Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Hasan, Jürgen Jähnert,
Sebastian Zander, Burkhard Stiller

Authentication, Authorization, Accounting, and Charging for the Mobile Internet

TIK-Report
Nr. 114, June 2001

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Hasan, Jürgen Jähnert, Sebastian Zander, Burkhard Stiller:
Authentication, Authorization, Accounting, and Charging for the Mobile Internet
June 2001
Version 1
TIK-Report Nr. 114

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology (ETH) Zurich

Institut für Technische Informatik und Kommunikationsnetze,
Eidgenössische Technische Hochschule Zürich

Gloriastrasse 35, ETH-Zentrum, CH-8092 Zürich, Switzerland

Authentication, Authorization, Accounting, and Charging for the Mobile Internet

Hasan[1], Jürgen Jähnert[2], Sebastian Zander[3], Burkhard Stiller[1]
[1] Computer Engineering and Network Laboratory TIK, Swiss Federal Institute of Technology ETH Zürich, Switzerland
[2] Rechenzentrum der Universität Stuttgart RUS, Germany
[3] GMD Fokus, Berlin, Germany
Corresponding Address: ETH Zürich, TIK, Gloriastrasse 35, CH-8092 Zürich, Switzerland, Fax: +41 1 632 1036
E-Mail: hasan@tik.ee.ethz.ch, jaehnert@rus.uni-stuttgart.de, zander@fokus.gmd.de, stiller@tik.ee.ethz.ch

Abstract

Mobile data services across the Internet pave the path for a society of tomorrow. Users will be able to access data, information, and services independent of their location, which will ease the way of business and private life, such as for the traveling field engineer repairing electronic devices at the customers' premises by downloading a new control software or the family on vacation accessing on their Personal Digital Assistant local maps and information on tourist attractions. Having these applications in mind, the Internet technology as it exists today has to be enhanced by a number of different features. An important one is the infrastructure for Authorization, Authentication, Accounting, and Charging (AAAC) those mobile services. These functions will ensure that mobility will not happen into places where not intended or even forbidden and it will enable a commercial operation of a network, which offers services to be sold, such as with varying Quality-of-Service (QoS) or different security degrees. Therefore, existing approaches, such as the traditional AAA (Authorization, Authentication, and Accounting) Architecture of the Internet Research Task Force (IRTF) have to be enhanced and equipped with performing and suitable functionality.

1 Introduction

The evolution of mobile data services outlines a trend towards the coexistence of a variety of wired and wireless overlay networks managed by several actors and covering both indoor and outdoor environments. This popularity of mobile devices is increasing rapidly due to the technology which allows users to connect their host to a visited domain and gain full Internet connectivity from that domain. This trend leads to an important paradigm shift for usage of Internet resources, where security and economic aspects play a major role. Based on an initial authentication and authorization process, these mobile hosts have to be allowed to consume distinct resources in the visited domain, *e.g.*, to generate Internet traffic with a better than best-effort service. In the visited domain, the service definition will probably differ from Service Level Agreements (SLA) valid in the home domain [16]. Furthermore, the Internet Service Provider (ISP) involved in connecting the visiting domain with the home domain probably supports an SLA which differs from the SLA agreed in the home domain. The need for this kind of service from a local domain requires authorization of the mobile user, which directly leads to authentication. In many cases, the ISP of a visited domain offers this service to a mobile user only, if it is assured that he gets paid for the service. This requires an adequate accounting and charging concept, considering the quality of the service provided in the visited domain as well as other service-relevant characteristics. A client requiring resources of a visited domain is requested to provide credentials, which can be authenticated before access to these resources is permitted.

Within the Internet Engineering Task Force (IETF) and Internet Research Task Force (IRTF) architecture and infrastructure of Authorization, Authentication and Accounting (AAA) services is defined and standardized [17], [18], [6], [2], and [12]. The provision of this service in the Mobile IP environment will require inter-domain exchange of this authentication, authorization, accounting, and billing information [7]. Several Internet drafts are proposed to take into account issues on Mobile IPv6 and policy-based networking [10], [4]. An entity requesting this kind of credentials is the AAA local authority (AAAL). In general, the AAAL itself may not have enough information stored locally to carry out a verification of credentials and must consult the home authority (AAAH), which is in charge of the representation of the mobile users permissions towards other networks. These types of credentials are related to a lifetime. The communication between the two authority instances must be secure. Once authenticated, the mobile user must be authorized to access services and resources within the foreign domain.

1.1 Application Scenario and Objectives

Within this context, consider the following sketchy scenario, which demands for an open and modular AAAC Architecture (Authentication, Authorization, Accounting, and Charging) for dealing with all of these aspects in an integrated manner. A medium sized enterprise with a number of external consultants besides the staff working in offices operates a small local access network to perform their communication needs. It may require a defined level of service determined by Quality-of-Service (QoS)-aware applications, such as stringent bandwidth allocations for Computer Supported Collaborative Work (CSCW) applications between local and remote consultants, on-line and real-time exchange of information on technical component availability in the market, and loss-free and secured delivery of financial data. The detailed set of access and utilization rules depend on the project group carrying out the work, which in turn requires the determination of flexible and adaptable policies for establishing and maintaining these working relations. They must be provided in an secure, open, modular, and flexible manner. These essential requirements are important, *e.g.*, because adaptivity and openness at the level of policies will allow for reuse of the same set of modules in many different usage scenarios.

Therefore, the objectives of the AAAC work embedded in the 5th Framework EU project MobyDick (Mobility and Differentiated Services in a Future IP Network) encompass the facilitation of the deployment of an ubiquitous Mobile IPv6 infrastructure through a best-suited and pragmatic use of an evolutionary AAAC architecture based on the IRTF AAA Architecture proposal [12]. Based on the background scenario described above, the set of important communication, functionality, and performance requirements needs to be addressed to allow for a suitable solution for access policies, authorization, authentication, accounting, and charging and its mobile devices. Within this context different mobility mechanisms will be taken into account, since current systems do only support terminal mobility which means that systems do not support mobility transparent to the user currently using a mobile end-system.

In terms of communication requirements the need for mobile IPv6-based services with service differentiation methods is demanding. The Internet environment, in turn, requires the inclusion of further functional requirements. A variety of value-added service add-ons, such as secure access (authenticated and authorized), QoS-controlled service provisioning which is integrated into the mobility management, and multicast services for CSCW applications, are essential for integrated communication solutions. Depending on the service utilized by an external access, charging for resource-intensive communication tasks is required allowing for different access technologies and qualities for roaming users. Within this context is should be mentioned that the requirements to an integrated AAAC framework differ significantly when different business models are deployed. A pre-paid business model, which obtained high popularity in the currently deployed 2GPP (2nd Generation Partnership Project) networks has completely different requirements than a fixed contract business model.

1.2 Terminology

The overall terminology applied for the AAAC work will follow the terminology document of MobyDick as presented in [9]. Additional AAAC-internal terms are enlisted at this stage within this document's section only. The terminologies related to policy are taken from [19], which are not covered by [9].

- **AAAC Architecture**
  The AAAC Architecture defines the overall architecture of those modules and components required to offer a full set of AAAC tasks.
- **AAAC System**
  AAAC System is used to identify the particular module required in the AAAC Architecture to perform as an AAAC client or as an AAAC server.
- **AAAC System Architecture**
  The AAAC System Architecture is used to specify the particular architecture required in the AAAC System to design and implement AAAC clients and servers.
- **Policy Decision Point (PDP)**
  A logical entity that makes policy decisions for itself or for other network elements that request such decisions [20].
- **Policy Enforcement Point (PEP)**
  A logical entity that enforces policy decisions [20].
- **Policy Repository (PR)**
  A policy repository can be defined from three perspectives:
  (a) A specific data store that holds policy rules, their conditions and actions, and related policy data. A database or directory would be an example of such a store.
  (b) A logical container representing the administrative scope and naming of policy rules, their conditions and actions, and related policy data. A "QoS policy" domain would be an example of such a container. This logical perspective will be used within this document.
  (c) A more restrictive definition than the prior one exists in [13]. A Policy Repository is a model abstraction representing an administratively defined, logical container for reusable policy elements.
- **Policy Server**
  A policy server determines a marketing term whose definition is not precise. Originally, [13] referenced a policy server. As the RFC evolved, this term became more precise and known as the Policy Decision Point (PDP). Today, the term policy server is used in marketing brochures and other literature to refer specifically to a PDP or to another entity that uses/services policies.

1.3 Related Work

The IETF and the IRTF are concerned about AAA issues. While the working group of the IETF discusses and standardizes some AAA protocols [12], [17], [18], the IRTF works on the definition of an overall AAA Architecture (AAAArch) [8]. Therefore, the IRTF has discussed how the relevant architecture elements are arranged, which are combined, where they are located, and how they will interact. The proposal depicted in Figure 1 is based on AAAArch [18]. The entities encompass a generic AAA Server, an Application Specific Module (ASM), a policy repository, and an event log. The AAA Server evaluates policies by applying Rule-based Engines (RBE), whereas the ASM interacts with the service equipment. The RBE resides inside an AAA server. The AAA server receives service requests from the Service Equipment (SE) via an ASM or form other AAA servers. On one hand, a request received by the AAA server is inspected by AAA servers considering policies stored in the Policy Repository (PR). To evaluate policy conditions, it may be necessary to consult other AAA servers or the status of the service

equipment. This is done firstly by sending requests to other AAA servers and secondly via an ASM. ASMs are needed additionally to enforce policy actions. Therefore, ASMs configure the service equipment and provision a service. On the other hand, policy actions are taken by the AAA server itself. It holds session states, records accounting data, and logs actions [12], [11]. The number of protocols used in this architecture according to Figure 1 include (1) as a special AAA protocol, which is assumed to be standardized in the research group, (2) a particular Application Programing Interface (API) or the AAA protocol also, (3) depending on the implementation of the PR, Light-weight Directory Access Protocol (LDAP) or an API (4) an application specific protocol.

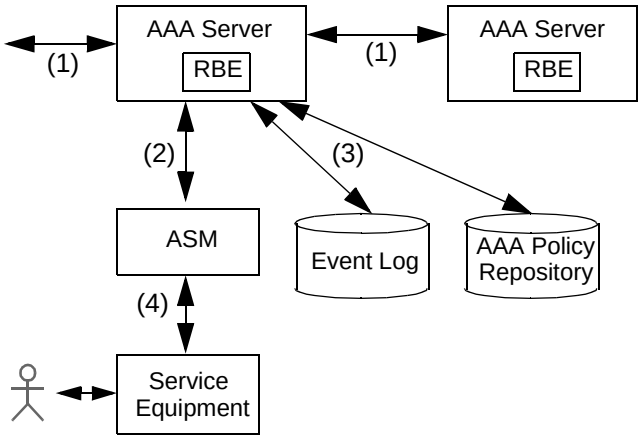

Figure 1: AAA Architecture and Policy Views

1.4 Outline

While Section 2 presents and discusses major requirements of an AAAC Architecture, the proposed Architecture is introduced in Section 3. Use cases are exemplified in Section 4, which are based on the technology and application scenario sketched above. Finally, Section 5 concludes the current state of the work and proposes next steps to be addressed.

2 Requirements

The overall requirements to be addressed for a new AAAC Architecture encompass at least the following five areas, where mobility is the driving force, where QoS-awareness outlines a potential integration of QoS into the AAAC Architecture, where security ensures that mobile services are well-maintained, where pricing enables the commercial deployment, and where performance investigations will enable a future scalability of the implementation of AAAC intended.

2.1 Mobility

The development of wireless Local Area Network (LAN) technology and the deployment of wireless equipment with higher link bandwidth and capacity has made wireless access to Intranets and the Internet more popular. This popularity is strengthened by the trend in current developments of cellular communication technology toward an all IP based communication network. The need to be able to access services from any domain has placed different requirements concerning mobility, security, and charging. While mobility in cellular networks is an inherent characteristic, mobility in IP-based network has other problem areas.

Mobility can be seen as the capability to access services from different access points and to preserve this access, while moving from one access point to another access point. This capability should be provided by the service user as well as the service provider. There are three types of mobility, which in a sense show different degrees of mobility: terminal mobility, personal mobility, and service mobility. The terminal mobility enables mobile devices to receive continued access to services, independent of their location and while moving. The personal mobility enables a person to access services irrespective of his location and the terminal he is using. While terminal and personal mobility define the mobility of service users, the service mobility defines the capability of the network to provide a set of users the subscribed services irrespective of their current locations.

Each terminal and person is given a unique identity, which binds the terminal and person to a specific home organization. Personal mobility involving terminal mobility needs a mapping of respective identities for the purpose of routing and charging. Service mobility will require either terminal and/or personal mobility, however, the service provided will depend on the capabilities of the terminal and the visited network. Terminal mobility necessitates the support of handover, when changing access points and paging when placing in idle mode.

Mobile IP is a protocol that allows a network node (the Mobile Node, MN) to migrate from its home network to other networks, termed foreign network, either within the same administrative domain or to other administrative domains [7]. Mobile IP transforms the mobility problem into a routing problem. Features in IPv6 like address auto-configuration, neighbor discovery, and several routing options (destination options and home address options) ease the design and implementation of Mobile IPv6 (MIPv6) compared to Mobile IPv4 (MIPv4). In IP-based networks the problem of terminal mobility within or across administrative domain is solved by Mobile IP together with micro-mobility protocols dealing with handover and paging.

2.1.1 Requirements Imposed by Mobile IPv6 on AAAC

Requirements imposed by Mobile IP on AAA are discussed in detail in [7]. The following summary covers the most important points, which are enhanced at this stage to the AAAC view, including traditional AAA tasks and the charging task.

- Mobile Terminal (MT) and Home AAAC (AAAC-H) need to authenticate each other before access to services is permitted.
- The attendant's task is performed either by a DHCPv6 server (Dynamic Host Configuration Protocol) for statefull address auto-configuration or by IPv6 routers for stateless address auto-configuration.
- In the proposed standard Mobile IP specification a MT has to be configured with a home address, the address of an HA (Home Agent), and a SA (Security Association) with that of the HA. Using AAAC features would only require the MT to be configured with its NAI (Network Address Identifier) and a secure shared secret for use by the AAAC-H (AAAC Home Entity). The MT's home address, the address of its HA, the SA between the MT and the HA, and the identity (Domain Name System's name or IP address) of the AAAC-H can be dynamically determined as part of the Mobile IP initial registration.
- If MT is identified by a NAI and obtains dynamically an IP home address, the AAAC should be able to select a HA for use with the newly allocated home address. Mobile IP requires that the home address assigned to the MT belong to the same subnet as the HA providing services to the MT.
- If MT already knows the address of its HA, the AAAC-H must be able to coordinate the allocation of a home address with this HA designated by the MT.
- The AAAC server must be able to validate MT's certificates and to identify a MT's NAI, which is unique and of the form "user@realm"
- The AAAC server must be able to obtain or to coordinate the allocation of a suitable IP address for the MT.
- MT and AAAC-H have to share a SA.
- The attendant and its local AAAC server has to share a SA.
- AAAC-F (AAAC Foreign Entity) has to share or dynamically establish a SA with the AAAC-H. To provide for a scalable solution, an AAAC Broker Entity (AAAC-B) may be used, which has SAs with both AAAC-F and AAAC-H. The broker can act as a proxy between AAAC-F and AAAC-H or help AAAC-F and AAAC-B in establishing a SA by relaying a shared secret key to them, which will be used to set up the SA.
- AAAC server must be able to distribute keys for subsequent Mobile IP registration. The keys can be used to create a SA between the MT and a HA (if it not already exists) as well as the MT and the local attendant.
- The AAAC protocol should enable transport of Mobile IP registration messages as part of an initial registration sequence to be handled by AAAC servers. Any Mobile IP data transported via AAAC servers should be considered opaque to servers.
- After an successful authentication the MT should be allowed to use services and resources, at least for a minimal Mobile IP functionality.
- Local attendants should obtain authorization from a local AAAC server for QoS requirements placed by the MT.
- Either AAAC-H or AAAC-F can demand the attendant to terminate service to the MT.
- In cases where the MT accesses only local services, the AAAC-F may be able to locate a local HA in the current domain for use with MT.
- An AAAC server should be able to configure the firewall in the same domain to enable data traffic from the MT.

2.1.2 Requirements Imposed by a Handover

With respect to mobility in the wireless environment, handover and paging are highly relevant issues. There are two handover dimensions to be taken into account: technology and domain, both on link-layer as well as network layer. While a handover on the link-layer deals with technical aspects, a handover on upper layer considers business policy aspects. This distinction has a major impact on mobility management tasks and supporting them. Safely assuming an all-IP infrastructure, the technology dimension to be considered will lie in the link layer and the physical layer only. This results in the assumption that no inter-technology handover in the network layer will take place.

Whenever possible, a new authentication and authorization sequence should not involve the AAAC-H, if a MT moves from one point of attachment to another. This can be done as follows by having the MT supplying the NAI of the previous attendant:

- For a handover within the same administrative domain, the same AAAC-F should be able to provide the needed authentication without involving AAAC-H. The new attendant should be able to get the necessary information, *e.g.*, session keys from the AAAC-F or from the previous attendant.
- For a handover between foreign domains, the AAAC-F in the new domain may contact the AAAC-F in the previous domain to verify the authenticity of the MT and/or to obtain session keys.
- Accounting information of the old domain and the new domain will be kept and associated with the MT's identity currently authenticated.

2.2 QoS-awareness

The MobyDick architecture is targeted at the offering of QoS-enabled services. Within the project the Differentiated Services (Diffserv) architecture [3] is used to provide this QoS. The AAAC Architecture has to deal with this service provisioning architecture to control service access and to be able account and charge for the provided QoS at a later stage. This means that the AAAC System needs to interface the Diffserv architecture via a specific Application-specific Module (ASM). Here it is assumed that an inter-domain QoS setup is facilitated by a Bandwidth Broker (BB) architecture as described in [14] and [15].

The current Internet 2 QBone Bandwidth Broker discussion describes a two-tier model, where a Bandwidth Broker accepts Resource Allocation Requests (RAR) from users belonging to its domain or RARs are generated by upstream Bandwidth Brokers from adjacent domains. Each Bandwidth Broker will manage one service domain and subsequently provide an authorization based on a policy, which decides whether a request can be honored or not. A Resource Allocation Answer (RAA) confirms or rejects a request or it may indicate an "in progress" state. The RAR/RAA-based model implies that this is a distributed service, where the first authorization is pull-based and the other authorizations are either pull or agent-based [17].

In case of a pull-based authorizations only, the first BB, basically the service equipment which receives the RAR from the user, contacts the local AAAC System via an ASM. The authorization request encapsulates data from the RAR needed for authentication and authorization. If the BB receives a positive answer from the AAAC system, it forwards the RAR to the next downstream BB, where the procedure is repeated. Upon reception of a RAA, a bandwidth broker configures network elements in his domain for the service requested. This can be done via SNMP (Simple Network Management Protocol), COPS (Common Open Policy Service), or CLI (Command Line Interface). In case all authorizations besides the first are agent-based, the AAAC System forwards the RAR via the AAA protocol to the AAAC System of the next downstream BB, which performs authorization. This is repeated until the final domain is reached. If all authorization requests succeeded, an RAA is passed back by the AAA protocol to the first BB, which forwards the RAA to the user. At every AAAC System the RAA is passed to the BB via the ASM to enable a setup between network elements in his domain. In addition, either the service equipment (*e.g.*, the BB) or a separate system needs to be configured to collect accounting information for each domain. This can be done via the use of accounting policies [3]. In case of a pull-based authorization policies must be passed in the AAA response from the server to the service equipment, while for agent authorizations policies are passed with the service equipment configuration request.

Targeting at the goal to support both, user and terminal mobility, each user is virtually linked with a user specific profile. This profile is located from a functional point of view in the home network of the mobile user or at a position which is closely linked with the home area, *e.g.*, access to this profile is established via an AAAH entity. This profile is similar to the Home Location Register (HLR) of the 3GPP (3rd Generation Partnership Protocol) architecture, but is extended with additional user specific, QoS-related information. The AAAC Architecture is responsible to transport this information in close interaction with the mobility management from the AAAH to the AAAL server via the AAA protocol. The profile contains the QoS a user whishes to operate on, regardless of his point of connection. The QoS-specific part of the user profile could contain either a kind of service reference (*e.g.*, olympic services), if home and foreign provider have an agreement on the semantics, or a set of detailed service parameters, like minimal bandwidth, average bandwidth, maximum bandwidth, or maximum loss rate. To support different types of QoS for different applications, these specifications must provided on a per-port basis. The QoS profile is used at the AAAL to decide, whether the user can be authorized, and to configure the QoS provisioning service equipment via an ASM. The exact content of the QoS profile is left open at this stage for further study within the MobyDick project.

2.3 Security

An AAAC System provides for the necessary security support to deploy services to mobile users and equipment. In order to perform the required tasks concerning security, AAAC Systems will need support of security services and/or infrastructure.

2.3.1 Security Association

To accomplish the authentication task and to guarantee message integrity and privacy while exchanging sensitive information, Security Associations (SA) among AAAC and MIPv6 entities should be established for the necessary lifetime. Several different algorithms exist to provide this security requirement. Some algorithms rely on public key infrastructure, while others based on shared secrets (symmetric keys).

2.3.2 Replay Protection

Intervening nodes, including attendant and AAAC-F, during an authentication process must not be able to learn any secret information about mobile user's credentials, which will enable them to reconstruct and re-use credentials for future authentication and authorization processes. Therefore, in MIPv6 a MT has to share or dynamically establish an SA with the AAAC-H.

2.3.3 Non-repudiation

Event sequence tracking or reconstruction is important particularly in the areas of financial transactions, where transactions must be irrefutable. Systems with irrefutable transactions provide non-repudiation services. Non-repudiation services generate, maintain, and validate irrefutable evidence of events in every transactions. All authorization and accounting messages should be transferred and saved in a non-repudiative way. The ISO (International Organization for Standardization) non-repudiation model is related to events of sending or receiving a message.

Two types of non-repudiation services are distinguished: non-repudiation of origin (NRO) and non-repudiation of receipt (NRR). NRO gives the recipient the evidence that the sender has sent the specified message. NRR provides the sender the evidence that the recipient has received the specified message. Non-repudiation service comprises of four phases: evidence generation, evidence transfer and storage, evidence verification, and disputes resolution. There are two approaches in implementing non-repudiation services: with and without a Trusted Third Party (TTP). Without the help of a TTP, secrets are released gradually. In early efforts, a TTP acted as a delivery agent to provide non-repudiation of submission (NRS) and non-repudiation of delivery (NRD). Current non-repudiation protocols reduced the involvement of TTPs to deal with keys only rather than with the content of transferred messages. A non-repudiation protocol is fair, if it provides the originator and the recipient with valid irre-

futable evidence after completion of the protocol, without giving a party an advantage over the other party in any possible incomplete protocol run.

2.4 Pricing, Economy, and Deployment

Price differentiation is a concept which is widely deployed in various business sectors of our daily life. The introduction of such concepts lead to a profit maximization. An open and flexible AAAC architecture should support such mechanisms as well. Starting point from a technical point of view is the metering process. Based on this, accounting and auditing mechanisms are to be deployed. To be able to present feedbacks to customers, metering and charging becomes time-critical, since a user must be informed on the price of a resource he is about to consume.

To deploy an AAAC architecture, firstly a service concept is required. Figure 2 depicts a possible service provisioning concept from an economical point of view. A subscriber, *e.g.*, the medium size enterprise, agrees on a Service Level Agreement (SLA), which is valid for all members of the company (all users). A single user utilizes resources or services of a service provider, which has to be contracted by a network provider for delivering the service to the user. The network provider maintains a business relationship to the subscriber and charges the subscriber for both, his and the service provider's services delivered to the user. In order to implement such a model, many issues are to be considered. Each user receives a personal identifier, which could be used to receive any kind of services, independent of its location and terminal currently used according to the SLA agreed by its subscriber.

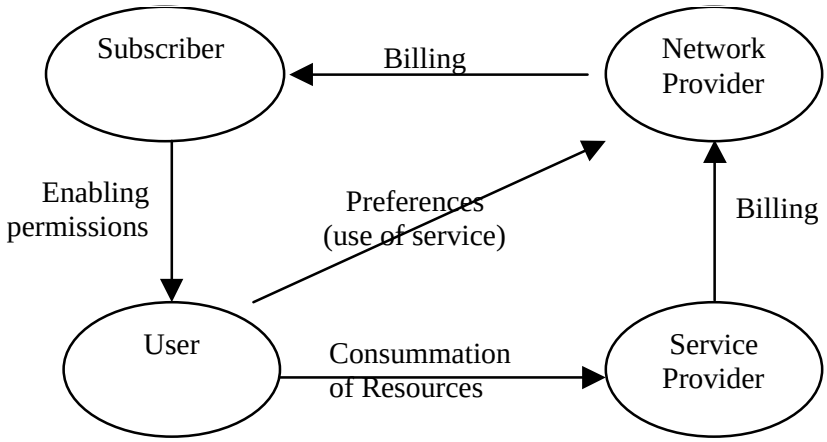

Figure 2: Service Concept of Next Generation Mobile Networks

2.5 Performance

Mobile IP requires every registration to be handled between an attendant and the HA. This registration can be performed after the authentication, but to reduce the time needed for an initial mobile IP registration, the registration message should be piggy-backed during the AAAC authentication in case HA and AAAC-H are in the same administrative domain. AAAC-F and AAAC-H need to interface the attendant and the HA to handle this registration message.

3 Architecture

The AAAC Architecture considered within the MobyDick project deals with the handling of information required to ensure that a mobile node, mainly a mobile host, is correctly granted access to networking resources in an Internet domain, it normally does not belong to. In addition, it deals with those data that are collected to provide charging for those service used by the mobile node. Within the project the auditing functionality is considered inherently. To allow for a complete and structured approach in preparing a generic AAAC Architecture, the underlying network technologies, which are considered relevant, are introduced at the beginning. Based on the overall valid assumptions undertaken detailed requirements are listed, which form the basis for the description of modules and components as well as their location within a functionality and infrastructure view. Requirements on AAAC System are derived from these investigations, which defined the basis for the specification of its modules, interfaces, protocols, and components

3.1 Basic Considerations

To enable a clear and common understanding on reasoning in various cases for or against certain solution proposals, a set of unique assumptions for the AAAC Architecture is defined. These assumptions are valid with respect to technology, protocols, and scenarios. The work on the AAAC Architecture requires the definition of the underlying technology to be considered with respect to their technical characteristics. On one hand, this includes protocol interfaces, which allow for the collection and access of data available within these protocols. On the other hand, it delimits the current areas of important technologies to be considered inherently with the AAAC Architecture approach undertaken.

Next to the underlying technology, the business model to be deployed has an impact on the AAAC architecture. Here, first the service concept has to be mentioned, but also charging strategies like pre-paid charging, which gained a lot of subscribers in the GSM market, has different requirements to the AAAC architecture than traditional postpaid charging concepts. Especially the prepaid charging concept rises up timely critical policing requirements which could be both, user-centric or subscriber-centric. So performance and scalability issues play an important role on an open and scalable AAAC architecture supporting various service provisioning concepts.

Basically, the AAAC Architecture can be regarded from two points of view: the user and the provider perspective. Without discussing it in any detail or explicitly the user perspective is provided by his QoS and mobility requirements (cf. work package 2 and 3 referred to in [5]), however the user view's requirements are at some stages of interest, but the complexity of allowing for access and mobility will basically remain similar for the AAAC Architecture, independent on those assumptions discussed in the following. The perspective of importance is the provider perspective, which is considered within this document, since the complexity of the AAAC Architecture as well as of interaction mechanisms to be developed will vary depending on those assumptions.

In addition, both wired and wireless technologies will be considered for a AAAC Architecture. Therefore, they should support at least user and device mobility, and may be in the future application or server mobility, which will become an important issue in a all-mobile networking scenario.

3.2 AAAC Architecture

The AAAC Architecture has been derived from the generic architecture proposed by the AAAArch group [12]. It consists of AAAC Systems which can be either an AAAC server or an AAAC client. The protocol to be operated between the AAAC server and the AAAC client is termed AAA protocol, however, for a sufficient transfer of appropriate data it may be an enhanced version of either RADIUS or DIAMETER. An AAAC client has no services to offer, however, instead it can request services only, if using the agent authorization model [17]. An AAAC server operates an interface to several Application-specific Modules (ASM), which provide either a service (*e.g.*, Interface to Mobile IP, QoS, content service) or accounting or charging functionality, which is viewed as an important service on its own. The AAAC server also has an interface to external authentication modules to be able to use different authentication techniques. The accounting component may get usage data input from a underlying metering component. Accounting can be a separate service or integrated within the service provided [4]. In case of integrated accounting, the service and accounting is performed by a single component, which interfaces the AAAC server via one ASM. Charging is implemented as an external module which also communicates via an ASM with the AAAC server. The charging module generates input for a subsequent billing process. Figure 3 shows the AAAC architecture.

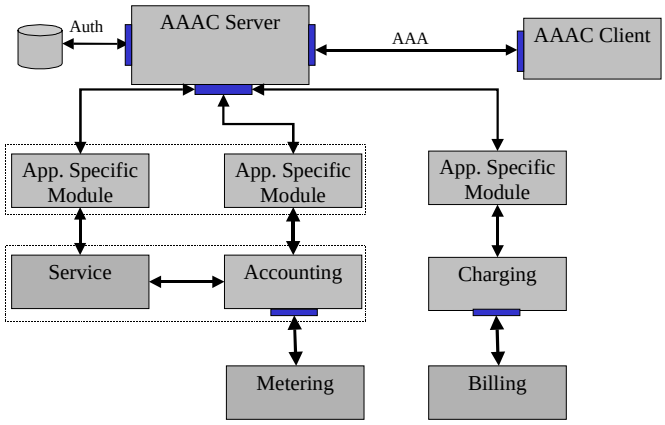

Figure 3: AAAC Architecture

AAAC servers or clients communicate via a standardized inter-domain AAA protocol. An AAA client only needs to use a subset of the AAA protocol messages. The communication between ASM and AAA server is intra-domain. An ASM also can act as an AAA client if the pull authorization model is used [14]. However the ASM also has to control the service provision and to deliver accounting data. The authentication interface also is an intra-domain interface.

3.3 AAAC System

Figure 4 shows the architecture of the AAAC System. The AAAC server has three external interfaces to authentication via an authentication interface, to account and charge via the ASM interface and an AAA protocol interface for inter-domain communication with other AAAC servers and clients.

The central component of the AAAC server itself is the control module. The control module processes AAAC transactions. Transactions originate either from another AAAC server or client via the AAA protocol or from an ASM. AAA protocol messages are processed by the AAA protocol handler while messages from ASMs are processed by the ASM interface. When the control module receives an AAA message it is processed according to internal policy rules and the state of the particular session.

These rules may enforce the control module to contact the authentication module, another AAAC System or an ASM for fulfilling the request. Upon arriving of a new request the AAAC server creates an initial session record which holds the state of the overall session as well as the state of short lived protocol transactions. A service being authorized and accounted for may be offered over a period of time and may consist of different sessions and transactions. All relevant information relative to the session are maintained by the session management component and are stored in the session data repository. After a session is terminated the session record is freed. During the runtime of the session all data relevant for later auditing are collected by the auditing component and are stored in the auditing database. The policy repository contain policy rules which define how the authentication, authorization, accounting and charging process work. These policies are evaluated by the rule based engine which is part of the control module. The parameters for these policies - if not locally known - need to be retrieved via one of the external interfaces. The session management, auditing and policy retrieval component use protocols such as LDAP or ODBC. to retrieve their data from the particular repositories. Security mechanism which are needed for the inter-domain AAA protocol are handled by a security module. These include the following capabilities:

- Authentication of peer AAA servers,
- Integrity protection for data elements exchanged between AAA peers, and
- Confidentiality protection for data elements exchanged between AAA peers.

When processing AAA transactions, the generic AAA server has no knowledge of the specific service the user is requesting. Therefore, it will have no hard-coded knowledge of how to process AAA transactions. For instance, if a request for service is received from a user it must be decided which AAAC servers in what administrative domains must be consulted for authorization and where policies reside which must be applied to this request. When accounting messages are received for a session, it must be decided where they need to be forwarded to. These decisions are encoded in a set of policies that may be accessed by the server from the PR In general, there will be one policy per service per each message type that can be received over any of the communication interfaces. These policies are also evaluated at the rule based engine.

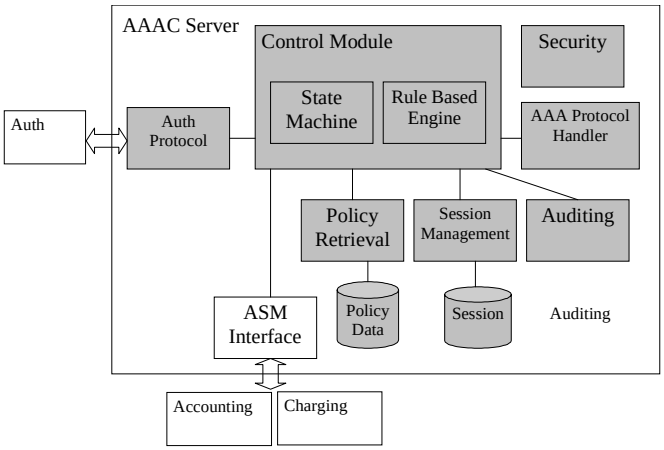

Figure 4: AAAC System Architecture

ASMs are required to configure and provision a service into the service equipment. ASMs may translate high level service policies to the low level device-specific policies required by the service equipment. ASMs may also evaluate “service proprietary” policies. These are policies which require application-specific knowledge in order to evaluate the result. For each type of service offered a separate ASM is needed.

3.4 AAAC Module Interfaces

The set of AAAC module interfaces include the following ones, which currently are being specified in more detail.

- AAA protocol: Enables the communication between AAAC servers in different domains.
- Authentication protocol: Defines the message types and exchange sequences for the authorization process within AAAC.
- ASM interface: Defines the interactions and data to be exchanged between an AAAC server and the underlying entity.
- Repository interfaces: Enable a generic access procedure for various data stored in AAAC-local data bases.
- Metering/accounting interface: Determines the data types and sequences of transfer for metering and accounting data.
- Charging/billing interface: Determines the procedures to complete the full charging and billing cycle for usage of services.

4 Use Cases

To enable the design and implementation of the AAAC Architecture, the logical view needs to be mapped onto the physical view. Obviously, there exist a number of alternative choices, while each of them shows a certain list of advantages and drawbacks. Any of these mappings is termed a scenario. However, the particular exploitation of a subset of modules, i.e. logical functionalities, may be performed differently. Therefore, the introduced “Use Cases” below determine a dedicated use of modules, a specified mapping of modules onto components, and the solution to a specific application problem, where certain service levels shall be achieved or special policies shall be applied.

As shown in Table 1, the following use cases are of interest to MobyDick and they form a sample of realistic situations. Two major groups of use cases are envisioned. Stationary or semi stationary hosts, e.g. dial-in access with no movement of the user, another example is Wireless LAN access with very few movement of the mobile node. So, within this group there will be few inter and many intra-domain handovers. Intra-technology handovers within a domain can occur as well, but inter technology handovers within a single domain are not considered. The second group are mobile nodes with a high level of mobility, *e.g.*, UMTS (Universal Mobile Telecommunication System) access in a train travelling between two cities. This example would result in many inter domain or inter technology handovers, this may include many intra-domain or intra-technology handovers as well. Within each major group, many subgroups are possible. As an example, imagine a company that wants to grant access to their private network for their own employees. Typically employees work in their office (semi-stationary, group 1), to work convenient, they will need end-to-end QoS, they would not need auditing, but secure data transport might be appropriate.

Table 1: MobyDick Use Cases

No.	Customer/User Activity	End-to-end QoS	Auditing	Secure Data Transport
1	**(Semi-) stationary host**	yes	yes	yes
		no[a]	no	no
2	**Mobile host**	yes	yes	yes
		no[a]	no	no

a. Best effort service class in diffServ environment.

Figure 5 shows an example of use cases. On the right, a mobile host is being connected inside domain B. The mobile host is connected inside domain A (lower left). Data transport through the network is shown in green color. In this example the Internet services use some sort of an end-to-end QoS guarantee, statistical or deterministic. The AAAC Architecture will be able to interact closely with the module of the QoS Manager/QoS Broker, which will be defined in close collaboration of WP2 QoS. A.o. AAAC may allow for the maintenance of QoS-related information, which are required anyway to perform AAAC tasks.

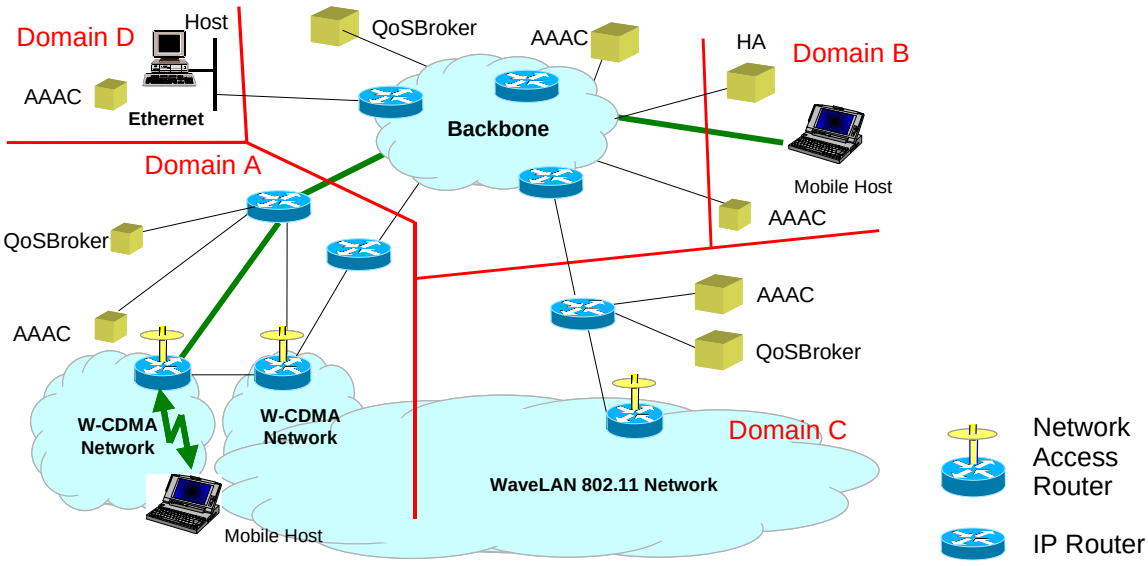

Figure 5: Use Case Example

Whereas the use case for the dial up is quite easy, the mobile host is the most complex one, since it will integrate QoS and mobility issues into a future-oriented case for policy-based end-to-end QoS provisioned Internet services.

5 Summary and Conclusions

This paper presented an overview on the MobyDick AAAC architecture comprising the complex inter-operation across organizational bodies of local operators and backbone service providers, charging aspects of different access network technologies, business models, and security considerations on the access and utilization of mobile networking services. Planned features of the MobyDick AAAC architecture are tariff announcements, online charge indication, and inter-domain support for both, fixed and mobile users, where multiple providers competitively offer QoS-enhanced IP services, and where end-users dynamically select providers based on QoS availability and tariffs.

Based on the support for configuration potentials for service-specific and user-specific features by allowing for policy-based functions, the application scenario used as a motivation is applied to the MobyDick architecture. The IETF's and IRTF's AAA architecture [1], [12] served as the basis of the MobyDick AAAC approach, however, based on the dedicated services consid-

ered and designed, a major enlargement and detailing of this basis was necessary, since within the IETF and IRTF not all aspects have been covered sufficiently. These extensions and their major requirements have been discussed, to be integrated into the overall MobyDick architecture. Furthermore, the paper investigated mobility scenarios and requirements for authentication and authorization with respect to Mobile IPv6.

Finally, the outlook comprises work to be refined which is based on the defined service concept of the net generation mobile networks. The inter-relation between accounted for information and charging based on the services provided including mobile services and users (such as SLA definitions, parameter identification, mapping definition, and pricing model design) are considered closely at this stage. This covers the exact content of the QoS profile, which describes the contract between providers and customers.

Acknowledgements

The authors like to extend many thanks to P. Kurtansky, T.V. Prabhakar, D. Singh as well as to their project partners within work package WP4 of the MobyDick project, namely FTW, Vienna; NEC CCRL, Heidelberg; T-Nova Berkom, Berlin; and University of Madrid, UC3M.

References

[1] *AAA Architecture Research Group*; Internet Research Task Force, Chairs: J. Vollbrecht, C. de Laat, Information available at the URL http://www.phys.uu.nl/~wwwfi/aaaarch/, April 2001.

[2] B. Aboba, P. Calhoun, S. Glass, T. Hiller, P. McCann, H. Shiino, P. Walsh, C. Perkins, B. Patil, D. Mitton, S. Manning, M. Beadles, X. Chen, S. Sivalingham, A. Hamed, M. Muson, S. Jacobs, B. Lim, B. Hirschman, R. Hsu, H. Koo, M. Lipford, E. Campbell, Y. Xu, S. Baba, E. Jaques: *Criteria for Evaluating AAA Protocols for Network Access;* Internet Engineering Task Force, RFC 2989, November 2000.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Whang, W. Weiss: *An Architecture for Differentiated Services;* Internet Engineering Task Force, RFC 2475, Informational, December 1998

[4] G. Carle, S. Zander, T. Zseby: *Policy-based Accounting;* Internet Draft Informational, work in progress, March 2001.

[5] H. Einsiedler (Edt.): *MobyDick Technical Annex;* Version 2, Mach 29, 2001.

[6] S. Farrell, J. Vollbrecht, P. Calhoun, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence: *AAA Authorization Requirements;* Internet Engineering Task Force, RFC 2906, August 2000.

[7] S. Glass, T. Hiller, S. Jacobs, C. Perkins: *Mobile IP Authentication, Authorization, and Accounting Requirements;* Internet Engineering Task Force, RFC 2977, October 2000.

[8] IRTF, Internet Research Task Force: *AAA Architecture Research Group*; Information available at the URL http://www.phys.uu.nl/~wwwfi/aaaarch/, April 2001.

[9] J. Jähnert (Edt.): *MobyDick Terminology;* MobyDick Deliverable, April 2001.

[10] M. Khalil, H. Akhtar, K. Pillai, E. Qaddoura: *AAA Interface for IPv6 Handoff;* Internet Draft Standard, October 2000.

[11] C. de Laat: *Structure of a Generic AAA Server;* Internet Draft, draft-irtf-aaaarch-generic-struct-00.txt, February 2001.

[12] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence: *Generic AAA Architecture;* Internet Engineering Task Force, Experimental RFC 2903, August 2000.

[13] B. Moore, E. Ellesson, J. Strassner, A. Westerinen: *Policy Core Information Model - Version 1 Specification*; Internet Engineering Task Force, RFC 3060, Proposed Standard, February 2001.

[14] R. Neilson, J. Wheeler, F. Reichmeyer, S. Hares: *A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment;* Version 0.7, August 1999.

[15] B. Teitelbaum, P. Chimento: *Qbone Bandwidth Broker Architecture;* June 2000.

[16] D. Verma: *Supporting Service Level Agreements on IP Networks;* Macmillan Technology Series, 1999.

[17] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence: *AAA Authorization Framework;* Internet Engineering Task Force, RFC 2904, August 2000.

[18] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence: *AAA Authorization Application Examples;* Internet Engineering Task Force, RFC 2905, August 2000.

[19] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A.-N. Huynh, M. Carlson, J. Perry, S. Waldbusser: *Terminology;* Internet Draft, draft-ietf-policy-terminology-03.txt, work in progress, April 2001.

[20] R. Yavatkar, D. Pendarakis, R. Guerin: *A Framework for Policy-based Admission Control;* Internet Engineering Task Force, RFC 2753, Informational, January 2000.