Section IV
Liability:
From IT Security to Liability

Max Mühlhäuser
Technische Universität Darmstadt, Germany

Andreas Heinemann
Technische Universität Darmstadt, Germany

GENERAL INTRODUCTION TO SECURITY AND LIABILITY

This part of the book covers liability in ubiquitous computing (UC). The term *liability* was deliberately chosen for two reasons. Firstly, due to the established understanding of common terms like *information assurance*, *dependability*, *accountability* and the like. Secondly, to provide a superseding term for **security for ubiquitous computing**, which—unlike traditional IT security—encompasses the following developments:

1. Absorption of security into information assurance, which itself is a more wide-ranging area within computer science.
2. Confluence of IT security with (everyday) real word security and liability issues
3. Tradeoffs between confidentiality and traceability.

The first observation is a general trend in computer science, whereas the second and third will experience rapidly increasing importance in ubiquitous computing specifically. Therefore, the first observation will be briefly addressed now while the others will be treated in distinct sections.

The reader should note that for each observation a sound definition is still missing and is being discussed intensely within the research community. Therefore, we must restrict ourselves to the following informal definition: *liability* means IT security for ubiquitous computing, including "absorption of security into information assurance," "confluence of IT security with real world security/liability," and reflection of the "confidentiality/traceability tradeoffs."

Before we discuss "absorption of security into information assurance" further, we very briefly address security in computer science.

Security in computer science pursues three "classical" goals (also called the *CIA triad*). In terms of data security, these goals are:

1. **Confidentiality:** Data should not be revealed to unauthorized[1] parties

2. **Integrity:** Data should not be altered by unauthorized parties
3. **Availability:** Data should not be made inaccessible by unauthorized parties

Besides data, these goals may concern executables, humans, identities, computer networks, and so forth. What exactly may be concerned is often determined by a concrete application.

Orthogonal to the goals above, a number of general security measures can be distinguished. These measures are achieved by more concrete security means (see the chapter "Security for Ubiquitous Computing"). A concrete means may serve several measures (and goals). The most common measures are:

1. **Concealment:** Ensures access denial for unauthorized parties (example means: encryption, access barriers).
2. **Authentication:** Ensures the correctness of claims – often of claimed identity (example means: digital signatures).
3. **Authorization:** Ensures the possession of rights, for example the right to read/write/alter or execute a file.
4. **Non-Repudiation:** Ensures verifiability of elements of (non) actions—often the identity of the originator of an action or the details of an action.
5. **Anonymity** (contrary to **Non-Repudiation**)**:** The identity of an originator of an action remains confidential/stays anonymous.
6. **Non-Observability** (contrary to **Non-Repudiation**)**:** Execution of an action itself is not verifiable.

This list deliberately left pseudonymity, that is, the use of pseudonyms, aside. We regard the use of pseudonyms as a means that allow an entity to authenticate against a third party but to (some extent) stay anonymous.

Our first observation, **Absorption of Computer Security into Information Assurance**, as stated earlier, is a trend in computer science to broaden the view on security issues. Security in the *classical* sense looks into *intentional* (malicious) causes for malfunctions. Since unintentional causes may yield to similar effects and may be avoided with similar measures, it is reasonable to take *reliability*, *correctness*, and *safety* into consideration as well. Reliability aims to protect from failures, that is, an unintentional change in a component's behaviour. Correctness makes sure there is no design defect present and safety—being a fairly new goal—is a measure to protect against catastrophic effects and catastrophes as causes. Note that *classic* security together with reliability, correctness, and safety is traditionally subsumed under the term *dependability*.

In the context of risk management (i.e., protection measures for restoring a correct function after a malfunction occurs in correlation with the generated costs to deploy these measurements) dependability has been recently termed *information assurance*. In the context of ubiquitous computing, information assurance emphasizes the "CIA triad" together with authentication and non-repudiation. The reason for this is found in the other two observations, namely the confluence of computer security with real world security and a prevalent conflict between confidentiality and traceability. The next section discusses this in greater detail.

LIABILITY IN UBIQUITOUS COMPUTING

Confluence of Computer Security with Real World Security

Ubiquitous computing, in its final stage, will support virtually everything we do in our everyday life. This comes with an extensive penetration of computers and embedded systems in everyday objects. Smart spaces will be the norm. As a consequence, we will depend even more on computers than today. What we see is a confluence of computer security with real world security. Three key aspects must be considered:

a. Due to the exponential growth of both the number of computers involved in our actions and the frequency of independent security-related actions, there is a need for more scalable security than public key infrastructures (PKI) offer. PKI do not scale for two reasons. First, UC nodes vary a lot in respect of computing power and so forth. On nodes with poor resources, PKI-based solutions make use of a shorter key length that might be breakable by resource–rich nodes. Second, in order to check a certificate's validity, a node needs to ask the certificate issuer whether the certificate was revoked or not. This can't be done online and in real time for zillions of nodes, zillions of times per millisecond.
b. In UC settings, we carry out everyday actions supported by computers. Thus, computers have to comply with real-world liability issues in all kinds of contexts. In a business context, UC enabled transactions need to comply with warranty issues, guaranteed services, and guaranteed payments as familiar to the user. Further, compliance with legal regulations, for example, obedience of privacy protection laws, is required. Finally, user actions carried out in a UC world need to comply with access control issues as deployed in the real word, for example, access control to premises, buildings, rooms, appliances, or individual operations of appliances.
c. Another aspect emphasizes the confluence of computer security with real world security, namely the fact that computers will act on behalf of humans and they will interact with humans. Thus, computers have to reflect natural, *humane* concepts relevant for security, for example, *trust*, *reputation*, *recommendation* and the like. We regard trust as a key concept here and discuss trust in detail in the chapter "Trust and Accountability." Also, human-computer interaction (HCI) and *ease-of-use* become key issues, since computers need to convey security issues to humans and need to support them in taking security-related actions.

In Figure 1, points (a) through (c) are listed under "Confluence of comp. security and real-world security."

Tradeoffs Between Conflicting UC Goals

As computer security and real-world security merge, conflicting goals that are present in real life among different parties of a society (individuals, organizations, society as a whole) have to be balanced in UC as well. For one, an individual's right to have privacy has to be traded off against his responsibilities/obligations. As an example, think of *automated working hours recording* based on employee tracking. This might be stated in a contract between employee and employer and illustrates a typical privacy/responsibility trade-off (other terms might be: *conceal/reveal* trade-off, *freedom/control* trade-off, or *autonomy/compliance* trade-off).

We give two examples from the past, where conflicts have already popped up. With the first worldwide recognition of AIDS and the fear of a new epidemic, a public debate on how to deal with HIV patients' medical records arose. Should a patient's records remain private or should the doctor report his HIV patients? The conflicting interests of an individual and the society as a whole are obvious. As our second example, we state the procedure change in selling prepaid mobile phones. While initially it was possible to buy prepaid phones anonymously, today, one has to register with a certified ID for a purchase in many countries due to the intensive use in organized crime as a means of making anonymous phone calls.

Summarizing this paragraph, the liability challenge in UC with respect to conflicting goals is to provide means for adjusting an inevitable trade-off to respective cultural, ethical, and juridical settings—and to its evolution over time [see Langheinrich (2001) for a brief history on how the understanding of individual's privacy changed in the course of time]. In addition, a chosen trade-off has to be understandable by users, that is, its implications need to be conveyed to the user as clearly and simply as possible.

OVERVIEW OF FURTHER CHAPTERS

As *Figure 1* illustrates, we found it inappropriate to turn the key challenges described above one-to-one into book chapters. The reason is the evolutionary development of scientific methods and approaches, which applies to liability like to any other scientific field. This means that we have to look for sound existing research domains that we observe as converging sources for liability in UC. In this respect, we found the three most important fields to be as follows:

The next three chapters present these fields in greater detail. The chapter "Accounting and Billing, Guarantees and Contracts" presents accounting and billing as done by telecommunication companies. We address how contracts are settled electronically and how they are enforced. This includes a treatment on how far the above-mentioned privacy/responsibility trade-offs are supported today. By looking into guarantees and contracts, this chapter discusses how this area contributes to the confluence of computer security and real world security.

The chapter "Security for Ubiquitous Computing" covers security issues by distinguishing three typical UC settings, namely *mobile computing*, *ad hoc interaction*, and *smart spaces.* The discussion should make the reader aware of the broad and varied security challenges and risks present in different settings. The risks derive from the inherent characteristics present in UC settings and the challenges are introduced by typical UC resource and infrastructure limitations. Also, this chapter includes a brief introduction in computer security in general and—based on this knowledge—presents a number of selected measures for liability in UC in detail.

The chapter "Trust and Accountability" is split into two parts. The first part covers trust, a concept that is familiar to humans in real life and helps them to interact in the presence of uncertainty. It outlines how trust can help humans in a UC setting as well. The discussion focuses on trust modelling and the propagation of trust via recommendations. The second part shows how accountability can be enforced in the context of resource sharing in distributed systems. We place the focus on reputation and micropayment schemes.

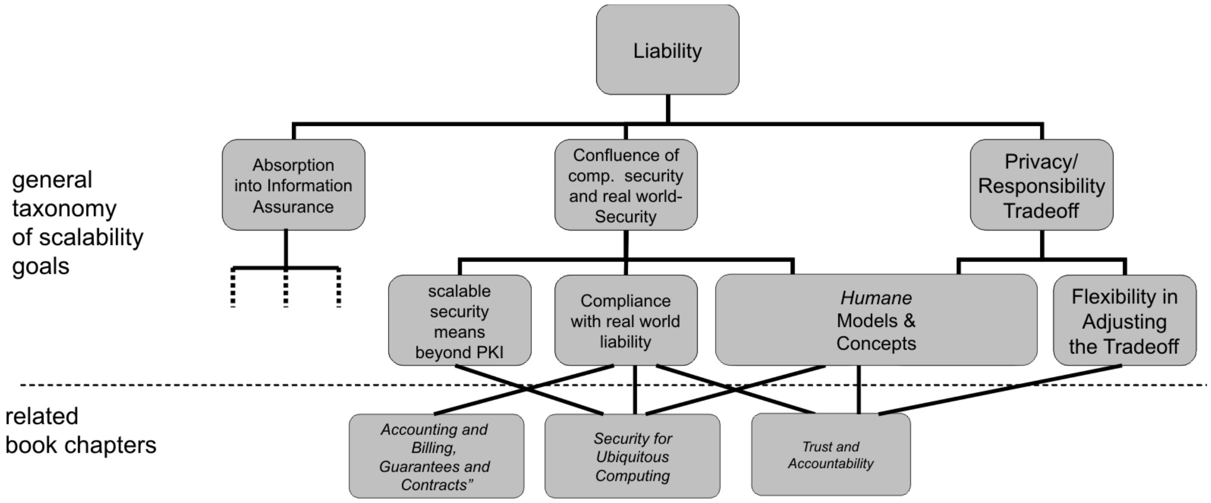
A Precautionary Remark

One would wish to see a better coverage of liability in the remainder, but the topic is still in its infancy—appropriate means, models and concepts are just emerging. Especially the confluence of computer security with real-world security is pending since security for UC must first come up with appropriate and comprehensive solutions, before these can penetrate confluence. As a consequence, the even broader absorption of security in information assurance, as stated earlier, must be left out for the time being.

In order to recall our discussion on UC liability more easily, *Figure 1* breaks down liability and its coverage within this book in a visual manner.

We conclude this introduction by giving the reader pointers to literature that cover security in ubiquitous computing. The short list presents

Figure 1. Liability in UC and its coverage within this book


Liability
general taxonomy of scalability goals
Absorption into Information Assurance
Confluence of comp. security and real world-Security
Privacy/ Responsibility Tradeoff
scalable security means beyond PKI
Compliance with real world liability
Humane Models & Concepts
Flexibility in Adjusting the Tradeoff
related book chapters
Accounting and Billing, Guarantees and Contracts”
Security for Ubiquitous Computing
Trust and Accountability

a selection of survey work with a focus on open/unsolved issues and challenges.

REFERENCES

Görlach, A., Heinemann, A., Terpstra, W.W., & Mühlhäuser, M. (2005). Location privacy. In A. Boukerche (Ed.), *Handbook of algorithms for wireless networking and mobile computing* (pp. 393-411). Chapman & Hall/CRC.

Haque, M., & Ahamed, S.I. (2006). Security in pervasive computing: Current status and open issues. *International Journal of Network Security*, *3*(3), 203-214.

Langheinrich, M. (2001). Privacy by design - Principles of privacy-aware ubiquitous systems. In *Third International Conference on Ubiquitous Computing (UbiComp)* (pp. 273-291). Springer.

Ranganathan, K. (2004). Trustworthy pervasive computing: The hard security problems. In *PerCom Workshops* (pp. 117-121). IEEE Computer Society.

Stajano, F. (2002). *Security for ubiquitous computing*. John Wiley & Sons.

Stajano, F., & Crowcroft, J. (2003). The butt of the iceberg: Hidden security problems of ubiquitous systems. In T. Basten, M. Geilen, & de Groot, H. (Eds.), *Ambient intelligence: Impact on embedded system design* (pp. 91-101). Dordrecht, The Netherlands: Kluwer Academic Publishers.

ENDNOTE

[1] It is obvious that this entails the question: Authorization is done by whom? By the sender, by the receiver, or by a third party?

Chapter XIV
Accounting and Charging:
Guarantees and Contracts

Burkhard Stiller
University of Zürich, Switzerland

David Hausheer
University of Zürich, Switzerland

Jan Gerke
University of Zürich, Switzerland

Peter Racz
University of Zürich, Switzerland

Cristian Morariu
University of Zürich, Switzerland

Martin Waldburger
University of Zürich, Switzerland

ABSTRACT

Charging for IP-based communications determines the overall term for metering or monitoring, accounting, pricing, charge calculation, and billing. These five actions are detailed in this chapter to provide a clear view on their interdependencies as well as their relations to distributed computing. Since an ubiquitous computing approach does require communication means between all entities involved, the provisioning of these communication channels is supported typically by commercial service providers—covering network, transport, and value-added services. Thus, the legal and contractual relationships between customers and providers as well as technological choices of protocols, mechanisms, and parameters define the area of interest here.

INTRODUCTION

Services being offered in a networking environment may range from traditional network access services through to value-added services provided by third party providers. The focus in this chapter has been placed on Internet and Internet Protocol (IP)-based services due to their great importance for basic communication as well as value-added services. In the case of ubiquitous computing, the areas of distributed communications as well as distributed computing merge to form an integrated approach in which those services mentioned provide an integrated portfolio to users and customers. As soon as these service providers and customers are identified, a contractual relation becomes necessary to formulate this contract in a legally enforceable manner. These contracts cover in general any type of service specification to be delivered from a provider to a customer. Specifications must be represented in an automatically interpretable manner in Service Level Agreements (SLAs) and may include computing cycles on a computing cluster, memory in storage area networks, value-added Web services representing business applications, network access, or Quality-of-Service (QoS) support, all of which showing a possible service guarantee level (if at all), numerical values for certain parameters to be monitored, and predefined delivery conditions. While SLAs for overall Information Technology (IT) services in a more traditional sense have been established for quite some time, SLAs in a communications and computing environment which can be interpreted without human interaction still face the problem of being hard to achieve.

The basis for these SLAs and their enforcement can be found in respective accounting mechanisms and protocols, which specify the set of essential operations and functions to be offered in a network. Note that accounting in this context addresses technical accounting questions, and it is not focused on financial accounting means. Since multiple providers may compete in a market-like situation in their service offerings, the need for such a technical service differentiation has to be complemented with suitable mechanisms which enable a service provider to account for these services and—optionally—their service usage. This type of accounting may serve a number of different purposes, such as network management supervision, determining resource bottlenecks in given topologies, or summarizing resource usage in view of subsequent charging. Typically, in a distributed computing environment all of these purposes are highly relevant, since a steady update and change of an existing networking infrastructure takes place, driven by networking equipment vendors, Internet Service Providers (ISP), and third party providers offering alternative value-added services.

Thus, a combination of traditionally pure technology-driven enhancements in network functionality with more recent economically controlled mechanism additions becomes essential for an operable, efficiently manageable, and future-proof communications and networking approach. The basics of each of these two fields, their application in a highly distributed environment, and a number of selected mechanisms will be laid out in this chapter.

Outline

This chapter on accounting and charging as well as guarantees and contracts has been structured into five main sections. While key terminology is defined first, the section “Technologies and Services” provides an overview of relevant technologies and services, which includes roles, accounting, and contracts. “Charging Approaches” extends this view into key charging, sometimes termed billing in public networks, covering basic charging principles, network and transport charging, and Web services and value-added service charging. Finally, “Future Research Directions” draws conclusions and offers a glimpse into major future issues and problems.

TERMINOLOGY

A clear and commonly used definition of key terms is essential. The list below outlines the basic terminology, which covers the most relevant terms related to accounting and charging of Internet services.

- **Account:** An *account* is defined as a repository which can be used to keep and aggregate accounting information, *for example*, the amount of data volume uploaded or downloaded or the number of CPU (Central Processing Unit) cycles used.
- **Accounting:** *Accounting* is the process of collection and aggregation of information on system activities as well as resource and service usage, stored in accounting records. Accounting has manifold purposes and accounting records can therefore serve as input for various subsequent processes, such as charging, network management, traffic management, traffic analysis, trend analysis, or capacity planning.
- **Accounting record:** *Accounting records* hold the accounting data collected by the accounting process.
- **Accountability:** *Accountability* is "the quality or state of being accountable" or the capacity "to account for one's actions" (Merriam-Webster Inc., 2005).
- **Auditing:** "*Auditing* is the verification of the correctness of a process with respect to service delivery. Auditing is done by independent (real-time) monitoring or examination of logged system data in order to test the correctness of operational procedures and detect breaches in security. Auditing of accounting data is the basis for after-usage proof of consumed resources and customer charges." (Rensing, Karsten, & Stiller, 2002).
- **Billing:** *Billing* is the process of consolidating charging information on a per customer basis into a bill.
- **Charge:** *Charge* is the monetary value of a certain service usage and it is the result of the charge calculation for a particular service and user.
- **Charge calculation:** *Charge calculation* is the process to calculate the charge for a particular service usage based on the related accounting records and charging scheme. Charge calculation maps technical values into monetary units.
- **Charging:** *Charging* is used in this section as a synonym for charge calculation. In other more general cases it has been applied to the overall process described from the start of the metering process to the writing of the final bill.
- **Charging/Pricing scheme:** The *charging scheme*—sometimes termed pricing scheme as well—contains the charge calculation rules and prices for services, settled by pricing. The charging scheme is used during the charge calculation.
- **Charging record:** *Charging records* hold the charging data computed during the charge calculation process. Call Detail Records (CDR) determine an example of a dedicated charging record.
- **Customer:** *Customer* is an entity having a business relation with a provider.
- **Guarantee:** A *guarantee* determines a formal assurance that a physical product or an electronic service will be provided under predefined conditions or that it will meet a certain predefined specification.
- **IP flow:** An *IP Flow* is defined as a unidirectional sequence of packets with common characteristics between two endpoints. The common characteristics typically include

source and destination IP addresses, source and destination ports and IP protocol number.

- **Metering:** *Metering* is the process of observing user and system activities by gathering resource and service usage data on network components.
- **Pricing:** *Pricing* defines the method that a particular role (Application Service Provider, Internet Service Provider, or Telecommunication Service Provider) applies to determine the price for a particular service. This includes in a fully distributed approach the collection of information from local resources and/or other roles depending on the pricing strategy that is followed by the peer.
- **QoS:** *Quality-of-Service* (QoS) defines a certain level of performance and quality for all types of data communications, which is expressed in parameter sets according to the special standardization organization involved in the respective communication system's approach. QoS shall be measurable or, in more recent terms, it may determine the perceived QoS of a user in an objective manner.
- **Resource:** A *resource* is a "source of ... supply that can be drawn upon when needed" (Web WordNet 2.0, 2005).
- **Service:** A *service* defines a set of functions and capabilities offered by a provider to a customer. A *value-added service* is defined as a service, which provides value due to extensions of a pure network access service, such as an IP access.
- **Service level agreement:** A *Service Level Agreement* constitutes a contract between an Internet Service Provider (ISP) or a third party service provider and a customer, which may be an ISP or a third party provider as well, to define legally binding service delivery specifications and to commit the ISP and the third party service provider to a required level of service, in case of network service to QoS specifications. The specifications within the Service Level Agreement can be interpreted automatically and require no human interpretation.
- **Session:** A *session* defines the use of a particular service or resource, for example, the download of a file or the use of some amount of computing power. A session always has two session partners, a provider and a consumer.
- **Tariff:** A *tariff* specifies how service usage needs to be accounted and charged for. It is represented by a specific *tariff formula* and a set of *tariff parameters* previously agreed upon between the service provider and the service consumer.
- **User:** *User* is an entity accessing and using a service.

As outlined in Kurtansky and Stiller (2005) the terminology for charging and accounting used

Table 1. Correlation of terminology in IP-based networks and 3G mobile networks (Kurtansky & Stiller, 2006)

IP-based Networks	3G Mobile Networks
Metering	Collecting charging information
Accounting	Charging
Accounting records	Charging Data Record
Charging options	Billing arrangements, Payment methods
Prepaid/postpaid charging	Pre-paid/post-paid billing
Charging mechanism	Charging mechanism
Billing and parts of charging	Rating (Parts of)
Inter-/Multi-Domain Charging/Billing	Accounting

in specifications for the Internet and for different mobile networks, addressing mainly Third Generation (3G) releases, looks different. Thus, Table 1 outlines in the left-hand column the terms used on the Internet and in the right-hand column 3rd Generation Partnership Project's (3GPP, 2005) vocabulary definitions (ETSI, 2005).

TECHNOLOGIES AND SERVICES

To be able to define interactions between providers and customers, a set of suitable roles and partners needs to be determined initially. Additionally, the underlying technology in terms of accounting for services is essential to understand how accounting works, which protocols are in use in the Internet, and which accounting models exist. Finally, the contractual side is discussed, combining the set of roles and relevant accounting parameters to ensure that legally binding Service Level Agreements can be constructed.

Roles and Partners

Considerations on contractual agreements, guarantees, accounting, and billing in ubiquitous computing imply services to be investigated which are offered commercially in a (potentially) fully competitive environment. This initial position determines the set of relevant roles and key players. Accordingly, this section develops the suitable role model for commercial service offerings.

The term *commercial* in this context means that service and resource usage need to be compensated. Compensation is often given by means of financial resources expressed in a widely used currency, turning the currency into a universal intermediary. Resource and service usage, is, however, also conceivable as being compensated by any kind of accepted value expressed by a currency that is accepted by contractual parties.

In ubiquitous computing, two distinct *service* (here in terms of an electronic product, thus, embracing the respective economic notion) types are of relevance for commercial offerings. Pervasiveness requires communications infrastructure to be in place, so that the first service type accordingly embraces network access. On top of this network service, value-added services are offered, determining the second service type category.

Commercial service provisioning involves a wide range of functional steps besides the pure service provisioning phase. These steps comprise support mechanisms which are on the one hand required by legal determinations (including those that are mainly externally imposed, but also self-regulations) and which constitute on the other hand business-critical data. Figure 1 identifies these functional steps in the respective applicable sequential and parallel order as invoked upon a service request received.

Every service is provided and used, while it is provided by making use of resources. Accordingly, the base service provisioning role model consists of three general roles: a service provider, a service user, and a resource provider role, which each see instantiations specific to assumed circumstances. By inclusion of technical and business roles and structured based on the general value chain modeling approach (Porter, 1985), Figure 2 shows an example mapping of business roles to their corresponding value chain components.

Those functional steps as well as those business roles visualized in Figure 1 and Figure 2, respectively, need to be specified in further detail in order to match a specific real-world environment. Accordingly, the main challenges in concreting are determined as follows: Functional steps, such as auditing, need to be mapped to contractual agreements both, in terms of human and machine readable form (cf. the sections “Legal Contracts” and “Service Level Agreements”). Furthermore, these steps have to be technically implemented by underlying mechanisms. Regarding role models, increased complexity needs to be handled, as a role model is required to reflect those various characteristics of an actual cooperation taking place

Figure 1. Functional steps invoked upon service request

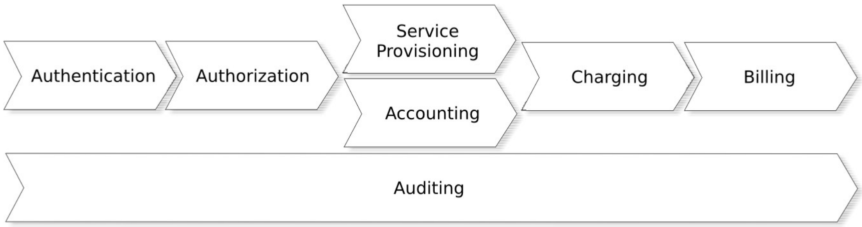
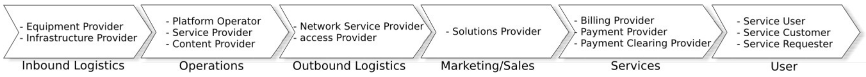


Figure 2. Example business role to value chain mapping



in a value network that potentially incorporates multiple legally independent and geographically dispersed organizations, probably even in a global context.

Metering

Metering is the process of capturing data related to network resource consumption, such as bandwidth, loss error rate, delay, or jitter. This data may further be used for accounting and charging purposes, Intrusion Detection Systems (IDS), or network planning. Metering data may generate inside an active network device (such as a switch or router) or passive nodes may be added on network links for monitoring all the traffic flowing through that link. The node on which the measurement process runs, acts as an *observation point.*

Depending on the final purpose, metered data may be captured with different granularities. The granularity required from the metering device impacts directly the computing requirements the metering device shall have. If an operator is only interested in the load of the links within its network a few simple counters attached to each network link would be enough to achieve this task. If, however, the final goal is to feed this data into an IDS or to perform usage-based charging more advanced and computing intensive mechanisms are needed.

Protocols

Two of the most frequently used protocols for retrieving information from metering devices are SNMP (Simple Network Management Protocol) (Case, Fedor, Schoffstall, & Davin, 1990) and Netflow (Netflow Services and Applications, 2007).

SNMP is an IETF (Internet Engineering Task Force) defined protocol that allows the transmission of management information to/from a network device. These protocols enable network administrators to monitor and manage network performance, find problems within the network and plan for network growth. Each network device in a SNMP-managed network has a management agent that resides on that device. The management agent collects information from different counters and configuration parameters of the managed device and makes them available to a Network Monitoring Station (NMS) via the SNMP protocol. Typical metering information that may be retrieved using SNMP contains interface counters (e.g., for measuring the amount of input/output bytes and packets on the interface, number of errors for input/outgoing transmissions, amount of unicast/multicast packets, size of a routing table, device up-time). The Real-time Traffic Flow Measurement Architecture (RTFM) (Brownlee, 1999) and the Remote Management Information Base (RMON) (Waldbusser, 2006) use SNMP for transporting measured data.

Netflow is a protocol developed by Cisco Systems dedicated to collecting information about IP traffic. Similar protocols to Netflow have been implemented by other network vendors and embedded in their routers (cflowd used by Juniper Networks or NetStream developed by Huawei Technology). Netflow collects information about the network flows passing through a network device or a network link. A network flow may be defined in different ways, but the most typical and widely used definition is a unidirectional sequence of packets having the following IP header fields in common: *IP source*, *IP destination*, *IP protocol*, *source port*, *destination port*. The collected information for a network flow is used to create a flow record when the router detects that the flow has finished. A wide variety of information may be placed in a flow record, but the following is present in all flow records: source and destination IP address, source and destination port number, IP protocol number, number of bytes and packets in the flow. Additional information that may be included is: timestamps for first and last packet observed, Type of Service (ToS) value, TCP (Transmission Control Protocol) flags observed in the flow, input and output interface number, or source and destination Autonomous System (AS) number. The IETF standardizes the IP Flow Export (IPFIX) protocol (IP Flow Information Export, 2007) as the future protocol for transporting flow data.

Metering High-Speed Network Links

Most of the traditional IP flow metering mechanisms scale badly when bandwidth is increased, mainly due to the decrease of time available for processing a single packet (e.g., 4 nanoseconds for an Optical Carrier OC-192 link). Sampling mechanisms have been designed to reduce packet processing work while still achieving a high accuracy of the results (Estan & Varghese, 2002).

Distributed architectures for metering IP flows on high-speed links have been recently proposed in Duffield and Grossglauser (2001), Han, Kim, Ju, and Won-Ki Hong (2002), Kitatsuji and Yamazaki (2004) and Mao, Chen, Wang, and Zheng (2001). They aim at the distribution of the flow-processing task to multiple nodes. Another direction when talking about high-speed link measurements is development of dedicated hardware. Endace Systems (*DAG* Cards, 2007) developed hardware cards that are specialized for packet capturing. The European project SCAMPI (SCAMPI, 2007) investigated strategies for monitoring systems operating at 100 Gbps and beyond.

Open Issues and Research Challenges

Probably the most critical issue in network-related measurements today is metering high capacity links. The IETF working group PSAMP on Packet Sampling (Packet Sampling Working Group, 2007) is standardizing mechanisms for network elements to sample subsets of packets and to transport the sampled data. Another aspect dealt with within the IPFIX (IP Flow Information Export, 2007) working group is the standardization of a protocol for carrying flow records. Another interesting and still quite open field in network measurements is related to Intrusion Detection. Performing IP measurements at multiple places within an IP network and correlating these measurement results in order to obtain an overview on the overall network health determines an important topic to be investigated.

Accounting Principles

In order to be able to keep track of and charge for the provision and use of services and resources, the core function of an accounting mechanism is essential. Thus, the main goal of an accounting mechanism (in the following called *accounting scheme*) is to ensure *accountability* (Dingledine, Freedman, & Molnar, 2001) by providing functionality that enables keeping track of contribution and consumption of resources by service providers and users within a particular application. As such, accounting can serve as a basis for a charging mechanism (cf. the section "Charging Services") or be used as a non-monetary incentive to contribute resources and to punish selfish behavior like freeloading.

Vital accounting mechanisms are the processing of accounting events describing the amount of used resources, the application of respective tariffs, as well as the creation and maintenance of accounts to store and aggregate the accounting information and to keep track of the account balance. One of the main challenges of an accounting mechanism is clearly to bind the accounting information to a real identity or making re-entries of providers and users under a new identity costly and thus unattractive.

Accounting schemes may implement any specific type of accounting, from simple local or centralized accounting to more sophisticated remote or token-based accounting. Individual accounting schemes usually fulfill specific requirements with respect to efficiency, scalability, and economic flexibility, as well as security and trustworthiness, among which there is always a trade-off. An overview of the design space of accounting schemes is given in the following. To generalize the design options, the term *peer* is used as an umbrella term to refer to any entity involved in a particular application. A peer may act as provider and user for several services at the same time.

Local Accounting

With this design option, peers keep accounting information locally, for example, based on receipts that are issued by the counterpart. Local accounting scales very well, however, it typically features bad security properties, as a peer may easily modify accounting information locally, for example, by forging receipts. Such a scheme would therefore only be suitable in trusted environments where security is not important, or for local uses. To increase the trustworthiness of local accounting, receipts could be signed by both transaction partners; however, fraud is still possible through collaboration between peers.

Examples for local accounting schemes are, for example*,* the Peer-to-peer (P2P) system BitTorrent's tit-for-tat mechanism (Cohen, 2003) or eMule's credit system (eMule, 2003). A related approach is SeAl (Ntarmos & Triantafillou, 2004), which creates digital receipts of every transaction and stores them locally. Another possibility to increase trustworthiness is to require peers to make their local accounting information public and auditable as proposed in Ngan, Wallach, and Druschel (2003).

Token-Based Accounting

Token-based accounting is similar to local accounting as it stores accounting information in tokens which are used by peers in exchange for the use of a service and can be aggregated locally. Tokens are different from receipts, as they are typically issued (and signed) by a trusted token issuer, for example, a bank or a quorum of peers (cf. Hausheer, Liebau, N., Mauthe, Steinmetz, & Stiller, 2003a; Hausheer, Liebau, Mauthe, Steinmetz, & Stiller, 2003b). Consequently, token-based accounting usually has a high level of trustworthiness.

The idea of this approach goes back to Chaum, who is regarded as the inventor of eCash, that is electronic payments (cf., for example, Chaum, 1982). One of the main drawbacks of token-based accounting is that tokens may be forged or spent twice like any other virtual currency. Thus, appropriate mechanisms have to be in place which take these problems into account. Double spenders may, for example, be punished by being exposed, as suggested by Chaum, Fiat, and Naor (1990).

An example for token-based accounting in P2P systems is Mojo Nation (Mojo Nation, 2002), where tokens are issued by a central bank. Other token-based mechanisms designed for P2P systems are, for example, Token Accounting (Liebau et al., 2005), PPay (Yang & Garcia-Molina, 2003) or the approach presented in Kamvar, Yang, & Garcia-Molina (2003). A related concept is the Digital Silk Road (Hardy & Tribble, 1993), which has been proposed in the context of agoric systems.

Remote Accounting

In contrast to local accounting, remote accounting is based on the idea that accounting information is held remotely on other peers. Remote peers are third party peers, which are typically different from the peers currently providing or using a particular service that needs to be accounted for. Using remote accounting, accounting information can be distributed and replicated over several peers, which, if designed appropriately, can increase the reliability and availability of the accounting data. In addition, greater credibility or trustworthiness can be achieved when many peers are involved in doing the accounting.

The concept of remote accounting is very general and covers several potential subtypes, such as central, hybrid, and distributed accounting. An overview of the possible variants is given in the following.

Central Accounting

This is the simplest form of remote accounting and is only mentioned for reasons of completeness. Using this type of accounting, accounting information is kept in a centralized place, that is on a central server. An example for an approach which is based on a central server (trusted third party) is proposed in Horne, Pinkas, and Sander (2001). Another central solution is GridBank (Barmouta & Buyya, 2003) which focuses on accounting for Grid services. Central accounting is simple to maintain and control and is thus usually highly trusted. However, such central elements represent a single point of failure and do not scale for a large number of peers.

Hybrid Accounting

Hybrid accounting features the simplicity of central accounting, while being more scalable with respect to the number of peers. In hybrid accounting a dedicated set of peers (so-called super-peers) are used to hold accounting information. Super-peers are typically peers which are highly trusted by a group of peers (clients) attached to them. If the size of such a group is limited, the hybrid approach scales quite well.

However, appropriate incentives need to be given to super-peers to provide the extra accounting efforts. For example, every peer may periodically pay a flat fee to its super-peer covering the costs for keeping and updating the accounting data. So far, the only accounting scheme which is partially based on a hybrid approach is the Token Accounting (Liebau et al., 2005). This approach uses several super-peers as a quorum of trustworthy peers to sign tokens.

Decentralized Accounting

Fully decentralized accounting seems to be most promising approach for distributed applications. It completely distributes the accounting load over all peers. As all peers are equally involved in doing the accounting the scheme scales very well and no payments are necessary to compensate for any accounting costs.

This approach is, for example, followed in Karma (Vishnumurthy, Chandrakumar, & Sirer, 2003), where the consumer sends an account update to the provider, which forwards it to its account holder (called bank). The provider's bank then sends the account update to the consumer's bank, in order to request permission to update the provider's account. If this is confirmed, both accounts are updated accordingly, and the two peers will be notified about the successful transaction so that the service transfer can start.

An important aspect of decentralized accounting is the redundancy of accounts. *Non-redundant* accounting describes the case where every account is held by only one peer, while *redundant* accounting refers to accounts being replicated over several peers. A non-redundant accounting approach supersedes the need for any synchronization between accounts; however, it has some severe drawbacks. If for any reason a particular peer goes offline, accounts held by that peer would temporarily not be accessible anymore. If a peer completely withdraws from the network, the corresponding accounting data would permanently be lost. In addition, a malicious peer could easily modify and misreport the balance of an account it is responsible for. The use of redundancy, that is the replication of accounts over several peers can increase the robustness of decentralized accounting.

PeerMint (Hausheer & Stiller, 2005) uses multiple remote peers to store and aggregate accounting information in a trustworthy and scalable way. It applies a structured P2P overlay network to map accounts onto a redundant set of peers and organize them in an efficient and scalable manner. Unlike Karma (Vishnumurthy et al., 2003) and similar work (cf. Agrawal, Brown, Ojha, & Savage, 2003; Ngan et al., 2003; Ntarmos & Triantafillou 2004), PeerMint uses session mediation peers to maintain and aggregate session information about transactions between peers. This minimizes the possibilities for collusion among peers trying to increase their account balance without actually contributing resources. The scheme is secure in that it ensures the availability and integrity of the accounting data. However, it does not provide confidentiality or privacy, as every peer is, in principle, able to access the accounting data of any other peer.

Another decentralized accounting approach is described in Agrawal et al. (2003). Similar ideas are also pursued in the context of Grid computing (cf. Thigpen, Hacker, McGinnis, & Athey, 2002).

Open Issues and Future Problems

This section has provided an overview of different accounting principles existing today, covering the complete design space from local to remote accounting and from centralized to fully decentralized schemes, each with certain benefits and drawbacks and suitable for particular use cases. However, the use of Internet services in a ubiquitous manner will further increase. Correspondingly, scalability will become the major challenge that has to be addressed by future accounting mechanisms.

Thus, new accounting schemes have to be developed, which will be able to cope with the increased accounting load without compromising on the accuracy. As discussed above, fully decentralized accounting schemes are the most promising approach to store and aggregate accounting information in a scalable and accurate manner. However, in terms of efficiency decentralized accounting mechanisms still lag behind centralized schemes due to a quite high communication overhead. By further optimizing the communication of emerging fully decentralized accounting mechanisms, the efficiency of these schemes can be enhanced without reducing their scalability.

Accounting Protocols

Accounting protocols provide means to transfer accounting data on service and resource usage, enabling a commercial service provisioning. AAA (Authentication, Authorization, and Accounting) protocols enable additionally the communication for user authentication and authorization of service access and resource usage. In the following sections an overview of the most relevant protocols are provided.

RADIUS

The Remote Authentication Dial In User Service (RADIUS) protocol (Rigney, Willens, Rubens, & Simpson, 2000) was introduced to support user authentication in dial-up and terminal server access services, and it is the most widely used AAA protocol in IP networks. RADIUS is a client-server based protocol. Network components requiring AAA support, like a Network Access Server (NAS), operate as RADIUS clients. They request authentication and authorization from the RADIUS server and act according to the response of the server. RADIUS servers are responsible for authenticating the user, authorizing the service request, and informing the client about the result. Requests are forwarded in general based on realms, which are administrative domains users belong to. RADIUS servers can operate as a proxy, forwarding requests to another server if they cannot satisfy the request locally. In this case, the server acts as a client toward the other server. This allows a chain of servers with a more flexible configuration. Request forwarding is commonly used in roaming scenarios, where two or more administrative domains are involved in the service provisioning. RADIUS accounting (Rigney, 2000) extends the protocol with the support of accounting record transfer.

Diameter

The Diameter protocol (Calhoun, Loughney, Guttman, Zorn, & Arkko, 2003) considered as the next generation AAA protocol, is a flexible AAA protocol, consisting of the Diameter base protocol and various Diameter applications. The base protocol defines Diameter entities and specifies the message format together with common functionalities, including Diameter message transport, capability negotiation, error handling, and security functions. Diameter applications enable the flexible extension of the protocol, defining service-specific commands and attributes.

Diameter clients such as a NAS (Network Attached Storage) device are components performing access control and collecting accounting data. Diameter servers are responsible for authentication, authorization, and accounting in a particular realm. In contrast to RADIUS, Diameter allows also server-initiated messages, that is, any node can initiate a message. In that sense, Diameter is a peer-to-peer protocol. Thus, the server can, for example, explicitly instruct the access device to terminate the service of a certain user. Besides Diameter clients and servers, the protocol provides explicit support for agents which can be used to make message routing and message processing more flexible. A Diameter agent provides either relay, proxy, redirect, or translation services.

Accounting support in Diameter was considered from the design on and the base protocol includes basic accounting support. The accounting process is organized in sessions, where sessions provide the means to correlate accounting records belonging to the same service. Diameter supports start, stop, interim accounting records and as well as records for one-time events.

To provide reliable data transfer, Diameter runs over TCP or the Stream Control Transmission Protocol (SCTP). For fail-over purposes Diameter nodes maintain a connection with at least two peers per realm at the same time. Additionally, transport connections are explicitly monitored with watchdog messages to be able to react to failures. Messages are sent typically to the primary peer, but in case of fail-over they are sent to the secondary peer. Diameter explicitly defines the use of IPSec or Transport Layer Security (TLS), providing hop-by-hop security for secure communication between peers.

Similar to RADIUS, Diameter message attributes are coded in Attribute-Value-Pairs (AVP), enabling the transfer of any kinds of parameter in a common representation format. In RADIUS the number of possible attributes is limited to 255 due to the 1 byte long attribute type. In Diameter the AVP code is extended to 4 byte length to provide enough space for future attributes. Additionally, different flags are assigned to AVPs, indicating encryption, mandatory, and vendor-specific AVPs. Additionally, grouped AVPs, consisting of several other AVPs, are supported. Diameter enables the definition of new protocol commands and AVPs in a flexible manner, building Diameter extensions in the form of Diameter applications.

There are several Diameter applications extending the protocol with application specific attributes and messages. The network access server application (Calhoun, Zorn, Spence, & Mitton, 2005) provides the extension for network access services. It defines authentication, authorization, and accounting messages and attributes for network access environments. It derives several AVPs from RADIUS to provide interoperability. The Diameter Extensible Authentication Protocol (EAP) application (Eronen, Loughney, Guttman, Zorn, & Arkko, 2005) specifies Diameter messages and AVPs necessary to support EAP based authentication. The Diameter mobile IPv4 application (Calhoun, Johansson, Perkins, Hiller, & McCann, 2005) provides AAA functionality for mobile IPv4 services, combining mobile IPv4 components and the Diameter AAA infrastructure. The Diameter credit-control application (Hakala, Mattila, Koskinen, Stura, & Loughney, 2005) specifies an extension for real-time credit-control, required in prepaid scenarios.

IPDR

The Internet Protocol Detail Record Organization (IPDR.org, 2007) is an open consortium developing specifications for a common usage data representation and exchange format for IP-based services. The IPDR reference model (IPDR.org, 2004a) is divided into three layers. The *network and service element layer* includes the network and service elements required for the IP-based service provisioning. The *mediation layer* has an interface to the network and service element layer and to the business support system layer and it contains the components responsible for the collection of usage information. The *Business Support Systems (BSS) layer* provides business operation functions of a provider like customer care or billing. The BSS comprises all systems and functions that are required for the business processes of a commercial enterprise. The BSS also exchanges settlement data with foreign BSSs either directly or via a clearinghouse. The model does not define the physical deployment of these entities in a network environment.

To support a flexible and extensible service specific accounting data representation, IPDR defines the IPDR document (IPDR.org, 2004a, IPDR.org, 2004b, & IPDR.org, 2004c), which is a unified data scheme in Extensible Markup Language (XML) format. The IPDR document enables the integration of any kind of service specification. There are common document formats specified for some well-known services, for example, usage information for Voice-over-IP (VoIP) service is specified in IPDR.org (2004d). In order to make the IPDR document transmission more efficient, accounting data can also be represented in the XDR (eXternal Data Representation) format. The IPDR XDR format (IPDR.org, 2004b) is a compact, binary representation of IPDR XML documents.

Further Accounting Protocols

The Terminal Access Controller Access Control System (TACACS+) protocol (Finseth, 1993) is an AAA protocol developed by Cisco Systems. It supports reliable transport of AAA messages over TCP, which makes it resistant against packet loss. The protocol supports start, stop, and interim messages for accounting purposes. Regarding security the protocol provides hop-by-hop authentication, integrity protection, and message confidentiality.

The Simple Network Management Protocol (SNMP) (Case, Mundy, Partain, & Stewart, 2002) is widely deployed in intra-domain management applications. It can be used to collect accounting data typically by polling network equipment in regular intervals. It supports the transfer of accounting records in the form of SNMP Management Information Base (MIB). But SNMP-based accounting has limitations in terms of efficiency and latency. Additionally, SNMP has security deficiencies and problems in inter-domain deployment.

The Common Reliable Accounting for Network Element (CRANE) protocol (Zhang & Elkin, 2002) is another protocol to transfer accounting records. The protocol uses reliable transport protocols, that is TCP and SCTP, and application layer acknowledgments as well. A client can have several simultaneous connections to different servers, which enables fail-over in case of server failure. Security can be supported by IPSec (IP Security) and TLS. The accounting data transmission and representation format is based on templates, which can be negotiated between client and server. The use of templates enables an efficient and compact data transmission.

Next Steps in Accounting

Accounting protocols have been developed and used in IP networks for a long time. At the beginning the focus was on network access services within a single provider domain and accounting protocols supported mainly network access related parameters—like session duration, NAS identifier, NAS port—and IP traffic related parameters—like number of bytes, number of packets transferred. With the conversion of communication networks, IP has become the network technology for all kinds of networks. More complex IP-based service infrastructures are emerging, providing content and value-added services as well.

This results in the core requirement for service-oriented accounting and not only accounting for bits and bytes. Therefore, future accounting protocols should be able to transfer service-oriented accounting records and should provide a flexible accounting record format, since new services will appear frequently. Accounting record formats based on XML, for example, IPDR records, and the AVP format of the Diameter protocol fulfill this requirement. Additionally, the multi-domain aspect, like in Grid networks (cf. the section "Accounting Models"), becomes more important, since services are provided over several provider domains. Therefore, accounting protocols shall support inter-domain interactions.

Because of increasing network link speed and network traffic, a decentralized accounting approach and accounting record processing might become necessary, determining new challenges. Since mobility will further evolve in IP networks, accounting protocols should also become mobility-aware. Additionally, accounting protocols should support prepaid services (cf. the section "Charging Approaches") in the future, because of the high popularity of prepaid charging.

Accounting Models: AAA and A4C

The AAA Architecture (Authentication, Authorization, and Accounting) (De Laat, Gross, Gommans, Vollbrecht, & Spence, 2000) covers a highly effective approach to integrating authentication, authorization, and accounting into a common architecture. This has been extended to achieve the A4C Architecture, which additionally includes, besides traditional AAA functionality, Auditing and Charging functionality. An important aspect of commercial service provisioning is the interaction of A4C infrastructure of different service providers or network operators.

In detail, *Authentication* refers to verifying a user's identity. The authentication will allow later mapping of service usage information to individual users. Figure 3 shows an example of an authentication process. For the mobile terminal to use the access network, the NAS (Network Access Server) needs to know the identity of the user or the device that wants to attach to the network. Using an access link protocol (such as PANA (Protocol for Carrying Authentication for Network Access) (PANA Working Group, 2007) credentials may be encapsulated and sent to the AAA Server via NAS. Depending on the credentials given, the AAA Server may instruct the NAS to allow or deny user's access.

During the *authorization* process a user is allowed or denied access to the service he requested. Authorization typically relates to the service to be provided, so the IETF AAA architecture defines ASMs (Application Specific Modules) to be contacted for deciding whether a user will or will not be allowed to access a specific service.

Accounting is the process of collecting service usage information. This information usually generates during the metering process (see the section "Metering"). According to the AAA architecture the accounting information is sent by an accounting client to an AAA server. In mobile scenarios, multi-domain AAA communication is required. The IETF AAA architecture allows inter-domain AAA interactions by placing an AAA server in every administrative domain. Figure 4 shows an example of a multi-domain AAA communication. The *Mobile Node* which is a client of *Home ISP* may attach to the *Foreign ISP*'s network if a trust relationship exists between the two ISPs. Authentication, authorization and accounting requests will be relayed by the foreign ISP's AAA server to the AAA server of the home ISP. Based on the existing trust relationship, the authentication and authorization performed by the home AAA server may be applicable in the foreign domain by the foreign AAA server. The foreign AAA server may also make local authorization decisions (e.g., even if the user would be authorized to use a certain amount of bandwidth in the home domain, while he is visiting the foreign domain he may have other bandwidth limitations).

The A4C approach determines an extension to the generic AAA architecture by incorporating SLA auditing as well as charging functions. This concept has been developed in several European projects such as Moby Dick (2007), Daidalos (2007), and Akogrimo (2007) as well as industry projects such as DAMMO (Eyermann, Racz, Schaefer, Stiller, & Walter, 2006).

Decentralized Accounting

As introduced in the section "Accounting Principles," decentralized accounting implements the idea of holding accounts on several remote, that is, third-party peers. Figure 5 defines the case of decentralized redundant accounting as used in PeerMint (Hausheer & Stiller, 2005). In this model two types of accounts are distinguished, *session accounts* and *peer accounts*. While session accounts are used to keep accounting information within a particular session, peer accounts aggregate information from several sessions, for example, the total amount of data volume uploaded and downloaded by a particular peer. Peer accounts may also be used to keep information about a peer's reputation or trustworthiness based on its behavior in the past, such as cheating or running a malicious attack.

Both session and peer accounts are held by several independent peers. For every session there is a corresponding *tariff.* Its main purpose is to specify how service usage needs to be accounted for. As such it is used to process *accounting events*

Figure 3. Authentication

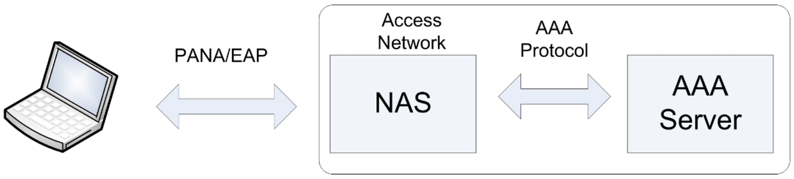
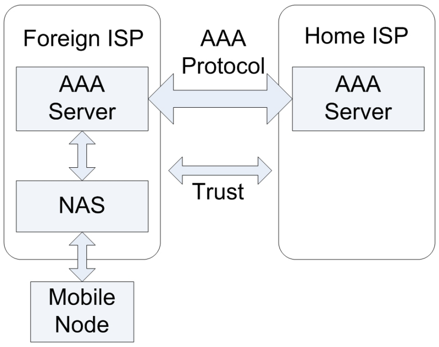


Figure 4. Multi-domain AAA interaction



which are generated by the service instances running on both the provider and the consumer side of a session. It depends on the tariff applied when and by how much the balance of a particular session or peer account is updated. Based on the result of a tariff evaluation, a generic *balance update* is created and forwarded to a particular account. Note that the term balance update is used rather than *charge* to make clear that this does not necessarily imply a monetary payment.

Provider and consumer peers involved in a session send their balance updates to a redundant set of session mediation peers which are responsible for holding the session account for the current session (1). Each session mediation peer checks if the two peers agree and updates the session account accordingly. Whenever a session account triggers a peer account update, the mediation peers send a balance update to the peers holding the respective peer accounts (2).

The two phases may be repeated several times independently. To overcome Byzantine failures (Lamport, Shostak, & Pease, 1982), the resulting account balance is agreed upon using majority decisions. Only if the majority of mediation peers report the same balance update, the peer accounts will be updated. Whenever a peer goes offline or permanently withdraws from the P2P network a new peer takes over its task. The new peer (shown as dashed circle) obtains the current balance from the other account holders.

Grid Accounting

The latest Grid research focused primarily on the accountability of Grid services from a technical perspective and on a metalevel of Virtual Organizations (VO). VOs are seen as the appropriate organization model representing Grid infrastructures that allow for Grid service provisioning across administrative domains. There are many existing grid accounting models. The most prominent ones are APEL (Byrom, Walk, & Kant, 2005), DGAS (Anglano, Barale, Gaido, Guarise, Patania, Piro, Rosso, & Werbrouk, 2006), GASA (Barmouta & Buyya, 2003), GRASP (GRASP, 2006), GSAX (Beardsmore, Hartley, Hawkins, Laws, Magowan, & Twigg, 2002), MOGAS (Lim, Thuan Ho, Zhang, Sung Lee, & Soon Ong, 2005), and Nimrod/G (Barmouta, 2004). All those existing approaches either provide for mechanisms for handling resource usage records or offer a usage tracking service.

When considering commercial Grid services, however, economic and financial principles need to be respected in Grid accounting. This means, for instance, that actual costs have to be allocated to the resource usage of a service. Since VOs are based on the concept of resource virtualization, complexity in service provisioning is increased due to the inherent need for the management of heterogeneous systems and diverse resources located in different service domains. These facts demand a Grid accounting approach based on accountable units that reflect accepted accounting systematics, thus addressing the apparent gap of already existing accounting models. Accordingly, such accountable units are the key means for bridging the respective notions of financial and technical (Grid) accounting. They represent the relevant set of base building elements applicable to every Grid service. This means that every Grid service can be composed from these accountable units, whereas not all elements need to be used in a given service. They embrace four basic hardware functionalities, namely *processing*, *storage*, *transferring*, and *output*.

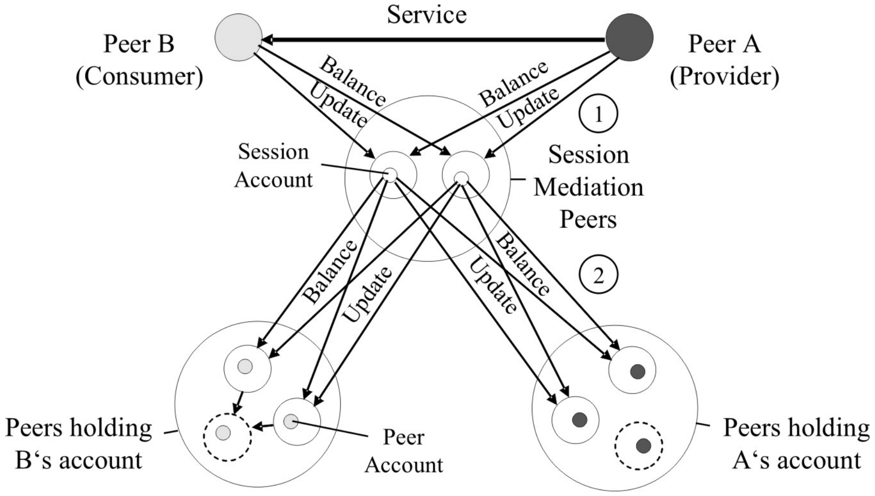
The accounting model introduced in Göhner, Waldburger, Gubler, Dreo Rodosek, and Stiller (2007) allows any service provider in a VO to calculate costs incurred when providing one specific service request. The model relies on two well-known accounting systems. On the one hand, it uses the Traditional Cost Accounting System (TCAS). In TCAS, cost elements originating from financial data are allocated in a first step to cost centers and in a second step to cost objects. This is where the presented accountable units come into play. With that, TCAS determines for the accountable units of a given service the corresponding cost rates. On the other hand, Activity Based Costing (ABC) (Kaplan & Bruns, 1987; Kaplan & Atkinson, 1998) is used. It is driven by the concept of activities. Cost objects are perceived to consume activities, whereas activities consume resources, which are seen as the cost driving event. These costs are assigned to cost objects not by the use of rough percentages (as is the case in TCAS) but rather by identified cause-and-effect relationships. Figure 6 gives an overview of the accounting system and illustrates the central role of its accountable units.

ABC is highly flexible in terms of configurability and applicability. Activities can be defined with the desired level of abstractions so that activities (composed from the basic accountable units) form in several rounds of abstraction the components of a complete IT product. Flexibility, however, not only holds a chance for fine-granular configuration of the accounting systematic but also for a risk of inefficiency. The process of accounting itself is costly itself. Thus, the most difficult task is to find the appropriate level of abstractions needed—in particular with respect to the number of accountable units used—in order to model Grid services with the help of the accounting model.

LEGAL CONTRACTS

Legal contracts are in their very essence promises that are given in exchange with a corresponding value. There are various types of contracts which differ in some fundamental aspects, such as the governing legal determinations to be compliant with. For instance, different contracts may require

Figure 5. Decentralized redundant accounting in PeerMint



different levels of formality (oral or written form). Similarly, contracts under private law need to be differentiated from contracts under administrative law, since they are based on other principles, which leads to potentially conflicting assumptions. In the case of private law contracts, those basic principles embrace—in terms of a non-comprehensive list—the following rationales:

- *Good faith* (bona fides) assumes that contractual parties act honestly according to their respective knowledge. Good faith prevails for contracts under private law, whereas contracts under administrative law recognize the corresponding principle of legal certainty.
- *Pacta sunt servanda* means that contracts are legally binding. This results in the understanding of obligations that are incurred when concluding a contract. Accordingly, contracts typically involve procedures and remedies for the case that a contractual term was breached.
- Contracts need to be concluded *knowingly* (scienter) and *intentionally*. This means on one hand that contractual parties have to be aware of a contract, while on the other hand, contracts require consent from all involved parties, so that contracts are perceived as negotiated agreements.
- *Incompleteness*: Consent is not possible to be assumed for a given aspect if the respective contractual terms are uncertain or incomplete. This aspect is of high importance for automated contract negotiations without human interaction (cf. the section "Service Level Agreements").

Although the above-mentioned basic principles may appear obvious at first, they show some key consequences that are fundamental for civil law as such. A decision on whether an agreement in fact constitutes a contract respecting the full set of mandatory requirements is not always taken unambiguously. For instance, contractual law recognizes so-called quasi-contracts that are inconsistent to a certain extent with the principle of concluding a contract intentionally.

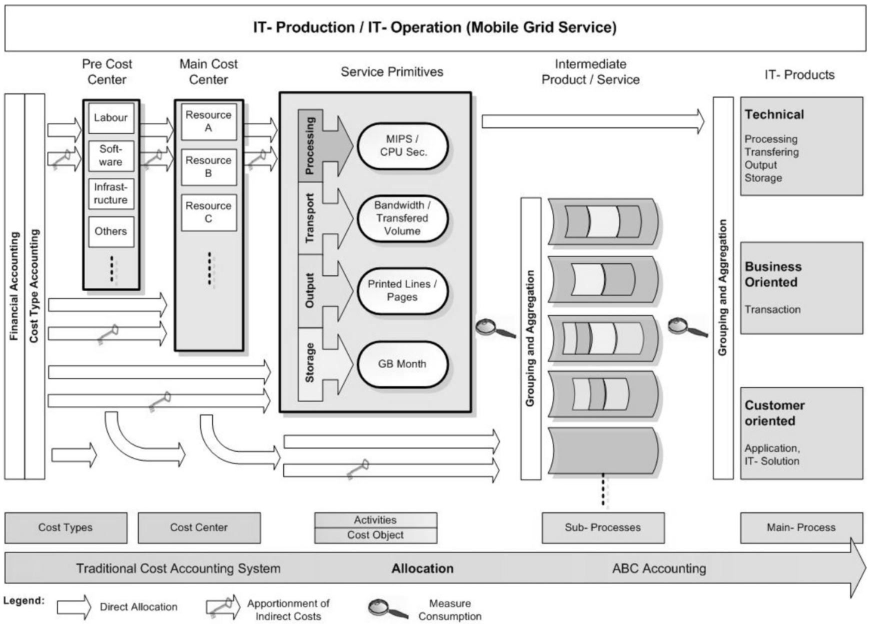
In contrast to the difficult task of finding an answer on whether an agreement can be termed contract, the process of contract conclusion is well defined. Figure 7 models the contract formation process in state diagram form for contracts that fall under United Nations law for the international trade of goods (UNCITRAL, 1980). The depicted automaton visualizes the different possibilities available for the so-called offerer (sender of a proposed agreement, called offer) and offeree (the receiver of an offer, an altered or a counter-offer, respectively) in order to consent to or dissent from a contract. It includes details on forming a contract, whereas it abstracts from the applicable specifics of contract termination.

Upon receipt of an offer, the offeree can either:

- Assent fully, which leads to an acceptance of the offer, rendering the offer a contract,
- Assent but alter it in non-material points, which leads to an acceptance of the slightly modified offer, automatically rendering the offer a contract under the new terms if the original offerer does not object,
- Dissent, which leads to rejection of the offer,
- Reply with a counter-offer which includes changes to the original offer that are of material nature, leading to final acceptance only if explicit consent with the counter-offer is received by the original offerer,
- Ignore the offer, which leads to rejection of the offer. For this reason, offers are equipped with a time frame in which they remain valid.

Material components in this type of contract are the respective terms on price, payment, quality and quantity of the goods, place and time of delivery, liability determinations, and settlement of disputes (UNCITRAL, 1980).

Figure 6. Grid accounting model overview (Gubler, 2006)



During contract negotiation, the user and the provider both aim to maximize their respective welfare. Thus, if the provider P offers a service S with tariff t(S) and cost c(S) to the user U with utility function u(S), the user attempts to alter the offer of S so that his welfare:

$$w(U) = u(S) - t(S) \quad (1)$$

is maximized, while the provider attempts to make a counter offer with a service S which maximizes

$$w(P) = t(S) - c(S). \quad (2)$$

It must be noted that these welfare functions do not take transaction costs into account, for example, the user's cost for waiting for an urgently required service or the provider's cost for reserving resources for the service which could be used otherwise are not taken into account. Transaction costs force user and provider to negotiate a compromise, instead of eternally exchanging altered offers and counter offers which maximize their respective welfare.

Furthermore, a user may negotiate the same service with different providers and choose the one offering him the highest welfare. Thus, in order to maximize the welfare within a service market with many participants, more complex pricing mechanisms, such as the Vickrey Welfare Procurement Auction (VETO) (Gerke & Stiller, 2005), have to be employed, in order to maximize the social welfare of the overall service market and to allocate services and service welfare fairly.

Service Level Agreements

A Service Level Agreement (SLA) is a representation of a contract which specifies the terms of service delivery in such a way that it can be interpreted automatically. This implies that the information within an SLA does not require human interpretation, as the normal contract does (cf. the section "Legal Contracts"). The information within an SLA must enable service user and service provider to carry out their tasks during service usage and service delivery, respectively. Thus, every piece of information contained within an SLA belongs to one of two groups:

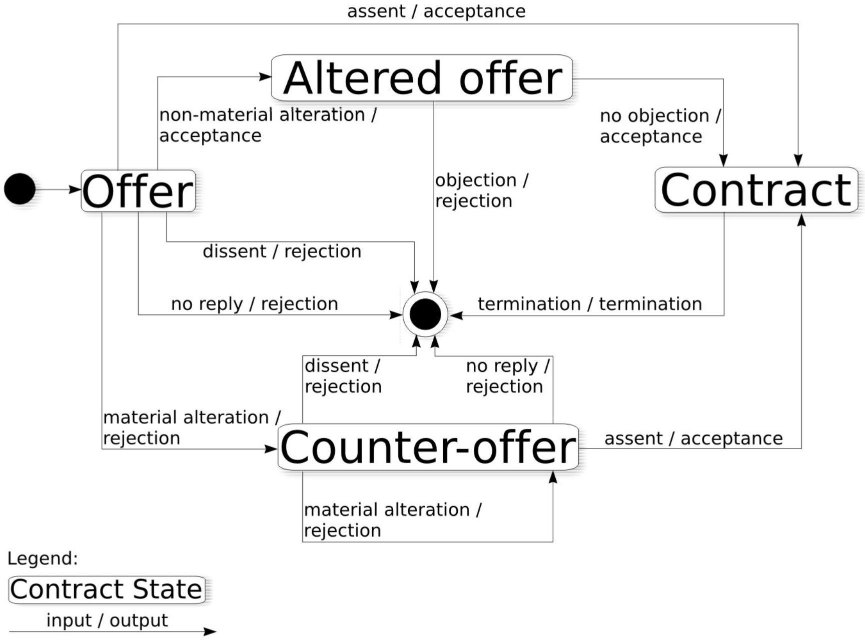
1. Information related to service delivery itself, that is, how it is delivered and used.
2. Information related to the accounting, charging for and payment of the service.

The information related to service delivery is listed in the following:

- **Identities:** The service user and the service provider must be specified through unique identifiers.
- **Service functionality:** The SLA must describe the functionality of the service, for example, that a video with a specific content is to be downloaded.
- **Service parameters:** During contract negotiation, the service user and the service provider negotiate service parameters, for example, the bandwidth a video streaming service will use. These parameters have to be specified in the SLA by key/value pairs.
- **Duration:** The duration of the service delivery phase has to be specified. This can happen in different forms, for example, a fixed start and end time, a fixed duration or even an unlimited duration with a possible service abort by either user or provider. Terms of duration extension can also be specified.

Every service property specified within an SLA has a name, which must be unique within the service description. Furthermore, its specification contains a description of the property, for example, a description of the functionality the property represents. Finally, the specification contains possible values of the property, as

Figure 7. State diagram of contract formation process for contracts governed by United Nations law for the international trade of goods (UNCITRAL, 1980)



well as an explanation of what these values stand for. For some properties, the values do not have any meaning, which is then also specified in the explanation.

Furthermore, in order to enable accounting, charging, and paying for a service, the SLA has to specify the following five types of information not directly related to service delivery and usage:

1. It must specify what information about service usage is collected by the service-using instance and when it is sent to the provider.
2. The tariff must be specified which is used to calculate charges according to the information received about service usage. Various charging schemes can be employed, for example, a fixed amount has to be paid for the service usage or the amount is proportional to the time the service is used.
3. A rule must be specified which determines when the tariff is applied, that is, when the charge is calculated.
4. The method of payment must be specified, for example, bank transfer, credit card, or electronic online payment.
5. Reaction rules and actions can be specified which determine how a service provider reacts to the payment behavior of the service user, for example, the provider might stop the service delivery if the user does not pay.

Information about service usage is specified in so-called *accounting events*. For identification, every event has a name which must be unique within an SLA. The specification of every event must contain the specification of a condition when the event is sent. Furthermore, every event can contain numerous pieces of information about service usage at the time the event is sent. These pieces of information are called *properties*, as they are very similar to service properties. Analogously, their specification must also describe what they represent.

A tariff specification must define an algorithm which takes the received accounting events as input and returns the charge calculated from these events. Thus, the tariff represents a flexible QoS promise made by the service provider to the service user. The tariff should be specified in such a way that when the events show a low QoS, the returned charge is also low. For a very low QoS or a service which was not delivered at all, the tariff can even return negative charges, which means that the provider has to pay compensation fees to the user.

The tariff application role makes it possible to use different types of charging schemes, which charge the service user at different points of service delivery, for example, prepaid or postpaid charging schemes. Various payment methods could be used, for example, off-line or on-line methods. Finally, the specification of reaction rules and actions ensures that a service provider can properly respond to a service user which does not fulfill the SLA. To this end, an SLA, and especially reactions and compensation payments specified within this SLA, must be legally enforceable. Therefore the SLA must be signed by both user and provider. The process of creating and exchanging such a countersigned SLA between both user and provider is not trivial and can only be completely resolved with the help of a trusted third party. This, as well as the duration of validity of service offers and their enforceability, are discussed in Gerke (2006).

CHARGING APPROACHES

Based on the underlying functions of metering and accounting, the path for charging has been opened. While this chapter assumes to have pricing models in place—good overviews of IP-based pricing approaches can be found here—charging forms a major part, which is introduced through standards and recommendations. It is subsequently applied to network and transport services as well as services in more general terms. Finally, value-added service charging concludes this section.

Charging Views, Options, and Mechanisms

Charging is a highly relevant term in the domain considered. Based on the Webster's Dictionary (Merriam-Webster Inc., 2005) "to charge" is explained as "to impose or record a financial obligation." Therefore, the concrete mapping of this term into IP-based communications leads to "charge calculation," which determines the task to calculate the cost of a resource usage by using the price for a given resource unit collected in an accounting record, which in turn determines a particular resource consumption. Thus, charge calculation specifies a function which translates technical values that are also accounted for into monetary units. In turn, the monetary charging information is included in charging records which are utilized for billing purposes. Since prices typically are available for particular resources, the use of accounting records and such prices allow for a customer-specific charge calculation. In general, standards and research work tend to agree on a common understanding of tasks required for charging.

As outlined in Stiller (2003) the European Telecommunications Standardization Institute ETSI (ETSI, 1999) offers a charging definition as follows: "Charging is the determination of the charge units to be assigned to the service utilization (that is the usage of chargeable related elements)." Additionally, Karsten, Schmitt, Stiller, and Wolf (2000) define the full process: "Once these accounting records are collected and prices are determined in full pricing schemes on unit service, for example, encompassing different quality levels for services or service bundles, the data for an invoice need to be calculated. The process of this calculation is termed charge calculation, performing the application of prices of unit services onto accounted for records determining the resource consumption. Thus, the charging function transforms mathematically unequivocal technical parameter values into monetary units. These units need to be collected, if they appear at different locations in the given networking environment, and are stored in charging records. Of course, accounting as well as charging records determine a critical set of data which need to be secured to ensure its integrity when applied to calculate monetary values or when used to compute an invoice's total."

For ATM (Asynchronous Transfer Mode) services the charging process is also termed "rating and discounting process" (Songhurst, 1999) and it is "responsible for the charge calculation according to a specific pricing policy and using the collected usage data." Therefore, these charging mechanisms correlate the service usage and calculate the charge the customer is faced with after the completion of the service utilization. Finally, as outlined in TERMINOLOGY, the charging terminology used in specifications for different mobile networks is different (Kurtansky & Stiller, 2005).

As defined in the terminology section, the charge calculation step calculates the charge for a given service consumption based on accounting records and respective tariffs defined in the SLA. Thus, charge calculation mechanisms are used to implement two different charging options.

- The *prepaid charging option* defines a way in which customers have to be in possession of a certain amount of financial units—typically termed credits or credit points—prior to service usage. Periodical credit checks during service usage are performed and credits are deducted upon the service usage.
- In case of the *postpaid charging option* service charges are collected from the provider's side for a certain period of time. They are debited to the user account after that period, typically by sending an invoice or charging a credit card.

The two charge calculation mechanisms in place differ as follows:

- The *on-line charging mechanism* determines the charge calculation process, which has to be performed in real-time. This implies that the underlying support functions besides the charge calculation—especially accounting and metering—have to be performed in real-time as well.
- In case of an *off-line charging mechanism* no fixed time constraints are defined. Thus, the processing time of the charge calculation may happen at any reasonable time after the service usage.

Additionally, hot billing defines a certain type of charging support, in which the final and last service usage needs to become available in real-time, such as for a phone bill during check out in a hotel. However, it is the operator's own definition of hot billing, such as short time or real-time or volume limits for Charge Detail Record (CDR) closure as well as priority (3GPP, 2005), which diversifies those actual mechanisms required in a given networking and service situation.

Thus, by discussing those two option and mechanism combinations in general, a prepaid on-line charging scheme (e.g., traditional phone card) is as useful as a postpaid off-line combination (e.g., traditional monthly phone bill), while a prepaid off-line charging scheme (e.g., the use of a credit card with a signature on the slip only) may be possible, but may imply risks, and a postpaid on-line charging scheme (performing an on-line account check, but not debiting the money) is possible, but inefficient and not useful.

Charging Components: Focus on Network and Transport

Charging may be applied to a number of different components within communications. Traditionally, as shown in Figure 8 the transport area in grey comprises the access, the connection, and usage-based components. On top of those, content charging may be in place. While any of these components may be considered as a general service being charged for—which is described in a dedicated form in the section "Charging Services" below—the content part has developed in the meantime into a more general value-added service charging, as discussed in the section "Charging Value-Added Services."

Focusing on the traditional transport, the number of packet-based approaches—mainly based on the Internet and IP—often term this component as network charging as well, since the packet determines the unit of interest and the respective layer in the Internet/Department of Defense Reference Model has been termed "Network Layer" as well. Thus, the access component enables a provider to charge for the physical access to a network, such as the ADSL (Asymmetric Digital Subscriber Line) cable or a WLAN (Wireless Local Area Network) access point.

Typically, the charge itself and the respective pricing model will be based on the physically available bandwidth or capacity this point of presence is able to offer to the user. Therefore, this component will be in many cases reflected in a flat fee, or a range of flat fees, depending on the physical layer characteristics.

However, a usage-based component can be found in many data communications. This determines a pre-defined, measurable parameter, which describes the resource usage of a customer on top of the physical access. For example, it may account for the time the user is accessing the network, where sending or receiving packets does not count in terms of volume. Just pure access will be accounted for, typically timed between an authenticated login and a logout of a user at a Network Access Server or an access router. Furthermore, an accounting for volume may be possible, where the amount of bytes, packets, or cells will be metered and utilized to determine subsequent charges.

Of course, these components determine the potential for a charging scheme to be defined for a given technology. While a provider-centric and economic view may prefer all three components to be in place, a provider-centric and technology-driven view may not favor such a complex and costly approach. Therefore, a trade-off between these two contradicting goals needs to be found. Additionally, the customer has to be integrated in such a decision as well, if a charging approach may become acceptable in a market situation, mainly in the sense of offering services and charging options, which are incentive-compatible. Only a competitive price/quality ratio will enable providers to charge for their network services in a viable manner.

Charging Services

In order to charge for the use of services, several sub tasks have to be carried out: The process of measuring service usage (accounting), charging (applying the tariff specified within the SLA), and paying is depicted in Figure 9 and described in the following.

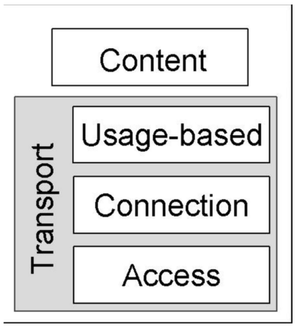
1. During the service delivery and usage, the service using instance repeatedly sends measurements of the service usage to the service user's charging module. These measurements consist of the service properties as observed up to this point of service usage.
2. The charging module forwards the received information, that is, the accounting record, about service usage to the service provider's charging module in an accounting event.
3. Steps 1 and 2 are repeated throughout the service usage phase. How often and when

exactly the measurements are made and sent, is specified in the SLA.

4. After the service delivery has been completed, the service provider's charging module applies the agreed upon tariff to the accounting events received and thus calculates the charge. It then sends a bill specifying this charge to the service user's charging module.
5. The service user's charging module has to ensure that the bill is paid. After the payment has been made, it sends a confirmation of payment back to the service provider's charging module.

Whenever an event describing service usage or service delivery is sent, it follows exactly the event's specification in the SLA, extended with a specification of the measurement values and the time the event is sent. Furthermore, in order to enable the identification of the corresponding SLA and to prevent replay attacks, the SLA identifier and a running sequence number are included in every event. Each event is signed by the service user, in order to make it impossible to refute the service usage afterwards. Thus, enforcing adherence to the SLA by both user and provider is restricted to the granularity of service usage events specified within the SLAs. Ultimately, discrepancies between user and provider can only be resolved through trusted third parties witnessing the service delivery. However, this approach is often more expensive than the service charge itself, rendering it unusable. Rather, on the one hand, providers will stop service delivery or not even start service delivery to users they deem non-trusted, because of their past behavior. Such information can be gained through blacklists or reputation mechanisms. On the other hand, users may simply refuse payment for parts of service delivery which did not comply with the SLA. In turn, if they continuously do this without cause, they will end up being blacklisted or with a very low reputation.

Figure 8. Components of charging



Every bill a provider sends to a user contains the charge for service usage. Furthermore, in order to precisely identify the service which the user is being charged for, the bill contains the SLA. Finally, in order to enable the user to comprehend the charging process, the bill contains all events which were used within the tariff to calculate the charge.

The proof of payment message consists of the bill that was paid. Additionally, it can contain a declaration by the entity which handled the payment, for example, a bank that the payment was made. Charging and payment does not necessarily have to take place only after the service delivery. It could also take place before service delivery, that is, in a prepaid manner (Kurtansky & Stiller, 2006), or at intervals during the service delivery, for example, by making use of token payments such as described in Liebau et al. (2005). Still, which method is used for payment and when it takes place does not change the general interactions between the modules as described in this section, but only the order in which they occur.

Figure 10 illustrates how the five types of charging-relevant information contained within the SLA (cf. the section "Service Level Agreements") are used in the service charging process. The process is started within the charging module prior to the start of the actual service delivery. From this point on, it repeatedly checks whether the tariff application rule is fulfilled. When this is case, the tariff is used to calculate the charge for service usage, which is then used to charge the service user. The service user pays using the payment method specified in the SLA. Then, the reaction rules are applied to the behavior of the service user, resulting in actions such as a continuation of the service, a change of the service parameters or a stop of the service delivery. Finally, if the service is still running, the process returns to its beginning, that is, checks whether the tariff application rule is fulfilled.

It is important to note that the process described in this section focuses on the normal sequence of accounting, charging and paying when service user and service provider behave as they should. However, since money is involved, there is a strong incentive to cheat, for example, for the service user to fake measurements, in order to receive compensation payments for a correctly delivered service. Thus, additional mechanisms have to be provided to prevent such cheating or make it unprofitable. Such mechanisms are described in Gerke (2006) and include a complete discussion of possible attacks, as well as counter measures such as using witnesses or balancing expected revenues of service deliveries against expected losses.

Charging Value-Added Services

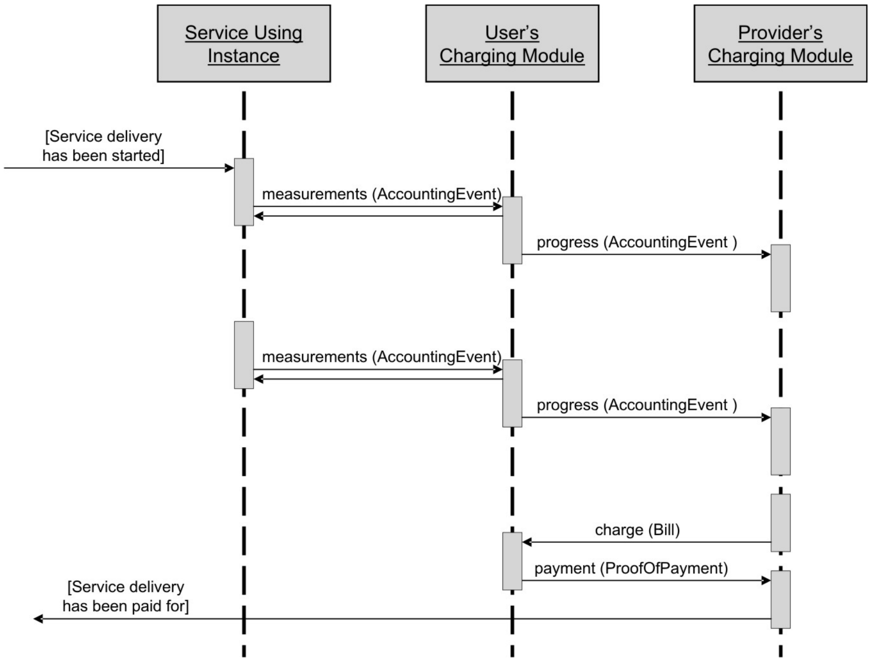
Value-added services (VAS) are usually referred to as non-core services, offering additional, higher-level application services to the user in contrast to the standard service offering. In the IP world VASs include services going beyond standard network access and data transport services. In the telecommunication world services beyond standard voice calls are usually termed as VASs. Value-added services include enhanced messaging, multimedia communication, audio and video streaming, gaming, and electronic content services. Although the charge calculation for value-added services is in general the same as discussed in the sections “Charging Components: Focus on Network and Transport” and “Charging Services,” charging for value-added services might combine charges for related services into a single charging record.

Value-added services are offered either by the network provider itself or by third-party Value-added Service Providers (VASP). A VASP might be tightly coupled with the network provider using its network infrastructure or offer its service independent from the network provider. In the tightly coupled case, the VASP and network provider have a contractual agreement and the service is delivered over the network operated by the network provider. If VASP and network provider are independent, accounting and charge calculation are performed separately and the user receives separate bills from both providers.

The network provider usually has a special role in value-added service provisioning, since the service delivery is performed by its network infrastructure and users typically access a large number of VASPs over a limited number of network providers. If the VASP and network provider have a contractual agreement, accounting, charge calculation, and billing can be delegated partially or completely to the network provider. This tightly coupled case also allows providers to apply more sophisticated charging schemes for VASs, incorporating charges both for network usage and application usage. The user can access services from various VASPs without having a direct contractual relationship with each VASP and the network provider can prepare in this case a single, itemized bill for all accessed services.

The accounting infrastructure (cf. the section “Accounting Models: AAA and A4C”) should enable metering and accounting data collection on the network layer and application layer in a multi-domain environment. This implies that accounting records originating from different service components and different domains should be able to be correlated in a single service delivery session. Additionally, this enables the support of a single, itemized bill.

Figure 9. Accounting, charging, and payment

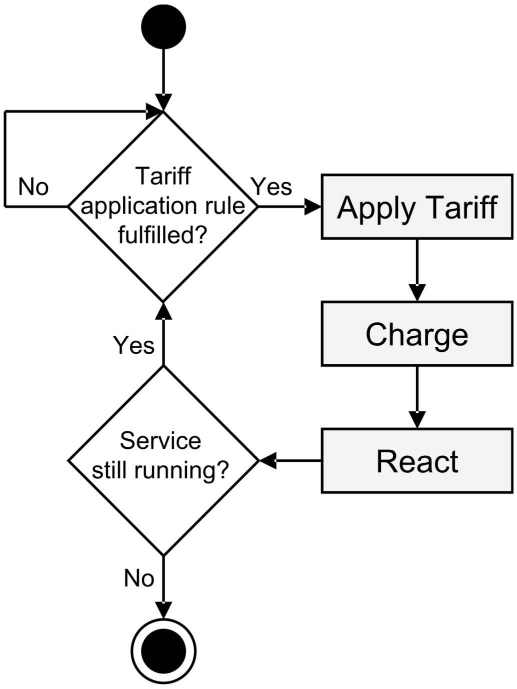

Service Using Instance
User's Charging Module
Provider's Charging Module
[Service delivery has been started]
measurements (AccountingEvent)
progress (AccountingEvent)
measurements (AccountingEvent)
progress (AccountingEvent)
charge (Bill)
payment (ProofOfPayment)
[Service delivery has been paid for]

Since several entities, that is, user, network provider, VASP, are involved in the service consumption and delivery, as well as in the charge calculation process, security measures are essential for the accounting infrastructure. In particular the secure accounting record transfer between domains, providing data origin authentication, integrity, and confidentiality, the secure and trustworthy access of VASPs, and billing for VASs are of key importance.

FUTURE RESEARCH DIRECTIONS

The Internet as a common platform for various advanced services faces constant changes at the moment. Almost every month new services provided on top of the Internet are being introduced, from P2P file-sharing and Internet telephony to Grid computing services and IPTV (IP Television) applications. Moreover, the Internet is currently

Figure 10. Service charging process



facing a new challenge that is commonly known as the Web 2.0, in which the Web browser is merging with the traditional desktop and end-users start to become service providers themselves. Thus, tomorrow's Internet will potentially see a radically different environment where the traditional notion of service consumer and service provider roles no longer holds.

These new applications and market situations will require appropriate new accounting schemes and business models to be put in place. While in the past, service consumers and service providers were clearly two separated roles and the service provider typically was a trusted entity when it came to charging and billing issues, the new environment will see these barriers of roles melting and the new providers may no longer be trusted. Moreover, many new services offered over the Internet such as Skype (2007) or Joost (2007) are provided completely free of charge by making use of the end-users own hardware resources as a distribution and management platform. Another example of this trend is BitTorrent which requires users who download files over this infrastructure to participate in the distribution of the files themselves in a tit-for-tat manner.

One potential solution to this challenge is to come up with new accounting and billing schemes, such as the P2P trading scheme sketched in Hausheer (2006), that combine the benefits of traditional accounting and micro-payment mechanisms with emerging barter-trade patterns like the one implemented by BitTorrent. These new schemes need to take into account that users are able to compensate service consumption to some extent by providing their own services to other users. Based on these new schemes the following new types of service traders can be distinguished: *balanced users*, who use more or less as much as they provide, *over-providers*, who use less than they provide, as well as *over-consumers*, who use more than they provide.

The basic idea of such a new scheme is the following: as long as a user's account is balanced, that is the difference between service consumption and service provision does not exceed a certain threshold, the user does not have to pay or get paid for services. However, when a user starts overusing services and thus exceeds the given threshold, the overuse has to be compensated, otherwise the user will not be granted service anymore. This compensation can then either be achieved by providing services or by paying an amount of money to another user who has reached the threshold in the opposite way. Thus, similarly, a user who is under-using services is able to redeem the accumulated credit for money.

Hence, in principle such a scheme supports remuneration by providing services to other users, but it also allows users to compensate for under or overuse through the transfer of money. Note that since the money transfer is not bound to a single service transaction but rather independent from them, such a scheme can aggregate the outcome from several service transactions and remunerate them together, which will reduce the transaction costs.

The core requirement for new service-oriented accounting—going beyond the accounting for bits and bytes—is a future accounting protocol with the possibility to transfer service-oriented accounting records. Flexible accounting record formats forever changing and new services appearing frequently determine the second requirement. Thirdly, the multi-domain aspect, like in Grid networks (cf. the section "Accounting Models"), is becoming more and more important, since services are provided over several provider domains. Furthermore, accounting protocols should support prepaid services (cf. the section "Charging Approaches") in the future, because of the high popularity of prepaid charging in 3G type networks. Last but not least, the increasing network link speed and network traffic has to be tackled by a decentralized accounting approach and a distributed accounting record processing might become necessary.

Additionally, the negotiation of a Service Level Agreement (SLA) as well its parameters will make sense in some dedicated application cases. This negotiation shall be guided by purely economic incentives, such as the highest welfare achievable (cf. the section "Legal Contracts"). Therefore, the investigation of highly scalable, economically-driven, and technically feasible mechanisms determines an urgent need for suitable solutions in support of charging and accounting for guaranteed services and contracts.

Finally, since new service providers cannot necessarily be trusted anymore, new accounting and billing mechanisms have to take into account that users may act maliciously by providing false accounting records in order to increase their own benefit or simply to harm the system. Thus, new mechanisms like reputation schemes have to be developed and put into place which are able to keep track of a user's behavior in the past in order to possibly predict his trustworthiness in future transactions. Also, appropriate levels of redundancy need to be applied to compensate for the unreliability of individual traders and achieve a high robustness of such new schemes.

REFERENCES

3GPP (2005). *Technical specification group services and system aspects: CR 32215 PS domain charging;* 3GPP TSG-SA5 (Telecom Management), Tdoc S5-054463.

Agrawal, A., Brown, D., Ojha, A., & Savage, A. (2003). *Towards bucking free-riders: Distributed accounting and settlement in peer-to-peer networks*. Jacob School of Engineering Research Review, UCSD.

Akogrimo (2007). *EU IST project.* Retrieved October 16, 2007 2007, from http://www.akogrimo.org

Anglano, C., Barale, S., Gaido, L., Guarise, A., Patania, G., Piro, R., Rosso, F., & Werbrouk, A. (2006). *The distributed grid accounting system (DGAS).* Retrieved October 16, 2007, from http://www.to.infn.it/grid/accounting/main.html

Barmouta, A. (2004). *Authorization and accounting services for the world wide grid*. Unpublished master thesis, University of Western Australia.

Barmouta, A., & Buyya, R. (2003). GridBank: A grid accounting services architecture (GASA) for distributed systems sharing and integration. In *Proceedings of the 17th Annual International Parallel & Distributed Processing Symposium (IPDPS 2003) Workshop on Internet Computing and E-Commerce*, Nice, France.

Beardsmore, A., Hartley, K., Hawkins, S., Laws, S., Magowan, J., & Twigg, A. (2002). *GSAX grid service accounting extensions.* Retrieved October 16, 2007, from http://www.doc.ic.ac.uk/~sjn5/GGF/ggf-rus-gsax-01.pdf.

Brownlee, N. (1999). *RTFM: Applicability statement.* IETF, RFC 2721, October 1999.

Byrom, R., Walk, J., & Kant, D. (2005). *User guide for APEL - Accounting using PBS event logging.* Retrieved October 16, 2007, from http://hepunx.rl.ac.uk/edg/wp3/documentation/apel/apel-user-guide.pdf

Calhoun, P., Johansson, T., Perkins, C., Hiller, T., & McCann, P. (2005). *Diameter mobile IPv4 application*. IETF, RFC 4004, August 2005.

Calhoun, P., Loughney, J., Guttman, E., Zorn, G., & Arkko, G. (2003). *Diameter base protocol*. IETF, RFC 3588, September 2003.

Calhoun, P., Zorn, G., Spence, D., & Mitton, D. (2005). *Diameter network access server application*. IETF, RFC 4005, August 2005.

Case, J., Fedor, M., Schoffstall, M., & Davin, J. (1990). *A simple network management rotocol.* IETF, RFC 1157, May 1990.

Case, J., Mundy, R., Partain, D., & Stewart, B. (2002). *Introduction and applicability statements for internet-standard management framework*. IETF, RFC 3410, December 2002.

Chaum, D. (1982). Blind signatures for untraceable payments. *Advances in Cryptology: Crypto 1982* (pp. 199-203). New York: Plenum Press.

Chaum, D., Fiat, A., & Naor, M. (1990). Untraceable electronic cash. *Crypto 1988* (Vol. 403) (pp. 319-327). Springer Verlag.

Cohen, B. (2003). *Incentives build robustness in BitTorrent*. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*. Berkeley, California.

DAG Cards (2007). Retrieved October 16, 2007, from http://www.endace.com/networkMCards.htm

Daidalos (2007). *EU IST Project*. Retrieved October 16, 2007, from http://www.ist-daidalos.org

De Laat, C., Gross, G., Gommans, L., Vollbrecht, J., & Spence, D.(2000). *Generic AAA architecture*. IETF, RFC 2903, August 2000.

Dingledine, R., Freedman, M., & Molnar, D. (2001). Accountability. *Peer-to-peer: Harnessing the power of disruptive technologies* (1st ed). O'Reilly & Associates.

Duffield, N. G., & Grossglauser, M. (2001). *Trajectory sampling for direct traffic observation. IEEE/ACM Transactions on Networking, 9*(3), 280-292.

eMule Project (2003). *eMule FAQ: Rating & score*. Retrieved October 16, 2007, from http://www.emule-project.net/faq/rating.htm.

Eronen, P., Hiller, T., & Zorn, G. (2005). *Diameter extensible authentication protocol (EAP) application*. IETF, RFC 4072, August 2005.

Estan, C., & Varghese, G. (2002). *New directions in traffic measurement and accounting.* UCSD technical report CS2002-0699, February 2002.

ETSI (1999). *Internet Protocol (IP) based Networks. Parameters and Mechanisms for Charging.* ETSI TR 101 734 V.1.1.1, Sophia Antipolis, France, September 1999.

ETSI (2005). *Digital cellular telecommunications system (Phase 2+). Vocabulary for 3GPP Specifications (3GPP TR 21.905 Ver. 6.10.0 Release 6)*, ETSI TR 121 905 V6.10.0, 2005.

Eyermann, F., Racz, P., Schaefer, C., Stiller, B., & Walter, T. (2006). Diameter-based accounting management for wireless services. In *Proceedings of the IEEE Wireless Communications and Networking Conference 2006 (WCNC'06)*, Las Vegas, Nevada.

Finseth, C. (1993). *An access control protocol, Sometimes Called TACACS*. IETF, RFC 1492, July 1993.

Gerke, J. (2006). *A Generic Peer-to-Peer Architecture for Internet Services*; Dissertation ETH No. 16673, ETH Zürich, Switzerland, 2006

Gerke, J., & Stiller, B. (2005). *VETO — Enabling a P2P-based market for composed services.* In *Proceedings of the IEEE 30th Local Computer Networks Conference (LCN'05)*, Sydney, Australia.

Göhner, M., Waldburger, M., Gubler, F., Dreo Rodosek, G., & Stiller, B. (2007). An accounting model for dynamic virtual organizations. In *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007)*, Rio de Janeiro, Brazil.

GRASP (2006). Retrieved October 16, 2007, from http://eu-grasp.net/english/default.htm

Gubler, F. (2006). *Accountable units for grid services in mobile dynamic virtual organizations*. IFI Diploma Thesis, University of Zürich.

Hakala, H., Mattila, L., Koskinen, J. P., Stura, M., & Loughney, J. (2005). *Diameter credit-control application*. IETF, RFC 4006, August 2005.

Han, S. H., Kim, M. S., Ju, H. T., & Won-Ki Hong, J. (2002). The architecture of NG-MON: A passive network monitoring system for high-speed IP networks. In *Proceedings of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management* (pp 16-27). October 21-23, 2002.

Hardy, N., & Tribble, E. (1993). *The digital silk road*. Retrieved October 16, 2007, from http://www.agorics.com/agorics/dsr.html

Hausheer, D., Liebau, N., Mauthe, A., Steinmetz, R., & Stiller, B. (2003a). Token-based accounting and distributed pricing to introduce market mechanisms in a peer-to-peer file sharing scenario. In *Proceedings of the 3rd IEEE International Conference on Peer-to-Peer Computing*, Linköping, Sweden.

Hausheer, D., Liebau, N., Mauthe, A., Steinmetz, R., & Stiller, B. (2003b). *Towards a market managed peer-to-peer file sharing system using token-based accounting and distributed pricing.* TIK Report Nr. 179, ETH Zürich, TIK, August 2003.

Hausheer, D., & Stiller, B. (2005). PeerMint: Decentralized and secure accounting for peer-to-peer applications. In *Proceedings of the 2005 IFIP Networking Conference*, University of Waterloo, Waterloo Ontario Canada.

Hausheer, D. (2006). *PeerMart: Secured Decentralized Pricing and Accounting for Peer-to-Peer Systems*. Diss. ETH Zürich No. 16200, Shaker Verlag, ISBN 3-8322-4969-9, Aachen, Germany, March 2006.

Horne, B., Pinkas, B., & Sander, T. (2001). Escrow services and incentives in peer-to-peer networks. In *Proceedings of the Electronic Commerce (EC'01)*, Tampa, Florida.

IPDR.org (2004a). *Business solution requirements: Network data management-usage (NDM-U)*. Version 3.5.0.1, November 2004.

IPDR.org (2004b). *IPDR/XDR file encoding format*. Version 3.5.1, November 2004.

IPDR.org (2004c). *IPDR/XML file encoding format*. Version 3.5.0.1, November 2004.

IPDR.org (2004d). *Service specification – Voice over IP (VoIP)*. Version 3.5-A.0.1, November 2004.

IPDR.org (2007). *IPDR website*. Retrieved October 16, 2007, from http://www.ipdr.org

IP Flow Information Export (2007). *IETF*. Retrieved October 16, 2007, from http://www.ietf.org/html.charters/ipfix-charter.html

Joost (2007). Retrieved October 16, 2007, from http://www.joost.com

Kamvar, S., Yang, B., & Garcia-Molina, H. (2003). Addressing the non-cooperation problem in competitive P2P systems. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, California.

Kaplan, R. S., & Bruns, W. (1987). *Accounting and management: A field study perspective*. Harvard Business School Press.

Kaplan, R. S., & Atkinson, A. A. (1998). *Advanced management accounting* (3rd ed). Prentice Hall.

Karsten, M., Schmitt, J., Stiller, B., & Wolf, L. (2000). Charging for packet-switched network communications - Motivation and overview. *Computer Communications, 23*(3), 290-302.

Kitatsuji, Y., & Yamazaki, K. (2004). A distributed real-time tool for IP-flow measurement. *International Symposium on Applications and the Internet*, 2004 (pp. 91- 98).

Kurtansky, P., & Stiller, B. (2006). State of the art prepaid charging for IP services. In *Proceedings of the 4th International Conference on Wires/Wireless Internet Communications (WWIC 2006)*, Bern, Switzerland.

Kurtansky, P., & Stiller, B. (2005). Time interval-based prepaid charging of QoS-enabled IP services. In *Proceeding of the 1st International Workshop (WINE 2005)*, Hong Kong.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems, 4*, 382-401.

Liebau, N., Darlagiannis, V., Mauthe, A., & Steinmetz, R. (2005). *Token-based accounting for P2P-systems*. 14. Fachtagung Kommunikation in Verteilten Systemen 2005 (KiVS 05), February 2005.

Lim, D., Thuan Ho, Q., Zhang, J., Sung Lee, B., & Soon Ong, Y. (2005). MOGAS, A multi-organizational grid accounting system. *SCS International Journal of Information Technology, 11*(4), 84-103.

Mao, Y., Chen, K., Wang, D., & Zheng, W. (2001). *Cluster-based online monitoring system of web traffic.* In *Proceedings of the 3rd International Workshop on Web Information and Data Management*, Atlanta, Georgia.

Merriam-Webster, Inc. (2005). *Merriam-Webster online dictionary*. Retrieved October 17, 2007, from http://www.m-w.com

Moby Dick (2007). *EU IST Project*. Retrieved October 17, 2007, from http://www.ist-mobydick.org

Mojo Nation (2002). *Technical overview.* Retrieved October 17, 2007, from http://www.mojonation.net/docs/technical_overview.shtml.

NetFlow Services and Applications. (2007). *Cisco white paper*. Retrieved October 17, 2007, from http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm

Ngan, T., Wallach, D., & Druschel, P. (2003). *Enforcing fair sharing of peer-to-peer resources.* In *Proceedings of the 2nd International Workshop on P2P Systems (IPTPS)*, Berkeley, California.

Ntarmos, N., & Triantafillou, P. (2004). SeAl: Managing accesses and data in peer-to-peer sharing networks. In *Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P 2004)*, Zurich, Switzerland.

Packet Sampling Working Group (2007). *IETF*. Retrieved October 17, 2007, from http://www.ietf.org/html.charters/psamp-charter.html

PANA Working Group (2007). Retrieved October 17, 2007, from http://www.ietf.org/html.charters/pana-charter.html

Porter, M. E. (1985). *Competitive advantage: Creating and sustaining superior performance*. New York: Free Press.

Rensing, C., Karsten, M., & Stiller, B. (2002). *AAA: A survey and a policy-based architecture and framework*. *IEEE Network, 16*(6), 22-27.

Rigney, C. (2000). *RADIUS accounting*. IETF, RFC 2866, June 2000.

Rigney, C., Willens, S., Rubens, A., & Simpson, W. (2000). *Remote Authentication Dial In User Service (RADIUS)*. IETF, RFC 2865, June 2000.

SCAMPI (2007). *EU IST Project*. Retrieved October 17, 2007, from http://www.ist-scampi.org

Skype (2007). Retrieved October 17, 2007, from http://www.skype.com

Songhurst, D. (Ed.) (1999). *Charging communication networks: From theory to practice.*, Amsterdam, The Netherlands: Elsevier Publisher.

Stiller, B. (2003). A survey of charging internet services. In S. Aidarous & T. Pleyvak (Eds.), *Managing IP Networks*. IEEE Press & Wiley InterScience.

Thigpen, W., Hacker, T., McGinnis, L., & Athey, B. (2002). Distributed accounting on the grid. In *Proceedings of the 6th Joint Conference on Information Sciences* (pp. 1147-1150). 2002.

UNCITRAL (1980). United Nations Commission on International Trade Law. *United Nations* *convention on contracts for the international sale of goods (CISG)* (pp. 1-47). April 1980.

Vishnumurthy, V., Chandrakumar, S., & Sirer, E. G. (2003). KARMA: A secure economic framework for peer-to-peer resource. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, California.

Waldbusser, S. (2006). *Remote network monitoring management information base Version 2.* IETF, RFC 4502, May 2006.

Web WordNet 2.0 (2005). Retrieved October 17, 2007, from http://wordnet.princeton.edu/cgi-bin/webwn

Yang, B., & Garcia-Molina, H. (2003). *PPay:* Micropayments for peer-to-peer systems. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'03)*, Washington, DC.

Zhang, K., & Elkin, E. (2002). XACCT's common reliable accounting for network element (CRANE) protocol specification version 1.0. IETF, RFC 3423, November 2002.

ADDITIONAL READING

3GPP (2005), *Overall high level functionality and architecture impacts of flow based charging*, 3GPP TR 23.125 v6.5.0, 2005.

Aboba, B., Arkko, J., & Harrington, D. (2000). *Introduction to accounting management.* IETF RFC 2975, October 2000.

Bailey, J. P. (1997). The economics of internet interconnection agreements. In L. McKnight & J. P. Bailey (Eds.), *Internet economics* (pp 155-168). Cambridge, MA: MIT Press.

Bhushan, B., Tschichholz, Leray, E., & Donnelly, W. (2001). Federated accounting: Service charging and billing in a business-to-business environment. In *Proceedings of the IFIP/IEEE Integrated Management Symposium*, Seattle, Washington.

Bubendorfer, K., & Komisarcczuk, P. (2006). *A position paper: Towards an utility computing and communications infrastructure.* Retrieved October 17, 2007, from http://www.mcs.vuw.ac.nz/~kris/publications/CIC.pdf

Carle, G., Smirnov, M., & Zseby, T. (1998). Charging and accounting architectures for IP multicast integrated services over ATM. In *Proceedings of the 4th International Symposium on Interworking (Interworking'98)*, Ottawa, Canada.

Estan, C., & Varghese, G. (2003). *New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice*. ACM Transactions on Computer Systems, August 2003.

IETF (2006). *Next steps in signaling (NSIS).* Retrieved October 17, 2007, http://www.ietf.org/html.charters/nsis-charter

Kühne, J., Reimer, U., Schläger, M., Dressler, F., Fan, C., Fessl, A., Klenk, A., & Carle, G. (2005). Architecture for a service-oriented and convergent charging in 3G mobile networks and beyond. In *Proceedings of the 6th IEE Conference on 3G & Beyond (3G 2005)*, London, UK.

Koutsopoulou, M., Alonistioti, A., Gazis, E. & Kaloxylos, A. (2001). Adaptive charging accounting and billing system for the support of advanced business models for VAS provision in 3G systems. In *Proceedings of the IEEE Intl. Symposium on Personal Indoor and Mobile Radio Communication (PIMRC 2001)*, San Diego, California.

Koutsopoulou, M., Kaloxylos, A., Alonistioti, A., Merakos, L., & Kawamura, K. (2004). Charging, accounting and billing management schemes in mobile telecommunication networks and the internet. *IEEE Communications Surveys, 6*(1).

Lamanna, D., Skene, J., & Emmerich, W. (2003). SLAng: A language for defining service level agreements. In *Proceedings of the 9th IEEE Work-*

shop on Future Trends of Distributed Computing Systems (FTDCS'03), San Juan, Puerto Rico.

Maniatis, S. I., Nikolouzou, E. G., & Venieris, I. S. (2004). End-to-end QoS specification issues in the converged all-IP wired and wireless environment. *Communications Magazine, IEEE, 42*(6), 80- 86.

Rigney, C., Willats, W., & Calhoun, P. (2000). *RADIUS extensions*. IETF, RFC 2869, June 2000

Ryan, C., Brazil, J., de Leastar, E., & Foghlú, M. (2002). Workbook approach to algorithm design and service accounting in a component oriented environment. In *Proceedings of the IEEE Workshop on IP Operations and Management*, Dallas, Texas.

Schulzrinne, H., & Hancock, R. (in press). *GIMPS: General internet messaging protocol for signaling*. Internet-Draft.

Sprenkels, R., Parhonyi, R., Pras, A., van Beijnum, B.L., & De Goede, B. (2000). An architecture for reverse charging in the internet. In *Proceedings of the IEEE Workshop on IP-oriented Operations and Management (IPOM00)*, Craow, Poland.