SOA in the CoNSIS Coalition Environment

Extending the WS-I Basic Profile for using SOA in a tactical environment

Hartmut Seifert, Markus Franke
Dept. Networks and Architectures
Industrieanlagenbetriebsgesellschaft mbH
Ottobrunn, Germany
seifert@iabg.de

Anne Diefenbach, Peter Sevenich
Communication Systems Group
Fraunhofer FKIE
Wachtberg, Germany
anne.diefenbach@fkie.fraunhofer.de

Abstract—**This article describes the elements necessary to allow SOA based CCIS systems to operate in a mobile tactical environment. All elements which are mandatory to allow a SOA based implementation to react to bandwidth limited, jammed and temporarily unavailable network connections and to span a common information domain across various coalition partners are listed in comparison to the WS-I Basic Profile.**

Keywords-autarchy, non-hierarchical, fully distributed, SOAP-based security mechanisms, schema-based compression, NNEC

I. Introduction

Future Command, Control and Intelligence Systems (CCIS) will not be developed from scratch anymore. Military operators are interested in specific military functionalities, and they do not care about how these functions communicate and cooperate with each other as long as a defined service level agreement is fulfilled.

One approach to realize such a system is to base it upon a service oriented architecture (SOA), where military applications will use commonly available core services (like basic messaging or repository services) to provide their capabilities to human operators.

This decomposition is originally derived from the NATO Network Enabled Capability (NNEC) feasibility study [1], where various steps are described to come from the current position of stove-pipe systems to a coalition wide shared services approach.

The SOA paradigms, and the tools and frameworks developed to help implement them, are intended for use in fixed, broadband company networks. Web services, the technology commonly used to realize a SOA (and the one suggested by NNEC), along with their numerous advantages also come with some disadvantages, such as a very large overhead. Military networks, however, often do not meet the conditions expected in civil company networks. Military networks in the tactical domain, especially mobile networks, suffer from low bandwidth, large delays, frequent disruption and the threat of jamming. Therefore, steps need to be taken to mitigate these disadvantages and ensure adequate performance. This paper takes the Web Services Interoperability (WS-I) Basic Profile [2], developed by an industry consortium to ensure interoperability between web services, and suggests several additions to be prescribed to web services in a military environment. The resulting architecture is described in principle and verified within a national implementation, the Reference Environment for Services “Referenzumgebung Dienste” (RuDi).

RuDi was the German SOA framework used in the international project CoNSIS (Coalition Networks for Secure Information Sharing). Work in CoNSIS, performed in five distinct tasks, aimed to develop a comprehensive, federated environment comprising heterogeneous networks from different nations in which to securely share information. This article is based on work carried out in Task 2, the Information Services task, concerned with connecting SOA frameworks from different nations – namely Germany and Norway.

II. Limitations for SOA in mobile Systems

A. Mobile Node Autarchy

In battlefield scenarios, such as a convoy operation as assumed in the CoNSIS scenario, mobile units may become cut off from communication with an upper command level (e.g. HQ). In this case, the units still need to be able to complete the operation they were sent out on – which means that any necessary information or supporting routine must be available locally – in our scenario, within the convoy. The SOA framework must be set up in such a way as to be able to provide the required services to the users (consumers) with the best quality of service achievable under current conditions. In the most extreme case, this means that all critical services must be completely available within each node (e.g. vehicle within the convoy), as stipulated in [3].

B. Bandwidth Limitations

The communications within a mobile environment is the most limiting factor for SOA: Military communications are either radio based (traditional or ad-hoc radio networks) or supported by tactical satellite communications. In both cases, the available bandwidth is significantly lower than in backbone systems (from a couple of kilobits per second up to a lower megabits per second rate, which is shared between a number of nodes). In comparison to several hundred megabits per second up to a few gigabits per second for stationary systems, this is a severe limitation.

To cope with these limitations, several optimizations are necessary within the WS-I Basic Profile:

- Services should, where possible, be executed on local nodes. A remote invocation of services (e.g. within a portal) consumes too much bandwidth.
- The main information exchange between military users is message oriented. For this purpose, a standard SOA service, a notification broker, is used to provide one or more users (consumers) with the same kind of information. To save bandwidth, the notification broker has to use the broadcast capability of radio networks to distribute the same information to a number of consumers in a multicast mode (necessary multicast extension to the notification broker). To enhance the performance further, it is recommended to distribute theses multicast messages in simplex mode (without acknowledgement transfers from the recipients). This avoids unnecessary send/receive mode changes in the involved radio systems.
- XML coding of messages within a SOA environment is not bandwidth efficient. To reduce the network load, message compression is required. To allow a maximum compression, the usage of schema-aware algorithms is highly recommended. Efficient XML Interchange (EXI) [4] is the W3C recommendation, and the more experimental mechanism XENIA [5] achieves compression results of about 97% of the original message size.
- In a highly dynamic environment, service availability will change heavily over time. To provide users with the most recent service availability under current restrictions, service discovery mechanisms need to be able to cope with these frequent changes. CoNSIS uses WS-Discovery [6] with a few extensions, as described more fully in [7]. RuDi uses this protocol proactively to report periodically about the service availability in different nodes. However, as this service generates a significant load within the radio network, administrators have to carefully decide how to set the period of retransmission of WS-Discovery messages.

C. *Secure SOA in a coalition environment*

Obviously, a military network has to be secured to protect the confidentiality, integrity and authenticity of the data. Here too, the WS-I offers a profile to specify how the different web service security standards can be made to work together: the Basic Security Profile [8]. However, a military environment places more strident requirements on security than a civil one, and moreover the security measures usually need to be formally accredited. So web service security can only serve as an additional safeguard, superimposed on traditional, accredited measures. But SOA, with its highly distributed architecture, also introduces new challenges. Within a single, national implementation of a SOA environment a hierarchical model with well-defined roles and access rules may work. In a coalition environment, the various information domains may overlap various technical domains (here used as a synonym for national domains). To operate within such an environment, the following assumptions are made:

- The basic protection of information is being done at network level. Modern approaches like HAIPE 3.x or NINE 2.x are used to encrypt IP packets at the network layer (hopefully end-to-end, realistically within a single technical domain).
- To protect messages individually, all information types and services are encapsulated within a SOAP message and protected individually on their route between the involved communication partners by their individual certificates. This allows a specific protection and encryption of SOAP messages, based on the individual conditions of the participating roles. In addition, these SOAP messages are enhanced by XML labels to allow label switching at any domain boundary between participating partners. This way, differently classified information domains can be interconnected. To support the exchange of protected information across domain boundaries, cross domain certification is required.

III. Principles of RuDi Realisation

To cover the previous limitations, the German SOA environment in CoNSIS, RuDi, was specified and realized in a specific way. The design principles are introduced in this section.

A. *Principles for Service Access*

To get sufficient information about available services in a mobile node, each user (consumer) has to contact his local service registry. This registry informs the user under which conditions (including the bandwidth limitations to access a remote service) a service can be used.

To generate and maintain the information on different radio links, methods like PPPoE [9] for the calculation on physical link conditions and multi-topology routing (MTR) [10] to manage end-to-end conditions within the radio network are used.

Figure 1 shows that the principles of cross-domain service invocation as realized in CoNSIS are the same as with local service invocation. To invoke a service, a consumer first contacts their local service registry (part of the local SOA infrastructure) to find an appropriate provider. The registry lists all service providers available to consumers in their domain, no matter whether those providers are located in the same domain or different ones. To achieve this, synchronization between infrastructures across domains is mandatory. This can be realized either by a peer-to-peer synchronization (SyncD, for details see STANAG 5524 ed. E) or, more applicable for tactical domains, using WS-Discovery [6].

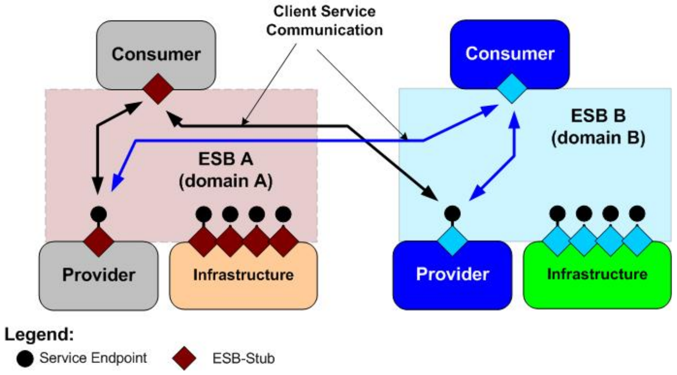


Figure 1: Service use across technical domains

The above service registry is a standard UDDI v.3 with a schema extension for network conditions (add-on, optional, therefore compatible with standard UDDI as specified in the WS-I Basic Profile).

B. Overcoming Bandwidth Limitations

The handling of multicast information in an (ad-hoc) radio network is not simple. To avoid a larger group management behavior, within tactical radio networks Simple Multicast Forwarding (SMF) [11] is used.

As various radio networks may be interconnected, it is necessary to provide a routing strategy which allows the integration of various link capabilities within a heterogeneous network overlay. Here the MTR approach [10] is used. It allows the end-to-end definition of a sequence of paths for a specific capability (e.g. for a minimum bandwidth between end nodes). The propagation of local link capabilities is here generated and provided by PPPoE [9]. If any radio systems do not support PPPoE (which is currently the case with most of the military radios in NATO), a PPPoE proxy or simple manual configuration may be used.

Some core services were modified to be more bandwidth efficient. The WS-Notification [12] service for example, in which a broker distributes topic-sorted notifications published by one party to multiple subscribed consumers, now uses multicast to send out notifications if more than one consumer is subscribed to a topic. The broker will also no longer expect acknowledgements, nor will consumers send any. Apart from saving bandwidth, this also means that radios do not have to switch between sending and receiving mode as frequently.

To learn about published or available services from partners, WS-Discovery [6] is used. Based on the local policy, each domain will announce the services which are available for partners. WS-Discovery also uses multicast. When used in pro-active mode, each active WS-Discovery provider distributes available services periodically on a specific multicast address. This can induce a very heavy load indeed for radio networks, depending on the number of available services to be distributed and the period for re-transmission. Practical experience in the CoNSIS experiment at WTD 81 in Greding in June 2012 has shown that the period, at least for land based systems, should not be in a range of a few seconds, but between 30 seconds and 1 minute [13].

Last but not least, XML, on which web services are based, is very verbose. In bandwidth constrained environments, exchanging larger XML messages will take a long time. This is a fundamental problem when using radios, independent of their type.

The obvious solution is to use compression. As mentioned above in section II.B, the most efficient ones are those which are able to interpret the specific coding schema of the source itself. Based on this schema information, the original source can be compressed in a range of about 97% which makes an XML input more transferable in narrowband radio networks. In the case of XML structures, the underlying schema information (which is by definition identical between source and sink) can be retrieved both by the sender and the receiver from one common or various distributed repositories within the network. The only requirement in a coalition environment is that the same schema files are used on all communication sites. This is an aspect of mission pre-planning.

The remaining question, then, is which compression algorithm to use. In CoNSIS, RuDi implemented both GZIP and XENIA [5], as the most efficient algorithm. The Norwegian framework supported GZIP and EXI [4], the standard-based choice. The compromise then was to use GZIP in cross-nation communication, but of course this solution is not desirable. A recommendation of which algorithm to use is to be worked out.

Please note that if using WS-Security [14] in combination with compression, the XML body of the relevant SOAP message must be compressed prior to their encryption. Otherwise compression is no longer possible.

The last aspect in the area of radio networks is the change in end-to-end transport services: As the transmission time and link availability may vary tremendously within interconnected radio networks, CoNSIS has mainly given up the concept of using TCP as the principal transport protocol. Instead, CoNSIS is using SOAP directly over UDP. The original specification [15] allows only the transfer of one UDP packet, which means that the SOAP message must fit completely into one UDP frame. This is not the case in every military network, with the consequence that the UDP specification has to be enhanced to allow segmentation and re-assembly as well as the recovery of lost or erroneous UDP frames. RuDi achieved this by using a wireless session protocol [16]. This approach is called the reliable UDP protocol [17].

C. Security Considerations in a Tactical Environment

Within a single, national implementation of a SOA environment a hierarchical model with well-defined roles and access rules may work. In a coalition environment, the various information domains may overlap various technical domains (which is used as a synonym for national domains). To operate within such an environment, the following assumptions are made:

Usage of SOAP

SOAP (Simple Object Access Protocol) is the basic element of web services. SOAP is a W3C protocol standard. It enables standardized communication between distributed applications and objects, particularly in the SOA/ESB environment.

Web services use SOAP for information exchange purposes and HTTP as a medium of transport. In their basic form, both SOAP as communication protocol and HTTP as transport protocol do not support any security requirements. Instead, data is transmitted in plain text. HTTP is, therefore, usually employed via SSL 3.0 and/or TSL 1.0 (HTTPS) to ensure the secure exchange of SOAP messages.

In RuDi an additional "object protection" transmitted together with the original SOAP message is being implemented for information objects transmitted via SOAP.

The OASIS standard WS-Security has established itself as the primary technology in this context. WS-Security defines how to use existing standards such as XML Encryption, XML Signature and X.509 certificates together. WS-Security is an essential enhancement of the SOAP standard. It is applied to fulfill the requirements with regard to message integrity, confidentiality, authenticity, and the authenticity and authority of the entities involved. It makes use of authentication and authorization based on SAML (Security Assertion Markup Language).

The security architecture is probably the most problematic area for interoperability, as not only the used protocols are of interest, but the way how the security is handled on the sending and receiving side may differ. First results in CWIX 2012 have shown that this area needs a deeper agreement between partners.

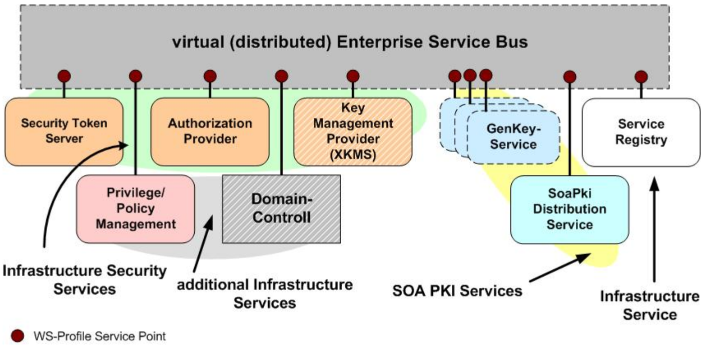
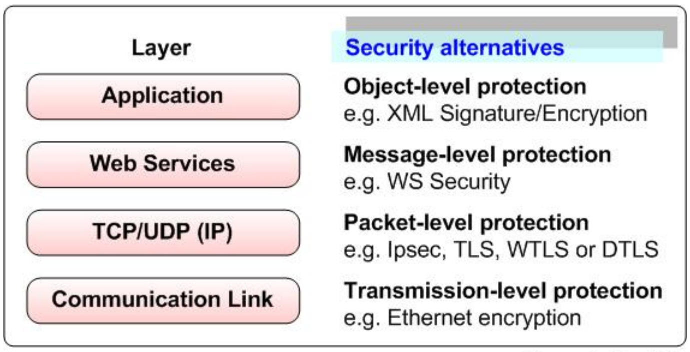

virtual (distributed) Enterprise Service Bus
Security Token Server
Authorization Provider
Key Management Provider (XKMS)
GenKey-Service
Service Registry
Privilege/ Policy Management
Domain-Controll
SoaPki Distribution Service
Infrastructure Security Services
additional Infrastructure Services
SOA PKI Services
Infrastructure Service
WS-Profile Service Point

Figure 2: IT security services of the SOA (ESB) infrastructure

The basic IT Security Architecture is based upon the RuDi Outline Concept Chapter 6.5.2 and Chapter 6.9.1 [18].

In the BI-SC AIS/NNEC SOA Implementation Guidance [19] NATO document, the layers depicted in Figure 3 are classified as IT security technology that RuDi is also based upon.



Source: [nc3-guide]

Figure 3: Example of a layer-based security architecture

The IT security architecture of the Reference Environment for Services is based on two components:

- **Elementary / basic protection** (basic security of the classified Bundeswehr LAN on network level)

  Elementary protection: the Virtual Private Network (VPN) technology protects the transmitted information from undesired access by setting up a "tunnel" protected by powerful cryptographic mechanisms that is going through insecure networks.

  Basic protection: The encryption by means of HAIPE, NINE or SINA with its IPSec internet protocol is based on the network level.

- **Object protection** (security on information level)

  Object protection is used to secure objects – information objects – by way of authentication, encryption, integrity securing and labeling (signature).

Generally, the SOA framework is an addition to the existing elementary / basic protection, which it uses. The elementary / basic protection is not described and analyzed in any more detail in this document.

RuDi implements object protection on the information level (application and web services) in the IT security architecture.

Certificate / Key Structure in RuDi

Put in technical terms, a X.509 certificate is a data construct that includes a user's public key, their personal data and a digital signature of the relevant certificate authority (CA). RuDi security is based upon certificates in accordance with the X.509 v3 standard. Using a root certificate, every user is able to verify whether the information signed by means of these certificates and the following certificate chain is authentic.

An OpenDS is used as directory for certificates. It is based on the ACP 133 [20] format (ed. C or D) in order to be smoothly interoperable in the NATO context.

Figure 4 shows an example of the CA structure (grayed out boxes) and an extract of the respective (Sub) CA certificate store.

In the example Mission RootCA Straussberg forms the highest-level Root CA (layer 1). The SubCAs for operations (missions, layer 2) that may be located on a stationary IT system in the home country are derived from this Root CA.

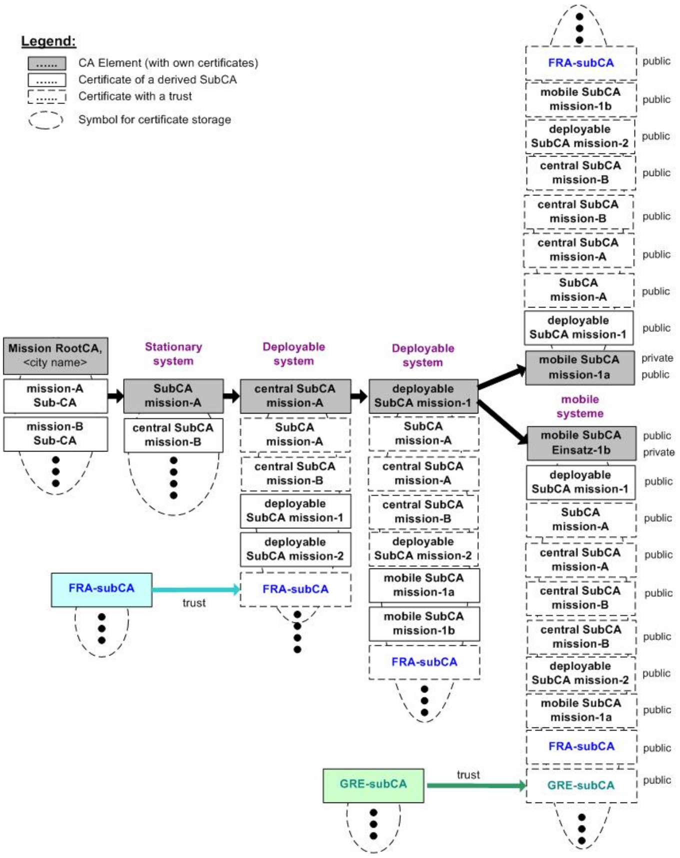


Figure 4: CA structure and extract of certificate store

In the example the SubCA mission-A is derived from the central SubCA-mission-A and, therefore, inherits its trust relationships. This means that the central SubCA-mission-A has a trust relationship with the central SubCA-mission-B in the example above.

Further deployable and autonomous IT systems may exist in the theater of operations. These systems derived from the central SubCA-mission contain a separate SubCA. These SubCAs form the deployable SubCA mission of layer 4.

Depending on the requirements of the theater of operations, mobile and autonomous IT systems (SubSubCAs) are distributed from the deployable SubCA mission and, if applicable, also from the central SubCA-mission. These IT systems (may) contain separate limited mobile SubCAs (layer 5).

The establishment of trust relationships may be required independent of the CA layer. The example in Figure 4 illustrates the establishment of a trust relationship to a Greek SubCA in the mobile SubCA operation-1b.

CA* and/or SubCA* are introduced due to the fact that technical domains are autonomous but not every technical domain of layer 5 automatically needs to have its own CA* or SubCA*. Like a CA or SubCA, the CA* and/or SubCA* receives a share of information from its respective higher-level CA. This CA* or SubCA* does, however, not form a CA or SubCA in organizational terms, but it remains assigned to its issuing CA or SubCA.

Synchronizations used to compare changes in the trust relationship and in the revocation list are required due to the CA structure and the direct trust relationship. Figure 5 illustrates the synchronization relationships resulting from the CA structure and the direct trust relationships in the example in Figure 4.

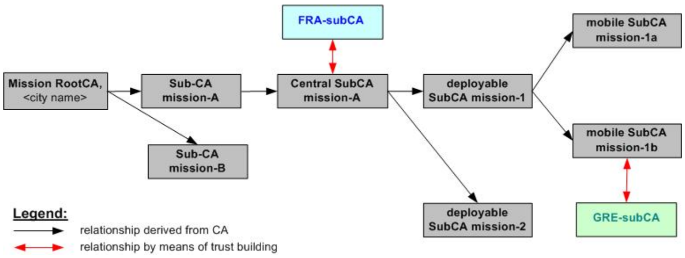

FRA-subCA
mobile SubCA mission-1a
Mission RootCA, <city name>
Sub-CA mission-A
Central SubCA mission-A
deployable SubCA mission-1
Sub-CA mission-B
mobile SubCA mission-1b
Legend:
relationship derived from CA
relationship by means of trust building
deployable SubCA mission-2
GRE-subCA

Figure 5: CA structure & trust synchronization relationships

Details of the procedures to create and maintain the various certificates in RuDi are described in [18], chapter 3.

SOA runtime security

The SOA Runtime Security is concerned with all steps and capabilities of IT security during service use and comprises:

- Authentication (checking the identity – identification);
- Authorization (check of authorization – access authorization);
- Encryption (digital encryption);
- Signature (electronic signature/ identification).

The definition of "addressing and identification" forms the basis of the SOA Runtime Security.

For all considerations concerning IT security it has to be noted that the IT security mechanism is regarded across technical domains (see Figure 6). It must be ensured that the mechanism is the same, both domain-internally and externally, and that simplifications working only domain-internally are not being established too quickly.

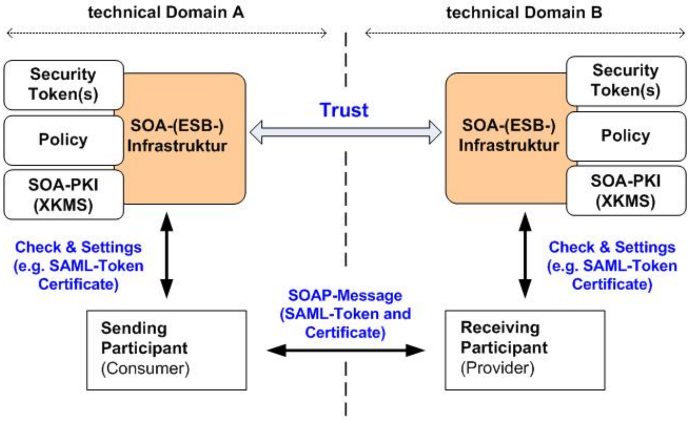

technical Domain A
technical Domain B
Security Token(s)
Policy
SOA-PKI (XKMS)
SOA-(ESB-) Infrastruktur
Trust
SOA-(ESB-) Infrastruktur
Security Token(s)
Policy
SOA-PKI (XKMS)
Check & Settings (e.g. SAML-Token Certificate)
Check & Settings (e.g. SAML-Token Certificate)
SOAP-Message (SAML-Token and Certificate)
Sending Participant (Consumer)
Receiving Participant (Provider)

Figure 6: Federation and trust configuration

This means that the transmitter and receiver instance may be located in different technical domains, each with its own SOA (ESB) infrastructure.

IV. Conclusion

While web services have come a long way in ensuring interoperability, the profiles written for use in civil systems are insufficient for use in tactical military networks. We have in this article introduced the areas which have been identified in CoNSIS as the areas in which further specifications are necessary. Solutions for these areas have been developed in the German national project Reference Environment for Services RuDi and are being verified in various field tests including CoNSIS. Following the full analysis of the final CoNSIS field experimentation of June 2012, if the proposed elements are proven mature they may be included in a profile for SOA in tactical networks to be recommended to NATO for standardization, to be included e.g. in STANAG 5524 (NISP) [21].

References

[1] T. Buckman, “NATO network enabled capability feasibility study”, v2.0, NC3A, October 2005

[2] Web Services Interoperability Organisation, WS-I Basic Profile version 1.1, ISO/IEC 29361:2008

[3] A. Diefenbach, M. Gerharz, S. Hunke, T. Lüke, and J. Tölle, Abschlussbericht SOA-Keimzelle Analyse Referenzarchitektur (SKAR), January 2010

[4] W3C, Efficient XML Interchange (EXI) Format 1.0, http://www.w3.org/TR/2011/REC-exi-20110310/, March 2011

[5] Christian Werner, “Xenia: Die smarte XML-Kompression aus Lübeck”, FOCUS MUL, 24. Jahrgang Heft 3, September 2007.

[6] OASIS, Web Services Dynamic Discovery (WS-Discovery) version 1.1, http://docs.oasis-open.org/ws-dd/discovery/1.1/os/wsdd-discovery-1.1-spec-os.docx, July 2009

[7] T. Hafsøe Bloebaum and K. Lund, ”CoNSIS: Demonstration of SOA interoperability in heterogeneous tactical networks”, MCC 2012, Gdansk, Poland, in press

[8] Web Services Interoperability Organisation, WS-I Basic Security Profile 1.1, January 2010

[9] B. Berry, “PPP over Ethernet (PPPoE) extensions for credit flow and link metrics”, IETF RFC 5578, February 2010

[10] P. Psenak, “Multi-topology (MT) routing in OSPF”, IETF RFC 4915, June 2007

[11] J. Macker, “Simplified multicast forwarding”, IETF draft-ietf-manet-smf-14, March 2012.

[12] OASIS, Web Services Notification, consisting of WS-BaseNotification 1.3, WS-BrokeredNotification 1.3 and WS-Topics 1.3, October 2006.

[13] CoNSIS-SC, CoNSIS final report, October 2012, unpublished

[14] OASIS, Web Services Security 1.1, February 2006

[15] xmlsoap.org, SOAP over UDP, September 2004

[16] Wireless Application Protocol Forum, Wireless Session Protocol specification, WAP-230-WSP, July 2001

[17] CoNSIS-SC, “Reliable UDP”, CoNSIS-DEU-Task1-DU-002.doc, November 2010

[18] M. Franke, H. Seifert, “IT security architecture”, IT-AmtBw, Project RuDi, E/IB1S/9A031/6F125, January 2012

[19] NATO, “BI-SC AIS/NNEC SOA implementation guidance”, final draft 1.0, December 2009

[20] CCEB, Common Directory Services and Procedures, ACP 133(D), July 2009

[21] NATO, NATO Interoperability Standards and Profiles, ADatP-34(F), December 2011