Reference: SCHLÄGER, C. & GANSLMAYER, M. (2007) Effects of Architectural Decisions in Authentication and Authorisation Infrastructures. Proc. of the 2nd International Conference on Availability, Reliability and Security (ARES'07). Vienna, Austria, IEEE.

Effects of Architectural Decisions in Authentication and Authorisation Infrastructures

Christian Schläger
Dpt. of Information Systems,
University of Regensburg, Germany
christian.schlaeger@wiwi.uni-regensburg.de

Monika Ganslmayer
R&L Inc.,
Landshut, Germany
monika.ganslmayer@rl-ag.com

Abstract

AAIs – Infrastructures for Authentication and Authorisation provide services for service providers on the Internet. Especially if combined with an attribute infrastructure these AAIs can offer additional functionalities like a single sign-on, enhanced privacy, strengthened trust and security, or improved usability. In respect to security and privacy, the AAI acts as a mediator within the client service provider relationship, or, more likely, the client federation relation. Since an AAI is a loosely coupled combination of services architectural decisions influence its effects on privacy and security focusing either on customer demands or service provider requirements. This work shows how architecture and allocation decisions alone can shape the security and privacy contribution of AAIs leading to different levels of contentment for the user groups.

1. Introduction

Service providers on the Internet are familiar with basic infrastructures supporting security services necessary to conduct business. These infrastructures usually support authentication and authorisation (so called AAIs). Providers can chose between prominent frameworks like Liberty's Identity Federation Framework [3], Microsoft's .NET Passport [11], Shibboleth [4], or the Spanish PAPI [5]. Depending on various factors these architectures are suited differently to address topical demands of service providers and customers in Internet-based transactions.

For service providers (SPs) these demands include a higher level of security through fine grained access control (AC) and additional information about customers and their reputation as well as the possibility to outsource security services to 3rd party providers, providing cheaper or better services [18]. Users require better usability with a single sign-on (SSO), central maintenance of account data, and the possibility to pass their reputation and trustworthiness from one provider on to another as well as the protection of their privacy, e.g. through the usage of pseudonyms rather then their real names. Of course, this list of user and provider demands is not complete. For an in depth discussion of stakeholders' demands see [17] and [18].

The functionality of AAIs can be divided into various sub-services: Authentication, authorisation, access control decision computing, and access control enforcement. Frameworks and products differ in as so far as services they include and how they provide a special service. Not all AAIs support all security sub-services. Microsoft's .NET Passport for example is only able to provide a SSO. In [18] AAIs have been clustered into different levels according to their capacity. In addition, the different AAIs have different paradigms they follow. Liberty's ID-FF for example distributes services among its federation members. PAPI consolidates all security sub-services at one single point, creating a proxy. Both issues, whether security sub-services are provided centrally or locally; by a third party or locally by the SP himself affect AAIs and how they are able to fulfil the requirements.

This work bases on an extended idea of AAIs. Through the integration of an attribute infrastructure and attribute-based access control (ABAC) additional functionalities are gained (see [15] and [18]). With XACML – the eXtensible Access Control Markup Language [12] – an open standard has been proposed that is able to express access policies [2]. XACML enables building complex policies that derive an access control decision from object and subject attributes, the first referring to resources, the latter to users. An important standard for the exchange of security information between service and identity providers is SAML – the Security Assertion Markup Language [13] – also maintained by the OASIS group.

Integrating an attribute infrastructure changes the chain of security services. The results are extension for the assignment of object attributes (resources), subject attributes (customers), environment attributes (e.g. time), and the creation of a corresponding policy as given in Fig. 1.

For the evaluation four main paradigms have been identified in existing frameworks and products. To compare these approaches an extension of the AAIs has been made to enhance their functionalities with ABAC.

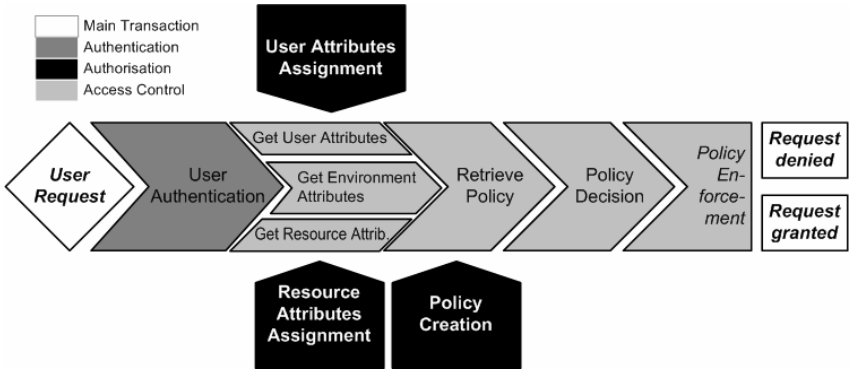

Figure 1. Attribute infrastructure security services

2. Related Work

2.1. Infrastructures for Authentication and Authorisation

AAIs make it possible to combine service outsourcing strategies with strengthened security and more flexible and suitable techniques. A special benefit lies in the accumulated user data over a federation: User profiles, buying patterns, and earned privileges. Identities could be transferred from one service provider to another making it possible to always use up-to-date address data or proof a good reputation acquired at one federation member. Comparative surveys on existing AAIs can be found in [10] and [17]. Schläger and Nowey [15] provide an AAI perspective focusing on risk assessment and risk identification.

The idea of outsourcing non-functional tasks has been discussed in the field of software engineering. A good résumé can be found in Tanenbaum and van Steen [19].

Katsikas et al sum up requirements in providing secure e-commerce [7]. The shown need for flexible and dynamic access control in e-commerce can be addressed with attribute-based access control as presented in [2] or [20].

As far as architectural decisions are concerned, two main approaches can be found: Firstly, the centralised approach of Microsoft's .NET Passport or PAPI and secondly, the distributed approach of Liberty's ID-FF or Internet2's Shibboleth. The reference architecture proposed by Schläger et al [18] mediates between these paradigms.

2.1.1. .NET Passport by Microsoft. Microsoft introduced Microsoft .NET Passport in 1999 to offer a single sign-on service in the Internet. We describe the processes as stated in the official Microsoft documents [11] and according to our own analysis:

When trying to log in at a resource, the user is forwarded to the .NET Passport log-in page. The resource's Passport ID is transported to passport.com using URL encoding. If it is registered and valid the user is forwarded to passport.net. The user authenticates with his username and password and is redirected to passport.com. Passport.com writes four cookies in the user's browser cache. Following to this, the user is forwarded to the resource. Finally, two more cookies are written to allow the user's single sign-on when he or she accesses the resource the next time.

2.1.2. Identity Federation framework by Liberty Alliance: In contrast to Microsoft .NET Passport, the Liberty Alliance develops only concepts and standards which allow compatibility between different implementations by third party companies. In general, Liberty offers the same functionality as Microsoft .NET Passport, however, it relies heavily on open standards like SAML and allows the use of federations and Circles of Trust (CoT) between different authentication services [3]. Services are distributed in the CoT making involved SPs act as identity and attribute provider for other members. Users decide upon their entry point into the federation for every

transaction. Via the SSO he or she takes his or her attributes from his momentary Identity Provider (IdP) to the SP. Consequently, transmitted attributes depend on the used IdP. The SP checks if the user was authenticated via a SAML request. The IdP sends back a SAML authentication assertion. Additional attributes can be send via a SAML attribute assertion. The SAML communication is always redirected via the user's browser.

2.1.3. Attribute-based AAI by Schläger et al. In [18] Schläger et al propose a reference architecture for an AAI including attribute-based access control. ABAC functionalities are integrated via the open standards SAML and XACML. The idea bases upon the Liberty ID-FF. However, Liberty's distributed paradigm is not followed. In order to mediate between provider and customer demands the architecture is only partly distributed. The user chooses his IdP among a list of 3rd party Identity Providers. He uses the one he trusts most. An external Policy Decision Point (PDP) evaluates the user's access request based on policies and user, environment and resource attributes.

The initial request is directed at the SP. The SP requests an authentication and access control decision from the AAI consisting of at least one IdP and a PDP. The IdP authenticates the user and requests his related attributes from all members. The authorisation request and the user attributes are transferred to the PDP. The PDP queries the SP for the resource attributes and uses the respective policies loaded at its initialisation. The computed access control decision is forwarded to the SP via the user's browser. Complying with the idea of a generic architecture the SP enforces this decision with its own means.

2.2. ABAC, SAML and XACML

The basic idea of attribute-based access control (ABAC) [2] [14] [20] is to use object and subject attributes as the basis for authorisations. For subjects, attributes can be static ones like a subject's position or role in a company or dynamic attributes like age, current location or an acquired subscription for a digital library. For objects, metadata properties, e.g., the subject of a document, can be used. Subjects and objects are both represented by a set of attributes and related values. Permissions are defined between subject and object descriptors which consist of sets of attributes, conditions, and an operation that is to be executed on the objects denoted by the descriptor.

XACML is an XML-based standard to describe attribute-based authorisation rules and policies. Furthermore, it specifies rules to process and combine these authorisation rules and policies. XACML entities comprise a Policy Enforcement Point (PEP), a Policy Decision Point (PDP), a Policy Information Point (PIP) and a Policy Administration Point (PAP). [12]

SAML is an XML-based standard to describe security information which is communicated between system entities and domains [13]. An integration with XACML has been proposed for example by [1].

3. Elements of an attribute-based AAI

AAIs not supporting authorisation and access control services can only stay superficial. They reduce themselves to mere single sign-on approaches neglecting the momentum gained through an integration of all security sub-services. Furthermore, AAIs need to integrate an attribute-based approach. The flexibility of ABAC guarantees fine grained control over privileges and rights as well as subsumes the traditional access control approaches. A holistic AAI consists of the four sub-services depicted in Fig. 1: Authentication, attribute and policy collection, policy decision, and enforcement

As each sub-service can be provided autonomously for the infrastructure, AAI architects need to decide for each sub service whether to provide it in a centralised manner, totally distributed over the federation, or in a manner in between. Additionally, service provider have the choice to integrated a sub-service from one central 3rd party provider, from a variety of providers, or not at all, resulting in an in-house realisation of the service.

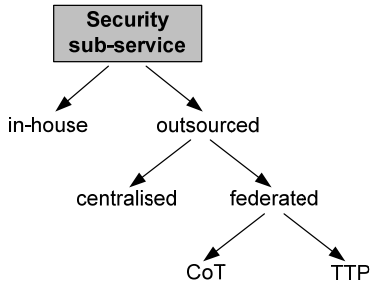

Figure 2. AAI security sub-services decision tree

Starting with the choice of the SP a decision tree evolves for an AAI as given in Fig. 2. Pros and cons of the four choices will be discussed for each of the aforementioned security sub-services. Criteria for the assessment are derived from [17] and [15]. The most important ones are:

-- generally: efficiency and availability
-- user specifically: usability and privacy
-- provider specifically: security and outsourcability

3.1. Authentication

Generally, the realisation of one single central authentication point results in a bottle-neck with a single point of failure. From the user perspective privacy issues arise in this case due to the enforced single point of authentication and trust. An identity provider can easily track a user profile. As a consequence the adoption of the service is doubtable. However, one single system of authentication for a large variety of services can result in higher usability. For providers the AAI is ideal for outsourcing strategies. One partner, bound by a legal contract with a certain service and liability level, is responsible for the services. Assuming the centralised service is run by a professional the security level might be higher [16].

In a CoT various SPs act as IdPs. The variety of providers strengthens availability. However, efficiency of the service is at a low. To trust this service a multitude of agreements is needed. As a consequence Liberty CoTs today are limited to a small number of participants. For privacy requirements the distribution of user data is ideal. Agreeing with [6] the distribution of data over a variety of provider usually enhances privacy and makes the creation of user profiles very hard. As a user needs to choose an IdP for his transactions first, usability can be slightly less. For SPs a CoT is neither ideal from a security perspective nor from the idea of outsourcing non-functional tasks: In a CoT the weakest member sets the security level for the whole circle [16]; synergies are not realised as all services need to be proffered still.

If various 3rd party providers are accepted as IdPs availability issues can be neglected. The user is not forced to accept a trust relation but can rather choose the IdP of his liking. On the other hand SPs are confronted with a limited number of 3rd party providers making efficient management and exchanges feasible.

Table 1 gives a summary of the stated possibilities. Column two depicts the case that no AAI is involved and the service is provided in-house. This case is used as the normalised reference (~) for occurring changes.

Table 1. Pros and cons of outsourcing authentication to an AAI

Authentication	*in-house*	*centralised*	*CoT*	*3rd Party*
Efficiency	~	++	-	+
Availability	~	--	++	+
Usability	~	++	-	++
Privacy	~	--	++	+/-
Security	~	++	--	++
Outsourcing	~	++	--	++

3.2. Attribute and policy collection

According to the XACML standard, policies are loaded at start-up by the PDP. Using one central policy simplifies its maintenance and up-to-date state. The more PDP provider are active, the likelier policy inconsistencies. For the attribute collection two approaches are possible [9]: Pull and Push. Using the Pull Model, the PDP explicitly asks every possible source for attributes relevant to the decision request. This can eliminate the aforementioned separation of identity and attributes. Still, the PDP would need lesser data. Using the Push Model the PDP gets all necessary attributes with the decision request. In this case the collecting party can be the IdP.

The centralised approach uses a central database for policy and attributes, maintaining user information and changes. Benefits lie in the efficient attribute gathering and the possibility of outsourcing to a professional provider. Nevertheless, Passport lacks adoption as providers seem hesitant to let their customer data be stored and maintained externally. Privacy issues evolve as a central IdP and attribute provider has extensive knowledge. Provided that attributes, policies, and management are correct a central approach can be seen as a very secure way to control access. Due to additional data about the customer from other SPs the decision process is more sophisticated and trusted.

The distributed approach in a CoT lacks efficiency and outsourcing benefits. Although availability can be guaranteed through redundancy, the attributes are scattered over the federation and access control decisions can not use the full potential of additional user information. On the other hand, privacy protection is very high. The idea of distributing attributes or user information is in itself a so called Privacy Enhancing Technology [6].

Table 2. Pros and cons of outsourcing attribute management to an AAI

Authentication	*in-house*	*centralised*	*CoT*	*3rd Party*
Efficiency	~	++	--	+
Availability	~	--	++	+
Usability	~	not applicable		
Privacy	~	--	++	+
Security	~	++	-	+
Outsourcing	~	++	--	+

Comparing the methods of attribute collection gives the best idea in how far the centralised and distributed approach can differ. Still, a 3rd party provider could mediate in a federated environment between these effects. A limited number of providers could guarantee common standards and efficiency, availability, and outsourcing benefits. If a user can choose his trusted IdP, privacy can be protected and yet a PDP could get

enough data to perform a secure and trusted decision. The potential to mediate can be seen in Table 2.

3.3. Policy decision

The policy decision itself is based on attributes and properties, not on identities. Based on a policy and attributes the PDP decides if usage is granted. This separation, naturally, enhances privacy. Users do not interact with the PDP directly. The PDP is a very important point in the chain of security services and trust is needed from the service providers in this entity.

A centralised approach needs to be especially reliable for the policy decisions. Still, an approach guaranteeing availability could use the benefits of effectiveness and correctness. However, the central management of policies can lead to lesser flexibility in expressing rules. Due to its importance and availability requirements the outsourcing of a PDP to a reliable provider implies a big convenience for SPs.

In a distributed environment every SP should maintain a PDP making it independent from an external provider. This approach is, again, far apart from being effective. If a SP uses his own PDP solely for his own transactions and decisions it might be possible to formulate more precise rules. On the other hand the idea of outsourcing this non functional service is neglected. Following the CoT idea a SP's PDP will not only be used by the SP providing the service but by other SPs as well. In this case an SP has to bear a considerable overhead in keeping multiple policies up-to-date.

Table 3. Pros and cons of outsourcing policy decisions to an AAI

Authentication	*in-house*	*centralised*	*CoT*	*3rd Party*
Efficiency	~	++	--	-
Availability	~	--	++	++
Usability	~	not applicable		
Privacy	~	++	++	++
Security	~	+	-	+/-
Outsourcing	~	++	--	++

If a PDP is offered by various 3rd party providers, principally, the same problems arise as in a CoT. However, outsourcing benefits for SPs exist. The effort for policy management ranges between the centralised and the distributed approach. Table 3 summarises the approach.

3.4. Policy enforcement

The policy enforcement point, intercepting the user request, is required at the beginning and the end of the transaction. The decision is enforced through the decoding of a XACML authorisation statement concerning the client in question. Its availability is crucial. Outsourcing the PEP is only practical as a proxy solution (see PAPI [5] and [18]). The alternative is the local enforcement of the AC decision.

4. AAI ARCHITECTURES

This work analyses four paradigms of AAI architectures. The first two are motivated from a software engineering point of view. The third and fourth base on the demands and requirements listed in the introduction. The four paradigms are:

1) A completely centralised architecture,
2) a completely distributed architecture,
3) a user centred architecture, and
4) a provider centred architecture.

Using effects and benefits analysed in section III AAIs have been designed following one of the aforementioned patterns of thought. In contrast to the previous section, this approach could be considered top-down. The aim is not to dissect the AAI into independent modules but rather to take a holistic approach, using the analysed modularised elements. Each architecture is described by an UML 2.0 sequence diagram using SAML, XACML nomenclature and, for clarity reasons, by a cross chart. The cross chart depicts the constructed architecture arranging AAI sub-service according to two dimensions, derived from Fig. 2.

4.1. The centralised AAI approach

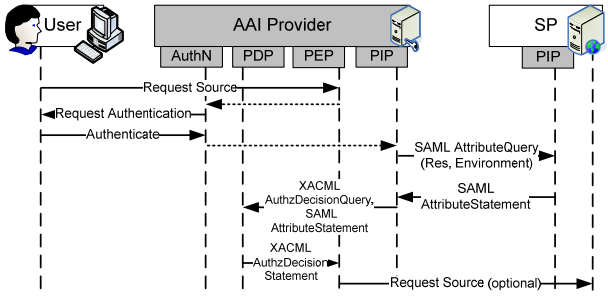

Figure 3. Centralised AAI reference (UML 2.0)

Using Microsoft's .Net Passport as a basis, we follow the centralised third party approach. If possible, security sub-services should be handled by the infrastructure. The user's request is intercepted by the AAI provider who collects user attributes from his central database and adds environment and resource attributes from the requesting SP. After computing an AC decision using the central policy, the AAI enforces the decision. If access is granted, the AAI fetches the resources from the SP and refers it. See Fig. 3.

Allocating the sub-service in our cross chart (Fig. 4) we see that, although central service providing was a basic requirement, attributes about resources and the environment need to be retrieved from the SP. The AAI acts as a proxy between the user and the resource.

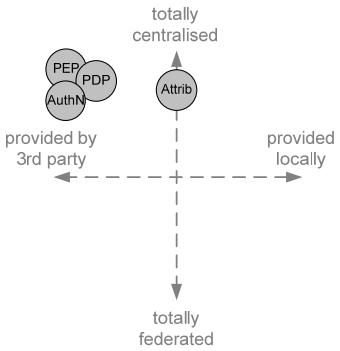

Figure 4. Centralised AAI reference (dimensions of allocation)

4.2. The distributed AAI approach

Contrasting the centralised approach, the distributed AAI architecture tries to choose as many entities as possible for each service. As a consequence, every member of such a Circle of Trust needs to offer every security sub-service to all other members in the federation. The only exception is the PEP. As stated in section 0 it is not feasible to let other CoT members enforce the AC decision. The only opportunity to do this would be a proxy approach. A completely distributed attribute-based AAI works as depicted in Fig. 5: The user requests a resource at SP-1. He chooses to authenticate himself at SP-2. This service provider adds known attributes about the user to the authentication assertion and sends both back via the user's browser to SP-1. The user's attributes are completed with resource and environment attributes and relayed to any PDP in the federation, e.g. SP-3. The returned policy decision is enforced by the local PEP from SP-1.

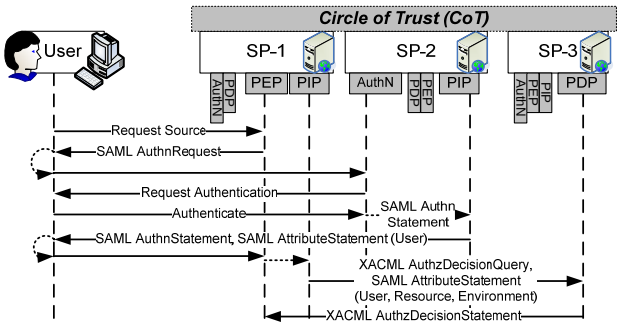

Figure 5. Distributed AAI reference (UML 2.0)

This procedure bases on the Liberty Alliance's ID-FF. Fig. 6 shows the sub-service's allocation. All except the PEP are placed exactly opposite the centralised version of Fig. 4.

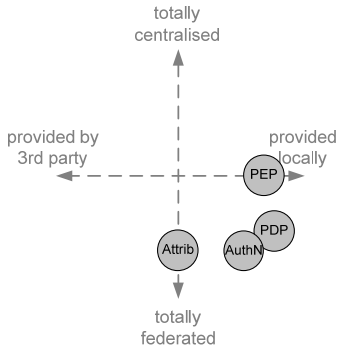

Figure 6. Distributed AAI reference (dimensions of allocation)

4.3. User focused AAI approach

A user focused architecture needs to especially respect privacy and usability requirements. Consequently, identities and attributes need to be separated, a trust relation mustn't be forced upon the user, and the rules of data canniness obeyed. Additionally, the infrastructure should still be usable. Wanted functionalities include account and profile management, reputation sharing, and SSO. The afore presented, completely distributed architecture might be privacy preserving, however, it is not able to meet usability demands.

The original software engineering idea of reference monitors [8] already states that context information from the application must be respected. Consequently, the enforcement of a policy decision in an AAI needs to be realised at the requested resource. The target system is able to use more detailed, up-to-date, and system specific information for its decision.

The architecture presented in Fig. 7 separates identity and attributes for the AC decision. The decision is computed at a trusted third party (TTP). The TTP's PIP collects user attributes from the chosen IdP and environment as well as resource attributes from the SP. The IdP authenticates the user and extends user profile data stored at his site by attributes from all federation members. To do this, he needs a common user identifier. The access request itself, however, needs to be handled by an opaque ID (see e.g. Liberty ID-FF [3]). The protocol's extent results from the laborious user attribute requests. Ideally, the IdP stays with the user for every transaction in the federation. This provider stores and manages his account. Still, attributes like the user's reputation need to be asked from SPs. Additional privacy is gained as the IdP is not aware of the requested resource.

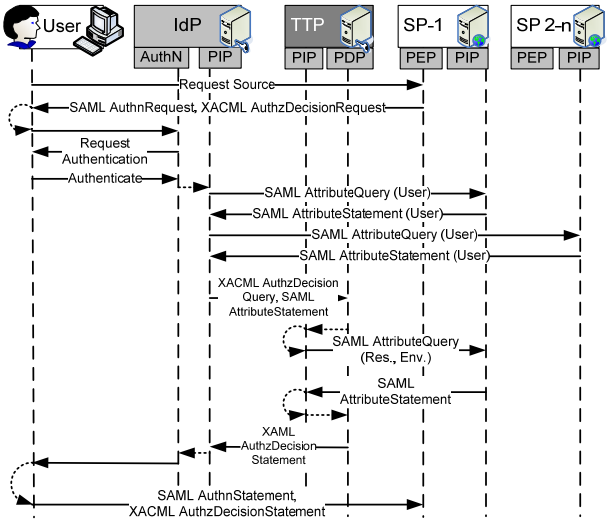

Figure 7. User focused AAI reference (UML 2.0)

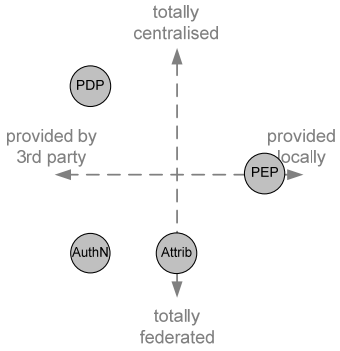

Figure 8. User focused AAI reference (dimensions of allocation)

Allocating the sub-services, we see that the source for attributes is the federation of SPs in addition to the main user database of the authenticating IdP. The decision is enforced locally. (Fig. 8)

4.4. Provider focused AAI approach

The criteria for a provider-centred approach are security and outsourcability. Security, in this case, derives from expressive user attributes enabling fine grained access control and trusted decisions. In contrast to the user focused approach fewer parties are involved and user data stays with the SPs. Changes as well as user behaviour should be mirrored back to the SPs. In the provider focused approach only one AAI provider is active. Due to the same reasons stated in the user focused approach, each SP maintains a PEP.

In this architecture a request is intercepted by the SP's PEP. The AAI is questioned about the user identity. After the SAML assertion has been sent back, the SP decides upon authorisation by requesting a decision from the AAI. Using the push-model the SP attaches necessary attributes with the XACML request. If further data needs to be gathered, other SPs are questioned about the user, this time using the pull model. The decision is send back to the SP. (Fig. 9)

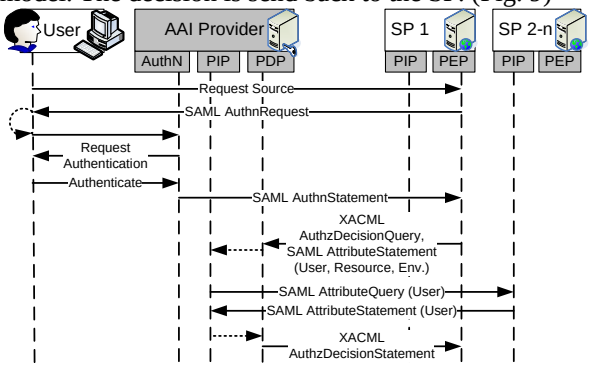

Figure 9. Provider focused AAI reference (UML 2.0)

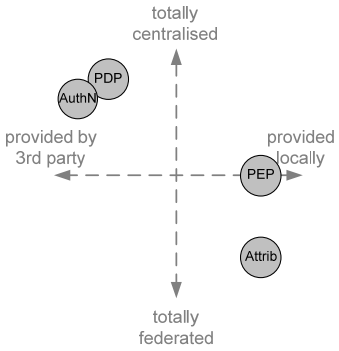

Figure 10. Provider focused AAI reference (dimensions of allocation)

Following the paradigm of a provider focused AAI, most sub-services are provided by one AAI provider. In contrast to the completely centralised option, here, the user data stays in the realm of the SP. Only what seems to be necessary for the decision process is transferred to a centrally managed PDP. If the SP sees a necessity, more user data is fetched from other members of the AAI provider knowing the customer. Although services should be outsourced if possible and practical, the SP still operates a PIP. This service is needed to retrieve the attributes from CRM, ERP, or other database systems. Fig. 10 shows the allocation.

5. Conclusion

This paper presents a thorough analysis of AAI architectures, focusing on the modularisation and allocation of security sub-services. The usage of an AAI is always reasonable if synergies between one or more service providers want to be exploited. Through the integration of attribute-based access control additional functionalities are gained and the security

and trust level of access control decisions is strengthened.

Based on existing frameworks, open standards, and commercial products a methodology has been deducted dividing the chain of security services in various security sub-services, each able to operate autonomously. Allocation possibilities have been identified according to a decision tree (Fig. 2).

Analysing the pros and cons of the diversification of security sub-services rules and effects have been presented that shape an AAI and its contribution for the involved parties. Using four identified paradigms of constructing AAI architectures, we have shown that extreme forms of allocation like the completely federated or centralised approach result in suboptimal infrastructures with doubtable adoption by any of the involved parties. However, using basic requirements of users and SPs, infrastructures are possible mediating between the stakeholders.

Solely through the decision on the allocation of sub-services an immanent impact on security, usability, privacy, and outsourceability is obtained. The functionalities of the sub-services stay the same for every architecture. This has been shown by four contrasting approaches and resulting distributions of security sub-services. In AAIs service orientation and service allocation affect each other directly.

References

[1] Anderson, A., Lockhart, H.: SAML 2.0 profile of XACML. http://docs.oasis-open.org/ (2004).

[2] Busch, S., Muschall, B., Pernul, G., Priebe, T.: Authrule: A Generic Rule-Based Authorization Module. In: Proc. of the 20th Annual IFIP WG 11.3 Working Conference on Data and Applications Security (IFIP 11.3), Sofia Antipolis, France (2006).

[3] Cantor, Scott, Kemp John: Liberty ID-FF and WSF Protocols and Schema Specification. http://www.projectliberty.org/specs/draft-liberty-idff-protocols-schema-1.2-errata-v3.0.pdf (2005).

[4] Cantor, Scott: Shibboleth Architecture, Protocols and Profiles, 10 Sept. 2005. http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-latest.pdf (2005).

[5] Castro-Rojo, R., Lopez, D. R.: The PAPI system: point of access to providers of information. Computer Networks 37 6, Pages 703-710 (2001).

[6] Federrath, F.: Privacy Enhanced Technologies: Methods, Markets, Misuse. Proc. 2nd Internat. Conference on Trust, Privacy, and Security in Digital Business (TrustBus), Goteborg, Denmark, 2005. LNCS, Volume 3592, Pages 1-9 (2005).

[7] Katsikas, S. K., Lopez, J., Pernul, G.: Trust, Privacy and Security in E-Business: Requirements and Solutions. Proc. 10th Panhellenic Conference on Informatics (PCI), Volas, Greece, 2005. LNCS, Volume 3746, Pages 548-558 (2005).

[8] Lampson, B., Abadi, M., Burrows, M., and Wobber, E.: Authentication in Distributed Systems: Theory and Practice. ACM Transaction on Computer Systems, Vol. 10, No. 4, Pages 265-310, (1992).

[9] Lopez, G., Gomez, A.F., Marin, R., Canovas, O.: A Network Access Control Approach Based on the AAA Architecture and Authorization Attributes. In. Proc of the 19th IEEE Internat. Parallel and Distributed Processing Symposium (IPDPS'05)-Workshop, Pages 287-295 (2005).

[10] Lopez, J., Oppliger, R., Pernul, G.: Authentication and Authorization Infrastructures (AAIs): A Comparative Survey. Computers & Security 23 7, Pages 578-590 (2004).

[11] Microsoft: Microsoft.NET Passport Review Guide. http://download.microsoft.com/download/a/f/4/af49b391-086e-4aa2-a84b-ef6d916b2f08/passport_reviewguide.doc (2004).

[12] OASIS eXtensible Access Control Markup Language TC: eXtensible Access Control Markup Language (XACML). http://www.oasis-open.org/committees/ (2006).

[13] OASIS Security Services TC: Security Assertion Markup Language (SAML). http://www.oasis-open.org/committees/ (2006).

[14] Priebe, T., Fernandez, E.B., Mehlau, J.I., Pernul, G.: A Pattern System for Access Control. Proc. of the 18th Annual IFIP WG 11.3 Working Conference on Data and Application Security (DBSec 2004), Sitges, Spain (2004).

[15] Schläger, C., Nowey, T., Montenegro, M.: A Reference Model for Authentication and Authorisation Infrastructures Respecting Privacy and Flexibility in b2c eCommerce. 1st Internat. Conference on Availability, Reliability and Security (ARES) 2006, IEEE Computer Society, Pages 709-716 (2006).

[16] Schläger, C., Nowey, T.: Towards a Risk Management Perspective on AAIs. Proc. 3rd Internat. Conference on Trust, Privacy & Security in Digital Business (TrustBus), Krakow, Poland, 2006. LNCS, Volume 4083, Pages 41-50 (2006).

[17] Schläger, C., Pernul, G.: Authentication and Authorisation Infrastructures in b2c e-Commerce. Proc. of the 6th Internat. Conference on Electronic Commerce and Web Technologies (EC-Web), Goteborg, Denmark, 2005. LNCS, Volume 3590, Pages 306-315 (2005).

[18] Schläger, C., Sojer, M., Muschall, B., Pernul, G.: Attribute-based Authentication and Authorisation Infrastructures for E-Commerce Providers. Proc. of the 7th Internat. Conference on Electronic Commerce and Web Technologies (EC-Web), Krakow, Poland, 2006. LNCS, Volume 4082, Pages 132-141 (2006).

[19] Tanenbaum, A. S., van Steen, M.: Distributed Systems: Principles and Paradigm. Upper Saddle River, NJ: Prentice Hall (2002).

[20] Yuan, E., Tong, J.: Attribute Based Access Control (ABAC) for Web Services. Internat. Conference on Web Services (ICWS) 2005, IEEE Computer Society, Pages 561-569 (2005).