International Journal of Digital Earth,
2012, 1–18, iFirst article

Taylor & Francis
Taylor & Francis Group

Pervasive geo-security – a lightweight triple-A approach to securing distributed geo-service infrastructures

Bernd Resch[a,b]*, Bernhard Schulz[c], Manfred Mittlboeck[a] and Thomas Heistracher[c]

[a]*Research Studio iSPACE, Research Studios Austria, Salzburg, Austria;* [b]*SENSEable City Laboratory, Massachusetts Institute of Technology, Boston, MA, USA;* [c]*Department of Information Technology and Systems Management, University of Applied Sciences, Salzburg, Austria*

(*Received 15 September 2011; final version received 7 March 2012*)

Security has recently become a major concern in distributed geo-infrastructures for spatial data provision. Thus, a lightweight approach for securing distributed low-power environments such as geo-sensor networks is needed. The first part of this article presents a survey of current security mechanisms for authentication and authorisation. Based on this survey, a lightweight and scalable token-based security infrastructure was developed, which is tailored for use in distributed geo-web service infrastructures. The developed security framework comprises dedicated components for authentication, rule-based authorisation and optimised storage and administration of access rules. For validation purposes, a prototypical implementation of the approach has been created.

Keywords: pervasive security; distributed geo-service infrastructures; triple-A; geo-authorisation; standardised sensor networks; service protection; Geographic Information Systems (GIS); geoinformatics; spatial data infrastructure; Digital Earth; Digital Earth Architecture

1. Introduction

Until recently, provision of geospatial data happened exclusively via exchanging hardcopies such as CDs or DVDs. However, in the emerging vision of 'Digital Earth' (Craglia *et al.* 2008) geo-data are increasingly provided via service-based interfaces such as Open Geospatial Consortium (OGC) Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS), Sensor Observation Service (SOS), Web Processing Service (WPS) or Sensor Alert Service (SAS).

A central requirement originating from this recent spread of service-based data provision is security in geospatial data services. This need is also rooted in the INSPIRE directive (European Commission 2010), which states in article 4, paragraph 2 that 'where spatial data sets and services are made available [...] community institutions and bodies shall make every possible effort to avoid unauthorised use of spatial data sets and services'. In other words, geospatial data have to be provided following criteria such as trustworthiness, completeness and up-to-dateness.

However, existing commercial solutions are mostly expensive and very complex to implement. This lack of appropriate security mechanisms is one central reason why (public) institutions are oftentimes reluctant to open their data repositories. Recent research and position papers on Digital Earth (Craglia *et al.* 2008, 2012) often emphasise the need for sensor systems, for socio-technical awareness-raising, for accurate simulation models or for multi-disciplinary methodological research, but often neglect insufficient security solutions as a limiting factor towards the realisation of the vision of Digital Earth.

In fact, lightweight security becomes a major concern in the context of service-oriented architectures (SOA) because data provision interfaces are exposed via the Internet raising the need for comprehensive but simple security mechanisms. In this regard, the paradigm of 'separation of concerns' is a key criterion. It means that single points of failure should be avoided in distributed infrastructures in that each component performs a dedicated task instead of establishing an all-in-one solution. In the context of geo-infrastructures this means that separate services should be created for data provision (OGC services as mentioned above) and for security to keep the system maintainable and configurable.

The basic goal of this research was to create a lightweight approach for securing distributed low-power environments. The security infrastructure comprises methods for authentication (confirming the user's identity), geo-authorisation (defining and enforcing spatial access rights) and optimised storage and administration of access rules. It shall be mentioned our research does not explicitly define strict performance or security level requirements, but tries to leverage existing technologies and methods in particular usage scenarios for low-power computers and distributed service infrastructures for spatial data provision.

1.1. Requirements for a lightweight approach to securing geo-web service infrastructures

Service-based geo-data provision in many cases involves the OGC Open Web Service (OWS) request-response model (Figure 1), which traditionally implements communication between client and server via HTTP GET and POST, or via SOAP. Using OWS, geospatial data can be provided via a variety of services as mentioned earlier.

From a data provider's perspective, the security system has to be able to authenticate users and to restrict access to data-sets by filtering according to bounding polygons, data layers, accuracy of the provided data, temporal intervals and complexity and exclusivity of provided algorithms.

Particular security challenges of distributed geo-infrastructures comprise communication over the public Internet, the need for mutual authentication and the

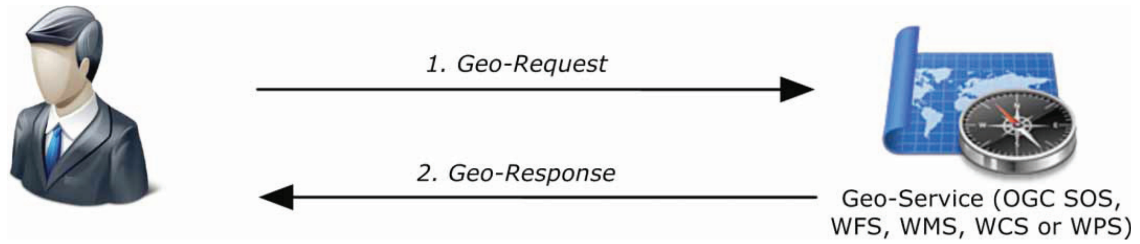

Figure 1. Unsecured geo-service infrastructure.

requirement of resource-saving security mechanisms to guarantee near real-time performance using low-power embedded devices.

This article presents a survey of existing security mechanisms for authentication and authorisation. Based on this survey, lightweight and scalable security infrastructure was developed, which is tailored for use in distributed service environments. The developed security framework comprises dedicated components for authentication, rule-based authorisation and optimised administration and storage of access rules.

Summarising, the key requirements for the security infrastructure are:

- Lightweight architecture for low-power environments,
- Support for distributed geo-service infrastructures,
- Cost-efficient implementation,
- Compatibility with OGC-compliant geospatial filtering,
- Building upon cost-free and license-free technologies.

The article is organised as follows: after this introduction, a short section on related work is presented to illustrate different security framework approaches, specifically for the geospatial domain. Thereafter, an evaluation of authorisation techniques, mechanisms to store and administrate geo-authorisation, and authentication possibilities are presented with a particular focus on distributed geo-infrastructures. Following the evaluation results, Section 4 describes a new approach towards a lightweight security infrastructure for distributed low-power environments including a prototypical implementation and a discussion of benefits and expected impact. The final section of the article contains a short conclusion.

2. Related work

Within the **GENESIS** project (GENESIS Consortium 2011), a generic security infrastructure was developed for large-scale Geographic Information Systems (GIS). The project's frame required the design of a security framework for SOA. Thus, the Security Assertion Markup Language (SAML) (OASIS 2011a) is used for authentication, Web Service Security (WSS) (OASIS 2011b) is used to secure services and the eXtensible Access Control Markup Language (XACML) (OASIS 2011c) is used to define and enforce security policies.

As the earlier mentioned technologies rely on SOAP-based communication, they are only suitable for powerful geo-information infrastructures, but not for distributed service infrastructures such as pervasive sensor networks.

The user management, authorisation and authentication policy of the **ORCHESTRA** architecture (ORCHESTRA Consortium 2009) is divided into the following components:

- User management,
- Authentication,
- Authorisation.

The authorisation service gives a compliance value as response to a service requesting an authorisation decision for a given authorisation context. The authorisation service provides its functionality through the following interfaces: ServiceCapabilities, AuthorisationService and XAuthorisationAdministration. The authentication service verifies genuineness of principals using a set of given credentials.

The authentication mechanism is up to the service implementation. Which credentials an authentication service needs, as well as the way they are passed is specific to the authentication mechanism used. The authentication service provides its functionality through the following interfaces: ServiceCapabilities, AuthenticationService and UsernamePassword-Mechanism.

As the ORCHESTRA authentication and authorisation services have been designed according to a variety of established ICT security standards for high-performance environments [Kerberos, X.509, LDAP, OGC GeoDRM (Open Geospatial Consortium 2011), KeyNote Trust Management System, amongst others], their suitability for distributed geo-infrastructures involving low-power embedded sensors is rather limited.

The same applies to the Secure European System for Applications in a Multi-vendor Environment (**SESAME**) (Ashley and Vandenwauver 1998). SESAME is a security architecture, which builds on the Kerberos authentication system. In addition, it uses public key based security for authentication and role-based access control for authorisation purposes. Generally speaking, the SESAME architecture is very complex, which prevents its deployment in distributed environments using low-power computers.

The **52North Security and GeoRM** software suite (52°North Initiative 2011) is basically targeted to securing distributed geo-services. As it uses WSS in a Java-based implementation to transmit secured messages, it is not suitable for low-power environments.

The open-source project **GeoServer** (OpenGeo 2011) defines a separate security component, which is directly integrated into the service implementation. This component enables layer-level security (protecting single geo-layers) and service-layer security (protecting the geo-service as a whole). However, these two service types cannot be combined. This restriction together with the fact that the requirement of 'separation of concerns' is not fulfilled make GeoServer built-in security not a viable choice for distributed environments. Furthermore, GeoServer authentication is based on basic HTTP authentication, which is not suitable for usage in comprehensive distributed geo-service infrastructures due to extensive administration requirements and a lacking trust relationship.

Simple Distributed Security Infrastructure (SDSI) (Rivest and Lampson 2001) combines a simple public-key infrastructure design with a means of defining groups and issuing group-membership certificates. Although SDSI aims to reduce infrastructure complexity, it is not suitable for the requirements defined above because the concept of flexible signatures is expensive to maintain, dedicated geo-filtering is not supported and interoperability on service level is limited due to proprietary protocol extensions.

Table 1 summarises existing approaches towards securing distributed geo-service infrastructures including their suitability for the requirements described in sub-section 1.1.

Table 1. Existing approaches to securing geo-service infrastructures.

	Suitability	Reference
GENESIS	SOAP-based communication not suitable for low-power environments	http://www.genesis-fp7.eu
ORCHESTRA	Security technologies, which make up the infrastructure, not suitable for low-power	http://www.eu-orchestra.org
SESAME	Complex security architecture	http://www.esat.kuleuven.ac.be/cosic/sesame.
52N Security	Java-based WSS implementation not suitable for low-power environments	http://www.52north.org/security
GeoServer	Limited security functionality and single point of failure	http://docs.geoserver.org/
SDSI	No support for geo-filtering; limited interoperability	http://groups.csail.mit.edu/

3. A state-of-the-art analysis of authentication and geospatial authorisation mechanisms

Within the presented research, a particular focus was on geo-authorisation technologies. However, the use of authorisation is by definition only useful in combination with according authentication mechanisms. Thus, authentication and geo-authorisation technologies are evaluated along with each other. Then, geo-authorisation mechanisms are assessed separately, followed by an overarching summary.

3.1. Evaluation of existing authentication and authorisation mechanisms

Before treating geo-authorisation mechanisms, available possibilities for implementing authentication and authorisation mechanisms (OAuth 1.a, HTTP authentication, SAML 2.0, WS-Security, XACML and Shibboleth 2.3) have to be assessed.

As authentication mechanisms themselves are already widely implemented in data providers' ICT infrastructures, no separate evaluation of those mechanisms has been performed within this research, but the evaluation has been performed assessing authentication and authorisation mechanisms together at once. This is due to the fact that most existing authentication mechanisms can be rather easily coupled with a variety of authorisation mechanisms.

3.1.1. Evaluation methodology

The evaluation has been performed according to eight parameters where the evaluation scale has been divided into three classes: good (+), moderate (O) and poor (−). Table 2 shows the evaluation results accounting for these criteria:

- *Diffusion*: spread and support within current security implementations
- *Communication Overhead*: message size, number of communication partners involved, number of overall steps
- *Available Software Libraries*: availability of extensively tested, well-supported and continuously maintained program libraries

Table 2. Evaluation summary of existing authentication and authorisation mechanisms.

	OAuth 1.a	HTTP authentication	SAML 2.0	WS-security	XACML	Shibboleth 2.3
Diffusion	O	+	O	O	+	O
Communication overhead	+	+	O	–	–	–
Available software Libraries	+	+	+	O	O	O
Implementation effort	O	+	O	–	–	–
Encryption	Transport Layer (HTTPS)	Transport layer (HTTPS)	Public–private keys	Public–private keys	Public–private keys	Public–private Keys
Trust relationship	Shared secret	None	Certificate	Certificate	Certificate	Certificate
Bindings	HTTP	HTTP	SOAP	SOAP	SOAP	HTTP
Fitness for complex use cases	O	–	O	+	+	O

- *Implementation Effort*: effort for establishing the security implementation depending on the communication complexity, message exchange and supported programming languages
- *Encryption*: concept/technology used for securing the communication and the services
- *Trust Relationship*: required information, which the communication partners have to know from each other a priori
- *Bindings*: underlying communication protocols
- *Fitness for Complex Use Cases*: suitability for the complex use cases using geo-infrastructures (cascading services, security in service-oriented infrastructures, heterogeneous IT architectures and data formats).

3.1.2. Discussion

Looking at the evaluation summary in Table 2, **OAuth** (OAuth Community 2011) – a standard allowing a user to grant a third party access to their information stored with another service provider – seems to be a good option for the developed security framework due to its broad support by numerous big players in the area of web 2.0 (Twitter, Google, Facebook, etc.), the availability of a variety of implementation libraries, and its efficient communication structure. Generally speaking, OAuth is intended to grant third-party access to data services. However, it is characterised by limited native support of geospatial access rules and thus has to be coupled with special mechanisms to integrate geo-access rules.

It is also evident that **basic HTTP authentication** (W3C Networking Group 1999) – simple provision of user name and password credentials – seems to be a viable security method due to its broad support in various web browsers, the availability of numerous libraries and its low implementation effort. This authentication technology cannot be used for the complex uses cases (Figure 4) due to its simplicity and lacking support for SOA. However, **OpenID** (OpenID Foundation 2011) seems to be a very promising technology to implement ubiquitous authentication coupled with authorisation as it broadly supports access to a variety of portals.

Security Assertion Markup Language (**SAML**) (OASIS 2011a), which is an XML-based OASIS standard for exchanging authentication and authorisation data, is a practical option to handle complex use cases. In effect, the communication overhead (through the use of SOAP) and the exchange of certificates require a substantive amount of resources. On the contrary, the number of exchanged messages is relatively small.

Like SAML, **WS-Security** (OASIS 2011b) uses SOAP as its underlying message exchange protocol. Due to its complexity, it is well suited for securing distributed infrastructures. Through its message-based security concept, WS-Security is very well designed for the use in SOA.

The same applies to **XACML** (OASIS 2011c), which is an OASIS-standardised XML-based access control policy language. The functional separation in the overall XACML infrastructure is well defined, which allows for securing complex geo-data infrastructures. Again, this naturally increases the implementation effort through the mandatory use of Policy Decision Points (PDP) and Policy Enforcement Points (PEP).

Finally, Shibboleth (Internet2 Middleware Initiative 2011), which internally uses SAML and WS-Security, is not very common and is characterised by an enormous implementation effort. Furthermore, there are only few program libraries, preventing the use of Shibboleth in the developed security infrastructure.

3.2. Evaluation of existing geo-authorisation mechanisms

In a next step, possibilities for restricting access to geo-data services based on geospatial parameters have been assessed.

3.2.1. Evaluation methodology

The evaluation has been performed according to nine parameters. The evaluation scale has been divided into three classes: good (+), moderate (0) and poor (−). The evaluation results accounting for the criteria listed below are shown in Table 3.

- *Standardisation*: standardisation of the geo-rule format.
- *Diffusion*: future market perspectives, spread and support within current security implementations.
- *Direct Support for Geo-Services*: spectrum of (OGC) geo-services, which can be protected using in-request filter transport.
- *Filtering Capabilities*: parameters, according to which data can be filtered (bounding box, map layer, time spans, etc.).
- *Billing Support*: support for an underlying billing infrastructure (integration of different accounting mechanisms such as per-query, per-layer, per-month or flat-rate).

Table 3. Evaluation summary of mechanisms for storing geo-authorisation rules.

	GeoXACML	CQL	OGC filter encoding	Proprietary XML dialect	INI configurable file
Standardisation	+	O	+	–	–
Diffusion	O	O	+	–	–
Direct support for geo-services	O	O	+	–	–
Filtering capabilities	+	O	+	n/a	n/a
Billing support	O	–	–	n/a	n/a
OGC filter conversion	O	+	+	O	O
Communication overhead	O	+	O	n/a	n/a
Complexity of the infrastructure	O	O	n/a	n/a	n/a
Fitness for complex use cases	+	O	O	–	–

- *OGC Filter Conversion*: effort to convert the rules to OGC Filter (Vretanos 2005) representation as used in OGC geo-web-services.
- *Communication Overhead*: message size, payload overhead.
- *Complexity of the Infrastructure*: complexity of the security architecture.
- *Fitness for Complex Use Cases*: suitability for the complex use cases using geo-infrastructures (cascading services, security in service-oriented infrastructures, heterogeneous IT architectures and data formats).

Before discussing the outcomes of the evaluation of available geo-authorisation mechanisms, GeoXACML is separately described and assessed as it is currently a very promising approach towards securing spatial data infrastructures (SDI).

3.2.2. Access control for geographic information – GeoXACML

Generally speaking, GeoXACML (Matheus and Herrmann 2011) extends the OASIS XACML standard. It establishes a policy language that uses XML encoding to express complex access rights, such as spatial access rights. This standardised policy language allows for interoperable processing, exchange and collaborative creation of policies independent from the underlying service based architecture. Besides the policy language, GeoXACML and XACML respectively describe a general architecture and information flow model. This architecture, which is shown in Figure 2 (blue: OGC components, orange: additional components for access control) enables a clean separation of the access control system from the web service it is protecting.

The GeoXACML concept can be used to establish access control for protecting OGC web services to regulate the access to geospatial data. An essential benefit of GeoXACML is that it can be incorporated into an existing service infrastructure by extension, without modification of the currently existing software components implementing OGC specifications.

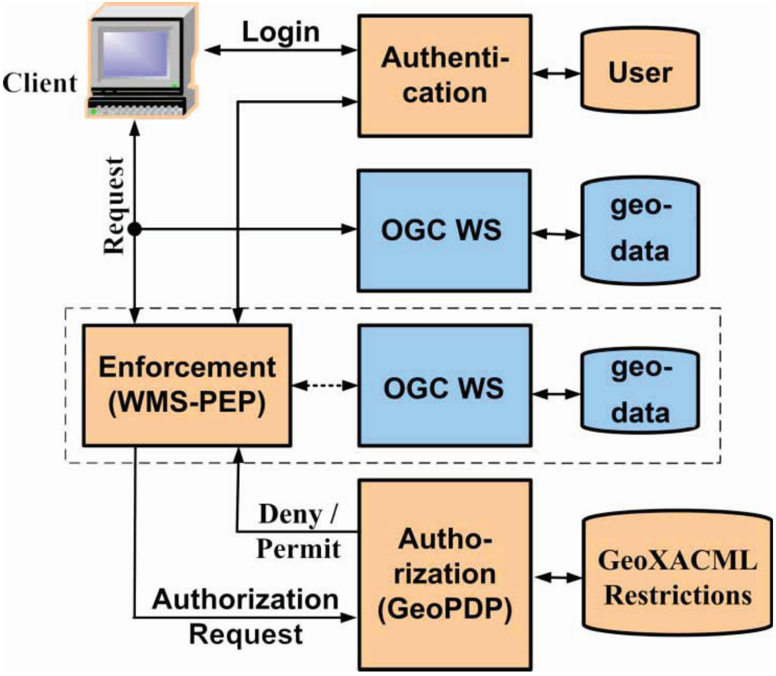

Figure 2. GeoXACML architecture (modified from AM Consult 2009).

GeoXACML introduces access control specific software components according to the XACML architecture. These components are the PEP that intercepts the communication between the OGC web services. Login and authentication of user identities are provided by the Authentication component. The central component of the access control extension is the Policy Decision Point (GeoPDP) that derives the authorisation decisions based on GeoXACML policies.

Even though GeoXACML offers pre-processing access control (rule enforcement before the data query), the OWS-6 GeoXACML Engineering Report (Hermann and Matheus 2009) brought up following issues concerning the use of GeoXACML for securing OGC web services:

- Sources of unnecessary freedom can result in losing interoperability in the access control system (in spite of the usage of a standardised rule language such as XACML or GeoXACML).
- XACML attributes should not be used at all to represent information on OWS requests or responses. Instead PEPs and policy writers should always use the *ResourceContent/AttributeSelector* approach to represent and reference OWS specific information.
- On the one hand, the *resource-parent*, *resource-ancestor* and *resource-ancestor-or-self* XACML attributes are optional to implement. On the other hand, the *resource-parent*, *resource-ancestor* and *resource-ancestor-or-self* attributes shall be present in an XACML-based infrastructure.
- The *AttributeSelector* mechanism as currently described in the specification is only intended to select text nodes. In some cases, this might not be sufficient.
- The authorisation semantics for an OWS are usually independent of the used protocol encoding. The same access rights need to be fulfilled no matter if the user sends a key-value-pair (KVP) encoded request or a corresponding XML-encoded request.

In addition, several other shortcomings of GeoXACML and related access control and filtering mechanisms for OGC geo-web-services have been identified through the evaluation in the course of the presented research:

- GeoXACML partly uses the OGC Filter specification. However, the harmonisation of the OGC Filter compliant structures and the GeoXACML rule representation still needs improvement. One essential problem is that OGC web services do not support property filtering.
- Furthermore, OGC web service requests cannot always be transformed into GeoXACML rules without loss of information when using XACML attributes. Thus, a combined *ResourceContent/AttributeSelector* approach (using a *ResourceContent* element as input for GeoXACML rules in combination with the common *AttributeSelector* element) seems to be the most suitable way to overcome this shortcoming.
- The OGC Filter specification does not support unit of measure (UOM) filters. Through this reduced capability of expression, the available access rights are dependent on the capabilities of the OGC web service request. Again, this issue requires increased harmonisation efforts between OGC filtering, OWS service requests and GeoXACML development. Furthermore, a consistent and standardised UOM registry should be urgently developed and introduced in the geospatial community.
- Due to this lacking functionality, the OGC SAS, which is an essential element for event-based data provision in distributed geo-service infrastructures, defines its own filter structure. However, in order to create a consistent secure geo-service infrastructure, the OGC web services have to be fully compliant with the OGC Filter specification.

3.2.3. Discussion

GeoXACML (Hermann and Matheus 2009, Matheus and Hermann 2011), a standardised policy language that uses XML encoding to express complex access rights, is characterised by its ability to handle complex geo-data infrastructure security requirements. As GeoXACML is a specific extension to XACML, it also uses PDPs and PEPs resulting in a rather complex overall infrastructure. Thus, GeoXACML is not well suited for low-power environments.

Generally speaking, the OGC GeoXACML standard is not really suitable for use in low-power distributed environments. Also, the exact market perspectives of GeoXACML are currently not clear as its development is not pursued with great ambition. Furthermore, several viable alternatives exist to represent geographic rules, which are mostly used in context-specific implementations. However, GeoXACML seems to be a seminal approach towards a holistic description of geo-rules, and simple integration into OGC web service architectures and IT security infrastructures. A key step will be to harmonise GeoXACML with the OGC Filter specification as far as possible to reach maximum compatibility with existing geo-web-service deployments and developments.

Like GeoXACML, also CQL (The Library of Congress 2008) (Contextual Query Language, previously known as Common Query Language) is a standardised formal language for representing queries to information retrieval systems. However, CQL is not very widely used in real-time geo-data environments as it only supports WFS and WMS, but neither SOS nor SAS. Furthermore, CQL cannot be used to express complex queries for filtering data layers and conversion to OGC Filter rules requires fairly sophisticated transformation procedures.

Open Geospatial Consortium (OGC) Filter Encoding (Vretanos 2005), which provides filter structures according to geospatial operations and features (e.g. a geographical bounding box), is very widely used, particularly by OGC web services (WFS, WMS and WCS). OGC SOS and SAS partly define their own filters, which can rather easily be mapped to OGC Filter compliant structures. However, OGC filtering only supplies the pure filter structures and does not provide native integration into existing IT infrastructures. In effect, OGC Filter has to be combined with other IT security technologies, which increases the implementation effort.

Proprietary XML dialects and INI configuration files have a range of advantages (rather simple implementation, good compatibility with OGC Filter or optimised communication overhead), but the facts that they are not standardised and not applicable to complex use cases, make these methods unusable for the developed security suite.

4. Securing (Geo) web services – a lightweight triple-A approach

This section presents the developed approach towards a security infrastructure for low-power environments. Sub-section 4.1 presents the technology choice for the security architecture, sub-section 4.2 describes the methodology behind the conceptual approach and sub-section 4.3 presents the prototypical implementation.

4.1. Motivation of technology choice

According to the evaluation of authentication and combined geo-authorisation methods and technologies presented in Section 3, it seems evident that the following technologies are best suited for distributed low-power geo-infrastructures.

- OAuth for authorization
- OGC Filter Encoding for storing authorisation rules
- OpenID for authentication

An essential aspect is that those technologies allow for rather simple transformation of OGC Filter based rule encoding (as used by OGC WFS, WMS, WCS, SOS and partly SAS), for example, through eXtensible Stylesheet Language Transformation (XSLT). Like this, standardised OGC geo-web-service requests can be transformed into more specialised and comprehensive rule languages.

Even though SAML and XACML are characterised by an elevated implementation effort, they enable the deployment of complex security mechanisms in distributed SOA. This applies particularly to cascading geo-services as shown in Figure 4. However, they are not usable for low-power environments due to performance issues.

This procedure also fulfils the requirement of ‘separation of concerns’, for example, that geo-services provide geo-data without natively implementing security mechanisms allowing for a more flexible service chain and technology infrastructure.

For the particular use in distributed geo-infrastructures, OpenID (OpenID Foundation 2011) seems very well suited being a lightweight authentication protocol and offering the well-known concept of user name and password for authenticating users.

Specific security implementations such as the 52° North Security Suite (52°North Initiative 2011) (a comprehensive Java-based architecture for securing geo-services using XML-based policies), the ConTerra securityManager (con terra 2011) (a commercial product for securing geo-services using SAML and XACML) and GeoShield (Institute of Earth Science 2011) (a security suite for OGC web services, which is still in very early development) have been briefly investigated. They have not been used in the actual implementation presented in sub-section 4.3 due to their implementation in Java, their commercial licenses and their limited support and availability, respectively.

4.2. Methodology – a lightweight approach to securing distributed geo-infrastructures

As mentioned in the introductory section, the aim of this research was to create a lightweight approach for securing distributed low-power environments, comprising methods for authentication, authorisation and optimised storage and administration of access rules.

4.2.1. Introduction of a security layer

To account for the requirement of 'separation of concerns', the geo-data service (e.g. OGC WFS or SOS) shall not be responsible for handling security and account information. Thus, a separate security layer has to be introduced between the user (data requestor) and the data service (data provider).

Practically speaking, the security layer comprises two components: (1) a *security service*, which serves as a proxy to handle geo-requests and (2) an *authentication, authorisation and accounting service* (AAAS), which handles user account data, enforces access rights and deals with the accounting process. The overall security architecture including the basic data flow is depicted in Figure 3.

In effect, the user sends their geo-data request to the security service, which communicates with the AAAS to check and apply authentication and authorisation rules, and forwards the request to the geo-service. For the user, this process is totally transparent as the data request (e.g. WFS GetFeature or SOS GetObservation) is sent to a web service endpoint as usual.

The benefits of this procedure are the compliance with the 'separation of concerns' requirement and the unloading of the geo-data service, which can be an embedded device in the case of pervasive sensor networks using OGC SOS.

To prevent the potential disadvantage of every geo-service maintaining its own security database, a central repository is used. This repository contains parameters, which are necessary for authentication (user credentials), authorisation (access rights) and accounting (billing information).

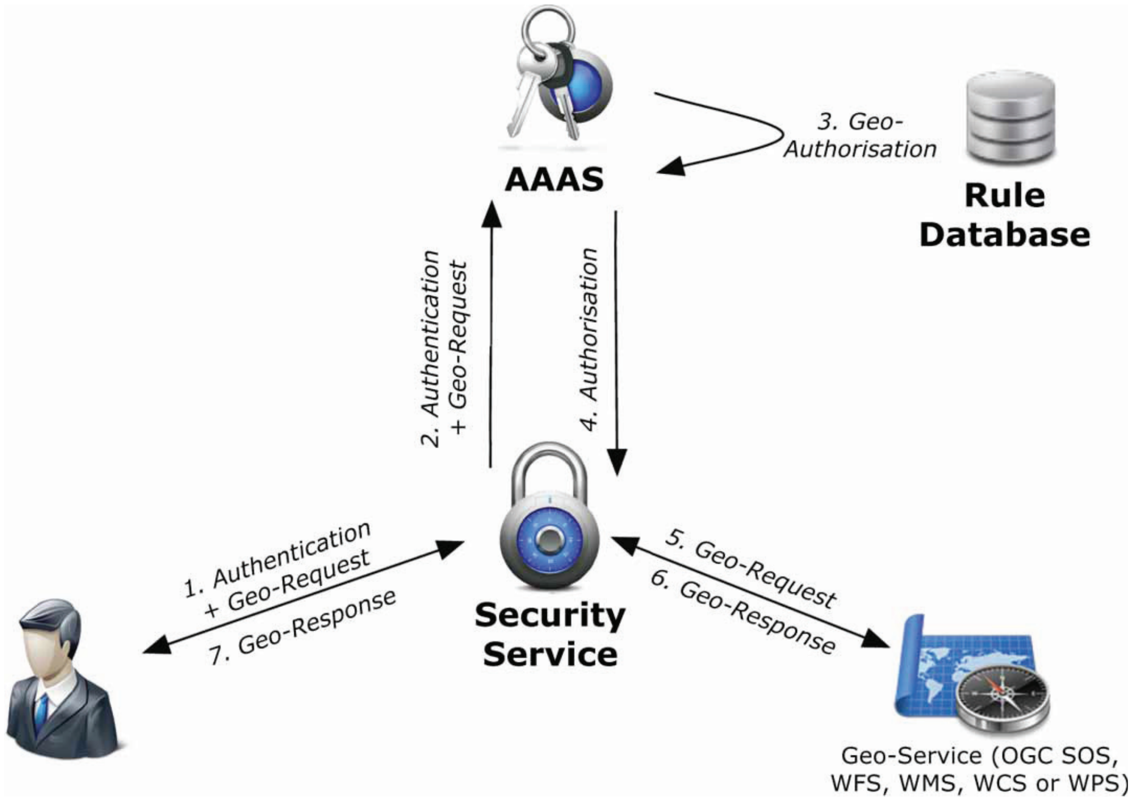

Figure 3. Secured geo-service infrastructure with security service and AAAS.

4.2.2. Using token-based security

Basically, security in distributed environments can be established using message-based encryption, point-to-point security and token-based security.

Using message-based security, every single message is encrypted before being sent to the recipient. This process can result in substantial performance losses, particularly in low-power systems.

Point-to-point security mechanisms establish a secure channel between sender and recipient before transmitting data. This method is, for instance, implemented in the well-known Transport Layer Security (TLS) protocol to allow for secure HTTP-based data transmission over the Internet.

However, considering distributed low-power service environments, token-based security seems to be the preferable method. Token-based security defines an architecture involving user, data services and authentication services. To request data from a data service, the user first has to obtain a security token from the authentication service. This token is then sent along with the request to the data service. The data service gets the token verified by the authentication service and sends the requested data back to the user.

The essential advantages of token-based security systems are that the client has to authenticate itself only to the authentication service (and not to the data service), and that a separate security entity is introduced into the system. Furthermore, it is impossible for eavesdroppers to draw conclusions from a user's token to connected access rights and credentials. Also, the validity of a token can be temporally limited, for example, for two months, or even only for a single request in high-security scenarios. Thus, the entire infrastructure becomes more secure through the introduction of a token-based system.

For the accounting part of the AAAS, no generic solution can be found as it highly depends on the concrete application context. The prototypical implementation described in sub-section 4.3 includes a simple logging mechanism to store the number of accesses.

Apart from the basic infrastructure shown in Figure 1, the developed security concept shall also be applicable to complex usage scenarios using cascading services with a special focus on embedded services (such as pervasive SOS). Figure 4 shows an example of such a complex scenario, in which an OGC WPS compliant service requests data from other services such as OGC SOS, WFS and WCS.

Handling this kind of use case requires 'cascading security' in that every service, which is queried, has to have access to a shared AAAS to verify user credentials and check access rights. Naturally, if the geo-data services are running in the same network, a single security service and AAAS can be used to secure all those services.

4.3. Prototypical security infrastructure implementation

To verify and validate the presented architecture, a prototypical implementation has been developed containing a security service to handle geo-requests and an AAAS to deal with authentication, geo-authorisation and accounting.

The security service has been implemented using *PHP* (Hypertext Pre-processor) as it offers plenty of ready-to-use libraries for handling HTTP-based communication. For handing token exchange and access to the service provider, we used the *Zend OAuth library* (Zend Technologies Ltd. 2011). To forward geo-data requests

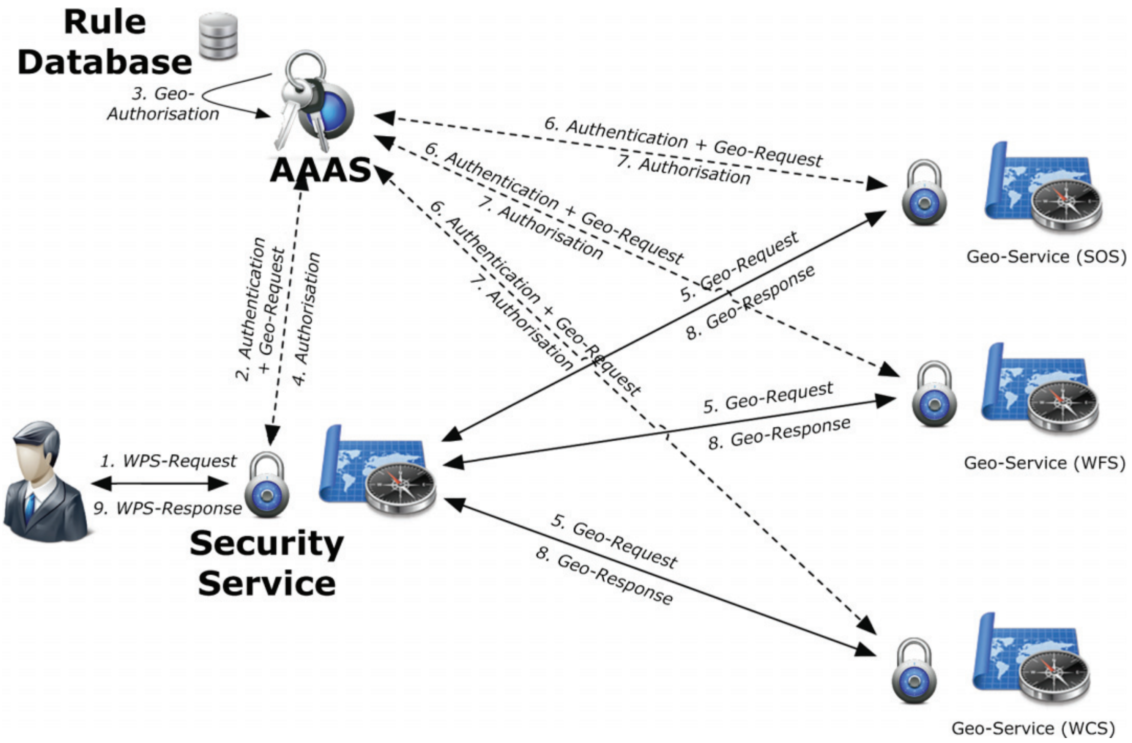

Figure 4. Secured geo-service infrastructure for cascading services.

(e.g. SOS GetObservation) to the according geo-service, we used the *Zend HTTP-Client library.*

To implement the AAAS, which is responsible for handling authentication, authorisation and accounting requests as well as for verifying tokens, we used Arjan Scherpenisse's *OAuth* library (Scherpenisse 2011), which has been extended by a user database, so that the AAAS can simultaneously handle various requests for access rights. In practice, the license broker receives the geo-data request and the token and looks for according filters in the database. Consequently, the geo-data request is modified according to the access rights in the database and returned to the security service.

To enable login functionality we used the *Zend OpenID* library for user name and password based authentication. As mentioned earlier, a rudimentary accounting mechanism has been integrated into the prototypical implementation. Every request for geo-data has been logged in the accounting database. The exact billing method and resulting access restrictions have to be solved specifically for each application.

4.4. Discussion and expected impact

The principal novelty in the presented solution is the lightweight triple-A (authentication, authorisation, accounting) solution for securing web services using openly available technologies and methods. Furthermore, our approach dedicatedly focuses on geo-data provision including geospatial filtering capabilities in low-power environments.

A central benefit of the proposed solution is its simplicity in terms of implementation and maintenance. This arises from the design decision to use a security service proxy, which acts as a coordination point between the client, the geo-web service and the AAAS. Like this, the solution can be applied to a number of (OGC) web services without having to modify the services themselves. Through the simplicity of the security service it can be implemented on embedded systems without any problem – unlike most existing security solutions, particularly SOAP-based ones.

Another essential advantage of the presented solution is that all used technologies are freely available. Thus, it enables cost-efficient security in geo-web service infrastructures without extensive license costs. This is a particular benefit over existing commercial solutions, which are mostly expensive and/or very complex to implement. This lack of appropriate security mechanisms is a main reason why (public) institutions are reluctant to open their data repositories. Thus, our solution can help overcome technological security barriers to open data access.

Moreover, the implementation using OpenID for authentication shows that it is easily possible to integrate external authentication services. In consequence, the developed solution does not constitute an isolated application, but it suitable for usage in large-scale implementations involving plenty of users worldwide.

The economic significance of the developed approach is twofold. First, the solution's flexibility and modularity provides security for geo-service infrastructures in a very simple way, involving low implementation effort and in a cost-efficient manner. The system enables user-tailored data provision and billing through the definition of dedicated access rights and pricing conditions. The solution allows for the implementation of different accounting methods such as per query, per feature, per geographical area, per data-set and so on depending on specific user requirements. Thus, comprehensive and complex framework directives for data usage are rendered unnecessary and billing could, for example, also happen via trusted online payment systems such as PayPal (http://www.paypal.com). The integrability of external authentication services enables data access as well to private users providing their credentials (user name and password) as to large organisations maintaining their own authentication servers.

The second economic benefit lies in the solution's simple architecture. The lightweight nature of the whole system allows its installation on laptops, smart phones or tablet computers. This can be an essential advantage in case of emergency – for example, when the existing data infrastructure has been damaged – because ad-hoc connectivity and secured data access can be easily provided.

Finally, it shall be mentioned that the degree of security is dependent on the used technologies themselves. This concerns the implementation of the security service proxy, the AAAS and the accessibility of the rule database. Using token-based security in connection with OpenID for authentication and OAuth for authorisation is considered a sufficient level of security.

The developed prototype has been validated and tested on a variety of geo-web services and hardware platforms and is basically ready for use in production environments. Before deployment, some code optimisation, checks for functional sufficiency and basic technology choices have to be performed, for example, whether PHP is a suitable technology for the security service.

5. Conclusion

Through increased provision of geospatial data via service-based interfaces such as OGC WFS, WMS, SOS and others, security in distributed geo-infrastructures has become a central requirement. In contrast to previous approaches, which are mostly comprehensive implementations compliant to existing ICT standards, we propose a lightweight approach for securing distributed low-power environments such as geo-sensor networks. The basic requirements, which we identified for the security solution, are to provide a lightweight architecture for low-power environments, to support distributed geo-service infrastructures, to allow cost-efficient implementation, to feature compatibility with OGC-compliant geospatial filtering and to build on cost-free and license-free technologies. Furthermore, an underlying requirement is 'separation of concerns', that is, the functional detachment of the single components.

The first part of this article presents a survey of current security mechanisms for authentication and authorisation techniques, and separately for geo-authorisation mechanisms. Based on this survey, a lightweight and scalable security infrastructure was developed, which is tailored for use in distributed service environments.

The developed security framework comprises dedicated components for authentication, rule-based authorisation, and optimised administration and storage of access rules. These components are united in two services, (1) a *security service*, which serves as a proxy to handle geo-requests and (2) an *authentication, authorisation and accounting service* (AAAS), which handles user account data, enforces access rights and deals with the accounting process.

The implementation uses OAuth for authorisation, OGC Filter Encoding for storing authorisation rules and OpenID for authentication. As accounting is a highly application-dependent process, no generic solution can be created. The implementation presented in this article uses a simple logging mechanism for storing the number of geo-requests.

Our solution provides the possibility to secure geo-web services in distributed low-power infrastructures using a variety of open and freely available technologies. It is very simple to implement and maintain, and thus can also be used on embedded systems. Furthermore, the web services themselves do not have to be modified using our approach. Moreover, our solution can be flexibly used in a variety of scenarios and environments ranging from singular implementations to large-scale systems through integrability of external and existing authentication systems. Also, economic significance is given as argued in sub-section 4.4.

These aspects show considerable benefits over existing commercial solutions, which are mostly expensive and complex to implement because it enables cost-efficient security in geo-web service infrastructures without extensive license costs. Thus, we believe that our solution can be an important step towards the realisation of the vision of 'Digital Earth' through the elimination of security issues and consequently through overcoming barriers to open data access.

Acknowledgements

This work has been funded by the European Commission (FP7 project GENESIS, reference No. 223996) and the Austrian Federal Ministry for Science and Research. The authors would like to thank all involved groups at the Research Studios Austria – iSPACE.

References

52°North Initiative, 2011. *Security & geo-rights management community* [online]. Muenster, Germany, 52°North Initiative. Available from: http://52north.org/communities/security [Accessed 24 August 2011].

AM Consult, 2010. *GeoXACML – access control for geospatial data* [online]. AM Consult. Available from: http://www.geoxacml.org [Accessed 21 November 2011].

Ashley, P. and Vandenwauver, M., 1998. *Practical intranet security – overview of the state of the art and available technologies.* Boston, MA: Kluwer Academic Publishers.

con terra, 2011. *con terra – GIS expert for spatial data infrastructures FME & ESRI technology* [online]. Muenster, Germany, Con terra. Available from: http://www.conterra.de [Accessed 21 June 2011].

Craglia, M., *et al.*, 2008. Next-generation Digital Earth: a position paper from the vespucci initiative for the advancement of geographic information science. *International Journal of Spatial Data Infrastructures Research*, 1 (3), 146–167.

Craglia, M., *et al.*, 2012. Digital Earth 2020: towards the vision for the next decade. *International Journal of Digital Earth*, 5 (1), 4–21.

European Commission, 2010. *COMMISSION REGULATION (EU) No 268/2010 of 29 March 2010 Implementing Directive 2007/2/EC of the European Parliament and of the Council as Regards the Access to Spatial Data Sets and Services of the Member States by Community Institutions and Bodies Under Harmonised Conditions* [online]. Brussels, Belgium, European Commission. Available from: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri = OJ:L:2010:083:0008:0009:EN:PDF [Accessed 3 June 2011].

GENESIS Consortium, 2011. *GENESIS FP7* [online]. Cannes, France, GENESIS Consortium. Available from: http://www.genesis-fp7.eu [Accessed 19 June 2011].

Hermann, J. and Matheus, A., 2009. *OGC OWS-6 GeoXACML engineering report* [online]. Open Geospatial Consortium. Document Version 0.3.0. Available from: http://www.opengeospatial.org [Accessed 6 June 2011].

Institute of Earth Science, 2011. *GeoShield – division of geomatics* [online]. Institute of Earth Science, Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Canobbio, SUI. Available from: http://istgeo.ist.supsi.ch/site/projects/geoshield [Accessed 17 August 2011].

Internet2 Middleware Initiative, 2011. *Shibboleth* [online]. Ann Arbor, MI, Internet2 Middleware Initiative. Available from: http://shibboleth.internet2.edu [Accessed 28 May 2011].

Matheus, A. and Herrmann, J., 2011. *Geospatial eXtensible Access Control Markup Language (GeoXACML) Version 1 Corrigendum* [online]. Wayland, MA, Open Geospatial Consortium. Document Version 1.0.1. Available from: http://www.opengeospatial.org [Accessed 28 June 2011].

OASIS, 2011a. *Standards OASIS – Web Services Security SAML Token Profile 1.1* [online]. Burlington, MA, OASIS Consortium. Available from: http://www.oasis-open.org [Accessed 18 June 2011].

OASIS, 2011b. *Standards OASIS – Web Services Security 1.1* [online]. Burlington, MA, OASIS Consortium. Available from: http://www.oasis-open.org [Accessed 18 June 2011].

OASIS, 2011c. *Standards OASIS – eXtensible Access Control Markup Language (XACML) v2.0* [online]. Burlington, MA, OASIS Consortium. Available from: http://www.oasis-open.org [Accessed 18 June 2011].

OAuth Community, 2011. *OAuth community site* [online]. South Orange, NJ, OAuth Community. Available from: http://oauth.net [Accessed 10 September 2011].

OpenGeo, 2011. *Welcome – GeoServer* [online]. New York, NY. Available from: http://www.geoserver.org [Accessed 24 June 2011].

Open Geospatial Consortium, 2011. *OGC geospatial digital rights management reference model* [online]. Wayland, MA, Open Geospatial Consortium. Document Version 06-004r3. Available from: http://www.opengeospatial.org [Accessed 21 June 2011].

OpenID Foundation, 2011. *OpenID foundation website* [online]. San Ramon, CA, OpenID Foundation. Available from: http://openid.net [Accessed 21 June 2011].

ORCHESTRA Consortium, 2009. *Orchestra overview* [online]. ORCHESTRA Consortium. Available from: http://www.eu-orchestra.org [Accessed 15 June 2011].

Rivest, R.L. and Lampson, B., 2001. *CIS: SDSI (A Simple Distributed Security Infrastructure)* [online]. Cambridge, MA, MIT CSAIL. Available from: http://groups.csail.mit.edu/cis/sdsi.html [Accessed 27 December 2011].

Scherpenisse, A., 2011. *MiracleThings* [online]. Amsterdam, NED. Available from: http://miraclethings.nl [Accessed 19 August 2011].

The Library of Congress, 2008. *CQL: Contextual Query Language (SRU Version 1.2 Specifications)* [online]. Washington, DC, The Library of Congress. Available from: http://www.loc.gov [Accessed 3 September 2011].

Vretanos, P.A., ed., 2005. *OpenGIS filter encoding implementation specification* [online]. Wayland, MA, Open Geospatial Consortium. Document Version 1.0.0. Available from: http://www.opengeospatial.org/standards/filter [Accessed 14 June 2011].

W3C Networking Group, 1999. *HTTP authentication: basic and digest access authentication* [online]. W3C Networking Group. Available from: http://www.ietf.org/rfc/rfc2617.txt [Accessed 4 September 2011].

Zend Technologies Ltd., 2011. *PHP web application server – PHP development tools – PHP Training – Zend.com* [online]. Munich, Germany, Zend Technologies GmbH. Available from: http://www.zend.com, 2011 [Accessed 18 August 2011].