University of Twente
Enschede - The Netherlands

Thesis for Master Telematics,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Design and Analysis of Communication Systems (DACS),
University of Twente

Providing AAA with the Diameter protocol for multi-domain interacting services

By Wendy Ooms
June 15, 2007

Supervising committee:
Dr. Ir. Georgios Karagiannis (University of Twente),
Dr. Ir. Aiko Pras (University of Twente),
Dr. Ir. Oskar van Deventer (TNO),
Drs. Jurjen Veldhuizen (TNO).

Abstract

In this thesis, we present a solution for the problem of how to provide authentication, authorization and accounting (AAA) for multi-domain interacting services. We studied the case of 'FoneFreez', a service that provides interaction between different basic services, like telephony and television. Because several parties are involved in this value chain, e.g. television provider, telephony provider etc., secure interaction between multiple domains must be assured. A part of this security issue can be resolved using AAA. In this study the AAA protocol Diameter is used for that purpose, which is the successor of the RADIUS protocol. How the Diameter protocol can be used for AAA in multi-domain service interaction is subject of this study.

Diameter is a very flexible protocol. The adoption by 3GPP boosted the number of network products that implement Diameter. Diameter is mainly used for end-user authentication, authorization and accounting, and is specifically designed for roaming situations.

In this study, the solution has to fulfill requirements that are elicited from the FoneFreez case. The requirements mainly cover AAA for the end-user and AAA between the different parties.

Two different specifications are found that use Diameter to provide AAA for multi-domain service interaction. It depends on the application which one is best suited. The first solution specifies that the different parties are loosely connected. This solution can be implemented in stages. The second solution specifies a tight coupling between the parties. The second solution is best suited for situations where the service is intensively used. In the end it is verified that both specification fulfill all requirements.

The conclusion of this thesis is therefore that the Diameter protocol can be reused in its existing form, to provide AAA for multi-domain interacting services. We found that authentication between different parties is better done with the Kerberos protocol. It is recommended that further research is done into the exchange of identities for our problem. Furthermore the provisioning of quality of service by Diameter for multi-domain interacting services should be explored.

Preface

What is in front of you is my thesis for the Master Telematics. The thesis is the last part of the Master program to be concluded before receiving the Masters degree.

I conducted my thesis at TNO-ICT in Delft under the supervision of Oskar van Deventer and Jurjen Veldhuizen. They offered me the opportunity to learn from the work they performed and acquire practical experience in the field of Telematics. During my thesis about AAA, I learned a lot of the Diameter protocol. With this report I would like to pass on the knowledge I gained about Diameter.

Georgios Karagiannis and Aiko Pras supervised me on behalf of the DACS chair at the University of Twente. I would like to thank my all four of my supervisors for guiding me in the right direction at times of confusion.

My thanks go to TNO and my colleagues for giving me the opportunity to acquire practical experience and for their assistance during my thesis. I would like to use this opportunity to thank my parents for enabling my education, and their support in the process. For help during my thesis, my special thanks go to: Hans Stokking, Fabian Walraven, Mike Schenk, Frank Fransen, George Huitema, Henk Ensing, Frens Rumph, Johan Boekema, Klaas Wierenga and Elly Ooms.

Wendy Ooms,
June 15, 2007

Contents

Abstract iii

Preface v

Contents vii

List of figures ix

List of tables xi

List of abbreviations xii

1 Introduction 1

1.1 Problem description 2
1.2 Relevance of the problem 3
1.3 Reader's guide 5

2 Methodology 7

2.1 Waterfall model 7
2.2 Scenario based requirements elicitation 7
2.3 Case study 8
2.4 Literature 9
2.5 Identification of phases 9
2.6 Design methodology 10
2.7 Verification of requirements 11
2.8 Conclusion 12

3 Theoretical background 13

3.1 AAA 13
3.2 RADIUS 18
3.3 Introduction to Diameter 18
3.4 Interacting services 21
3.5 Realms 22
3.6 IMS 23
3.7 Conclusion 25

4 Case description 27

4.1 Architecture 27
4.2 Interaction 29
4.3 Business roles model 30
4.4 Implementation of business roles 32
4.5 Conclusion 33

5 Requirements 34

5.1 Scenario 34
5.2 Actors and use cases 35
5.3 Functional requirements 36
5.4 Non-functional requirements 37
5.5 Constraints 38
5.6 Acceptance criteria 38
5.7 Conclusion 38

6 Solution phases 40

6.1 Initialization phase 40
6.2 Log-on phase 45
6.3 Operational phase 49
6.4 Conclusion 50

7 Alternative solutions 51

7.1 Two designs 51
7.2 Interactions belonging to the different designs 53
7.3 Comparison of specifications 57
7.4 Conclusion 60

8 Fulfillment of the requirements 62

8.1 Functional requirements 62
8.2 Non-functional requirements 65
8.3 Constraints 65
8.4 Conclusion 66

9 Conclusion and future work 67

9.1 Results 67
9.2 Discussion 68
9.3 Conclusion 69
9.4 Future work 70

References 72

Normative references 72
Informative references 77

Appendix A Diameter 81

A.1 History 81
A.2 Diameter framework 82
A.3 Diameter base protocol 85
A.4 Comparison with other protocols 90
A.5 Diameter applications 95

Appendix B Diameter commands 114

Appendix C Identity management 115

Appendix D Path to solution 119

D.1 One realm situation 119
D.2 Services in different realms 120
D.3 Application server in separate realm 122
D.4 All business roles in different domains 124

Appendix E Interaction diagrams 126

Appendix F Future for Diameter 129

F.1 Standard 129
F.2 Supporting firms 130
F.3 Industry forces 131
F.4 Market mechanism & environment 132
F.5 Conclusion 132

List of figures

Figure 1 Waterfall model 7
Figure 2 Path to solution 11
Figure 3 Three-party authentication model [Nakhjiri et al, 2005] 14
Figure 4 Agent sequence [RFC 2904] 15
Figure 5 Pull sequence [RFC 2904] 16
Figure 6 Push sequence [RFC 2904] 16
Figure 7 Accounting overview [ICOM, 2006] 17
Figure 8 Diameter framework 19
Figure 9 Simplified IMS reference architecture [Fried et al, 2006] 24
Figure 10 IMS inter-domain architecture [Sher et al, 2006] 25
Figure 11 Architecture FoneFreez 28
Figure 12 Sequence diagram case 30
Figure 13 Business roles model 31
Figure 14 Implementation of business roles 32
Figure 15 Actors and use cases 36
Figure 16 Realm hierarchy [Microsoft, 2006] 41
Figure 17 Eduroam Certificate authority [Eertink et al, 2005a] 43
Figure 18 Multi-domain authorization I [RFC 2904] 47
Figure 19 Multi-domain authorization II [self edited] 47
Figure 20 Revenue sharing 48
Figure 21 Cash flows 49
Figure 22 Hop-by-hop design 51
Figure 23 End-to-end design 52
Figure 24 Certificate authorities 53
Figure 25 Hop-by-hop authentication 55
Figure 26 End-to-end authentication 56
Figure 27 Diameter framework 82
Figure 28 Relay agent 83
Figure 29 Proxy agent 84
Figure 30 Redirect agent 84
Figure 31 Translation agent 85
Figure 32 Diameter protocol header [RFC 3588] 85
Figure 33 AVP header [RFC 3588] 87
Figure 34 Diameter Mobile IPv4 interaction 96
Figure 35 Mobile Security Associations [RFC 4004] 97
Figure 36 Interaction NASREQ 99
Figure 37 First interrogation after authentication and authorization [RFC 4006] 101
Figure 38 Authorization messages used for first interrogation [RFC 4006] 102
Figure 39 One-time event [RFC 4006] 103
Figure 40 EAP authentication [RFC 4072] 104
Figure 41 EAP authentication alternative [RFC 4072] 104
Figure 42 Diameter SIP application architecture [RFC 4740] 105
Figure 43 Authentication and authorization procedure [RFC 4740] 106
Figure 44 Locating the SIP server of the recipient [RFC 4740] 107
Figure 45 User profile update [RFC 4740] 107
Figure 46 QoS request authorization [Alfano, 2006] 109
Figure 47 Server-side initiated QoS parameter provisioning [Alfano, 2006] 110
Figure 48 DHCPv6 architecture 111
Figure 49 DHCP request [Vishnu, 2006] 111
Figure 50 Server-side push configuration [Vishnu, 2006] 112
Figure 51 Liberty architecture [Liberty, 2005] 115
Figure 52 Two identity providers federated to a service provider [Liberty, 2005] 116
Figure 53 Two federated identity providers [Liberty, 2005] 116

Figure 54 Alternative I 117
Figure 55 Alternative II 118
Figure 56 User authentication 118
Figure 57 AAA in one domain 119
Figure 58 Services in different domains, connected AAA..... 121
Figure 59 Services in different domains, separate AAA..... 121
Figure 60 Application server in different domain, connected AAA 123
Figure 61 Application server in different domain, separate AAA.....123
Figure 62 All business roles in different domains, connected AAA 124
Figure 63 All business roles in different domains, separate AAA 125
Figure 64 Registration phase end-to-end design 126
Figure 65 Registration phase hop-by-hop design 127
Figure 66 Operational phase.....128
Figure 67 Success factors telecom standards [Sweers et al, 2007].....129

List of tables

Table 1 Comparison of designs 57
Table 2 Fulfillment of the requirements 62
Table 3 Diameter command codes defined in the base protocol [IBM, 2006] 86
Table 4 3GPP interfaces [IANA, 2006] 112
Table 5 Overview Diameter commands 114

List of abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ATM	Automated Teller Machine
AVP	Attribute Value Pair
BGCF	Breakout Gateway Control Function
CA	Certificate Authority
CHAP	Challenge-Handshake Authentication Protocol
CSCF	Call Session Control Function
DACS	Design and Analysis of Communication Systems
EAP	Extensible Authentication Protocol
ETSI	European Telecommunications Standards Institute
HSS	Home Subscriber Service
HTTP	HyperText Transfer Protocol
I-CSCF	Interrogating-Call Session Control Function
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPTV	IP television
ITU	International Telecommunication Union
MGW	Media Gateway
NAI	Network Address Identifier
NAS	Network Access Server
NASREQ	Network Access Server Requirements
NGN	Next Generation Networking
P-CSCF	Proxy-Call Session Control Function
PDU	Protocol Data Unit
PIN	Personal Identification Number
PKI	Public Key Infrastructure
QoS	Quality of Service
RADIUS	Remote Authentication Dial In User Service
RFC	Request For Comments
SAML	Security Assertion Markup Language
SCIM	Service Capability Interaction Manager
S-CSCF	Servicing-Call Session Control Function
SCTP	Stream Control Transmission Protocol
SDL	Specification and Description Language
SEG	Security Gateway
SIM	Subscriber Identification Module
SIP	Session Initiation Protocol
SOAP	Simple Object Access Protocol
TACACS	Terminal Access Controller Access-Control System
TCP	Transmission Control Protocol
TISPAN	Telecoms & Internet converged Services & Protocols for Advanced Networks
TLS	Transport Layer Security
TNO-ICT	Netherlands Organization for Applied Scientific Research - Information and Communication Technology
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
VoIP	Voice over IP
WLAN	Wireless Local Area Network

1 Introduction

Basic services like telephony and television have been developed a long time ago, at the first half of the previous century. The services became a commodity and their supply chains matured, which are the so called 'stovepipes'. For example the companies that supplied telephony had nothing to do with television and the supply chain from beginning till the end was arranged to deliver solely telephony.

Next to the basic services like telephony and television, value added services are developed. Value added services are services that use the basic services and try to add extra value for the customer, e.g. Electronic Program Guide (EPG) or telephone conferencing services. There is also a group within the value added services that provide added value based on service interaction. This is the interaction between basic services e.g. telephony and television.

The trend can be seen of leaving the stovepipes in telecommunication and adopting the horizontally internet model. The traditional model of one operator playing both the role of service provider and network operator is outdated. Due to this trend, new models need to be found that disengage the different roles. When multiple providers are part of the supply chain, interaction between these providers is necessary. Because every provider resides in its own domain, the services cross multiple domains. When enabling service interaction over multiple domains several issues arise.

Security of multi-domain service interaction is one of these issues. How does the interaction service know if a user is allowed to use the service? And how can the providers that reside in different domains trust each other? For example a user has a different telephony and television provider, and some service is in place to enable interaction between these two basic services. How can the user be billed for the different services? Or how does the television service know that the interaction service is allowed to intervene? For this reason authentication, authorization and accounting (AAA) is needed for multi-domain service interaction.

AAA can be provided by AAA protocols and an example of such protocol is Diameter [RFC 3588]. Diameter is the newest AAA protocol developed in 2001 from the older AAA protocol RADIUS.

At the Netherlands Organization for Applied Scientific Research - Information and Communication Technology (TNO ICT) research is done in the field of multi-domain service interaction. How can this be enabled, and how can we offer a secure solution? The next section presents the problem description and the thesis delimitation.

1.1 Problem description

TNO ICT encounters the Diameter protocol in different fields of research, for example in their research on the IP Multimedia Subsystem (IMS). To be able to have full understanding of the IMS architecture, knowledge about every component is required. Diameter was such a component for which more insight was needed on the functionality and application of the protocol, to enable a good overview of the complete architecture.

Within their field of work, TNO ICT acknowledges the need for advice about interacting services. A demo, called FoneFreez, was built to investigate how interacting services from different realms can be managed and how different platforms can be interconnected. The service that is built is an IP television service that interrupts when the phone (IP based) is ringing. The AAA support for this demo was outside the scope of that project.

The main objective of this assignment is to study the functionality and application of the Diameter protocol specification and to reuse and/or extend this specification to provide AAA for interacting services from different realms.

To achieve this objective a specification has to be made, on how to integrate AAA support in the project mentioned above.

The goals of this assignment are: to draw up requirements, make a specification and evaluate whether the specification fulfills these requirements.

The main research question that has to be answered by this assignment is:

Is it possible to reuse and/or extend the Diameter protocol specification, according to the rules defined in the Diameter base protocol, to provide AAA for interacting services from different realms in IMS like architectures?

The research sub-questions are:

- Which requirements should be applied?
- What does the case of the project look like and which IMS components are used?
- What are the currently available Diameter-based architectures for interacting services from different realms?
- How can the AAA architectures be mapped on the given case?
- Which specification can be derived?
- Does the specification fulfill the requirements of this study?

The terms used in this research question are further explained in chapter 3.

1.2 Relevance of the problem

It is important to perform research on how to add AAA as stated in the main research question, to enable commercial exploitation of service interaction from different realm. This section is divided in three subsections. First it answers why service interaction, as described above, is relevant. Furthermore, it describes why it is important to add AAA support to the interacting services. Finally it is described why the solutions that can be found in literature are not sufficient.

1.2.1 Service interaction at the service provider level

Service interaction can take place at different levels, at the service provider or at the residential gateway of the consumer. In this thesis the interaction at the service provider level will be discussed. The party providing the service interaction is defined as the broker.

The triple play functionality that several providers offer is the immediate cause to look for solutions that provide interaction at the service provider level. The television, telephony and internet services used to be offered by separate parties. With triple play, the services are offered by the same provider, which can add value on top of the basic services. The service providers need the value added services to distinguish themselves from the other triple play providers. Interaction services can be such value added services.

The reason to enable service interaction at the service provider level is that in this way the service provider can control the service interaction. When the service interaction takes place at the residential gateway, the service provider has no

influence on what is happening with the service the client uses from this provider. The service interaction embedded in the network gives control to the service provider and helps to maintain customer relations.

Furthermore, research in interacting services is important to enable different applications to interact, without the need for building them again. The basic services like television and telephony can be reused by interaction services. An example of such an interaction service is the FoneFreez service built at TNO-ICT.

The service interaction in the TNO project 'FoneFreez' is done with a Service Capability Interaction Manager (SCIM). The SCIM is an IMS function, but is not completely standardized by 3GPP yet. In the FoneFreez project the SCIM is used to provide interaction between different service providers (IPTV, IP telephony). More about interacting services can be found in section 3.4.

1.2.2 AAA for security

At this moment the FoneFreez project has no possibility to provide AAA. The AAA concept is explained in section 3.1.

The user cannot be authenticated by the IP television service or to the IP telephony service. Also the services are not authenticated or authorized by the broker, which is needed since these services belong to different administrative domains and are managed by different service providers.

Authorization is needed to sustain intervention of the broker at the service providers. It is necessary to check if the request to intervene indeed comes from the correct broker and if the request is allowed for that user.

At the moment, accounting for the service interaction can only be done on a flat fee basis. To enable usage based charging, accounting functionality must be added to the design developed in the FoneFreez project. This consists of two components: accounting for the IPTV and IP telephony platforms and accounting for the service of the interaction service. The broker must be able to charge the interaction service for the usage of the service interaction. The support of accounting for the IPTV and IP telephony platforms is outside the scope of this assignment.

1.2.3 AAA solutions in literature are not sufficient

In literature few is written about AAA architectures. Most of the work is done in a subgroup of the AAA workgroup of the IETF [AAAARCH, 2004]. The subgroup closed in October 2004, after delivery of several informational RFCs about AAA architectures. Some work can be found in this document about multi-domain AAA. This is further described in chapter 3.

Some products that provide service interaction are on the market, where the AAA aspect is not pointed out explicitly. Also service interaction in a multi-domain environment is not described in literature [Aepona, 2007]. Because these products are vendor specific no general solution for adding AAA to interacting services from different realms exists.

Research has not been done for the Diameter protocol on how to provide AAA for service interaction from different realms. In this thesis a generic solution is developed for AAA for service interaction from different realms, based on the standardized protocol Diameter.

1.3 Reader's guide

The structure of this report follows from the methodology that is used for the thesis. The two most important methods used are the 'case study methodology' and the creation of a new design according to the waterfall model. The methodology is discussed at length in chapter 2.

The 'case study methodology' is used, because the case of FoneFreez is relevant for this thesis, and the main research question should be studied in this context [Perry et al, 2004]. This means that the requirements on the specification are derived from the case study and the specification must fit within the boundaries set by the case. This is done using 'scenario based requirements elicitation' [Whittle et al, 2004].

Regarding the waterfall model, this thesis describes only the first two phases; due to lack of time only the requirements phase and the design phase are considered. The implementation, verification and test phase are out of scope of this study. First the requirements are drawn up, and then the specification is made.

Furthermore chapter 3 gives some theoretical background about AAA and other terms used in the problem description. Chapter 4 describes the case study; the

FoneFreez project at TNO. This is followed by the requirements; the constraints on the specification that are derived from the case in chapter 5. The solution is presented in the chapter 6 and chapter 7. In chapter 8 is discussed whether the specification fulfills the set of requirements/constraints, followed by the conclusion of this thesis in chapter 9.

In the appendices background information is provided. Appendix A discusses the Diameter protocol in more depth. Appendix B gives an overview of the Diameter command codes. In Appendix C the identity management issue of this thesis is studied. The path to the solution is discussed in Appendix D. Appendix E provides the interaction diagrams of the solutions. In Appendix F a TNO-ICT model is used to predict the success of the Diameter protocol.

2 Methodology

This chapter discusses the methodology used for answering the main research question and the research sub-questions. First the method for the main research question is discussed, after which the methods for the research sub-questions are described.

2.1 Waterfall model

The objective of this study is to design a specification for the given problem. When creating a new design standard approaches are known. One of these approaches is described in the waterfall model. The first person to come up with the waterfall model was Royce in 1970. After that several interpretations of the model have arisen. The waterfall model is a sequential software development model that is used to develop new software in a standardized approach. Figure 1 shows a graphical representation of the model.

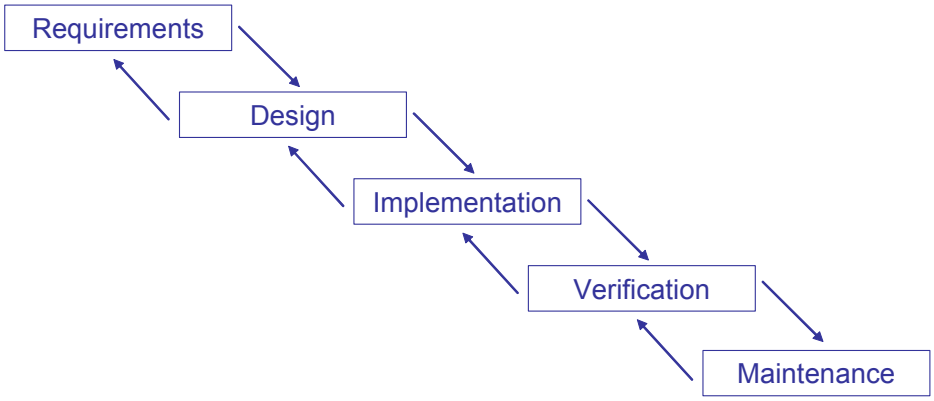

Figure 1 Waterfall model

To find the answer to the main research question on the use of Diameter of this thesis the first two phases of the waterfall model are passed through; requirement analysis and design. Due to time constraints the implementation, verification and maintenance phases are out of scope. The phases appear in the following research sub-questions.

2.2 Scenario based requirements elicitation

For the first research sub-question on requirements, the requirements must be elicited that should be applied. Requirement analysis can be done using different methodologies. Interviews, workshops, prototyping or use cases are used to gather requirements [Sharp et al, 2007]. The methodology used here is the scenario based

requirements elicitation [Whittle et al, 2004]. A scenario is used to derive the requirements from. The scenario is drawn up using experts that are familiar with the FoneFreez project. One possible scenario is studied and the requirements follow from this scenario. The scenario itself is described in chapter 4. From that scenario the requirements described in chapter 5 are derived. The requirements are divided in four different types: functional requirements, non-functional requirements, constraints and acceptance criteria.

This method is chosen because of the adopted case study method described in the next section. Given the scenario, this type of requirements elicitation was well suited.

In this thesis the requirement specification is a solution description, rather then a problem description [Wieringa et al, 2003]. The requirements specification describes the desired functions and quality attributes of a solution.

2.3 Case study

For the answer to the second research sub-question on the project case and its IMS components, the case study method is used. With this method, the case of the project is described and the IMS components that are used are identified.

A case study is an empirical research method. Case studies can be helpful when doing research that needs to be studied in context [Perry et al, 2004]. In this thesis a case study is done because the environment where the AAA must be added is very important. Without this context of interaction services from different realms, the study could not be performed. The case study is used to investigate the problem and identify the requirements which the AAA solution for interacting services from different realms has to meet.

In this thesis the single case study of the project at TNO-ICT is used to generalize to an overall solution for adding AAA to interacting services from different realms. Generalizing from a single case study is a difficult process which is often criticized by researchers [Aha, 1992]. Nevertheless, the FoneFreez project is believed to be found suitable for generalization. This because the interacting services is a concept that is already generalized in literature [Aepona, 2007] and the multi-domain issue is studied. Because the case where the AAA is added can be generalized, the AAA solution is also likely to be generalized.

2.4 Literature

A literature study is performed to find the current Diameter based architectures for interacting services from different realms. This will give the answer to the third research sub-question on current AAA architectures. An overview of the literature study can be found in chapter 3. Appendix A describes more about the Diameter protocol in depth.

First generic AAA literature is reviewed, to find out what AAA exactly means and its limitations. After which the Diameter protocol documentation is studied in detail, to give a thorough understanding of the protocol and its possibilities. The RADIUS protocol is reviewed to place the Diameter protocol in context and find out the differences between the two AAA protocols. Finally service interaction, realms and IMS is explored to understand all the aspects of the main-research question.

2.5 Identification of phases

For mapping the AAA architectures found in the literature study, another methodology is needed. Different phases are identified to find a way to map the AAA architectures on the case.

The distinguishing of the phases is done according to the phases in [Das et al, 2004]. The registration phase and authentication phase are set apart because of the frequency that they appear. The registration phase appears once for every user, where the authentication phase appears multiple times. Every time a user logs-in to the interaction service, the authentication phase is passed through. After the authentication phase the service can take place as meant to be. This phase was not mentioned in [Das et al, 2004], but is called in this thesis the operational phase.

The registration phase and authentication phase are renames for clarity. The registration phase also includes the preparation of the service and is renamed the initialization phase. The authentication phase is not only authentication but also authorization and part of accounting that is involved with the log-on process, so this phase is renamed to the log-on phase.

After the identification of phases, expert opinions can be used to verify the correctness of the mapping. Experts are used because of the lack of understanding of the complex matter this case describes. There are different kinds of experts: 'hands-on' experts, scientific experts and process experts. In this thesis different scientific

experts will provide qualitative information about the subject that lies within their specialty. Individual interviews and expert groups are methods used to retrieve this information. Another known method for retrieving this kind of information is the Delphi method [Turoff, 2002], but because of the time and complexity of this method it is not used in this thesis information retrieval.

2.6 Design methodology

In this thesis neither protocol nor protocol component is implemented. For these types of designs several methodologies can be used. For example: top-down or bottom up. For designing a specification for the problem described in this thesis, no standard methodology was found. This section describes the way to come to an answer to the fifth research sub-question; which specification can be derived.

In this section the path is described to come to a complete design for AAA in service interactions from different realms. First the problem is simplified by looking at service interaction in one realm. Next the problem is refined by adding realms until the situation contains a different realm for every party.

Figure 2 shows the possible paths to the solution. Number four is the design in which every party involved is placed in a separate domain. Number one is the design where they are all in the same realm. To come from one realm to design four, there are two possibilities. When proceeding to the right, the service providers are placed in separate realms. When proceeding downward, the application service provider is placed in a separate realm. So design two is the design where the two service providers are located in separate realms. When placing the application service provider from that design in a separate domain, the solution is reached as described by design four.

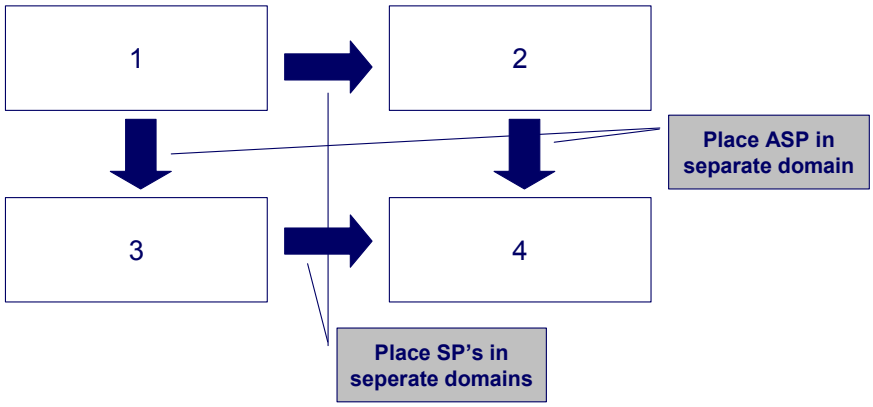

Figure 2 Path to solution

Section D.1 describes design one. Design two and three are described in sections D.2 and D.3. The solution, where every role is placed in a different domain, is described in section D.4.

Unified Modeling Language (UML) is a standardized specification language for object modeling [Braun et al, 2001]. In this thesis UML is used to present the specification in a standardized method.

The following diagrams of UML are used in this thesis: use case diagram and sequence diagram. The business roles model and functional designs could not be comprised in UML. Because the design is not detailed to the level of software development, class diagrams and physical diagrams are not needed.

2.7 Verification of requirements

There are different methods to verify if the requirements are fulfilled. One method is building the solution according to the specification and use tests to verify the requirements. Another is a formal method, like the Specification and Description Language (SDL) to verify requirements [SLD, 2007]. Both methods could not be used in this thesis. The first method was not appropriate because only the specification was drawn up, and the proof of concept was out of scope. The second method can be used if the requirements can be formalized. The type of specification is not to that depth that formal requirement validation can be used. For formal requirements validation checkers like Spin [Spin, 2007], states and actions must be specified, which is not the goal in this study.

In this thesis the specification is drawn up, and for each requirement must be described which messages of the specification fulfill this requirement. Experts are used to review the requirements and determine the fulfillment. This method was also used in the European project Cybervoting [Forsgren et al, 2001].

2.8 Conclusion

Different methods are used to answer the research sub-question and the main research question. A method that is used to predict the feasibility of the Diameter protocol is a model that is developed at TNO-ICT. This model identifies the key success factors for telecom standards. This model and the fulfillment of the model can be found in Appendix B.

The most important standard methods used are: the case study method, the scenario based requirements elicitation, and study of literature. The other methods used in this study are mapping through phase identification, design by problem refinement and verification of requirements by experts.

3 Theoretical background

In this chapter the concept of AAA is explained and other terms used in the problem description are further defined, like the Diameter protocol, interacting services, realms and IMS. In grey textboxes simple examples are given to illustrate the theory in this chapter.

3.1 AAA

AAA stands for Authentication, Authorization and Accounting. This section looks into the meaning of AAA, and the models used for authentication, authorization and accounting.

Authentication is the verification of the identity of the entity. An entity can be a user or the device a user has, like a computer or the SIM of his mobile phone. With authentication someone can prove that it is really the person or device he or it claims to be. This prevents from impersonations from other parties. Authentication consists of three sorts: user authentication, message authentication and device authentication [Thales, 2006].

Joe wants to get some money from his bank account. He goes to the ATM machine of his bank in inserts his bankcard. The ATM machine wants to know if it is really Joe that tries to withdraw money. The ATM machine asks for the PIN code belonging to the bankcard. When Joe enters his PIN code correct, the bank has *authenticated* Joe as the owner of the card.

Authorization is the determination whether the requesting entity is allowed access to a particular resource. Authorization is the process of determining if the user has the right to access the network or use services, like the print server from that network. Furthermore, authorization is needed for resource reservation and quality of service support.

Now Joe can enter the amount of money he wants to withdraw. The ATM machine checks with the bank if the amount Joe is asking for, is not more than he has on his account. If there is enough money left in his account, the ATM is *authorized* to hand out the requested amount to Joe.

Accounting is the collecting of information about resource usage for the purpose of capacity planning, auditing, billing or cost allocation. For example, records are kept about the duration a user surfs the Internet.

Joe's balance must be updated to process the withdrawal. The withdrawn amount is deducted from his account. *Accounting* is the registration of the withdrawal.

Re-authentication is the renewal of the authentication by the client upon request of the server. When a session lifetime has expired, or when an error has occurred in the path, re-authentication can be necessary to ensure trust.

When Joe enters a wrong PIN code, the ATM machine asks again for the PIN code of Joe. Joe is *re-authenticated* by entering this PIN code again.

3.1.1 Authentication models

As mentioned above, there are three different levels of authentication: user authentication, device authentication and message authentication. User authentication is the verification of the identity of the user; this can be done using authentication protocols like Diameter and RADIUS. Device authentication is sometimes needed when the device is from another, not trusted domain; here protocols like Kerberos can be used. Message authentication is used to authenticate messages without their context of a session. Digital signatures can be used to provide message authentication. [Thales, 2006].

For authentication a two-party model and a three-party model exists. The two-party model is used when two peers interact. A client and server are directly interconnected with no involvement of the middle nodes like gateways or proxies.

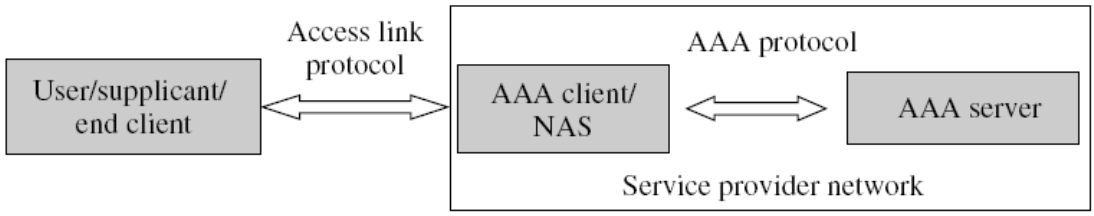

Figure 3 Three-party authentication model [Nakhjiri et al, 2005]

A three-party authentication model is shown in Figure 3. In this example a user wants to access a network, like the network of his internet service provider. The user

wants to connect to the network and connects to the first edge device, which is the Network Access Server (NAS). The NAS acts as an AAA client who connects to the AAA server to authenticate the user. The AAA server makes the decisions regarding granting access to the user.

Joe goes to the ATM machine to withdraw money. He enters his PIN code, and with that the ATM machine verifies with the bank if this is correct. Also authorization is granted by the bank to give Joe is money. In this case, Joe's contact with the bank goes through the ATM machine. The ATM fulfills the role of the *AAA client*. The bank takes the decisions about authentication and authorization and fulfills the role of the *AAA server*.

3.1.2 Authorization models

The informational RFC [RFC 2904] about AAA authorization frameworks distinguishes three different architectural models for authentication: the agent sequence, the pull sequence and the push sequence.

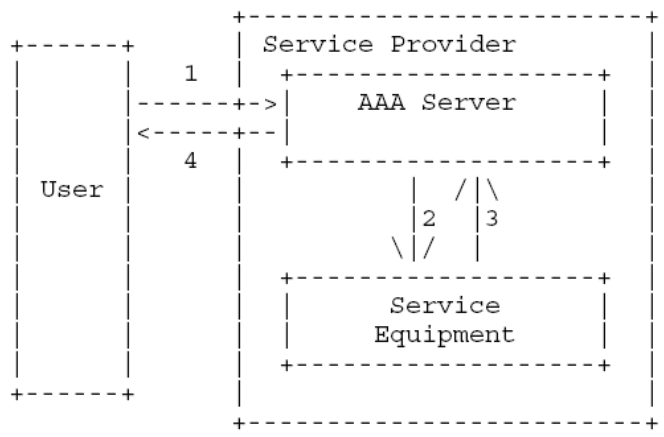

Figure 4 Agent sequence [RFC 2904]

In the scenario of the agent sequence showed in Figure 4, the user contacts the AAA entity first. The AAA server authorizes the user, and the service equipment is notified. The service equipment can set up the service and notifies the AAA server that it is ready, which notifies the user. The user and service equipment can precede the communication directly, without the AAA server functioning as an agent. An example of this situation is when a user requests Internet access. The user is first connected to the AAA server of the internet service provider. When the AAA server has authenticated the user, the proxy of the service provider is notified and the connection is established.

Before Joe can use his bank card, he has to prove to the bank that he is really Joe. He visits the bank office and shows his passport. Now he has proven he is Joe and is authorized by the bank that he can use the ATM machines. The system is updated that the ATM machines can accept Joe's card. Joe is informed that he is allowed to use the ATMs.

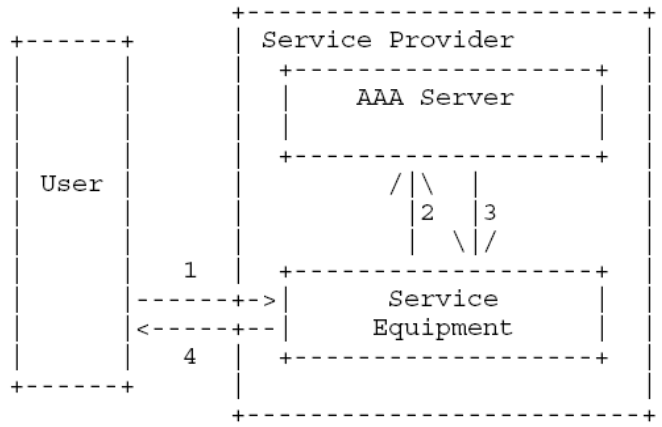

Figure 5 Pull sequence [RFC 2904]

Figure 5 shows the pull sequence. The user directly requests the service from the service equipment, which authorizes the user by placing a request at the AAA server. An example of this situation is when you pay with your credit card and the store checks with the credit card company if the card is still valid.

Joe withdraws money from his bank account. He goes to the ATM machine. The ATM machine contacts the bank and the bank authorizes that he has enough money, and can withdraw the requested amount. The ATM hands the amount to Joe.

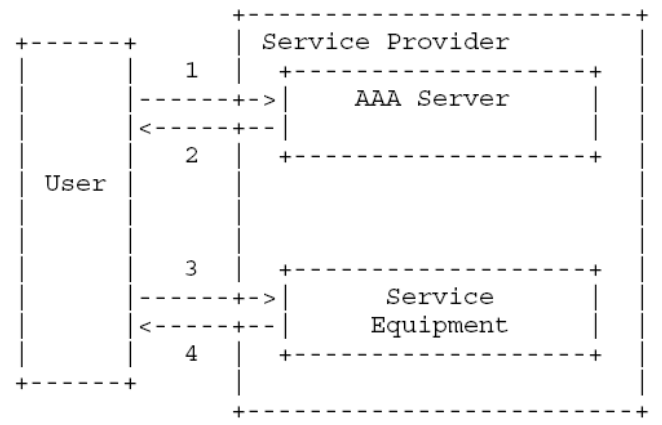

Figure 6 Push sequence [RFC 2904]

The push sequence is shown in Figure 6. The user receives a token from the AAA server with which the user can request the service and prove that it is authorized to use the service. An example of this situation is if you are going to the theater. First you buy a ticket at the box office. Before entering the auditorium the attendant requests for your ticket, which is prove that you have paid.

Most bankcards also have a 'chipknip', a kind of e-wallet. Before Joe can use his 'chipknip' in a store to pay for small purchases, the bank must give authorization. This authorization token is the credit that is placed on the chip of the bankcard. With this token the store knows that Joe has enough money on this 'chipknip' to pay for the product he wants to buy.

3.1.3 Accounting overview

The billing process describes all the sub processes needed for getting an invoice to the user for having used services. Figure 7 gives an overview of all the sub processes covered by billing, in particular accounting.

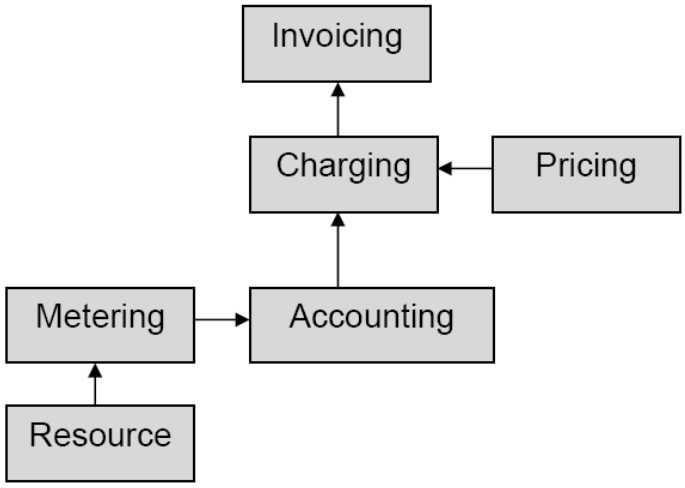

Figure 7 Accounting overview [ICOM, 2006]

Assume that a user consumes services generated by a resource. Then *metering* is the process which collects consumption statistics at a specific resource in the network. *Accounting* is the collection of this metering information, stored in accounting records. *Charging* combines the pricing information (set by *pricing*) and the accounting records and calculates the charging records, i.e. the amounts the user has to pay. Finally, *invoicing* is the process of consolidating the charging records on a per customer basis and sending an invoice to the user [Hinard et al, 2006].

AAA servers are able to collect metering information from the resource, for example from the Network Access Server about how long a user is surfing on the Internet. They order this information in accounting records, which are kept per user. This is what is meant by accounting in AAA. Note that invoicing and the handling of cash flows are never done by AAA protocols.

If Joe is abroad and wants to withdraw cash from his bank account, it is possible that he has to pay for that transaction service. The foreign bank has a price, e.g. 2 Euro, which a withdrawal costs and calculates for Joe how much transactions he has done. The foreign bank sends an invoice to Joe's bank for the withdrawals and the transaction costs.

3.2 RADIUS

RADIUS [RFC 2865] stands for Remote Access Dial-In User Service and provides authentication, authorization and accounting. The protocol was developed for providing authentication to the network access process.

The term AAA was first used with the RADIUS protocol. Predecessors of RADIUS like TACACS+, were also protocols that could do authentication, authorization and accounting, but these three services are separated in the protocol. RADIUS is the first protocol that combines these three services in its messages.

The RADIUS protocol was standardized in 1997 by the IETF in [RFC 2039]. Later this RFC was superseded by RFC 2865, which is the current standard of the RADIUS protocol.

The AAA protocol RADIUS was developed in the early 90s. At that time Internet was used differently; people were using dial-in to connect to it. With the development of web 2.0 and the ever increasing capabilities of routers and Network Access Servers (NAS), the demands changed and created the need for a replacement of the RADIUS protocol.

3.3 Introduction to Diameter

In September 2003 a new AAA protocol Diameter was standardized. The Diameter protocol was developed to resolve the issues that RADIUS left open. In new application areas like Wireless Local Access Network (WLAN) and Voice over IP

(VoIP), Diameter is better suited and gives better support for roaming users. A full comparison of the Diameter and RADIUS protocol is given in Appendix A.4.1.

The Diameter protocol is standardized by the IETF in [RFC 3588]. The development of Diameter is currently focused on supporting access to IP networks. Because of the flexibility of the protocol, it can be used for generic purposes in the AAA domain.

The Diameter protocol consists of the Diameter base protocol and Diameter protocol applications as shown in Figure 8. The applications are extensions of the Diameter base protocol.

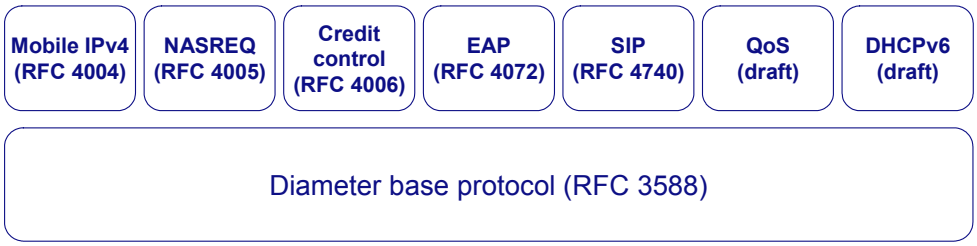

Figure 8 Diameter framework

In the base protocol the functionality is implemented that is common in all supported services, like mechanisms for reliable transport, message delivery and error handling. The base protocol must be supported by all applications.

A Diameter node is a client, agent or server. A Diameter client is a device at the edge of the network that performs access control. A Diameter agent can be a relay, proxy, redirect or translation agent. In Appendix A.2.1 the differences between these agents are explained. The Diameter server handles the authentication, authorization and accounting requests for a specific realm. A realm is an administrative domain where the server resides in.

A Diameter agent may act in a stateful manner for some requests while being stateless for others. An agent can also be one type of agent or server for some requests, but another type of agent or server for other requests [IBM, 2006; RFC 3588].

Diameter runs over Transmission Control Protocol (TCP) [RFC 793] or Stream Control Transmission Protocol (SCTP) [RFC 2960]. The different Diameter nodes are interconnected in a peer-to-peer structure. The Diameter framework enables push and pull application models and architectures as earlier described in subsection 3.1.2

[HP, 2002]. The Diameter base protocol defines the protocol header and the necessary Attribute-Value Pairs (AVPs). Diameter uses Attribute-Value pairs to send data or AAA specific information. Diameter AVPs are defined in the base protocol or Diameter application documents. The applications can extend the protocol by defining new messages and AVPs and append them to the Protocol Data Units (PDUs).

The Diameter protocol does not share common PDUs with RADIUS. For backward compatibility with legacy protocols, a translator is necessary to translate the Diameter and RADIUS PDUs.

The two most important Diameter applications for this thesis are briefly described here. A full overview of Diameter applications is given in Appendix A.5.

3.3.1 NASREQ application

The NASREQ application is the direct replacement of the authentication and authorization part of the RADIUS protocol. This application specifies the inter-working between the Diameter and the RADIUS protocol, for backward compatibility.

The NASREQ application defines extra commands for authentication. First an *AA-Request* (AAR) is sent to the server with the credentials of the user and after authentication an *AA-Answer* (AAA) is sent back. The *Re-Authentication-Request* (RAR) can be used by the server to verify if the user is still using the service. The client sends back a *Re-Authentication-Answer* (RAA), where after an AAR and AAA message should follow. The session can be terminated by the server or client. The server can send an *Abort-Session-Request* (ASR) or the client can send a *Session-Termination-Request* (STR). The accounting is done by the *Accounting-Request* (ACR) and *Accounting-Answer* (ACA) messages. The message flow can be found in Appendix A.5.2.

3.3.2 Credit control application

The Diameter Credit control application provides real-time credit-control for different end-user services. The application is only concerned with credit authorization for prepaid subscribers. Some accounting features are already specified at the base protocol, but these are not sufficient for real-time accounting for prepaid subscribers.

Two types of events can be seen at the application: session based credit-control and one-time events. Price enquiry, user's balance checks and refund of credit on the user's account is usually done in one-time events. For these two types of events, there are two different credit authorization models: authorization with money reservation and credit authorization with direct debiting.

The money reservation model is session based and works as follows: the server rates the request from the client and reserves a suitable amount of money from the user's account. Resources corresponding to the amount are returned to the user. When the user runs out of resources or ends the service, the client reports back to the server how much is used. The server returns money when resources where left over or can make a new reservation. The money reservation model is session based. A credit-control session always consists of first, possibly intermediate and final interrogations.

Credit authorization with direct debiting is a one-time event. The server directly deducts the right amount of money for the request from the user's account.

Two messages are added by this Diameter application: *Credit-Control-Request* (CCR) and *Credit-Control-Answer* (CCA). Message sequences can be found in Appendix A.5.3.

3.4 Interacting services

Value added services based on only one basic service like telephony or television, are no longer the only kind of value added services. Value added services that combine two basic services are developed and a feature is added on top of that. An example of such a value added service is the FoneFreez service, where the television is paused during the phone call. The FoneFreez service is a very simple example of a value added service based on service interaction.

Services that intervene in each others behavior are called interacting services. In this case the interacting services do not interact directly but use a manager to control the interaction. The Service Capability Interaction Manager (SCIM) is an example of a decision making entity that is connected to all parties involved. The SCIM is an IMS function as later described in this chapter. The manager has interfaces with different value added service parties to interfere with the service of the basic service providers.

The service interaction in FoneFreez is a first example of service interaction which shows the possibilities that service interaction enables. There are other services thinkable that will be useful for a large public. For example when watching television and the phone rings, information appears on the television screen about the caller. When it is an important phone call the user can press a button to answer the call. Video telephony is used to give the user a picture of the caller on his screen. This type of service interaction combines two services, but also more services could be combined in service interaction. When adding presence information for example, three basic services are combined. A user can have a buddy list with presence information, which is used in a list on your television screen to see who is in the opportunity to answer a call.

3.5 Realms

A realm is an administrative domain often associated with ownership. An administrative domain is the collection of resources (hosts, routers and the interconnecting networks) under the control of a single administrative authority. Such an authority can be an internet service provider. The internet service provider has a network that is registered under one domain, e.g. xs4all.nl. The difference between a domain and a realm is that a domain can consist of multiple realms. In this thesis the terms realms and domain are used interchangeably.

A user has a home realm, with which the user maintains an account relationship. The user is registered with an internet service provider for internet connectivity. He has to pay for the service the internet service provider delivers to the user. The Network Access Identifier (NAI) is used to find out to which realm the user belongs. With the UserID@realmID formatted identifier the realm of the user is identified. An example of such a identifier is Wendy@xs4all.nl. More about NAI can be found in [RFC 2486].

Most of the time only one realm is involved when talking about a corporate network or home network. But because the Internet is a gathering of interconnected networks, the number of realms in the Internet is very large and the interconnections of these realms become interesting.

An example of an issue with multiple realms is roaming. Roaming is the situation where the user is connected to a network that is not the network of its own internet service provider. In this case the user is located at a different realm as to which it

belongs, and the realm which the user is visiting has to verify authentication and authorization of the user at its own realm.

Joe wants to use an ATM of another bank. He inserts his bankcard and enters his PIN code. This is verified by the ATM because this information was stored on the card itself. To withdraw cash, the ATM machine must first contact Joe's own bank, to find out whether he has enough money on his account. His bank answers to the ATM machine that it is ok, and the cash can be handed out.

3.6 IMS

IMS stands for IP Multimedia Subsystem and is a standardized framework used by telecom operators to provide mobile and fixed multimedia services in an all IP environment. Its purpose is to make network management easier and to provide better interoperability, roaming between networks and enable network convergence.

3.6.1 IMS architecture

IMS is an architecture that consists of functions and standardized interfaces which are defined by the 3rd Generation Partnership Project (3GPP). There are different releases of IMS and it is work in progress. At the moment 3GPP is working on release 7. Work is also done by the standardization body Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) from the European Telecommunication Standards Institute (ETSI), which calls its IMS standard 'Next Generation Networking' (NGN). The International Telecommunication Union (ITU) is also active on cooperation with ETSI and 3GPP on the standardization of IMS. The protocols used for communication within IMS are standards of the IETF [3GPP, 2007; ETSI, 2007; ITU, 2007; Bertrand, 2006].

The IMS system transports signaling information, the data is transported over transport networks like Universal Mobile Telecommunication System (UMTS), fixed and mobile networks. The IMS architecture is a horizontal architecture and consists of three layers: the application plane, the control plane and the transport plane. The application plane contains application and content servers that run value added services for users. The Service Capability Interaction manager (SCIM) provides an interface to the control plane to enable combinations of the applications that run on the application servers. The control plane contains different functions like the Home Subscriber System (HSS), Call Session Control Function (CSCF) and border gateways (BGCF) that control calls and sessions, Media Resource Function Controller (MRFC),

and support functions like provisioning and charging. The transport plane consists of Media Gateways (MGW), routers and switches for the backbone and access networks, both fixed and mobile [Ericsson, 2004].

A simplified version of the IMS architecture is shown in Figure 9. The complete architecture is too complex to represent in one picture. The CSCF handles the call and functions as a SIP server. The CSCF is decomposed in different types of session control functions: serving (S-CSCF), interrogating (I-CSCF) and proxy (P-CSCF) call session control functions.

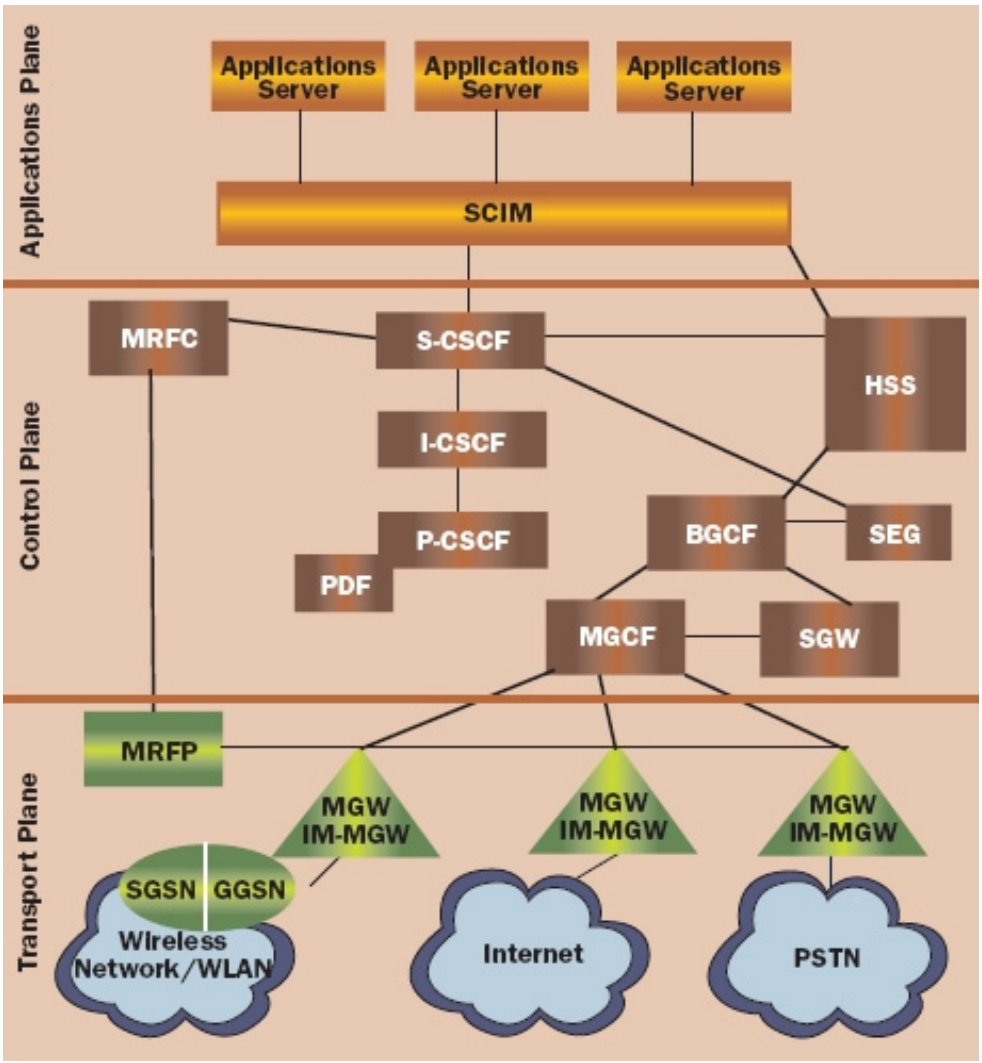

Figure 9 Simplified IMS reference architecture [Fried et al, 2006]

The Diameter protocol is used for provisioning of AAA in the IMS architecture. More details on Diameter and IMS interfaces are given in Appendix A.5.8.

3.6.2 IMS inter-domain

The IP multimedia subsystem (IMS) is managed by one operator. If multiple operators want to communicate using IMS applications, they must interconnect their

networks. It is possible that both networks are IMS networks, or only one of them. The IMS networks are interconnected using the Za interface, which is not further specified yet [Sher et al, 2006].

Networks from different security domains are interconnected through Security Gateways (SEG). The SEG's ensure that the IMS network is securely connected to other networks and protects the traffic between the networks and the IMS core.

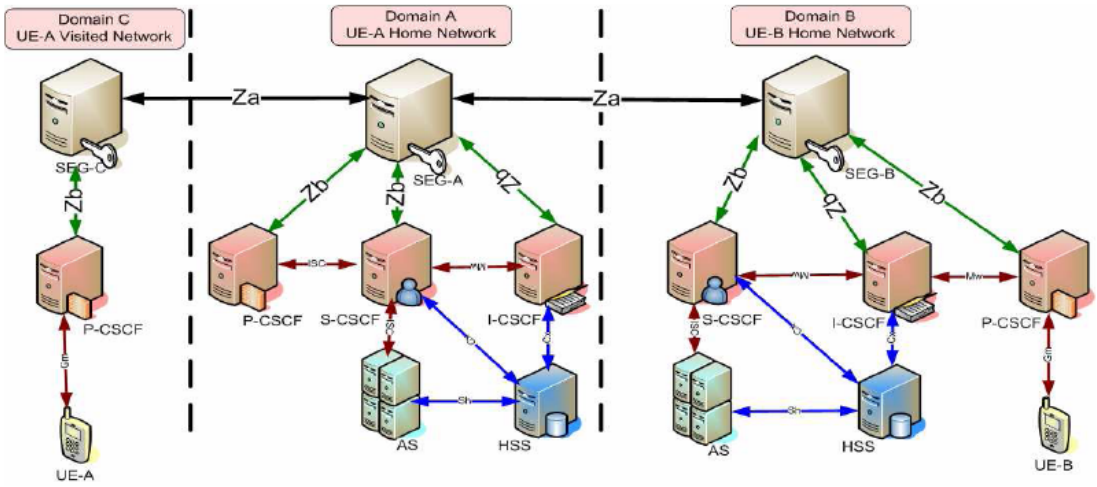

Figure 10 IMS inter-domain architecture [Sher et al, 2006]

With the architecture shown in Figure 10, not only IMS networks can be interconnected, also other IP networks can be connected to an IMS network. In this architecture the applications from the IMS domain can be used in other domains as well [Mayer, 2006].

The IMS network is developed to run in a single trusted administrative domain under management of one operator. It is not possible in the current design to split the IMS network and divide it over multiple realms. The IMS network should be managed by one single party. As described above, multiple parties with their own IMS network can be interconnected, but IMS is not developed with the idea that multiple parties manage a single IMS network.

3.7 Conclusion

The three A's in AAA stand for authentication, authorization and accounting. Diameter is a protocol that can provide AAA to end-users. Diameter is especially designed to handle issues that arise when multiple realms are involved. Service interaction in this thesis is defined to be between services from different parties. AAA for interacting services from different realms is an unsolved issue.

Diameter is used in the IMS reference architecture at different interfaces. The IMS architecture is used by telecom operators for multimedia services. Service interaction is part of the IMS functions, but there it is not possible to realize multi-domain service interaction as is the case in this thesis.

4 Case description

This chapter describes the case that is used to derive the requirements and architecture used in this thesis.

The service is called FoneFreez and consists of an IP television service and an IP telephony service. The FoneFreez application interrupts the television service when the user receives a phone call. The functionality of the service is described below.

When the user comes home after a day of work, the FoneFreez service is activated. The user starts to watch television to relax at the end of the day. Suddenly the phone rings, and the user walks to the phone to answer it. When he picks up the phone the television show he was watching is paused. After the telephone conversation, the user hangs up the phone and the television show proceeds where it left off, from the moment the user picked up the phone.

First the architecture of implementation of the FoneFreez service is described and the interaction between the devices is shown. This is followed by the different business roles that are distinguished in the FoneFreez service.

4.1 Architecture

The current architecture of the FoneFreez service, as built by TNO is shown in Figure 11.

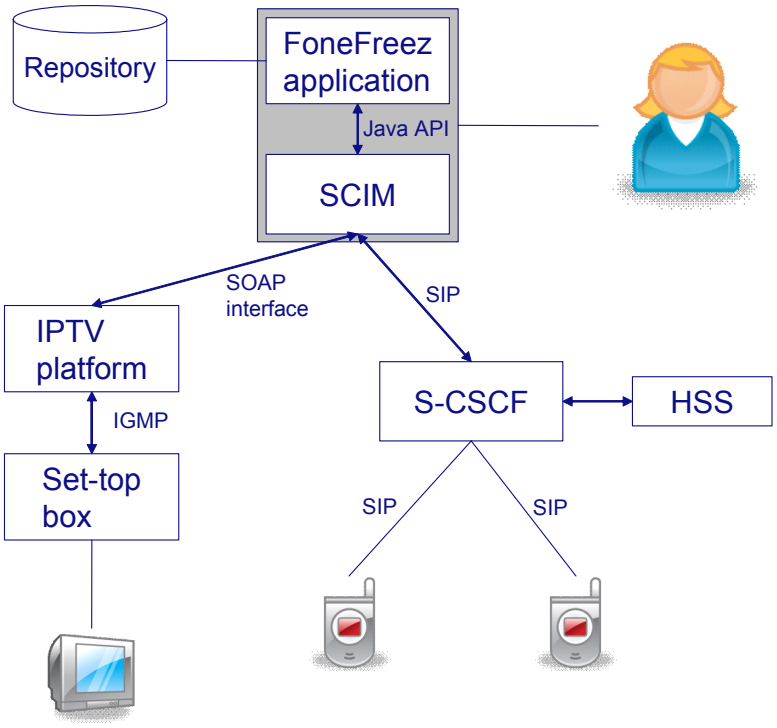

Figure 11 Architecture FoneFreez

The two different services, IP television and IP telephony, can be used stand alone, without the FoneFreez application. The Service Capability Interaction Manager (SCIM) provides the interaction between the television and telephone service and decides on which events to take action. In this implementation all components reside in the same realm; the TNO's research realm. It is also possible that they are located in different realms, this is taken into account at the design of the implementation of the project.

The IP telephony services architecture is based on the IMS architecture. The Serving-CSCF (S-CSCF) functions as the SIP server and the HSS stores the user profiles. The SIP protocol is used to communicate between the telephones, the S-CSCF and the SCIM. Because the SCIM is part of the IMS architecture, it is commonly located in the same realm as the IP telephony service.

The IPTV service consists of a streaming server. The server has a Simple Object Access Protocol (SOAP) interface on which other applications like FoneFreez can interact with the television service. Multicast is used to provide the channels to the user. The user uses a set-top box connected to the television, to receive the television signal.

The two functional entities, the FoneFreez application server and the SCIM, are in this case combined in one physical device. Next to the application server a repository resides. In this repository the combinations of the ID that the user has at the telephony service and the ID the user has at the television service are linked together. In this way the FoneFreez service can interrupt the correct television service when the phone is picked up.

4.2 Interaction

Initially, the user has a relationship with the IPTV service and the IP telephony service. When the user wants to use the FoneFreez service, see Figure 12, it sends its ID's of both services to the FoneFreez server. These are then stored in the repository. The FoneFreez service checks if both services are part of the group of services the FoneFreez application has relations with. The application server requests the SCIM to retrieve all the events from that user from the IP telephony service. The FoneFreez application defines what action to take when an event occurs. Then the SCIM acts on the event and makes a SOAP-call to the IP television platform.

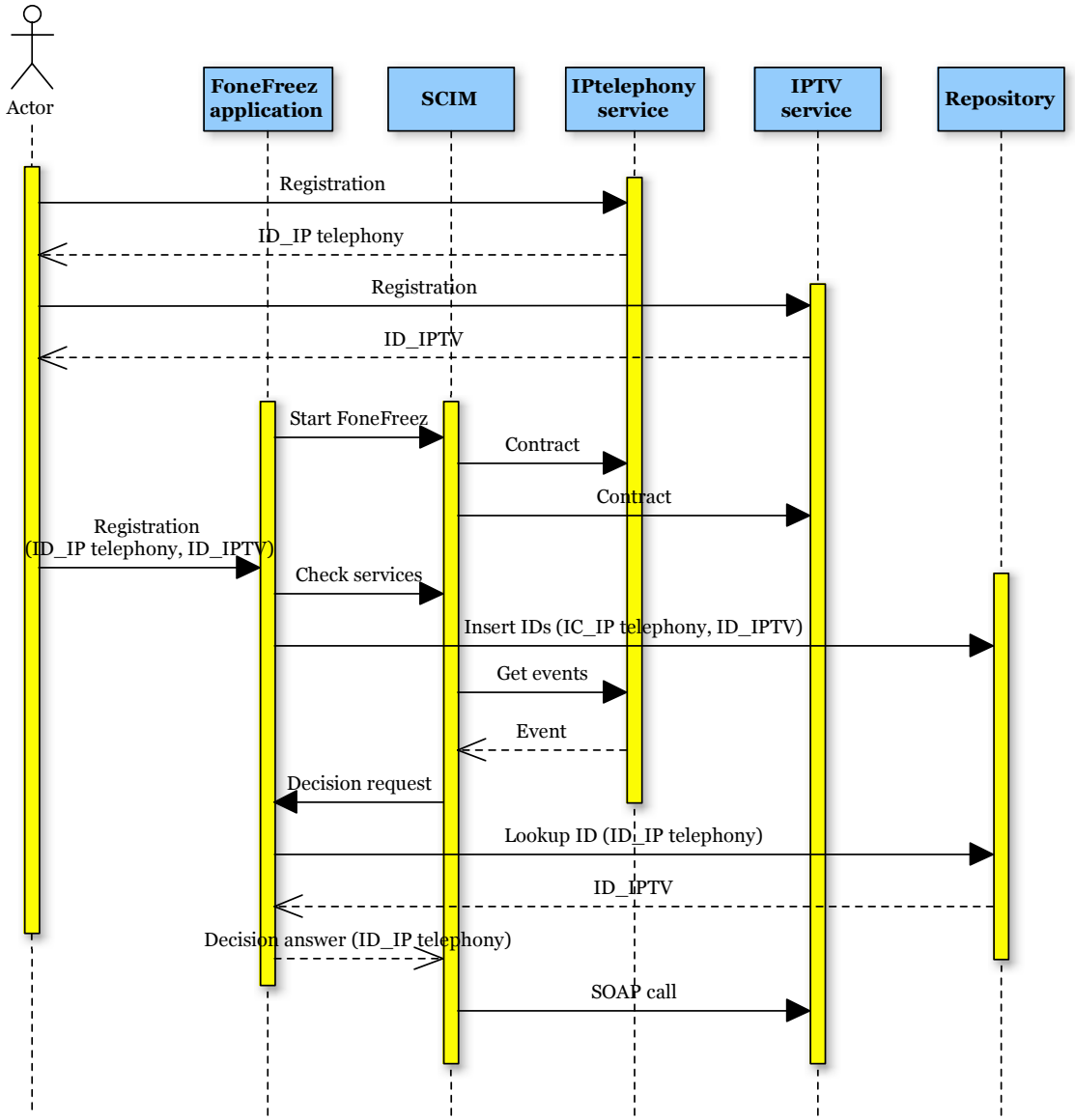

Figure 12 Sequence diagram case

4.3 Business roles model

The business roles that are identified for this FoneFreez case are:

- End-user
- Service provider providing IPTV
- Service provider providing IP telephony
- Application service provider providing the interaction service
- Broker providing the interaction between the service providers and the application service provider.

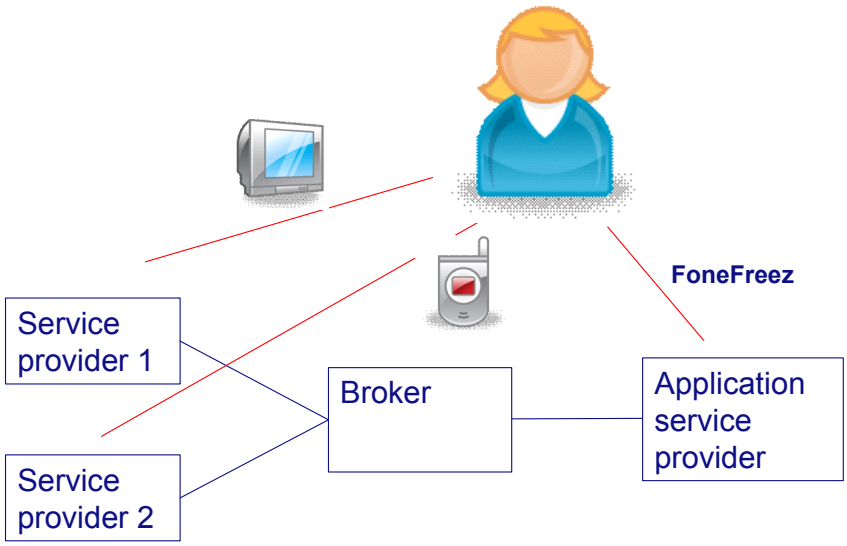

Figure 13 Business roles model

As can be seen in Figure 13, three different roles have a relationship with the end-user. The FoneFreez service is a complementary service and is not a packager for the IPTV and IP telephony service. This is why the relationships of the end-user with the service providers remain to hold after the user decides to subscribe to the FoneFreez service.

The reason why there are four different roles identified apart from the end-user, is that with three roles the possible functionalities are not decomposed sufficient. A conceivable alternative is the combination of the broker and application service provider, or the combination of one of the service providers with the broker.

In the implementation of FoneFreez the broker and application service provider are implemented in one device. This is a logical explanation if you look at the FoneFreez service with little value added on top of the interaction. When more value is added, the separation of the broker and application service provider is desirable.

In the current IMS implementations, the broker is combined with service provider. In this implementation it is possible to have a third party application service provider. Because of both implementations mentioned above, four different roles should be identified.

The business roles can be fulfilled by four different actors, but also combinations can be made. An actor can fulfill different business roles, e.g. the broker and application service provider roles can be fulfilled by one party as done in de FoneFreez case. A

minimum of one actor and a maximum of 4 actors (besides the end-user) are allowed in this case.

In theory the business roles could be further decomposed in smaller roles, but that is not done for this case. For the reason that it is feasible that the roles defined above are played by one actor and not more actors. How the roles are further decomposed is left to the actor implementing the role.

4.4 Implementation of business roles

The roles are implemented with different functional entities. The application service provider is implemented by an application server functional entity. The role of the broker is implemented by an interaction manager. The service provider roles are implemented by service platforms. These service platforms consist of more functional entities like in the IP telephony situation, an S-CSCF and HSS entity, but this level of decomposition is not needed at this stage. The functional entities each in their own domain, separated by the dashed lines, are shown in Figure 14.

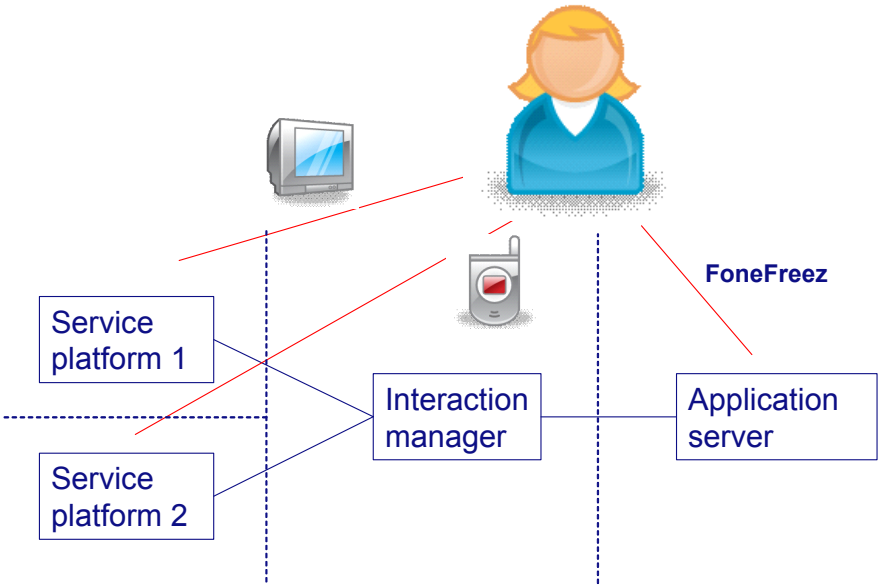

Figure 14 Implementation of business roles

In the implementation of the FoneFreez service the broker role is implemented using the service capability interaction manager (SCIM). The SCIM is at the moment specified by 3GPP in a way that it can only support SIP-SIP interaction and for example not SIP-HTTP interaction. For that reason an 'interaction manager' functional entity is defined which can interconnect different services, for example IPTV, presence and IP telephone services. Another reason to renounce from the SCIM is that the interface between the SCIM and the S-CSCF is a static connection,

and these functions cannot reside in different realms in the current IMS specification. The interaction manager, as meant in this document, is the implementation of the broker, which provides interaction between different application servers and underlying services.

4.5 Conclusion

In this chapter for the case study, the case FoneFreez is described. The FoneFreez project built at TNO is used in this thesis as case study. The FoneFreez service is a service interaction service that provides interaction between IP telephony and IP television service. Four different business roles are identified: end-user, application service provider, broker, and service provider. The roles are implemented by the following functional entities: application service provider by an application server, broker by an interaction manager and service provider by a service platform (two times). We renounced from the SCIM as implementation of the broker, because it does not support other interfaces than SIP and it can not be placed in a separate realm under the current specification.

5 Requirements

This chapter describes the requirements on the specification and the constraints which the solution has to meet. In case of multiple possible solutions the acceptance criteria are used to evaluate which solution is best [Dodd, 2003].

The method that is used to formulate the requirements is denoted as scenario-based requirement elicitation [Whittle et al, 2004]. Scenario-based requirement elicitation means to generate requirements from scenarios. The scenario is drawn up with the help of experts from different fields of work: multi-domain, IMS, security, accounting. From the scenario the requirements follow logically, for example when the scenario is: 'the user registers with the service', then the requirements are: 'the user is identified at the service, is authenticated and authorized to use the service.'

First the scenario is described followed by the use-cases found in the scenario. The following sections of this chapter describe the functional and non-functional requirements. Then the constraints are given which the solution has to meet, followed by the acceptance criteria.

5.1 Scenario

This section describes one possible scenario that takes place when AAA functionality is added to the case described in the previous chapter. The scenario combines the events that can be seen from the end-user perspective and the events that take place between the other business roles.

A user registers first with an IPTV service at its cable television provider and IP telephony service at its telecom provider. The user is identified, authenticated and authorized to use both services. Accounting information about the usage of the services is maintained.

Then the user registers with the interaction service. He enters his credentials and information about which operators provide his telephony and television services. To be able to verify if the user is also registered at the IPTV provider and IP telephony provider, he enters his telephone number under which he is registered at the IP telephony provider and his address on which the IPTV provider delivers the television signal. The interaction service verifies with the IPTV and IP telephony provider if the user is indeed registered there.

After registration the interaction service starts and the user is authenticated and authorized to use the interaction service. The interaction service is authenticated to and authorized by the broker. The broker is authenticated and authorized by the IP telephony service from the telecom provider, and authenticated and authorized by the IPTV service from the cable television provider. They authorize the broker to intervene at their services.

The user starts to watch TV. A call for the user reaches the IP telephony service. The call is identified and authenticated at the broker. The broker contacts the interaction service, which identifies the user and looks up the users TV session and authorizes the call to continue to the user. The TV session is interrupted as the user picks up the phone. The broker makes a SOAP-call for the given session of the user on the IPTV platform and is authorized to place the call. When the user hangs up, the SIP server identifies the user and contacts the broker. The broker contacts the interaction service, to find out which action to take, and is authorized by the IPTV service to place a SOAP-call on the IPTV service to resume the users session.

At the interaction service the accounting records are maintained about the frequency and duration that the service is used. The user can be billed for the usage of the service. The costs are deducted from the pre-paid account of the user. The broker can register the usage of the interaction by the IPTV service and IP telephony service. All parties maintain accounting records about the usage of the service for which they can bill each other later or agree on a flat fee.

5.2 Actors and use cases

From the business roles and the scenario described above several use cases can be derived. The UML notation used for depicting the use cases in a use case diagram requires actors, which are the business roles. As described in section 4.3, in this case every business role is a different actor.

In Figure 15 the actors and their use cases are shown. The use cases are all authentication, authorization and accounting, for clarity depicted in one use case. The use cases in the box are part of the specification that will be developed in this assignment. The use cases outside the box are out of scope.

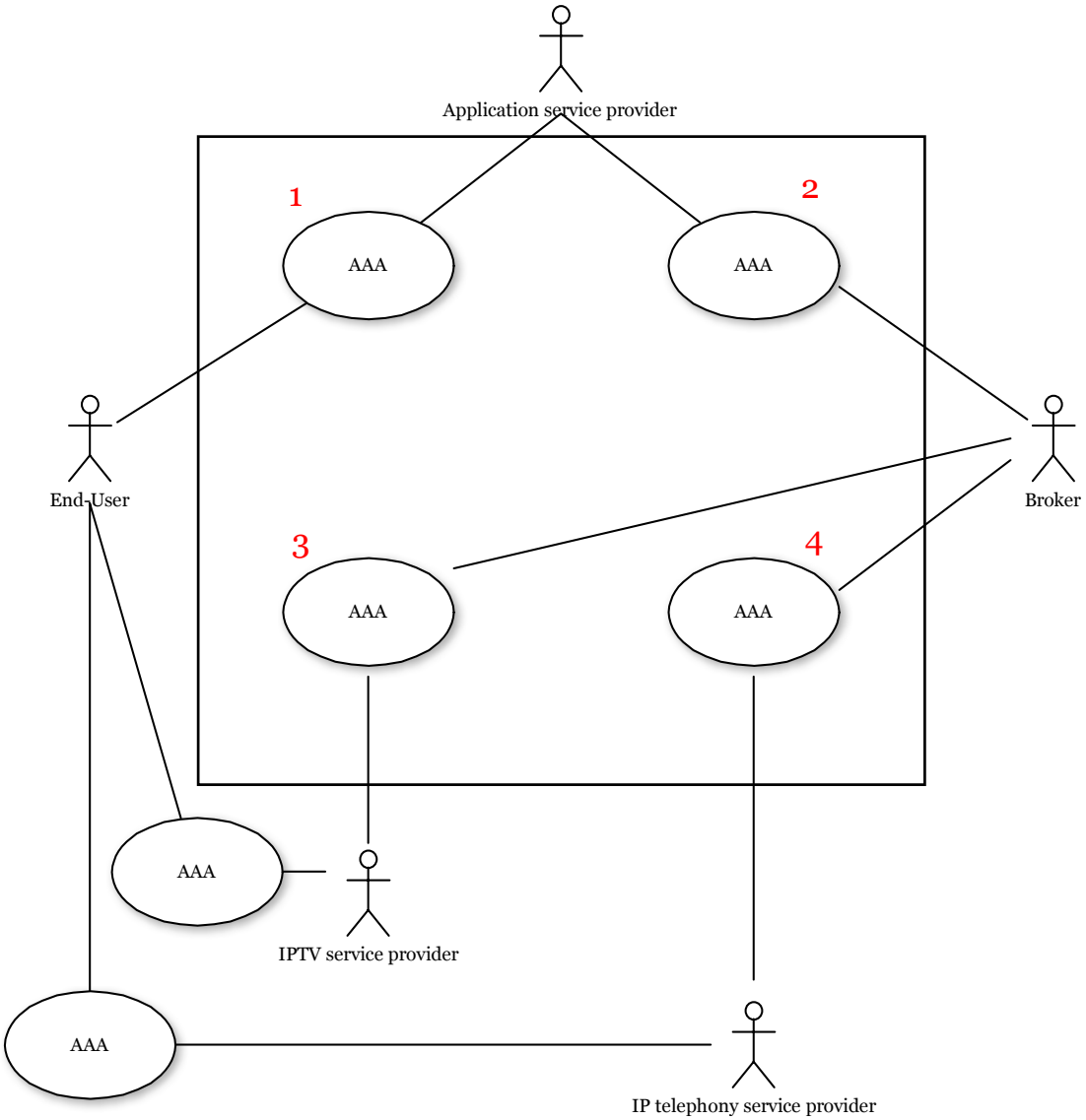

Figure 15 Actors and use cases

The numbers 1 to 4 are used at the requirements in section 5.3.

5.3 Functional requirements

This section describes the functional requirements that will be used as criteria during the specification of the AAA support for interacting services from different realms. These requirements are derived from the scenario described in section 5.1.

These requirements belong to the use case identified with number 1 in Figure 15:

- Identification of the interaction service user to the interaction service
- Authentication of the interaction service user to the interaction service
- Authorization of the interaction service user to the interaction service
- Accounting of the usage of the interaction service per user
- Re-authentication of the user to the interaction service

These requirements belong to the use case identified with number 2 in Figure 15:
- Authentication of the interaction service to the interaction manager
- Authorization of the interaction service to the interaction manager
- Accounting of the usage of the interaction manager by the interaction service
- Re-authentication of the interaction service to the interaction manager

These requirements belong to the use case identified with number 3 in Figure 15:
- Authentication of the interaction manager to the IPTV service
- Authorization of the interaction manager for a particular user at the IPTV service
- Authorization of the interaction manager for the execution of a particular event at the IPTV service
- Identification of the user at the IPTV service by the interaction manager
- Accounting of the usage of the IPTV service by the interaction manager
- Re-authentication of the interaction manager to the IPTV service

These requirements belong to the use case identified with number 4 in Figure 15:
- Authentication of the interaction manager to the IP telephony service
- Authorization of the interaction manager for a particular user at the IP telephony service
- Authorization of the interaction manager for the execution of a particular event at the IP telephony service
- Identification of the user at the IP telephony service by the interaction manager
- Accounting of the usage of the IP telephony service by the interaction manager
- Re-authentication of the interaction manager to the IP telephony service

5.4 Non-functional requirements

This section describes the non-functional requirements that will be used as criteria during the specification of the AAA support for interacting services from different realms. These requirements are derived from the scenario described in section 5.1.

- Limited interactions added between realms
- Maximal added delay to the IP telephony and IPTV service is 0,5 sec.
- Ability to recover from failure

5.5 Constraints

This section describes the constraints that will be used as criteria for the specification of the AAA support for interacting services from different realms. These constraints must be met by the specification. When a specification does not meet one of these requirements, is it not a valid solution. These constraints are derived from interviews with experts and subsequent analysis.

- Interaction is enabled using an interaction manager
- Different realms exist for the interaction service and interaction manager and the IPTV and IP telephony service
- More than one user can be identified and use the service
- The IPTV and IP telephony service can be used stand alone, without the interaction service
- The user is billed on a prepaid basis
- Extensions to the Diameter protocol must follow the rules as defined in the Diameter base protocol 'Approach to Extensibility'

5.6 Acceptance criteria

This section describes the acceptance criteria that are used to differentiate between the possible solutions. For each specification is measured to what extent it is compliant to the criteria described below. These criteria are derived from interviews with experts and subsequent analysis.

- Extendable, more operators and more applications can be added
- Level of trust between domains
- Minimal number of components that are needed to realize the architecture
- Minimal number of interactions/packets between realms
- Reuse of Diameter architecture in services that are based on IMS
- Easy identity management – exchanges of identities
- No/minimal alterations to the Diameter specification

5.7 Conclusion

This chapter describes one possible scenario with AAA for the given case. From the scenario several use cases are derived. The requirements are derived based on the scenario based requirements elicitation technique. The functional requirements consist of the authentication, authorization and accounting for the end-user and the

authentication, authorization and accounting between the different realms. The constraints are given which the solution has to meet and the acceptance criteria are described which are evaluated, in case of multiple possible solutions.

6 Solution phases

To map the solutions found in literature to the FoneFreez case, different phases are distinguished. The three types of authentication: device authentication, user authentication and message authentication can be mapped on different phases.

First trust between the different parties must be established and the connections made. Authentication between the devices is done in this phase. Then the user registers itself at the FoneFreez service. This all happens in the initialization phase. In this phase it must be checked whether the user is also registered at the television and telephony provider.

The next phase is the log-on phase. When the user comes home and wants to start the service, he logs-on at the FoneFreez service. The authentication and authorization must be done for the end-user, but also between the parties for that session.

After the log-on phase, the operational phase starts. Then the actual interaction service comes into place. In this phase re-authentication is possible and on message level authentication and authorization must be done. The usage of the service by the different parties must be registered in accounting records.

This chapter describes the different phases and issues that are of importance in every phase.

6.1 Initialization phase

The initialization phase consists of trust establishment; this is especially relevant when the entities reside in different domains. The starting point is where the physical connections already exist. But before it is possible to transport information over the connections, trust must be established. Several mechanisms for the creation of the trust relationship are available: by contract or protocols that help establish trust [HP, 2002b]. In this thesis protocols for dynamic trust establishment are considered, because this improves the flexibility of the solution.

After the trust is established registration starts, and the credentials of the user are entered in the AAA server of the interaction service. Diameter is not able to enter these credentials. The credentials must be entered using some other mechanism like: direct insert at the database by an administrator or web service that has the right to

insert entries in the database. When the user is registered at the interaction service it must be checked whether the user is also registered at the television and telephony providers. It is not desirable that the identity under which the user is known at the different parties is exchanged, so the exchange of identities must be done with respect for the privacy of the user. This management of identities is an issue during registration in the initialization phase.

The initialization phase is a one time event, which only takes place once for each user. For every user this phase must be passed through.

This section discusses how trust can be established between parties and how trust is established within Diameter. Furthermore the identity management issue is amplified, for verifying the registration of the user at the service providers.

6.1.1 Types of trust

There are different sorts of inter-realm trust: one-way trust, two-way trust and hierarchical trust.

In a one-way trust situation, a user from domain A can use the resources in domain B but not the other way around. Domain A gave its secret to domain B. For users from domain B is it not allowed to use resources from domain A, because domain B did not give its secret to domain A.

In a two-way trust situation resources can be used in both directions, because both domains know each others secret.

Hierarchical trust is only relevant when there is a realm hierarchy. Realm hierarchy is a tree of different realms as can be seen in Figure 16. For example realm A is tno.nl, realm B is dnv.tno.nl, realm C is ict.tno.nl and realm D is wir.ict.tno.nl.

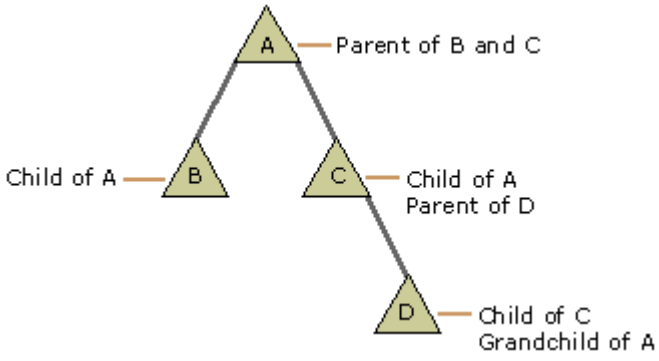

Figure 16 Realm hierarchy [Microsoft, 2006]

Hierarchical trust is a chain of trust that is built on the hierarchical relationships. If there is two-way trust between every realm in the example, then realm B can use resources from realm D based on the hierarchy.

Trust can be transitive. When there are three parties and two of them directly trust the same party (A trusts B, B trusts C), automatically an indirect trust relationship is established between A and C. Transitivity can be the case with one-way and two-way trust situations [Gleason, 2002].

6.1.2 Ways to establish trust

Trust can be established by arranging a contract between two parties. In this contract a shared secret can be agreed on, that is used to connect the servers with. But this shared secret can also be dynamically established.

Authentication can be done at different levels. PKI and Kerberos are lower level key- or certificate-exchanging mechanisms to establish trust [Oasis, 2004]. When trust is established and a connection is setup between the two parties, authentication and authorization is needed to negotiate which rights one domain has in the other domain.

Inter-realm authentication for building trust is realized using Kerberos [RFC 1510]. Kerberos is an entity authentication protocol, most used to mutually authenticate two servers that reside in different domains. It protects authentication information from threats like eavesdropping and uses a secret key encryption mechanism [Michiels, 2003]. Kerberos is used to provide inter-realm authentication between different servers, like the application server and the interaction manager. However it cannot be used to authenticate specific events, because this is authentication on a higher level and belongs to the operational phase.

Diameter uses Public Key Infrastructure (PKI) to dynamically establish trust between Diameter peers. This is shown in the Eduroam project [Eertink et al, 2005a]. Figure 17 shows two parties. The client's home realm is shown on the right, but he is now visiting at the realm on the left. He can log on, and the Diameter server of the visited domain has to contact the client's home AAA server for authentication. The DNS structure is in place to find the Diameter server belonging to the end-user. A third party certificate authority (CA) is used to establish the trust between the two

Diameter servers. The CA exchanges keying material with which the Diameter servers can built a trust relationship.

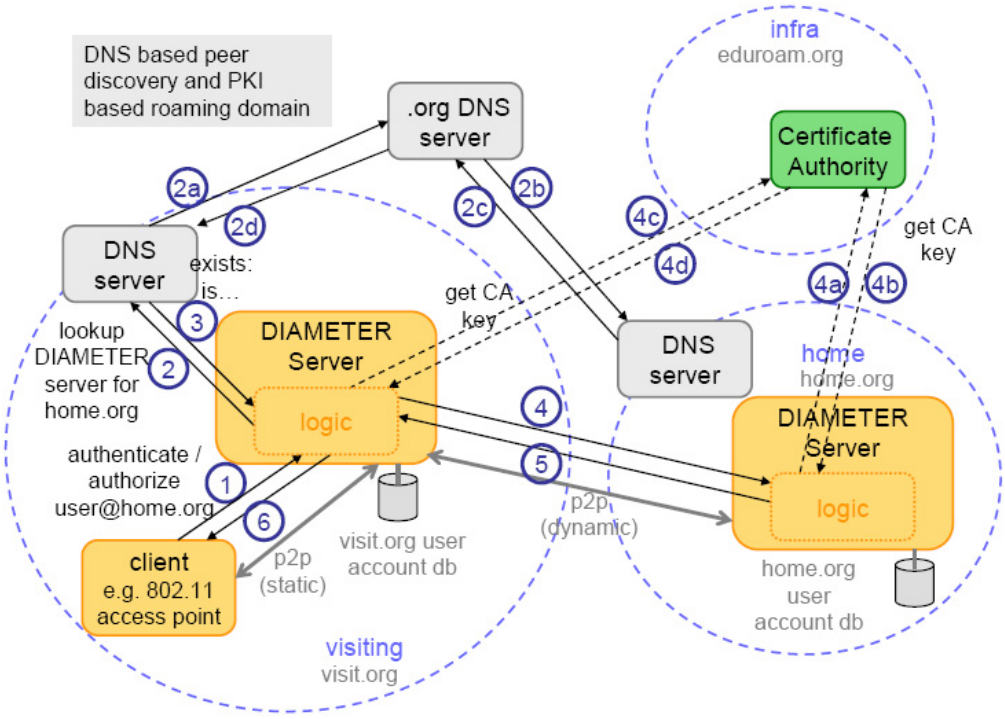

Figure 17 Eduroam Certificate authority [Eertink et al, 2005a]

The certificate authorities are especially for Diameter and must not function as certificate authorities for other purposes like web services. As stated in the RFC of Diameter: “In general, it is expected that those root CAs will be configured so as to reflect the business relationships between the organization hosting the Diameter peer and other organizations.” It is because of security considerations that the CA is especially for Diameter, so that the Diameter peer is configured to disallow connectivity with any arbitrary peer [RFC 3588].

In Diameter the lower level protocol TLS is used to exchange certificates and establish a lower level connection [RFC 3588].

6.1.3 Identity management

When the user registers with the interaction service, it must be verified that the same user is a user of the IPTV and IP telephony service. The user is known at the interaction service under a different identity than at the IPTV or IP telephony service. It is important that all entities know which user is meant, but with respect for the privacy of the user.

A body that is concerned with this issue is Liberty Alliance [Liberty, 2007]. The mission of the Liberty Alliance Project is to establish an open standard for federated network identity through open technical specifications. The theory about the federation framework is described in Appendix C.

Another initiative for providing exchange of identities in a secure manner is Shibboleth [Shibboleth, 2007]. This is a product from Internet2 [Internet2, 2007] and provides authorization in a distributed system. With Shibboleth federations can be built too.

To enable the verification of the identity of the user with the other services, a circle of trust between all parties must be established. The multiple identity providers in that case must be linked by another party which finds out by which identity the user is known at the different identity providers. It depends on the architecture chosen how this is arranged. The Diameter server can function as identity provider or as a service provider.

To find out if the user is registered at the service providers is difficult, because information about how the user is identified there must be available. This issue about the privacy of the user is important when dealing with different parties that know the user under different identities. This type of identity management problem is different from most problems, because there are three different identities of the user. In most problems the user is known under one identity in the network, and this identity has to be exchanged with a server from a different realm in a secure manner.

The user has a responsibility to prove his identity at the different parties. At registration the user can give information about how he is known at the service providers, but this is private information and vulnerable for misuse. Information about the user with which he can be identified at the service providers, are for example his telephone number and address. When the user enters another address, the television service is interrupted at another person's house. This type of misuse occurs when the user enters information at the FoneFreez application without verification of the correctness. This is why the user has the responsibility to authenticate itself to every party.

There are three different alternatives for the identity management issue. The liberty alliance federation framework can be used to exchange identities, but it is also

possible to extend the Diameter protocol. The current specification of the Diameter protocol is only for authentication, authorization and accounting. Identity management is important before authentication, and Diameter could play that role. The last alternative is tunneling an identity management protocol like SAML [Cover, 2007] over the Diameter protocol. The tunneling is described in [Dame, 2007]. Which alternative is best suited for this situation is not further researched.

6.2 Log-on phase

After registration the user can start the interaction service. First the user must be authenticated and authorized by the FoneFreez service. Authentication between the parties is done with the Kerberos protocol as described in the previous section. After that authorization must be arranged between the different parties. Also accounting plays a role in this phase.

6.2.1 User authentication and authorization

There are different authentication methods in Diameter for user authentication: username – password, challenge – response (CHAP) and the extensible authentication protocol (EAP) [RFC 3748]. The username – password option is the less secure solution and the EAP the most secure. The username – password and CHAP methods are included in de Diameter NASREQ application. The EAP method is separately specified in the Diameter EAP application as described in A.5.4.

For the user authentication no changes are needed in the Diameter NASREQ application.

Authorization is done with the same messages of the NASREQ application as authentication. Often authorization happens at the same time as authentication, in the same interaction.

6.2.2 User accounting

There are different types of accounting. It can be done by a flat rate model or by a usage based model. In case of a flat rate model, e.g. a fixed price per month is agreed on. The usage based model can be divided in static charging, with fixed rates for specific services, or dynamic charging, depending on the frequency of usage or time of day. The usage based model can use prepaid or postpaid charging.

Accounting for the interaction service is done based on a prepaid model. The Diameter Credit control application is used for authorizing credit for prepaid subscribers [RFC 4006]. The Diameter Credit control application supports both static and dynamic charging.

The AAA server that is used for authentication and authorization can be used for the accounting, but it is also possible to have an AAA server especially for accounting purposes.

Besides the Diameter Credit control entity, a party is needed that supports the transactions. The Diameter Credit control server can handle the prepaid model, but the Diameter protocol is not able to send money from one place to another. A third party must be involved to enable the transactions; have billing functionality. The three parties that have a relationship with the end-user also need a credit control server to enable real-time accounting.

Two types of messages are used to support user accounting; the ACR and ACA messages and the CCR and CCA messages. The first two messages, described in the Diameter base protocol are used to provide accounting data to the accounting server. The credit control request (CCR) is sent from the client to the server to request credit authorization for a given service. Before and during the service delivery, the credit authorization process takes place.

In the credit control application there are two approaches to perform the first interrogation: the credit control messages after the users authentication and authorization, or the credit control messages during the authentication and authorization. A reason to use the first alternative, shown in Figure 37 of subsection A.5.3, is the situation where authentication and authorization are decoupled from the actual service request; otherwise the second alternative should be used. The second alternative is shown in Figure 38 of the same subsection. In the situation of the case considered here the second alternative can be used, because the authentication and authorization of the user is combined with the service request.

For the accounting for the user, no changes are needed to the Diameter Credit control application. Because of choice for the second alternative, also CCR and CCA messages are transported between the application server and the AAA server. The Diameter client at the application server must support the credit control application.

6.2.3 Authorization between entities

In [RFC 2903] and [RFC 2904] different alternative designs are presented to handle AAA in multi-domain situations. These alternatives are shown in the next two figures. The concept is that the agent sequence, push sequence and pull sequence, as described in subsection 3.1.1, can be used interchangeably.

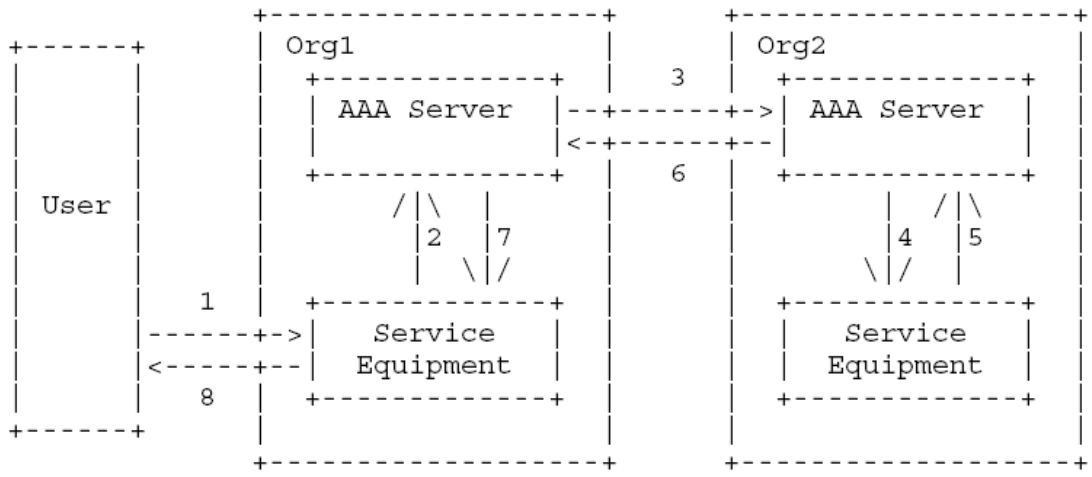

Figure 18 Multi-domain authorization I [RFC 2904]

Alternative I uses the agent sequence. The service equipment can be an application server. After the authorization is finished, the service can be delivered between the service equipments.

Where to find the AAA server of interaction manager is programmed in the application server. The trust between the AAA servers is established as described in subsection 6.1.2, with the help of certificate authorities.

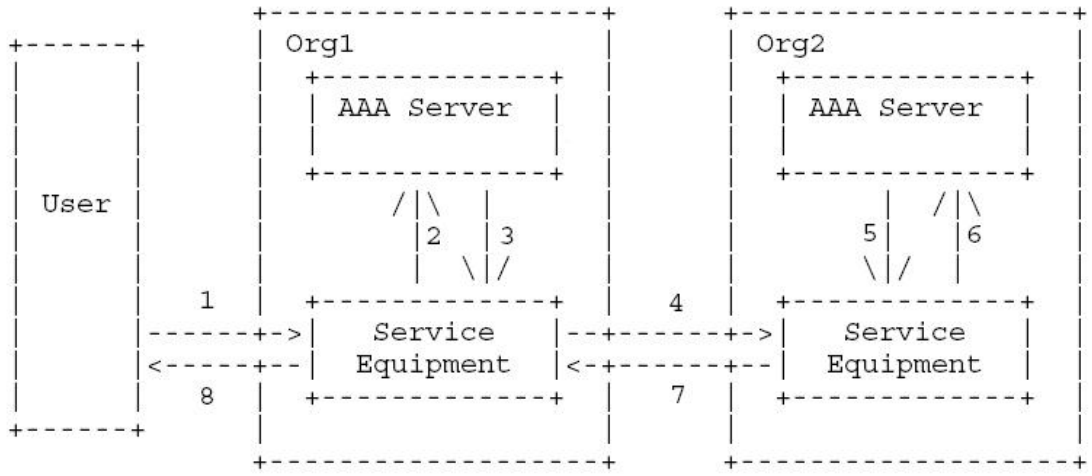

Figure 19 Multi-domain authorization II [self edited]

Alternative II uses the pull sequence. The service equipment requests authorization at the AAA server for its service. If the authorization is granted, the next service in the chain is asked for permission. In the case of FoneFreez, the service equipment is a

Diameter client for the next domain, e.g. the application server is a Diameter client for the broker. The broker maintains a list of application servers, as the application server maintains a list of end-users, and so on.

6.2.4 Accounting between entities

In the case that there are parties involved that have no direct relationship with the end-user, the revenue sharing comes in to play. Revenue sharing means that more than one party has to share the revenues of the end-user.

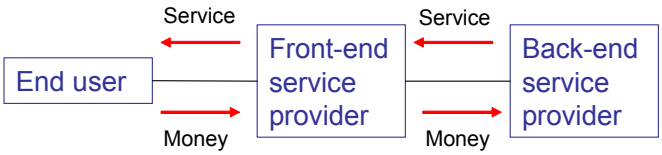

Figure 20 Revenue sharing

The back-end service provider delivers a service to the front-end service provider as shown in Figure 20. The front-end service provider adds extra service on top of that and delivers it to the end-user. The end-user pays the front-end service providers for the received service, and part of that payment is forwarded to the back-end service provider. The revenue sharing principle can be used in a chain of two different parties, but also in longer chains. At every party in the chain a part of the payment will be deducted from the amount before the money is forwarded to the next party.

It depends on the business relationship in which way the money flows. It is stated that the service providers and application service provider have their own relationship with the end-user and perform accounting as described in 6.2.2. This subsection describes the accounting between the service providers, broker and application service provider.

In the situation of accounting between the entities, metering is done at both sides or at the side of the party that collects the money; this depends on the level of trust.

In the FoneFreez case, the application service provider adds extra service and collects the money for this interaction service. The broker has to be paid for its intermediation services. This money will come from the application service provider. The service providers will not alter their rates because of that the interaction service is added, because as described before, the service providers can also run stand alone, without the interaction service. The broker will have to pay the service providers for

the effort they have to perform to send the events to the broker or to supply the interface to intervene with their services. The cash flow is shown in Figure 21.

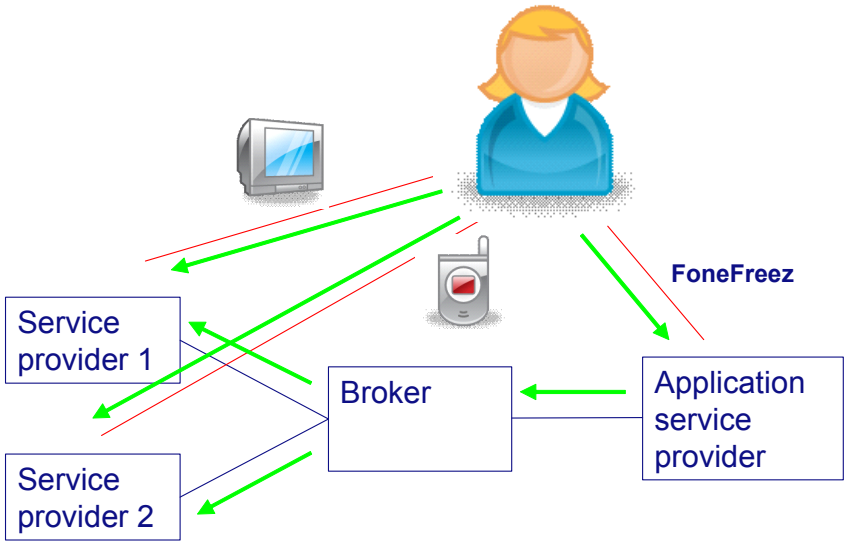

Figure 21 Cash flows

The cash flow will stay in the same direction as long as the number of relationships with the end-user is not changing; otherwise it is possible that the money flows in opposite direction. Even if the different roles are played by one actor, the internal cash flow will maintain the same direction. The internal cash flow is the money that flows in one and the same party between e.g. different business units.

There is no difference in metering if the charging is done prepaid, postpaid or with a flat fee, because in every situation it is desirable to meter the usage of the services at every entity. Only in the case of less than four different actors, the number of metering places can change. This is because it might not be necessary to meter at every entity if all the roles are played by the same actor.

6.3 Operational phase

During the operational phase the actions of the FoneFreez service take place as described in section 4.2. During the standard FoneFreez actions, authorization of the messages and re-authentication are important.

The authorization of the messages is mainly important in this case when the broker intervenes in the actions of the service provider providing television. In Diameter authorization is done with the same messages as authentication, but with another code that indicates that solely authorization is needed.

Re-authentication is relevant when for some time no interactions have occurred. Re-authentication takes time, and must be used sparingly, because otherwise too much delay is added to the FoneFreez service.

6.4 Conclusion

The phases that can be distinguished in the FoneFreez case are the initialization phase, log-on phase and operational phase. The important issues at the initialization phase are the trust establishment between the different parties and the exchange of the identities of the user with respect for privacy. In the log-on phase the user authentication, authorization and accounting, authorization between the entities and accounting between the different parties is important. In the operational phase authorization of messages and re-authentication are relevant topics.

The most important phase for the solution is the log-on phase. In this phase the Diameter protocol plays an important role, which is further described in the next chapter.

7 Alternative solutions

This chapter describes the specification suitable for the requirements stated in chapter 5 and the issues described in the previous chapter. The specification is a combination of an architecture and interactions between the entities of the architecture.

Two different architectures were found that both suit the requirements. The path to come to these two specifications is described in Appendix D using the methodology described in section 2.6.

In this chapter first the two designs are introduced. Then their different interactions are discussed and their differences are pointed out.

7.1 Two designs

The problem of this thesis is about service interaction from different realms. For the situation where all the business roles reside in different domains two alternatives are possible, following from the different architectures as described in subsection 6.2.3. The first alternative is shown in Figure 22.

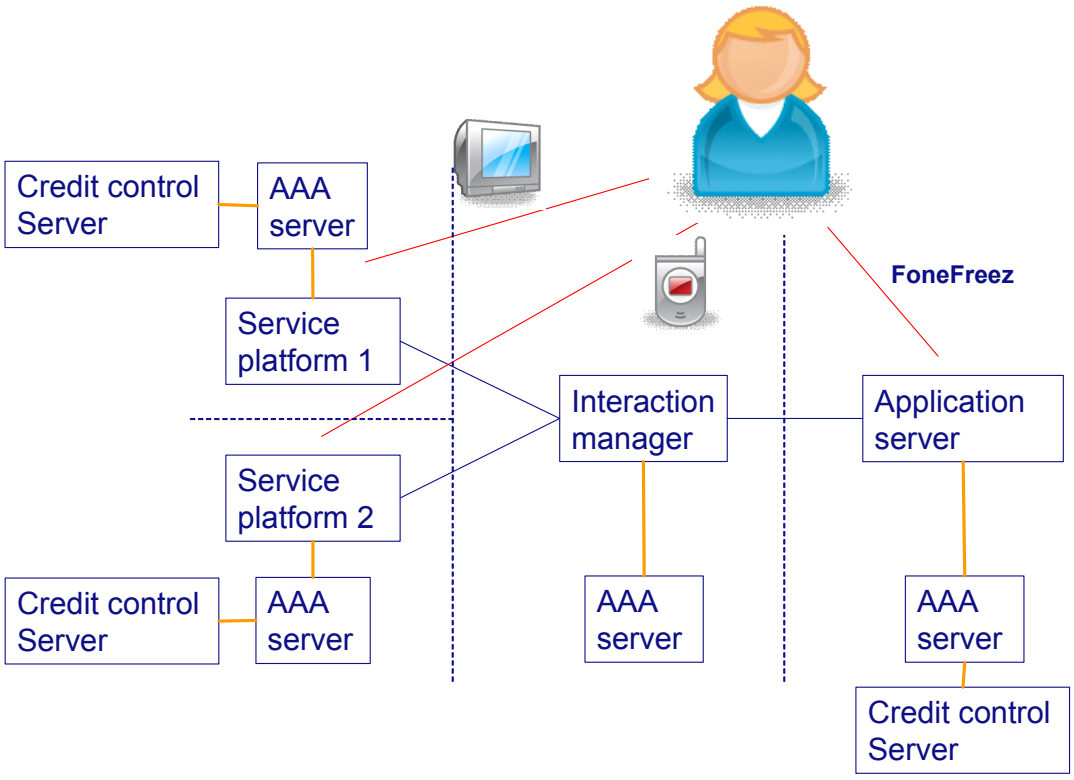

Figure 22 Hop-by-hop design

This alternative shows the realms with their own AAA server. The authentication between the domains is done in a hop-by-hop way, as explained later. Every domain must authorize when there is an incoming message, if this is an allowed action. The identity exchange takes place over the normal connections and at the service provider it is verified if the user is also registered there.

In Figure 23 the alternative is shown, where the AAA entities are interconnected.

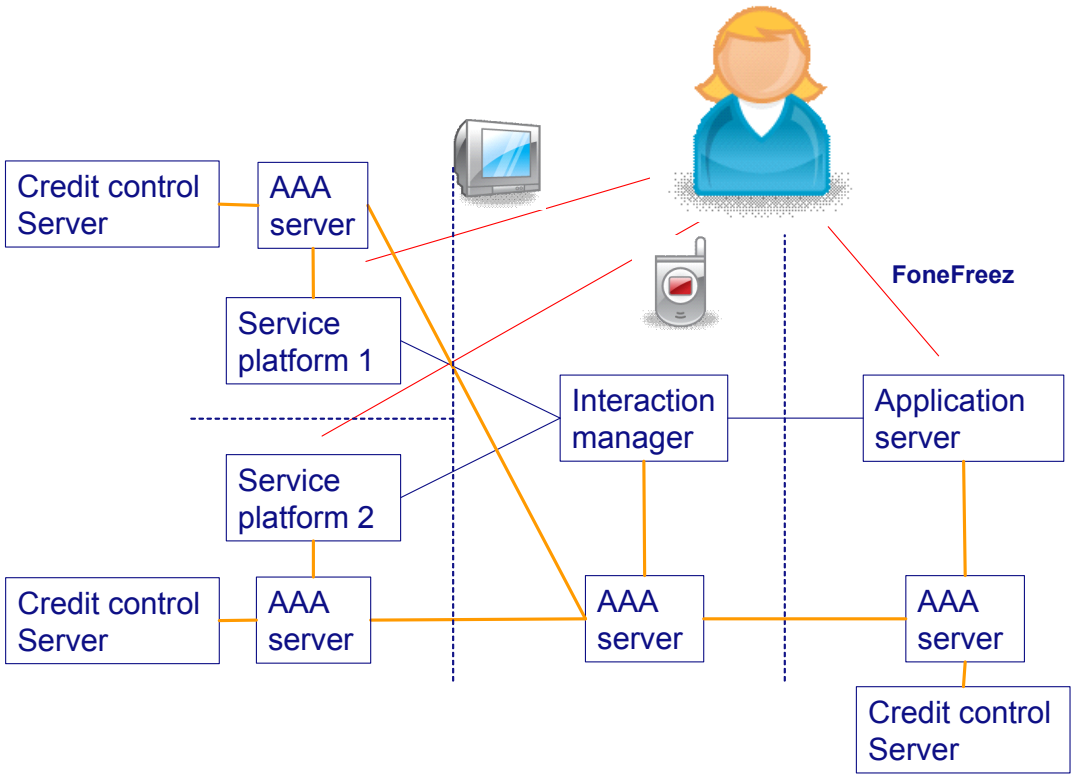

Figure 23 End-to-end design

The AAA entity in the broker's domain functions like a relay or proxy to enable the connection between the application service provider and the services. In this way the trust relations can be built between the different parties. Furthermore the identity of the user can be discussed, so they all know that they are talking about the same user. More about identity management for this alternative can be found in the next section. In this alternative the authentication can be done based on an end-to-end principle, also explained later.

The broker has no credit control server in this situation, this is because in the requirements is stated that only the user is accounted for on a pre-paid basis. The broker needs a third party to enable billing of the application service provider for the

usage of the interaction manager. The accounting for the usage is done at the AAA servers. The third party billing provider is left out.

7.2 Interactions belonging to the different designs

Due to the different interfaces designed in the previous section, the interactions that happen between the domains are also different. The differences are considered during the phases, as distinguished in chapter 6.

7.2.1 Initialization phase

In the first phase the trust is established. For trust establishment in the end-to-end design, certificate authorities (CA) are needed. The CA's are placed, reflecting the business relationship.

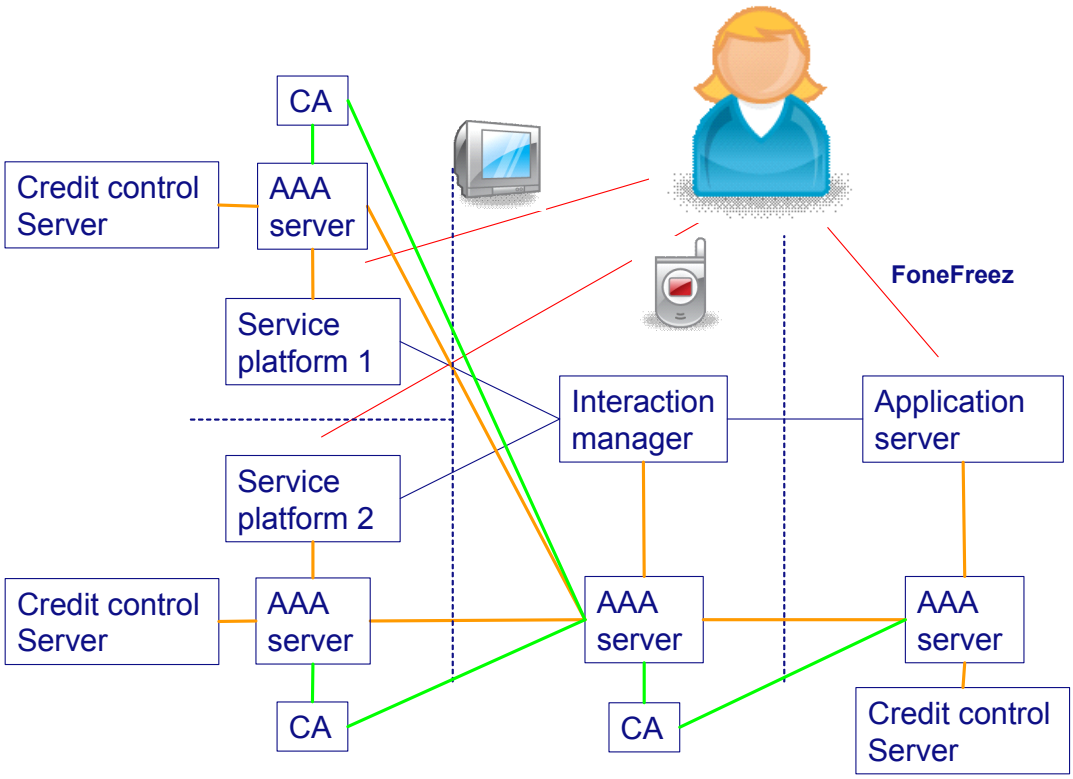

Figure 24 Certificate authorities

The only difference between Figure 24 and Figure 23 are the CAs and their connections in green. The trust between the entities is realized by trust transitivity (see subsection 6.1.1). Indirect trust relations are created between the entities due to the trust between the AAA server from the different realms and the trust of entities within the realm.

In the hop-by-hop design no certificate authorities are needed for the AAA architecture. The trust between the entities is realized in a way that is out of the scope of this thesis, but protocols like Kerberos can be used to establish trust. It is possible that there are also certificate authorities used for that realization of trust. These CA's should be others than the CA's used in the AAA architecture, because these CA's should be Diameter specific for safety reasons as described in subsection 6.1.2.

Identity management in both designs is a challenge as described in subsection 6.1.3. In the hop-by-hop design the interaction manager can provide the exchange of identities. In the end-to-end design the AAA architecture could play a role in exchange of identities. The identities can be stored in different places, the broker or application service provider. Who stores the identities depends on the business relations and the privacy information needed from the user (see Appendix C).

When the identities are exchanged, Diameter uses these identities to authenticate the user at the different domains. The Diameter nodes can change the contents of the username field in the Diameter messages. When an authentication request is sent to the service providers, the broker can change the username in the message, to the name under which the user is known at the service provider. This prevents the revelation of the identity of the user to other parties than the ones directly connected.

7.2.2 Log-on phase

The main difference between the designs is the level of intelligence needed in the entities. In the first design the intelligence lies within the entities, which only use the AAA servers for authentication and authorization of entities and users. In the second design the intelligence of the authentication lies within the AAA structure. This is where the hop-by-hop and end-to-end concepts come from.

In the hop-by-hop design shown in Figure 25, every domain uses its own AAA server for authentication and authorization of the end-user and other parties. Figure 65 of Appendix E shows the interactions between the different entities and the messages that are exchanged.

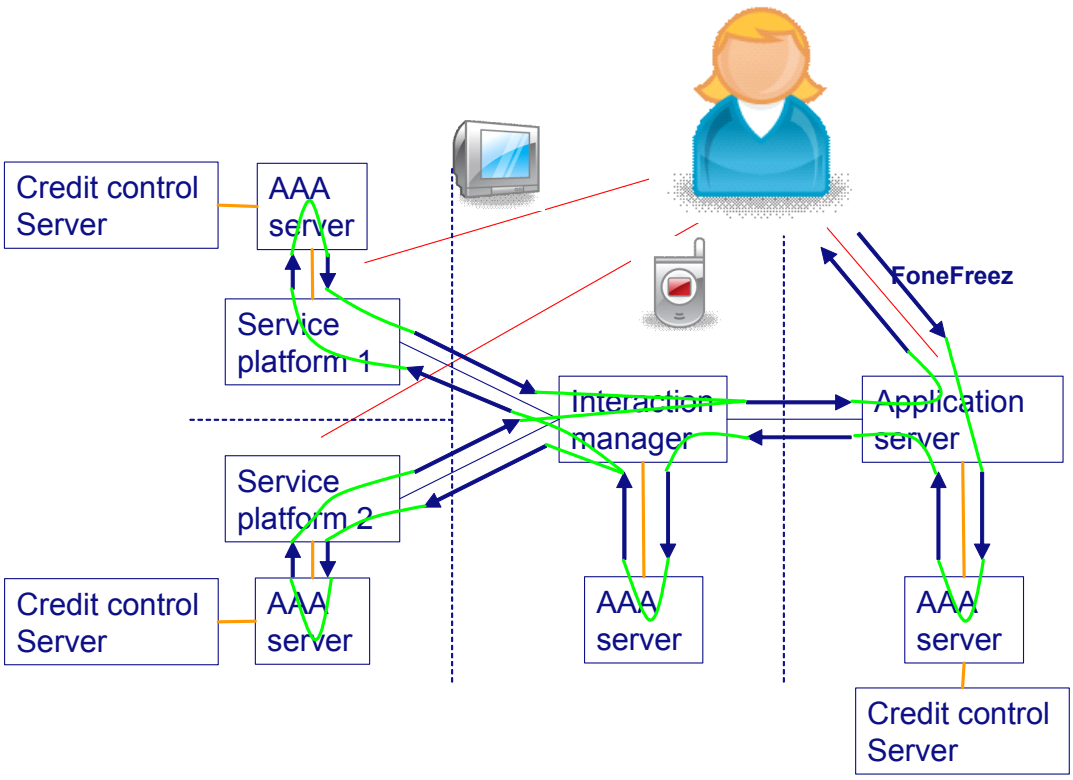

Figure 25 Hop-by-hop authentication

The user is authenticated at every party to make sure that the upcoming interventions are allowed for this particular user. This must be done every time the user logs-on to comply with the possibly updated policies. It is feasible that the service provider has new policies for intervention, for example with the television provider that only the pause functionality for that user is allowed.

In the end-to-end architecture shown in Figure 26, the AAA servers deal with the authentication procedure. Figure 64 of Appendix E shows the flow of messages in the end-to-end situation. These messages are all Diameter messages, except for the messages from and to the user.

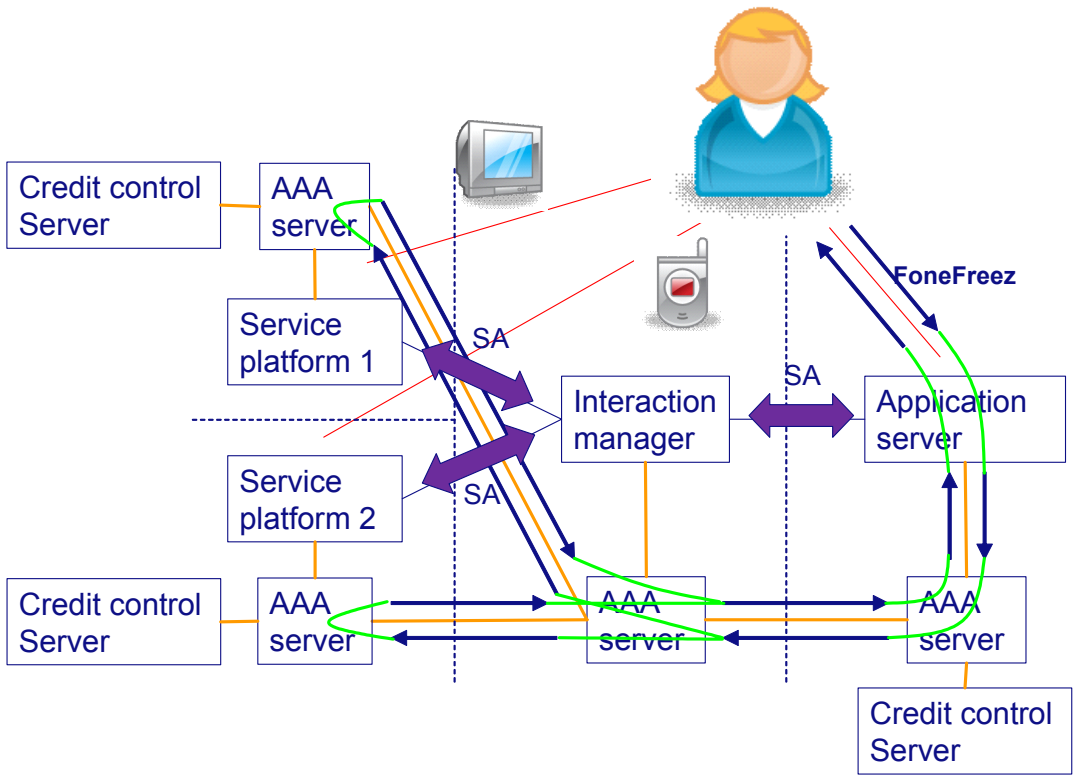

Figure 26 End-to-end authentication

When the messages between different entities are exchanged, a security association is realized between the different parties. A security association is the establishment of a shared secure connection. The security association is used in the Diameter mobile IPv4 application, as described in Appendix A.5.1. Because the AAA servers of both domains built a trust relationship and exchanged secrets, these can be used in a transitive way to enhance the trust between the entities.

In both designs the Diameter NASREQ application can be reused for authentication authorization and accounting. The entities in the network need the knowledge when to address the AAA server for some authentication and authorization. In the hop-by-hop design, the client side of the AAA needs a lot of modifications in the implementation to enable entity and user authentication, but also to decide to forward the message to the next realm. In the end-to-end design this intelligence lies within the AAA architecture. It is most likely that the application server sends two messages to the different service providers, which are transported over the AAA architecture.

7.2.3 Operational phase

Re-authentication and authorization of messages are relevant in this phase. Both can occur multiple times during a session.

Re-authentication of the user to the interaction service can be done every moment in time. It is feasible that the user is re-authenticated when the application server needs to make a decision about what to do. When using re-authentication here, it is possible that too much delay is added. Perhaps it is better to re-authenticate at a time when the application server is not processing other requests for that user.

The authorization of messages is most relevant when the broker intervenes at one of the service providers. When the interaction manager sends an event to the television service provider to pause the stream, the television service provider will check if this action is allowed for this user. If the telephony service provider sends an event to the interaction manager, the interaction manager decides what to do as part of the service, and authorization of the message is not needed.

7.3 Comparison of specifications

The specifications described above are compared using the acceptance criteria described in section 5.6. For the fulfillment of the criteria the extent of the fulfillment is expressed using --, -, + and ++, where -- is least fulfilled and ++ is most fulfilled.

Table 1 Comparison of designs

	Acceptance criteria	**Hop-by-hop**	**End-to-end**
1	Extendable, more operators and more applications can be added	+	++
2	Level of trust between domains	+	++
3	Minimal number of components that are needed to realize the architecture	+	-
4	Minimal number of interactions/packets between realms	-	+
5	Reuse of Diameter architecture in services that are based on IMS	-	--
6	Easy identity management – exchanges of identities	-	+
7	No/minimal alterations to the Diameter specification	++	+

The motivation for every criterion is discussed in the next subsections.

7.3.1 Extendibility

In the end-to-end specification the intelligence lies within the AAA architecture, whilst in the hop-by-hop specification it lies in the entities that provide the service. When more operators and applications are added in end-to-end situation, in comparison with the other specification, more interfaces need to be configured between the AAA entities. For every operator or application added in hop-by-hop, the interaction manager needs an update about how to forward AAA messages. The implementation of the hop-by-hop specification could be done in stages, where the implementation of the end-to-end specification must be done at once for all the parties involved for a particular user.

Both specifications are extendible but the end-to-end specification is the easiest to extend. This is because standard interfaces are used and no alterations to the interaction manager for the provisioning of AAA are needed.

7.3.2 Level of trust between domains

The trust is established in different ways. In the end-to-end specification trust transitivity is used to enhance the trust between the different domains. In hop-by-hop specification the trust between the entities is established by authentication of the entity in the other domain.

It is not explicitly that the level of trust is higher in one design or the other, but that it is easier realized. In the end-to-end specification, trust is enhanced during the authentication procedure and no other messages are needed for this. The hop-by-hop specification uses the Kerberos protocol to authenticate the devices of the different domains, but also needs messages for authentication of the end-user in the different domains.

7.3.3 Minimal number of components

The number of components is larger in the end-to-end design, because certificate authorities are needed especially for the Diameter architecture as described in subsection 7.2.1. They are not needed in the hop-by-hop design. In both situations AAA servers in every domain are needed, as well as Credit control servers at the parties that have relations with the end-user, for end-user prepaid credit authorization.

7.3.4 Minimal number of interactions/packets

The number of packets during the operational phase is the same in both designs. The end-to-end has the advantage that authentication and authorization between the entities can go in parallel with the other messages via the AAA architecture. This can be an advantage when the user is re-authenticated, that no delay is added to the regular process.

The number of interactions is different at the log-on phase as described in subsection 7.2.2. In the end-to-end specification the Diameter interface added, is a standard interface, while the interface in the hop-by-hop specification becomes more complex, because the complexity lies within the entities. Next to the eight Diameter messages needed in the hop-by-hop design, also six application specific messages are needed. In the end-to-end specification eight Diameter messages are sufficient.

Because the standardized interfaces are favorable, and the number of interactions for this specific case is less, the end-to-end specification performs better on this criterion.

7.3.5 Reuse in services based on IMS

In the hop-by-hop specification only one interface from the IMS network to the broker is needed. This connection can be established using security gateways as described in subsection 3.6.2. The IMS network can use the interaction manager from another domain and have its own AAA structure. This is not a currently used architecture in IMS, because of business reasons, but the possibility is present.

In the end-to-end specification two interfaces lead from the broker's domain to the IMS domain. A direct connection between AAA servers is established in this design. This is less in line with the IMS architecture as found in literature nowadays. In roaming situation there is no direct communication between the AAA entities from the different IMS networks. The call is first forwarded to the home realm, and there the Cx interface is used to authenticate the user [TS 24.228].

Both specifications do not fit seamlessly into the IMS network. The hop-by-hop specification has the best prospect, because the connection with the IMS architecture is possible in theory. The end-to-end specification has properties like the interconnection of the AAA servers, which are not in line with the IMS architecture.

7.3.6 Easy Identity management

Identity management is an issue in the phase that precedes the log-on phase where the given specification is important. The AAA architecture of the end-to-end design enables that the knowledge of the identities is stored in places where the username needs to be replaced. In the hop-by-hop design the identities must be replaced by the entities e.g. the interaction manager, so more intelligence must be added there.

7.3.7 No/minimal alterations to the Diameter specification

For the hop-by-hop specification no alterations to the Diameter specification are needed. The intelligence lies in the entities and they decide when to use the AAA server for authentication or authorization. The Diameter NASREQ application is used for authentication and authorization. The Diameter credit control application is used for accounting.

For the end-to-end specification also no alterations are necessary. For optimizing the authentication process it is possible that alterations to the AAA server of the broker are desirable. The authentication message from the application service provider could be split up in two different authentication messages for the service providers in the AAA server of the broker. This could be more favorable in stead of the application service provider sending two messages to the service providers. Alterations to the NASREQ application are not needed, but might be desirable.

7.4 Conclusion

This chapter presented two specifications for the problem. Both specifications have positive and negative sides, as discussed in the acceptance criteria. When looking solely at the scores, the end-to-end specification outperforms the hop-by-hop specification. Which specification is better actually depends on the area of application.

The end-to-end specification provides better trust and is suitable in situations where the parties use the service intensively and don't mind to be tightly connected. An example of such a situation could be an interaction in the telecom industry. It is common in this industry to have tight connections with the parties in the supply chain. This provides control for the service providers, which is favored by providers from the telecom industry.

The hop-by-hop specification is likely to be adopted by the parties that use the internet model. The hop-by-hop has less coupling of different interfaces and the independency of the different parties stays high. This causes providers to be less in control as in the end-to-end situation. Furthermore the rollout of this specification can be done in steps, where the end-to-end specification must be installed at once. The hop-by-hop specification has at the moment the best prospect to connect to an IMS network.

8 Fulfillment of the requirements

This chapter describes how the requirements as discussed in chapter 5 are fulfilled. These requirements are divided in four categories which correspond to the different sections in chapter 5.

The following table shows fulfillment of the requirements per category.

Table 2 Fulfillment of the requirements

Type of requirement	Fulfilled
Functional requirement	√
Non-functional requirement	√
Constraints	√

The requirements are presented with the exception of the acceptance criteria, because they are used in the previous chapter to compare the different designs. It is not mandatory that the acceptance criteria are fulfilled by the specifications. This is why they are not presented in this chapter.In section 7.3 is discussed to what extent the acceptation criteria are met for both specifications.

Both specifications, i.e. the hop-by-hop and end-to-end specification are compliant to all requirements. Every section discusses for each requirement how it is fulfilled by the specifications.

8.1 Functional requirements

These requirements belong to the use case identified with number 1 in Figure 15 of section 5.2:

- Identification of the interaction service user to the interaction service

The user is identified in Diameter by its NAI (section 3.5).

- Authentication of the interaction service user to the interaction service

The user is authenticated at the AAA server of the application service provider using the AAR/AAA messages of the NASREQ application.

- Authorization of the interaction service user to the interaction service

For the interaction service, the authorization happens with the authentication messages. The user is authorized to use the service after it is verified if the user also has a relationship with the IPTV and IP telephony provider. How to check the relationships with the end-user is not included in the specification.

- Accounting of the usage of the interaction service per user

The accounting of the usage of the service is done at the AAA server of the application service provider. The ACR and ACA messages are used to gather accounting information. The credit authorization on pre-paid basis, is done by the credit control server.

- Re-authentication of the user to the interaction service

The user can be re-authenticated by using the RAR/RAA messages, but it is not specified when the user is asked for re-authentication.

These requirements belong to the use case identified with number 2 in Figure 15:

- Authentication of the interaction service to the interaction manager

Authentication between the entities is done by Kerberos. The user authentication at the interaction manager is done using Diameter. The difference between hop-by-hop and end-to-end are explained in subsection 7.2.2.

- Authorization of the interaction service to the interaction manager

The authorization of the interaction service happens together with the authentication messages.

- Accounting of the usage of the interaction manager by the interaction service

The accounting is done at the AAA server of the broker. This server gathers accounting information from the interaction manager by using the ACR and ACA messages.

- Re-authentication of the interaction service to the interaction manager

Re-authentication can be done using RAR/RAA messages, but re-authentication must be used scarcely because of possible added delay to the interaction service.

These requirements belong to the use case identified with number 3 in Figure 15:

- Authentication of the interaction manager to the IPTV service

Authentication between the entities is done by Kerberos. The user authentication at the interaction manager is done using Diameter. The difference between hop-by-hop and end-to-end are explained in subsection 7.2.2.

- Authorization of the interaction manager for a particular user at the IPTV service

The authorization of the interaction service happens together with the authentication messages.

- Authorization of the interaction manager for the execution of a particular event at the IPTV service

The message that the interaction manager sends to the IPTV platform is authorized by the AAA server of the IPTV service provider. If the event is not allowed, the AAA

server declines the message and the event is not executed. The AAR and AAA messages are used for this purpose.
- Identification of the user at the IPTV service by the interaction manager
The user has an identity at the IPTV service but the interaction manager does not know this identity. It is not specified by which identifier the user is identified at the IPTV service. The interaction manager must use one of the identity exchange mechanisms described in subsection 6.1.3.
- Accounting of the usage of the IPTV service by the interaction manager
The accounting is done by the AAA server of the IPTV service provider. The IPTV platform meters the usage by the interaction manager and the accounting information is passed on to the AAA server using the ACR/AAA messages. The user accounting is done in de same way as described at the interaction service.
- Re-authentication of the interaction manager to the IPTV service
Re-authentication can be done using RAR/RAA messages, but re-authentication must be used scarcely because of possible added delay to the interaction service.

These requirements belong to the use case identified with number 4 in Figure 15:
- Authentication of the interaction manager to the IP telephony service
Authentication between the entities is done by Kerberos. The user authentication at the interaction manager is done using Diameter. The difference between hop-by-hop and end-to-end are explained in subsection 7.2.2.
- Authorization of the interaction manager for a particular user at the IP telephony service
The authorization of the interaction service happens together with the authentication messages.
- Authorization of the interaction manager for the execution of a particular event at the IP telephony service
This situation is not part of the scenario discussed in this thesis. Because of the symmetry of the design, the requirement should also be valid for the IP telephony provider, as it is valid for the IPTV provider as described above.
- Identification of the user at the IP telephony service by the interaction manager
The user has an identity at the IP telephony service but the interaction manager does not know this identity. The interaction manager must use one of the identity exchange mechanisms described in subsection 6.1.3. It is possible that the phone number of the user is used to negotiate about the user's identity, but this is not specified.
- Accounting of the usage of the IP telephony service by the interaction manager

The accounting is done by the AAA server of the IP telephony service provider. The IP telephony platform meters the usage by the interaction manager and the accounting information is passed on to the AAA server using the ACR/AAA messages. The user accounting is done in de same way as described at the interaction service.

- Re-authentication of the interaction manager to the IP telephony service

Re-authentication can be done using RAR/RAA messages, but re-authentication must be used scarcely because of possible added delay to the interaction service.

8.2 Non-functional requirements

- Limited interactions added between realms

The number of AAA messages after initialization and registration is limited. During the operational phase the number of interactions between the realms has not significantly increased.

- Maximal added delay to the IP telephony and IPTV service is 0,5 sec.

During the operational phase re-authentication is possible but not necessary, because the level of trust that exists after the registration phase is high enough. Furthermore it is presumed that the time that the user uses the interaction service is limited within a 24 hour period. When the user ends the session and connects again, the service is started again and authentication is done. Because no re-authentication during the operational phase is needed, no more delay is added to the current case.

- Ability to recover from failure

Diameter is a protocol that is built to be able to recover from failure. As long as the original service was able to recover from failure, the new specification can also recover after for example a sudden disconnection.

8.3 Constraints

- Interaction is enabled using an interaction manager

The interaction manager is used to provide the interaction functionality. All application service providers and service providers are connected to the broker, which is implemented by the interaction manager.

- Different realms exist for the interaction service and interaction manager and the IPTV and IP telephony service

There are four different realms present in the specification; for the application service provider, which provides the interaction service; for the broker which is implemented by the interaction manager; and for both service providers which provide the IPTV and IP telephony service.

- More than one user can be identified and use the service

Diameter is a protocol that scales well according to the specification of the protocol. A large amount of users can use the Diameter service simultaneously. In this specification also a large amount of users can be authenticated and authorized.

- The IPTV and IP telephony service can be used stand alone, without the interaction service

The specification is designed with in mind the stand alone ability of the service providers. The IPTV and IP telephony services have their own AAA entity, which can perform user authentication, authorization and accounting. At both the IPTV and IP telephony service an interface to the broker is added, but the usage of this interface is not crucial to the delivery of the IPTV and IP telephony services to the end-user.

- The user is billed on a prepaid basis

For this functionality the Credit control servers are added in the specification. The cash flow needed for this functionality is not delivered by Diameter, and a third party should be involved to add billing functionality.

- Extensions to the Diameter protocol must follow the rules as defined in the Diameter base protocol 'Approach to Extensibility'

No extensions of the Diameter protocol are needed.

8.4 Conclusion

All requirements are fulfilled by both specifications. Not every requirement could be met using Diameter because Diameter is for end-user AAA. Some of the requirements are inter-domain AAA for different entities, for which Kerberos is suitable.

9 Conclusion and future work

This chapter discusses the conclusions and future work. First the results of the thesis are summarized, followed by the discussion. The research question is answered in the conclusion section. Finally the possible future work, which follows from this thesis, is considered.

9.1 Results

This section discusses the answers on the research sub questions, which are described in section 1.1.

- Which requirements should be applied?

The requirements on the solution of this thesis are divided in four categories: functional requirements, non-functional requirements, constraints and acceptance criteria. The functional and non-functional requirements describe the AAA behavior of the specification. The constraints are showstoppers when they are not met. The acceptance criteria are used to differentiate between possible solutions. The most important functional requirements are end-user AAA and AAA between entities of different realms.

- What does the case of the project look like and which IMS components are used?

The FoneFreez project of TNO-ICT is used as case study. In this project an added value service is built that provides service interaction from different realms. The interaction consists of an IPTV and IP telephony provider that intervene in each others behavior, managed by a service capability interaction manager (SCIM). This case uses the SCIM, application server and CSCF IMS components. The SCIM IMS component is renounced, because of the lack of specification by 3GPP on the SCIM. Instead of the SCIM an interaction manager is used to provide the interaction functionality.

- What are the currently available Diameter based architectures for interacting services from different realms?

A literature study is done to find generic AAA architectures and Diameter specific architectures. The general architectures and specific Diameter knowledge is used to elaborate on the subjects that are important when developing an AAA specification for interacting services from different realms. These subjects are different types of trust establishment, authorization architectures, authentication models and

accounting models. The accounting models are derived from expert opinions, the other subjects from literature.

- How can the AAA architectures be mapped on the given case?

The AAA architectures are mapped on the case by defining different phases, initialization, log-on and operational, and their subsequent issues. The main AAA architectures used are two inter-domain authorization models. These two possible approaches for authorization between entities, led to two different specifications for the given case.

- Which specification can be derived?

Two specifications are derived: hop-by-hop and end-to-end. These are compared using the acceptance criteria. The best specification for the case cannot be given simply; this depends on the application area. In the end-to-end specification the different parties are tightly connected, and in de hop-by-hop specification a loose coupling is used.

- Does the specification fulfill the requirements of this study?

The fulfillment of the requirements is discussed in chapter 8. As described in this chapter the functional and non-functional requirements are met and the specification is compliant with the constraints.

9.2 Discussion

During the thesis, criticism on the FoneFreez project was heard from different experts. The feasibility of the service interaction in the network is questionable. There are business models that prefer control over the service interaction, for whom the FoneFreez concept is desirable. On the other hand, there are a lot of reasons why service interaction at the network level as seen in FoneFreez, is not in the best interest of the user. A model where a packager bundles different services and provides service interaction for the user is more user-friendly. Identity management is different in that case, and user-friendly solutions like single-sign-on can be provided by the identity federation.

During the thesis the area of application for Diameter was found. The most important finding for this thesis was that Diameter provides authentication, authorization and accounting for end-users, this in contrast to authentication, authorization and accounting between entities from different parties. Kerberos is a better suited

protocol to provide this kind of AAA. It is not desirable to provide this type of AAA with Diameter, because this is not the purpose of the protocol.

The flexibility of the Diameter protocol is large, which is also its biggest disadvantage. Due to the flexibility, the protocol can be used for a large range of applications, but it is less likely to become a standard in that way. 3GPP uses the flexibility of the protocol to its full extent, which is not always in line with the purpose of the Diameter protocol in my opinion. For example, the following is stated in the specification of the Rx interface: "Existing Diameter command codes from the Diameter base protocol RFC 3588 and the NASREQ Diameter application are used with the Rx specific AVPs. An Rx specific Auth-Application id is used together with the command code to identify the Rx messages. NOTE: The notion of NAS (Network Access Server) is not used here, NASREQ is just used for protocol purposes, not for its functional meaning." [TS 29.211]

In this thesis only the FoneFreez case is studied. Generalizing from one case is not advisable, but some lessons learned in this thesis can be used in more general ways. For example knowledge was gained during the analysis of the Diameter protocol, the application area's of the protocol and limitations were found. Furthermore the multi-domain AAA architectures with Diameter can be applied in several situations outside the FoneFreez project.

The verification of the requirements in this thesis is not done by testing or by using formal methods. The requirements are inspected by experts. This is not a very reproducible method. To ensure the verification of the requirements a proof of concept should be built.

9.3 Conclusion

The research question that is posed by this thesis is:

> Is it possible to reuse and/or extend the Diameter protocol specification, according to the rules defined in the Diameter base protocol, to provide AAA for interacting services from different realms in IMS like architectures?

The answer to this question is that the Diameter protocol specification can be reused in a way that it provides AAA for interacting services from different realms in IMS like architectures. The specifications given in this thesis provide AAA following the

rules defined in the Diameter base protocol, the Diameter NASREQ application and the Diameter credit control application, with the help of Kerberos. The specifications can be used in IMS like and non-IMS architectures and is suited to deliver AAA in both situations. One of the two given specifications (hop-by-hop specification) is currently best suited to be connected to IMS architectures. The usage of the other specification (end-to-end specification) in IMS networks is for further study which is described in the next section.

As described in this thesis is it possible to provide AAA with Diameter to the specific case of FoneFreez. In this project a demonstrator of service interaction in multi-domain is built. On the other hand, this specification is also suited to deliver AAA in a more general inter-domain service interaction architecture.

The objective of the study is reached, to find out how Diameter can add AAA to interacting services from different realms. The solution in this thesis provides authentication, authorization and accounting of the end-users and between entities from different domains. Service interaction can now be delivered with a multiple party architecture in a secure manner.

Diameter is a very flexible protocol, which is designed to function in a multi realm environment. The applications that are standardized are suitable for many AAA problems. How Diameter functions in one of these problems is demonstrated in this thesis. As shown in the analyses given in Appendix F, Diameter will become widely adopted with the success of IMS, and is the next AAA protocol to follow up RADIUS.

9.4 Future work

After this thesis a proof of concept should be done. Due to time constraints the objective of this thesis was to deliver a specification. The implementation of this specification should be done to prove that the specification actually fulfills the requirements and delivers the desired functionality.

Quality of service (QoS) is the control of specific resources, like bandwidth or processing power. This can be done by the Diameter quality of service application. Because the application is not yet standardized, quality of service was not considered in this thesis. For this particular case of FoneFreez, quality of service was not that interesting, but other service interaction applications could benefit from QoS support by the Diameter protocol. For example when service interaction is bound to a

particular bandwidth or level of service, which is often used for important telephony signals.

The full process of identity management was out of scope of this thesis. With the exchange of identities privacy is a very important aspect. Because privacy issues are a whole other research area, this was not considered in this thesis. The exchange of identities of the user for the FoneFreez project is a necessity. This should be studied further before the FoneFreez concept can be commercially exploited.

The third party billing provider was not considered in this thesis. How this billing provider is positioned and what functionality it has to have, should be studied.

The end-to-end specification could be used in IMS networks. If multiple IMS cores are connected and several identities of the user exist, the connections of HSS's can provide added value. The application of the end-to-end specification in IMS networks is a topic that should be further explored.

References

The references are divided in normative and informative references. The normative references are for finding the place where the information of that part of the thesis was retrieved. The informative references just refer to places where the reader can find more information, when interested in the subject.

Normative references

[AAAARCH, 2004] John Vollbrecht and Cees de Laat, *Authorization, Authentication and Accounting ARCHitecture research group*, closed October 18, 2004, www.AAAARCH.org

[Aepona, 2007] Aepona Ltd, *Universal Service Platform, Service Interaction*, 2007, www.aepona.com, http://www.aepona.com/about/proposition_images/service_interaction.pdf

[Aha, 1992] David W. Aha, *Generalizing from a case studies: a case study*, Research center, RMI Group, Applied Physics Laboratory, The John Hopkins University, 1992

[Alfano, 2006] F. Alfano, *Diameter Quality of Service Application*, IETF draft, work in progress, expires April 23, 2007, draft-tschofenig-dime-diameter-qos-01.txt

[Bertrand, 2006] Gilles Bertrand, *The IP multimedia subsystem, an overview*, GET/ENST Bretagne, December 2006, http://www.rennes.enst-bretagne.fr/~gbertran/files/IMS_an_overview.pdf

[Braun et al, 2001] David Braun, Jeff Sivils, Alex Shapiro, Jerry Versteegh, *Unified Modeling Language (UML) Tutorial*, Kennesaw State University, Spring 2001, http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm

[Chaouchi et al, 2002] H. Chaouchi, G. Pujolle, H. Afifi, Kim HahnSang, *A Trial towards Unifying Control Protocol: COPS versus Radius/DIAMETER and Mobile IP*, appears in the proceedings of Mobile and Wireless Communications Network 2002

[Cisco, 2001] Cisco Systems Inc., *Authentication, authorization and accounting,* 2001,

http://www.cisco.com/en/US/products/ps6638/products_data_sheet09186a00804fe332.html

[Forsgren et al, 2001] Olov Forsgren, Ulrich Tucholke, Sebastien Levy, Stephan Brunnessaux, *Report on electronic democracy projects, legal issues of Internet voting and users (i.e. voters and authorities representatives) requirements analysis, 3 List of requirements,* Report for European project Cybervote, IST-1999-20338, 2001, *http*://www.eucybervote.org/Reports/KISTA-WP2-D4V3-v1.0-02.htm

[Dame, 2007] DAMe Project, *Design alternatives for the DAMe project: A proposal for extending the eduroam infrastructure with authorization mechanisms,* University of Murcia, January 2007, *http*://dame.inf.um.es/htmldocs/design_alternatives.htm

[Das et al, 2004] Manik Lal Das, Ashutosh Saxena, Ved P. Gulati, *A Dynamic ID-based Remote User Authentication Scheme*, Consumer Electronics, IEEE Transactions on Volume 50, Issue 2, May 2004

[Dodd, 2003] Martin Dodd, *Transaction Log and Data Exchange Standards, The requirements document,* presentation from UNFCCC Pre-Sessional Consultations on Registries, Bonn, Germany, 2 June 2003, http://unfccc.int/files/meetings/workshops/other_meetings/application/vnd.ms-powerpoint/mad020603.pps

[Eertink et al, 2005a] Henk Eertink, Arjan Peddemors, Roy Arends, Remco Poortinga, Rok Papez, Klaas Wierenga, *Onderzoeksrapport over Diameter, Inter-Domain Roaming in eduroam-ng: An Overview and Comparison of Advanced Roaming Protocols*, Surfnet and Telematica Instituut, December 25, 2005, http://www.surfnet.nl/publicaties/surfworks2005/indi-2005-012-35.pdf

[Eertink et al, 2005b] Henk Eertink, Arjan Peddemors, Roy Arends, Klaas Wierenga, *Combining RADIUS with Secure DNS for Dynamic Trust Establishment between Domains*, presentation Terena network conference 2005, Poland, http://www.lab.telin.nl/~arjan/pub/tnc05-ea-eertink.pdf

[Ericsson, 2004] Ericsson, *IMS – IP multimedia subsystem, the value of using the IMS architecture,* white paper, October 2004,

http://www.ericsson.com/technology/whitepapers/ims_ip_multimedia_subsystem.pdf

[Fried et al, 2006] Jeff Fried, Duane Sword, *Making IMS Work: Current Realities, Challenges And Successes*, Empirix Inc., May 1, 2006, http://www.bcr.com/architecture/ip_networking/making_ims_work_20060501119 9.htm

[Gleason, 2002] Bernard W. Gleason, *Call to Action!*, presentation at the JA-SIG uPortal conference, Vancouver, British Columbia, June 10, 2002, http://web.princeton.edu/sites/isapps/jasig/2002summerVancouver/Presentations/call%20to%20action/img19.html

[Gustafson et al, 2001] Ulf Gustafson, Jan Forslöw, *Network design with Mobile IP,* INET 2001 proceedings, Stockholm, Sweden, http://www.isoc.org/inet2001/CD_proceedings/T40/inet_T40.htm

[Hinard et al, 2006] Yoann Hinard, H. Bettahar, Y. Challal, A. Bouabdallah, *AAA based security architecture for multicast content distribution*, Compiegne University of Technology, Heudiasyc lab. France IEEE proceeding ISCN'06, June 2006

[HP, 2002] Hewlett-Packerd Company, *Introduction to Diameter*, white paper, September 2002, http://docs.hp.com/en/T1428-90011/T1428-90011.pdf

[HP, 2002b] HP*, Installing, Configuring and Administering the Kerberos Server V 2.0 on HP-UX 11i,* HP 9000 Networking, Edition 2, Hewlett-Packard Company, 2002, http://docs.hp.com/en/T1417-90003/T1417-90003.pdf

[IANA, 2006] IANA, *AAA parameters*, December 1, 2006, http://www.iana.org/assignments/aaa-parameters

[IBM, 2006] Jeffrey Liu, Steven Jiang, Hicks Lin, *Introduction to Diameter, Get the next generation AAA protocol*, IBM, 24 January 2006, http://www-128.ibm.com/developerworks/library/wi-diameter/index.html

[ICOM, 2006] Description project 1, *ICOM 6006: Distributed Systems, Project 1 Description*, January 2006, http://ece.uprm.edu/~wrivera/Distributed_Systems/Project1.pdf

[Liberty, 2005] Liberty Alliance, *Liberty ID-FF Architecture Overview*, Version: 1.2-errata-v1.0, 2005, http://www.projectliberty.org/liberty/content/download/318/2366/file/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf

[Mayer, 2006] Georg Mayer, Miikka Poikselkä, Hisham Khartabil, Aki Niemi, *The IMS, IP Multimedia Concepts and Services*, Second edition, John Wiley and Sons Ltd, 2006

[Microsoft, 2006] Microsoft, *Active Directory Domain Hierarchy,* part of Windows 2000 Server Resource Kit, Microsoft Corporation, 2006, http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib/dsbb_act_zjfb.mspx?mfr=true

[Michiels, 2003] E.F. Michiels, *Telematics systems security*, lecture notes, University of Twente, November 2003

[Nakhjiri et al, 2005] Madjid Nakhjiri and Mahsa Nkhjiri, *AAA and Network Security for Mobile Access*, Wiley, 2005

[Oasis, 2004] Oasis, *Trust Models Guidelines*, for SSTC by the Liberty Alliance, February 2, 2004, work in progress, http://www.oasis-open.org/committees/download.php/6158/sstc-saml-trustmodels-2.0-draft-01.pdf

[Perry et al, 2004] Dewayne E. Perry, Susan Elliott Sim, Steve Easterbook, *Case studies for Software Engineers*, ICSE 2004 tutorial, 2004

[RFC 2748] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., and A. Sastry, *The COPS (Common Open Policy Service) Protocol*, RFC 2748, January 2000

[RFC 2904] Vollbrecht, J., Calhoun, P., Farrell, S., Gommans, L., Gross, G., de Bruijn, B., de Laat, C., Holdrege, M., and D. Spence, *AAA Authorization Framework*, RFC 2904, August 2000

[RFC 3127] Mitton, D., St.Johns, M., Barkley, S., Nelson, D., Patil, B., Stevens, M., and B. Wolff, *Authentication, Authorization, and Accounting: Protocol Evaluation*, RFC 3127, June 2001

[RFC 3539] Aboba, B. and J. Wood, *Authentication, Authorization and Accounting (AAA) Transport Profile*, RFC 3539, June 2003

[RFC 3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, *Diameter Base Protocol*, RFC 3588, September 2003

[RFC 4004] Calhoun, P., Johansson, T., Perkins, C., Hiller, T., and P. McCann, *Diameter Mobile IPv4 Application*, RFC 4004, August 2005

[RFC 4005] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, *Diameter Network Access Server Application*, RFC 4005, August 2005

[RFC 4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, *Diameter Credit-Control Application*, RFC 4006, August 2005

[RFC 4072] Eronen, P., Hiller, T., and G. Zorn, *Diameter Extensible Authentication Protocol (EAP) Application*, RFC 4072, August 2005

[RFC 4740] Garcia-Martin, M., Belinchon, M., Pallares-Lopez, M., Canales-Valenzuela, C., and K. Tammi, *Diameter Session Initiation Protocol (SIP) Application*, RFC 4740, November 2006

[Savola, 2003] P. Savola, *Mobility support in RADIUS and Diameter*, Helsinki University of Technology, May 28, 2003, http://www.tml.tkk.fi/Studies/T-110.551/2003/papers/13.pdf

[Sharp et al, 2007] Helen Sharp, Yvonne Rogers, Jenny Preece, *Interaction Design: Beyond Human-Computer Interaction*, 2nd ed. John Wiley & Sons Ltd., 2007

[Sher et al, 2006] Muhammad Sher, Thomas Magedanz, *Developing Network Domain Security (NDS), Model for IP Multimedia Subsystem (IMS)*, Journal of Networks, vol. 1, no. 6, November/December 2006

[Sultan, 2006] Alain Sultan, *La normalization des réseaux de prochaine génération (NGN)*, February 1, 2006, http://www.etsi.org/ictroadshow/Presentations/5-%20v5_Sultan_Rennes.ppt

[Sweers et al, 2007] Bart-Jan Sweers, Josie Scarr, Richard Tee, *Key success factors of telecom standards*, TNO whitepaper, January 29, 2007

[Thales, 2006] Thales e-security, *Advances authentication*, whitepaper 2006, http://www.thales-esecurity.com/Whitepapers/documents/Advanced_Authentication_.pdf

[TS 24.228] 3GPP, *TS 24.228, Technical Specification Group Core Network and Terminals; Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 5)*, V5.15.0, September, 2006

[TS 29.211] 3GPP, *TS 29.211, Technical Specification Group Core Network and Terminals; Rx Interface and Rx/Gx signalling flows (Release 6),* V6.3.0, December 2005

[Vishnu, 2006] R. Vishnu, *DHCP Diameter Application*, IETF draft, work in progress, expires March 4, 2007, draft-ram-dhc-dhcpv6-diam-app-01.txt

[Whittle et al, 2004] Jon Whittle and Ingolf H. Krüger, *A Methodology for Scenario-Based Requirements Capture*, Proceedings of the ICSE 2004 Workshop on Scenarios and State Machines (SCESM), 2004, http://www-cse.ucsd.edu/~ikrueger/publications/WhittleKrueger_SCESM04_final.pdf

[Wieringa et al, 2003] Roel Wieringa, Hans Heerkens, *Requirements Engineering as Problem Analysis: Methodology and Guidelines*, Technical report, University of Twente, July 14, 2003

Informative references

[3GPP, 2007] website 3rd Generation Partnership Project, visited January 2007, www.3GPP.org

[Cover, 2007] Robin Cover, *Security Assertion Markup Language,* Technology report, March 16, 2007, *http*://xml.coverpages.org/saml.html

[Dime, 2007] IETF Diameter Maintenance and Extensions (Dime) workgroup, *Dime Status Pages*, visited January 2007, http://tools.ietf.org/wg/dime/

[Fenner, 2007] Bill Fenner, *Draft/RFC Dependency Tool*, visited January 2007, http://www.rtg.ietf.org/~fenner/ietf/deps/index.cgi

[ETSI, 2007] website European Telecommunications Standards Institute, visited January 2007, www.etsi.org

[IETF, 2007] website International Engineering Task Force, visited January 2007, www.ietf.org

[Internet2, 2007] website Internet2, visited April 2007, www.internet2.edu

[ITU, 2007] website International Telecommunication Union, visited January 2007, www.itu.int

[Liberty, 2007] website Liberty Alliance project, visited February 2007, www.projectliberty.org

[RFC 768] Postel, J., *User Datagram Protocol*, STD 6, RFC 768, August 1980

[RFC 793] Postel, J., *Transmission Control Protocol*, STD 7, RFC 793, September 1981

[RFC 1034] Mockapetris, P., *Domain names - concepts and facilities*, STD 13, RFC 1034, November 1987

[RFC 1035] Mockapetris, P., *Domain names - implementation and specification*, STD 13, RFC 1035, November 1987

[RFC 1334] Lloyd, B. and W. Simpson, *PPP Authentication Protocols*, RFC 1334, October 1992

[RFC 1492] Finseth, C., *An Access Control Protocol, Sometimes Called TACACS*, RFC 1492, July 1993

[RFC 1510] Kohl, J. and C. Neuman, *The Kerberos Network Authentication Service (V5)*, RFC 1510, September 1993

[RFC 1994] Simpson, W., *PPP Challenge Handshake Authentication Protocol (CHAP)*, RFC 1994, August 1996

[RFC 2039] Kalbfleisch, C., *Applicability of Standards Track MIBs to Management of World Wide Web Servers*, RFC 2039, November 1996

[RFC 2486] Aboba, B. and M. Beadles, *The Network Access Identifier*, RFC 2486, January 1999

[RFC 2608] Guttman, E., Perkins, C., Veizades, J., and M. Day, *Service Location Protocol, Version 2*, RFC 2608, June 1999

[RFC 2750] Herzog, S., *RSVP Extensions for Policy Control*, RFC 2750, January 2000

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

[RFC 2903] de Laat, C., Gross, G., Gommans, L., Vollbrecht, J., and D. Spence, *Generic AAA Architecture*, RFC 2903, August 2000

[RFC 2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, *Stream Control Transmission Protocol*, RFC 2960, October 2000

[RFC 3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, *SIP: Session Initiation Protocol*, RFC 3261, June 2002

[RFC 3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, July 2003

[RFC 3344] Perkins, C., *IP Mobility Support for IPv4*, RFC 3344, August 2002

[RFC 3525] Groves, C., Pantaleo, M., Anderson, T., and T. Taylor, *Gateway Control Protocol Version 1*, RFC 3525, June 2003

[RFC 3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, *Extensible Authentication Protocol (EAP)*, RFC 3748, June 2004

[RFC 4301] Kent, S. and K. Seo, *Security Architecture for the Internet Protocol*, RFC 4301, December 2005

[RFC 4346] Dierks, T. and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, RFC 4346, April 2006

[RFC, 2007] IETF, RFC bibliographic listing, visited April 2007, ftp://ftp.rfc-editor.org/in-notes/rfc-ref.txt

[SDL, 2007] website Specification and Description Language (SDL), visited April 2007, http://www.sdl-forum.org/SDL/index.htm

[Shibboleth, 2007] website Shibboleth project by Internet2, visited April 2007, http://shibboleth.internet2.edu/

[Spin, 2007] website Spin LTL model checking tool, visited April 2007, http://spinroot.com/spin/whatispin.html

[Turoff et al, 2002] Murray Turoff, Harold Linstone, *The Delphi method: Technique and application*, 2002, http://www.is.njit.edu/pubs/delphibook/

Appendix A Diameter

The AAA protocol RADIUS was developed in the early 90s. At that time Internet was used differently; where people were using dial-in to connect to it. With the development of web 2.0 and the ever increasing capabilities of routers and Network Access Servers (NAS), the demands changed and created the need for a replacement of the RADIUS protocol.

In September 2003 a new AAA protocol Diameter was standardized. The Diameter protocol was developed to resolve the issues that RADIUS left open. In new application areas like Wireless Local Access Network (WLAN) and Voice over IP (VoIP) Diameter is better suited and gives better support for roaming users.

This chapter discusses the Diameter protocol as standardized by the IETF. First a short description of the standardization process is given, followed by an overview of the existing Diameter applications in the Diameter framework. The Diameter base protocol is discussed and the differences with other protocols like RADIUS and COPS are pointed out. Finally the Diameter applications as standardized by IETF and other proprietary applications are discussed.

A.1 History

Diameter was developed by Pat Calhoun in 1996 while working at Sun Microsystems. The protocol was designed as an improved version of the RADIUS protocol because of its shortcomings [HP, 2002]. The Diameter protocol consists of a base protocol [RFC 3588] and extensions. The Base protocol was documented in an RFC in 2003 and is now in the phase of Proposed Standard. At time of writing five Diameter applications are part of the Standards Track and are in the phase of Proposed Standard.

Work on the Diameter protocol is done in the AAA working group of the IETF. This group will focus on the development of the Diameter protocol. Maintenance and extensions are done in the Diameter Maintenance and Extensions (DIME) working group [Dime, 2007]. This group focuses on the maintenance of the Diameter protocol and applications, and will make sure that the work done on RADIUS will also be available for Diameter.

Over the years that Diameter already exists, a lot of drafts have been written and expired. Drafts about Diameter are written in various working groups. A complete overview of drafts and RFCs that depend on a particular RFC or draft can be found with [Fenner, 2007].

The development of Diameter is currently focused on supporting access to IP networks. Because of the flexibility of the protocol, it can be used for generic purposes in the AAA domain.

A.2 Diameter framework

The Diameter protocol consists of the Diameter Base protocol and Diameter protocol applications as shown in Figure 27. The applications are extensions of the Diameter Base protocol.

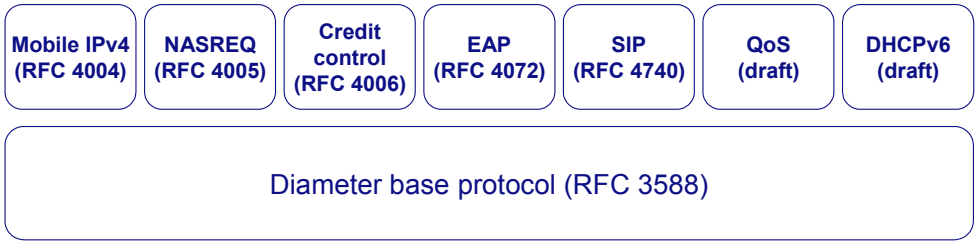

Figure 27 Diameter framework

In the base protocol the functionality is implemented that is common in all supported services, like mechanisms for reliable transport, message delivery and error handling. The base protocol must be supported by all applications.

Diameter runs over Transmission Control Protocol (TCP) [RFC 793] or Stream Control Transmission Protocol (SCTP) [RFC 2960]. The different Diameter nodes are interconnected in a peer-to-peer structure. The Diameter framework enables push and pull application models and architectures [HP, 2002]. The Diameter base protocol defines the protocol header and the necessary Attribute-Value Pairs (AVPs). The applications can extend the protocol by defining new messages and AVPs and append them to the Protocol Data Units (PDUs).

For backward compatibility with legacy protocols, the Diameter protocol does not share common PDUs with RADIUS. A translator is necessary to translate the Diameter and RADIUS PDUs.

A.2.1 Diameter agents

A Diameter node is a client, agent or server. A Diameter client is a device at the edge of the network that performs access control. A Diameter agent can be a relay, proxy, redirect or translation agent. In the following section the differences between these agents will be explained. The Diameter server handles the authentication, authorization and accounting requests for a specific realm. A realm is an administrative domain where the server resides in.

A Diameter agent may act in a stateful manner for some requests while being stateless for others. An agent can also be one type of agent or server for some requests, but another type of agent or server for other requests [IBM, 2006; RFC 3588].

Relay agent

A relay agent routes messages to other Diameter nodes based on the information found in the messages. A relay agent is allowed to modify the messages by inserting and removing routing information, but is not allowed to modify other parts of the messages. The relay agent has a realm routing table which contains a list of supported realms and known peers. In Figure 28 the order of the flow of messages is shown from Diameter nodes in two different realms.

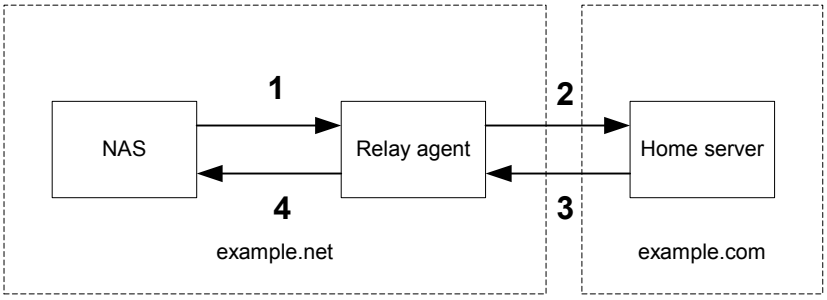

Figure 28 Relay agent

Proxy agent

Proxy agents also route messages, but they can modify those messages to implement policy enforcements. Due to the fact that the proxy agent can modify messages, no end-to-end security is possible. There is no difference in flow compared to the relay agent, as can be seen in Figure 29.

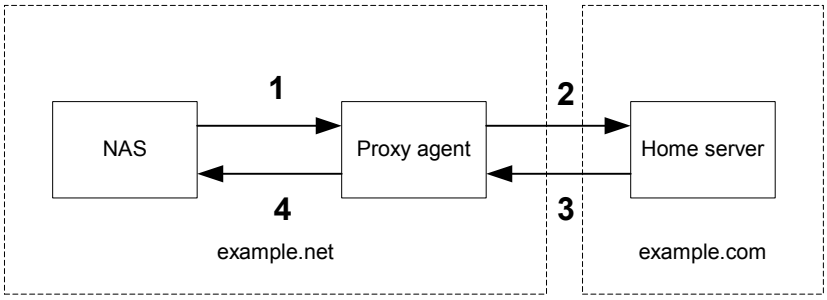

Figure 29 Proxy agent

Redirect agent

A redirect agent can tell to an agent where to find another Diameter server, for example a home-server for a particular user. Redirect agents can be used when the Diameter routing configuration needs to be centralized. Because redirect agents do not handle messages, they also do not modify the messages. They only return an answer to the Diameter agent needed to set up a direct communication path. When a request enters the relay agent, the redirect agent is asked where the home server is located. The relay agent can setup a connection with the home server, as shown in Figure 30.

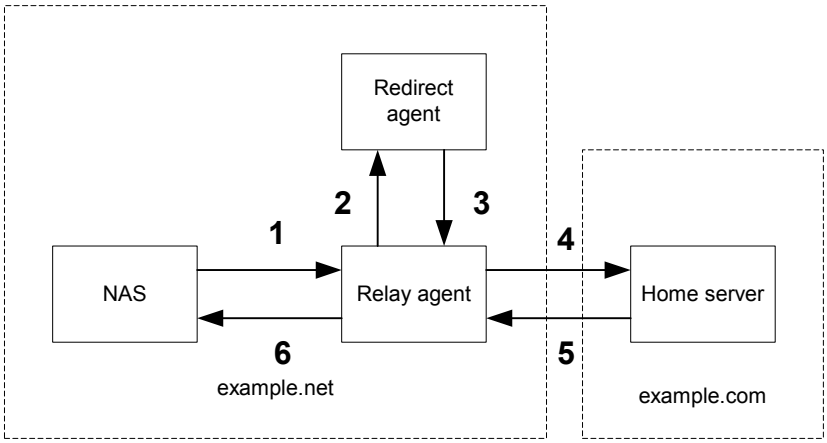

Figure 30 Redirect agent

Translation agent

The translation agent provides the protocol translation between two protocols, like RADIUS and Diameter or the even older protocol TACACS+ [RFC 1492] and Diameter. The translation agent can only perform this translation for the Diameter applications it is familiar with. The translation agent always resides in the Diameter domain, while it is a Diameter agent. The RADIUS client or server is not aware of the existence of another protocol and sees the Diameter translation agent just as another RADIUS agent.

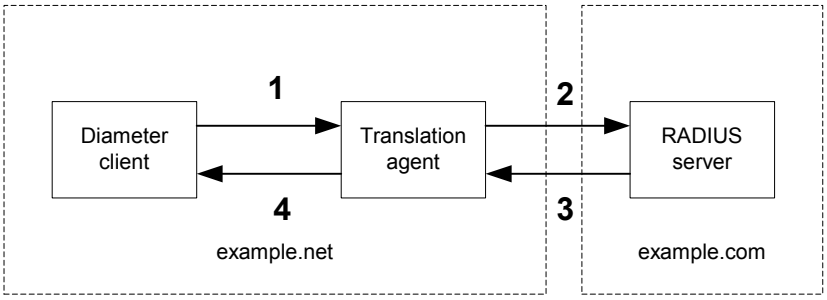

Figure 31 Translation agent

A.3 Diameter base protocol

In the Diameter base protocol the basic Diameter packet structure is described. The Diameter peer discovery and capability exchange mechanisms are explained in there. Furthermore the fail over mechanisms and error handling are explored. Finally the basic accounting features and security, as specified by the base protocol, are described.

A.3.1 Diameter header

The Diameter message consists of a Diameter header and one or more attribute-value pairs (AVPs) which carry authentication, authorization or accounting information or protocol specific data.

The Diameter header is 20 bytes, divided in fields as shown in Figure 32.

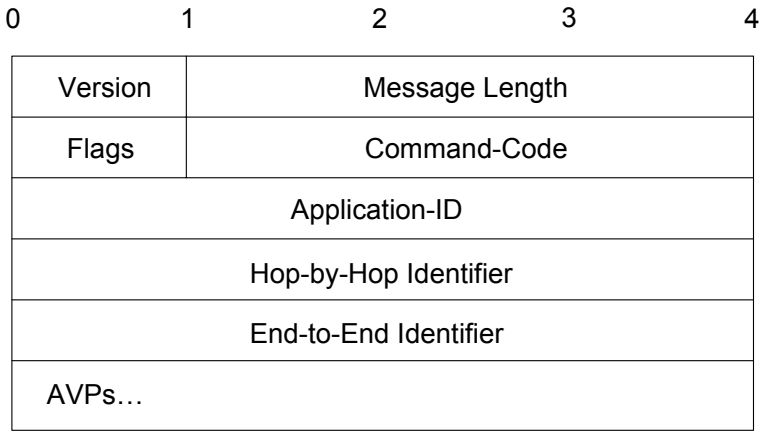

Figure 32 Diameter protocol header [RFC 3588]

The message length field indicates the length of the Diameter message including the header fields. In the flags field the first four bits are defined, the other four are reserved for later use:

- Request (R): If set, this message is a request, else an answer.

- Proxiable (P): If set, this message may be proxied. If cleared the message must be processed locally.
- Error (E): If set, this message contains an error message. This cannot be set in request messages.
- Potentially re-transmitted message (T): This is set when a retransmission is sent, to help remove duplicates. This flag can only be set in request messages.

The command code field contains the information about which command is sent. Command codes are further described in subsection A.3.2. In the application-ID field it is identified for which application this message is used. This can be a standard Diameter application or a vendor specific application.
The hop-by-hop identifier is a random generated value, increasing each hop when transporting a request. The answer message uses the same hop-by-hop identifier as found in de request message. The end-to-end identifier is used to identify duplicate messages. The answer message uses the same end-to-end identifier as in the request message. The difference with the hop-by-hop identifier is that the end-to-end identifier stays the same during transportation.

After the Diameter header the AVPs follow. More details about AVPs are given in subsection A.3.3.

A.3.2 Diameter command codes

Three command codes have been defined in the Diameter base protocol are shown in Table 3. A request and answer message have the same command code. In other Diameter applications more command codes are defined. An overview of command codes and messages from the Diameter applications can be found in Appendix B.

Table 3 Diameter command codes defined in the base protocol [IBM, 2006]

Message name	Abbreviation	Command code
Capabilities-Exchanging-Request	CER	257
Capabilities-Exchanging-Answer	CEA	257
Device-Watchdog-Request	DWR	280
Device-Watchdog-Answer	DWA	280
Disconnect-Peer-Request	DPR	282
Disconnect-Peer-Answer	DPA	282

A.3.3 Diameter AVP

Diameter uses Attribute-Value Pairs (AVP) to send data or AAA specific information. Diameter AVPs are defined in the base protocol or Diameter application documents.

The header of an AVP is shown in Figure 33 and it is placed after a Diameter protocol header or another AVP. Every AVP must be padded to align on a 32-bit boundary.

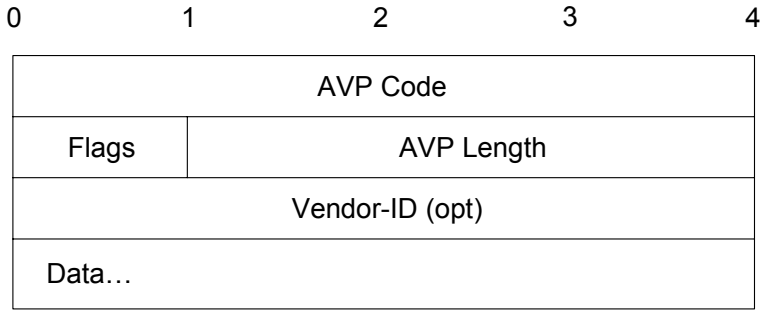

Figure 33 AVP header [RFC 3588]

The AVP code combined with the vendor-ID gives a unique identification. The AVP numbers 1 through 255 are reserved for RADIUS attributes. The first three AVP flags are defined as follows:

- Vendor-specific (V): Indicates whether the vendor-id field is present in the header.
- Mandatory (M): Indicates whether the support of this AVP is required.
- End-to-end encryption (P): Indicates the need for end-to-end encryption.

The AVP length field states the length of the AVP without padding.

A Diameter AVP data field can be of different formats: OctetString, Integer32, Integer64, Unsigned32, Unsigned64, Float32, Float64 and Grouped. In the RFCs the format per AVP is defined.
Applications can use AVP Derived Data Formats which are formats derived form the Basic AVP Data Formats. Commonly used AVP Derived Data Formats are: Address, Time, UTF8String, DiameterIdentity, DiameterURI, Enumerated, IPFilterRule, QoSFilterRule.

Grouped AVPs are used to nest more AVPs in an AVP. In this case the data field of the AVP contains more AVPs. [HP, 2002]

A.3.4 Diameter peers

Diameter is a peer-to-peer protocol. In this section the connection establishment and communication with peers is described.

A Diameter node should at least have two peers per realm, the primary and secondary peer, for robustness. The peers to whom the Diameter node is connected to are stored in the Peer Table.

Dynamic peer discovery makes the Diameter protocol more simple and robust. Peer discovery can occur when the client needs to discover a first-hop Diameter agent or when the client needs to discover another agent for further handling of a Diameter operation. The peer discovery mechanisms are based on existing IETF standards. The possible discovery mechanisms are: manually configured list of agent locations, Service Location Protocol (SRVLOC) [RFC 2608] and Domain Name Server (DNS) [RFC 1034; RFC 1035].

When two nodes want to establish a connection, they must use the capability exchange messages to find out the peer's identity and its capabilities. The capability exchange possibility can only be used by next-hop peers. After a *Capability-Exchange-Request* always a *Capability-Exchange-Answer* follows, even if the peers have no applications in common.

When disconnecting with a peer, the *Disconnect-Peer-Request* should be used to inform the peer of its intent to disconnect the transport layer, and that the peer shouldn't reconnect unless it has a valid reason to do so. If a peer disconnects without a Disconnect-Peer message the node will periodically attempt to reconnect.

A.3.5 Failover mechanisms and error handling

Two types of errors can occur: protocol errors and application errors. The protocol errors are errors at the base level of the protocol including routing problems etc. The application errors are problems with a function specified in a Diameter application.

If a transport failure is detected with a peer, the messages pending are forwarded to another agent, and the T flag is set. This is the failover mechanism of Diameter.

"In order for a Diameter node to perform failover procedures, it is necessary for the node to maintain a pending message queue for a given peer. When an answer message is received, the corresponding request is removed from the queue. The Hop-by-Hop Identifier field is used to match the answer with the queued request." [RFC 3588]

To support early connection failure detection, the Diameter protocol defines a *Device-Dogwatch-Request* message. When two connected Diameter nodes don't exchange messages for a certain length of time, this message is sent from either of these nodes to detect possible network problems [IBM, 2006].

For error handling purposes different Result-Code AVP values are specified to indicate if the request was completed successfully or an error occurred. Each Diameter answer message must have a Result-Code AVP. More about error handling is also specified in the Diameter applications.

A.3.6 Accounting

The general functionality for accounting is described in the base protocol. Accounting is based on a server directed model with capabilities for real-time delivery of accounting information. A server directed model means that the device that generates the accounting records follows the direction of an authorization server [RFC 3588; IBM, 2006].

If a Diameter client is successfully authenticated and authorized, it must send an *Accounting-Request* (ACR) to the server. The *Accounting-Answer* (ACA) is used for conformation of reception.

The accounted service can be a one-time event or of measurable length. If it is of type measurable length, then the *Accounting-Record-Type* AVP must have the value of START_RECORD, STOP_RECORD and possibly INTERIM_RECORD. In the Diameter application specifications sequences must be defined.

In the base protocol some AVPs are defined that must be present in accounting messages. For applications where a user receives service from different access devices (each with distinct Session-Ids), such as Mobile IPv4, the *Accounting-Multi-Session-Id* AVP, can be used for correlation [HP, 2002].

Batch accounting is not implemented in the Diameter base protocol.

A.3.7 Security

Using security in Diameter is mandatory. Either IP security (IPSec) [RFC 4301] or TLS (Transport Layer Security) [RFC 4346] should be used. TLS is recommended to be used as inter-domain security, in connections between administrative domains.

IPSec is to be used for intra-domain usage when pre-shared secrets are used as a security mechanism [RFC 3588].

End-to-end security by using TLS or IPSec is strongly recommended for all Diameter applications by the base protocol.

When the distribution of authentication keys takes place and there are untrusted Diameter agents in the path, IPSec or TLS must be used to eliminate the agents in the path.

A.4 Comparison with other protocols

Diameter provides some of the functionalities that other protocols also can provide, for example network access like RADIUS, policy control like Common Open Policy Service (COPS) [RFC 2748] and gateway control like H.248/megaco [RFC 3525]. In this section the differences between Diameter and the protocols RADIUS, COPS and H.248 are discussed.

A.4.1 Diameter vs. RADIUS

Diameter was designed to overcome certain shortcomings of the RADIUS protocol. In this subsection the shortcomings are described and how they are overcome by the Diameter protocol. This comparison is derived from [HP, 2002; Cisco, 2001].

Strict limitation of attribute data

In RADIUS the length of the data in the AVPs is defined in a field of one octet, hence only 255 bytes of data can be send in an AVP. In Diameter the attribute length field is three octets.

Retransmission

The identifier field in the header of the RADIUS packet is used to recognize retransmissions. The identifier field is one octet, so the maximum number of outstanding messages between a RADIUS client and a RADIUS server is 255. In Diameter the end-to-end identifier field is used for retransmission purposes and is four octets.

In Diameter lost packets are retransmitted at each hop, because the loss of a packet is already noticed by the next hop.

RADIUS uses the connectionless protocol User Datagram Protocol (UDP) [RFC 768] and has no standard scheme to regulate the UDP flow. Diameter runs over TCP or SCTP and therefore contains flow control and congestion avoidance mechanisms.

Failure detection

Diameter can detect a local failure of a peer and therefore can take care of proper failover in case of network congestion, temporary network failure in the path to the home server or failure at the home server. In RADIUS the NAS cannot distinguish what went wrong if there is no timely response to a request and assumes that it was a failure at the next-hop server, so it connects to another peer.

Hop-by-hop transport failure detection allows failover in Diameter to occur at the appropriate place; proxies can locally failover to an alternate next-hop peer.

In RADIUS the server cannot tell the client if it is running or going down. In case of the client failing over to an alternate RADIUS server, the client does not know if the alternate server is even reachable. Diameter solves this issue by using keep alive messages and messages indicating that a server is going down.

Silent discarding of packets

If the client sends packets with the wrong information or the packet contains errors, the RADIUS server silently discards them, while the Diameter server notifies the client by sending an error message. In RADIUS the client does not know the difference between the sending of wrong messages and absence of the server. The client will repeat to try and send the message or send it to another server.

End-to-end message acknowledgement

In RADIUS the client cannot tell, in case that no response is received, if the request was lost on its way to the server, or the response was lost during the transport back to the client. In Diameter the client receives a response or acknowledgement of the request and can tell if the request reached the server.

No unsolicited server messages

Server initiated messages are not allowed in the RADIUS protocol and when needed a solution outside RADIUS has to be found or proprietary extensions are needed. In Diameter two server initiated messages are supported; the server requesting that the

client terminates a specific user session and a re-authentication request for a specific user.

End-to-end security

Only hop-by-hop security is supported by RADIUS, so every hop can easily modify information or confidential information can be compromised. Diameter supports hop-by-hop security and end-to-end security. Digital signatures and encryption of AVPs make sure that the information is secured from end-to-end.

RADIUS has no mechanism preventing replay attacks. The same message can be sent over and over again, which can result in denial of service if the server has a limited amount of concurrent sessions for a user. By using end-to-end encryption and authentication, this is prevented in Diameter.

RADIUS requires that there is a common shared secret between two peers, even if IPsec is used. In Diameter communication can be secured by either IPsec or TLS.

No support for vendor-specific commands

Vendor-specific attributes are supported in RADIUS, but no vendor-specific commands. This results in interoperability problems as vendors create private command codes. Diameter supports both vendor specific attributes as well as commands.

No alignment requirements

In RADIUS there are no alignment requirements which can result in unnecessary burdens for processors. In Diameter a 32-bit alignment requirement is specified. Every Diameter message header and AVP header has to align on the 32-bit boundaries, which makes the process of treating headers and AVPs as byte aligned characters unnecessary.

Session control

In Diameter session management is independent of accounting. Accounting information can be routed to a different server than authentication/authorization messages. The session is terminated by a specific session-termination message rather then an Accounting Stop message as it was in RADIUS.

Trust establishment

In RADIUS trust is based on a pre-configured set of roaming domains, while in Diameter trust is based on a pre-configured set of roaming Certificate Authorities (CAs). The key management is done in RADIUS by DNSsec. Diameter requires Public Key Infrastructure (PKI) management with the CAs. [Eertink et al, 2005b]

A.4.2 Diameter vs. COPS

Common Open Policy Service (COPS) offers the possibility to exchange policy information between a policy server and its clients, being a policy transport protocol. The protocol was created for the general administration, configuration, and enforcement of policies and is described in RFC 2748. The policy server is called Policy Decision Point (PDP) and the client is called Policy Enforcement Point (PEP).

The main difference with Diameter is that COPS is a policy control protocol that can deliver AAA, while Diameter is an AAA protocol that can deliver policy control.

There are two models: the outsourcing model and the configuration (provisioning) model (COPS-PR).

In the outsourcing model, the PDP contains all policies and the PEP needs to consult the PDP every time it needs to make a decision. The PDP does all the processing of analyzing information and taking decisions and then sends the result back to the PEP.

In the configuration model, also known as the provisioning model, the PEP lets the PDP know which decision-making capabilities it has. The relevant policies are then downloaded to the PEP. A policy information base is used as a repository for the policies.

Both models are also found in the Diameter QoS application as described in subsection A.5.6.

The comparison is derived from [RFC 3127].

End-to-end connectivity

In COPS the protocol runs from the PEP to the server, and the PEP can be placed at different devices like the end-user device or the network access node. Diameter always runs from the edge node from the network to the AAA server.

Inter-domain

Diameter is known for its inter-domain functionality, and helps to resolve roaming issues. COPS has no inter-domain support but only intra-domain functionality.

EAP

COPS claims better interoperability with the Extensible Authentication Protocol (EAP) [RFC 3748], because both protocols support enforcement procedures. For Diameter a special Diameter EAP application is developed to support EAP over Diameter. [Chaouchi et al, 2002]

Security

COPS has hop-by-hop security like Diameter (IPSec or TLS) but no end-to-end security. Diameter provides end-to-end security.

Stateful

In COPS there is no dynamic discovery of peers, like Diameter. The PDP must know all its PEP's and remains state information about them. COPS is stateful in two ways. First, requests from the client PEP are installed or remembered by the remote PDP until they are explicitly deleted by the PEP. Second, the server may respond to new queries differently because of previously installed Request/Decision state(s) that are related [RFC 2748].

A.4.3 Diameter vs. H.248

H.248 is a standard created by the ITU-T, which is based on the IETF protocol named Megaco [RFC 3525]. The predecessor of both protocols is the Media Gateway Control Protocol (MGCP). This protocol was developed for IP telephony signaling.
The differences between H.248 and Megaco are described in the beginning of [RFC 3525].

H.248 is an extensible protocol and there are applications that have similar functionality as parts of the Diameter protocol.

H.248 and Diameter are compared, because at one point in 3GPP the choice had to be made to use Diameter or H.248. Actually these protocols are not each others replacement, because H.248 has more SIP like functionality, that Diameter does not contain. Diameter quality of service is about authorization of resource reservations, and no gateway control itself.

A.5 Diameter applications

A set of basic applications for Diameter are developed by IETF, but others parties can also make Diameter applications to fulfill specific needs. IANA allocates Diameter application ID's to parties that request a new application. The rules defined in the Diameter base protocol on the request for a new application ID must be followed.

There are, at the moment, five applications for Diameter standardized by the IETF: Mobile IPv4, NASREQ, Credit control, EAP and SIP. These and two current IETF Internet drafts are described in this section. For an overview of the messages added by the applications see Appendix B.

3GPP developed additional applications to work in the interfaces of their IP Multimedia Subsystem. A short overview of these applications is given at the end of this section.

A.5.1 Mobile IPv4 application

The Mobile IPv4 application allows mobile nodes to receive service from foreign service providers. The application allows the Diameter server to authenticate, authorize and collect accounting information for its clients. The Mobile IPv4 application cannot be used with the Mobile IPv6 protocol. More about Mobile IPv4 can be found in [RFC 3344].

Interaction diagram

In this application the Foreign Agent (FA) or Home Agent (HA) acts as the Diameter client, because the mobile nodes interact over IPv4 with the FA. The basic functionality of Mobile IPv4 is that the HA intercepts packets that are directed to the home address of the mobile user and encapsulates them. It sends the packets over the network through a tunnel to the FA to which the mobile node is connected.

The interaction between the devices is as follows. When a mobile node requests access on a foreign network, the FA will contact the authentication server of that domain (FAAA). The server will contact the authentication server of the home domain (HAAA) by sending an *AA-Mobile-Node-Request* (AMR). The authentication server of the home domain sends a *Home-Agent-MIP-Request* (HAR) to the HA. If the mobile node is allowed to connect the HA, the HA sends a *Home-Agent-MIP-Answer* (HAA) back. The HAAA sends back a *AA-Mobile-Node-Answer* (AMA) and the mobile node is accepted.
After this a Mobility Security Association (MSA) is established between the FA, the HA and Mobile Node. In the next section more details about the MSAs are given.

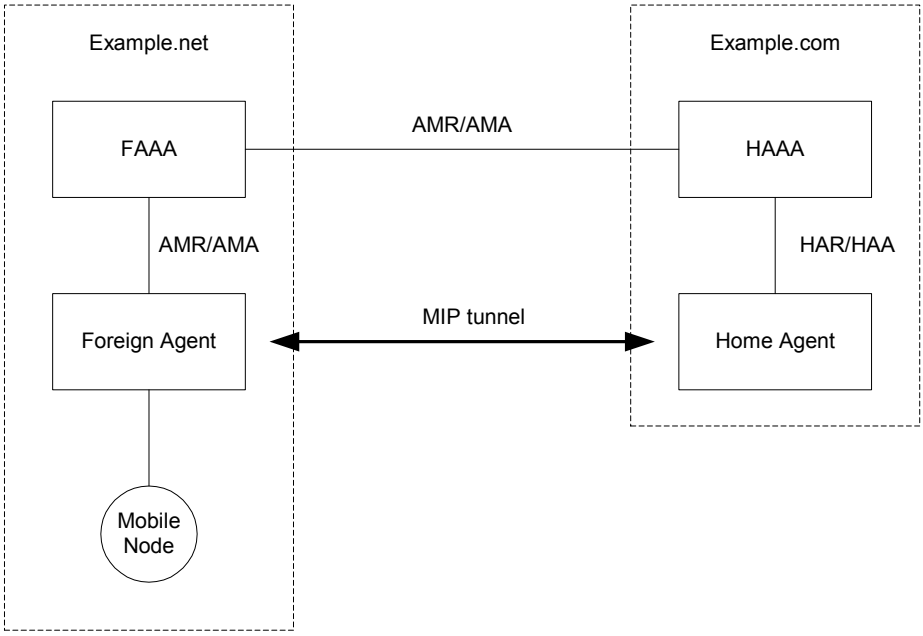

Figure 34 Diameter Mobile IPv4 interaction

The home agent uses the *acct-multi-session-id* AVP to identify the mobile node when handed over to another foreign agent. This way the accounting can continue for a particular session with a mobile node.

If the request from the mobile node does not ask for a particular Home Agent, the Diameter server will allocate one according to its internal policy [Savola, 2003].

The mobile node is identified by its NAI, an IETF standard RFC 2486, which is used to find the realm to which it belongs.

Key distribution

In Figure 35 the Mobile Security Associations (MSAs) can be seen that are established between the FA, HA and MN.

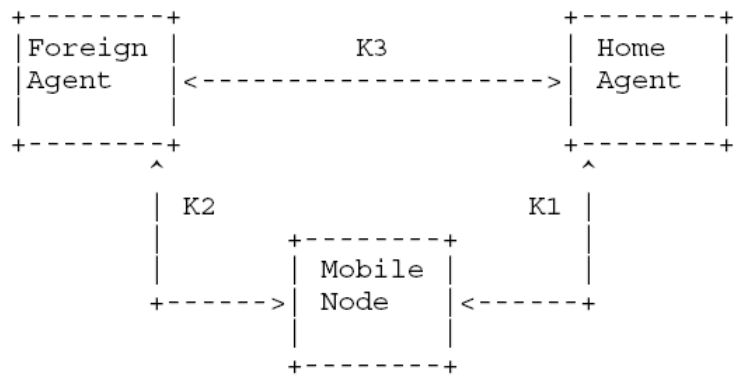

Figure 35 Mobile Security Associations [RFC 4004]

The MSAs are established as follows: After the authentication phase, in the authorization phase session keys are generated for the establishment of MSAs between the FA, HA and MN. The session keys are used to compute authentication extensions applied to Mobile IP registration messages. The session key is symmetric, which means the same for both directions between two entities.

The HAAA generates session keys and transmits them to the FA and HA. The HAAA also generates nonces that correspond to the same keys and transmits them to the mobile node. All keys and nonces are generated by the HAAA, even if a HA is dynamically allocated in the foreign network.

The MN receives a nonce for each key it needs, and the mobile node will use the nonce and the long-term shared secret to create the session keys. The generated session keys by the mobile node are equal to the session keys that the FA and HA have. Once the session keys have been established and propagated, the mobility devices can exchange registration information directly.

For scalability aspects the requirement of the number of pre-existing MSAs should be minimized. To solve this, the application includes a key distribution center, which is the Diameter server distributing keys and nonces as described above, to provide mobility across different domains [RFC 4004; Gustafson et al, 2001].

A.5.2 NASREQ application

This application [RFC 4005] is the direct replacement of the authentication part of RADIUS and offers secure authentication in the Network Access Sever (NAS) environment. In this application the interaction with RADIUS is taken into account. The interactions between Diameter and RADIUS as described in the RFC 4005 of this application are to be applied to all Diameter applications, so this RFC extends the Base protocol in this area.

Interaction diagram

In the interaction diagram of this application in Figure 36, the normal behavior of the NASREQ application is shown. First an *AA-Request* (AAR) is sent to the server and if allowed an *AA-Answer* (AAA) is sent back. The *Re-Authentication-Request* (RAR) can be used by the server to verify if the user is using the service. The NAS sends back a *Re-Authentication-Answer* (RAA), where after an AAR and AAA message should follow. The session can be terminated by the server or NAS. The server can send an *Abort-Session-Request* (ASR) or the NAS can send a *Session-Termination-Request* (STR).
The accounting is done by the *Accounting-Request* (ACR) and *Accounting-Answer* (ACA) messages. These are all the messages described in the NASREQ specification.

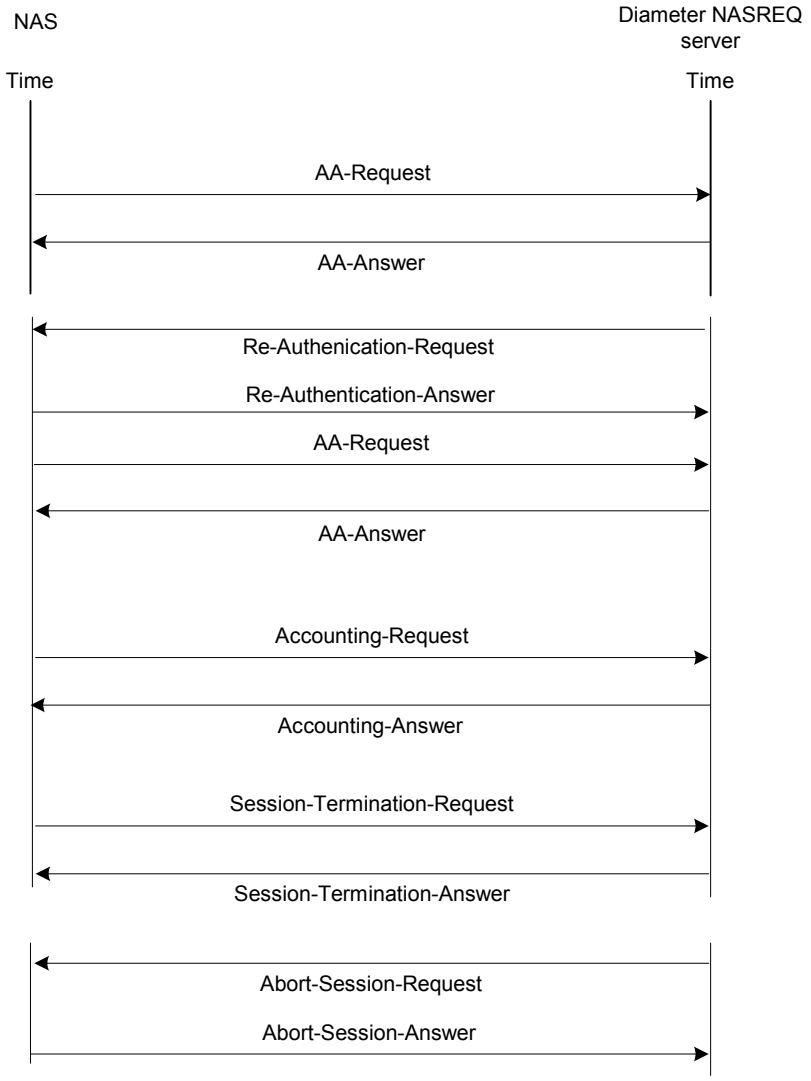

Figure 36 Interaction NASREQ

Interaction with RADIUS

AAA translation agents must follow special rules in the two possible situations:

- RADIUS request forwarded as Diameter request
- Diameter request forwarded as RADIUS request

"Some RADIUS attributes are encrypted. RADIUS security and encryption techniques are applied on a hop-per-hop basis. A Diameter agent will have to decrypt RADIUS attribute data entering the Diameter system and if that information is forwarded, the agent must secure it by using Diameter specific techniques.", from [RFC 4005].

If a RADIUS request is handled by a particular translation agent, the Diameter response always comes back at the same translation agent.

A.5.3 Credit control application

The Diameter Credit control application provides real-time credit-control for different end-user services. The application is only concerned with credit authorization for prepaid subscribers. Some accounting features are already specified at the base protocol, but these are not sufficient for real-time accounting for prepaid subscribers.

Two types of events can be seen at the application: session based credit-control and one-time events. Price enquiry, user's balance checks and refund of credit on the user's account is usually done in one-time events.

There are two different credit authorization models: authorization with money reservation and credit authorization with direct debiting.

The money reservation model is session based and works as follows: the server rates the request from the client and reserves a suitable amount of money from the user's account. Resources corresponding to the amount are returned to the user. When the user runs out of resources or ends the service, the client reports back to the server how much is used. The server returns money when resources where left over or can make a new reservation.
The money reservation model is session based. A credit-control session always consists of first, possibly intermediate and final interrogations.

Credit authorization with direct debiting is a one-time event. The server directly deducts the right amount of money for the request from the user's account.

Two messages are added by this Diameter application: *Credit-Control-Request* (CCR) and *Credit-Control-Answer* (CCA).

Credit-control sub-sessions can be used for certain applications, for example when multiple services are embedded in one user session.

Interaction diagrams

The first interrogation can differ depending on the architecture of the credit control system. The first approach uses credit-control messages after the user's authentication and authorization takes place. The interaction is shown in Figure 37.

In Figure 38 the second approach is shown. It uses service specific authorization messages for the first interrogation.

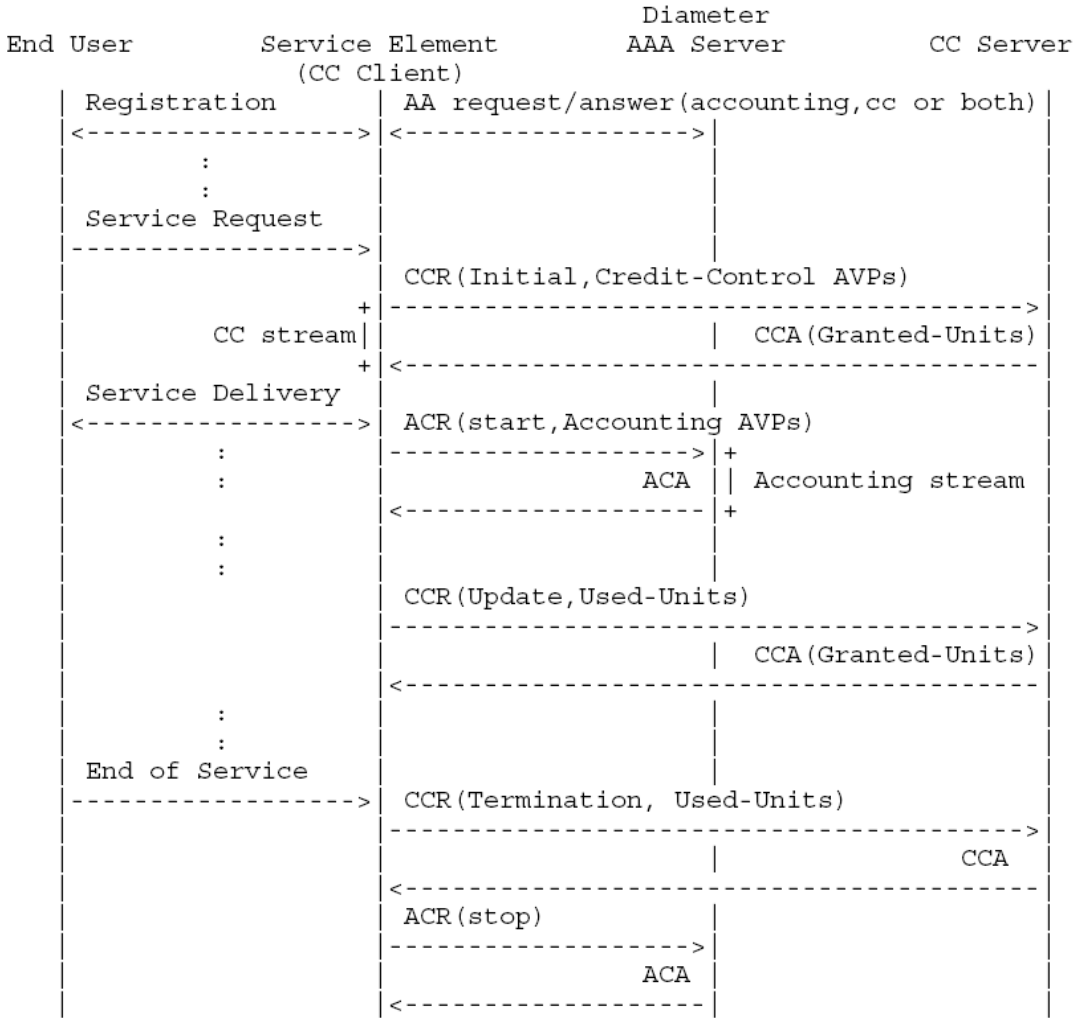

Figure 37 First interrogation after authentication and authorization [RFC 4006]

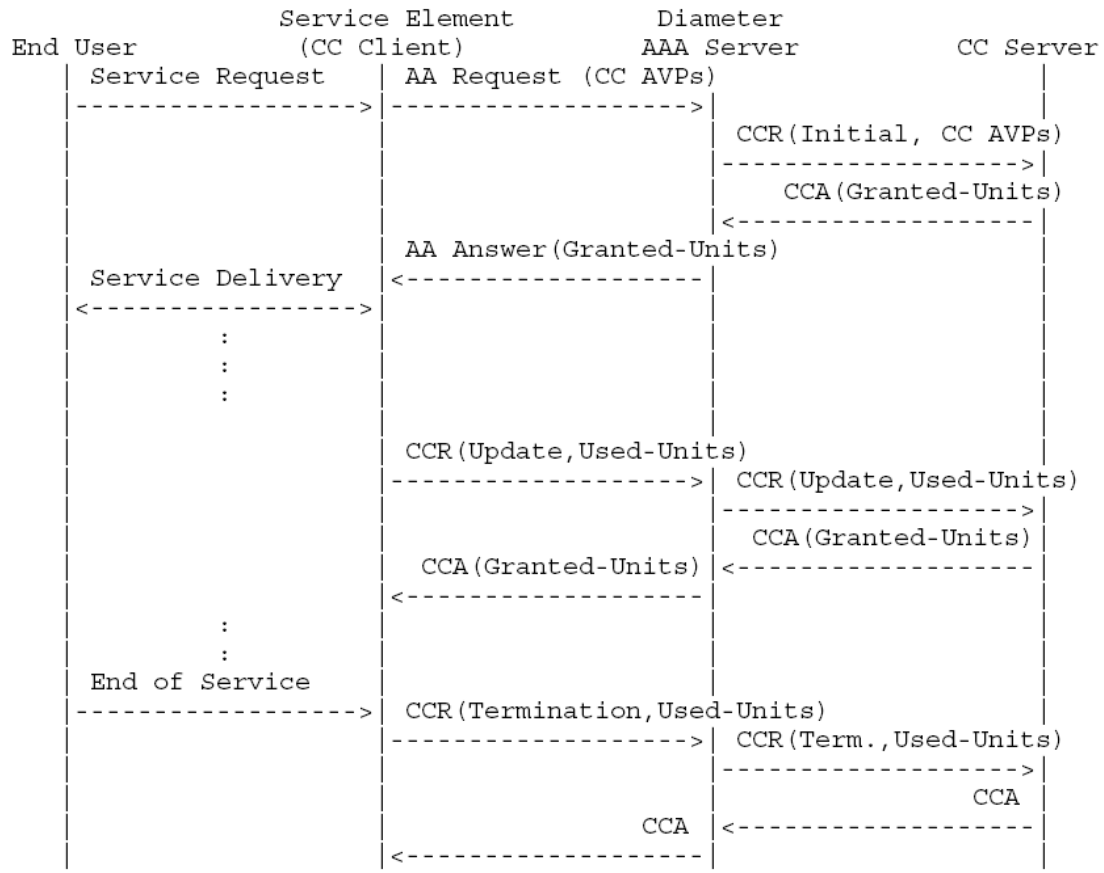

Figure 38 Authorization messages used for first interrogation [RFC 4006]

The intermediate interrogations can be used when the credit reservation is consumed, the validity time has expired or when the service had mid-session events. Multiple intermediate interrogations can happen during a session.

When a session is terminated, either by the user or CC client, a final interrogation must follow to settle the account and complete the credit-control session.

In case of a one-time event the interaction takes place as shown in Figure 39. One-time events are used for service price enquiry, balance check, direct debiting and refunding.

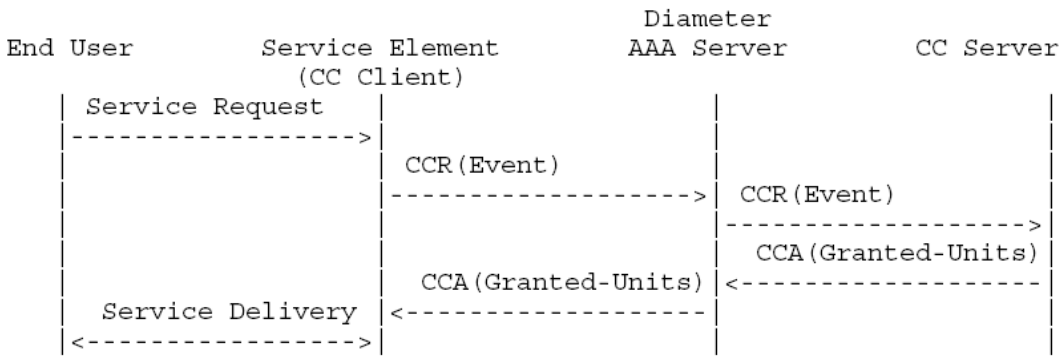

Figure 39 One-time event [RFC 4006]

Multiple credit-control servers can be used in the system for redundancy and load balancing. Architecture of the system and its interfaces are not given in the specification, because these are implementation specific.

When interoperability between RADIUS and Diameter is required, the Diameter AAA server will act as a translation agent and be the Diameter credit control client for the service elements that use other accounting techniques.

A.5.4 EAP application

In the Diameter EAP application the usage of the Extensible Authentication Protocol (EAP) over Diameter between the NAS and Diameter server is described. EAP [RFC 3748] can also be used over the data link layer between the user and the NAS, but that is not transported over Diameter and out of scope.

EAP is an authentication framework that supports multiple authentication mechanisms. If EAP is not supported, another protocol like Password Authentication Protocol (PAP) [RFC 1334] or Challenge-Handshake Authentication Protocol (CHAP) [RFC 1994] is used, but these are less secure.

The command codes *Diameter-EAP-Request* (DER) and *Diameter-EAP-Answer* (DEA) are specified in this application.

The user initiates the EAP request and sends it to its NAS. The NAS constructs a DER message and waits for the Server to respond with a DEA message.

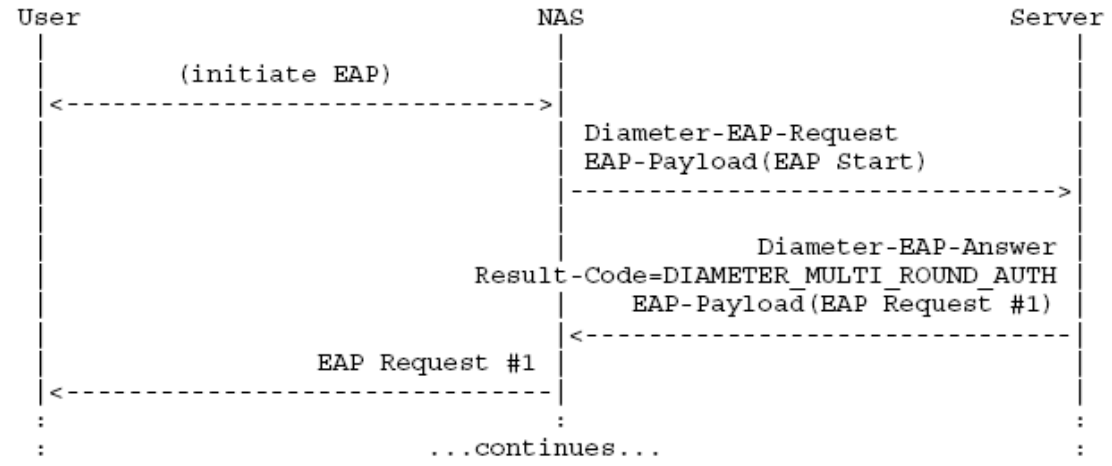

Figure 40 EAP authentication [RFC 4072]

In the DEA keying material for protecting the communication link between the user and NAS is included. After the flow sequence shown in Figure 40 is completed, the NAS receives an EAP response from the client and sends a second DER message with the client's EAP payload encapsulated. Note that this procedure is not shown in Figure 40.

An alternative approach is given in Figure 41. The NAS will issue the EAP-request/identity message to the client. The response is directly encapsulated into the DER, so no second DER transmission is necessary.

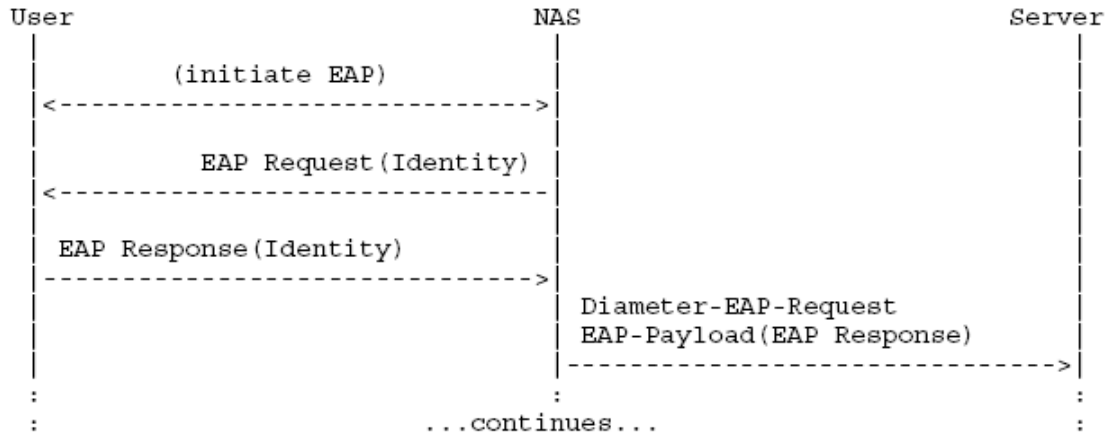

Figure 41 EAP authentication alternative [RFC 4072]

In the RFC about the Diameter EAP application four scenarios are included. The scenarios discuss how messages can be transported by hop-by-hop mechanisms since end-to-end security mechanisms are not defined for this application. The scenarios are about direct connection, direct connection with redirect, direct EAP authorization via agents, proxy agents.

A.5.5 SIP application

The Diameter SIP application [RFC 4740] is designed to be used in conjunction with the SIP protocol [RFC 3261]. It provides the functionality of authentication of the user of a SIP request and authorization of SIP resources. The SIP server and Diameter client are co-located in the same node.

For this application no particular sequence of events between SIP and Diameter are required, nor a mapping of SIP procedures to Diameter SIP application procedures.

In Figure 42 the architecture of the Diameter application for SIP can be seen. There is a single Diameter server that stores the user data. The Diameter SL is the Subscriber Locater, which has the responsibility to find the Diameter Server that contains the user-related data. For redundancy multiple Diameter servers can keep the data synchronized. Co-located with the SIP servers is a Diameter client which handles the Diameter messages for that server.

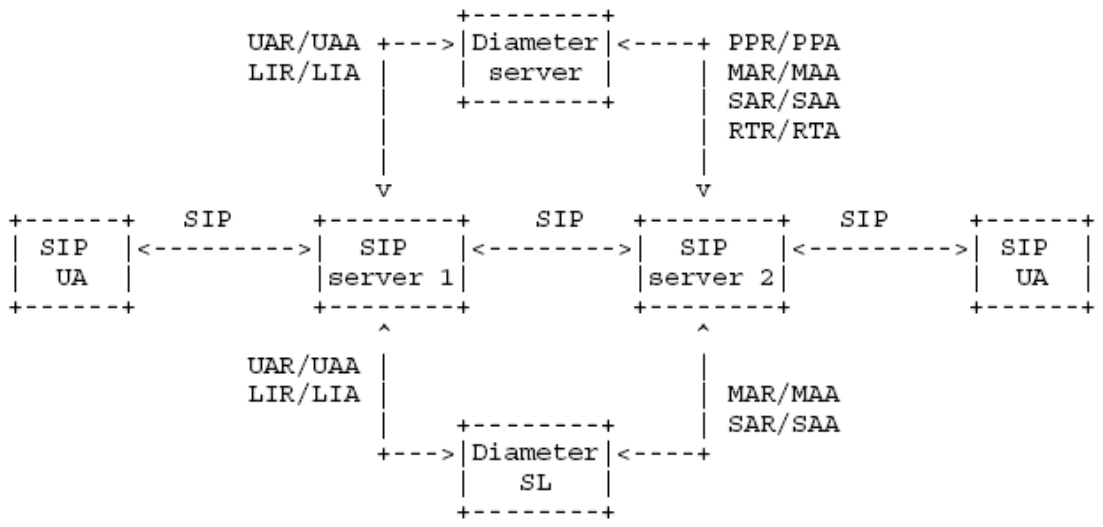

Figure 42 Diameter SIP application architecture [RFC 4740]

The first SIP server is located at the edge of the network and its responsibility is to locate the SIP server. The second SIP server requests and receives authentication and authorization data from the Diameter server.

Authorization and authentication of the user can be seen in Figure 43.

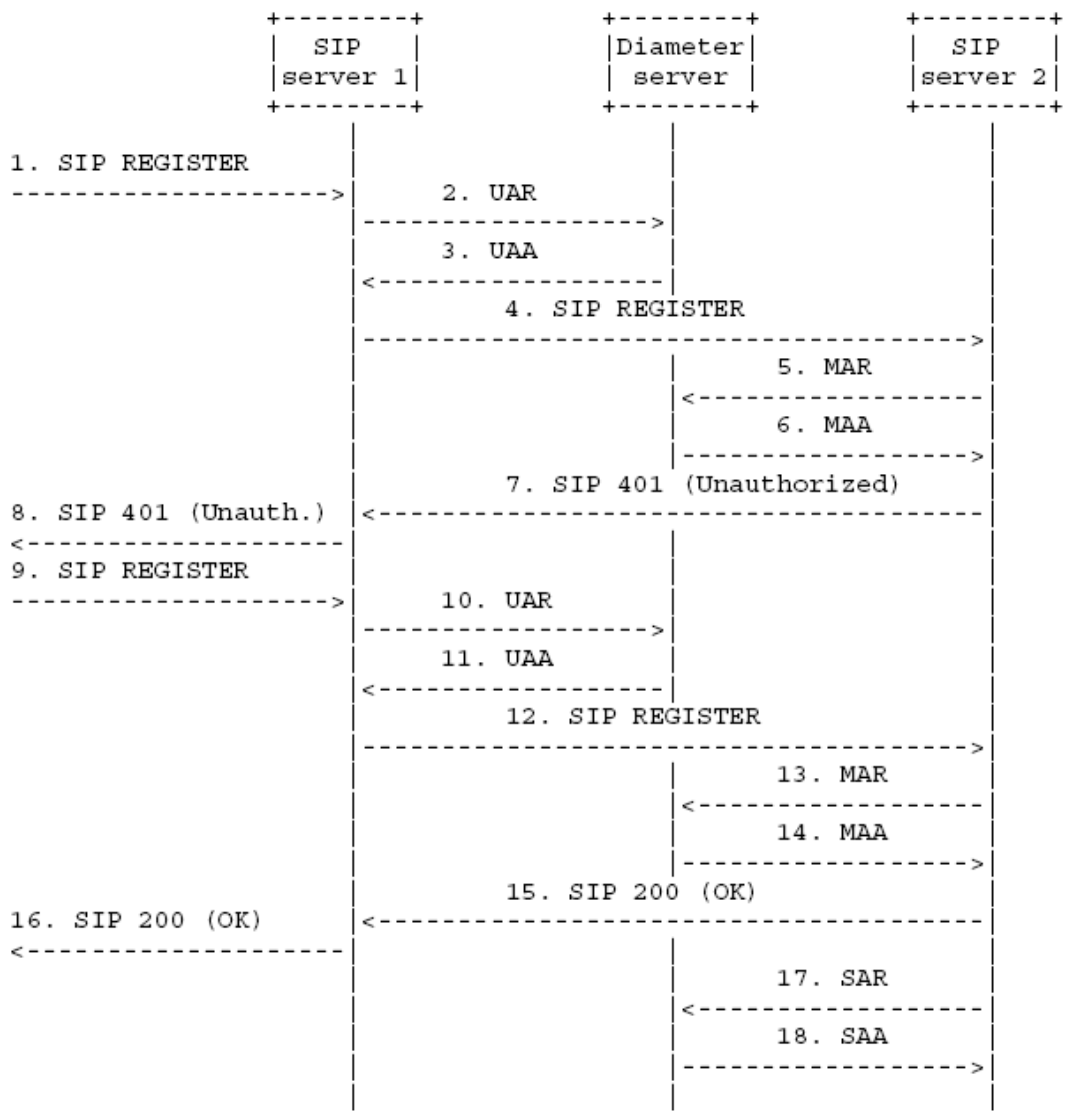

Figure 43 Authentication and authorization procedure [RFC 4740]

The first SIP server sends a *User-Authorization-Request* (UAR) to the Diameter server after it receives a SIP register request. From the Diameter server it receives the address of the SIP server that can handle the call. The second SIP server authenticates the user by sending a *Multimedia-Authentication-Request* (MAR) message to the Diameter server. In the second register request of the SIP server, the credentials from the user are included and the user is authenticated by the Diameter server. Because the first SIP server does not need to keep state, the SIP server allocated to the user has to be looked up again.

The *Server-Assignment-Request* (SAR) message can be used to retrieve the user profile from the Diameter Server or update information about the SIP server's address.

There is also another mechanism possible for authentication, which saves one round-trip. In that scenario the final authentication check is delegated to the SIP server. The messages 13 and 14 from Figure 43 are not necessary then. Due to utilization of MD5, the second SIP server can validate the credentials itself, without the extra intervention of the Diameter server.

In Figure 44 the procedure can be seen when the SIP server from the recipient of the SIP request needs to be found. The *Location-Info-Request* (LIR) message is sent to the Diameter server. The server returns the SIP URI(s) of the SIP server of the recipient. The procedure is the same for other than the SIP invite message.

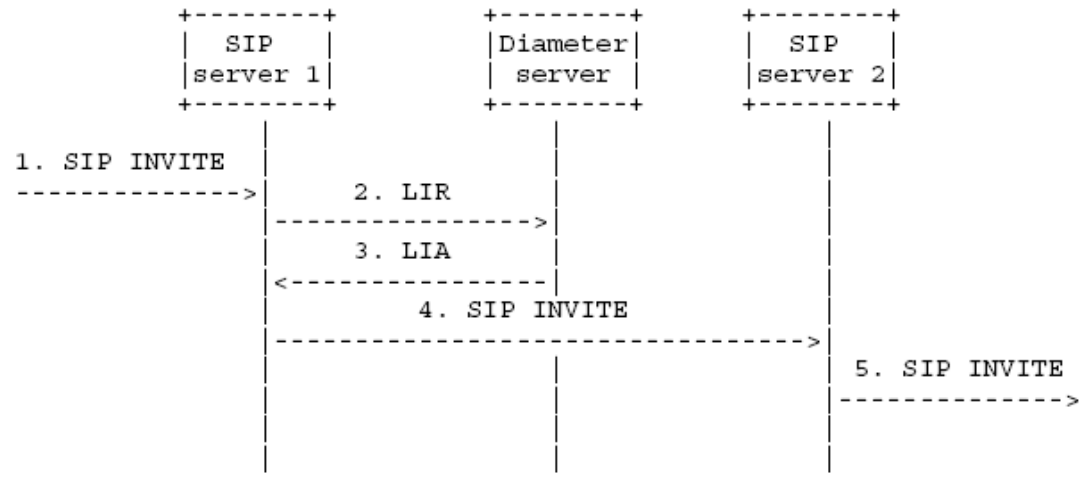

Figure 44 Locating the SIP server of the recipient [RFC 4740]

Updating the user profile can be done by sending a *Push-Profile-Request* (PPR) message to the SIP server as shown in Figure 45. In the request the Diameter server sends the updates user profile, which the SIP server acknowledges.

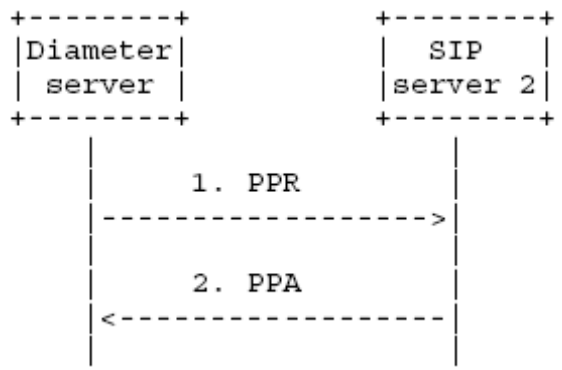

Figure 45 User profile update [RFC 4740]

In the Diameter SIP application a *Registration-Termination-Request* (RTR) and answer (RTA) are specified. These messages can be used when the Diameter server wants to terminate the SIP soft state and Diameter user sessions are not maintained.

When the sessions are maintained the server needs to use the ASR message. For termination by the Diameter client, in case that the sessions are maintained, the normal STR message is used. When the sessions are not maintained and the Diameter client wishes to terminate the SIP soft state, a SAR message must be used.

A.5.6 QoS application

The Diameter Quality of service application provides AAA for quality of service reservations [Alfano, 2006]. This means that a reservation request can be authenticated and authorized and that the resources consumed are accounted for.

A quality of service request must be made by protocols like the Resource Reservation Protocol (RSVP) [RFC 2750]. The network element receiving this request then processes this request and has to perform three different actions: admission control, authorization and resource reservation. The admission control means determining if there are enough resources to fulfill the request. The authorization server is contacted to perform authorization of the request. Then the resources are reserved.

There are two different models: the three party model and the token-based three party model. In the three party model the visited network is compensated for the resources consumed by the user via the home network. In the token-based three party model a token is used when authorization takes place at the application level, then the server will send a token to the network element which authorizes the request from the user.

The messages added by this application are: *QoS-Authorization-Request* (QAR), *QoS-Authorization-Answer* (QAA), *QoS-Install-Request* (QIR), *QoS-Install-Answer* (QIA). The first two messages are used for client initiated authorizations requests to the server. The last two messages are used server-side initiated QoS parameter provisioning, which means that the server is able to update installed QoS parameters.

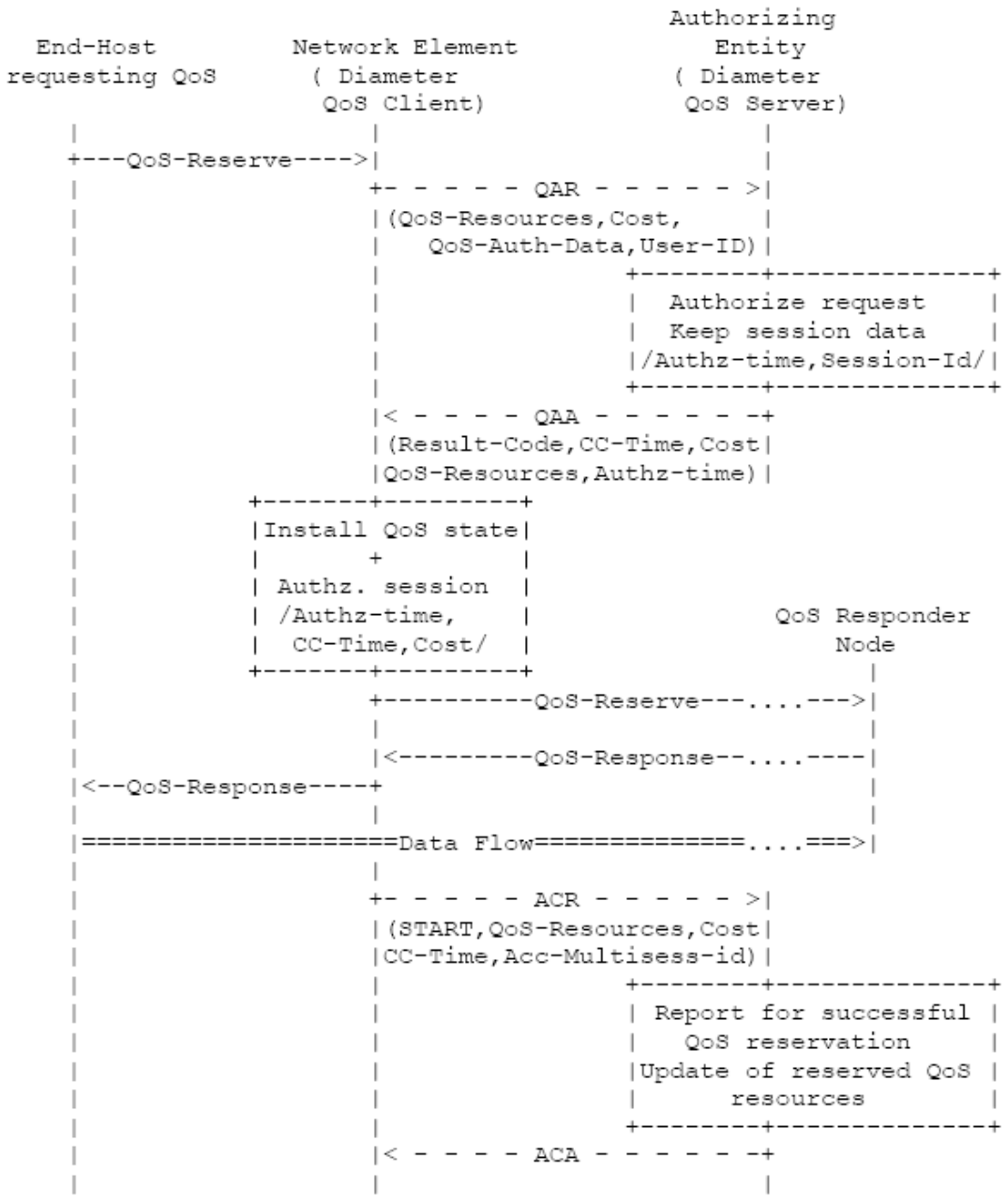

Figure 46 QoS request authorization [Alfano, 2006]

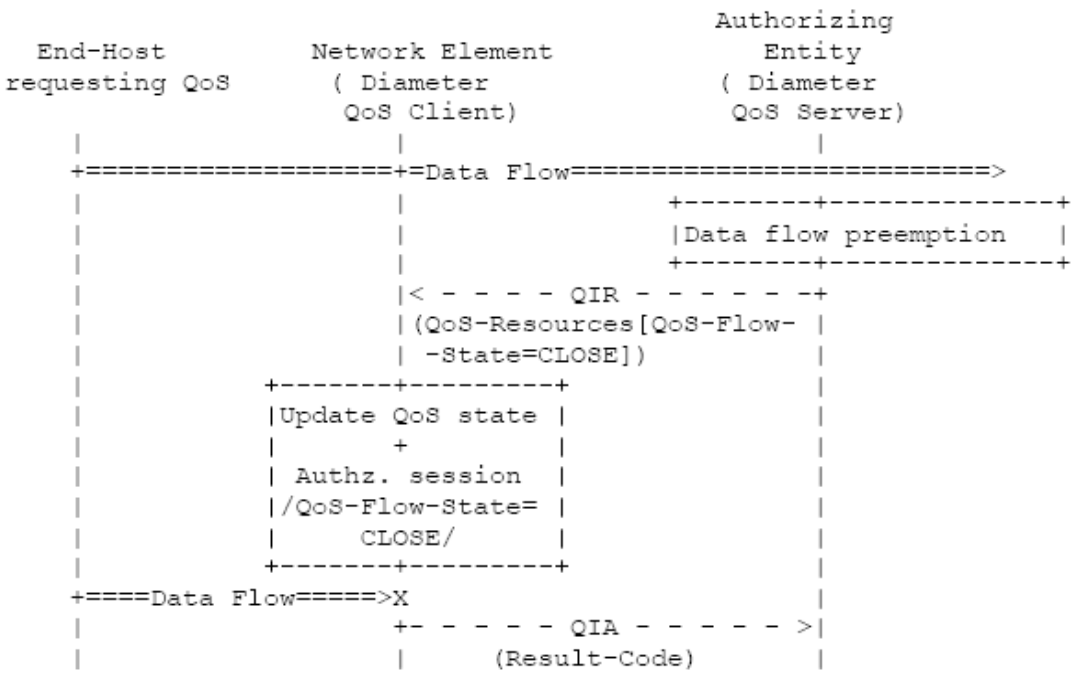

Figure 47 Server-side initiated QoS parameter provisioning [Alfano, 2006]

A.5.7 DHCPv6 application

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) application [Vishnu, 2006] provides an establishment of a Security Association between the HAAA and the DHCP server with Diameter. More details on DHCPv6 are given in [RFC 3315].

There are four messages specified in this application: *AAA-DHCP-Request* (ADR), *AAA-DHCP-Answer* (ADA), *Push-Configuration-Request* (PCR), *Push-Configuration-Answer* (PCA).

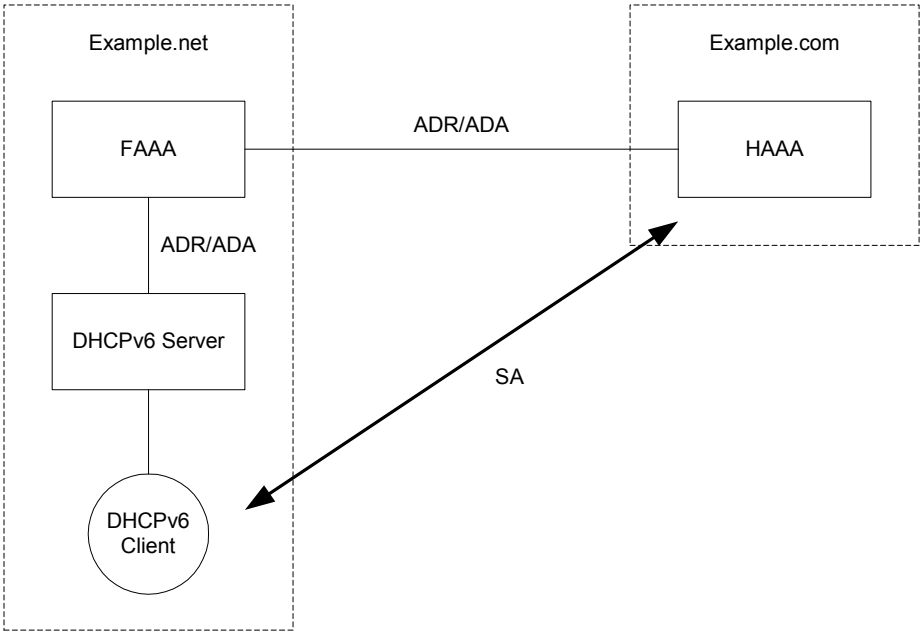

Figure 48 DHCPv6 architecture

In Figure 49 and Figure 50 the interaction diagrams of the Diameter DHCPv6 application is shown.

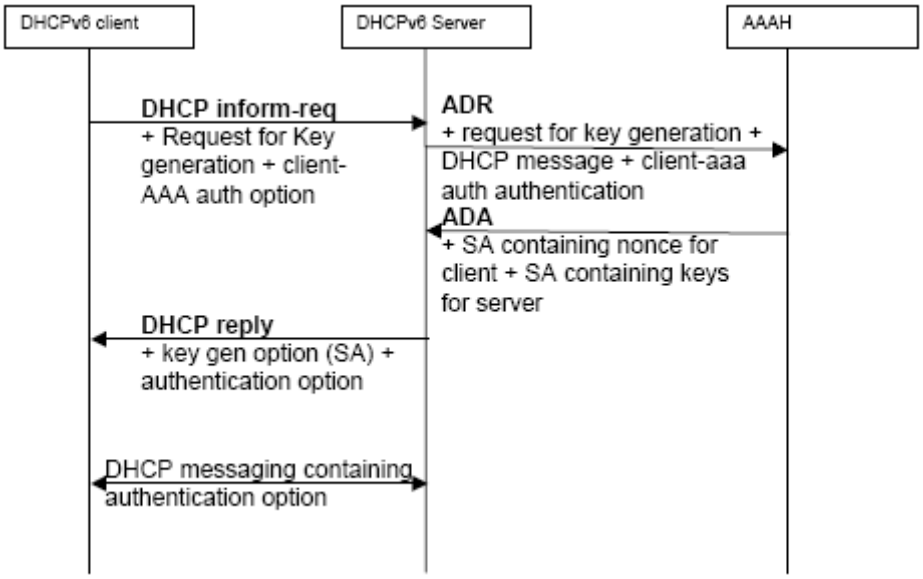

Figure 49 DHCP request [Vishnu, 2006]

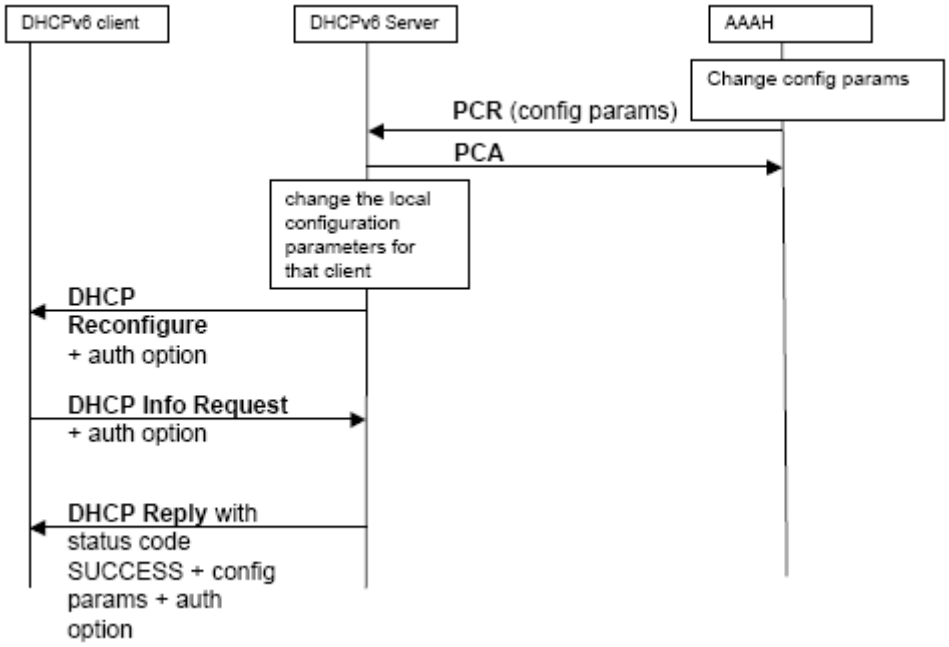

Figure 50 Server-side push configuration [Vishnu, 2006]

The DHCv6 Security Association is uniquely identified by the peer source and destination IP address and the Security Parameters Index (SPI). This is a connection between the DHCP client and the AAAH.

A.5.8 3GPP applications

In IMS, several interfaces are specified that use the Diameter protocol. The interfaces are defined as a Diameter application where the vendor is 3GPP. In the table below, the different interfaces are stated. Per interface the location (between functions) is given where it appears in IMS, what the general purpose of the interface is and in which document the interface is specified.

Table 4 3GPP interfaces [IANA, 2006]

Interface name	Location	Purpose	Specified in document
Cx	CSCF – HSS	Authenticating and authorization	3GPP TS 29.228 and 29.229
Sh	AS - HSS	User profiles	3GPP TS 29.328 and 29.329
Re/Rf	OCRP – RF	Charging	3GPP TS 32.296
Wx	HSS – AAA server	Charging WLAN	3GPP TS 29.234
Zn	BSF – NAF	Authentication	3GPP TS 29.109
Zh	BSF – HSS	Fetch keying	3GPP TS 29.109

		material	
Gq	PDF – AF	Not in release 7	3GPP TS 29.209
Gmb	GGSN – BM-SC	Exchange MBMS service control information	3GPP TS 29.061
Gx	PCRF – PCEF	Flow based charging GPRS	3GPP TS 29.210
Gx over Gy		Online charging GPRS	3GPP TS 29.210
MM10	MMS relay - MSCF	MMS	3GPP TS 29.140
Rx	CRF – AF	Charging	3GPP TS 29.211
Pr	PNA – AAA server	Presence I-WLAN	3GPP TS 29.234

Appendix B Diameter commands

In the following table an overview of the Diameter commands is given. These commands are standardized in the RFCs that form the Diameter specification. The table contains the full message name and its abbreviation, followed by the command code that is used in the command code field of the Diameter header. In the last column the RFC number is given in which the command is defined.

Table 5 Overview Diameter commands

Message name	Abbreviation	Command code	Defined in RFC
Capabilities-Exchanging-Request	CER	257	3588
Capabilities-Exchanging-Answer	CEA	257	3588
Device-Watchdog-Request	DWR	280	3588
Device-Watchdog-Answer	DWA	280	3588
Disconnect-Peer-Request	DPR	282	3588
Disconnect-Peer-Answer	DPA	282	3588
AA-Mobile-Node-Request	AMR	260	4004
AA-Mobile-Node-Answer	AMA	260	4004
Home-Agent-MIP-Request	HAR	262	4004
Home-Agent-MIP-Answer	HAA	262	4004
AA-Request	AAR	265	4005
AA-Answer	AAA	265	4005
Abort-Session-Request	ASR	274	4005
Abort-Session-Answer	ASA	274	4005
Accounting-Request	ACR	271	4005
Accounting-Answer	ACA	271	4005
Re-Authentication-Request	RAR	258	4005
Re-Authentication-Answer	RAA	258	4005
Session-Termination-Request	STR	275	4005
Session-Termination-Answer	STA	275	4005
Credit-Control-Request	CCR	272	4006
Credit-Control-Answer	CCA	272	4006
Diameter-EAP-Request	DER	268	4072
Diameter-EAP-Answer	DEA	268	4072
Location-Info-Request	LIR	285	4740
Location-Info-Answer	LIA	285	4740
Multimedia-Authentication-Request	MAR	286	4740
Multimedia-Authentication-Answer	MAA	286	4740
Push-Profile-Request	PPR	288	4740
Push-Profile-Answer	PPA	288	4740
Registration-Termination-Request	RTR	287	4740
Registration-Termination-Answer	RTA	287	4740
Server-Assignment-Request	SAR	284	4740
Server-Assignment-Answer	SAA	284	4740
User-Authorization-Request	UAR	283	4740
User-Authorization-Answer	UAA	283	4740

Appendix C Identity management

There are several different aspects of the work done at Liberty Alliance, but the part most interesting for this research is the identity federation framework. This framework enables the establishment of a circle of trust. “A circle of trust is a federation of service providers and identity providers that have business relationships based on Liberty architecture and operational agreements and with whom users can transact business in a secure and apparently seamless environment.” [Liberty, 2005]

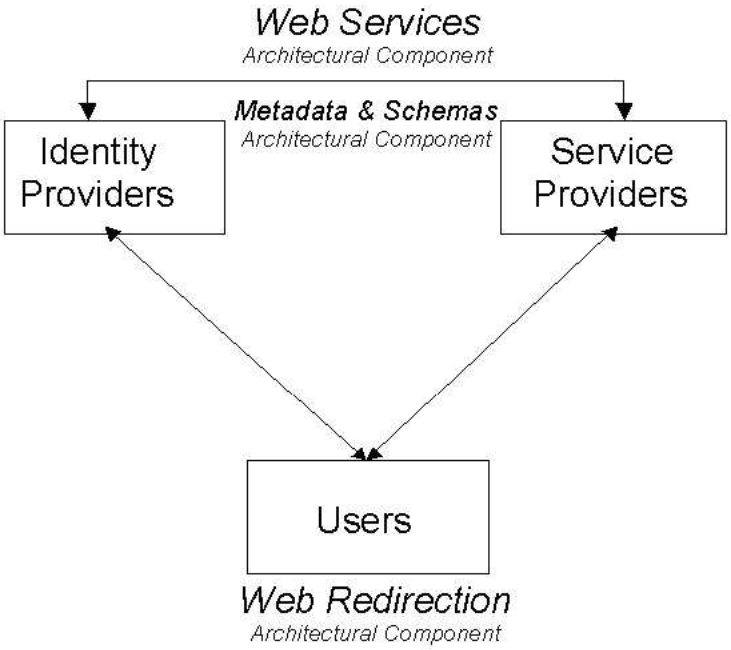

Figure 51 Liberty architecture [Liberty, 2005]

As can be seen in Figure 51, the important actors are the user, service provider and the identity provider. The identity provider is the entity that maintains the user’s credentials can verify the identity of the user. When the user wants to use the services of the service provider, it must use the token from the identity provider to authenticate itself at the service provider.

There are two alternatives when a user has more devices on which he has an identity, for example a mobile phone and a television. In this case there are multiple identity providers that need to be interconnected when a service handles both devices.
In Figure 52 the first alternative for Joe is shown of two identity providers in parallel. The service provider knows under which alias the user is known at the other identity providers.

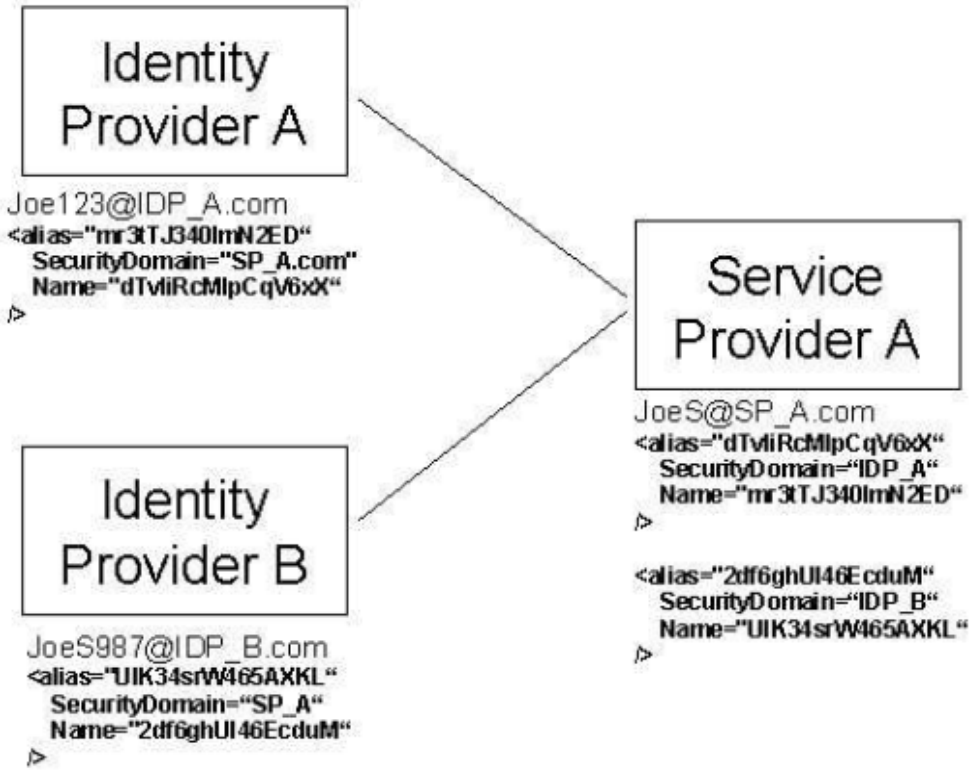

Figure 52 Two identity providers federated to a service provider [Liberty, 2005]

In Figure 53 the second alternative, two identity providers in series, is shown. The two identity providers are federated. Identity provider A is acting as both a service provider and an identity provider.

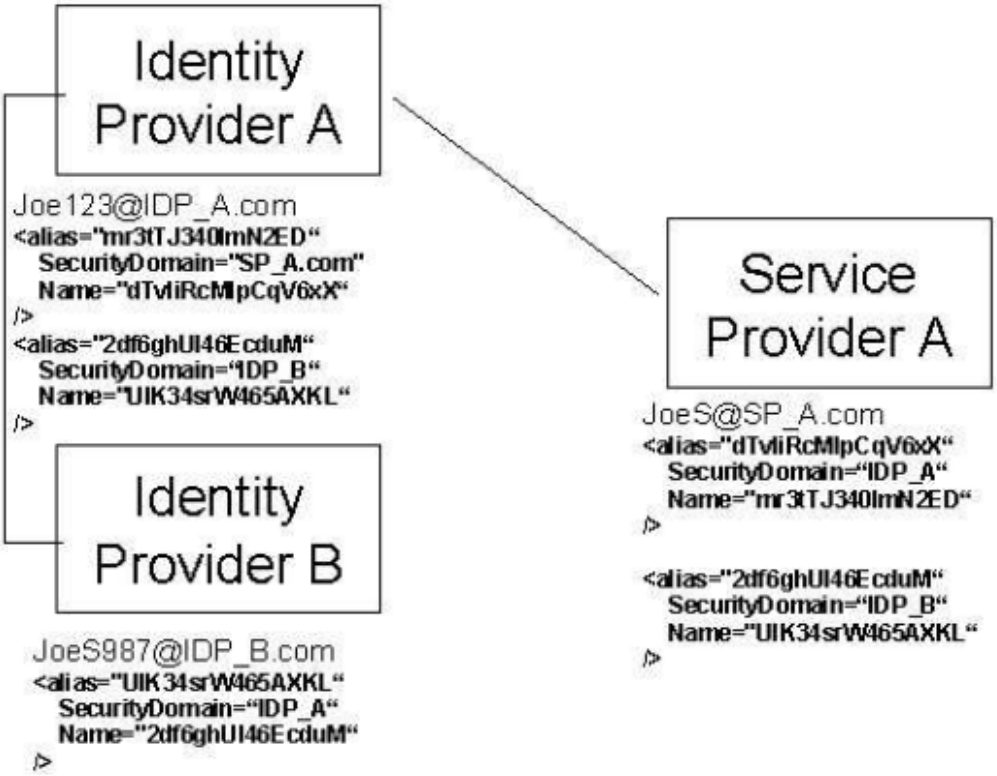

Figure 53 Two federated identity providers [Liberty, 2005]

When applying the theory from above to the problem of the thesis, identity management can be arranged as follows: The AAA server of the interaction manager can collect the ID's under which the user is known, from all other AAA servers that act as identity provider for the user, following the architecture of Figure 52. The alternative is the federated identity providers as shown in Figure 53. The both implementations for the FoneFreez case are shown in Figure 54 and Figure 55.

In the first alternative the AAA server of the interaction manager functions as a service provider that collects all the identities of the identity providers around itself. The AAA server of the interaction manager can verify if the user has accounts at both services and return to the AAA server of the application server so the user can be authorized.

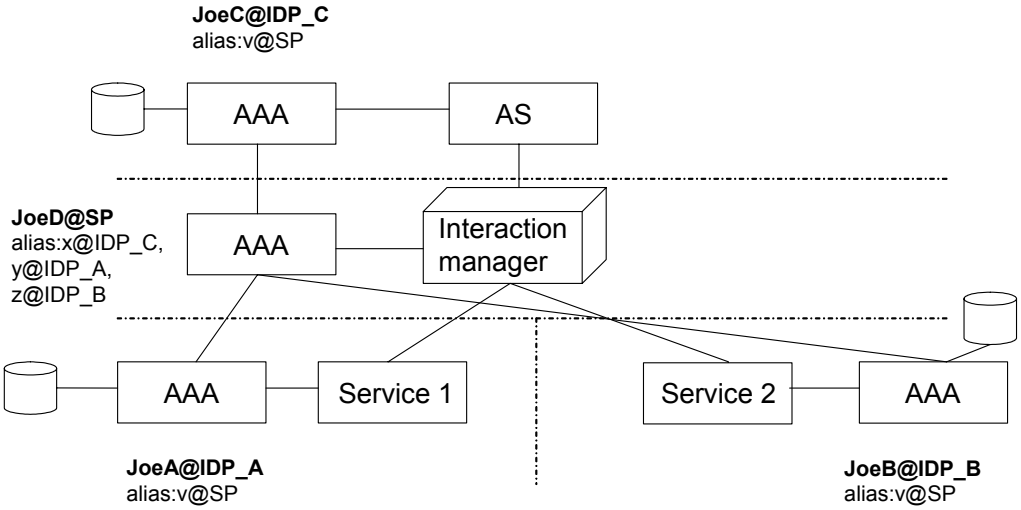

Figure 54 Alternative I

In the second alternative, shown in Figure 55, all the AAA servers function as identity providers that pass through the identity of the user to the application server. The application server in this case functions as a service provider. From that moment the application server knows under which identity the user is known in the different domains. This can be used to verify if the user has an account at both service providers.

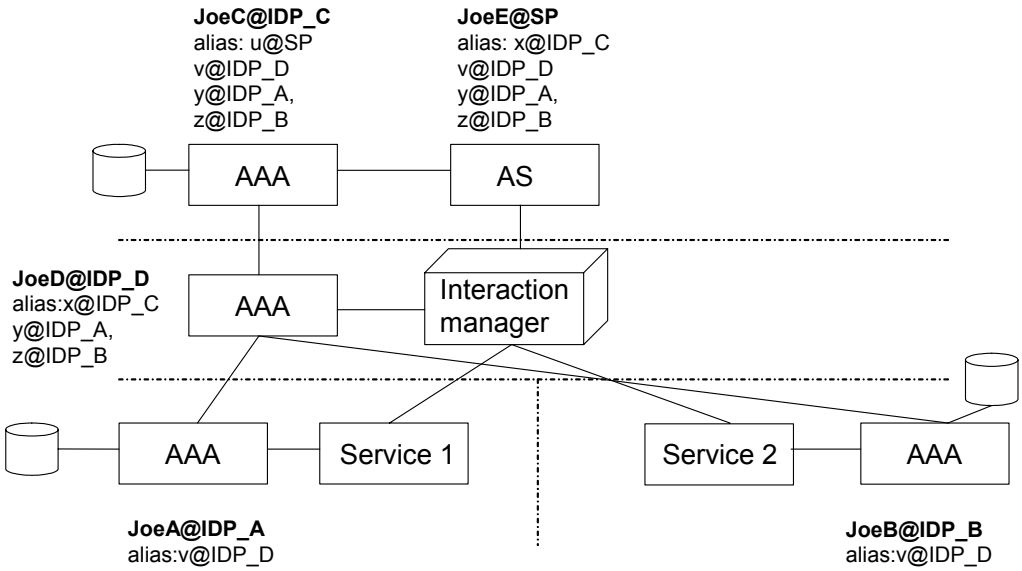

Figure 55 Alternative II

Alternative I is thought to be the best alternative, otherwise the application server must maintain a lot of knowledge about the users identity at the different places. This is not favorable because the application server gains too much information and is able to abuse its position.

To clarify what happens after the identities are exchanged, we show the implication for Diameter for the situation of Figure 54. When assuming that all identities have been exchanged, the user can be authorized to use the interaction service at the different components. The exchange is shown in Figure 56. The username is changed to the username known at the next component. In this way the authentication and authorization between domains is realized at the initialization phase.

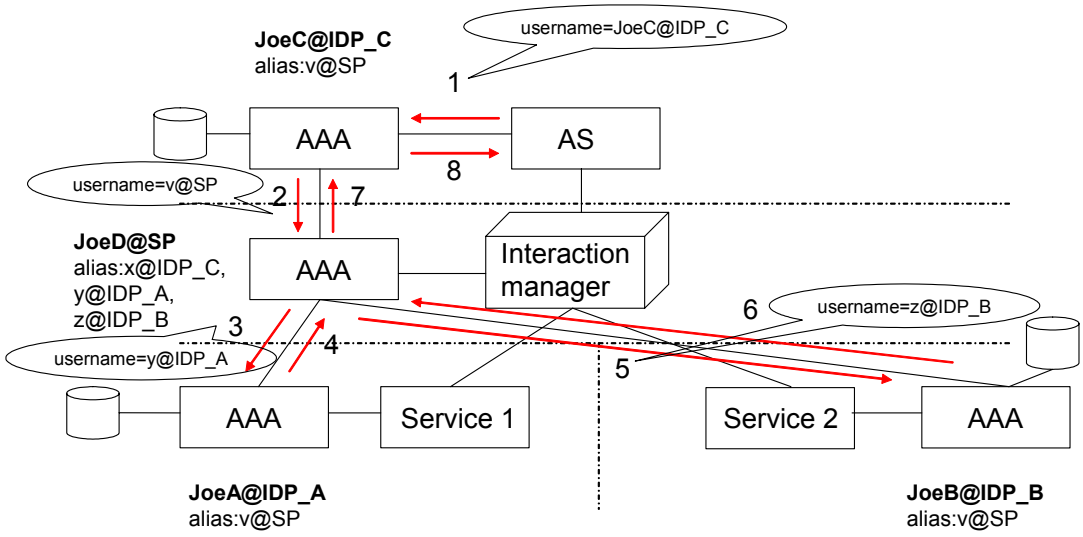

Figure 56 User authentication

Appendix D Path to solution

In this chapter the path is described to come to a complete design for AAA in service interactions from different realms. First the problem is simplified by looking at service interaction in one realm. Next the problem is refined by adding realms until the situation contains for every role a different realm. This methodology is described in section 2.6. The roles are implemented as described in subsection 4.4.

D.1 One realm situation

To come to a complete architecture first the simplest situation is considered. This situation is shown in Figure 57. Here all the services belong to a single actor and reside in the same realm. The application server, the interaction manager and the services platforms like the IPTV and IP telephony service are located in one domain.

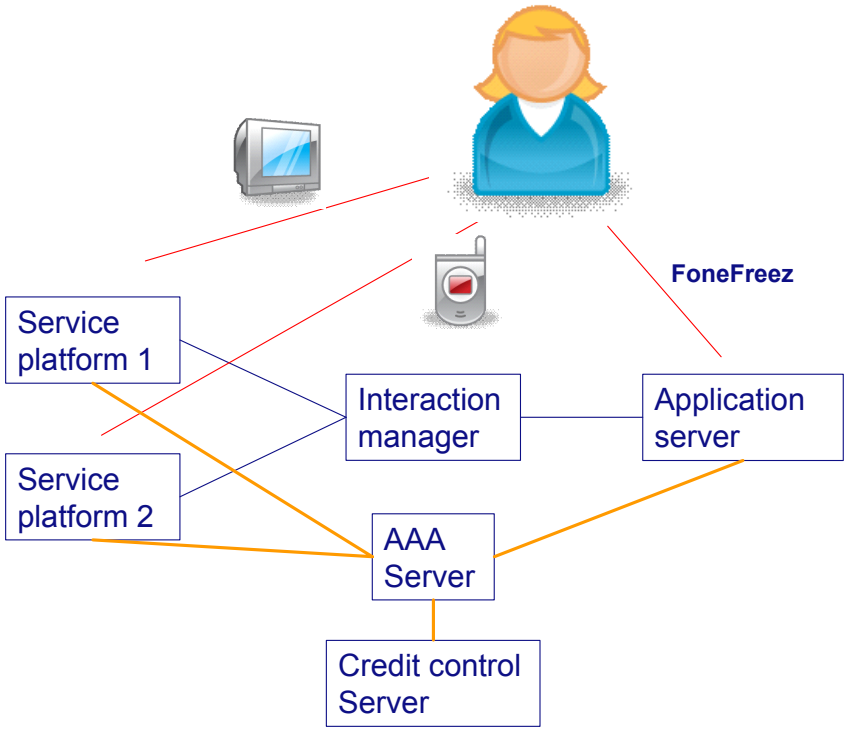

Figure 57 AAA in one domain

The AAA added in this situation consists of one AAA server that authenticates and authorizes the user and stores the user's credentials. Also a Credit control server is added to enable real-time credit authorization for the user.

The user registers with service provider 1 and service provider 2, this can be done under the same username and password. When the user wants to register with the interaction service, the application server contacts the AAA entity and the user is authenticated and authorized to use the interaction service. Because all the

information about the user runs through the same AAA entity, no issues about identity management appear. The AAA entity can authorize the user for the interaction service because it knows if the user is also registered with the proper IPTV and IP telephony service.

Accounting is done at the service platforms and at the application server. In this way the usage by the user can be monitored, but also the metering is done for the usage between the entities. Because they all reside in the same domain, and trust is assumed between the entities, the interaction manager does not need to meter for accounting itself. The service platforms and application server do not need direct contact with the credit control server as described in subsection 6.2.2.

In this situation the user can still use the services without the interference of the broker. The broker is allowed to intervene, because there are no issues of trust between the different services while they reside in the same domain.

D.2 Services in different realms

In this section the situation is considered that the application service provider and broker reside in the same domain, but the IPTV and IP telephony services have both their own realm. For this situation some alterations have to be made with respect to the situation where all business roles where fulfilled by the same actor, as described in the previous section.

To enable the use of the services without the interference of the broker, both need their own AAA entity to authenticate and authorize their users. Furthermore the application server needs its own AAA entity to authenticate and authorize the users of the interaction service. In this situation there must be negotiation on the identity of the user. It is possible that the user is known under different usernames at the three domains.

In the next two figures, the application service provider and broker reside in the same domain. In the first situation shown in Figure 58, the application service provider and broker are located in the same realm, while the services both reside in their own realm. The AAA entities are interconnected.

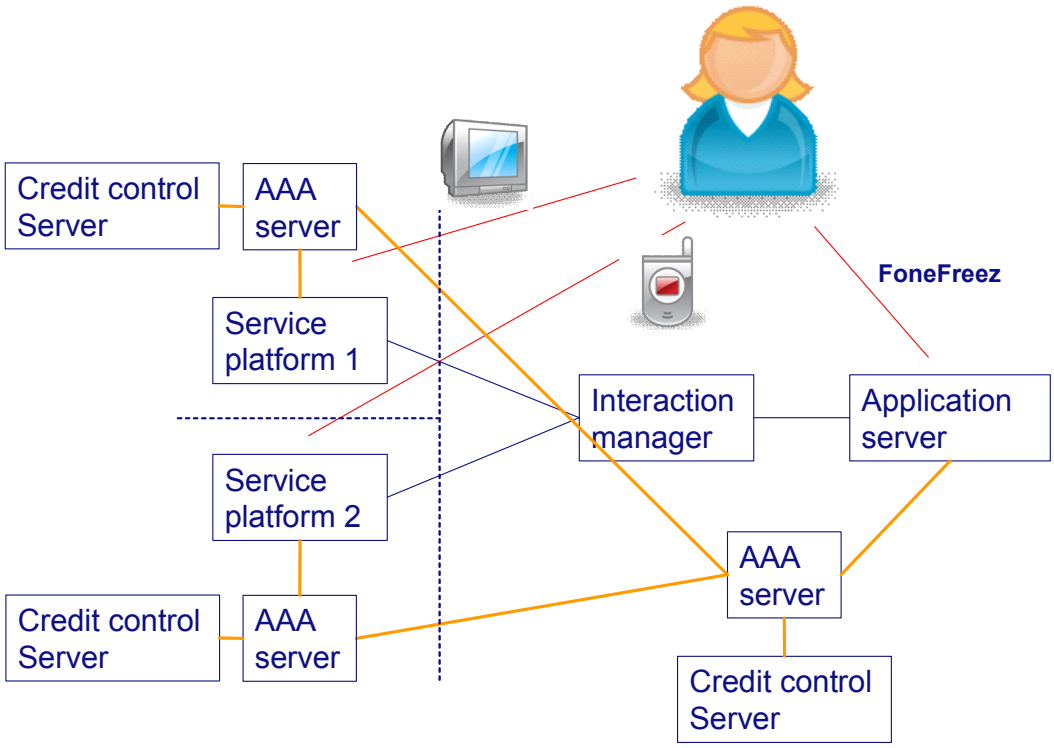

Figure 58 Services in different domains, connected AAA

An alternative design for this situation is shown in Figure 59. The different alternatives come from the possible difference in architecture as described in subsection 6.2.3.

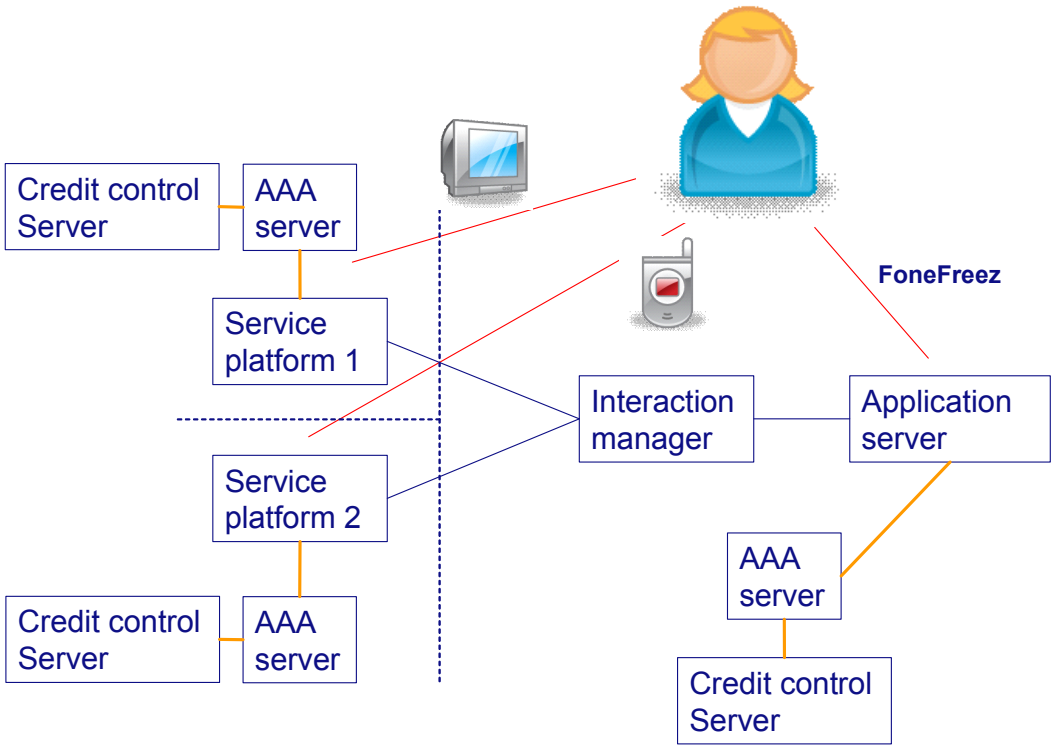

Figure 59 Services in different domains, separate AAA

The difference in design is mostly relevant at the initialization and log-on phase. The verification of the user at the services can be done using the AAA infrastructure (Figure 58) or through some interaction of the broker (Figure 59).

Because the broker and both services no longer reside in the same domain, there must be some kind of authentication and authorization of the broker, to justify its interference. The AAA entity of the broker can request authentication for the broker at the AAA entity of the both services as shown in the situation in Figure 58. Another option is that the service handles the authentication of the broker itself, but then it must maintain a list of brokers that are allowed to intervene as done in the situation of Figure 59.

Three credit control servers are needed to provide real-time credit authorization for the user. Metering is done at the service providers to enable charging the broker and application service provider for their usage of the service platforms. Metering at the broker is not directly needed because the broker receives money from the application service provider, and the broker and application service provider are implemented by the same actor in this section. The cash flow is described in subsection 6.2.4.

D.3 Application server in separate realm

The second situation is where the broker and the services reside in the same domain. When the application service provider resides in a different domain than the broker and the services, also some alterations are needed.

The application service provider needs its own AAA entity to authenticate and authorize its users. The services can share an AAA entity which contains the credentials of the users. In this situation the issue of identity management of the user is also relevant, because it is possible that the user is known under another identity at the application service provider and at the services. Here both services have the same identity for the user.

The first alternative is shown in Figure 60 where the AAA entities are interconnected.

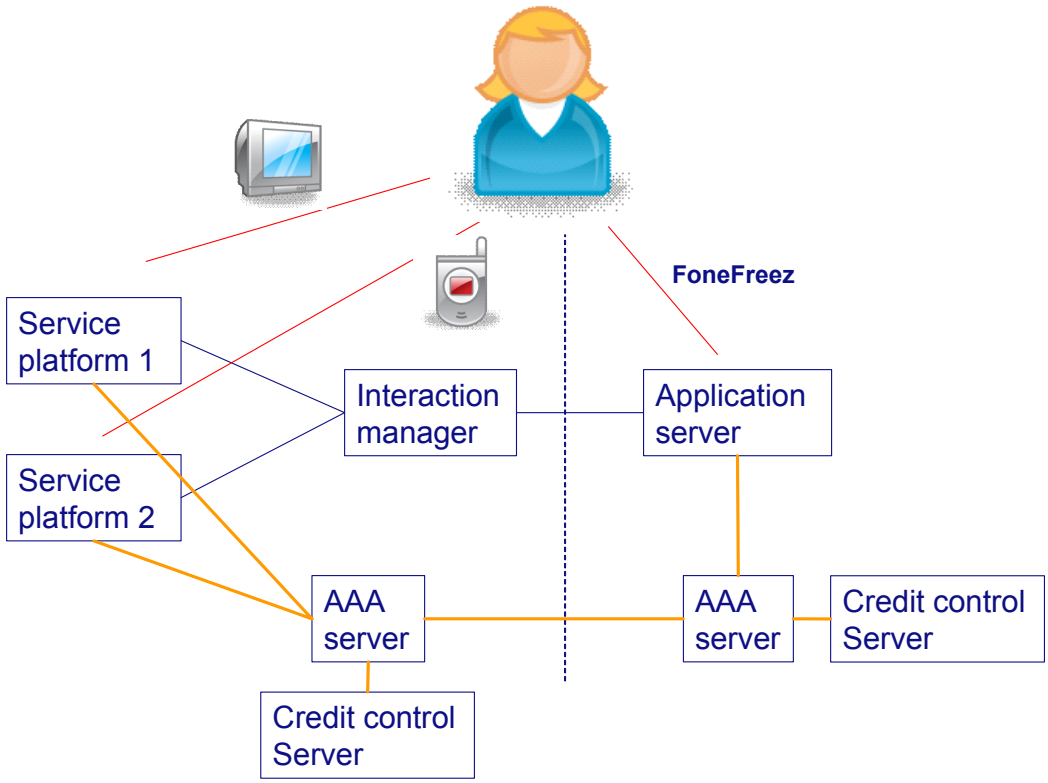

Figure 60 Application server in different domain, connected AAA

The second design is shown in Figure 61.

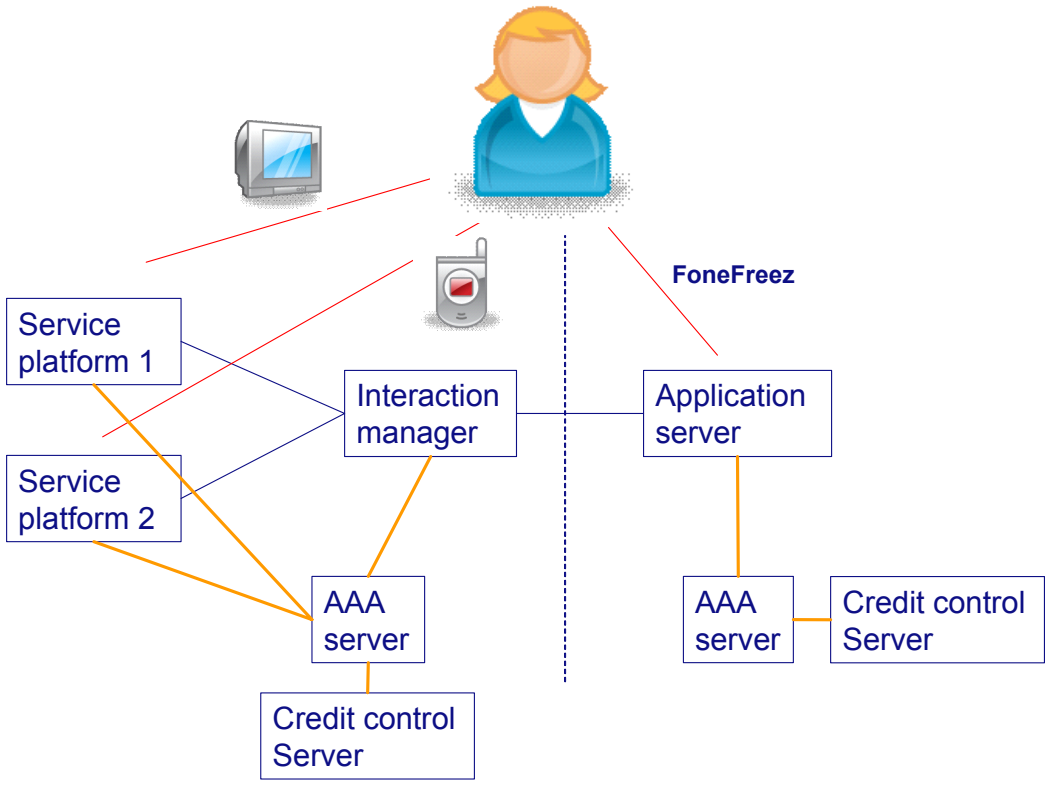

Figure 61 Application server in different domain, separate AAA

The application service provider needs to be authenticated to and authorized by the broker because it is not located in the same domain. The trust relation can be built between the different domains by the AAA entities (Figure 60) or by authentication at the broker (Figure 61). To enable this, the broker needs its own connection to the AAA entity.

The same difference as described in section D.2 is relevant here. In the initialization and log-on phase the AAA infrastructure can check if the users are registered at the services or the broker can verify this by contacting the services.

D.4 All business roles in different domains

For the final situation where all the business roles reside in different domains also two alternatives are possible, following from the different architectures as described in subsection 6.2.3. In Figure 62 the first alternative is shown, where the AAA entities are interconnected.

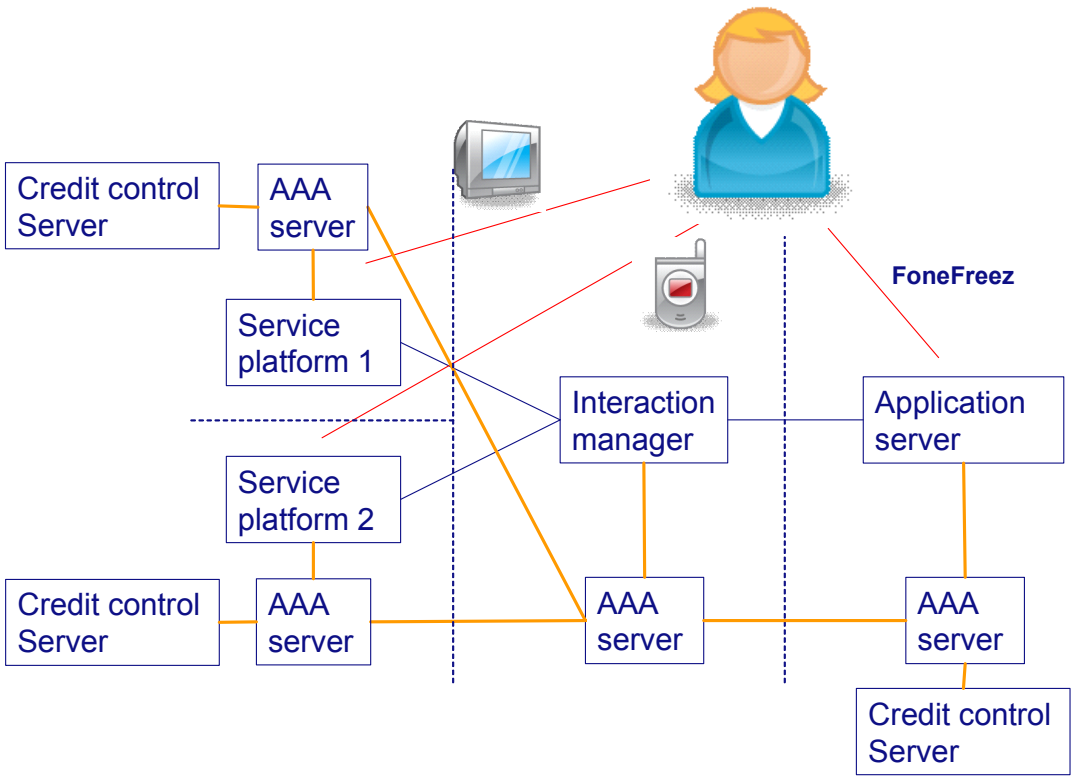

Figure 62 All business roles in different domains, connected AAA

The AAA entity in the broker's domain functions like a relay or proxy to enable the connection between the application service provider and the services. In this way the

trust relations can be built between the different parties and the identity of the user can be discussed, so they all know that they are talking about the same user.

The second alternative is shown in Figure 63.

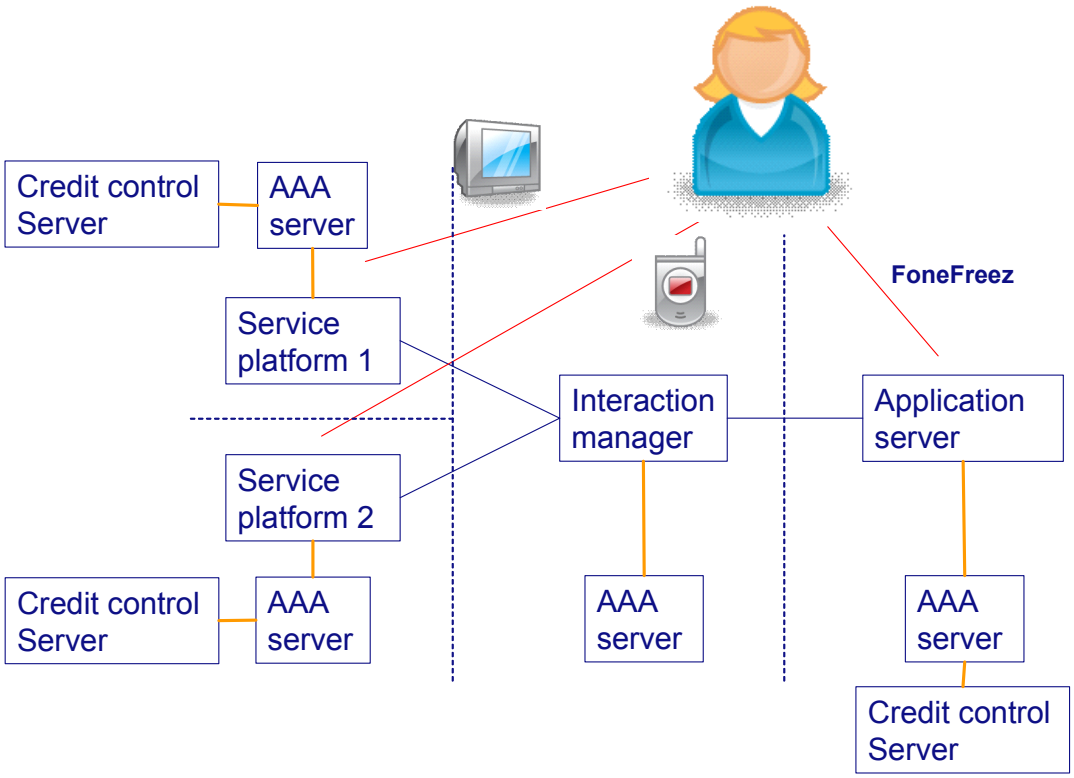

Figure 63 All business roles in different domains, separate AAA

It is possible that there is no connection between the AAA entities from the different domains. Then every domain must authorize an incoming message, if this is an allowed action. The identity exchange takes place over the normal connections and at the service provider it is verified if the user is also registered there.

The broker has no credit control server in this situation, because in the requirements is stated that only the user is accounted for on a pre-paid basis. The broker needs a third party to enable billing of the application service provider for the usage of the interaction manager. The accounting for the usage is done at the AAA servers. The third party billing provider is left out of the picture.

Appendix E Interaction diagrams

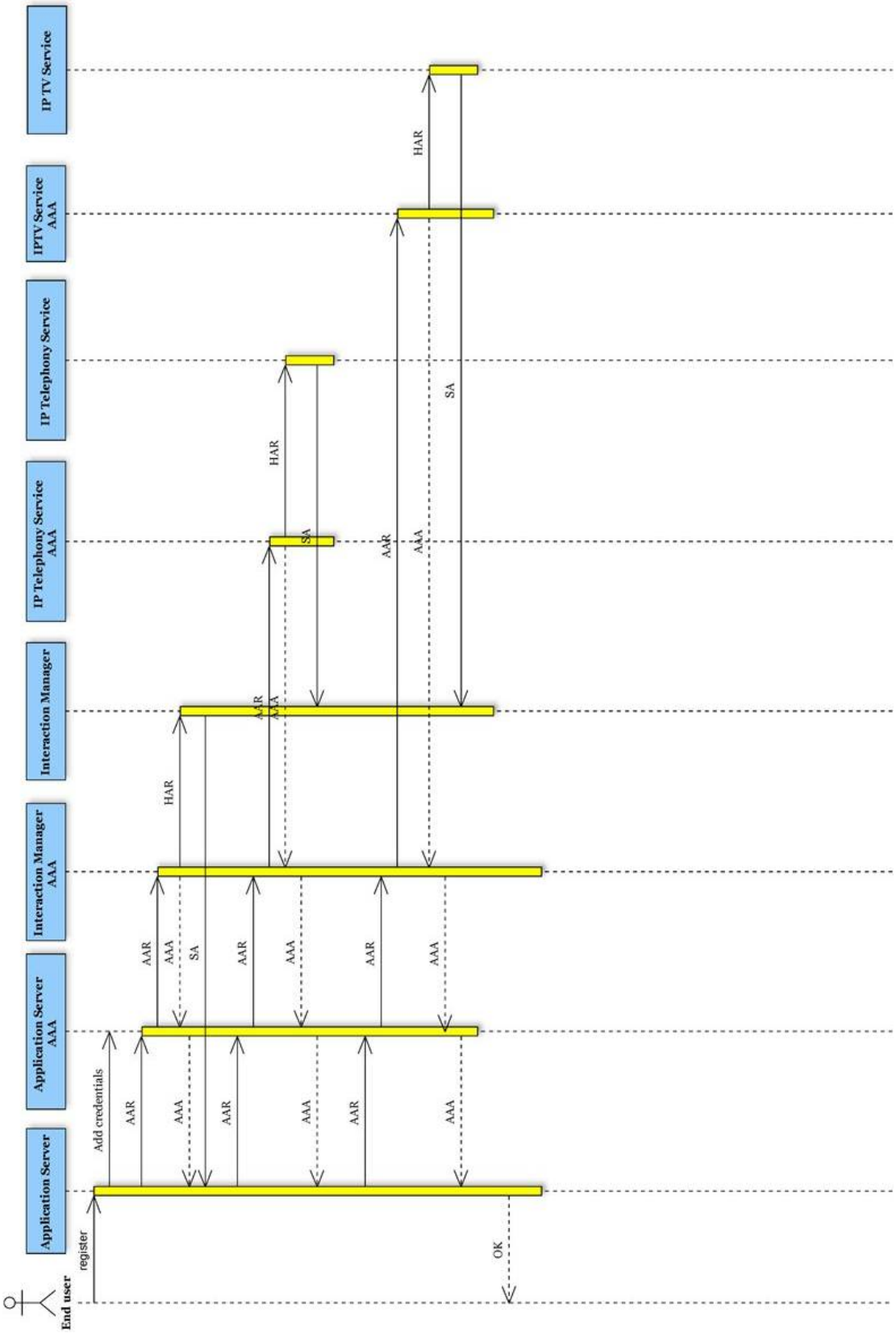

Figure 64 Registration phase end-to-end design

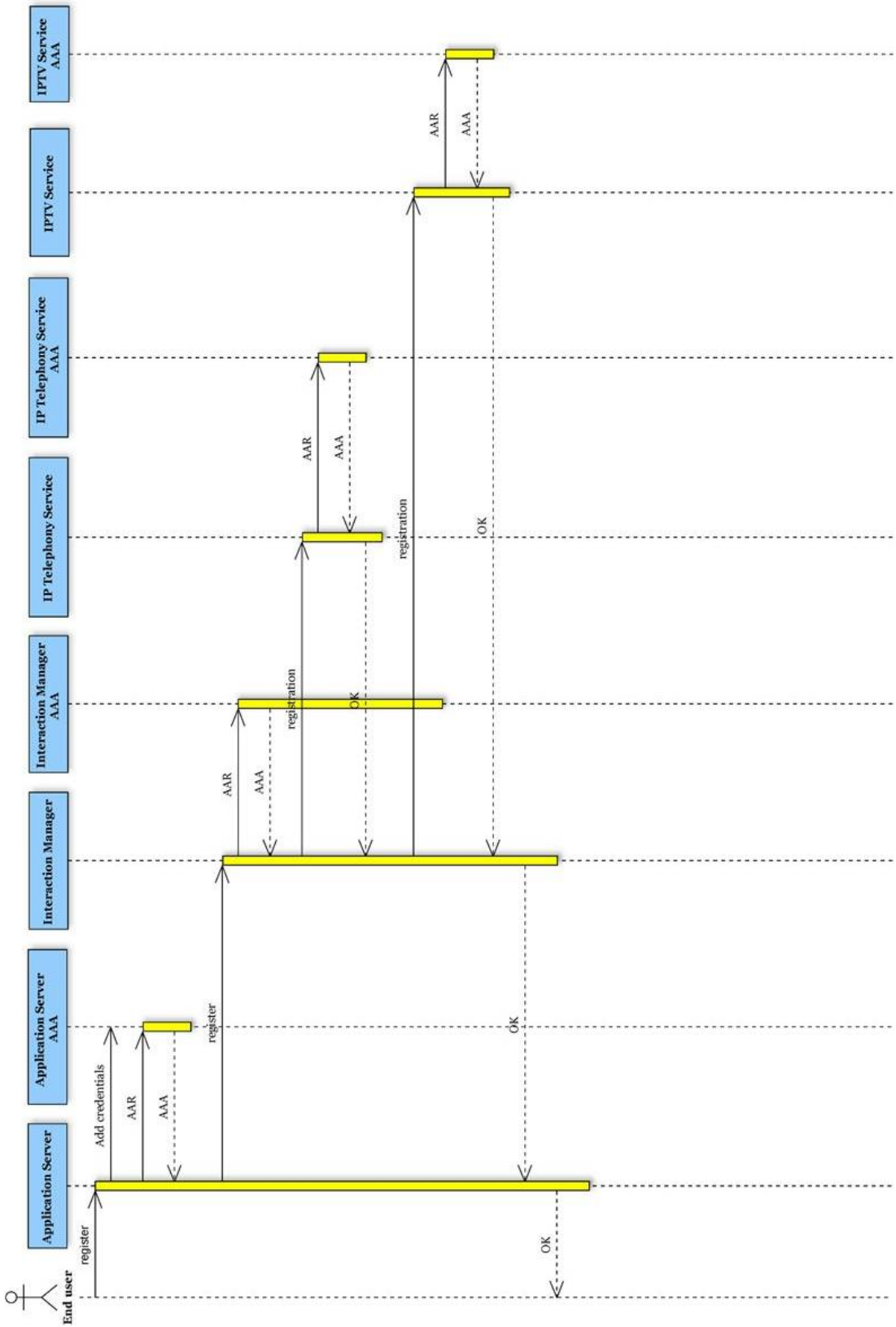

Figure 65 Registration phase hop-by-hop design

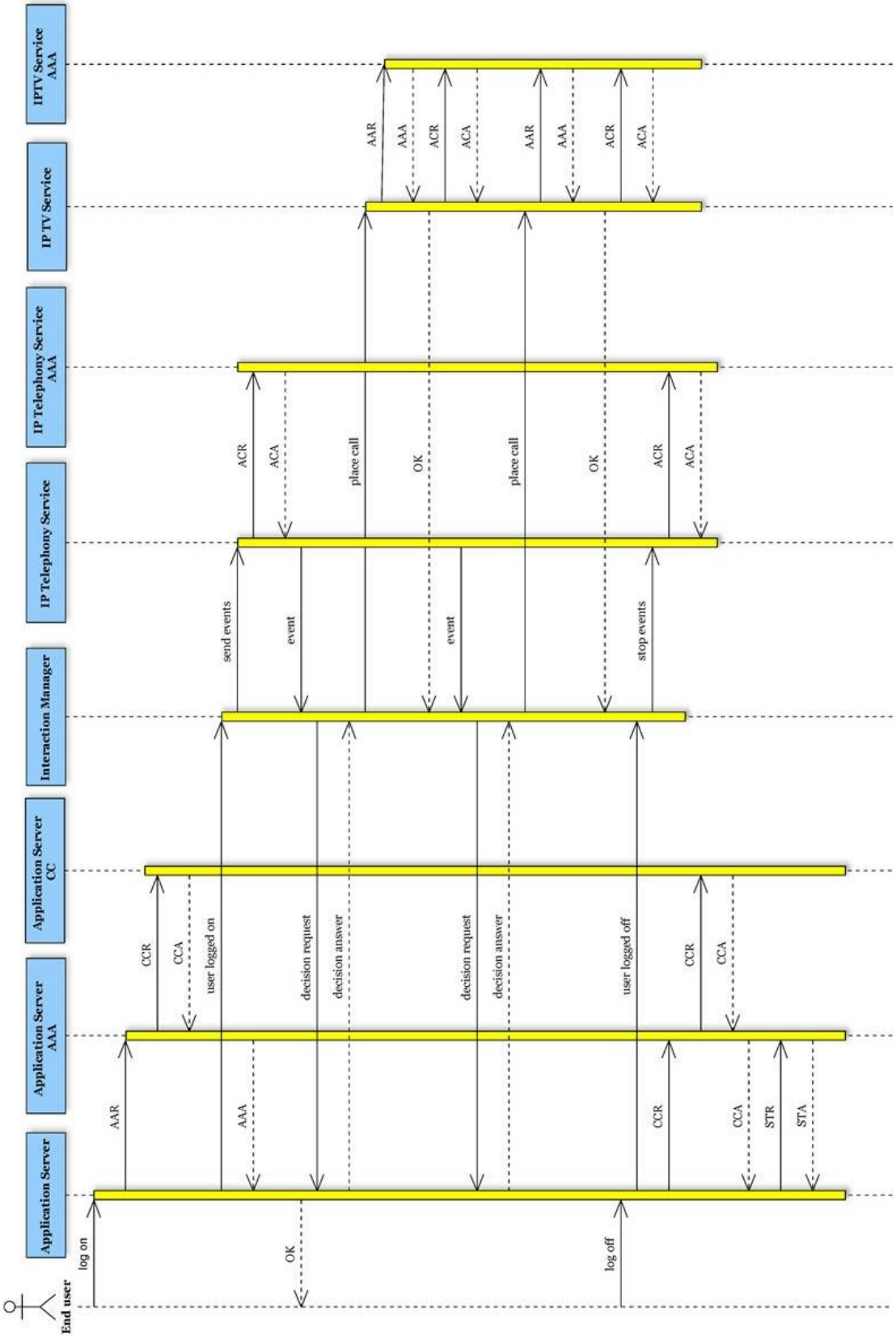

Figure 66 Operational phase

Appendix F Future for Diameter

To decide if Diameter is a viable protocol for future use, a model developed at TNO-ICT is used. The model identifies the key success factors of telecom standards [Sweers et al, 2007]. The factors are categorized in four clusters: standard, supporting firms, industry forces and market mechanism & environment. The model can be seen in Figure 67.

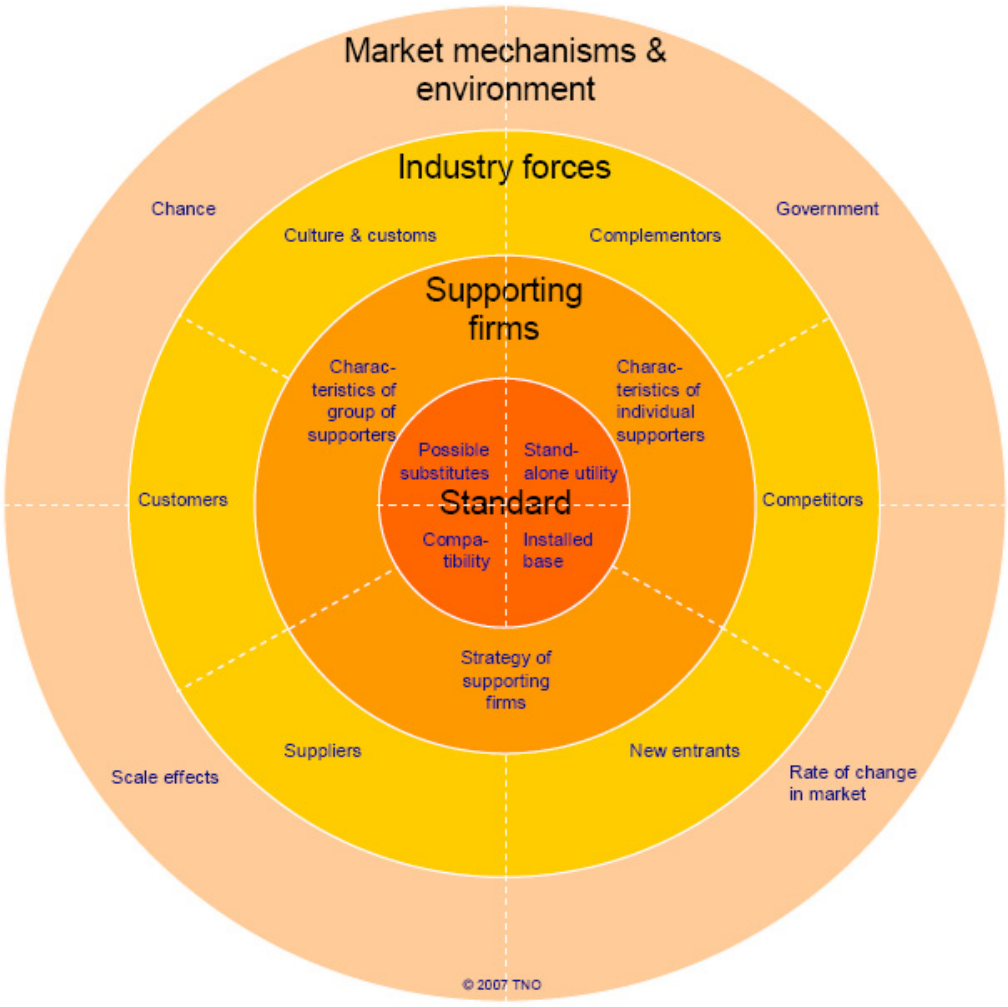

Figure 67 Success factors telecom standards [Sweers et al, 2007]

In the following sections the clusters are discussed, followed by an overall conclusion.

F.1 Standard

Four factors are identified that are directly related to the standard: stand-alone utility, installed base, compatibility and possible substitutes.

With respect to the stand-alone utility of Diameter, the Diameter standard outperforms other standards like RADIUS, as described in subsection A.4.1, but is a

protocol under development that is not yet complete. There are extensions of RADIUS that are not yet available for Diameter, like extensions for MIPv6.

The installed base of Diameter is not very large at the moment, but the installed base of its predecessor RADIUS is enormous. It is likely that the RADIUS users will switch to Diameter when implementing new AAA implementations.

Diameter is for the most part backward compatible with RADIUS as described in [RFC 3588]. It is also compatible with complementary standards like Mobile IPv4, SIP and EAP. The Diameter specification is very flexible and it consumes limited resources (time and costs) to develop new applications.

Possible substitutes for Diameter's network access functionality are: RADIUS and TACACS. With respect to the policy control functionality, COPS is a possible substitute. Switching to RADIUS involves limited costs, but with switching to the other protocols probably more costs are involved, for example because of more configurations on the network and services.

F.2 Supporting firms

This section discusses the factors for the directly supporting firms: characteristics of individual supporters, characteristics of groups of supporters and strategy of supporters.

The number of supporters of the Diameter protocol has recently increased, due to the adaptation of the Diameter protocol in IMS. 3GPP uses Diameter in its IMS interfaces, as is ETSI/TISPAN. 3GPP has some influential market representation partners like the GSM Association and hundreds of individual member companies [Sultan, 2006]. These are the characteristics of the groups of supporters of Diameter.

The individual supporters are specialized firms in telecommunication protocols. They try to influence the user by showing their products on worldwide congresses like 3GSM. There are a number of firms that have implementations available of the Diameter protocol stack and Diameter servers, e.g. Netbricks, Intellinet, Marben, HP Opencall etc.

The strategy of the supporters is not yet available for study. Because Diameter products recently entered the market, not much can be said about timing of entry decisions, pricing strategy and licensing strategies and marketing communications.

F.3 Industry forces

The industry forces originate from the firms that deliver complementing goods or services, competitors, new entrants, suppliers, customers and the dominant culture and customs in the industry.

The firms that deliver complementing goods or services are the firms that for example produce SIP products, where Diameter is often used for authentication and authorization. Another group of complementing goods is hardware on which the Diameter server is to be installed.

Competitors are firms that produce smartcards or other authentication mechanism other then Diameter, like Kerberos. Also firms that produce RADIUS products are compatitors, but mostly those firms also produce Diameter products.

For new entrants, barriers to entry are small in the case of Diameter. They must have knowledge of the open standard and be able to produce an implementation. It will be likely that new entrants will enter the market as long as this market is not yet saturated with Diameter products.

There are numerous suppliers that adhere to the standard, or claim to adhere to the standard. No tests by independent institutes are done yet to check if they are actually compliant to the standard. Because IMS is believed to be the next hype in the telecom market, the suppliers trust the commercial success of the services based on the standard.

The need of the customer is mostly Diameter for IMS solutions, but some customers do also want Diameter products to solve simple authentication issues in their networks. The tendency of customers to adopt Diameter because others did, is likely to be high. While IMS is hyped more customers believe they need IMS, which include Diameter products.

The dominant culture and customs in the industry also influence the adoption process. The widely adopted RADIUS protocol, will tribute to the success of

Diameter, because those customers are more likely to also adopt Diameter. Secondly, the firms are able to generate extra profit by using this standard, e.g. through IPR on Diameter products.

F.4 Market mechanism & environment

The factors that have impact in the market mechanism and environment are: government, rate of change in the market, scale effects and chance.

The standard is not prescribed by policy makers and therefore not enforced by the legislators.

The change in users needs is very high, because every time a new technology enters the market new AAA solutions are needed. Diameter is designed with respect to possible new technologies and explicitly made flexible to enable Diameter to also provide AAA for these technologies.

Diameter is, like RADIUS, developed to handle a great amount of users, and will scale very well. Economies of scale, which means that the costs of the standard decreases with the number of users of the standard, is valid for Diameter.

F.5 Conclusion

Where Diameter was not considered to be the best solution in several comparisons in the past, it is at the moment. This is because the main objective was that there were not enough implementations of Diameter and that not every part of the specification was finished [Eertink et al, 2005a; RFC 3127]. Due to the adoption of Diameter by 3GPP, the production of Diameter products and the development of extensions on the Diameter protocol experienced an explosive growth.

As long as IMS becomes widely adopted, the success of Diameter will grow. But the success of Diameter is not only dependent on the success of IMS. It is now mature enough to survive on its own as a good replacement of the RADIUS protocol.