya, “Inter-Cloud architectures and application brokering: taxonomy and survey,” *Software: Practice and Experience*. John Wiley & Sons, Hoboken, NJ, pp. 1–22, 2012.

[44] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, “NIST cloud computing reference architecture,” *NIST Special Publication*, vol. 500. p. 292, 2011.

[45] ITU-T, “Recommendation M.3060/Y.2401. Principles for the Management of Next Generation Networks.” International Telecommunication Union, 2006.

[46] ITU-T, “Recommendation Y.2234: Open service environment capabilities for NGN.” International Telecommunication Union, 2008.

[47] TMF, “GB921 eTOM Solution Suite Release 7.0,” TeleManagement Forum, 2007.

[48] R. Wieringa, “Requirements engineering: frameworks for understanding.” John Wiley & Sons, Hoboken, NJ, 1996.

[49] R. Wieringa, “Design science as nested problem solving,” *Proceedings of the 4th international conference on design science research in information systems and technology*. Philadelphia, PA, p. 8, 2009.

[50] A. Hevner, S. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, vol. 28, no. 1. University of Minnesota, Minnesota, pp. 75–105, 2004.

[51] T. Magedanz, F. Schreiner, and P. Weik, “Das IP Multimedia Subsystem,” *Arnold, F.: Handbuch der Telekommunikation. Loseblattausgabe Köln: Deutscher Wirtschaftsdienst*. p. Kapitel 12.2.5 (124. Ergänzungslieferung), 2007, ISBN: 978-3871560965.

[52] N. Blum, F. Schreiner, and T. Magedanz, “Emerging Web and Telecom Services : Prototyping FMC Services based on IMS and Web 2 . 0 for a Mobile Operator,” *11th International Conference on Intelligence in Service Delivery Networks, ICIN*. Bordeaux, France, 2007.

[53] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz, and B. A. Hamilton, “Reference model for service oriented architecture 1.0,” *OASIS Standard*, vol. 12. pp. 1–31, 2006.

[54] D. K. Barry, “Web services and service-oriented architectures: the savvy manager’s guide.” Morgan Kaufmann, San Francisco, CA, 2003.

[55] T. Erl, “SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl),” vol. 1. Prentice Hall Upper Saddle River, 2008.

[56] J. Follows and D. Straeten, “Application driven networking: Concepts and architecture for policy-based systems.” IBM Corporation, 1999.

[57] T. Phan and J. Han, “A survey of policy-based management approaches for service oriented systems,” *19th Australian Conference on Software Engineering, 2008. ASWEC 2008.* Sidney, Australia, pp. 392–401, 2008.

[58] D. Verma, “Simplifying network administration using policy-based management,” *IEEE Network: The Magazine of Global Internetworking*, vol. 16, no. April. IEEE Press, Piscataway, NJ, pp. 20–26, 2002.

[59] A. Westerinen, L. Rafalow, and R. Moore, “Policy Framework Architecture,” *IETF Network WG, Internet Draft*, 1999.

[60] J. Strassner, E. Ellesson, and B. Moore, “Policy framework core information model,” *IETF Policy WG, Internet Draft*, 1999.

[61] W. Bumpus, J. W. Sweitzer, P. Thompson, A. R. Westerinen, and R. C. Williams, *Common Information Model: Implementing the Object Model for Enterprise Management*. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[62] N. Damianou, “A policy framework for management of distributed systems,” *PhD Thesis, Imperial College*. London, UK, 2002.

[63] J. Lobo, R. Bhatia, and S. Naqvi, “A policy description language,” *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, AAAI ’99*. Orlando, FL, pp. 291–298, 1999.

[64] A. Uszok, J. M. Bradshaw, M. Johnson, R. Jeffers, A. Tate, J. Dalton, and S. Aitken, “KAoS policy management for semantic web services,” *Intelligent Systems, IEEE*, vol. 19, no. 4. IEEE, Los Alamitos, CA, pp. 32–41, 2004.

[65] A. S. Vedamuthu, D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, and U. Yalçinalp, “Web services policy 1.5-framework,” *W3C Recommendation*, vol. 4, pp. 1–41, 2007.

[66] TMF, “GB922 Shared Information / Data (SID) Model - Policies.” TeleManagement Forum, 2011.

[67] OMA, “PEEM Policy Expression Language Technical Specification.” Open Mobile Alliance, pp. 1–27, 2012.

[68] OSOA, “Service Component Architecture - Client and Implementation Model Specification for WS-BPEL.” Open Service Oriented Architecture Collaboration, 2007.

[69] J. Kephart and D. Chess, “The vision of autonomic computing,” *Computer*, vol. 36, no. 1. IEEE, Hawthorne, NY, pp. 41–50, 2003.

[70] A. Ganek and T. Corbi, “The dawning of the autonomic computing era,” *IBM systems Journal*, vol. 42, no. 1, pp. 5–18, 2003.

[71] IBM Group and others, “An architectural blueprint for autonomic computing,” *IBM White Paper*, 2006.

[72] D. Cannon and D. Wheeldon, *ITIL -- Service Operation*. TSO, 2007.

[73] D. Clifford, *ISO/IEC 20000*. IT Governance Pub., 2010.

[74] O. F. Rana and C. Germain-Renaud, “The convergence of clouds, grids, and autonomics,” *IEEE Internet Computing*, vol. 13, no. 6. IEEE Computer Society, Los Alamitos, CA, p. 9, 2009.

[75] R. Buyya, R. Calheiros, and X. Li, “Autonomic cloud computing: Open challenges and architectural elements,” *Third International Conference on Emerging Applications of Information Technology*. Kolkata, India, pp. 3–10, 2012.

[76] E. Casalicchio and L. Silvestri, “Architectures for autonomic service management in cloud-based systems,” *2011 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, Kerkyra, Greece, pp. 161–166, 2011.

[77] V. C. Emeakaroha, M. a. S. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. a. F. De Rose, "Towards autonomic detection of SLA violations in Cloud infrastructures," *Future Generation Computer Systems*, vol. 28, no. 7, pp. 1017–1029, Jul. 2012.

[78] ITU-T, "Y. 2001: General overview of NGN." International Telecommunication Union, 2004.

[79] ITU-T, "Y. 2011: General principles and general reference model for Next Generation Networks." International Telecommunication Union, 2004.

[80] ETSI TISPAN, "188 001 NGN management; OSS Architecture Release 1," vol. 1. European Telecommunications Standards Institute, pp. 1–28, 2005.

[81] ETSI GRID, "TR 102 767; Grid Services and Telecom Networks; Architectural Options." European Telecommunications Standards Institute, 2009.

[82] ITU-T, "Y.2111 Global Information Infrastructure, Internet Protocol: Aspects and Next Generation Networks: Next - Next Generation Networks – Quality of Service and performance - Resource and admission control functions in next generation networks." International Telecommunication Union, 2008.

[83] 3GPP, "TS 23.228, IP Multimedia Subsystem, Stage 2." Third Generation Partnership Project, 2006.

[84] J. Rosenberg, H. Schulzrinne, and G. Camarillo, "RFC 3261 SIP: session initiation protocol," Internet Engineering Task Force, 2002.

[85] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "RFC 3588 Diameter Base Protocol," Internet Engineering Task Force, 2003.

[86] OMA, "Presence SIMPLE Architecture." Open Mobile Alliance, 2006.

[87] OMA, "XML document management architecture," *Candidate Version*, vol. 2. Open Mobile Alliance, 2007.

[88] OMA, "Instant Messaging using SIMPLE." Open Mobile Alliance, 2009.

[89] OMA, "Push to talk over Cellular (PoC)-Architecture," vol. 1. Open Mobile Alliance, 2006.

[90] GSMA, "Rich Communication Suite (RCS) & Rich Communications Ecosystem (RCE)." White paper, GSM Association, 2009.

[91] J. Kallio, T. Jalkanen, and J. T. J. Penttinen, "Voice over LTE," *The LTE/SAE Deployment Handbook*. Wiley Online Library, Hoboken, NJ, pp. 157–187, 2012.

[92] M. Mouly, M.-B. Pautet, and T. Foreword By-Haug, *The GSM system for mobile communications*. Telecom Publishing, 1992.

[93] R. Fielding, J. Gettys, J. Mogul, and H. Frystyk, "RFC 2616 Hypertext transfer protocol–HTTP/1.1." Internet Engineering Task Force, 1999.

[94] OMA, “OMA service environment,” *Approved Version*. Open Mobile Alliance, pp. 1–36, 2007.

[95] S. Beddus, G. Bruce, and S. Davis, “Opening up networks with JAIN Parlay,” *Communications Magazine, IEEE*, vol. 38, no. 4, pp. 136–143, 2000.

[96] ETSI, “Parlay X Web Service Specification, Version 3.0,” *Published by ETSI as ES*. European Telecommunications Standards Institute, 2007.

[97] GSMA, “GSM World - OneAPI V1.0.” GSM Association, 2010.

[98] ITU-T, “Y.2234 : Open service environment capabilities for NGN.” International Telecommunication Union, 2008.

[99] OMA, “Reference Release Definition for Open Mobile Alliance Service Environment (OSE),” *Candidate Version*. Open Mobile Alliance, 2007.

[100] 3GPP, “TS 32.101 Telecommunication management; Principles and high level requirements.” 3rd Generation Partnership Project, 2012.

[101] J. Huang, “eTOM and ITIL: Should you be Bi-lingual as an IT Outsourcing Service Provider,” *BP Trends*, 2005.

[102] TMF, “GB927 The NGOSS Lifecycle and Methodology.” TeleManagement Forum, 2004.

[103] TMF, “GB922 Information Framework Concepts information framework.” TeleManagement Forum, 2011.

[104] TMF, “GB929 Application Framework (TAM).” TeleManagement Forum, 2013.

[105] TMF, “GB980 Integration Framework Suite.” TeleManagement Forum, 2013.

[106] TMF, “TR139 Service Delivery Framework Overview.” TeleManagement Forum, 2009.

[107] TMF, “IPsphere Framework, Technical Specification (Release 1).” TeleManagement Forum, 2007.

[108] C. Rudd and V. Lloyd, “Service Design: Office of Government Commerce (Itil).” The Stationery Office Ltd, St Crispins, Norwich, 2007.

[109] I. Sriram and A. Khajeh-Hosseini, “Research agenda in cloud technologies,” *arXiv: 1001.3259*, 2010.

[110] Y. Demchenko, C. Ngo, C. De Laat, J. Rodriguez, L. M. Contreras, J. A. G. Espin, S. Figuerola, G. Landi, and N. Ciulli, “Intercloud Architecture Framework for Heterogeneous Cloud based Infrastructure Services Provisioning On-Demand,” *IEEE International Conference on Advanced Information Networking and Applications*. IEEE, Barcelona, Spain, 2013.

[111] T. Metsch, A. Edmonds, R. Nyrén, and R. Nyr, "Open Cloud Computing Interface - Core," *Open Grid Forum*. Open Grid Forum, Muncie, IN, p. 17, 2011.

[112] S. Crosby, R. Doyle, M. Gering, M. Gionfriddo, S. Hand, M. Hapner, D. Hiltgen, M. Johanssen, J. Leung, F. Machida, and others, "Open virtualization format specification," *Standards and Technology, no. DSP0243 in DMTF Specifications, Distributed Management Task Force*, 2009.

[113] TMF, "TR178 Enabling End-to-End Cloud SLA Management." TeleManagement Forum, 2013.

[114] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," *IEEE Fourth International Conference on Internet and Web Applications and Services (ICIW'09)*. Venice/ Mestre, Italy, pp. 328–336, 2009.

[115] M. X. Makkes, C. Ngo, Y. Demchenko, R. Stijkers, R. Meijer, and C. de Laat, "Defining intercloud federation framework for multi-provider cloud services integration," *CLOUD COMPUTING 2013, The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization*. Valencia, Spain, pp. 185–190, 2013.

[116] D. Bernstein and V. K. Deepak, "P2302 Draft Standard for Intercloud Interoperability and Federation (SIIF)." IEEE Standards Draft, 2012.

[117] OASIS, "Topology and Orchestration Specification for Cloud Applications (TOSCA)." OASIS Committee Specification Draft 06 / Public Review Draft 01, 2012.

[118] A. V. Dastjerdi and R. Buyya, "A Taxonomy of QoS Management and Service Selection Methodologies for Cloud Computing," *Cloud Computing: Methodology, Systems, and Applications*, vol. 1. CRC Press, Boca Raton, FL, pp. 109–131, 2011.

[119] B. Rochwerger and D. Breitgand, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 1–11, 2009.

[120] C. Loomis, M. Airaj, M.-E. Bégin, E. Floros, S. Kenny, D. O'Callaghan, and others, "Stratuslab cloud distribution," in *European Research Activities in Cloud Computing*, D. P. and J. L. Vázquez-Poletti, Ed. Cambridge Scholars Publishing, 2012.

[121] A. J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, and others, "OPTIMIS: A holistic approach to cloud service provisioning," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 66–77, 2012.

[122] R. G. Cascella, L. Blasi, Y. Jegou, M. Coppola, and C. Morin, "Contrail: Distributed Application Deployment under SLA in Federated Heterogeneous Clouds," in *The Future Internet*, Springer, 2013, pp. 91–103.

[123] A. Hume, Y. Al-Hazmi, and B. Belter, "BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation," *Testbeds and Research Infrastructure.*

Development of Networks and Communities. Springer, Thessanoloki, Greece, pp. 81–96, 2012.

[124] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloud-computing system," *9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'09)*. IEEE, Shanghai, China, pp. 124–131, 2009.

[125] D. Milojičić, I. M. Llorente, and R. S. Montero, "OpenNebula: A Cloud Management Tool.," *IEEE Internet Computing*, vol. 15, no. 2. Los Alamitos, CA, 2011.

[126] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "OpenStack: Toward an Open-Source Solution for Cloud Computing," *International Journal of Computer Applications*, vol. 55, no. 3. Foundation of Computer Science, New York, USA, 2012.

[127] OpenStack Community, "OpenStack Compute Starter Guide - Cactus." CSS Corp Private Limited, 2011.

[128] E. Kurbatova and A. Tolstoy, "Cloud Computing with Amazon Web Services: A viable way forward?," *Conference of SibFU*. Сибирский федеральный университет, Siberia, 2011.

[129] ITU-T, "Recommendation P. 800.1, Mean opinion score (MOS) terminology." International Telecommunication Union, 2006.

[130] ITU-T, "Recommendation P.862: Perceptual evaluation of speech quality (PESQ)." International Telecommunication Union, 2001.

[131] R. Volk, "Integrated Service Management with OSS/J," *JOURNAL-COMMUNICATIONS NETWORK*, vol. 3, no. 3, pp. 125–131, 2004.

[132] M. Yousefvand and F. Ayazi, "An integrated functional architecture model for the Operations Support Systems: Using MTOSI-based standard APIs," *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*. St. Petesburg, Russia, pp. 292–297, 2012.

[133] Wikipedia, "Enterprise architecture framework - Wikipedia, The Free Encyclopedia," 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Enterprise_architecture_framework&oldid=606916804. [Accessed: 16-May-2014].

[134] The Open Group, "The Open Group Architecture Framework (TOGAF)." The Open Group, 2009.

[135] TMF, "TMF 053. The NGOSS Technology Neutral Architecture." TeleManagement Forum, 2005.

[136] TMF, "TMF 061. Service Delivery Framework Reference Architecture." TeleManagement Forum, 2009.

[137] "SIP Express Media Server." [Online]. Available: http://www.iptel.org/sems. [Accessed: 16-May-2014].

[138] I. Brandic, "Towards Self-Manageable Cloud Services," *33rd Annual IEEE International Computer Software and Applications Conference*, no. iv. IEEE, Seattle, WA, pp. 128–133, 2009.

[139] M. Godse and S. Mulik, "An approach for selecting software-as-a-service (SaaS) product," in *IEEE International Conference on Cloud Computing (CLOUD'09)*, 2009, pp. 155–158.

[140] E. Triantaphyllou, *Multi-criteria Decision Making Methods: A Comparative Study (Applied Optimization)*, 2000th ed. Springer, 2000.

[141] C. M. Brugha, "Structure of multi-criteria decision-making," *Journal of the Operational Research Society*, vol. 55, no. 11, pp. 1156–1168, 2004.

[142] Z. U. Rehman, F. K. Hussain, and O. K. Hussain, "Towards Multi-criteria Cloud Service Selection," *2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, Seoul, Korea, pp. 44–48, Jun-2011.

[143] W. Zeng, Y. Zhao, and J. Zeng, "Cloud service and service selection algorithm research," in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 2009, pp. 1045–1048.

[144] O. K. Hussain, F. K. Hussain, and others, "Multi-criteria IaaS Service Selection Based on QoS History," *IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. Barcelona, Spain, pp. 1129–1135, 2013.

[145] S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh, "Efficient service recommendation system for cloud computing market," *Proceedings of the 2nd International Conference on Interaction Sciences Information Technology, Culture and Human - ICIS '09*. ACM Press, New York, New York, USA, pp. 839–845, 2009.

[146] A. Quiroz, H. Kim, M. Parashar, N. Gnanasambandam, and N. Sharma, "Towards autonomic workload provisioning for enterprise grids and clouds," *Grid Computing, 2009 10th IEEE/ACM International Conference on*. Shanghai, China, pp. 50–57, 2009.

[147] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, Jan. 2012.

[148] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," *IEEE International Conference on Cloud Computing (CLOUD), 2011*. Ieee, Washington, DC, pp. 500–507, Jul-2011.

[149] P. Dinda and D. O'Hallaron, "Host load prediction using linear models," *Cluster Computing*, vol. 3. Springer, pp. 265–280, 2000.

[150] W. Dawoud, I. Takouna, and C. Meinel, "Elastic VM for rapid and optimum virtualized resources' allocation," *2011 5th International DMTF Academic Alliance*

Workshop on Systems and Virtualization Management: Standards and the Cloud (SVM). IEEE, Paris, France, pp. 1–4, 2011.

[151] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing SLA violations," *10th IFIP/IEEE International Symposium on Integrated Network Management (IM'07)*. Munich, Germany, pp. 119–128, 2007.

[152] A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko, and A. Tantawi, "Dynamic placement for clustered web applications," *Proceedings of the 15th international conference on World Wide Web*. ACM, New York, NY, USA, pp. 595–604, 2006.

[153] S. Gatziu, T. Kumar, and W. Holger, "Cloud Broker: Bringing Intelligence into the Cloud An Event-Based Approach," *IEEE 3rd International Conference on Cloud Computing (CLOUD)*. Miami, FL, pp. 6–7, 2010.

[154] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, 2009, pp. 103–110.

[155] W. Li, J. Tordsson, and E. Elmroth, "Modeling for dynamic cloud scheduling via migration of virtual machines," *Third International IEEE Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, Athens, Greece, pp. 163–171, Nov-2011.

[156] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers," *Future Generation Computer Systems*, vol. 28, no. 2. Elsevier, Amsterdam, Netherlands, pp. 358–367, 2012.

[157] W. Iqbal, M. N. Dailey, and D. Carrera, "SLA-Driven Dynamic Resource Management for Multi-tier Web Applications in a Cloud," *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*. IEEE, Melbourne, Australia, pp. 832–837, 2010.

[158] D. Breitgand, A. Marashini, and J. Tordsson, "Policy-driven service placement optimization in federated clouds," *IBM Research Division, Tech. Rep*, 2011.

[159] B. Rochwerger, D. Breitgand, D. Hadas, I. Llorente, R. Montero, P. Massonet, E. Levy, A. Galis, M. Villari, Y. Wolfsthal, and others, "An architecture for federated cloud computing," *Cloud Computing*. Springer, 2010.

[160] B. Rochwerger, D. Breitgand, and A. Epstein, "Reservoir-when one cloud is not enough," *IEEE Computer*, vol. 44, no. 3, pp. 44–51, 2011.

[161] C. Chapman, W. Emmerich, and F. Márquez, "Elastic Service Management in Computational Clouds," *2th IEEE/IFIP Network Operations and Management Symposium (NOMS2010) / International Workshop on Cloud Management (CloudMan 2010)*. Osaka, Japan, pp. 1–8, 2010.

[162] "Rackspace. Rackspace hosting." [Online]. Available: http://www.rackspace.co.uk/. [Accessed: 27-Jan-2014].

[163] "ElasticHosts. Flexible servers in the Cloud." [Online]. Available: http://www.elastichosts.com. [Accessed: 27-Jan-2014].

[164] G. V. S. Alastair C. Hume, Yahya Al-Hazmi, Bartosz Belter, Konrad Campowsky, Luis M. Carril, Gino Carrozzo, Vegard Engen, David García-Pérez, Jordi Jofre Ponsatí, Roland Kűbert, Yongzheng Liang, Cyril Rohr, "BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation," *Proceedings of the 8th International ICST Conference, TridentCom 2012*, no. January. pp. 81–96, 2012.

[165] Bonfire Staff, "BonFIRE User Documentation Rel. 3.0," 2014. [Online]. Available: http://doc.bonfire-project.eu/R3/#. [Accessed: 16-May-2014].

[166] D. Gracia-Perez, J. Lorenzo del Castillo, Y. Al-Hazmi, J. Martrat, K. Kavoussanakis, A. Hume, C. V. Lopez, G. Landi, T. Wauters, and M. Gienger, "Cloud and Network facilities federation in BonFIRE," in *Federative and interoperable cloud infrastructures (FedICI-2013)*, 2013, pp. 1–10.

[167] D. Bernstein, D. Vij, "IEEE P2302™/D0.2 Draft Standard for Intercloud Interoperability and Federation (SIIF)" Jan. 2012, [Online].Available: https://www.oasis-open.org/committees/download.php/46205/p2302-12-0002-00-DRFT-intercloud-p2302-draft-0-2.pdf [Accessed: 16-May-2014].

[168] 3GPP, "TS 23.218, IP Multimedia (IM) session handling; IM call model; Stage 2." 3rd Generation Partnership Project, 2013.

[169] 3GPP, "TS 29.228, IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signaling flows and message contents." 3rd Generation Partnership Project, 2013.

[170] T. Magedanz, S. Dutkowski, and Y. G. Gil-Laich, "The OpenPEEM as core for service orchestration within the Open IMS Playground at FOKUS," *International Conference on Intelligence in Networks (ICIN 2007)—Emerging Web and Telecom Services: Collision or Coopetition*. Bordeaux, France, 2007.

[171] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, "Time series analysis: forecasting and control." John Wiley & Sons, Hoboken, NJ, 2013.

[172] K. Keahey, "Nimbus: open source infrastructure-as-a-service cloud computing software," in *Workshop on adapting applications and computing services to multicore and virtualization, CERN, Switzerland*, 2009.

[173] A. Blotny and N. Blum, "A platform providing bidirectional service integration for the dynamic long-tail service market," *Intelligence in Next …*, pp. 0–5, 2010.

[174] D. Rolls, "Service Provisioning Markup Language (SPML) Version 1.0," *OASIS Committee Specification*, 2003.

[175] "SIPp, Open Source IMS benchmarking tool." [Online]. Available: http://sipp.sourceforge.net/index.html. [Accessed: 16-May-2014].

[176] D. Mosberger and T. Jin, “httperf—a tool for measuring web server performance,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 3. ACM, New York, NY, pp. 31–37, 1998.

[177] “HAProxy-The Reliable, High-Performance TCP/HTTP Load Balancer,” 2012. [Online]. Available: http://haproxy.1wt.eu/. [Accessed: 16-May-2014].

[178] N. Kandasamy, “A hierarchical optimization framework for autonomic performance management of distributed computing systems,” *26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006)*. Lisbon, Portugal, 2006.

[179] T. Balog, M. Brozowski, D. Hustace, and B. Reed, “OpenNMS,” *SNMP walker and network manager*, vol. 16, 2004.

[180] “Zabbix. An Enterprise-Class Open Source Distributed Monitoring Solution,” 2007. [Online]. Available: http://www.zabbix.com/. [Accessed: 16-May-2014].

[181] D. Jefford, K. Smith, and E. Fairweather, “Professional BizTalk Server 2006,” 2007.

[182] E. Friedman-Hill, “Jess, the rule engine for the java platform.” Sandia National Laboratories, Livermore, CA, 2008.

[183] M. Bali, “Drools JBoss Rules 5. X Developer’s Guide.” Packt Publishing Ltd, Birmingham, UK, 2013.

[184] “Updated Optimis Architecture Design Document - (updated version M12 - June 2011) | Optimis - Optimized Infrastructure Services,” 2011. [Online]. Available: http://www.optimis-project.eu/content/updated-optimis-architecture-design-document-updated-version-m12-june-2011. [Accessed: 16-May-2014].

Appendix I: Author's Dissemination

AI.1 Books

[1] Borcea, C., Bellavista, P., Giannelli, C., Magedanz, T., & Schreiner, F. (Eds.). (2013). Mobile Wireless Middleware, Operating Systems, and Applications - 5th International Conference, Mobilware 2012, Berlin, Germany, November 13-14, 2012, Revised Selected Papers. In MOBILWARE (Vol. 65). Springer. ISBN: 978-3-642-36659-8

AI.2 Journal / Book Articles

[1] N. Blum, T. Magedanz, F. Schreiner, and S. Wahle, „From IMS Management to SOA based NGN Management“, Journal of Network and Systems Management, Issue 17(1-2), June 2009, pp. 33-52, ISSN: 1064-7570 (Print) 1573-7705

[2] T. Magedanz, F. Schreiner, and S. Wahle, „From NGN to Future Internet Testbed Management - Collaborative Testbeds as Enabler for Cross-Technology, Cross-Layer, and Cross-Domain Communication and Network Research.“ Tele Kommunikation Aktuell, 62(5-6):20-40, 2008, pp. 1-20, ISSN 1619-2036

[3] N. Blum, T. Magedanz, F. Schreiner, „The Role of Service Brokers for Composed Services in an Open Service Environment”, TELEKOMMUNIKATION AKTUELL 1-2/2008, Verlag für Wissenschaft und Leben Georg Heidecker GmbH, Erlangen, pp. 3-21, ISSN: 1619-2036

[4] N. Blum, T. Magedanz, F. Schreiner, „Definition of a Service Delivery Platform for Service Exposure and Service Orchestration in Next Generation Networks", UbiCC Journal - Volume 3 Number 3, 2008, pp. 102-1, ISSN: 1994-4608

[5] N. Blum, P. Jacak, F. Schreiner, D. Vingarzan, P. Weik – „Towards Standardized and Automated Fault Management and Service Provisioning for NGNs" Journal of Network and Systems Management, Springer Netherlands, Volume 16, Number 1 / March 2008, pp. 63-91, ISSN: 1064-7570 (Print) 1573-7705 (Online)

[6] T. Magedanz, F. Schreiner, P. Weik, „Das IP Multimedia Subsystem“; Handbuch der Telekommunikation, Dr. Franz Arnold, 124. Erg.-Lfg. Kapitel 12.2.5.0, Franz Arnold (Hrsg.) Wolters Kluwer Deutschland GmbH, March 2007, ISBN: 978-3871560965

[7] N. Blum, T. Magedanz, J. Müller, F. Schreiner, “Telecom applications, APIs and service platforms”. In E. Bertin, N. Crespi, and T. Magedanz, eds. Evolution of Telecommunication Services. 2013 ed. Vol. 7768. N.p.: Springer, 2013. Print. Lecture Notes in Computer Science / Information Systems and Applications, Incl. Internet/Web, and HCI (Book 7768), pp. 25-46, ISBN 978-3-642-41568-5

AI.3 Publications

[1] B. Harjoc, T. Magedanz, F. Schreiner, „Automated Root Cause Analysis for IMS and SDP", in proceedings of NGMAST 2009 - Third International Conference on NEXT GENERATION MOBILE APPLICATIONS, SERVICES and TECHNOLOGIES, Cardiff, Wales, UK, 15-18 September 2009, pp. 27-32, ISBN: 978-0-7695-3786-3

[2] Niklas Blum, Thomas Magedanz, Florian Schreiner, and Sebastian Wahle, „Service Oriented Testbed Infrastructures: a Cross-Layer Approach for NGNs." ACM/Springer Mobile Networks and Applications (MONET), Special Issue: Advances In Wireless Test beds and Research Infrastructures, August 2009, pp. 413-424, ISSN 1383-469X (Print) 1572-8153

[3] B. Harjoc, T. Magedanz, F. Schreiner, D. Vingarzan, „Automated Fault Localization based on Unified Web Service and NGN Benchmarking". 14th IEEE Symposium on Computers and Communications (ISCC'09), Sousse, Tunisia. July 5 - 8, 2009, pp. 77 – 82, ISSN: 1530-1346, ISBN: 978-1-4244-4672-8

[4] N. Blum, T. Magedanz, F. Schreiner, „Management of SOA based NGN service exposure, service discovery and service composition." IFIP/IEEE International Symposium on Integrated Network Management, 2009, pp. 430–437, ISBN 978-1-4244-3487-9

[5] T. Magedanz, F. Schreiner, S. Wahle, „Service-Oriented Testbed Infrastructures and Cross-Domain Federation for Future Internet Research.", In Integrated Network Management-Workshops, 2009. IM '09. IFIP/IEEE International Symposium on, New York, USA, June 2009, pp. 101-106, ISBN: 978-1-4244-3923-2

[6] N. Blum, T. Magedanz, F. Schreiner, S. Wahle, „A Research Infrastructure for SOA-based Service Delivery Frameworks - The Open SOA Telco Playground at Fraunhofer FOKUS.", In TRIDENTCOM 2009, 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Washington DC, USA, April 2009, pp. 1-6, ISBN: 978-1-4244-2847-2, IEEE CN: CFP09364.

[7] N. Blum, T. Magedanz, F. Schreiner, „Services, Enablers and Architectures: Definition of a Connected Web 2.0 / Telco Service Broker to Enable New Flexible Service Exposure Models", Proc. of International Conference on Intelligence in Networks (ICIN), Bordeaux, 20 - 23 October 2008, pp.20-13, ISBN 978-1-60702-868-0

[8] F. Schreiner, S. Wahle, N. Blum, T. Magedanz: „Modular Exposure of Next Generation Network Services to Enterprises and Testbed Federations", HUT-ICCE 2008, 2nd International Conference on Communications and Electronics, Hoi An, Vietnam, June 4-6, 2008, pp.98-103, ISBN 978-1-4244-2425-2, IEEE CN CFP0816B-PRT

[9] J. M. Gonzalez, F. Gouveia, J. A. Lozano, T. Magedanz, F. Schreiner, „Towards Autonomic Communication Mechanisms for Service Composability Management“ ease, pp.197-203, Fifth IEEE Workshop on Engineering of Autonomic and Autonomous Systems (ease 2008), 2008, ISBN: 978-0-7695-3140-3

[10] N. Blum, F. Schreiner, T. Magedanz, „Emerging Web and Telecom Services: Prototyping FMC Services based on IMS and Web 2.0 for a Mobile Operator", International Conference on Intelligence in Networks (ICIN) 2007, Bordeaux, France, 8 - 11 October 2007

[11] F. Dinu, P. Jacak, N. Blum, T. Magedanz, F. Schreiner, „Automated Service Provisioning and Fault Management for OSIMS Based NGNs", In Proceedings of the 14th Annual Workshop of HP Software University Association, 2007, Infonomics--Consulting, Stuttgart, Germany, July, 2007, pp. 271-276, ISBN-13: 978-3-00-021690-9

[12] K. Campowsky, G. Carella, T. Magedanz, F. Schreiner, "Network-aware Cloud Brokerage for telecommunication services", IEEE CloudNet'12, 2012 1st IEEE International Conference on Cloud Networking, November 28-30, 2012, Université Pierre et Marie Curie, Paris, pp. 131-136, ISBN: 978-1-4673-2798-5

[13] P. Bellavista, K. Campowsky , G. Carella, L. Foschini, T. Magedanz, F. Schreiner, "QoS-aware elastic cloud brokering for IMS infrastructures", The Seventeenth IEEE Symposium on Computers and Communications (ISCC'12). July 1 - 4, 2012, Cappadocia, Turkey, pp. 157-160, ISBN: 9781467327121, ISSN: 15301346

[14] K. Campowsky , G. Carella, T. Magedanz, F. Schreiner, "Optimization of Elastic Cloud Brokerage Mechanisms for Future Telecommunication Service Environments", Praxis der Informationsverarbeitung und Kommunikation. Volume 35, Issue 3, June 2012, pp. 153-159, ISSN (Online) 1865-8342, ISSN (Print) 0930-5157

[15] W. Vandenberghe, B. Vermeulen, P. Demeester, A. Willner, S. Papavassiliou, A. Gavras, M. Sioutis, A. Quereilhac, Y. Al-Hazmi, F. Lobillo, F. Schreiner, C. Velayos, A. Vico-Oton, G. Androulidakis, C. Papagianni, O. Ntofon, and M. Boniface, "Architecture for the Heterogeneous Federation of Future Internet Experimentation Facilities" In Future Network and Mobile Summit 2013, Lisbon, Portugal, Jul 2013, pp. 1-17, ISBN: 978-1-905824-37-3

[16] K. Campowsky, G. Carella, T. Magedanz, F. Schreiner, "Elasticity as a Service for federated cloud testbeds", International Conference on Communications (IEEE ICC'13), 2nd Workshop on Clouds, Networks and Data Centers (CNDC), Budapest, Hungary, June 2013, pp. 256-260, ISBN: 978-1-4673-5753-1

[17] F. Schreiner, T. Magedanz, "QoS-aware Multi-Cloud Brokering for NGN Services." The Fifth International Conference on Communications and Electronics – ICCE, Da Nang, Vietnam, 2014, pp. 157-160, ISBN: 978-1-4799-5049-2

AI.4 Tutorials / Workshops

[1] A. Al-Hezmi, T. Magedanz, F. Schreiner; “Impacts of SOA on the Strategy for building Next Generation Network´s Multimedia Applications”, Workshop, IQCP SOA Forum, London, UK, May, 2006

[2] F. Schreiner, T. Magedanz: „Operations and Business Support Systems for Next Generation Networks and the IP Multimedia System”, halfday Tutorial at 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TRIDENTCOM, Innsbruck, Austria, 2008

[3] F. Schreiner, T. Magedanz: „Operations and Business Support Systems for Next Generation Networks and the IP Multimedia System”, halfday Tutorial at 5th Latin American Network Operations and Management Symposium (LANOMS), Petrópolis, Brazil September 10-12, 2007

[4] N. Blum, F. Schreiner: “Smart Bit Pipes: Open APIs and their Role in Emerging SOA SDPs for Converging Networks” 5th International FOKUS IMS Workshop, Berlin, 2009

AI.5 Conferences and Workshop Chair and TPC

Workshop Chair:

1) INFINITY Project Workshop, Tridentcom 2012, Thessaloniki, Greece, 2012
2) International FI-PPP Workshop, FOKUS FUSECO Forum 2013, Berlin, Germany, 2013

TPC Co-Chair:

1) MOBILWARE 2012 - 5th International ICST Conference on Mobile Wireless Middleware, OSs, and Applications Testbeds in TridentCom, Berlin, Germany, 2012
2) Future Internet Symposium 2011, Vienna, Austria, November 2011.

TPC Member:

1) 4th IFIP/IEEE Workshop on Distributed Autonomous Network Management Systems (DANMS 2011), Dublin, Ireland, 2011
2) ITU Kaleidoscope event, Cape Town, South Africa, December 2011
3) ITU Kaleidoscope conference, Kyoto, Japan, April 2013
4) CloudCom 2013 - Using and building Coud Testbeds UNICO Workshop, Bristol, UK, December 2013
5) Intercloud IC2E Workshop 2014 - Cloud Computing Interclouds, Multiclouds, Federations, and Interoperability, Boston, Massachusetts, USA, March 2014

6) ICC'14 NGN, K-BSC 2013, IEEE CloudNet'12, K-2011, ICC'11 CSMA, ICC'11 Workshop TH4 (T1) AMN, K-2010, SoftCOM 2010, DANMS'08, IPTComm'08

AI.6 Master and Diploma Theses Supervisor

1) S. Krämer, S, F. Schreiner (Supervisor); U. Bareth, (Supervisor): Design, Implementation and Evaluation of Automated "Composite Service Management Mechanisms for NGN Service Brokers in Accordance with SOA Principles", München, Univ., Dipl.-Arb., 2008
2) A. Fedulov, F. Schreiner (Supervisor), T. Magedanz (Supervisor), A. Timm-Giel (Supervisor), "Design and Implementation Of a Flexible Platform-Independent Elasticity Management System for CloudBased Services", Master Thesis, Technische Universität Hamburg-Harburg, 2011
3) G. Carella, F. Schreiner (Supervisor), P. Bellavista, "QoS-aware brokering support for IMS Infrastructures in cloud", 2011, Maser Thesis, University of Bologna, 2011
4) M. Umair, Jr. Maguire, Q. Gerald, F. Schreiner (Supervisor) "Performance Evaluation and Elastic Scaling of an IP Multimedia Subsystem Implemented in a Cloud," Master Thesis, KTH, Radio Systems Laboratory (RS Lab), 2013

AI.7 Conference Presentations and Workshops

1) Many IMS, SDP and OSS related talks 2006 - today
2) F. Schreiner "*Optimization of Elastic Cloud Brokerage Mechanisms for Future Telecommunication Service Environments*", 4th GI/ITG KuVS Fachgespräch on NGN SDPs, Vienna, 2011
3) F. Schreiner, "*Cloud Brokerage for Telecoms*", IIR Telecoms, Cloud Enablers Summit, London, 2012
4) F. Schreiner, "*How Cloud Computing and Cloud Brokering technologies help telecommunication service providers and enterprises to efficiently and economically optimize their IT and service platforms and service delivery models*", 6th international Workshop SENAI - Automação e TIC, Florianopolis/SC, Brasil, July 2012
5) G. Garella, T. Magedanz, F. Schreiner, "*Elastic Cloud Principles applied onto Telco SDPs and NFV*" 8th KuVS Fachgespräch NGSDP – "Competitive Service Delivery Infrastructures" , Vodafone Schulungszentrum,, Königswinter, Germany, April 17th, 2013

Appendix II: Detailed Specification of the NGN Cloud Broker

AII.1 Interfacing NGN and Cloud Platform Functions

NGN service control functions, after users have already registered to the network, coordinate the service invocation and session control of NGN-based telecommunication services. Based on user and service profiles, session control mechanisms are capable of controlling the access to services, in a user-specific fashion meaning that for each user a different set of services might be made accessible and invocations of one and the same service type might result in different invocations of different application service instances. For each new service domain, e.g. a new cluster of application serving nodes, the service control layer has to be informed about service end-points, invocation methods and the users for which this new service cluster should be made accessible. As soon as the session control layer has been provisioned, incoming service requests (from users authorized to access the service) are forwarded to the appropriate service end-point and session control mechanisms are applied for the duration of the session.

AII.1.1 NGN Service Control Platform Functions

The de-facto standard NGN session control protocol is the Session Initiation Protocol (SIP). SIP provides a set of session control methods which are used for user/subscriber registration purposes as well as for controlling different telecommunication service sessions, from simple messaging, to presence information exchange, to complex state-full voice and video telecommunication services with multiple participants. The session control layer is responsible for receiving and forwarding the incoming SIP requests to the appropriate destination.

The actual session control functions are SIP-based proxies and routing functions, which are responsible for registration (authentication and further authorization) of users, interconnecting users for simple messaging and simple voice / video call establishment and control and for coordinating intra-domain as well as cross-domain invocation and session of more complex, or roamed telecommunication services.

The IP Multimedia Subsystem

The IMS is de the-facto standard session control system for NGNs, depicted in Figure 133. The IMS is designed in a highly scalable fashion. For the end-user / client the Proxy Call Service Control Function (P-CSCF), is the entry point to NGNs. Already at the point of user / client registration, complex SIP transactions between user / client and network, as well as inside the NGN core are used to locate the correct databases which hold the user's profile and subscription information and to interrogate between functions for registering users and initializing the standard set of a user's subscribed services. After the correct user-/service-

profile database, the Home Subscriber Server (HSS) has been identified, the relevant data for further controlling a user's services and sessions are downloaded from the HSS to the main coordinating SIP function, the Serving Call Service Control Function (S-CSCF).

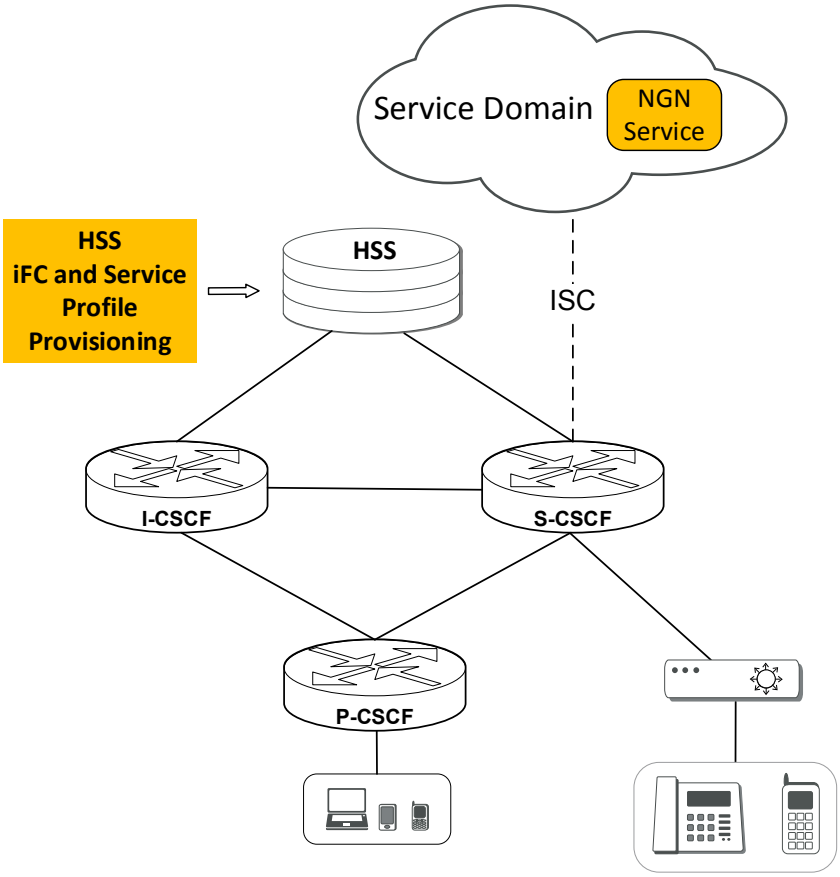

Figure 133: IP Multimedia Subsystem Core Functions, HSS Provisioning

This basic procedure is depicted in Figure 134, where the initial Filter Criteria (iFC) are downloaded from the HSS to the S-CSCF. This downloading process is conducted via Diameter-based information exchange containing the iFCs of a particular user / subscriber. iFCs define, to which SIP application server (AS) a specific service request needs to be routed from the IMS session control layer. This is being achieved by inspecting the incoming SIP request, analysing user-related fields, SIP methods and SIP headers. Based on user- and service-specific rules, which are based on iFCs, invocation of services and coordination of sessions is conducted in a user-specific fashion. The SIP AS can either reside inside the IMS / NGN domain or be deployed outside the NGN core network, but still it must be reachable through the SIP URL as specified in the iFC.

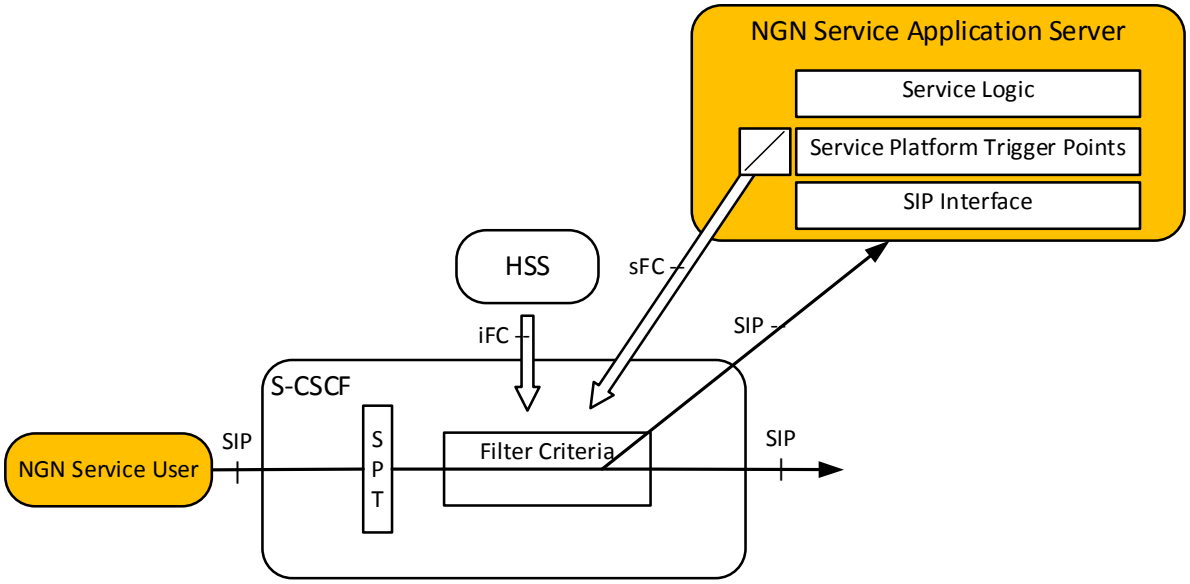

Figure 134: IMS Application Triggering Architecture, Filter Criteria [168]

A typical iFC is structured as shown in Figure 135. Of core importance for proper invocation of the correct application server are the trigger points, i.e. SIP request specific methods, headers which serve to identify the appropriate service request as well as application server specific data for locating and properly invoking the correct application server(s). So called “subsequent Filter Criteria” (sFCs) can furthermore be used to orchestrate a chain of sequential application server invocations, through which complex service orchestrations on the SIP level are realized.

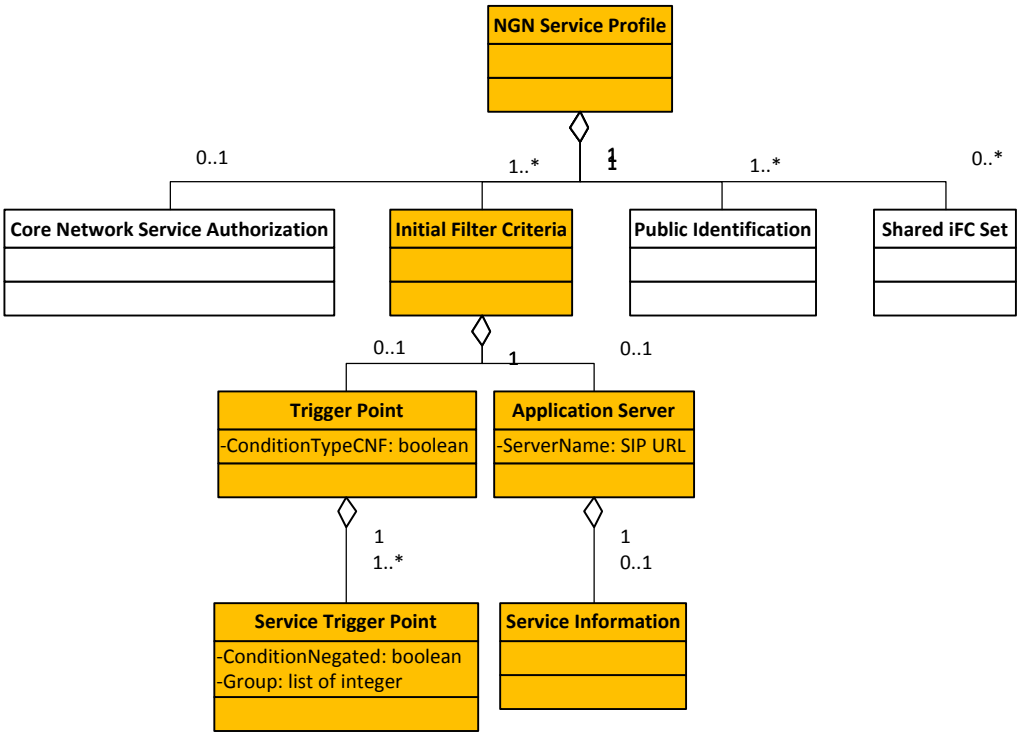

Figure 135: NGN Service Profile, iFC Structure [169]

AII.1.2 NGN Service Delivery Platform Functions

As shown in Figure 136 a typical NGN SDP comprises a broad range of functions for the actual service lifecycle management as well as for policy-based service access control and service coordination / orchestration. Service lifecycle management functions include functions for service creation, service development, service composition, service registration functions and service management functions. Functions controlling the access from a user perspective include user-specific policy enforcement functions as well as service discovery functions.

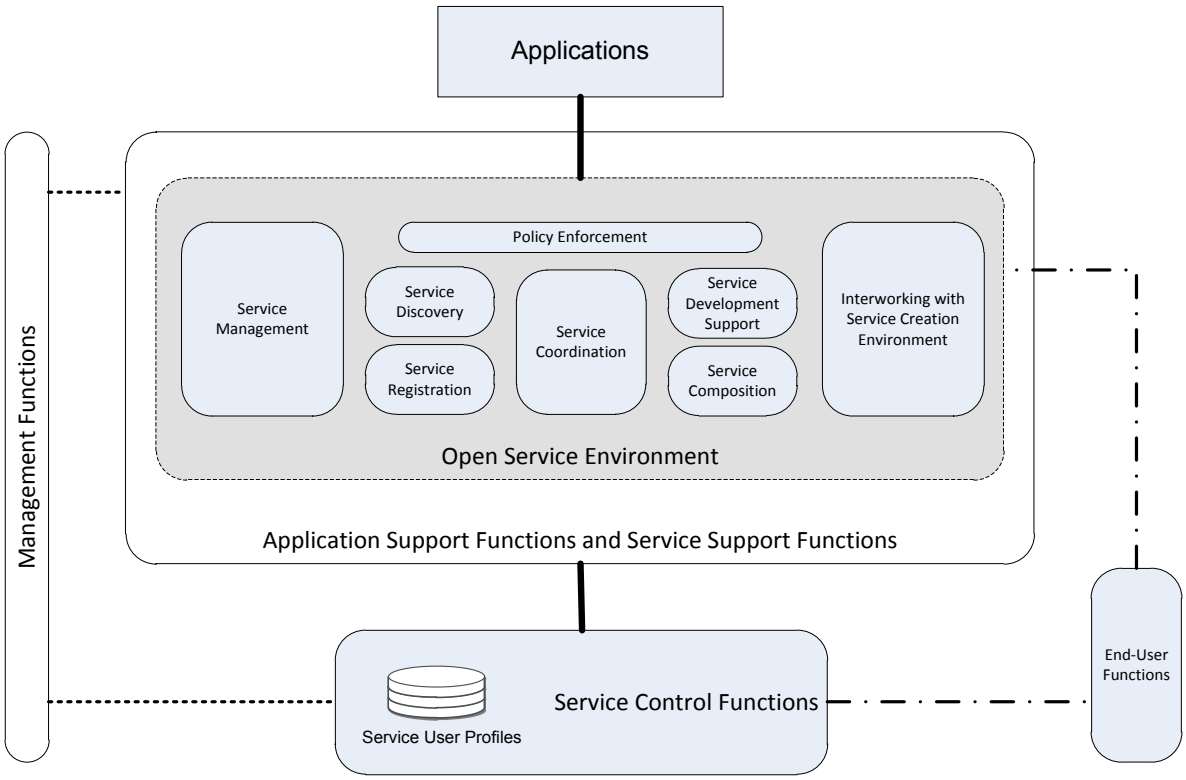

Figure 136: ITU-T Extended NGN architecture positioning the OSE [98]

Similar to the relevant NGN service control layer functions described in section AII.1.1, for the interworking of the proposed system with the functions at the NGN SDP layer, are functions for registering external services (the cloud-based NGN services) at the SDP, providing information about service end-points and service invocation mechanisms to the SDP. By doing so, cloud-based NGN services can be utilized as if they were deployed on a local SDP infrastructure at the premises of a telecommunication service provider.

Registering NGN services at the NGN SDP is important for letting the SDP as well as users know about the availability of a new / newly placed cloud-based NGN service. Apart from that, the SDP's policy evaluation and enforcement functions allow for defining specific, user- and service-related policies, through which particular user segments are providing access to one class of services whereas other user segments are providing access to other classes of services. Exploiting the spectrum of cloud-based resource offerings, provided at various QoS levels and at different prices, in [21] the opportunity for telecommunication

service providers to deploy their telecommunication services on different cloud platforms, while differentiating the delivery of those services, based on user segments was identified. This allows for infrastructure cost savings through QoS differentiated cloud-based service delivery.

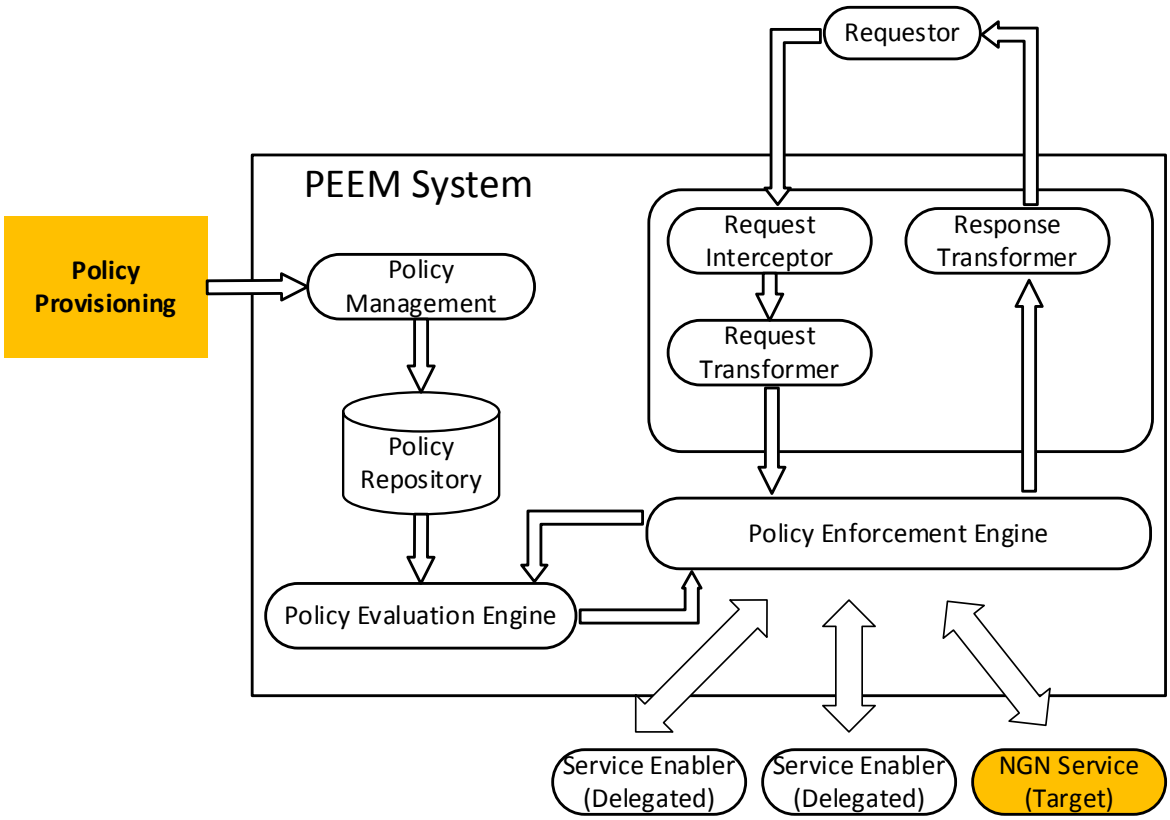

Figure 137: OMA OSE PEEM Reference Implementation [170]

AII.1.3 NGN Management Platform Functions

As explained in section 7.2.5, each time a new NGN service needs to be activated, as well as each time an NGN service is migrated to a different Cloud platform, the NGN Cloud Broker needs to provision NGN service control layer functions (described in the beginning of this section) as well as functions of the NGN service delivery platform (described in above section AII.1.2). As shown in Figure 138, as part of a SOA-based NGN management platform, service provisioning servers expose provisioning interfaces, capable of provisioning the service control platform, as well as the service delivery platform.

In order to provision the NGN service control layer, as explained in the beginning of this section AII.1.1, the HSS needs to be provisioned with NGN service profiles, comprising the initial filter criteria, based on which user requests can appropriately be forwarded to the actual service end-point. The initial filter criteria need to specify the service trigger point (which typically does not change during the run-time of the brokered NGN service), as well as the

application server URL (which changes as soon as the service is deployed on different locations).

In order to provision the NGN service delivery layer, service policies need to be updated within the NGN service environment's policy evaluation, enforcement and management platform. Therefore the policy repository (as explained in section AII.1.2) needs to be provisioned with service access and composition policies (which typically do not change during the brokering runtime) as well as with the service end-points (which change as soon as a service is migrated to a different cloud platform).

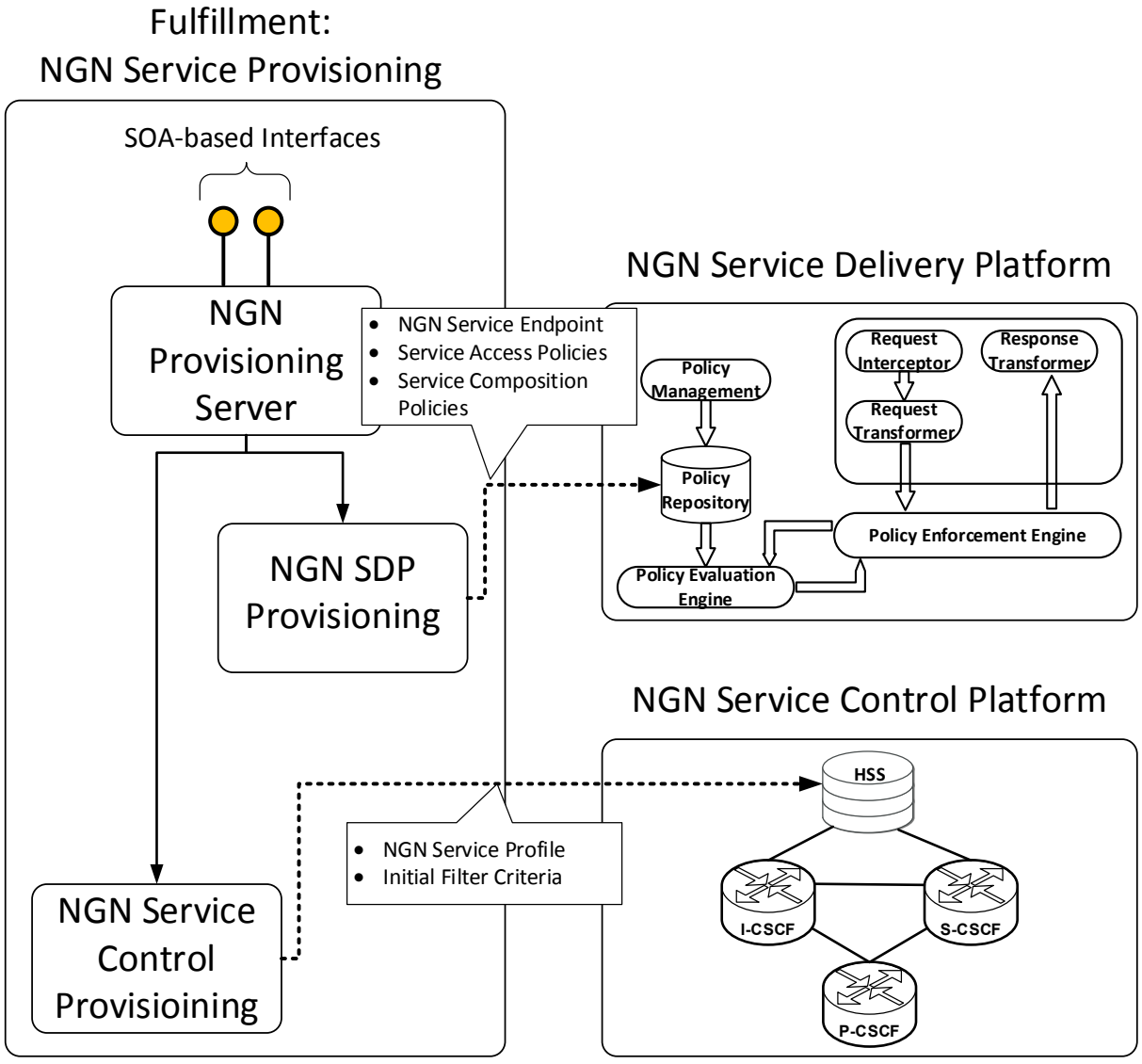

Figure 138: NGN Service Provisioning of NGN Service Control and Service Delivery Platform

In addition to the fulfillment related interworking requirements of the NGN Cloud Broker with the NGN management platform, there are two additional domains, with which the NGN Cloud Broker in a fully integrated solutions would need to interwork with:

- Systems for NGN service assurance

- Systems for NGN service billing

Full interworking with NGN service assurance systems would require the NGN Cloud Broker to receive and submit monitoring and fault information from/to the NGN management platform's service assurance systems.

Full interworking with the NGN billing systems would require the NGN Cloud Broker to continuously update the NGN management's resource data collection and processing elements with actual usage reports containing information about currently utilized resources of all Cloud platforms.

Regarding NGN service assurance interworking of the NGN Cloud Broker: Since fault management mechanisms, although taken into account for selecting alternative Cloud platforms, are not in the primary scope of this work, the specification and integration of the NGN Cloud Broker into NGN fault management systems was omitted. Furthermore, integration of the NGN Cloud Broker's monitoring system into the NGN management monitoring system would require the specify, of pushing monitoring data to the NGN monitoring system, and retrieving monitoring data from the NGN monitoring system, which was intentionally omitted.

Regarding interworking with NGN billing systems: The user-based charging and accounting for using brokered, multi-cloud-based NGN services is not impacted by the NGN Cloud Broker's system, as charging is carried out at the NGN service control layer (the IMS). Only for the supplier layer (i.e. the Cloud platform providers in case of this work), reporting of the current expenses for utilizing third parties' (Cloud) resource would be of relevance. This however, is currently not standardized as NGN enterprises / NGN service providers typically buy infrastructure resources on a long- to mid-term basis. Therefore it was refrained from specifying / formalizing this process. However a rough mapping of the process of assessing resource costs is provided in the evaluation chapter's section 9.7, where different Cloud platforms are used during the brokering phase and costs for using the dynamically scaled resources are asserted, mapping the capacity saving performance to potential cost savings.

AII.1.4 Cloud Service Provisioning Functions

Having introduced standards and best practices for cloud infrastructure management in section 2.7 and for cloud service management in section 2.8, the following relevant Cloud platform management functions are identified, as depicted in Figure 139 and as introduced in section 2.7.1:

Cloud Instance Management Functions:

- Virtual Machine Control Functions (start, re-start, reboot, delete, migrate, deploy, stop)
- Virtual Image Repository (list, enable, disable, register, update)

Virtual Network Management Functions:

- Virtual Network Control Functions (create, delete, monitor, list VM interconnections, optionally also traffic prioritization, encryption, isolation)

Cloud Instance Discovery/Lookup Functions:

- Instance type discovery (e.g. micro, small, medium, large,..)
- Hardware profile discovery
- Realm discovery
- Image discovery
- Cost information discovery

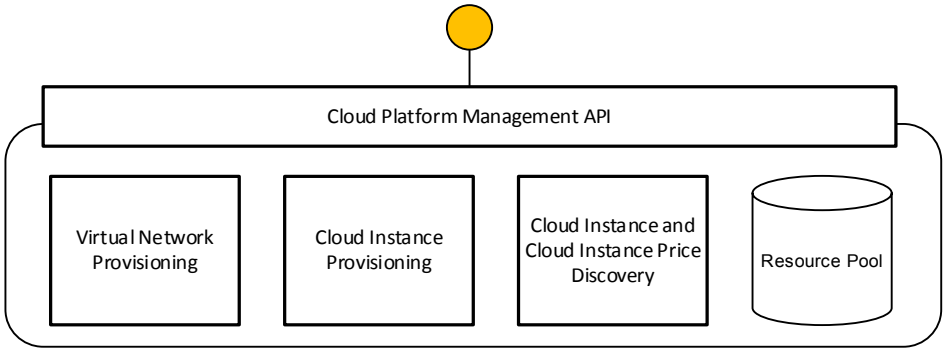

Figure 139: Cloud Platform Management Functions

On the one hand, the proposed system should be agnostic to the actual cloud infrastructure management mechanisms, being able to interwork with multiple cloud infrastructures simultaneously solely by controlling virtual infrastructure management APIs. On the other hand, the system should be capable of working in hybrid cloud scenarios, where in fact the deployment and management of a local cloud infrastructure becomes relevant.

AII.2 NGN Service and Service Scalability Functions

The NGN Cloud Broker core functions for resource allocation and service quality management should be sufficiently generic for supporting many different types of services, NGN specific (typically SIP-based) telecommunication services as well as standard web applications (e.g. 3-tiered web application architectures). Nevertheless the type of service invocation protocols (e.g. whether SIP, HTTP-based or both) plays a critical role for the overall provisioning process, particularly of the dependent NGN systems as well as for the utilized application scaling architecture.

AII.2.1 NGN Service Functions

Principally, there are two kinds of service "flavours" which need to be supported by any telecommunication service layer, i.e. SIP-based application servers and HTTP-based application servers. Whereas HTTP-based application servers are already widely used on cloud-based service platforms, already providing typical Web-based services to millions of users, SIP-based applications are of central importance to this work, as the SIP is the de-facto standard service and session management protocol for real-time telecommunication services, particularly for NGN/IMS-based telecommunication services.

There is a broad spectrum of standardized (typically standardized by the OMA) IMS-based telecommunication services, with varying complexities of required service architectures. As an example for rather complex services architectures, the Push-to-talk over Cellular (PoC) service makes use of several service enabling functionalities, including, presence service enablers, XML document management service enablers, media servers as well as the PoC server itself. Migration of a PoC service from one cloud infrastructure to another is a rather complex endeavor as session and presence states are needed to be maintained during the migration phase.

This work is to a lesser extend looking into those challenges that are related to service availability assurance of complex, state-full telecommunication services (during service migration processes). Rather than that, this work focuses on the efficient, service quality aware resource allocation for real-time telecommunication services.

AII.2.2 NGN Service Scalability Functions

As explained in section 2.8, scalability in clouds is typically achieved through horizontal scaling mechanisms (in contrast to vertical scaling mechanisms). Horizontal scaling implies that for each service dedicated systems for balancing the incoming workload across dynamically growing or reducing numbers of application serving instances must be provided. As explained in section 2.3.2, the direct invocation of an NGN services through the NGN service control layer involves IMS/SIP signalling, whereas the SOA/SDP-based orchestration of services, as explained in 2.3.3 involves SOA/HTTP-based service invocation.

Therefore for two types of load balancing functions are provided:

1) SIP load balancing systems, for purely IMS/SIP-based invocation of NGN services
2) HTTP load balancing systems for enabling SOA/HTTP-based access to NGN services

Both load balancing systems need to evenly distribute the workload across NGN service instances. Typically round-robin load-balancing algorithms, ideally supporting weighted load balancing are used for evenly distributing workload across service instances. It is worth noting that also here at the load-balancing layer certain optimizations can be applied, which can optimize the distribution / load-balancing of incoming load especially in cases of heterogeneous resource back-ends. In the case of heterogeneous back-end capacities,

weighted round-robin load balancing mechanisms / algorithms are required. In these cases, the load-balancer itself adapts the amount of workload allocated to a specific instance based on received feedback about the particular instance's resource utilization / load.[24]

Furthermore utilized load-balancing functions need to provide remotely-accessible provisioning interfaces.

AII.3 NGN Cloud Broker Functions

AII.3.1 NGN Cloud Broker - Initialization Functions

As explained in section 7.2.1 for Multi-Cloud NGN Service Lifecycle Management Initialization, the following service related information has to be provided by the NGN service provider (the user of the NGN Cloud Broker) to the NGN Cloud Broker:

- NGN Service Description
- QoS/SLA Constraints Description
- Cost Constraints

The initialization of the NGN Cloud Broker consists of several user-provided administration and configuration steps. It includes steps for configuring service related aspects as well as user-preferences including QoS/SLA constraints and cost constraints.

Service Descriptions include service control protocol, invocation methods, service dependencies (such as load-balancing in the simplest case). This information will later be needed i) to allocate, deploy and provision service components, ii) to configure/provision the service during resource instance deployment and VM start-up iii) to configure/provision the NGN session control layer as well as iv) the service platform-based orchestration engine.

QoS/SLA related configurations include network, resource performance / utilization thresholds and service related minimal performance requirements (if available), which might be part of available standards for telecommunication QoS requirements, as well as cloud infrastructure constraints like reliability, availability, location, security, trustworthiness constraints, which might be part of an initial SLA between service brokering service provider.

[24] As this additional complexity, however, did not provide additional insights into the actual performance of the system under investigation (the NGN Cloud Broker), the load-balancing architecture is simplified by only allowing homogeneous resources as back-end resources / application serving slaves. By doing so it was sufficient to utilize (un-weighted) round-robin algorithms of the load balancing systems, without limiting the general scope of this work.

Cost constraints, as initial parameters are also to be provided to the NGN Cloud Broker system by the user / administrator, which will later on define the policies and input parameters to platform selection and resource allocation algorithms.

Furthermore, also explained in 7.2.1 for Multi-Cloud NGN Service Lifecycle Management Initialization, the following resource related information has to be provided to the NGN Cloud Broker (however, the administrator of the NGN Cloud Broker is responsible for managing the cloud resource inventory, i.e. for providing this information):

- Cloud Resource Description
- Cloud Resource Costs and Pricing
- Cloud Platform Description

Finally for enabling service quality management and efficient resource allocation, as explained in 7.2.1, information about QoS dependencies and QoS impacting factors as well as cloud resource capacity related information need to be made available to the NGN Cloud Broker (however, this information should either be known a-priori, or provided by the administrator of the NGN Cloud Broker by means of benchmarking mechanisms):

- Cloud Resource Performance / Utilization vs. QoS
- Network Performance vs. QoS
- Workload vs. Cloud Resource Performance / Utilization

AII.3.1.1 NGN Cloud Broker - Service Registration Functions

According to the telecommunication *service model* introduced in section 7.1.1, the following service related information must be provided to the NGN Cloud Broker.

- Service Control Protocol (e.g. HTTP, SIP)
- Service Invocation Method (e.g. SIP INVITE)
- Service End-Point (e.g. IP, Port)

Note: Whereas the service control protocol and the invocation method should already be known by NGN Cloud Broker's initialization time, the service end-point is dynamically and repeatedly provided by NGN Cloud Broker during run-time.

According to the telecommunication *service QoS model* introduced in section 7.1.2, the following QoS related information must be registered in NGN Cloud Broker .

- Network QoS Requirements (e.g. Packet Delay, Jitter, Packet Loss, Bandwidth)
- Cloud Platform QoS Requirements (e.g. Availability)

- Service QoS Requirements (e.g. Service Execution Time)

Note: The QoS requirements of a given telecommunication service are either known at NGN Cloud Broker 's initialization time (based on experience, or based on telecommunication service standards) or identified by benchmarking mechanisms. For the latter, please refer to the section about Service Quality Model Establishment below.

According to the telecommunication *service interdependency model* introduced in section 7.1.3, the following service interdependency related information must be registered in NGN Cloud Broker.

- Service Dependencies (e.g. at application level, NGN service control and SDP level)
- Service Image Location (e.g. URL to service image)

Note: Whereas service dependencies should be known at NGN Cloud Broker initialization time, preparation of a virtual image that contains the actual service and making the latter available to cloud management systems needs to be conducted a-priori. Ideally, a pre-provisioned, packaged VM image or a location / URL to access such an image is being provided by the user / administrator. Alternatively, either partially automated image packing and creation mechanisms (e.g. taking formats like OVF as input), or manual image preparation and packaging mechanism can be used.

According to the telecommunication *service resource requirements* model introduced in section 7.1.4, the following service resource requirements related information must be registered in NGN Cloud Broker.

- Workload vs. resource consumption relationship (e.g. 50 calls/s ~ 80% CPU ECU)
- Virtual Resource Requirements (e.g. Type, Capacity)

Note: Whereas the virtual resource requirements of a given service should be known already before NGN Cloud Broker initialization time, information about relationship between service workload and capacity requirements either is available through past experiences, or needs to be found out through benchmarking mechanisms.

AII.3.1.2 NGN Cloud Broker - Cloud Platform and Cloud Resource Registration Functions

According to the *cloud infrastructure model* introduced in section 7.1.5, the following cloud infrastructure related information must be registered in NGN Cloud Broker.

- Cloud API (e.g. OCCI, OpenStack, Amazon), including end-point (e.g. URL)
- Optional: Location, Security, Reliability and Recommendation Information

Note: Whereas information about the actual cloud management API is mandatory for NGN Cloud Broker to interwork with a given public cloud infrastructure, additional non-functional information like location, recommendation might be important for specific services or specific users and might be the reason for non-applicability of a certain cloud infrastructure.

According to the *cloud infrastructure resource model* introduced in section 7.1.6, the following cloud infrastructure resource related information must be registered in NGN Cloud Broker .

- Virtual Resource Instance Types (e.g. micro, small, large)
    - Virtual Resource Instance Capacities (i.e. #CPUs, storage, network capacities)
    - Virtual Resource Instance Performances (i.e. compute performance in standard compute units)
- Virtual Resource Management Performance (e.g. start, update, stop delay)

Note: Whereas cloud infrastructure providers typically provide information related to the offered types of instances, including their capacities, there is yet no global standard for defining instance performances (e.g. the Amazon Elastic Compute Unit – ECU, is either not used by other cloud resource providers, or not sufficiently meaningful for specific applications).

According to the *cloud resource cost model* introduced in section 7.1.6, the following cloud resource cost related information must be registered in NGN Cloud Broker .

- Virtual Resource Costs
    - Long-term reservation costs
    - On-demand provisioning costs
    - Pricing model (e.g. min lease time)

Note: While information about long-term reservation and on-demand provisioning costs, including pricing models are typically be made available by cloud infrastructure providers, so called spot-market prices, which are subject to daily, even hourly change need to be made accessible through specific APIs.

The process of registering new platforms and new resources in NGN Cloud Broker

- Might include cloud infrastructure resource and load-based platform resource performance benchmarking with subsequent capacity computation.
- Mapping individual cloud infrastructure resource instances (micro, medium, large) to abstract, system-wide, cross-cloud domain compute, storage and network capacity metrics.

For NGN Cloud Broker to be able to select between a number of cloud infrastructure options a registry must be maintained, containing currently available cloud infrastructures, available resource types, resource costs, as well as additional information about location, recommendation, service reliability / service availability. The procedure of updating the cloud infrastructure repository can be conducted either manually or in a partially automated fashion. Weather done in a manual or partially automated fashion, for the overall performance and benefit that NGN Cloud Broker brings to an enterprise / service provider, frequent updates of the cloud infrastructure registry as well as a broad spectrum of entries is highly recommended (allowing to benefit from new offerings, options as well as to exclude critical platforms).

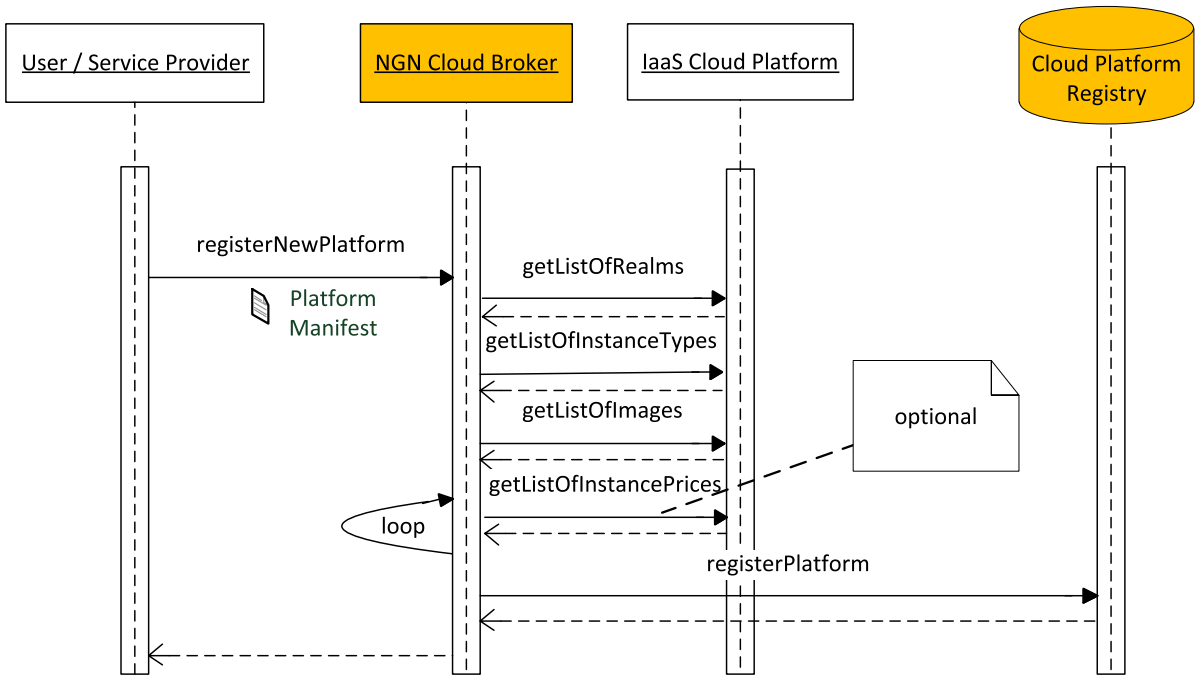

Figure 140: Cloud Platform and Cloud Resource Registration

Now that infrastructures have been listed in the registry, and their available cloud resource-types / including their specific costs have been listed in a cloud provider domain specific fashion in the registry, for NGN Cloud Broker to be able to compare resource capacities and costs of one cloud provider domain against offerings from another, a harmonized and unified metric for describing the capacity of a certain resource-type in a certain cloud provider domain. This will later become of importance during cost, capacity computation and resource allocation phase of NGN Cloud Broker.

AII.3.1.3 NGN Cloud Broker - Policy Registration Functions

Based on the considerations regarding policy-based management in section 2.1.2, NGN Cloud Broker's policies governing the resource allocation and platform selection process need to be

dynamically specified and registered. According to the policy information model introduced in section 7.1.8, the following policies have to be defined and registered in NGN Cloud Broker.

- Policies for Orchestration / Resource and Service Provisioning
    - Orchestration policies based service dependencies
- Policies for Platform Selection, Resource Allocation
    - Resource / service performance / QoS related policies
        - Service specific QoS target policies
        - Monitored metric's threshold policies
        - Policies for QoS computation
    - Platform selection process related policies
        - Platform performance related policies
        - Platform resource cost related policies

The process of registering policies in NGN Cloud Broker is illustrated in Figure 142. Based on service descriptions (incl. QoS, cost requirements) and resource descriptions (incl. capacities, performances and costs), policies are being generated, which govern the resource allocation, service orchestration and platform selection processes of NGN Cloud Broker.

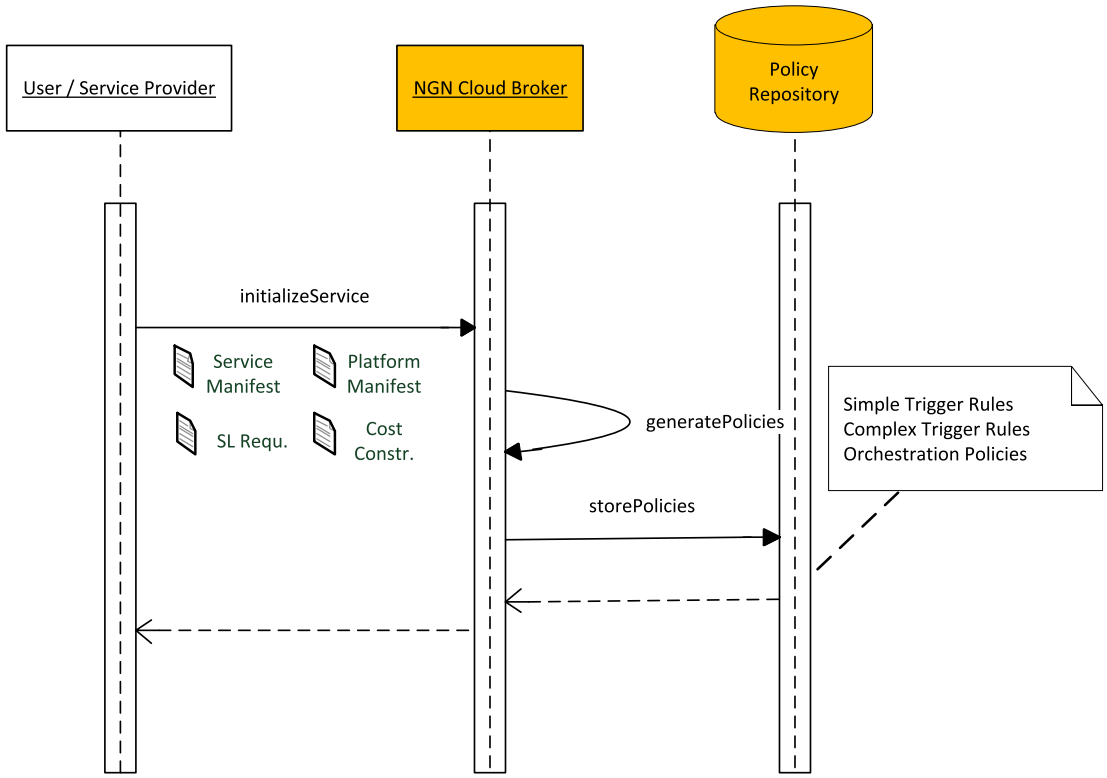

Figure 141: NGN Cloud Broker Policy Registration

AII.3.1.4 Cloud Resource and NGN Service Benchmarking and Quality Model Establishment

Network, resource and service specific parameters impacting the service quality in many cases are not known at NGN Cloud Broker's initialization time as described in above section AII.3.1.2 infrastructure and resource registration functions (*service specific QoS requirements*, *workload – QoS correlation*) and in section AII.3.1.1 service registration functions (*workload – resource utilization correlation*, *cloud resource performance*). For NGN Cloud Broker, i.e. the identification of appropriate policies, rules and thresholds, benchmarking / profiling mechanisms are used.

The service QoS model introduces in section 7.1.2 shows the factors that impact the End-to-End QoS. A de-composition of these QoS impacting factors is important for root cause analysis and counter measures. The service quality model establishment identifies all QoS impacting factors and for as long as they are measure-able, it specifies upper/lower critical thresholds. The NGN Cloud Broker itself enables the following two counter measures / fault resolution mechanisms 1) selection of an alternative cloud infrastructure and / or 2) provisioning of additional capacity. Nevertheless for dynamically selecting the currently optimal cloud platform, or in terms of failover management, for finding an alternative platform for a currently faulty / problematic platform, de-composition of QoS factors up-to network performance impacts is important.

Service Quality Establishment can be aided by telecommunication QoS standards, helping to identifying QoS impacting factors (e.g. network performance, resource performance, service performance factors) and their interdependencies. With the goal of defining upper or lower limits / boundaries of network, resource and service performances that are required to be maintained for assuring specific QoS classes of specific types of services. While there are NGN standards defining network performance objectives for IP-based telecommunication services, such as ITU-T standards [40], other impacting factors on the one hand strongly depend on the actual service, on the other hand on the performance of the actual resource. For establishing a holistic, service- and resource-specific service quality model, the following benchmarking / profiling techniques are employed:

- Service Quality Benchmarking

  By generating artificial workload, at discrete levels, the following relationships / correlations are determined:

  - QoS - Workload
  - QoS - Cloud Resource Performance / Utilization

  By emulating network performance parameters in a controlled environment, at discrete levels, the following relationships / correlations are determined:

  - QoS - Network Performance

- Cloud Resource Performance Benchmarking

  By generating artificial workload, at discrete levels, the following relationships / correlations are determined:

  - Cloud Resource Performance / Utilization - Workload

- Cloud Platform Performance Benchmarking

  By generating artificial workload, at discrete levels, the following relationships / correlations are determined:

  - Cloud Resource Performance / Utilization – Workload

The established service quality model provides the NGN Cloud Broker with thresholds and QoS relevant metrics required for assuring the QoS of a specific service. These policies are used for QoS, network and resource performance monitoring agent deployment and provisioning, and for the platform selection and resource allocation algorithms of the service brokering control loop.

AII.3.2 NGN Cloud Broker - Monitoring Functions

As explained in section 7.2.2, Multi-Cloud Resource Performance and NGN Service Quality Monitoring involves monitoring of QoS impacting factors as defined in section 7.1.2. In order to monitor the relevant network, resource and application performance metrics, monitoring agents are dynamically deployed at different layers, as shown in Figure 142.

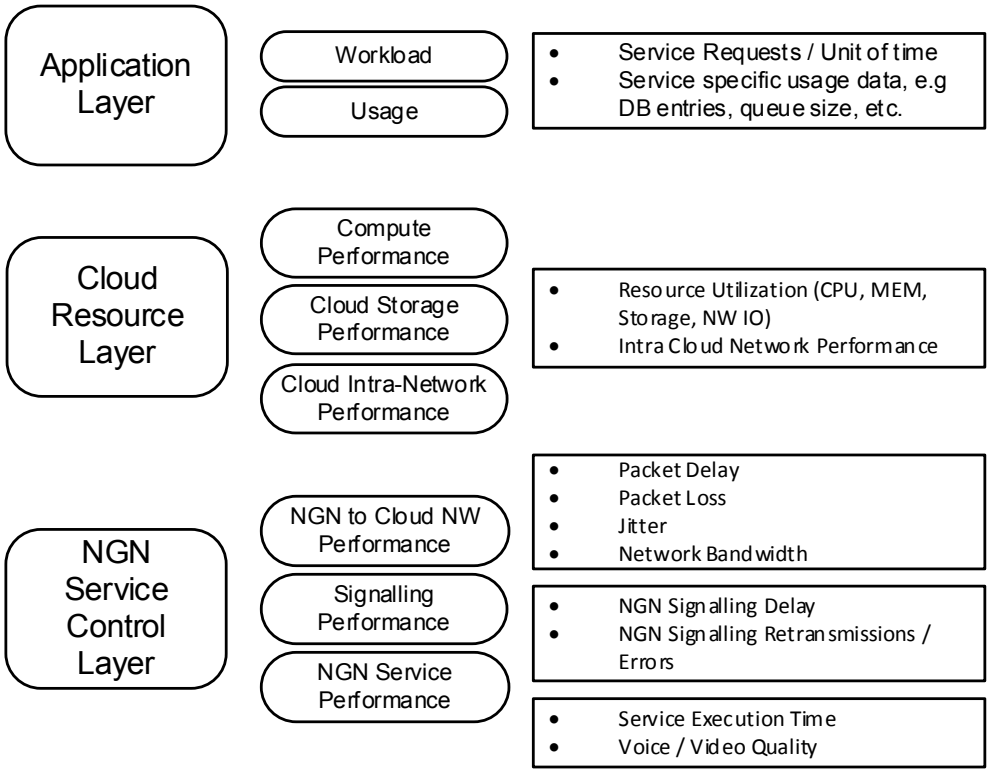

Figure 142: Layers for deployment of monitoring agents, measured performance parameters and metrics

In order to aggregate aforementioned QoS impacting performance metrics, the NGN Cloud Broker employs SOA-based orchestration of provisioning active and passive monitoring agents at the NGN as well as at the Cloud layer as shown in Figure 143. Whereas the NGN-based monitoring agents only need to be provisioned at initialization time, the Cloud-based agents are dynamically provisioned with each instantiated cloud resource. The different requirement of provisioning dynamism, relates to the fact that, the NGN-based monitoring agents do not need to be provisioned when NGN services change their location / are migrated from one Cloud platform to another.

For the passive NGN service control based agents, as well as for the active NGN user emulation agents, the actual NGN service identifier remains the same (this is why for NGN service users, the actual cloud brokering is completely transparent, identical to locally hosted NGN services). The NGN-based, NGN to Cloud network performance monitoring agents, also do not need to be re-provisioned if NGN services are moved between Cloud platforms, as they continuously monitor the network performance between the NGN and all available

Cloud platforms. Only for newly registered Cloud platforms, they need to be provisioned, in order to also start monitoring the network performance of the network connection between the NGN and the new platforms.

The dynamic provisioning of Cloud-based monitoring agents involves initialization and provisioning of monitoring agents inside the newly launched virtual machines. There are basically two ways of achieving this, 1) one-step provisioning of agents as part of the cloud resource provisioning request 2) two-step provisioning, where after the virtual machines are booted and reachable, in a consecutive step, the actual provisioning of monitoring agents takes place. One-step provisioning has several advantages. First, it significantly speeds up the process / reduces the time it takes from initialization of new cloud resources to monitoring cloud resource performance data. Second, it significantly reduces the networking complexity as cloud resources are not always reachable / accessible over the public internet, in which case NAT traversal, port forwarding or VPN or other mediation mechanisms need to be employed in order to provision monitoring agents / and other services after the startup of virtual machines. However, the one-step approach requires cloud management APIs allowing for so called contextualization of cloud provisioning requests. Within the contextualization, additional service specific information can be injected to the booting operating system of the virtual machine. In the case of dynamically provisioning monitoring agents, the following two parameters need to be provided to the monitoring agent:

- URL of monitoring aggregation function
- Metrics to be monitored

In the course of the booting process of the virtual machine, the contextualization information is read and used for provisioning applications launched at system start-up. The monitoring agents are initialized, register at the monitoring aggregation system and send the pre-defined monitoring data at the pre-defined frequency to the monitoring aggregation system. The monitoring data in subsequent steps of the NGN Cloud Broker is analyzed, used for planning consecutive resource allocations and cloud platform selection, which subsequently lead to the execution of further provisioning mechanisms, likely to involve the initialization or de-commissioning of monitoring agents, upon which a new loop is starting.

As with unified cloud computing interfaces such as OCCI, as introduced in section 2.7.1, as well as with cloud management platforms such as OpenStack and OpenNebula (introduced in section 2.12) supporting contextualization, it was refrained from specifying the two-step provisioning approach.

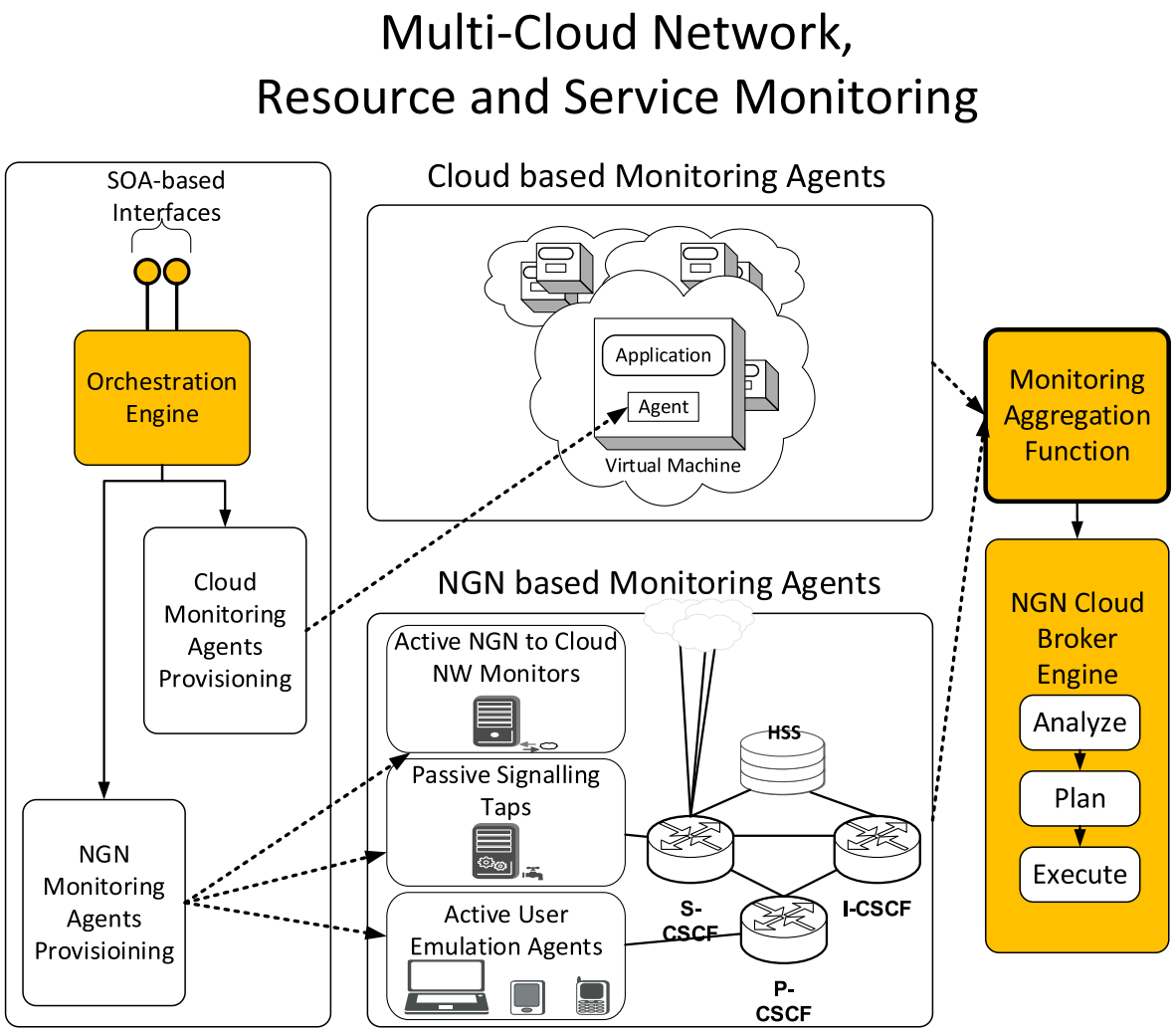

Figure 143: NGN Cloud Broker Multi-Cloud network, resource and service monitoring

AII.3.3 NGN Cloud Broker - Policy Evaluation Functions

For providing enhanced flexibility and configurability to the NGN Cloud Broker, allowing rapid definition of new service and resource specific policies exploiting the benefits of policy-based management as introduced in section 2.1.2 policy evaluation functions are integrated into the overall architecture. In principle, there are two policy evaluation and enforcement modes, shown in Figure 144 where the evaluation of policies is triggered based on local and global monitoring information.

In the first mode, the instant evaluation and enforcement mode, directly after aggregating monitoring data (we term monitored “singletons”, i.e. monitoring data of single elements, like VMs, single network performance monitoring data), a policy evaluation process is conducted in order to rapidly check, whether current states are acceptable (within acceptable range or violating thresholds).

In the second mode, policy evaluation is conducted *inside a MAPE cycle* of the NGN Cloud Broker, which imposes higher latency to re-act on critical states, as single cycles are carried out at significantly lower frequencies as monitoring information aggregation. In this mode, the analysis functions trigger the evaluation of typically more complex rules, since the analysis functions already conduct an analysis of multiple, system-wide monitoring information, e.g. the overall workload of all cloud instances on a cloud platform, all QoS impacting factors (i.e. network, platform, resource performance), all network performances of the connectivity between NGN and all used and available cloud platforms, etc. It is also here that the planning function triggers the evaluation of orchestration policies. This can, however, also happen as a result of events triggered by the first policy evaluation mode, e.g. where, in case of a sudden cloud platform outage, or significant network deterioration, orchestration policies are evaluated for platform selection.

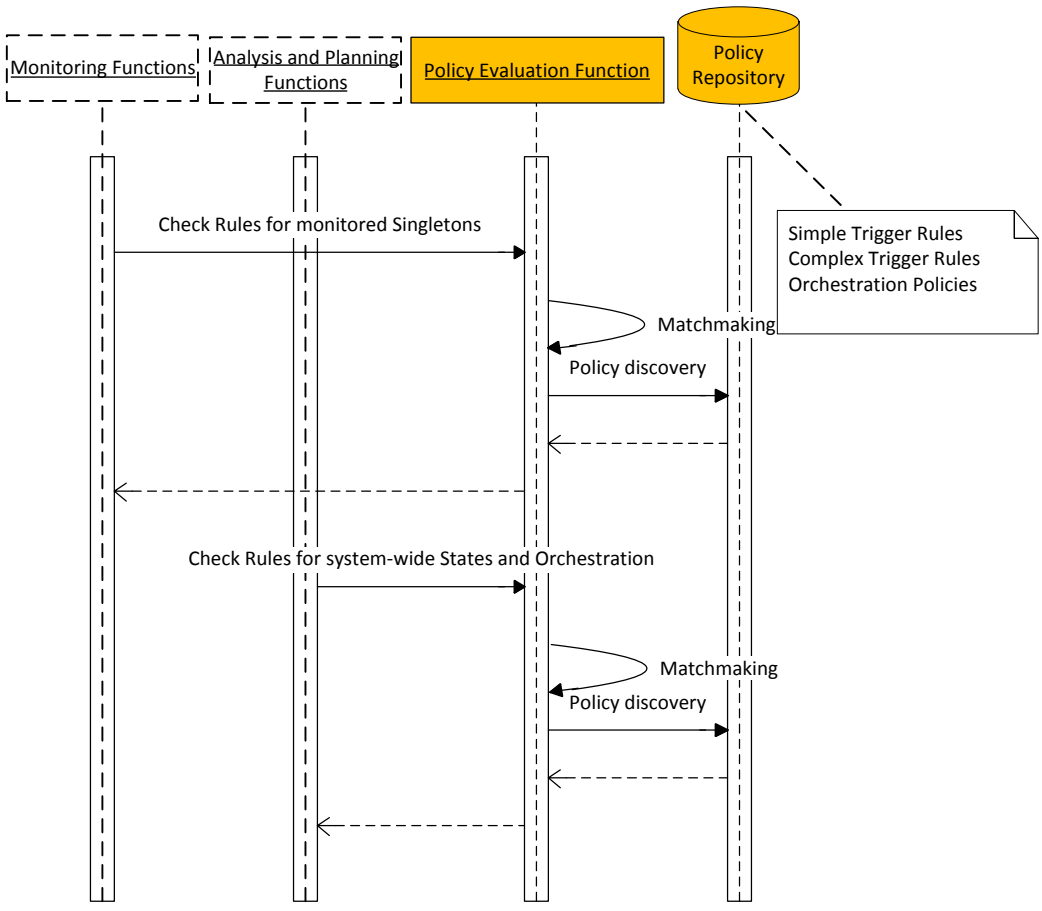

Figure 144: NGN Cloud Broker Policy Evaluation Function Call Flow

AII.3.4 NGN Cloud Broker - Service Orchestration Functions

As explained in 7.2.5, Multi-Cloud Resource Provisioning, Service Configuration and Activation Execution involves provisioning of cloud resources, service components, including

monitoring and loadbalancing components, NGN systems from service control to service delivery platform systems and the final provisioning of cloud usage related reporting data.

The service orchestration functions of the NGN Cloud Broker, as depicted in Figure 145, differentiate two intrinsically different resource allocation and service provisioning workflows:

A) Initial, full allocation and provisioning workflow upon selection of new or alternative cloud platforms
B) Subsequent, partial allocation and provisioning workflow for already provisioned cloud platforms (i.e. NGN service are deployed and already in operation)

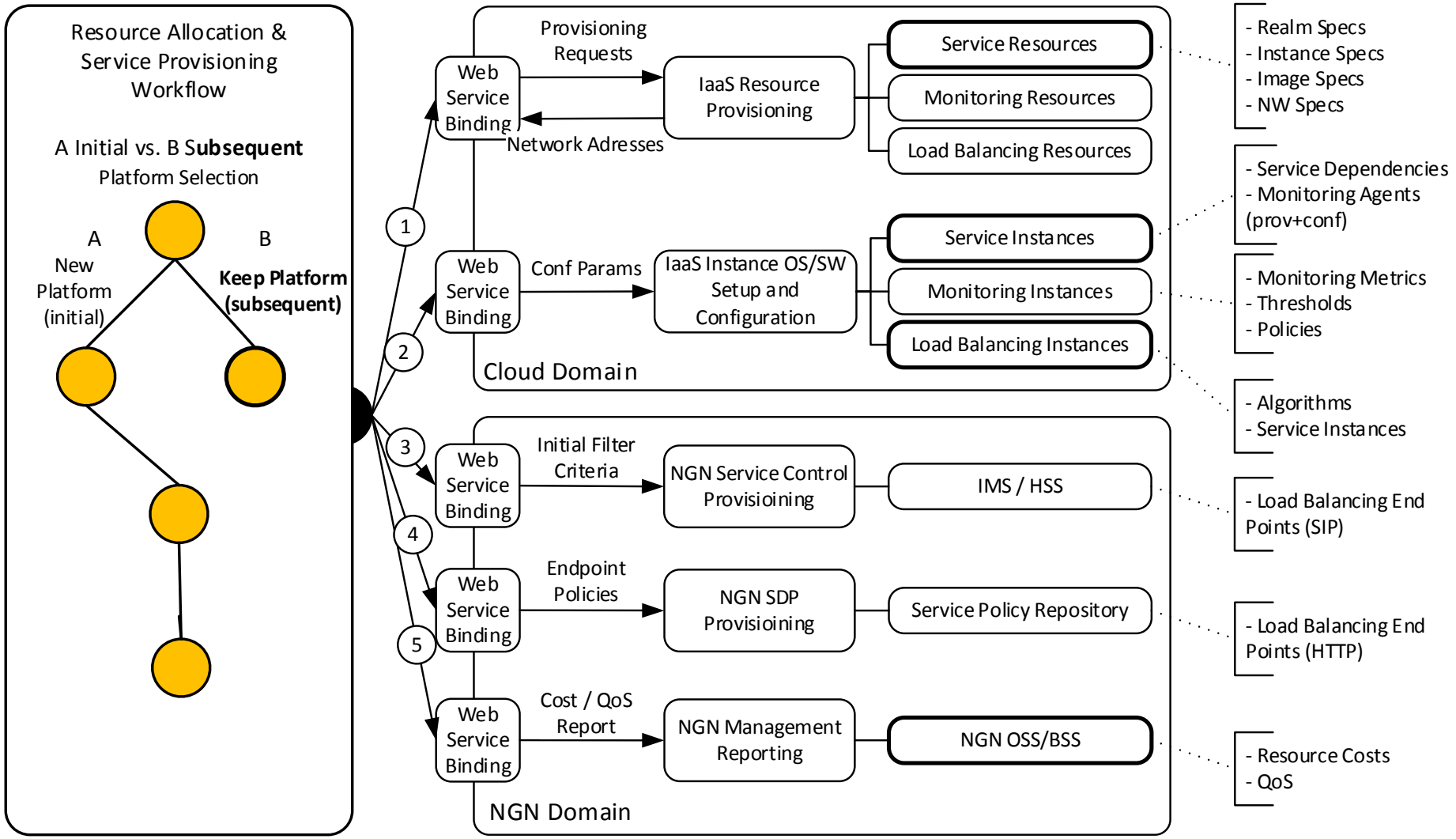

Figure 145: Initial and subsequent Cloud and NGN Provisioning Workflow

By modeling and realizing its management processes in a service oriented fashion, the NGN Cloud Broker adheres to the SOA paradigm. The NGN Cloud Broker's resource allocation and service provisioning lifecycle is realized as a SOA-based workflow consisting of loosely-coupled service invocations, which enables re-use and composition for different purposes, allowing flexible service chaining / provisioning service composition through which provisioning steps required by different services can be easily incorporated.

Whereas for the initial provisioning workflow, resources and services in the cloud domain as well as service in the NGN domain are provisioned, subsequent provisioning workflows only need to provision a subset of resources and services. This differentiation, however, is partially based on the assumption that neither monitoring aggregation, nor load-balancing components need to be scaled during service operations, which in some, heavy load, massive scale scenario is an invalid assumption. For overcoming this limitation however, the NGN Cloud Broker's SOA-based workflow approach can easily support also scalability of these service supporting functions (monitoring aggregation, load-balancing functions), which can be realized in the same way as the scalability of the actual service is realized, as shown and focused in this work.

The steps of the *full resource allocation and service provisioning workflow* are:

1) IaaS Resource Provisioning

   As described in section AII.1.4, either by making use of the native IaaS cloud provisioning APIs or through mediation functions as described in section 2.7.1, the actual on-demand cloud resource provisioning process is conducted through multiple resource provisioning requests, which specify the realm (i.e. the location of the cloud data centre if there are multiple locations), the instance type (i.e. provider specific size of small to large mixes of compute, storage and network resources), the image (i.e. the pre-configured OS that runs on the cloud instances) and the required network address allocation (i.e. private or public addresses for each specific instance).

   The network addresses of Cloud instances, weather private or public, usually allocated dynamically, are returned to the NGN Cloud Broker workflow logic for further provisioning and configuration steps.

   Whereas in subsequent workflows (i.e. the cloud platform is already in use), only resources for the actual service instances are being provisioned, the initial workflow requires deployment of additional service supporting functions. On the one hand, for each cloud domain one monitoring aggregation function needs to be allocated, which aggregated the monitoring data of monitoring agents allocated within each service instance. On the other hand, for each cloud platform at least one load-balancing function needs to be allocated, which serves as the core function for horizontal scaling of cloud-based service instances. Both, the monitoring aggregation functions, as well as the load-balancing function, need to be externally accessible for remote access, thus a public network address needs to be allocated for both functions.

2) IaaS Instance OS/SW Setup and Configuration

After the service and service support functions have been provisioned and started (or stopped in case of down-scaling) in step 1), subsequent configuration of new instances and of supporting functions have to be conducted.

For the newly provisioned service instances, service depending components have to be provisioned (e.g. in order to connect to other service components such as databases). Additionally, within each service instance a monitoring agent needs to be initialized and provided with the monitoring aggregation functions' network address for subsequent monitoring data provisioning.

For newly provisioned monitoring aggregation instances, service specific configurations of metrics (to be monitored), thresholds (for each metric) and policies for re-active incident resolution mechanisms need to be configured.

For load-balancing instances, the load-balancing algorithms and the current set of service instances' network addresses (usually private) need to be provisioned.

It should be noted that some IaaS provisioning APIs provide IaaS instance configuration *within* the actual resource provisioning request, known as "contextualization" (see also section AII.3.2, for contextualization of monitoring agents). Therefore, step 2) can either be integrated in step 1) or (as described here) issued as subsequent provisioning / configuration requests after service instances have been provisioned and started up.

3) NGN Service Control Provisioning

For the initial provisioning workflow, after cloud resources are allocated and cloud-based services have been provisioned, the NGN service control layer needs to be provisioned. Provisioning of the NGN service control layer, as described in AII.1.1 involves setup and configuration of iFCs in the HSS, providing Service Trigger Point and Application Server SIP URL. In the case of the proposed system, the Application Server URL is the public network address of the cloud-based load balancing function.

4) NGN SDP Provisioning

For the newly provisioned service to become available also at the NGN SDP level, additional provisioning of SDP-based service registries, as described in section AII.1.2 needs to be carried out. Typical SOA-based SDP registries are XML-based registries supporting the Universal Description, Discovery and Integration (UDDI) specification. Typically, a SIP-based service either provides an additional Web Service interface, or the SDP itself provides such Web Service (SOAP/REST) – SIP mediation functions. Whatsoever solution is provided, in the end, the SDP-based registry maintains a list of Web Service end-points of services. Therefore, the XML / Web Service description needs to be fed into the SDP-based service registry, which is a comparably straight

forward task, as the entire SDP already exposes Web Service interfaces, ready to be utilized.

5) NGN OSS/BSS - Cloud Resource Usage Reporting

After NGN Cloud Broker allocated resources in step 1) and activated the service through provisioning steps 2) - 4), the service is ready to be consumed. Typically no further steps are required for service execution, unless the service is entirely new to the portfolio of telecommunication service provider, needs to be advertised and subscribed to by telecommunication service users. These BSS processes are not in the scope of the work. Nevertheless, *NGN Cloud Broker's key performance parameters, 1) the QoS and 2) resource consumption / allocation / costs* are of central relevance and to the OSS and BSS of a telecommunication service provider as 1) allows the service provider to identify SLA violations (and conduct appropriate measures on the OSS and/or BSS level) and 2) is of importance for assessing current OPEX.

Service Decommissioning

As described in section 7.2.6, the full service management lifecycle finally also includes the de-commissioning of the service, brokered by the NGN Cloud Broker. For de-commissioning of the service the following orchestration steps are required.

- NGN Service Control un-provisioning
- Resource un-provisioning
- Monitoring data un-provisioning
- Policy un-provisioning

Service decommissioning represents a roll-back of the service orchestration cycle. As shown in Figure 146 the NGN service control layer elements are provisioned first, for user requests to be blocked / not anymore forwarded to cloud-based NGN service instances soon ceasing to exist. Subsequently, cloud infrastructure resources are released. Finally service management elements, monitoring elements, and service artifacts in NGN Cloud Broker s' registries and repositories are removed. Finally the actual orchestration script can be removed from the orchestration system.

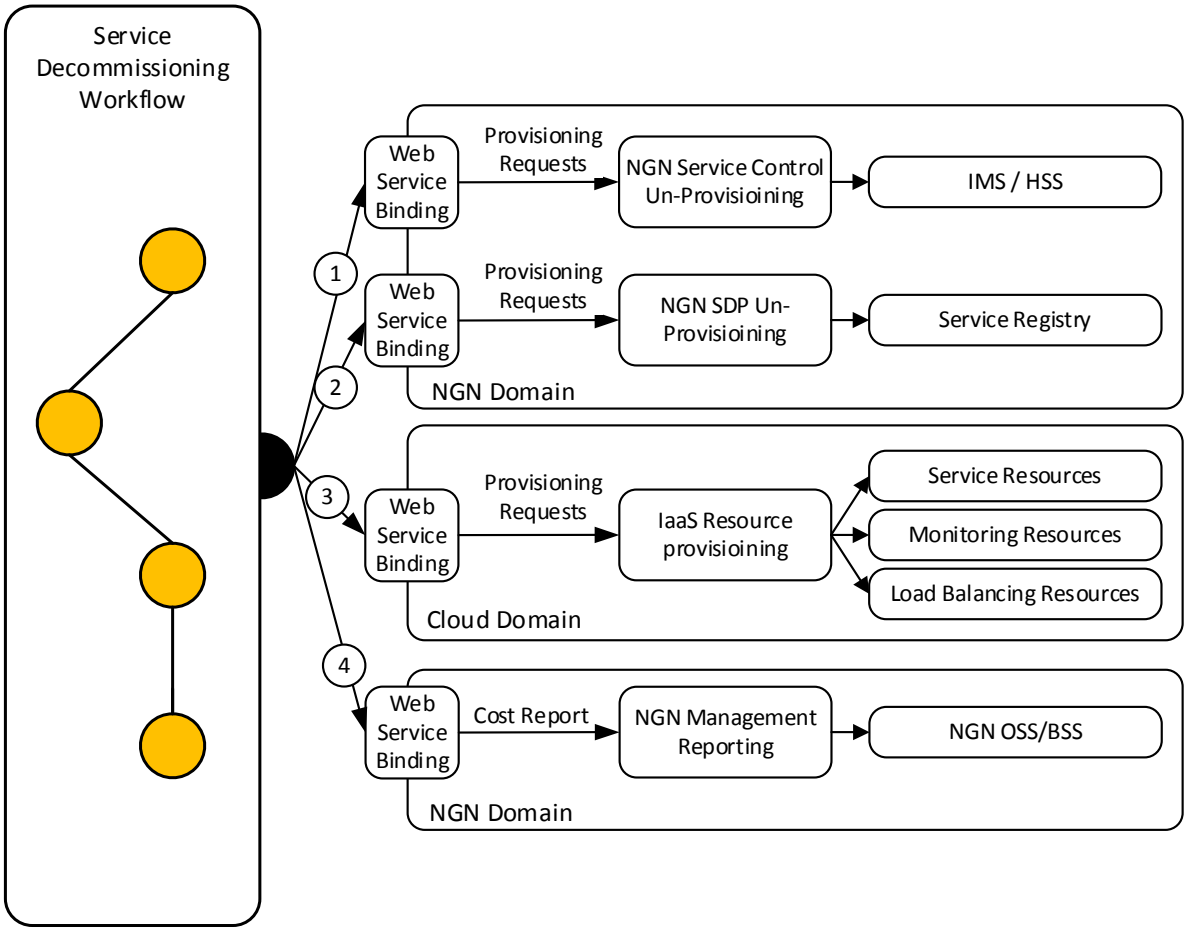

Figure 146: Service Decommissioning Workflow

Important for assuring that no further service requests are being served (and consequently interrupted/aborted) during the service decommissioning workflow is to first disable NGN-based service admission, before shutting down cloud resources. This assures that no more service requests are accepted and "gracefully" rejected and not aborted during service invocation or session. After un-provisioning the NGN layer, a "grace"-time is given for currently served sessions for finishing before conducting the subsequent service de-commissioning steps.

AII.3.5 NGN Cloud Broker - Cloud Brokering Functions

The NGN Cloud Broker's core brokering functions, are analyzing the current resource performance and NGN service quality and planning resource performance and NGN service quality improvement steps.

As explained in section 7.2.3, Multi-Cloud Resource Performance and Service Quality Analysis involves analysis of cloud resource performance, cloud platform performance, network performance between NGN and Cloud platforms and NGN service quality performance analysis.

And as explained in section 7.2.4, Multi-Cloud Resource Performance Control and QoS Improvement Planning involves capacity planning, planning of cloud resource allocations (taking into account cost calculations) and selection of optimal cloud platforms.

In contrast to the NGN Cloud Broker's analysis phase, which analyzes latest network, resource and service performance and workload and evaluates respective policies, as shown in Figure 147, the planning phase does not only need to take into account policies for re-acting on violations of certain thresholds, as this would be a purely re-active strategy only. For optimizing resource consumption while assuring QoS, the planning phase surpasses purely re-active control mechanisms, employing workload-prediction mechanisms in order to forecast required resource capacities. By utilizing service quality models as well as active end-to-end QoS measurements an optimization between QoS assurance and capacity saving (directly impacting costs) is sought to be optimized.

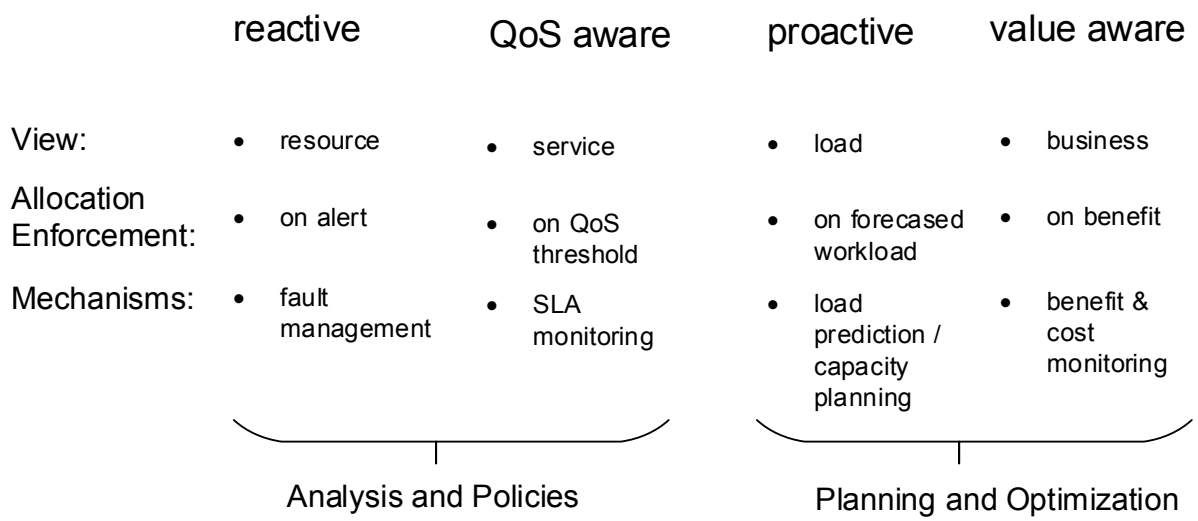

Figure 147: Reactive analysis and policy-based and proactive value aware resource allocation planning

As shown in Figure 148, the analysis and planning functions of the NGN Cloud Broker, receive the input from the NGN Cloud Broker's monitoring functions (described in section AII.3.2), benchmarking (described in section AII.3.1.4) and policy information (described in section AII.3.3). Based on the analysis and planning functions, the NGN Orchestration functions (described in section AII.3.3) are triggered, executing the required provisioning of resources, services and service dependent elements. This marks one complete cycle of the continuously executed control loop of the NGN Cloud Broker.

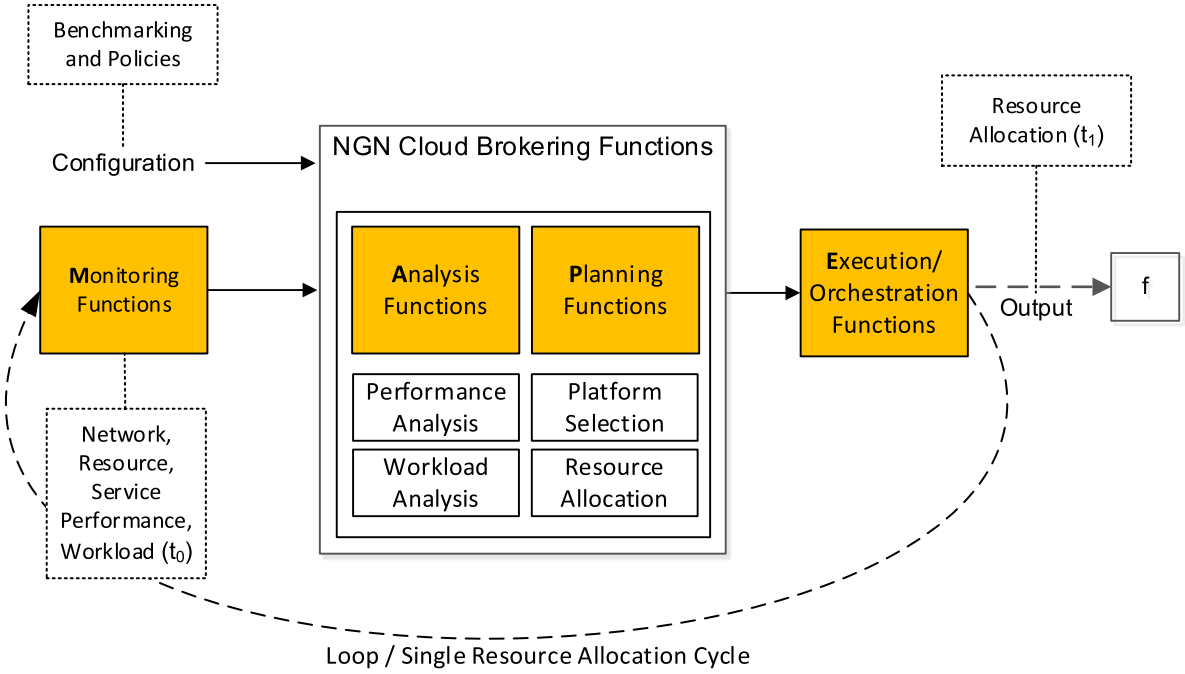

Figure 148: NGN Cloud Brokering Loop Input, Output and Core Brokering Functions

The goal of each control loop is to identify the current cost-optimal cloud platform and efficient (re-) allocation of cloud resources under specific constraints, of which QoS constraints are the main constraint focused in this work.

Whereas monitoring agents are contiguously sending data to the monitoring aggregation function, at the start of each new loop, NGN Cloud Broker requests the latest data for a specific service from the aggregation function upon which the following data is analyzed and evaluated against respective policies / thresholds.

- Workload
- Network, Platform, Resource performance data
- Resource utilization
- Service Quality

During the analysis phase, system-wide as well as instance specific utilization of resources, network performance, workload and service quality data is smoothed and averaged, and provided as input for the planning functions.

The planning phase, as depicted in Figure 149, consists of 1) workload prediction and capacity forecasting, 2) network performance analysis, 3) cloud platform filtering step, 4) optimal platform selection and resource allocation, based on which cloud resources are (re-) allocated and potentially migrated in the subsequent execution phase of the overall control loop. Certain performance indicators are recorded during the execution of each cycle, which are 1) the full duration of each cycle, as well as 2) the cloud platform provision duration (as

this provides vital information about the performance of each specific cloud platform). These performance measures define the maximum execution frequency of each cycle, are used to determine the performance of a cloud infrastructure platform (i.e. including the virtual infrastructure management system) and determine the look-ahead step / event horizon required for subsequent capacity forecasting steps.

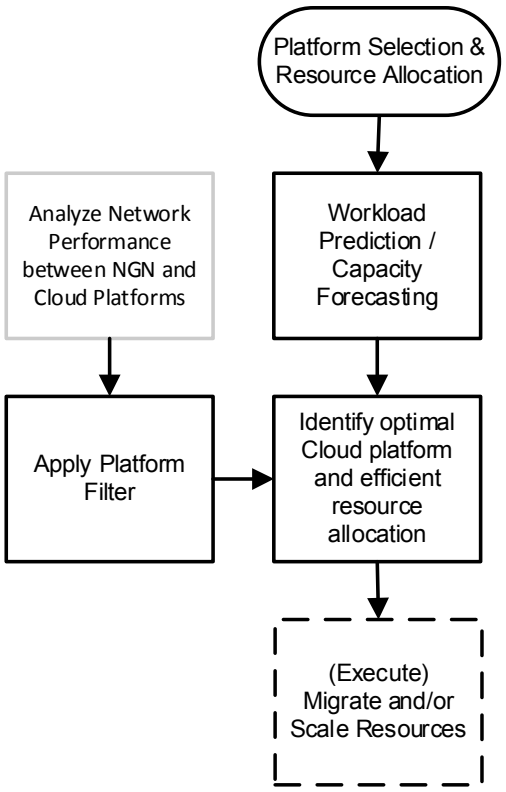

Figure 149: Platform Filtering and cost-optimal Resource Allocation

The core planning algorithms of NGN Cloud Broker, the platform selection algorithm and the capacity estimation / resource allocation algorithm are executed as described in sections AII.3.5.1 and AII.3.5.2.

AII.3.5.1 NGN Cloud Broker - Cloud Platform Selection Functions

For continuously selecting a cloud platform, which at a given point in time, 1) provides sufficient performance / service levels for delivering a specific NGN service 2) the lowest costs and 3) comply with further user-defined constraints, the following parameters of a given cloud platform are continuously analysed and evaluated.

Primary QoS-related Cloud Platform KPIs / filter criteria:

- Network Performance
- Cloud Platform Reliability / Availability
- Cloud Platform Instance Provisioning Performance

The initial phase of the cloud platform selection process applies a simple filtering mechanism to the list of available cloud platforms which satisfy cost and QoS constraints, as depicted in Figure 150. The result is a short-listing of eligible cloud platforms.

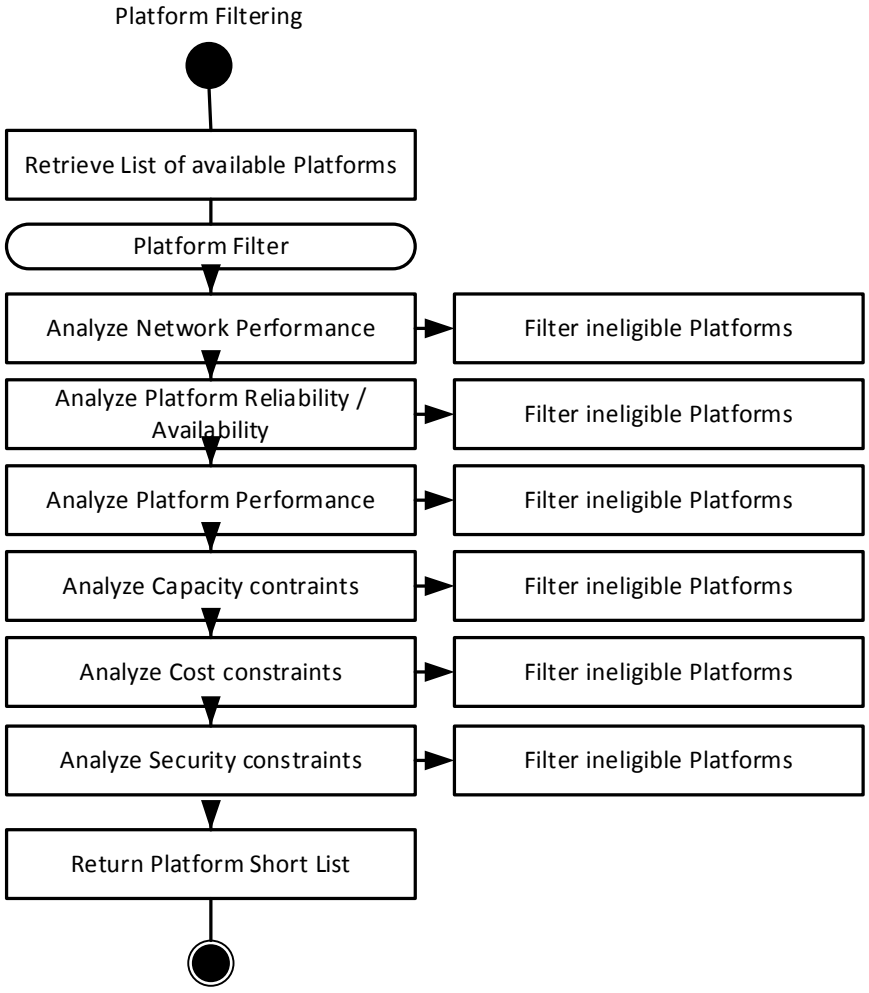

Figure 150: Cloud Platform Filtering

Whereas certain factors like cost or reliability, if within some pre-defined boundaries (limited by max/min values / policies), other factors like trust / security or location might be knock-out criterions. Whereas some cloud platforms need to be filtered and dropped from the list of currently eligible Cloud IaaS platforms, others are ranked according to pre-defined weights. A score is being calculated for each available IaaS Cloud platform, which determines their position in the ranking table.

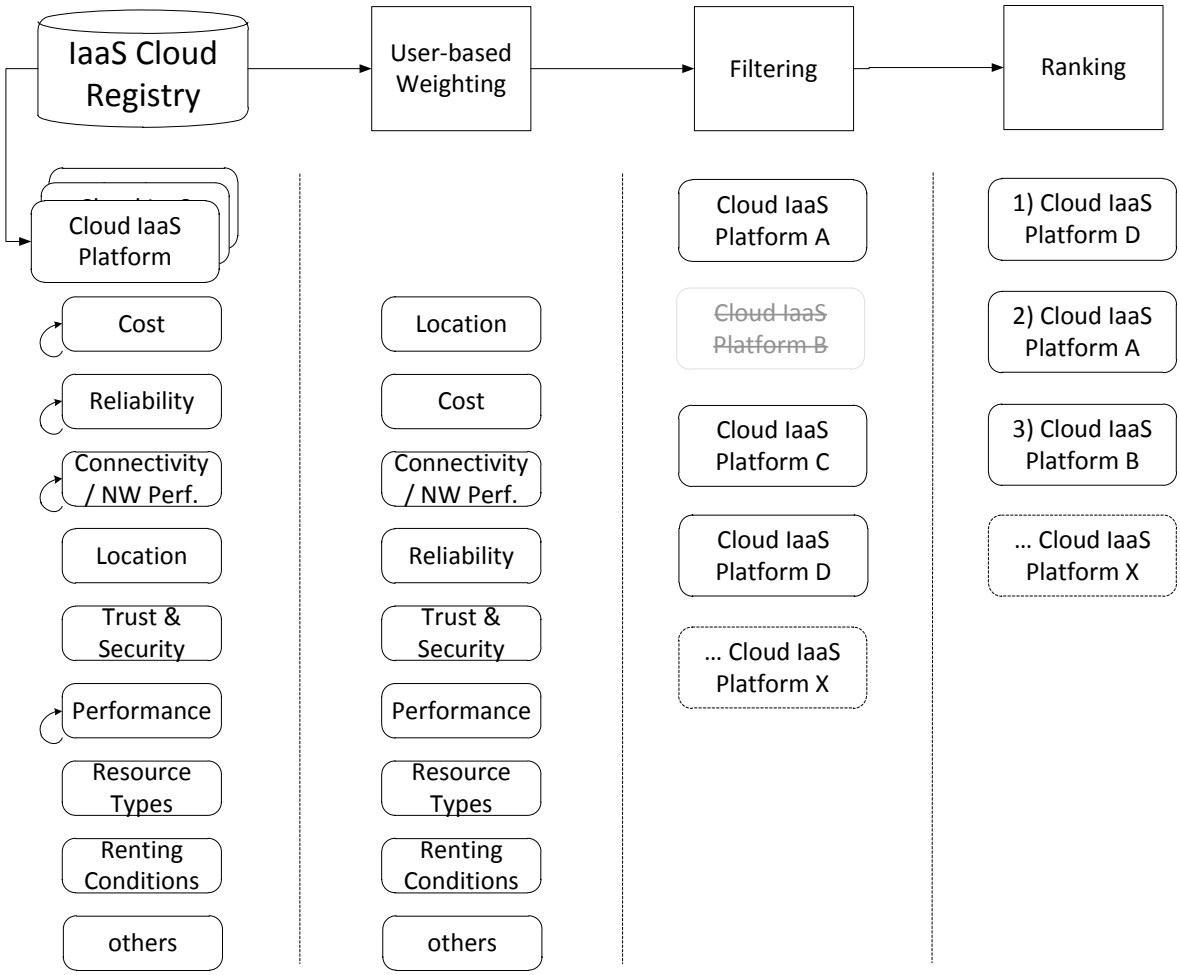

Figure 151: Cloud Platform Ranking

In [19] the cloud platform ranking algorithm as a multi-criteria decision making (MCDM) process (as introduced in section 2.9) is described. For dynamically ranking cloud platforms, users of the NGN Cloud Broker are required to define a set of specific KPIs, such as QoS related KPIs, or cost related KPIs, to provide specific weights, as a measure of importance of these KPIs in relation to each other. K_j is defined as KPIs of a cloud platform and λ_j as the weight associated to each specific KPI. The overall sum of all weights is

$$\sum_{1}^{m} \lambda_j = 1$$

Based on these parameters, the NGN Cloud Broker is dynamically ranking eligible cloud platforms at each given point in time. For normalizing the different KPIs for creation of a create ranking table, V_{ij} is defined as the normalized value for a KPI K_j.

For KPIs for which lower values represent better KPIs ($KPI-$) the following definition applies:

$$NV_{ij} = \frac{max\ - V_{ij}}{max} \lambda_j\ (for\ KPI-)$$

Here, max represents the maximum acceptable value for these KPIs.

For KPIs for which the greater values are better ($KPI+$) the following definition applies:

$$NV_{ij} = \frac{V_{ij} - \min}{V_{ij}} \lambda_j\ (for\ KPI+)$$

Here, min represents the minimum acceptable value for these KPIs.

Based on this, a new table is created which maps Cloud Platforms to KPIs according to the normalized values. For ranking cloud platforms, T_i is defined as the score of the i_{th} cloud platform. The top ranked cloud platform at a given point in time is determined by the maximum score T_i, i.e. highest values for those KPIs for which high values are better or lowest values for KPIs where low values optimal. By defining with KPI+ the first, and by KPI- the latter, the following is defined:

$$T_i = \sum_{j=1}^{m} VN_{ij}$$

AII.3.5.2 NGN Cloud Broker - Capacity Forecasting and Resource Allocation Functions

While cloud platform filtering and ranking mechanisms as described above, are utilized for creating a ranking table of eligible cloud platforms fulfilling the core constraints, the efficient resource allocation process, selecting cost-optimal cloud resource and auto-scaling the latter is explained in the following sections.

Workload Analysis and Capacity Forecasting

As an initial step, the current and historical workload data, globally as well as on individual cloud instances is analyzed, based on which the required capacity expected by the end of the resource allocation cycle is forecasted, as shown in Figure 152.

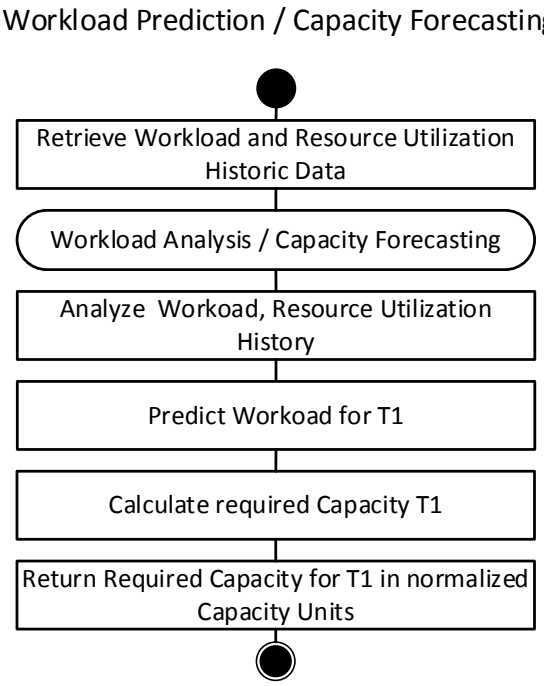

Figure 152: Workload Prediction and Capacity Forecasting Sequence

For workload prediction and capacity forecasting the NGN Cloud Broker uses a second order autoregressive moving average (ARMA) model, as described in [171], on historical workload data.

With parameters $\varphi_1, \ldots, \varphi_p$ and $\theta_1, \ldots, \theta_q$ and constant c and random white noise variable ε_t the auto regression (AR) model of order p (according to [171]) is denoted as:

$$X_t = \mathrm{c} + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i}$$

The moving average (MA) model of order q (according to [171]) is denoted as:

$$X_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

Together, the ARMA (p, q) (according to [171]) is denoted as:

$$X_t = \mathrm{c} + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

One-step linear forecasting of the AR process (according to [171]) is denoted as:

$$\hat{X}_{t+1} = \sum_{k=1}^{p} \varphi_i X_{t+1-k}$$

One-step linear forecasting of the MA process (according to [171]) is denoted as:

$$\hat{X}_{t+1} = \sum_{i=1}^{q} \theta_i \epsilon_{t+1-k}$$

Cloud Resource Allocation

For the determination of cost-optimal allocation of cloud resources, according to and as a simplification of [27], the following parameters are defined:

1) Eligible Cloud Platform's {cp1, cp2,…,cpm} instance types, with l := number of instance types = {IT1, IT2,…,ITl}
2) Eligible Cloud Platform's Instances' Capacities CCITj, where 1 ≤ j ≤ l := Compute Capacity of Instant Type l
3) Eligible Cloud Platform's Instance Type real Costs (by mapping benchmarked capacity of a cloud instance to costs of a cloud instance), Pjk := Hourly price for leasing instant CCITj in cloud platform cpk
4) Migration Costs (This cost is a combination of various factors such as cost of SLA violations, leasing cost of resources and a cost associated with the changes to the configuration.

For each combination, the algorithm computes:

$$\boldsymbol{Total\ Infrastructure\ Capacity -\ TIC_k} = \sum_{i=1}^{n} CCIT_j$$

which needs to be greater or equal to the required capacity and

$$\boldsymbol{Total\ Infrastructure\ Price -\ TIP_k} = \sum_{i=1}^{n} P_j$$

which is used to determine the cheapest allocation of cloud resources.

The overall resource allocation process of the NGN Cloud Broker is depicted in Figure 153. After the required capacity and the short list of eligible cloud platforms is taken as input, by comparing the currently allocated capacity with the forecasted capacity the process identify whether up-scaling or down-scaling of cloud resource capacities is required. While the down-scaling process needs to take care of potential under-provisioning of capacities, the up-scaling process can directly proceed to the following step of evaluating whether the migration of NGN services to alternative cloud platforms (if a new cloud platform is determined to provide increased cost-optimality) provides sufficient gains compared to remaining with the currently used cloud platform. This evaluation might depend on several

factors, of which the leasing schema (e.g. minimal time of leasing cloud resources) of a particular cloud platform provider might be of core relevance.

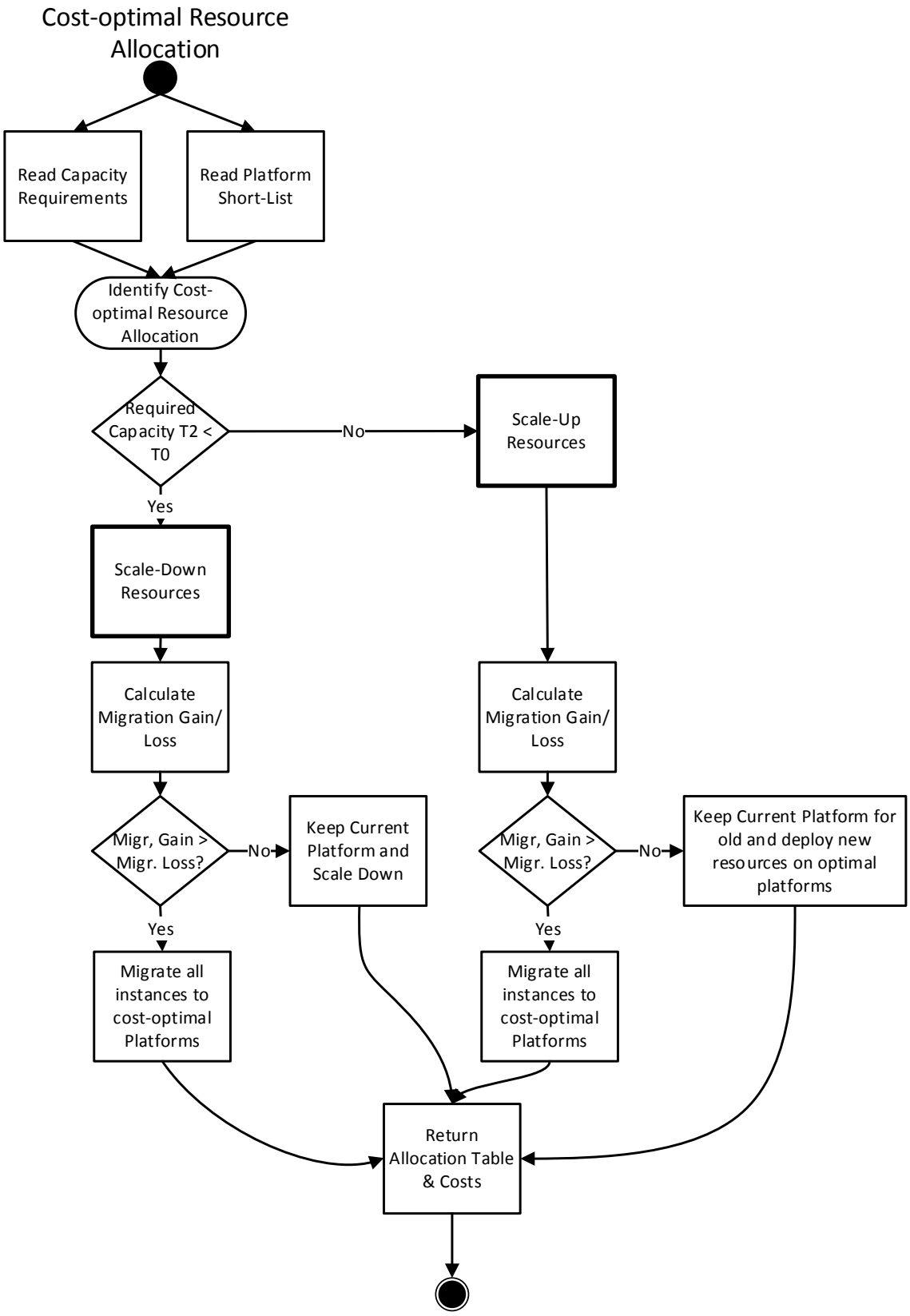

Figure 153: Cost-optimal Resource Allocation

Cloud Resource Up-Scaling

As described in [18] and [20], the overall up-scaling workflow is depicted in the sequence diagram in Figure 154. After retrieving the current utilization of single resources, the overall system utilization is calculated by the NGN Cloud Broker's analyzer. Based on policies defining the maximum allowed system utilization, the required capacity for serving the forecasted workload is computed by the NGN Cloud Broker's planner. After computing the cost-optimal allocation of cloud resources, the planner sends a request to deploy N additional cloud instances, or to instantiate a full set of cloud resources on an alternative cloud platform to NGN Cloud Broker's execution component / the orchestration function. The orchestrate takes care that cloud resources are provisioned, dependent services at Cloud level (e.g. load-balancers), NGN service control and NGN service delivery platform level are provisioned.

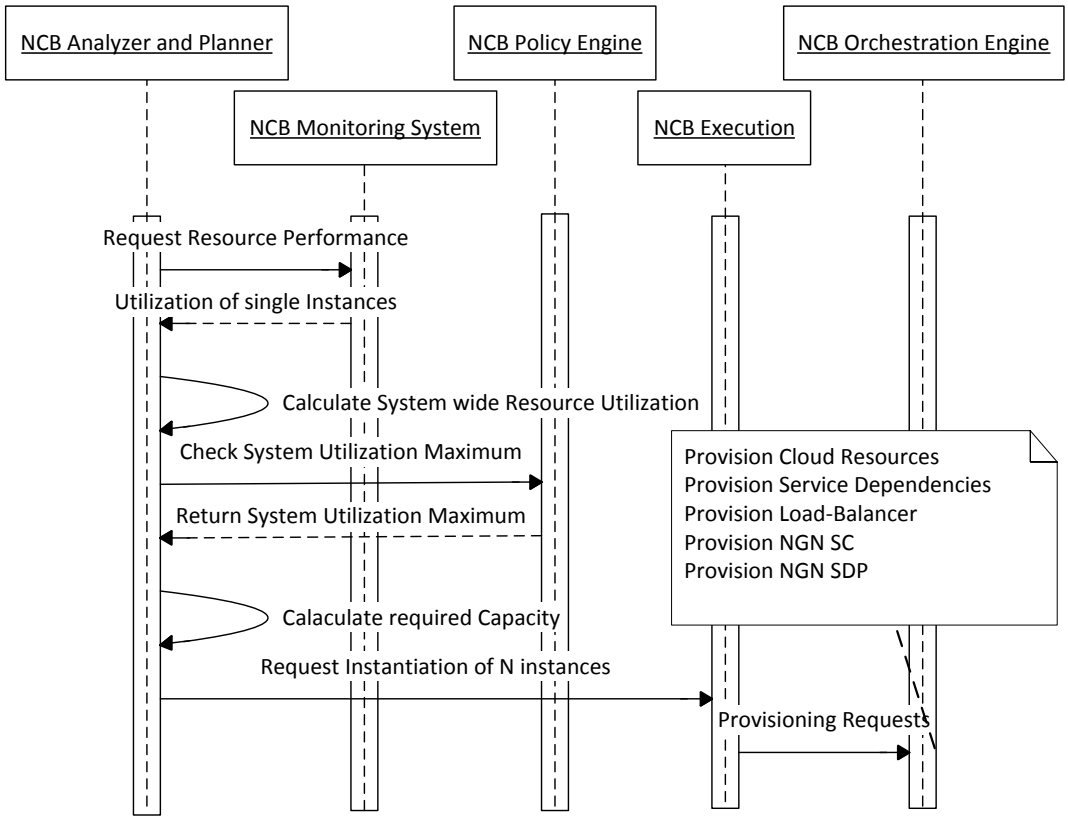

Figure 154: Resource Up-Scaling Sequence

Resource Down-Scaling

As described in [18] and [20], the overall down-scaling workflow is depicted in the sequence diagram in Figure 155. After retrieving the current utilization of single resources, the overall system utilization is calculated by the NGN Cloud Broker's analyser. Based on policies defining the minimal allowed system utilization, the required capacity for serving the forecasted workload is computed by the NGN Cloud Broker's planner. However, the NGN Cloud Broker needs to assure that removal of Cloud instances will not bring the overall system into a state of overload. Therefore the identification of obsolete instances assures that sufficient capacities will be available after release of such resources. After computing the cost-optimal allocation of cloud resources, the planner sends a request to remove the identified cloud instances. The orchestrate takes care currently ongoing sessions can be successfully finished, removing the obsolete cloud instances from the load-balancers first, waiting for a certain grace-period before obsolete cloud resources are decommissioned and dependent services at Cloud level (e.g. load-balancers), NGN service control and NGN service delivery platform level are provisioned.

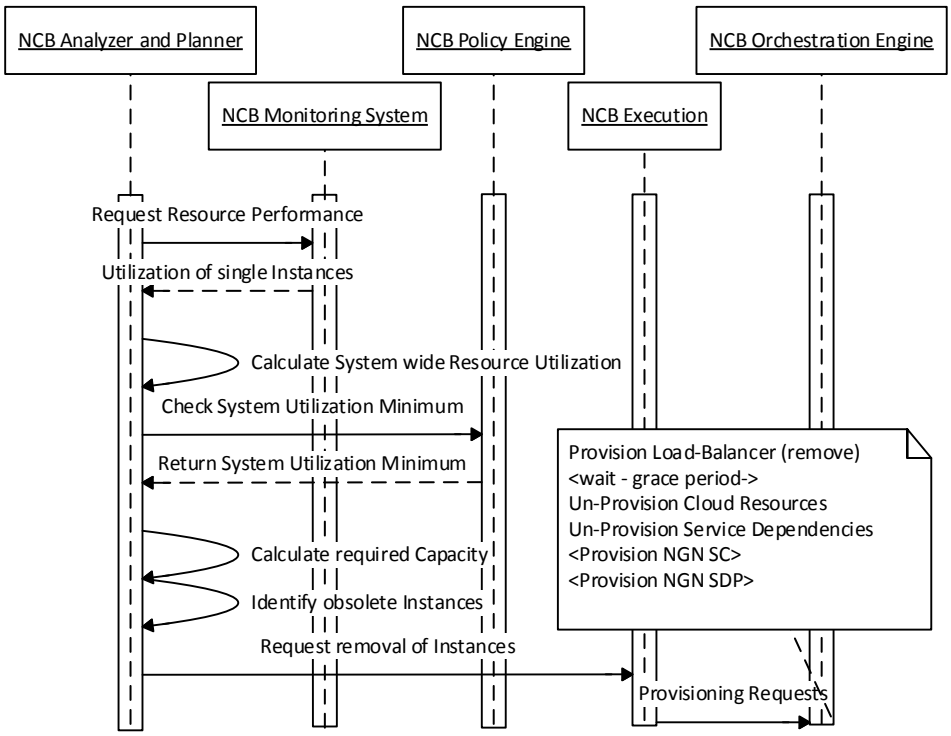

Figure 155: Resource Down-Scaling Sequence

Appendix III: Detailed Instantiation of the NGN Cloud Broker

AIII.1 Cloud Layer

Whereas the real-world multi-cloud instantiation and evaluation of the NGN Cloud Broker is carried out using multiple public cloud platforms, distributed across Europe, the local, private cloud environment is used for implementation and evaluations in controlled environments. For this, a physical data-center infrastructure, supporting several hardware virtualization mechanisms is used. A cloud infrastructure management platform which exposes open cloud infrastructure management APIs for remote management of the virtual environment was needed proving the provisioning capabilities explained in section AII.1.4.

Selected baseline technology

We compared the features of available cloud management frameworks, i.e. OpenNebula [125], Eucalyptus [124] and Nimbus [172], (details can be found in Appendix IV: Table 23) and Cloud management APIs (details can be found in Appendix IV: Table 24) available at the time (March 2010). OpenNebula was selected for management of the local virtual infrastructure mainly because of the support for the OCCI API and the support for contextualization (i.e. the ability to inject user-based configuration through the cloud management API to the virtual instance at startup time).

AIII.2 NGN Layer

AIII.2.1 NGN Service Control Platform

A reference NGN service control platform was required (as explained in section AII.1.1), which provided the basic NGN session control functions, i.e. NGN client authentication, authorization, session management. The IMS, the de-facto standard for the NGN session control layer (as introduced in section 2.3.2), is comprised of at least two different kinds of functions, i.e. the actual SIP-based service and session control functions, as well as a user database.

Selected baseline technology

The Open Source IMS (OSIMS) Core from Fraunhofer FOKUS was utilized, being the most well-known and feature rich reference implementation of an IMS-based NGN. The OSIMS provides standard-based session control functions, namely the Proxy Call State Control Function P-CSCF, the Interrogating CSCF and the Serving CSCF. Together with the Home Subscriber Server (HSS) theses CSCFs represent the core of NGN session control layers.

Utilization, Extension and Adaptations of Baseline technologies

In order for OSIMS to be provision-able through NGN Cloud Broker's provisioning and orchestration engine, OSIMS needed to be equipped with SPML-based provisioning adaptors and expose SPML-based interfaces to the NGN management platform-based provisioning Server described in section AIII.3. For the context of this work, mainly the provisioning connector for the OSIMS was relevant, for the dynamic provisioning of service profiles / iFCs, in which the actual end-points of the IMS service is specified, which in the case of this work are dynamically changing IP addresses of load-balancing systems, dynamically deployed at various IaaS cloud platforms, as explained in section AII.1.1.

AIII.2.2 NGN Service Delivery Platform

Service dynamically deployed on multiple cloud resources have to be registered at both NGN layers, the service control layer (as described in AII.1.1) as well as at the NGN service delivery layer (as described in section AII.1.2). Especially if (and that is common practice) SIP-based NGN service elements are also being made accessible through Web-Service interfaces (i.e. usually SOAP/REST interfaces, e.g. Parlay X or GSMA One API) at the SDP layer.

Selected baseline technology

The FOKUS Broker [8], shown in Figure 156, represents a reference implementation of a standard-based NGN SDP, i.e. an OMA standard-based implementation of an OMA OSE (introduced in 2.3.3), particularly including a PEEM at its core. NGN Cloud Broker is responsible for the provisioning of the PEEM upon initial service deployment on public cloud infrastructures and for the dynamic provisioning of the PEEM as soon as service end-points change due to service migrations across cloud infrastructures.

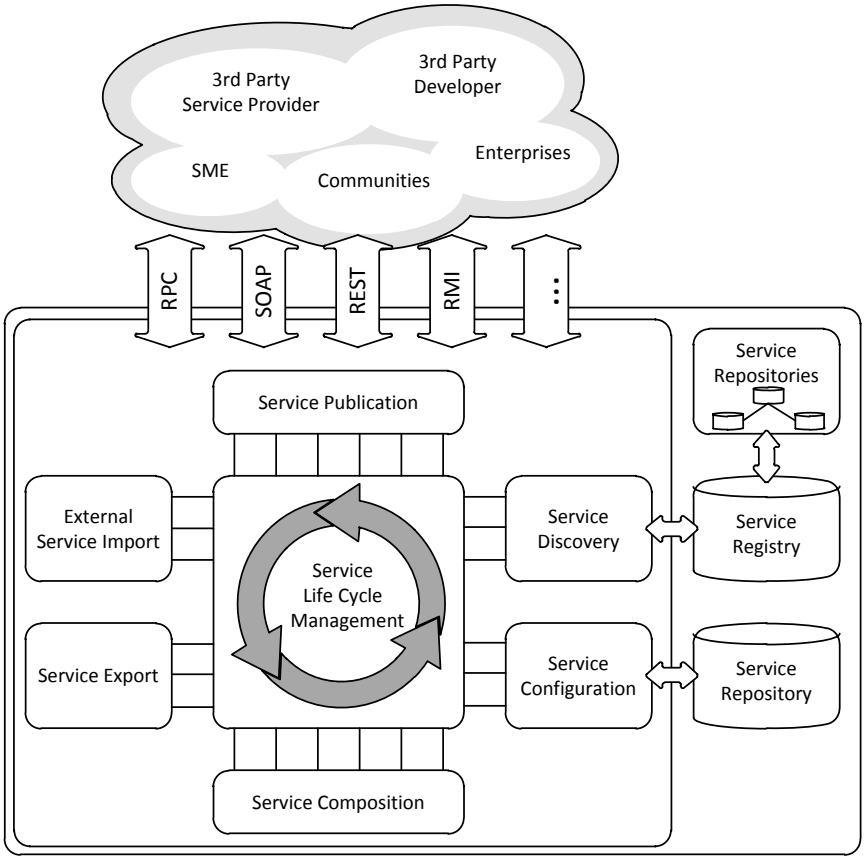

Figure 156: NGN Service Broker – the FOKUS Broker bsd. on [173]

Utilization, Extension and Adaptations of Baseline technologies

Similar to the IMS adaptations, also the FOKUS Broker needed to be equipped with SPML-based provisioning adaptors, allowing registering services at the NGN SDP layer. This required the provisioning of policy evaluation and enforcement and management provisioning, NGN service access control policies specification and provisioning of service orchestration and composition rules.

AIII.2.3 NGN Management Platform

Requirements

As described in section 8.2, in principal there are three NGN management systems that the NGN Cloud Broker has to interwork with, which are systems for the NGN service fulfillment, NGN service assurance, as well as the NGN billing systems.

The interworking with the NGN fulfilment system is needed for remotely provision NGN service control as well as NGN service delivery elements. Interworking with the NGN assurance system primarily relates to integration with NGN problem and fault management processes, including the reporting of QoS related issues. Interworking with NGN billing systems mainly relates to the reporting of currently reserved and utilized cloud resources as a

vital input for the telecommunication enterprises internal infrastructure cost / OPEX calculations.

For the instantiation of NGN Cloud Broker's infrastructure, NGN Cloud Broker's interworking with the NGN fulfilment system is of primary importance, as this is required for the actual operation of NGN Cloud Broker in contrast to NGN Cloud Broker's QoS reporting to NGN assurance systems as well as NGN Cloud Broker's reporting of current costs / OPEX to NGN billing systems.

Therefore an integrated reference implementation of an NGN fulfilment system was required able to dynamically provision various NGN resources for various cloud-based NGN services, whereas simplified NGN assurance and billing mechanisms (basically only able to record the incoming reports from NGN Cloud Broker) were sufficient.

Selected baseline technologies

The utilized technology for the NGN fulfilment system, i.e. the provisioning system of the NGN management system is an implementation of a SOA-based provisioning system, based on the SOA-based Service Provisioning Mark-up Language (SPML) specification, as standardized by OASIS [174].

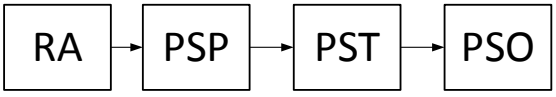

Figure 157: Service Provisioning Markup Language Domain Model, bsd on [174]

The SPML Domain Model is depicted in Figure 157. In SPML terms, a requesting authority (RA) is an entity sending SPML requests to the provisioning service provider (PSP). PSP processes provisioning requests received from the RA and returns SPML results. The provisioning service target (PST) is a target for provisioning actions. A provisioning service object (PSO) is an object on a PST. The two schema profiles of SPML are: Directory Service Markup Language (DSML) and XML Schema Definition (XSD). XSD Profile uses XML and DSML Profile uses DSML elements as a data model. The communication between RA and PSP is conducted via SPML requests, based on a request/response model. The SPML core operations are *list targets*, *add*, *lookup*, *modify* and *delete*.

Utilization, Extension and Adaptations of Baseline technologies

For the instantiation of the NGN Cloud Broker, the SPML-based provisioning engine[25], is used for the purpose of provisioning the NGN service control elements, namely the IMS home subscriber server, as well as the NGN service delivery platform elements, namely the

[25] P. Jacak, "Design and Implementation of an IMS / NGN Service Provisioning System," Technische Universitaet Berlin, Germany, 2008.

NGN service broker. For this purpose, provisioning adaptors for remotely provisioning NGN elements are being deployed at the NGN service control and service delivery layer. The provisioning engine, as part of the NGN management layer, is exposing web service interfaces, based on SPML (and partially also OSS/J, the SOA-based TMForum-based telecommunication management API standard).

AIII.2.4 NGN Client Layer

There are two types of NGN clients, "real" fully interoperable, SIP-based NGN user endpoints, as well as Browser-based clients (typically supporting only a small subset of NGN multimedia service controlling capabilities). For the real human-based testing, the myMONSTER IMS Client[26] is used, whereas for the scalability tests and long term testing, SIPp [175] is used as User Endpoint (UE) emulator and SIP/IMS load generator. For testing HTTP service performance, Httperf [176] is used as UE emulator and workload generator.

AIII.3 NGN Service Layer

The NGN Cloud Broker core functions for resource allocation and service quality management should be sufficiently generic for supporting many different types of services, NGN specific (typically SIP-based) telecommunication services as well as standard web applications (e.g. 3-tiered web application architectures). Nevertheless the type of service invocation protocols (e.g. whether SIP, HTTP-based or both) plays a critical role for the overall provisioning process, particularly of the dependent NGN systems as well as for the utilized application scaling architecture.

AIII.3.1 NGN Application Servers

Based on the explanations on the NGN service layer provided in section AII.2 a typical SIP-based real-time, voice-based NGN service was selected. The SIP Express Mediaserver (SEMS) [137] was chosen, which receives SIP-based service requests by IMS clients, upon which voice-based media (RTP streams, encoded as G711) are delivered to the IMS client. For a typical Web-based web service the HTTP-based Apache Web server is utilized, which receives HTTP requests by typical Web Clients, or the NGN SDP, upon which the service is delivered.

AIII.3.2 Components for Application Scalability

As explained in section 2.8, scalability in clouds is typically achieved through horizontal scaling mechanisms (in contrast to vertical scaling mechanisms). Horizontal scaling implies that for each service dedicated systems for balancing the incoming workload across dynamically growing or reducing numbers of application serving instances must be provided.

[26] IMS Client MONSTER, online: www.monster-the-client.org, accessed 21st May 2014

Therefore for the two types of focused services, load balancing systems needed to be provided. For HTTP load balancing, the HAproxy[27] [177] is utilized, a high-performance HTTP load balancer. For SIP load balancing, Cipango[28] a high-performance SIP load balancer was selected and adapted. Both load balancing systems were configured for supporting round-robin load-balancing algorithms and partially enhanced (in the case of Cipango) for dynamic remote provisioning.

It is worth noting that also at the load-balancing layer certain optimizations can be applied, which can optimize the distribution / load-balancing of incoming load especially in cases of heterogeneous resource back-ends. These functions relate to so called weighted round-robin load balancing mechanisms / algorithms, where the load-balancer itself adapts the amount of workload allocated to a specific resource based on received feedback about the particular resources' current utilization / load. As this additional complexity, however, did not provide additional insights into the actual performance of the system under investigation (NGN Cloud Broker), the load-balancing architecture was simplified by only allowing homogeneous resources as back-end resources / application serving slaves. By doing so it was sufficient to utilize (un-weighted) round-robin algorithms of the load-balancing systems, without limiting the general scope of this work.

AIII.4 NGN Cloud Broker

The following sections describe the utilized technologies for instantiating the NGN Cloud Broker's monitoring system (specified in section AII.3.2) and orchestration engine (AII.3.3).

AIII.4.1 NGN Cloud Broker - Monitoring System

Selected baseline technology

This work started with an NGN monitoring and management system called OMACO, the OSIMS Management Console [16] for monitoring and managing NGN service control and service delivery systems and services. The OMACO system was able to satisfy most of the requirements for multi-cloud monitoring, except for scalability related requirements. Therefore available open source monitoring and management systems were anaylzed and a comparison was conducted as shown in Appendix V: Table 25.

Although Nagios [178] and OpenNMS [179] were also promising candidates, the Zabbix [180] monitoring solution was selected, mainly due to Zabbix's well defined, RESTful JSON-

[27] HAProxy, reliable, high-performance TCP/HTTP Load-Balancer, online: http://haproxy.1wt.eu, accessed 21st May 2014

[28] Cipango, SIP/HTTP Servlets Application Server, online: http://www.cipango.org, accessed 21st May 2014

RPC-based API, which was an important criterion for the simplified integration of a monitoring solution into the NGN Cloud Broker's overall architecture, supporting remote configurations of *Items, Hosts, Host Groups, Triggers, Templates, Actions*.

Utilization, Extension and Adaptations of Baseline technologies

The most important and specific usage of Zabbix in NGN Cloud Broker's context is related to the following capabilities of Zabbix: 1) Dynamic integration (and de-commissioning) of monitoring agents, 2) Specification and activation of even triggering policies and 3) Capabilites for specification and integration of custom-made metrics. These functionalities and capabilities are further described in Appendix V: Utilized Monitoring Solution, where also examples for contextualization and policy descriptions are provided.

AIII.4.2 NGN Cloud Broker - Resource and Platform Registry

For the NGN Cloud Broker's resource and platform registry a simple relational database was required which stores the required data about each platforms resources and resource costs according to the resource model defined in section 7.1.5 and the cloud resource cost model defined in section 7.1.7. For realization of the NGN Cloud Broker's resource and platform registry a MySQL database is used.

Utilization, Extension and Adaptations of Baseline technologies

NGN Cloud Broker's cloud infrastructure platform and cloud resource description fed into NGN Cloud Broker's resource and platform registry was an extensible XML structure which listed each resource of each cloud platform provider in the following way.

```
<ProviderList>
        <name>IMSProvider</name>
        <description>List of cloud provider</description>
        <cloud_manager>http://192.168.144.150:2633/RPC2</cloud_manager>
        <providers>
                <provider>
                <id>4</id>
                <name>sleipnir</name>
                <ip_address>192.168.144.59</ip_address>
                <price>5</price>
                <bogomips> 4850.05</bogomips>
                <core>6</core>
                <ram>16384</ram>
                <minNumber>1</minNumber>
                </provider>
        </providers>
        <ranking>
                <responseTime>0.6</responseTime>
                <price>0.30</price>
                <cpu_number>0.10</cpu_number>
        </ranking>
</ProviderList>
```

The above platform and resource description XML list[29]contains:

- ID: i.e. the ID of the physical host in OpenNebula
- Name: i.e. the hostname of the node
- The ip_address is the IP of the node
- The price is an abstraction that should represent the price you pay to use this provider, for example €/h. This is useful for the ranking algorithm.
- The CPU is the CPU frequency of this host.
- The core represents the maximum number of virtual machine to create in this physical host.
- The RAM of this physical host.
- The minimum number of virtual machine to create in this host

AIII.4.3 NGN Cloud Broker - Policy Engine

For the management of the NGN Broker's policies, as described in section AII.3.3, policy engine was required. There is a broad range of polices and rules which govern the processes of NGN Cloud Broker. Whereas simple, re-active actions are conducted with the NGN Cloud Broker monitoring system itself (as described in AIII.4), for other, more complex policies a solution was sought, which 1) allowed for easy configuration / deployment of rules 2) provides convenient ways for integration and interworking and 3) provides sufficient performance as to not become a bottleneck in the overall resource allocation cycle.

Without having conducted an in-depth comparison of available rules engines consulting contemporary literature on performance and usability comparison of available rules engines [181] revealed only marginal differences of popular rules engines, such as Jess [182] or Drools [183] rule engines. Therefore the Drools [183] rules engine is used for the NGN Cloud Broker's policy evaluation processes. As such these policies are slightly more complex as the simple triggering rules that come with the monitoring system selected, however still simple policies as compared to the algorithms of NGN Cloud Broker's core resource allocation system.

Utilization, Extension and Adaptations of Baseline technologies

Firstly, the rules engine is used for comparing different QoS related performance measurements according to QoS performance aggregates, which are constellations where each single QoS metric (e.g. packet loss, jitter, packet delay) still resides below a certain QoS/SLA related threshold, but where the deterioration of several QoS factors in fact leads to a QoS violation (i.e. the effect of several factors contributes to an overall QoS problem).

[29] Carella Giuseppe, F. Schreiner (Supervisor), P. B. (Professor). (2011). "QoS-aware brokering support for IMS Infrastructures in cloud." Master Thesis, University of Bologna, 2011

Second, the rules engine was used for executing resource up-scaling and down-scaling mechanisms of resources without violating certain constraints. Constraints in this regard are the requirement that after down-scaling, remaining resources would not automatically be over-loaded so that end-less up- and down-scaling loops would occur. Other constraints are slightly more complex, involving costs of migration constraints, which need to be taken into account before migrating resources from one cloud platform to another, where the actual cost of migration does not rectify the actual gain from migration (especially important for borderline situations, where continuous migration would take place based on only marginal direct costs advantages). Here also constraints like minimal lease times were conveniently implemented, where the actual cost of a resource needed to be related to the minimal lease time.

An example of a simple drools rule for VM Up-Scaling used in the context of NGN Cloud Broker's Drools policies is given below. The example shows a simple Drools-based VM up-scaling policy.

```
rule "VM Up-Scaling"
    dialect "java"
        when
                t : Trigger( type == "vm", $vm : vm, $id : id )
        then
                System.out.println("Trigger + $id);
                VmGroup group = $vm.getGroup();
                cloudManager.deployVM(group);
                group.listVms();
end
```

AIII.4.4 NGN Cloud Broker - Orchestration Engine

For realizing the service orchestration functions of the NGN Cloud Broker described in section AII.3.3, taking into account SOA-based mechanisms (as described in section 2.1.1) and the benefits that SOA-based solutions provide like, ease of adaptation, ease of composition, versatility in use, a SOA-based orchestration engine was sought.

Different services require different provisioning workflows to be executed. For different services, different provisioning steps and mechanisms are required. From service to service managed by the NGN Cloud Broker, the orchestration workflow might contain different provisioning parameters regarding the service invocation protocol (e.g. SIP, HTTP where different load balancing mechanisms have to be used and repeatedly provisioned), specific dependencies (where different service elements have to be provisioned, from dependent databases, to additional service components) as well as invocation methods (where the NGN service control layer for instances needs to be provisioned in different fashions).

The baseline approach for coming up with a solution providing sufficient versatility, as motivated earlier are SOA techniques, as they allow for convenient and eased composition of such provisioning workflows. Moreover, SOA techniques are defined as the standard way for managing telecommunication services TMForum's SDF (as introduced in section 2.4.3) and now that more and more also entered the service management plane, it became clear to us (compare the author's work in [16] for SOA-based NGN management and the author's work

in [6] for SOA-based autonomic communication mechanisms for service compos-ability management) to extensively exploit SOA mechanisms not only for composing consumer oriented services, but also for composing both, management as well as consumer oriented services and service components.

Therefore a SOA-based workflow engine based on Business Process Execution Language (BPEL) is used, which allows for convenient modeling and execution of the required provisioning workflows. What adds to that choice is the author's research into automated composition mechanisms, through semantically enriching service descriptions in order for so called "reasoners" to be able to automatically compose complex workflows out of available service components. As the author showed in [3], [4], [9], [14], [16], [17], with sufficiently annotating service components and sufficiently describing service dependencies, service compositions can be carried out in a highly automated fashion. With growing numbers of heterogeneous services (to be managed on multiple cloud platforms) such mechanisms more and more become relevant for an application in such frameworks. However for the instantiation of the NGN Cloud Broker, focus was put more on the actual continuous process of optimizing the resource allocation process for a particular service instead of focusing on fully automated provisioning mechanisms (which however can be realized as an extension of the technologies used).

The selected BPEL-based service provisioning allows for both, the manual, yet graphically aided design of the orchestration and provisioning workflow, as well as the automated composition of workflows.

Utilization, Extension and Adaptations of Baseline technologies

The BPEL workflow for the orchestration of required provisioning tasks of a NGN services, conducts the aforementioned (section AII.3.3) steps of 1) IaaS Resource Provisioning, via OCCI interfaces, 2) IaaS Instance OS/SW Setup and Configuration, via SPML interfaces, 3) NGN Service Control Provisioning, via SPML interfaces and 4) NGN SDP Provisioning, via SPML interface.

Appendix IV: Utilized Cloud Platform and API

For the implementation of the local Cloud testbed, the following evaluation of available Cloud platforms, Cloud management APIs (as of March 2010) was carried out. The utilized Cloud Management API calls are introduced subsequently.

Table 23 Comparison of Cloud Infrastructure Management Frameworks (as of March 2010)

		OpenNebula	Eucalyptus	Nimbus
	implements	OCCI	-	-
	organisation	Distributed Systems Architecture Group	UCSB	University of Chicago
	IaaS support	x	x	x
	HaaS support	-		-
	SaaS support	-		-
	RESTful interface/support		x	?
virtualization platforms support	Xen	x	x	x
	KVM	x	x	- (planned)
	VMWare	x	- (planned)	-
on-demand access to cloud providers	Amazon EC2 (fully, partially)	? / x	? / x	- / x
	ElasticHosts (fully, partially)	? / x		- / -
	Amazon S3 (fully, partially)		? / x	?
	Amazon EBS (fully, partially)		? / x	?
	multiple users	x		
	image transferring / deployment	x		
	image cloning	x		
	virtual network management	x		
	service contextualization	x	-	x
	open source	x	x	x
	API	x		
	build your own private IaaS-cloud infrastructure	x	x	x
	current version number	v1.4	v1.6.2	v2.3
	monitoring		x	

OCCI primarily was identified to be the most relevant candidate as a cloud management APIs, due to its extensibility, openness, vendor-independence as well as due to its capabilities for searching for resources and differentiation of resource categories.

Table 24 Comparison of Cloud Management APIs (as of March 2010)

	OCCI	Flexi-Scale API	GoGrid API	Sun Cloud API	Amazon EC2	Rack-space Cloud Servers API	VM-Ware vSphere
authentication via HTTP	x	x	n/a	n/a	x	n/a	n/a
authentication via request signing	x	n/a	n/a	x	n/a	x	n/a
ephemeral compute resources	x	x	n/a	n/a	n/a	x	n/a
persistent compute resources	x	n/a	x	x	x	n/a	x
ephemeral storage resources	x	n/a	n/a	n/a	n/a	x	n/a
persistent storage resources	x	x	x	x	x	x	x
multiple storage resources	x	x	n/a	n/a	n/a	x	n/a
multiple network resources	x	x	n/a	n/a	x	n/a	n/a
static IPs	x	x	x	x	x	x	x
firewalling	?	n/a	x	x	n/a	x	n/a
load balancing	?	n/a	n/a	x	n/a	n/a	x
billing	?	x	x	n/a	n/a	n/a	n/a
resource categories	x	n/a	n/a	n/a	n/a	n/a	n/a
resource search	x	n/a	n/a	n/a	n/a	n/a	n/a
resource tagging	x	n/a	n/a	n/a	x	n/a	n/a
collections (pass-by-reference)	x	x	n/a	n/a	n/a	n/a	n/a
collections (pass-by-value)	x	x	x	x	x	x	x

OCCI Cloud Infrastructure Instances

The OCCI model for cloud infrastructure resources as shown in Figure 158 is comprised of a set of Compute, Storage and Network elements connected through Cloud infrastructure internal links and exposed through network interfaces.

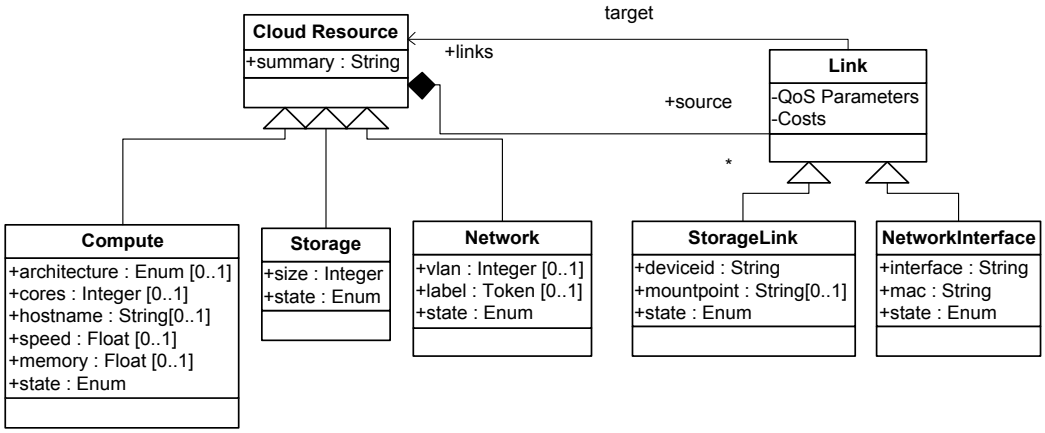

Figure 158: OCCI Cloud Infrastructure Resource Model [111]

Cloud Resource Elements

The root element required for all the Pool Resources (PRs) is named after the pool name, e.g. `COMPUTE`, `NETWORK` or `STORAGE_COLLECTION`. No attributes can be defined for the root element. Each one of Entity Resources (ERs) in the pool are described by an element (e.g. `COMPUTE`, `NETWORK` or `STORAGE`) with one attribute: `href`, a URI for the ER, as for example:

Compute Collection:

```
<COMPUTE>
  <COMPUTE href="http://www.opennebula.org/compute/234">
  <COMPUTE href="http://www.opennebula.org/compute/432">
</COMPUTE>
```

Network Collection:

```
<NETWORK>
  <NETWORK href="http://www.opennebula.org/network/234">
  <NETWORK href="http://www.opennebula.org/network/432">
</NETWORK>
```

Storage Collection:

```
<STORAGE>
  <STORAGE href="http://www.opennebula.org/storage/234">
  <STORAGE href="http://www.opennebula.org/storage/432">
</STORAGE>
```

The `COMPUTE` element defines a virtual machine by specifying its basic configuration attributes such as `NIC` or `DISK`. The following elements can be defined:

```
ID, the uuid of the virtual machine.
NAME, describing the virtual machine.
TYPE, a COMPUTE type specifies a CPU and memory capacity, valid types are small, medium and
large.
STATE, the state of the COMPUTE. This can be changed to stopped, suspended, resume, cancel,
shutdown done
DISK, the block devices attached to the virtual machine.
```

```
NIC, the network interfaces.
```

Example:

```
<COMPUTE href="http://www.opennebula.org/compute/32">
  <ID>32</ID>
  <NAME>Web Server</NAME>
  <INSTANCE_TYPE>small</INSTANCE_TYPE>
  <STATE>running</STATE>
  <DISK>
    <STORAGE href="http://www.opennebula.org/storage/34"/>
    <TYPE>OS</TYPE>
  </DISK>
  <DISK>
    <STORAGE href="http://www.opennebula.org/storage/24"/>
    <TYPE>CDROM</TYPE>
  </DISK>
  <NIC>
     <NETWORK href="http://www.opennebula.org/network/12"/>
     <MAC>00:ff:72:31:23:17</MAC>
      <IP>192.168.0.12</IP>
  </NIC>
</COMPUTE>
```

States of Cloud Resource Elements

Focusing on Compute Elements, OpenNebula defines the following states of a cloud compute element:

```
INIT         --> The VM just entered the system
PENDING      --> The VM is waiting for the scheduler
HOLD         --> The VM needs to be released
ACTIVE (*)   --> the VM just entered the VM life cycle manager
STOPPED      --> The VM is stopped, waiting for resume
SUSPENDED    --> The VM is suspended, waiting for resume in the same host
DONE         --> The VM has ended the lifecycle successfully
FAILED       --> The VM has ended the lifecycle due to a problem
```

The states and transitions between the cloud resource element states are depicted in Figure 159.

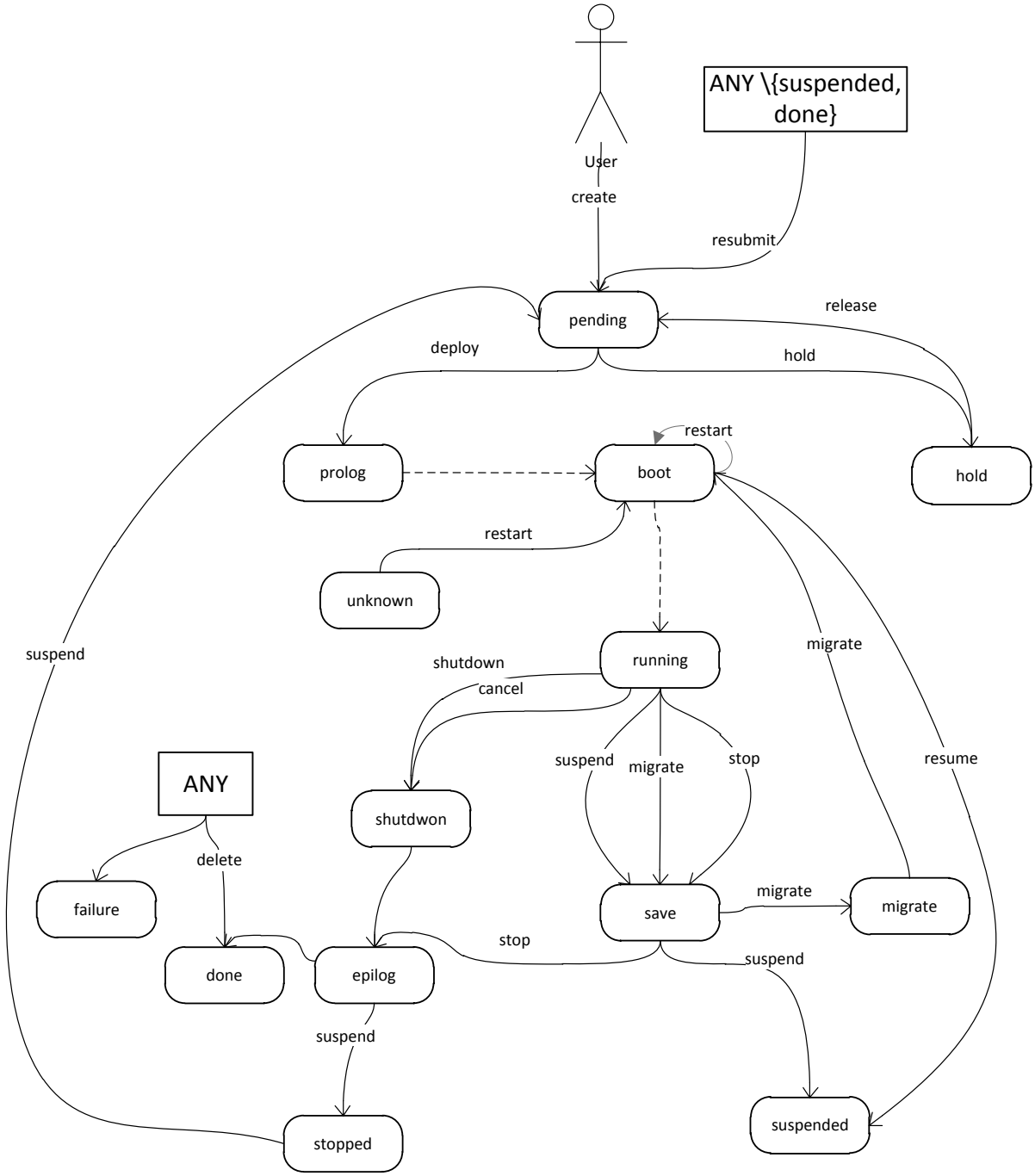

Figure 159: States of Cloud Compute Resources in OpenNebula[30]

For evaluating a Cloud Infrastructure (Management) Performance, the delay between user initiated creation of VMs and the state of having a virtual resource up and running is of key importance. The longer the duration of this transition, the more inflexible the overall resource allocation reacts to change (e.g. in workload related capacity requirements).

[30] Bsd. on OpenNebula, Managing Virtual Machines 4.4, online: http://archives.opennebula.org/documentation:rel4.4:vm_guide_2, accessed 21st May 2014

Cloud Infrastructure Resource Management OCCI API

The BonFIRE OCCI API is a RESTful service to create, control and monitor cloud resources based on the latest draft[31] of the OGF OCCI API specification[32]. There are two types of resources that resemble the basic entities managed by the OpenNebula system, namely:

Pool Resources (PR): Represents a collection of elements owned by a given user. In particular three pool resources are defined: COMPUTE, NETWORK and STORAGE.

Entry Resources (ER): Represents a single entry within a given collection: COMPUTE, NETWORK and STORAGE.

A `COMPUTE` entry resource can be linked to one or more `STORAGE` or `NETWORK` resources. The methods associated with each resource type are as follows:

Pool Resources (PR)

```
GET: to list all the entry resources in that pool resource owned by the user
POST: to create a new entry resource
```

Entry Resources (ER)

```
GET: to list the information associated with that resource
PUT: to update the resource (only supported by the COMPUTE resource)
DELETE: to delete the resource
```

The Cloud Infrastructure Resource Management Methods of the OCCI API are shown in Figure 160.

[31] OGF, Open Cloud Computing Interface Secification, online: http://forge.ogf.org/sf/go/doc15731, accessed 21st May 2014

[32] Open Cloud Computing Interface, online: http://www.occi-wg.org/, accessed 21st May 2014

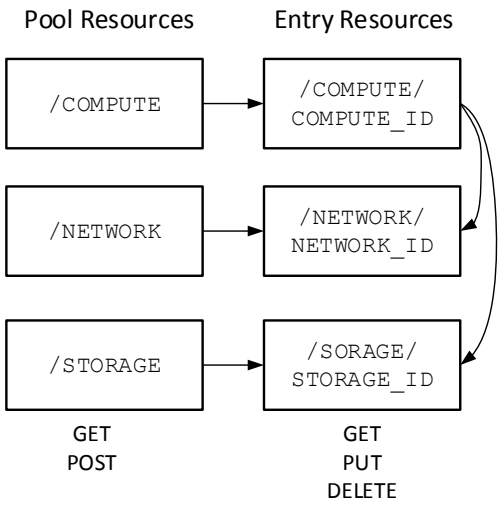

Figure 160: Cloud Infrastructure Resource Management Methods of OCCI API

Appendix V: Utilized Monitoring Solution for Multi-Cloud Monitoring

The comparison of applicable monitoring solutions for instantiating the NGN Cloud Broker is shown in Table 24.

Table 25 Comparison of Monitoring Solutions[33]

Name	Logical Grouping	Trending	Auto Discovery	Agent	SNMP	Triggers / Alerts	WebApp
Ganglia	Yes	Yes	Via gmond check in	Yes	Via plugin	No	Viewing
OpenNMS	Yes	Yes	Yes	Supported	Yes	Yes	Full Control
Zabbix	Yes	Yes	Yes	Supported	Yes	Yes	Full Control
Nagios	Yes	Yes	Via plugin	Supported	Via plugin	Yes	Full Control
	Distributed Monitoring	**Inventory**	**Data Storage Method**	**License**	**Maps**	**Access Control**	**IPv6**
Ganglia	Yes	Unknown	RRD, in memory	BSD	Yes	No	Unknown
OpenNMS	Yes	Limited	RRD, PostgreSQL	GPL	Yes	Yes	Limited
Zabbix	Yes	Yes	Oracle MySQL PostgreSQL SQLite	GPL	Yes	Yes	Yes
Nagios	Distributed Monitoring	Inventory	Data Storage Method	License	Maps	Access Control	IPv6

The most important Zabbix features utilized for implementing the NGN Cloud Broker are listed below:

Dynamic integration (and de-commissioning) of monitoring agents

Monitoring agents are started with the process of stating each virtual node and provided with the monitoring aggregation function's IP through contextualization. An example of the contextualization information provided at startup of VMs is provided below. In the example the URL of the monitoring aggregator, as well as the metric to be monitored, including frequency of monitoring is provided through conextualization.

```
<context>
    <aggregator_ip>10.1.1.16</aggregator_ip>
```

[33] bsd. on A. Fedulov, "Design and Implementation Of a Flexible Platform-Independent Elasticity Management System for CloudBased Services," Master Thesis, Technische Universität Hamburg-Harburg, 2011, supervised by the author

```
    <metrics><![CDATA[<metric> users,wc -l /etc/passwd|cut -d" " -f1, rate=20,
       valuetype=3, history=10 </metric>]]></metrics>
</context>
```

By doing so, each monitoring agent is provisioned for monitoring the required metrics (which can be active as well as passive measurements, system specific or infrastructure wide metrics), and automatically registers at the monitoring aggregation function using Zabbix's active agent auto-registration mechanisms[34] . This concept was later adopted in the BonFIRE project [164].

Specification and activation of even triggering policies

Triggers are specified in the following format:

```
{<server>:<key>.<function>(<argument>)}<operator><const>
```

In addition to logical operators (+ - * / ^ & |), the functions shown in Table 25 are supported.

Table 26 Functions for defining threshold-based policy enforcement / triggering actions in Zabbix [180]

Function	Parameters	Description
abschange	-	Returns absolute difference between last and previous values.
avg	sec or #num	Average value for period of time or for #num number of values
count	sec or #num	Number of historical values for period of time in seconds or number of last *#num* values matching condition.
dayofmonth	-	Returns day of month in range of 1 to 31.
delta	-	Same as max()-min().
last	sec or #num	Last (most recent) value. Parameter*: #num* – Nth value. last(#3) – third most recent value
max	sec	Maximal value for period of time
min	sec	Minimal value for period of time.
nodata	sec	Returns: 1 – if no data received during period of time in seconds. 0 - otherwise
sum	sec	Sum of values for period of time. Parameter defines length of the period in seconds.
time	-	Current time of the Zabbix Server in HHMMSS format.

Based on this a trigger reacting on a virtual cloud resources CPU utilization threshold might look as follows.

```
{HOSTNAMEorIP:system.cpu.load[all,avg1].last(0)}>5
```

[34] Zabbix Documentation Version 1.8, Agent Autoregistration, online: http://www.zabbix.com/documentation/1.8/manual/about/what_s_new#auto_registration_for_active_agents, accessed 21st May 2014

Subsequently to specifying trigger points, the actual action[35] to be carried out upon thresholds have been reached needs to be specified. Whereas a standard action of Zabbix is the sending of notifications to administrators, in the case of this work, the NGN Cloud Broker's resource allocation and platform selection mechanisms are triggered.

Capabilites for specification and integration of custom-made metrics

Monitoring agents are provisioned not only making use of the standard monitoring metrics of the Zabbix monitoring system, but also providing additional actively as well as passively measured metrics. For integration of user-defined metrics, the monitoring aggregation servers has to be provisioned with a new metric defining item (only once), and monitoring agents have to be provisioned with this new user-defined parameter the monitoring agent's configuration files. By doing so virtually any metric can be gathered through external programs, databases or system states and provided to the aggregation server on periodical basis (with frequencies freely defined by users).

[35] Zabbix Documentation Version 1.8, Agent Autoregistration, online:http://www.zabbix.com/documentation/1.8/manual/config/actions, accessed 21st May 2014

Appendix VI: Related Research Projects

EU FP7 Reservoir

The EU FP7 project Resources and Services Virtualization without boundaries RESERVOIR [119], in the EU context marked an important milestone in the area of Cloud service related research projects, as significantly more emphasis was put on the service management layer, in contrast to past projects focusing on the virtual execution environment management (VEEM) layer, as shown in Figure 161.

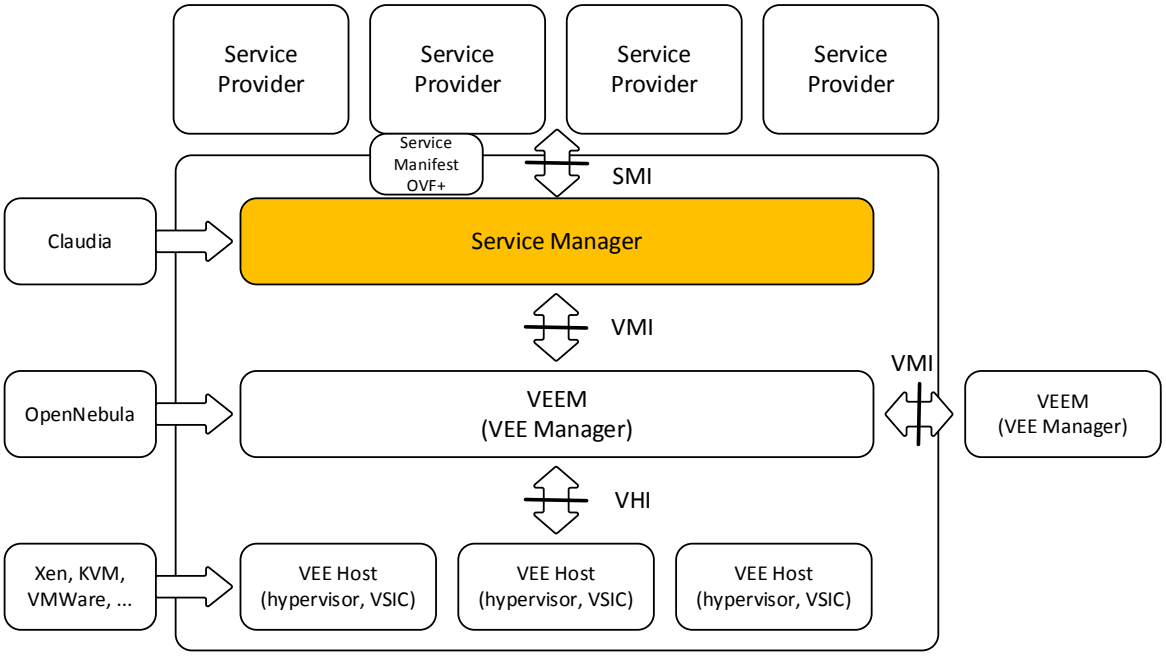

Figure 161: Reservoir Architecture (bsd on [119])

As shown in Figure 161, the Cloud service management layer (Claudia) of the RESERVOIR architecture comprised 1) a service lifecycle management system, 2) a system for optimizing scalability and 3) a monitoring system.

These three basic systems and their interworking processes will also play an important role in the context of this work. Unfortunately the Claudia software was poorly documented and it seemed that the official download only comprised parts of the required components. Also there was lack of evidence of Claudia's real-world applicability / evaluation, otherwise several of Claudia's components would have possibly been eligible as a starting point for the developments of this work. In contrast work conducted in the context of this thesis however, the scope of Claudia was limited to purely Web-based applications, did not incorporate cloud platform selection and only took very basic monitoring information for the optimization process into account.

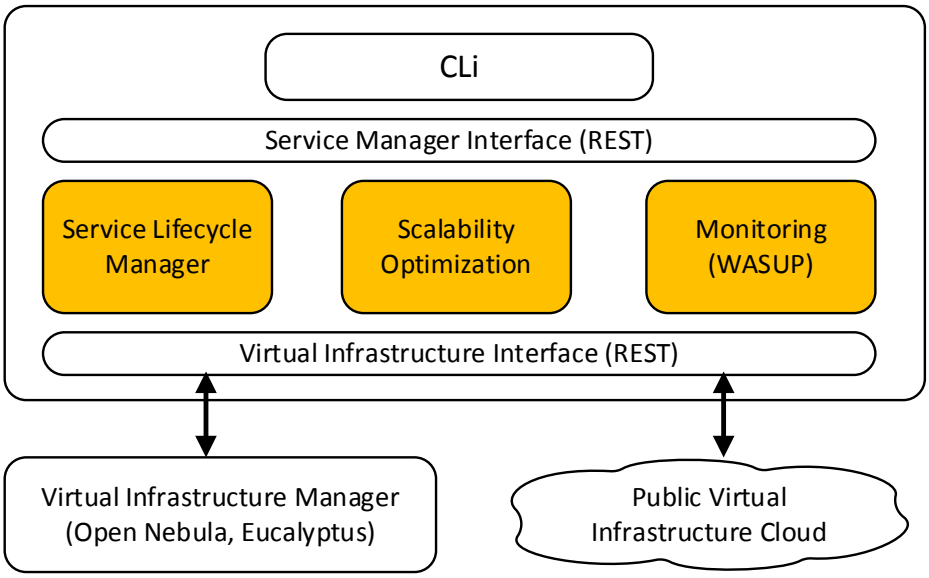

Figure 162: Cloud Service Management Layer (Claudia) in FP7 Reservoir[36]

EU FP7 OPTIMIS

The EU FP7 project Optimized Infrastructure Services - OPTIMIS[37] focuses on scenarios, where organizations seek to outsource their applications and services to third party cloud infrastructures. OPTIMIS develops a framework, which allows organizations to automatically outsource services and applications providing mechanisms for dynamically assessing trustworthiness, risks, costs and SLAs of available cloud platform providers. Shown in Figure 163 is the OPTIMIS cloud monitoring infrastructure, together with the OPTIMS framework for optimizing cloud resources and cloud-based services [184]. Similar to the earlier project RESERVOIR, the core components of the service management plane comprise a monitoring system (although by now several more metrics from the physical plane to the application plane are monitored), a cloud optimization system, and a service optimization system. Additionally, the monitoring and evaluation system however, integrates with components for evaluating trust and risk of cloud platform providers.

Although the OPTIMIS tools at the point of finalizing this thesis could not be tested, and although the scientific publications of OPTIMIS concentrated on simulations, in contrast to real world evaluations, the OPTIMIS approach of brokering and scaling services across multiple cloud platforms shares many similarities with the work conducted in the scope of this thesis.

[36] Claudia Cloud Service Management Architecture, online: http://claudia.morfeo-project.org, accessed 21st May 2014

[37] EU FP7 OPTIMIS, online: http://www.optimis-project.eu, accessed 21st May 2014

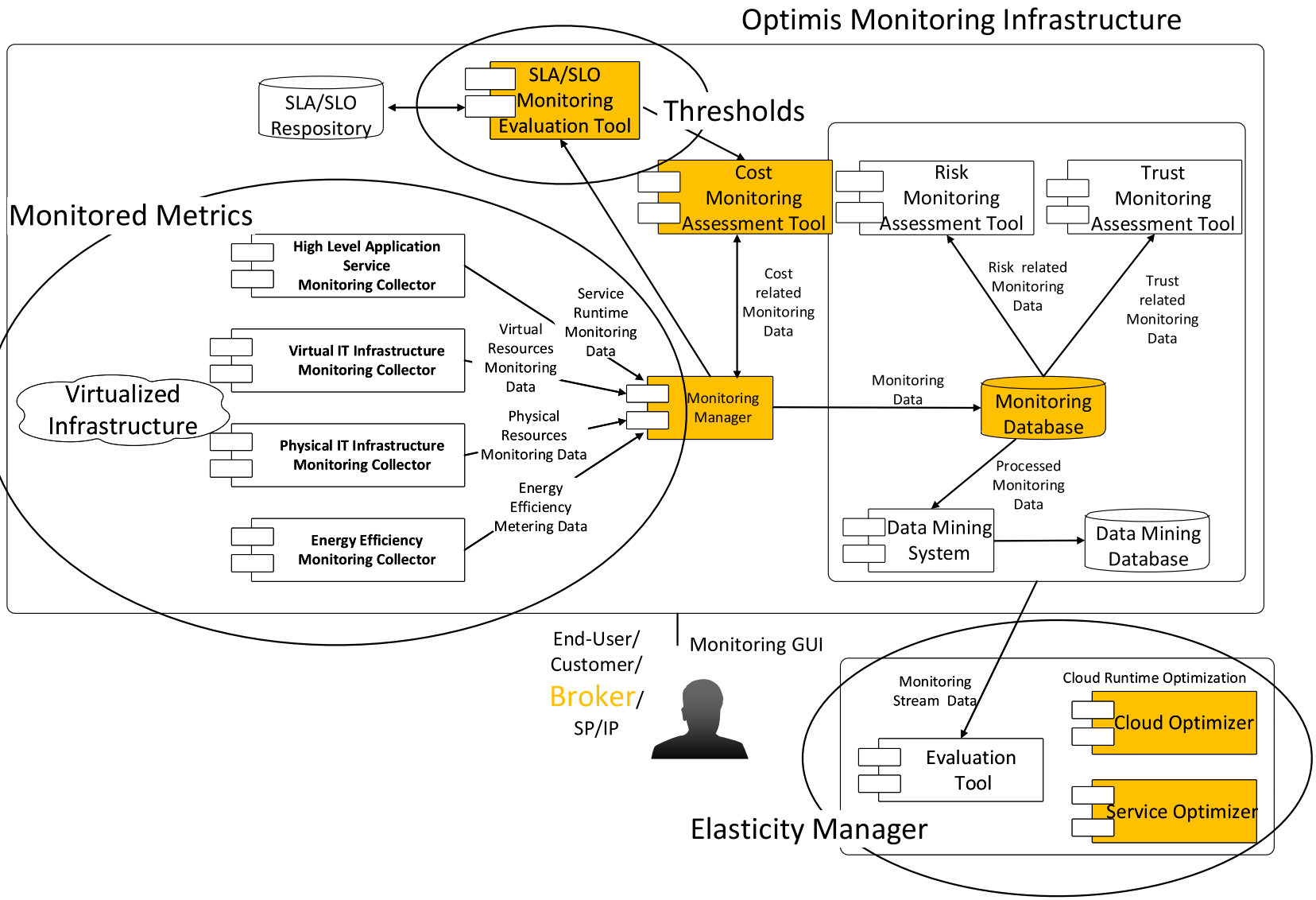

Figure 163: EU FP 7 OPTIMIS Cloud Monitoring Infrastructure and Elasticity Management (bsd on [184])

EU FP7 BonFIRE

The EU FP7 project "Building service testbeds for Future Internet Research and. Experimentation" – BonFIRE, operates several Cloud infrastructures across Europe and provides an environment to flexibly provision Cloud resources across those multiple sites. The BonFIRE architecture, as shown in Figure 164, is comprised of four architecturally different layers, 1) the tesbed layer where Cloud management tools receive provisioning request, 2) the Enactor layer, where complex provisioning requests are sequenced, 3) the resource management layer, where resources are reserved as well as 4) the experiment layer, where experimenters can either control multi-cloud resource via APIs or through a portal. The highlighted elements were particularly developed as an instantiation of the developments carried out in the course of this thesis (see also the impact section 10.2.3).

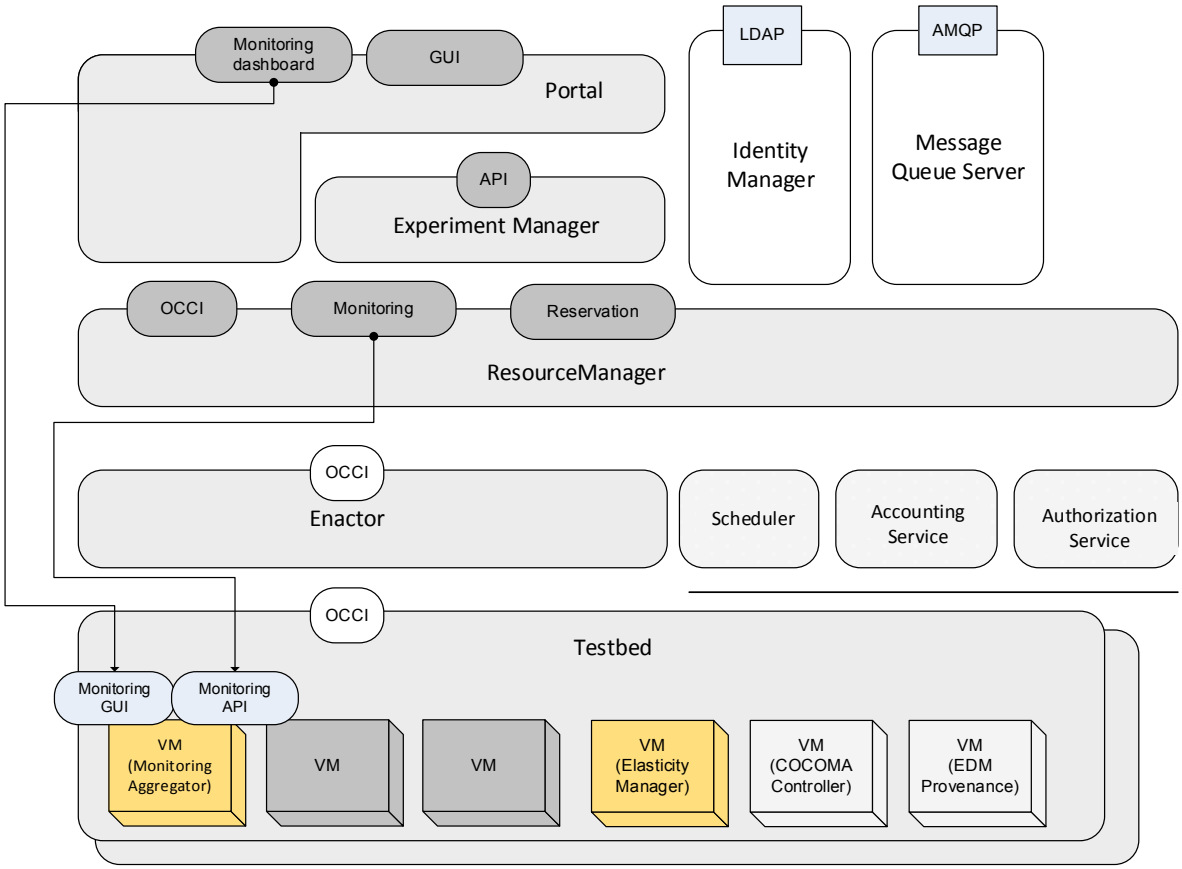

Figure 164: BonFIRE Architecture [164]

Acknowledgement

This work would not have been possible without the support from many colleagues, students, and friends.

First and foremost I would like to thank Niklas Blum for pushing me. I think the two of us managed the last decade including many stressful phases pretty well, and still being friends, for me, is something extraordinarily unique and exquisite.

But also sensei Konrad Campowsky's optimism and support needs to be honoured at this place, "it's all doable, no worries" – thanks a lot!

Several diploma-/master theses, under my supervision have contributed to this work, particularly I would like to thank Giuseppe Carella, Alexander Fedulov and Sebastian Krämer, who all delivered excellent results and successfully beard my moods. Particularly I would like to thank Giuseppe Carella and Roman Busse for their superior support.

Of course, also without the motivation of my friends Meike, Lia and Tina and the continuous support of my family, I would not have found the energy.

Finally and foremost I want to thank my doctoral adviser Prof. Dr. Thomas Magedanz for supporting me.