The Cloud Supply Chain: A Framework for Information, Monitoring, Accounting and Billing

Maik Lindner[1], Fermín Galán[2], Clovis Chapman[3], Stuart Clayman[3], Daniel Henriksson[4], and Erik Elmroth[4]

[1] SAP Research, Belfast, UK
m.lindner@sap.com,
[2] Telefónica I+D, Emilio Vargas 6, 28040 Madrid, Spain
fermin@tid.es,
[3] Departments of Computer Science and Electrical Engineering
University College London, Malet Place, WC1E 6BT, London, UK
c.chapman@cs.ucl.ac.uk|sclayman@ee.ucl.ac.uk,
[4] Department of Computing Science and HPC2N, Umea University, Sweden
danielh@cs.umu.se|elmroth@cs.umu.se

Abstract. Cloud computing is changing the way in which companies deploy and operate ICT based services. This paradigm introduces several advantages compared with traditional data centers, such as a great degree of flexibility, pay-per-use models, and rapid resource provisioning. However, the lack of a well defined supply chain for clouds and an associated information model is limiting the adoption of these technologies. This paper introduces the *Cloud Supply Chain*, which enables both consuming and providing organizations to clearly determine their position within such a supply chain. The *Cloud Supply Chain* is the result of our experience from building systems for supply chain businesses combined with our experience of building Service Cloud infrastructures within the RESERVOIR EU research project. This paper discusses the definitions and components of such a supply chain, together with all of the requirements with regards to services and an information model, which are the most pertinent topics for accounting and billing. The underlying basis for this work is a service provisioning process chain that includes service deployment, comprehensive monitoring, accounting and billing, delivering technical as well as business information. Our work presents the first definition of a *Cloud Supply Chain*, providing a foundation for researchers and businesses in this area.

Key words: Cloud Computing, Supply Chain, Accounting, Billing, Monitoring, Information Flow, Information Model, service provisioning

1 Introduction

The cloud computing service model combines a general organizing principle for IT delivery, infrastructure components, an architectural approach and an economic model. The resource acquisition, usage and maintenance capabilities of cloud computing infrastructures enable customers to access and use software Software as a Service (SaaS), Platform as a Service (PaaS) or Infrastructure as a Service (IaaS) offerings that lower their Total Cost of Ownership (TCO) if compared to traditional on and off premise data center models [5, 27]. The infrastructure which supports cloud computing enhances the customisation, flexibility and scalability of resource acquisition, usage and maintenance, such that greater masses and varieties of customers and applications can be served by a single data center [4, 22]. However, even with the technological know-how for sizing and scaling computational resources on demand, many large organisations are hesitant to engage in cloud computing, even if they consider it to be a viable model [17, 30]. One potential reason for this behaviour is that they feel uncertain about the impact of changes on their overall IT landscape and operations [27, 14].

Though the base concept of clouds has been known for decades, it is during the last decade that clouds have really taken off. As future systems will exploit the capabilities of managed services and resource provisioning further, the clouds will probably continue to grow in popularity also in the years to come. Clouds are of particular commercial interest not only with the growing tendency to outsource ICT, in order to reduce management overhead and to extend existing and limited ICT infrastructures, but even more importantly, they reduce the entrance barrier for new service providers allowing them to offer their respective capabilities to a wide market with a minimum of both entry costs and infrastructure requirements [18]. Thus, new service providers can focus on their main business rather than on building the infrastructure needed. In fact, the special capabilities of cloud infrastructures allow providers the advantage of experimenting with novel service types whilst removing the disadvantage of infrastructure provisioning, thereby reducing or eliminating the risk of wasting expensive resources.

What hinders companies from embracing the cloud, are not only technical hurdles (latency, legal aspects, etc.) and psychological effects (loss of control, etc.), but also the lack of a comprehensive overview of the complete supply chain plus the missing insight into information flow, the monitoring requirements, and the processes of acccounting and billing of cloud services. To overcome this gap, this paper introduces the *Cloud Supply Chain.* The contents of the paper are the results of our experience from building systems for supply chain businesses combined with our experience of building Service Cloud infrastructures within the RESERVOIR EU research project [1, 28].

We present a comprehensive framework of the cloud supply chain and we focus particularly on the infrastructure services, as they are the basis for all cloud services. As such, the presented research strand is based on a fundamental technical background and emerging technology enhanced by essential business research knowledge and future-oriented models. We have combined technical oriented research that has already taken place in RESERVOIR [28], with the business aspects of service provisioning along the supply chain concept.

The following sections of the paper are structured as follows. First, we define the concept of the cloud supply chain, motivated by the traditional supply chain theory and new settings on the ICT market (Section 2). Based on that, we then compare traditional and emerging supply chain concepts and include an analysis of functional and innovated products in ICT. Next, we introduce a cloud service provisioning model which defines the basis for monitoring, accounting, and billing (Section 3). This section includes the identification of service provisioning subprocesses in the supply chain and existing standards. On this foundation, we then describe the monitoring of cloud services, from which the drill-down into highly dynamic infrastructure monitoring and data representation and communication is done (Section 4). Based on the information model and monitoring for cloud services, the accounting and billing in the supply chain is described (Section 5). We provide an overview of the information flow, the billing models, as well as the accounting data management. Finally we finish the paper with a summary and an outlook of the future research (Section 6).

2 The Cloud Supply Chain

2.1 Definition

The application of the supply chain concept in the context of cloud computing is innovative and opens a new research field. The following definition, from [26], delivers a basis for this: “a supply chain is two or more parties linked by a flow of goods, information, and funds”. Applied to cloud computing, we propose the following variation:

A Cloud Supply Chain is two or more parties linked by the provision of cloud services, related information and funds.

It is important to mention that, as Cachon and Fisher show [3], sharing of information is not the only thing leading to costs within the supply chain, but also the management and restructuring of services, information, and funds for an optimization of the chain. In general a supply chain performs two types of functions [15], namely:

i) a physical function comprises the production of the product out of raw material or intermediate parts or components, and the transportation of all components to the right place, and
ii) a market mediation function ensures that the variety of products reaching the marketplace matches what customers want.

While for functional products the physical function dominates, the market mediation function is more important than the physical function for innovative products [26]. Here the mixed characteristics of the cloud supply chain lead to a high importance of the physical function like the provisioning of software services, as this is the core product of cloud services, but moreover the need for a strong market mediation function arises from the modular design of these services.

2.2 Components of the Cloud Supply Chain

The cloud supply chain of cloud services needs to be identified and then managed and controlled from both a business and technical perspective. The cloud supply chain represents a network of interconnected businesses in the cloud computing area involved in the end-to-end provision of product and aggregated service packages required by end cloud service customers. Therefore supply chain execution for the cloud is managing and coordinating the (partly) bi-directional movement of services, information and funds across the cloud supply chain. This includes (but it is not limited to it) the actual provisioning of infrastructure services, the monitoring of services like the provisioning of virtual machines and the information processes supporting accounting and billing processes. To capture this complex chain, it is needed to identify and clearly define the following components: the actors and the services exchanged (products along the cloud supply chain), as well as the flow in information and funds. All of these are described in the following sections and shown in Figure 1.

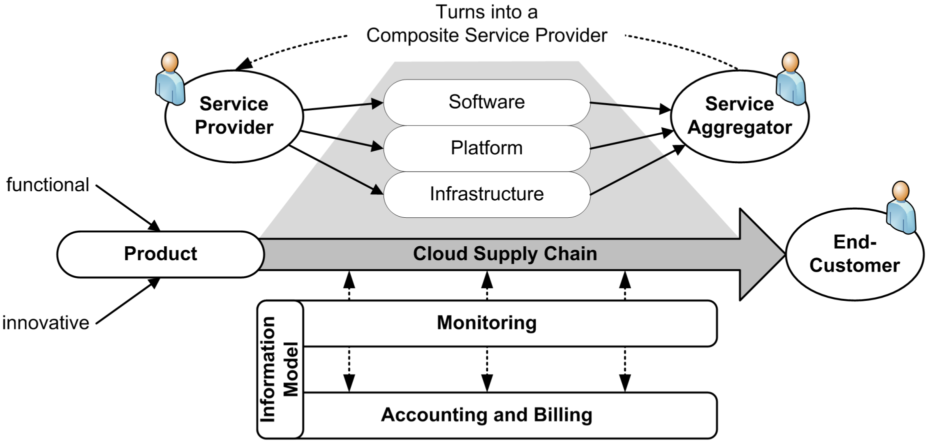

Fig. 1. Cloud Supply Chain

Main Actors Several actors have to be identified along the supply chain. *Service providers* can actually take several roles within the cloud supply chain. They might act as infrastructure[1], platform or software providers and directly be in contact with the end-customer. But they might also be a broker (which is a role of the actor service provider) or a business partner of a *service aggregators*, that uses the provided service and combines or enriches it with another service or new functionality. By doing so a composite service is created. As an example a composite service might be a piece of software that runs as a service on top

[1] Service providers providing infrastructure are sometimes called just *infrastructure providers.*

of a flexible provided platform. Thus, the product for the end-customer is software as a service provided in a flexible manner. When such a supply network is created, it is even more important to maintain visibility and transparency of all processes and data for monitoring and accounting and billing as one can imagine such a end-product can get easily quite complex and include many actors. The *end-customers* usually consume a product, that is a single or composite service, which is provided by a service provider over the cloud supply chain.

Products along the Cloud Supply Chain In general a supply chain has to be classified according to the product it supplies. Fisher [15] classifies products primarily on the basis of their demand patterns into two categories: products are either (a) primarily functional or (b) primarily innovative. On the one hand, functional products fulfill the following 3 criteria:

i) to satisfy basic needs that do not change much over time,
ii) have predictable and stable demand with low uncertaintie,
iii) have long life cycles, typically more than two years.

Due to their stability, functional products favour competition, which leads to low product margins and, as a consequence of their properties, to low inventory costs, low product variety, low stockout costs, and low obsolescence [21][15]. On the other hand, innovative products are characterized by:

i) additional (other) reasons for a customer in addition to basic needs that lead to purchase,
ii) unpredictable and variable demand, difficult to forecast,
iii) short product life cycles, typically three months to one year.

While companies selling innovative products can achieve higher profit margins for an innovative product compared to a functional one, innovative products require frequent innovations due to emulating competitors. Furthermore, innovative products will have low volumes per stock-keeping unit (SKU), high stock out costs, and high obsolescence [21].

In general the products coming out of emerging ICT are to be classified as innovative products, but have certain characteristics of functional products as well. Cloud services should fulfill basic needs of customers and favour competition due to their reproducibility. But they also show characteristics of innovative products as the demand is in general unpredictable (on-demand business model) and have due to adjustments to competitors and changing market requirements very short development circles. So cloud services as a product need to be classified as innovative, while they still feature characteristics of functional products.

Information and funds Regarding information and funds flows that characterize the cloud supply chain (see Section 2) the following can be clearly identified:

Funds. The service provider has a payment relationship with the cloud infrastructure provider by the use of IT infrastructure. Typically, the payment follows a pay-per-use model, which is one of the main drivers toward cloud computing adoption compared with the traditional exploitation fixed-rate of IT infrastructure. This flow uses to be unidirectional, from service provider to cloud infrastructure provider. However, some times it goes in the opposite direction, e.g. compensation penalties due to Service Level Agreement (SLAs) violations. SLAs allow service providers to protect their investment, allowing them to seek some form of financial compensation should the infrastructure not operate as planned.

Information. There are several pieces of information that are interchanged between the service provider and infrastructure provider along the different subprocesses in the service provisioning chain, analyzed in the following subsection.

Emerging Supply Chain Concept This mixed characterization of the supply chain in terms of the provided products is furthermore reflected when it comes to the classification of efficient vs. responsive supply chains. Whereas functional products would preferable go into efficient supply chains, the main aim of responsive supply chains fits the categorization of innovative product. A comparison of traditional supply chain concepts such as the efficient supply chain and responsive supply chain and a new concept for emerging ICT as the cloud computing area with cloud services as traded products is presented in Table 1.

	Traditional Supply Chain Concepts		Emerging ICT Concepts
	Efficient SC	**Responsive SC**	**Cloud SC**
Primary goal	Supply demand at the lowest level cost	Respond quickly to demand (changes)	Suppy demand at the lowest level of costs and respond quickly to demand
Product design strategy	Maximize performance at the minimum product cost	Create modularity to allow postponement of product differentiation	Create modularity to allow individual setting while maximizing the performance of services
Pricing strategy	Lower margins because price is a prime customer driver	Higher margins, as prices is not a prime customer driver	Lower margins, as high competition an comparable products
Manufacturing strategy	Lower costs through high utilization	Maintain capacity flexibility to meet unexpected demand	High utilization while flexibile reaction on demand
Inventory strategy	Minimize inventory to lower cost	Maintain buffer inventory to meet unexpected demand	Optimize of buffer for unpredicted demand, and best utilization
Lead time strategy	Reduce but not at the expense of costs	Aggressively reduce, even if the costs are significant	Strong Service Level Agreement (SLA) for ad-hoc provision
Supplier strategy	Select based on cost and quality	Select based on speed, flexibility, and quantity	Select on complex optimum speed, cost, and flexibility
Transportation strategy	Greater reliance on low cost modes	Greater reliance on responsive modes	Implement highly responsive and low cost modes

Table 1. Traditional vs. emerging supply chain concepts

From a research perspective, but even more from a business-driven perspective, it is important to understand the supply chains and the related *supply network*. In [19], Kranton and Minehart define the *supply network* as: "a [supply] network is a group of buyers, sellers, and the pattern of the links that connect them, [where] a 'link' is anything that makes possible or adds value to a particular bilateral exchange". Both research and business often focuses on supply chains as connections between exactly one seller and exactly one buyer due to the simplicity of the concept and analytical traceability. However, real world interactions often occur in network structures rather than in a bilateral manner, because:

i) multiple links may enable the pooling of risks,
ii) buyers may share sellers to ensure that sellers have sufficiently high demand to cover investment costs,
iii) more links may enable access to a variety of goods,
iv) sellers may have economies of scope or scales, if they have multiple buyers,
v) possible advantages of diversity and potential future benefits, e.g. buyers could take advantage of sellers investigating in different technologies,
vi) overcoming threshold values in a certain field that is impossible to overcome with one link. E.g., in many environments, a company's gain of adopting a technology may depend on others adopting the same technology.

Along this line it can be stated that the cloud market, composed of products and actors, will probably function as a supply chain with strong characteristics of a supply network.

Kranton and Minehart [19] have developed a model in which they present a theory of investments and exchange in a network. The model assumes that agents do not act cooperatively and that agents cannot write state-contingent, long-term binding contracts to set links, future prices, or side payments. They show the striking result that the change in the expected utility that any buyer sees from adding a link to some seller is precisely the overall social gain from adding that link. In other words, since building links between buyers and sellers is costly, there is a trade-off between building links and pooling risks. They also show that non-cooperatively behaving buyers and sellers can form a socially efficient network structure [26].

These findings directly feed into the discussion whether a cooperation of cloud service providers or single stand-alone clouds should be preferred from a business perspective. As mentioned, the trade-off is the important part to have in mind for this discussion. From observations of the current market situation, it can be derived that ambitious efforts are made in order to create standards, e.g. a common cloud-API [25] and even frameworks for a federation of clouds like RESERVOIR [29].

Federation is a form of symmetric service composition by which cloud providers can rely on third parties offering similar services in order to extend their own capacity and provide unlimited scale. Here again, the costs for building links and pooling risks will lead to a balanced market orchestration.

In the next section we analyze the consumption or provisioning of services along the cloud supply chain. The main focus hereby is set to the subprocess between software providers and infrastructure providers. The result coming out of this subprocess is a composite service, that can be consumed by e.g. an end-customer.

3 Cloud Service Provisioning Information Model

A necessary step to tackle is the development of an information model to provide a uniform view for the different actors involved in service provisioning along the cloud supply chain. In order to so, we need to analyze the different processes involved. Thus, we can say that the information model design is *process-driven*, as it has to consider the requirements of the processes. The analysis is done in the following subsections, after defining the high level requirements for the information model.

3.1 Information Model High Level Requirements

The are several high level requirements that the information model has to address:

Completeness. It has to provide a common model generally applicable to the cloud supply chain. In other words, it must be rich enough to model the information that any process may need.

Concepts and relationships. It has to include not only basic concepts (e.g. *service* or *bill*), but also the relationships between them (e.g. the *service* is related with the *bill* through a *associated-bill* relationship).

Abstraction. It has to provide model independent of low-level unrelevant details, setting the proper level of abstraction. This also involves implementation technology independence.

Standards alignment. It should be based on standards to achieve interoperability along the cloud supply chain. This aspect is specifically addressed in Section 3.3.

3.2 Service Provisioning Subprocesses

The overall service provisioning process involves several subprocesses. In order to be comprehensive enough, the information model needs to take all these subprocesses into account. The remainder of this section describes the most relevant service provisioning subprocesses, their information flows and the requirements they introduce to the information model: service deployment, service monitoring, service accounting and service billing[2]. A simplified process chain diagram shows the relationship between these subprocesses in Figure 2.

Service Deployment Its objective is to set up the service in the cloud infrastructure and make it available to its final users. The main information flow goes from service provider to infrastructure provider, as the former defines the service in terms of service components and associated meta-data (such as an associated SLA) and pass this definition to the infrastructure provider.

[2] This list is not complete, as other subprocesses could be considered as part of service provisioning (e.g. service undeployment). However, for the sake of briefness in this paper, we are addressing just the most significant ones.

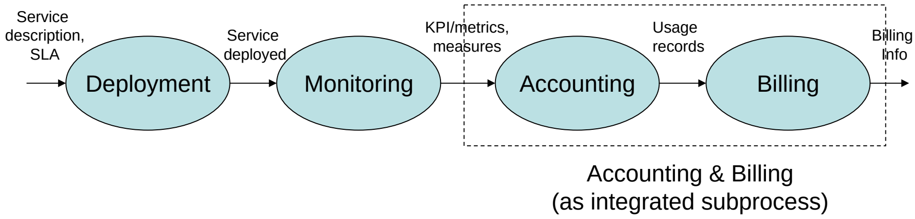

Fig. 2. Service provisioning subprocesses chain

From an information model point of view, this process introduces technical aspects related with the service structure, but also business aspects related with the service SLA to be enforced once the service is deployed. From a technical perspective, the most important are the ones related with service components (both at virtual machine and software component levels), service topology (relationships between components) and sizing (in terms of required hardware resources). Related with SLA, the service has to specify the particular *Service Level Objectives* (SLOs), which basically are a list of relevant *Key Performance Indicators* (KPIs) and target levels for each one of them. The SLA should also include penalties and compensations information, to be used when some of the contracting parties (service and infrastructure providers) broke the agreed terms.

Infrastructure Monitoring Its objective is to measure the metrics associated with the infrastructure itself. These metrics include those that are related to the performance and behaviour of the physical machines in the cloud. Such metrics are used by the cloud management to determine the *best* use of the available resources. From an information model point of view, infrastructure monitoring considers metrics from the virtual machines which actually run the services. Such metrics include CPU usage, memory usage, network usage, disk usage, etc. These metrics[3] are used for both resource utilization and for billing and accounting purposes. Other concepts to take into account are measurement (a particular value for the metric in a given moment of time) and measurement period (metric sampling frequency).

This subprocess relies on service deployment, as in order for a service to be monitored, it needs having being previously deployed.

Service Monitoring Its objective is to measure different metrics related with the status and operation of a running service. These metrics can be related with the performance and health of the service. Some of them are information addressed for the service provider, notified through the proper mechanics, e.g. a web-based dashboard. In this case, an information flow exists from infrastructure provider to service provider. On the contrary, other metrics are for internal infrastructure provider consumption, e.g. the ones used internally for accounting.

[3] The *metric* concept (from a monitoring point of view) is very semantically close to the *KPI* concept (from a SLA point of view, associated to service definition). In fact, both terms are used indistinctively in the literature.

In addition, service monitoring implements the SLA surveillance function and the proper alarms are triggered when SLAs get broken. Whether these alarms reach the service provider in the information flow or are kept as part of infrastructure provider internal information would depend on the particular cloud supply chain case.

As infrastructure monitoring, service monitoring relies in service deployment. From an information model point of view, service monitoring also uses metrics, but of a different nature and introduces new concepts (such as alarm, raised when a service metric crosses a given threshold).

Service Accounting and Billing We describe both subprocesses together, as they are very related. Service accounting objective is to obtain resource usage information, usually in the form of records. It relies on infrastructure and service monitoring, because of usage information is obtained from metric measurements. Accounting process could also consider the measurement of SLA violations provided by monitoring, that could be seen as “negative” usage records to be compensated on billing.

Next, the service billing uses the accounting records in order to produce billing information for the service provider, considering the different resource prices and particular billing rules, e.g. discounts per consumption volume, different prices depending on the daily hour (peak rate at business hours), etc. Although service accounting uses to be an internal process not exposed to the service provider, in the overall, the service accounting and billing chain involves an information flow from the infrastructure provider, which main element is the billing information.

From an information model point of view, there are several concepts related with accounting and billing. Firstly, the payment model, described in detail in Section 5.2, which determines the other concept to consider. For example, for post-paid model, invoice, billing period, etc. need to be taken into account, while for prepaid credit, balance, recharge, expiration threshold, etc. are considered. However, there are some concepts that make sense in both cases, such as customer, price, billing rule, etc.

3.3 Existing Standards

The different processes involved in the cloud supply chain should use a common information model in order to provide a stronger integration. In general, standards aims at interoperability and, in our context, they can help to ensure a common “vocabulary” in the service provision process widely understood and supported across the industry. There are also standards that help to classify the processes in which the information model is used under common frameworks.

In this section, we analyze the most outstanding information models defined by standardization bodies in the field of IT management and how each one can be applied to the cloud supply chain context. In particular, we analyze the Open Virtualization Format (OVF) [12], the Common Information Model (CIM) [10] and the Shared Information Datamodel (SID) [32]. In addition, we analyze how the service provision subprocesses defined above fit in the Enhanced Telecommunications Operations Maps (eTOM) [31] standards framework..

Standard Information Models OVF specifies a portable and vendor-neutral packaging mechanism for Virtual Appliances (VAs), which in the context of cloud computing are the services to be provisioned. This standard is very focused on deployment aspects, so the information model involved by the OVF descriptor (an XML-based meta-data format associated to the VA) suits very well to the requirements of the service deployment subprocess. OVF descriptor addresses service components (in the form of virtual machines), topology (virtual networks among virtual machines) and sizing (hardware resources description). With the appropriated extensions [16] it can also be suitable for the descriptions of service KPIs. However, it lacks the ability to describe the other business concepts related with service deployment, e.g. SLA, SLO, etc.

Regarding CIM, it is a language to define the management information used in distributed computing environments. CIM is object-oriented, technology independent and allows extensibility. In addition, the CIM Schema [11] is a set of management information models specified in CIM covering a broad landscape of the ICT domain, including systems, network, applications, etc. Thus, the CIM Schema and specially the *CIM_Metrics* subschema part of it are very appropriated to describe the technical aspects for the service monitoring and accounting subprocesses. In addition, CIM is indirectly used as part of OVF, in particular for the hardware resources description. However, it lacks some key concepts belonging to business domains needed in the cloud supply chain.

Similar to CIM is SID, part of the NGOSS (Next Generation Operation Support Systems) framework. SID is defined in UML, so it also uses an object-oriented approach. Moreover, SID is sometimes described as a federation of models rather than a stand alone model. In fact, CIM is usually included in this federation. What is important from the point of view of our work is that SID can complement CIM with the business aspects the later is lacking. In particular, SID includes modeling areas such as *Customer*, *SLA*, *Bill*, etc.

A summary relating the information model requirements of the different subprocesses analyzed in Section 3.2 with possible standards information models to address them is shown in Table 2.

Subprocess	**Technical perspective**	**Business perspective**	**Relevant standards**
Deployment	Components, topology, sizing	SLA, SLO, KPI, penalties, compensations	OVF/CIM, SID
Monitoring	Metric, measurement, period, alarm	-	CIM
Accounting	Usage record	-	CIM
Billing	-	Payment model, price, customer, billing rule, balance, credit, invoice	SID

Table 2. Information model requirements analysis per service provisioning subprocess

Standard Processes Frameworks Regarding standard processes frameworks, eTOM provides a neutral reference point for processes development and integration. It considers vertical and horizontal processes, the former related with a single business function involving a specific set of data and people, and the latter defined across several business functions and departments (Figure 3). Applied to the our context, eTOM explicitly considers service provisioning and billing as vertical processes within the Operations area. Monitoring could be considered within the *Assurance* vertical process. The classification for deployment and accounting is not so clear, but they would be seen as the particular projection of provisioning and billing in the *Service Management and Operations* horizontal process, respectively.

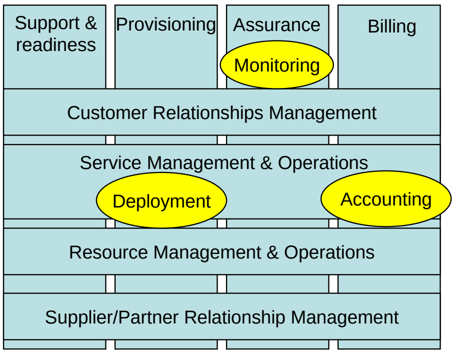
Support & readiness
Provisioning
Assurance
Billing
Monitoring
Customer Relationships Management
Service Management & Operations
Deployment
Accounting
Resource Management & Operations
Supplier/Partner Relationship Management

Fig. 3. Vertical and horizontal processes within the eTOM Operations area

4 Monitoring of Cloud Services along the Supply Chain

Key to facilitating the flow of information in a cloud based environment is a monitoring process via which various metrics regarding the operation of services and infrastructure can be circulated to all participants in a cloud supply chain. Service management processes such as accounting, billing and service level protection require the state of the service to be accurately represented throughout its lifetime. To achieve this, it is necessary to aggregate information from numerous sources to obtain an accurate picture of the provisioning process, correlating application level parameters with that obtained from the infrastructure.

This requires one or more communication channels to exist between components of the infrastructure, the cloud middleware implementing accounting and other processes, service level application elements and the various providers. In addition, a clear understanding must be shared of the type of data that it is important to collect and the means via which it is collected, represented and distributed via these channels.

Focusing primarily on IaaS clouds, we begin in this section by examining how monitoring generally fits in the cloud supply chain, identifying producers and consumers of monitoring data and it's use. We will then discuss state of the art and issues related to federation, data representation and communication.

4.1 Producers and Consumers

In the context of IaaS Clouds, such as RESERVOIR, interactions exist primarily between the infrastructure provider and the service provider. These interactions however are not purely uni-directional. To realise the overall service provisioning process it is necessary for information obtained at various levels of the infrastructure and from the service itself to be distributed at different stages of the service lifecycle.

We distinguish for this purpose the concepts of *producers* and *consumers.* Producers collect monitoring data from the overall environment, in the form of usage measurements. This is achieved typically via *software probes*, which provide the necessary mechanisms to interact with the infrastructure or service, formulating for example appropriate queries on a regular basis. This measurement data can come from numerous sources; this may be raw data gathered from probes in the underlying infrastructure, data gathered from probes embedded in the application, data which is the combination of the raw data into composed Key Performance Indicators (KPIs), or data derived from the analysis of historical raw data and KPI data held in a data store.

Consumers on the other hand will read the monitoring data and will enact some form of action, depending on the process at hand and rules determined by the service or infrastructure provider. It is the following service management processes that we seek to automate in an IaaS context:

Lifecycle Management. Each service deployed in a cloud may be composed of multiple components each with an associated state. A component may for example be awaiting the allocation of resources, deployed, running, in an error state, stopped or un-deployed. Determining when to enact a transition from one particular state to another requires monitoring data to be available regarding the current state of the service and the resources allocated to the service.

Service Level Agreement Protection. In order to ensure a particular level of service, and to be able to meet variations in demand or minimise cost, a service provider may wish to automate the run time reconfiguration of a service, allocating or de-allocating resources as required to meet higher level service level objectives.

Application level monitoring information must be shared between providers for this to be achieved. The state of the application service must be described and exposed in the form of one or more KPIs and enacting various resizing actions when conditions related to these KPIs are met.

Accounting and Billing. The subprocess of accounting and billing, which will be further detailed in the following section, require the collation of measurements into usage records which will summarise the overall level of activity of a service provider, and from which we can derive an appropriate measure of usage costs.

In these three contexts, we require multiple consumers and producers of monitoring data to exist in the overall environment, the operation of which being the responsibility of either the service or infrastructure provider. This is illustrated in Figure 4.

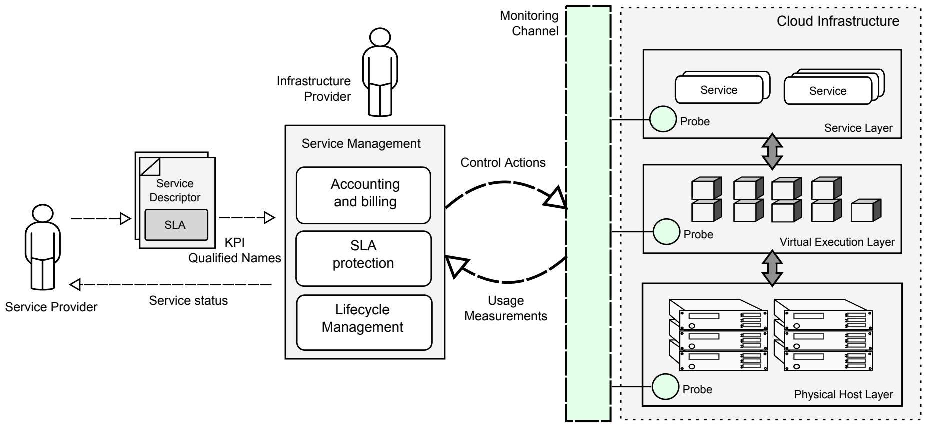

Fig. 4. Monitoring data flow

In this illustration a single monitoring channel is used to aggregate measurements obtained from probes embedded at the service level, within a virtual machine, responsible for the allocation and placement of service components on physical resources, or at the resource layer itself. While the virtual machine and infrastructure probes will be typically operated by the infrastructure provider, it is for the service provider to ensure that any application level monitoring information is supplied by appropriate probes capable of understanding the operation of the application itself.

The data generated by the probes is used as a basis for the sub-processes of collection and analysis. The former relates to the need to obtain and route a specific set of data from the probes to the components that require it. The latter relates to the need to compose and aggregate measurements as required and identify whether specific conditions are met for a particular course of action to occur, notifying specific components in the form of alarms or triggers.

It is important to recognise that monitoring closes the loop from the initial deployment, through execution, and back to the service management, informing the set of actions undertaken by service management components, and allowing immediate feedback to be obtained regarding their impact.

4.2 Monitoring Highly Dynamic Infrastructures

A management system for a cloud requires a monitoring system that can collect all the relevant data in an effective way. The monitoring system has to have a minimal runtime footprint and not be intrusive, so as not to adversely affect the performance of the network itself or the running service applications, taking into account the fact that there may be hundreds or thousands of probes generating measurement data.

Existing monitoring systems such as Ganglia [23], MonaLisa [24], and GridICE [2] have addressed monitoring of large distributed systems. They are designed for the fixed, and relatively slowly changing physical infrastructure that includes servers, services on those servers, routers and switches. However, they have not addressed or assumed a rapidly changing and dynamic infrastructure as seen in clouds. Unlike static physical environments, clouds will see services appear and disappear, or services migrated between clouds, still retaining their capabilities.

The design of the monitoring framework must hence be geared towards maximising *scalability*, *adaptability* to varying loads and rapid changes in service context, and *autonomy*, to ensure minimal intervention. In addition, it must support the distribution of loosely coupled service components across multiple physical locations, *isolating overall services* from one another and providing support for potential *migration* of components from one physical host to another.

4.3 Data Representation and Communication

We briefly address in this section the issue of how monitoring information is represented and distributed. Multiple encoding and transport mechanisms may be used, but there exists a tight coupling between deployment and monitoring which will drive the monitoring process.

Indeed service providers will describe their overall service composition, requirements and elasticity rules or service level objectives in the form a service descriptor or manifest. They will also describe the state of the overall application as a collection of KPIs. The service descriptor hence serves as a basis for identifying the key metrics that an infrastructure provider should listen for in the form of a regular stream of measurements.

As such it is crucial to specify the relationship between the standards and languages relied upon for software architecture and resource requirement description, such as OVF and CIM, previously described in 3.3 and the monitoring events obtained from the various probes embedded in the system.

This may be achieved in numerous ways. In [7], we present an approach to correlating key performance indicators with individual measurements obtained for the purpose of evaluation and analysis using a model driven architecture. This approach aims to complement the normal specification of the software description language (in this case OVF) by formally specifying its ties to the model of the underlying cloud computing infrastructure components and monitoring framework. KPIs are primarily identified using appropriate *qualified names*, which will serve as a basis for identifying corresponding events obtained from probes, as well as their source and purpose in the overall provisioning process.

Once the link between deployment and monitoring is defined, we can then examine the issue of data transportation. Beyond the transport of the measurement data itself, we must also consider the issue of management, how the probes are controlled, turned on and off and reconfigured, and the exchange of meta-data, which will consist of a data dictionary, specifying what the measurement source is, what KPI is associated with it, and any other important information that allow consumers to identify measurements relevant to them. Meta-data represents the information model, which should be distributed separately in order to minimise the amount of data sent with each measurement. Measurements themselves can hence be encoded as efficiently as possible in order to minimise network load.

This typically requires multiple communication planes: a control plane, an information plane and a data plane. An example of an implementation of a monitoring framework designed for clouds is described in [8]. The framework generally provides support for several transport mechanisms; in order to minimise the number of connections established between end points a number of solutions are in place, including the potential use of IP multicasting as a transport mechanism, and intermediate data aggregation points, which will be responsible for collecting data packets and processing these if necessary to produce new performance indicators.

5 Accounting and Billing in the Supply Chain

As mentioned in the previous section, the accounting and billing subprocess is one of the primary consumers of monitoring data. From a service management perspective the requirements and concerns related to accounting can be clearly isolated from the ones related to billing, but we argue that the tight and bi-directional information flow between those components makes them better modelled as an integrated subprocess. Section 5.1 shows how the accounting and billing subprocess is incorporated into the general information flow of the supply chain, and also discusses the information flow between those components.

Accounting is, similarly to monitoring, ultimately about data management. The main difference between the two is that accounting deals with data that has to be made available over a long period of time with high demands on consistency and durability, where as monitoring data have a considerably shorter lifespan and more relaxed demands on historical data availability.

Service billing is performed by applying an arbitrary complex pricing function on data received from the accounting component. Other factors can also be taken into account in the pricing function, such as time of day, historical usage, or any previous violations by the provider. This conversion process between usage and violations into a flow of funds (either positive or negative) also includes dynamic price setting mechanisms for the different type of resources making up the service.

Our previous work on accounting and billing in federated clouds [13] focused on the more technical aspects of accounting and billing in (federated infrastructure) clouds, and here large parts of that work is put into context of the cloud supply chain.

5.1 Information flow

Section 3.3 already emphasized the importance of a unified and coherent representation of data using a common information model, and having such a model in place mitigates one of the main technical hurdles also in the accounting and billing components.

As previously illustrated in Figure 2 and briefly mentioned in Section 3, the main responsibility of the accounting component is to transform and collate KPIs and metrics from the Monitoring component into *usage records* and *violation records*. Usage records summarize the combined usage on different parts of the system for a certain user and a given period of time, structuring the different monitoring infrastructure metrics and KPIs into a single view for the entire service. Violation records occur when the provider does not live up to the agreed terms with regards to, e.g., amount of infrastructure resources allocated or the availability and response time of an online software service. Each violation record relates to a single usage record, clearly defining the relation between expected resource utilization and the actual one.

The information flow between monitoring and accounting is, apart from control commands in some implementations, unidirectional and has no strict time dependencies (although the delay affects the responsiveness of the billing subsystem).

Compared to the relation between monitoring and accounting, accounting and billing have a much more complex relation in terms of information flow and we argue that this is the main reason why these two parts should be regarded as an integrated subprocess. The sections that follows will each present the base functionality of the billing and accounting components, respectively, and there will also be clear explanation on how this functionality affects the information flow.

5.2 Billing Models

Apart from the many ways of construction the pricing functions converting between resource consumption and credits (out of scope for this article) there are also a number of different payment models that are commonly used in cloud settings. These models are in no way unique to clouds and on the contrary they are well known to customers after being used for years in other utility markets, most notably the mobile phone industry.

The basic billing models are;

Post-paid. Consumers are billed for their previous consumption during, e.g., the past month.

Pre-paid. Credits are purchased prior to any allotment of resources, and the consumption of resources will directly affect the amount of credits until they are run out.

Flat rate. The same fee is billed the consumer regardless of the actual consumption. Note that this model is unsuitable for any kind of service without a maximum possible consumption, such as an elastic IaaS service, but can be used for distinct parts which themselves has a well-defined maximum, such as the utilization of the network.

Hybrid. Hybrid models between the above can also be used to affect the way in which the system is used. For instance, a pre-paid / post-paid hybrid where the consumer will be charged as per post-paid when all pre-paid credits have run out (on unfavourable terms compared to pre-paid) will make it favourable for the customer to use the pre-paid model without being totally cut of when credits are running low.

The payment model in use affects the information flow between the accounting and billing component. For example, the post-paid model requests a set of accounting data representing a specified period of time while as the pre-paid model relies on constant updates from the underlying accounting component. As the payment model can change dynamically, the behavior of the accounting component must be controlled from the billing component.

The billing component must also keep track of which usage records and violation records that have already been covered and accounted for to avoid processing the same records twice. This is not only a problem when using, e.g., the hybrid model between pre-paid and post-paid but is also important using the more basic schemes to avoid mismatches between different billing periods (mostly post-paid) and crash resilience (mostly pre-paid). The basic approaches to this problem is either keeping the information on previously processed records in the billing component or by communicating with the accounting component and mark the records as processed. The first solution means that the billing component must maintain a list of previously processed records very similar to that of the accounting component and that these datasets much be kept synchronized. The other solution, on the other hand, introduces another flow of information that goes in the opposite direction of the information flow.

5.3 Accounting Data Management

There are two main tasks of the accounting component, namely converting metrics and violation observation from monitoring into usage records and violation records and also offering persistent and consistent data storage mechanisms for the accounting data.

The record composition process depends to a large extent on the information model being used as different models will have different mappings between the monitoring information and the usage records used for accounting. There are also differences in terms of data representation and in many cases records has to be unified in terms of units and other semantically important fields that may or may not be explicitly defined in the information model. As the composition process depends largely on the information model, and defining the exact information model is well out of scope for this article, we will instead look at the actual data storage.

In a cloud setting where the amount of resources can be seen as unlimited, the accounting component is in effect expected to deal with infinite amounts of highly detailed data persistently over a considerable period of time[4]. The cloud supply chain concept pushes this to the extreme, as large fluctuations in demand is the norm and traditional storage resource provisioning becomes unmanagable.

There are two basic approaches to mitigating this problem; (i) making use of scalable storage where the storage capacity itself can be regarded infinite, or (ii) relax one or more of the requirements on the data to make it manageable. Both of these options are described in more detail here.

i) *Scalable Storage for Accounting Information*: As cloud computing is a lot about mitigating the problems and risks involved with resource provisioning, it might seem very natural to solve also the storage provisioning problem at hand using similar techniques. Although there are established storage systems, such as Cassandra [20] or BigTable [6], capable of storing large amounts of information and also supporting adding and removing resources (effectively supporting scaling-up and scaling-down), these systems have relaxed data consistency and only provide very little support for transactions as a trade off. Emerging system with a clear focus on transaction support in distributed scalable storage systems, such as the very recent ElasTras [9] and CloudTPS [33], are very interesting and might prove to be suitable solutions over time.

ii) *Relaxing Requirements on Accounting Data*; As an alternative, or rather a complement, to scalable storage back-ends is the process or relaxing requirements on the accounting data to reduce the amount of data that has to be managed. As the life-span of the data is usually fixed, and the amount of data depends on the amount of utilization, the only parameter left to consider is the level of detail of the data. In this scenario, this means aggregating several usage records into a single one spanning a larger period of time, forfeiting the lower resolution of each measurement for a more compact data representation.

Aggregating usage records to reduce the amount of data is a very natural and obvious way of mitigating the problem, but making this kind of data aggregation requires a more flexible information model for usage records, and also limits the number of possible billing models in the system. For example, if the usage is aggregated into a daily summary per consumer, the billing process would no longer be able to encompass different prices for day- and night time consumption. In effect, the current billing model(s) affects the manner in which the data can

[4] In some jurisdictions, financial data must be stored and made available on request for ten years.

be aggregated, which creates yet another dependency between the accounting component and the billing component that runs opposite the information flow.

6 Summary

In this paper we have presented the *Cloud Supply Chain.* The results of our work are the definitions and components of such a supply chain for clouds, together with all of the requirements. This supply chain derives from our experience of building systems for supply chain businesses combined with our experience of building service cloud infrastructures within the RESERVOIR EU research project.

In the paper, we presented a motivation for the cloud supply chain, and we described the main components of the chain, such as actors, products, and information and funds. We illustrated the concept of composite services along the supply chain and we also introduced the emerging supply chain concept as compared to traditional ones. We showed a comparison of supply chains and supply networks and concluded, that for cloud computing, the processes between the actors have a notion of both a supply chain as well as a supply network. Furthermore, we motivated the strong requirement of an underlying monitoring framework combined with a billing and accounting framework, both of which have to be based on a common information model.

To define such an uniform information model, the different processes involved in the cloud supply chain have to be examined, analyzing the different information requirements that each one is introducing. We have done this for the service provisioning process, decomposed it into several subprocesses (deployment, monitoring, accounting and billing), identifying information needs, both from a technical and business perspectives. Although this is not an exhaustive process decomposition, it illustrates pretty well the methodology to follow when designing a common information model and could be reproduced for other more fine-grained subprocesses within the cloud supply chain. Considering the key role of standards as a mean of achieve industrial agreement and interoperability, we not only examined the information model from an abstract point of view, but also considered how standard information models (such as OVF, CIM, and SID) are suited to cloud supply chain needs. We have also analyzed standard business processes frameworks, and found that eTOM could be very useful to describe different processes in the cloud computing supply chain.

For the cloud supply chain, we have discussed the issue of monitoring and identified some of the core processes and information that must be exchanged between providers in order to provide appropriate feedback to ensure the correct provisioning of cloud services. This requires multiple producers and consumers of monitoring data to exist, and different sets of interactions to take place during the overall service lifecycle alongside an appropriate monitoring framework, whose requirements were also presented.

Following the common information model and the monitoring section, we examined the relationship between accounting and billing, especially in terms of information flow, and have shown several examples of why the two are, from a modelling point of view, best regarded as a unified subprocess. For billing, different payment models and their implications on the information flow were briefly discussed. For accounting, we primarly focused on the core concepts of record composition and data management, and the implications of data management on the information flow.

Although this paper has set out the basic *Cloud Supply Chain*, in further research we will examine more subprocesses within the supply chain. This will incorporate more complex service scenarios including the role of brokers, various service providers, and federated cloud structures on all levels such as infrastructure, platform, and software. Furthermore, the implications for monitoring, accounting, and billing coming from composite services will be closely examined.

Acknowledgment

This work is partially supported by the European Union through the RESERVOIR project of the 7th Framework Program.

References

1. RESERVOIR Project. `http://www.reservoir-fp7.eu/`, 2008-2011.
2. S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone, and M. C. Vistoli. GridICE: A monitoring service for grid systems. *Future Gener. Comput. Syst.*, 21(4):559–571, 2005.
3. G. Cachon and M. Fisher. Supply chain inventory management and the value of shared information. *Management Science, 46(8):1032-1048*, 2000.
4. C. C. Centre. Moving from on premise to the cloud - the importance of partnership, 2010.
5. C. C. Centre. Solutions in the cloud vs on-premise software solutions, 2010.
6. F. Chang, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber. Bigtable: A distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI'06)*, 2006.
7. C. Chapman, W. Emmerich, F. Galán, S. Clayman, and A. Galis. Software Architecture Definition for On-demand Cloud Provisioning. In *Proceedings of the 19th International Symposium on High Performance Distributed Computing (HPDC)* , New York, NY, USA, 2010. ACM.
8. S. Clayman, A. Galis, C. Chapman, G. Toffetti, L. Rodero-Merino, L. Vaquero, K. Nagin, and B. Rochwerger. Monitoring Service Clouds in the Future Internet. In *Towards the Future Internet - Emerging Trends from European Research*, pages 115–126, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.
9. S. Das, D. Agrawal, and A. Abbadi. Elastras: An elastic transactional data store in the cloud. *HotCloud. USENIX*, 2009.
10. DMTF. CIM Infrastructure Specification. DMTF Document DSP0004 v2.6.0, Mar. 2010.
11. DMTF. CIM Schema, Mar. 2010. [Online, checked on May 2010] `http://www.dmtf.org/standards/cim/cim_schema_v2250`.

12. DMTF. Open Virtualization Format Specification. DMTF Document DSP0243 v1.1.0, Jan. 2010.
13. E. Elmroth, F. Galán, D. Henriksson, and D. Perales. Accounting and billing for federated cloud infrastructures. In *GCC '09: Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing*, pages 268–275, Washington, DC, USA, 2009. IEEE Computer Society.
14. J. Feiman and D. W. Cearley. Economics of the cloud: Business value assessments, 2009.
15. M. Fisher. What is the right supply chain for your product? *Harvard Business Review, pages 105-116*, 1997.
16. F. Galán, A. Sampaio, L. Rodero-Merino, I. Loy, V. Gil, L. M. Vaquero, and M. Wusthoff. Service Specification in Cloud Environments Based on Extensions to Open Standards. In *Proc. of the Fourth International Conference on COMmunication System softWAre and middlewaRE (COMSWARE 2009)*, June 2009.
17. GFI. On-premise vs. cloud-based solutions. Technical report, 2010.
18. E. E. Group. The future of cloud computing - opportunities for european cloud computing beyond 2010, 2010.
19. R. Kranton and D. Minehart. A theory of buyer-seller networks. *American Economic Review, 93(3):485-508.*, 2001.
20. A. Lakshman and P. Malik. Cassandra-A Decentralized Structured Storage System. In *Workshop on Large Scale Distributed Systems and Middleware (LADIS)*, 2009.
21. H. Lee. Aligning supply chain strategies with product uncertainties. *California Management Review, 44(3):105-119*, 2002.
22. X. Li, Y. Li, T. Liu, J. Qiu, and F. Wang. The mothod and tool of cost analysis for cloud computing. pages 93 – 100, Bangalore, 2009.
23. M. L. Massie, B. N. Chun, and D. E. Culler. The ganglia distributed monitoring system: Design, implementation and experience. *Parallel Computing*, 30:2004, 2003.
24. H. Newman, I. Legrand, P. Galvez, R. Voicu, and C. Cirstoiu. MonALISA : A distributed monitoring service architecture. In *Proceedings of CHEP03, La Jolla, California*, 2003.
25. OGF. Open cloud computing interface working group.
26. M. Paulitsch. *Dynamic Coordination of Supply Chains*. PhD thesis, 2003.
27. G. Research. Hype cycle for cloud computing, 2009, 2009.
28. B. Rochwerger, D. Breitgand, E. Levy, A. Galis, et al. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4), 2009.
29. B. Rochwerger, C. Váquez, D. Breitgand, D. Hadas, M. Villari, P. Massonet, E. Levy, A. Galis, I. Llorente, R. Montero, Y. Wolfsthal, K. Nagin, L. Larsson, and F. Galán. An architecture for federated cloud computing. *Cloud Computing*, 2010.
30. P. Roehrig, C. Ferrusi, and A. Shanahan. Major hurdles remain in enterprise cloud services - IT service providers are addressing the technical and business challenges for end users, 2009.
31. TMF. Business Process Framework Suite. Technical Report GB921 Release 8.1, Mar. 2010.
32. TMF. Information Framework (SID) Solution Suite. Technical Report GB922 Release 9.0, Apr. 2010.
33. Z. Wei, G. Pierre, and C. Chi. CloudTPS: Scalable Transactions for Web Applications in the Cloud. Technical Report IR-CS-53, Vrije Universiteit, February 2010.