

Architectures for access control in telecommunication networks

December 20, 2006

Abstract

In this document we discuss access control for telecommunication networks. We start by presenting some security requirements for telecom networks and an example of access control structure in a Radio Access Network. Then we show different alternatives for access control architectures for use in telecommunication networks and compare their advantages and drawbacks. Finally we draw some conclusions and present future directions for the work.

The document is an intermediate report of the PRIMA-NET project and is not intended to promote a specific architecture, but rather as a tool to help in the selection of the most appropriate one which will depend on other system and business requirements yet to be defined.

1 Introduction

This document reports the first set of results of the activity, architecture, in the *Privilege management in large distributed telecommunication networks (PRIMA-NET)* project. PRIMA-NET is a collaboration project between the Swedish Institute of Computer Science (SICS) and Ericsson Research, and the project is funded by Vinnova.¹ The goal of this activity is to investigate how access control mechanisms can be part of the overall architecture of a telecommunication network taking into consideration various requirements posed on access control procedures in the network.

Building a telecommunication network represents a substantial investment for a telecommunication operator, that is to be amortised by future revenues from subscriber traffic in the network. Any outage or disruption of service is likely to mean loss of revenue and hence will harm the operator economically. As any error in the configuration of such complex systems can impact service performance, a common sense security policy is to reduce the number of people who have access to management functionality to a minimum. To further reduce the risk for errors it is desirable to also be able to designate portions of the network and subsets of management functionality operating personnel has access to.

Telecommunication networks consist of a large number of geographically distributed nodes. Managing access to these nodes become itself an error-prone

¹Project number 2005-02501. <http://www.sics.se/primanet>

and security critical task. A secure and efficient management of access privileges to these distributed nodes increases the overall level of security of the network.

In this document we analyse some architectural requirements of such an access control system. We come forth with several design alternatives and discuss their advantages and drawbacks with respect to the requirements of telecommunication providers. At this stage we have not included considerations about the specific internals of the access control system and focus instead on which components are needed and these components are to be deployed.

2 Problem Definition

Telecommunication networks consist of *several integrated sub-systems*, which are partly *developed independently on different platforms* and using different software technologies (e.g. using different operating systems).

Moreover, a telecommunication network is typically *geographically spread* to support national and often international coverage of services. This requires *support for central as well as local management* of the nodes including management of access privileges to these nodes.

In addition there are *bandwidth restrictions* for Operation and Management traffic, as this traffic is weighed against the (directly) revenue generating traffic on the same network.

Another important constraint when dealing with telecommunications networks is the *extreme availability requirements on system uptime*. To get a rough idea of the order of magnitude, disrupting traffic on a telecommunication network of a large operator may cost 100.000 USD per minute or more. Furthermore business-to-business contracts regulating a telecommunication service can have severe penalty clauses for failure to conform with the availability requirements.

Today many of the sub-systems in a telecommunication network have their own proprietary security solution including some access control system. In current network management, SNMP [?] is a common tool used, but the usage is often limited to passive features - collecting logs and status information from network nodes. Setting parameters and actively affecting the operation of a node is typically done with manual command line interfaces, which are different for each manufacturer and device model. For heterogeneous networks built of such sub-systems, this most likely means increased costs both for integrating the different sub-systems and for training adequate personnel.

The study undertaken and reported in this report address the problem and scenario outlined above from an access control architecture point of view and subject to the following

Working assumptions:

- *Standardised access control*. In order to ensure interoperability and reduce the integration cost, a standardised access control solution is proposed to be used. If the standard is sufficiently mature and has widespread industry support, this also allows to take advantage of the previous experience of the standard makers and to have a solution that can easily adapt to future paradigm changes.

- *Well-defined policy language.* In current systems management of access control is often done manually, by assigning privileges according to policies written on paper in natural language. Such a form of privilege management is time consuming and error-prone, as the implementation of privileges is often left to a small group of administrators that have not been part of the decision making process. In such a situation misunderstandings can easily happen when interpreting natural language policies. We propose to use a well-defined policy language, together with a processing model that defines, how decisions can be calculated from a given set of policies. This will reduce misunderstandings and enforce a strict interpretation of policies.
- *Decentralised access control management.* Delegation mechanisms can help to spread the load of access control management, by allowing the delegation of well-specified authority to the corresponding decision makers. This reduces the overhead and speeds up the decision making process. Within the system, these decision makers would be able implement their decisions themselves, thus reducing the risk of misunderstandings. Such a decentralised solution is also more resilient against failures or sabotage.

2.1 Threats, Security services and Protection objects

Having established the need for an access control architecture we now consider how it fits into the rest of the security system. These considerations are based on the operational security current practises draft [2].

We start by enumerating generic threats and how they apply to telecommunication network management.

- *Unauthorised disclosure.* Here we are concerned about service data or configuration data, that an attacker may either use in order to prepare an attack that disrupts the service or to gain a competitive advantage.
- *Deception.* Here the threat would be an attacker inserting false service or configuration data into the system. This could lead to disruption of the service or the attacker could use services without paying.
- *Disruption.* The attacker in some way disturbs the telecommunication service. This can be vandalism, revenge or to gain a competitive advantage. The most common form is a denial of service, although more subtle degradation of service quality is also thinkable.
- *Usurpation.* The attacker takes control of some service. This means that the attacker not only uses this service without paying, but can also shut out authorised users of the service.

We now proceed by presenting security services that are commonly used. We focus on which role access control plays for those services or how they interact with access control.

- *Authentication.* We differentiate between actor authentication (including users and services) and data origin authentication. Access control is dependent on actor authentication in order to make access control decisions.

We therefore need to design the access control system in a way that permits to interact with different authentication systems (e.g. login/password schemes, SSH, SSL and other X.509 based PKIs).

Data origin authentication is part of communication security and should be covered by protocol at this level (e.g. SSL/TLS or IPSec). For this document communication security is considered out of scope.

- *Data Integrity and Confidentiality.* Like data origin authentication these security services are covered by a communication protocol and therefore not in the scope of this document.
- *Authorisation.* This is the goal we want to achieve with an access control architecture.
- *Auditing/Logging.* This will be the focus of some later document. However it should be noted, that the access control system is gateway function to system resources. It is therefore ideally suited to provide support for logging and auditing.
- *Denial of Service Mitigation.* While this is not the main focus of the access control architecture, it can have an important role in helping these efforts, by keeping unauthorised users away from critical services. It can however become part of the problem itself, if it can be misused to shut out users that should be authorised.

Finally we consider the objects that we want to protect against the aforementioned threats.

- *Physical devices.* This means that we want to prevent unauthorised physical access to the devices. Measures includes doors, locks and guards. Such protection is not in the scope of this document.
- *Data on device.* This includes configuration and log data. The data is to be protected for both in-band and out-of-band access.

A special case here are software updates, where the protection is very difficult to realise.

- *Data in transit.* Data that travels over the network is out of the scope of this document. We assume that some network security protocol (e.g. TLS or IPSec) will take care of this protection.

2.2 Application Example: A Radio Access Network

In this section we present a real world application example to which our architecture considerations will apply. We choose a WCDMA² Radio Access Network (RAN), consisting of *Radio Base Stations* and *Radio Network Controllers*. These network elements number in the thousands and are built using Ericsson's *Connectivity Packet Platform (CPP)*.

CPP is a platform product from which it is possible to develop a packet handling network node, e.g. an ATM switch, a Radio Base Station, a Radio Network Controller or a Media Gateway.

²Wideband Code Division Multiple Access

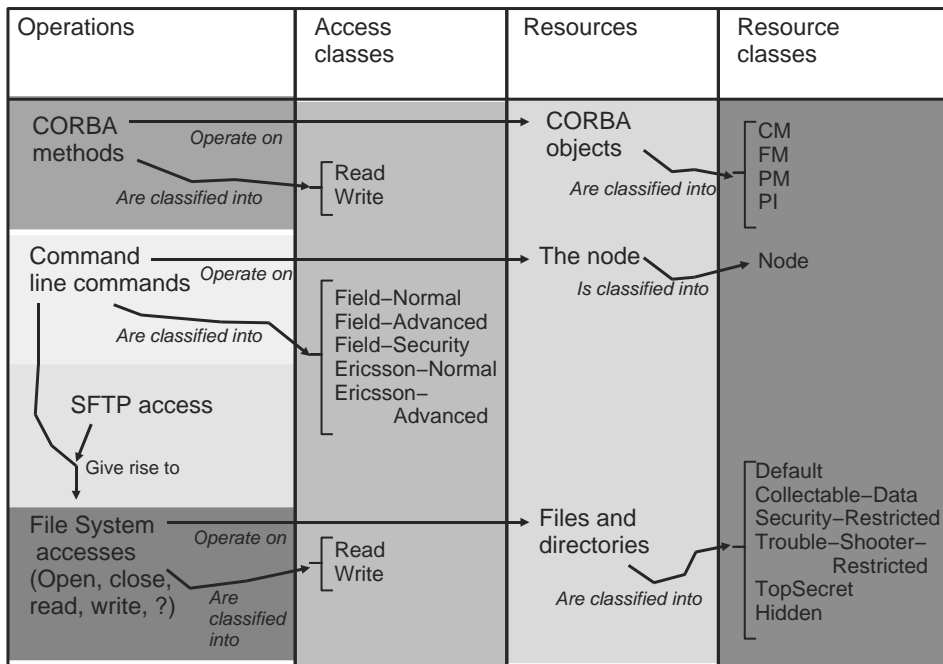


Table 1: Access and Resource classes in the Ericsson RAN architecture

CPP consists of two parts, an *Application Development Environment* which is not relevant in this example and the *CPP Node Component* which provides a distributed real-time control system, an element management system, media processing and a packet transport system with all the required hardware.

The network features a subnetwork with the task of providing a management environment for the RAN. The main operation and maintenance (O&M) interface to this subnetwork is the *Operations Support System for Radio and Core (OSS-RC)* another Ericsson product that fulfil the following tasks:

- Configuration management
- Supervision and fault management
- Performance management
- Security management

Network elements are managed through *CORBA interfaces, command line commands and file transfer services*. The operation and maintenance traffic is only guaranteed 64 kbit/s as the main part of available links are reserved for customer traffic.

In order to hide the low level complexity, Ericsson has defined a set of *Access classes* and *Resource classes* for this RAN architecture. These are shown in table 1.

Based on these Ericsson has defined a set of *Access profiles* as a prepackaged collection of permissions (i.e. access class operating on a resource class). These are summarised in table 2.

<u>PROFILE ReadOnly</u>	<u>PROFILE PM-Normal</u>
Read Default	<u>EXTENDS ReadOnly</u>
Read CM	Field-Normal Node
Read FM	Read Collectable-Data
Read PM	
Read PI	
	<u>PROFILE PM-Advanced</u>
<u>PROFILE CM-Normal</u>	<u>EXTENDS PM-Normal</u>
<u>EXTENDS ReadOnly</u>	Write Collectable-Data
Write CM	Write PM
Read Application-Data	
	<u>PROFILE SecurityManagement</u>
<u>PROFILE CM-Advanced</u>	<u>EXTENDS ReadOnly</u>
<u>EXTENDS CM-Normal</u>	Field-Normal Node
Read Trouble-Shooter-Restricted	Field-Security Node
Field-Normal Node	Read Security-Restricted
Write Application-Data	Write Security-Restricted
<u>PROFILE FM-Normal</u>	<u>PROFILE EricssonSupport</u>
<u>EXTENDS ReadOnly</u>	<u>EXTENDS ReadOnly</u>
Field-Normal Node	Field-Normal Node
Read Collectable-Data	Field-Advanced Node
	Ericsson-Normal Node
<u>PROFILE FM-Advanced</u>	Read Collectable-Data
<u>EXTENDS FM-Normal</u>	Write Collectable-Data
Write FM	Read Trouble-Shooter-Restricted
Field-Advanced Node	Write Trouble-Shooter-Restricted
Write Collectable-Data	
Read Trouble-Shooter-Restricted	

Table 2: An example access profile hierarchy with corresponding privileges

The profiles are intended to be interpreted the following way:

Read Only - Personnel collecting network configuration and performance data from the node. The Read Only profile gives the user permission to perform a reading operation over CORBA, such as reading the Managed Object Model but not changing anything. The user with a Read Only profile is also able to read all files in the system that do not have any read restrictions. The Read Only profile does not have any permissions for shell commands.

CM Normal - Personnel responsible for network configuration and planned area configuration. The CM Normal profile inherits permissions from the Read Only profile. In addition, the profile also gives permissions to perform write operations on the CORBA CM service, that is the Managed Object Model.

CM Advanced - Personnel responsible for software upgrades and hardware extensions. The CM Advanced profile inherits permissions from the CM Normal profile. In addition, the user with CM Advanced profile has access to shell commands that are needed for software upgrades and hardware extensions.

FM Normal - Personnel responsible for alarm handling, node supervision and sending trouble reports. The FM Normal profile inherits permissions from the Read Only profile. In addition, the profile also gives permissions to access shell commands that are needed for alarm handling, node supervision and sending trouble reports.

FM Advanced - Personnel responsible for troubleshooting. The FM Advanced profile inherits permissions from the FM Normal profile. FM Advanced also has access to shell commands and access to files that are needed for troubleshooting.

PM Normal - Personnel responsible for collecting performance data. The FM Normal profile inherits permissions from the Read Only profile. In addition, the profile also gives permissions to access shell commands that are needed for alarm handling, node supervision and sending trouble reports.

PM Advanced - Performance configurations. The PM Advanced profile inherits permissions from the PM Normal profile. PM Advanced also has access to shell commands and access to files that are needed for configuration of performance counters.

Security Management - Personnel responsible for O&M security on the node. The profile, Security Management inherits permissions from the Read Only profile. The Security Management profile also has access to shell commands and files that are needed to operate the O&M security.

Ericsson Support - Troubleshooters from Ericsson. The Ericsson support profile is just for Ericsson personnel and includes permissions to all commands and files that are needed for advanced troubleshooting. The Ericsson support profile does not have access to shell commands and files that are needed to operate the O&M security.

This example gives us the layout of the platform and a structure for the access control policies. For these we now need to find a good architecture of access control components in order to support organisation and network management without disrupting customer traffic.

2.3 Specific telecom requirements

In this section we list some requirements for an access control system that are particularly relevant in the telecom setting.

High reliability: High reliability of the access control system deals with the problem of maintaining availability and access control even in certain error situations. High reliability of the access control solution is especially important due to extreme availability requirements.

Interdependencies with external services: The access control solution should exhibit a minimum of interdependencies with external services. I.e., the access control solution should not fail only because some other services does. This is a particular instance of the availability requirement.

Central and local administration possible: Central and local administration of user rights should be possible. This means that it should be possible to change rights using both in-band methods as well as out-of-band methods.

Consistent data about access control state: It should always be clear in which state the access control data at some node is. If there is a central management point, it should have a picture of the actual state of the node it controls, and not only of what commands where sent out.

Low bandwidth use: The access control solution should not directly or indirectly require or cause a large amount of data to be transferred in the running system. Low bandwidth for operations and management traffic is desirable because the O&M traffic competes with the customer generated traffic that the operator can charge for, and so can be considered a cost. In a CPP system, O&M traffic is restricted to 64 kbit/s. Low bandwidth implies e.g. that it is not possible to do callbacks over the network for access control decisions and that massive, simultaneous updates access control information on all network elements are not an acceptable behaviour either.

End user privileges enforced in the Nodes: End-user privileges should be enforced in the network element. A good security practise is to apply access control close to the resource that needs to be protected. (A related principle with respect to time instead of space is *time-of-check vs time-of-use* which states that the access should be granted soon after the access rights are verified.)

Simplicity: The access control system should be easy to set up, use, and administrate. This can be considered a security requirement: A system that that is difficult to set up, use or administrate is likely to be bypassed or just managed in the wrong way, which in many cases constitute a security risk.

3 Architecture alternatives

In order to provide access control for telecommunication devices (Nodes) we need to design a distributed access control architecture. We choose to base this architecture on the RFC 2904 [7], which defines a set of standard access

control components. The reasons for this are that the RFC 2904 is widely used. Furthermore the access control standard XACML [1], that we plan to use for implementing this architecture, also follows the definitions of RFC 2904.

The components defined within the RFC are:

- The *Policy Decision Points (PDP)*, where access control decisions are made based on policies and other information
- The *Policy Enforcement Points (PEP)*, where a PDP's decision is enforced.
- The *Policy Retrieval Points (PRP)*, from where relevant policies are retrieved in order to be evaluated for a specific request.
- The *Policy Information Points (PIP)*, that provide additional information to PDPs, such as user roles, time of day etc.

In addition to these, the XACML standard also defines Policy Administration Points (PAP), as components that allows the administration of policies and policy information, including the distribution of this information to relevant PRPs and PIPs. The generic architecture of these components is shown in figure 1.

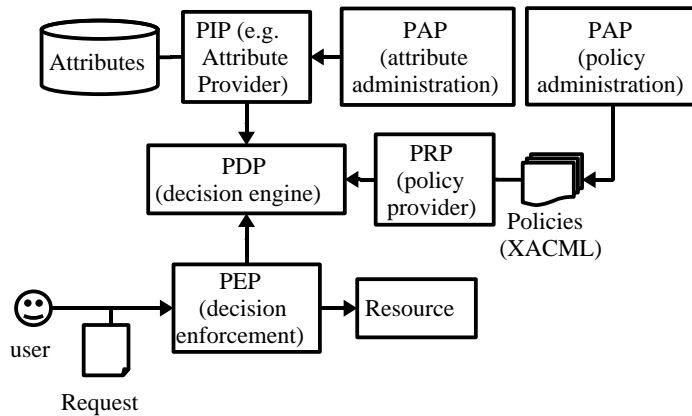


Figure 1: Generic access control architecture

XACML defines the notion of *attribute* to describe any information about users, resources and actions (e.g. roles, resource collections). Attributes are provided to an XACML PDP either via a PIP or through the PEP directly.

Since attribute management is out of the scope of XACML, we use a solution developed by SICS for the PIPs. These *Assertion Servers*³ are also capable to manage attribute hierarchies.

We use the XACML attributes in order to model the application presented in 2.2. We thus define the operations, access classes, resources and resource classes from table 1 as XACML attributes. We use attribute hierarchies to define the grouping of several operations into access classes and of several resources into resource classes. We can furthermore represent the access profiles as XACML

³Available from http://www.sics.se/spot/assertion_server.html

attributes and represent the *EXTENDS* relation by attribute hierarchies. The assignment of permissions to the access profiles is represented in policies that are stored in the PRP's.

Thus we are able to reproduce the design that Ericsson has provided for access control in their CPP platform.

In such a system the user and some of the components need to be authenticated to each other. The method of authentication is not relevant for this document, note however that besides the result of the authentication, the authentication method can be input data to the PDP. This data could be referred to by some policies that require the user to be authenticated by some specific mechanism deemed to be more secure for certain actions.

The data traffic between the components of the system needs integrity protection, if the components do not run in the same process. We assume that this is taken care of by some secure communication protocol such as TLS or IPsec, or by using "object security" such as digitally signed certificates/assertions. In the examples below, and without loss of generality, we assume the latter.

The main architectural question we explore in this section, is the physical location and interconnection of these components. For this we have designed three different such architectures and analysed their generic advantages and drawbacks. Furthermore we have tried to determine if any constraints from the telecommunications domain influence the decision on which architecture to choose.

3.1 Centralised authorisation enforcement

In this architecture all services providing access control data (PRPs and PIPs) are located centrally on a system called the *Operational Support System (OSS)*. The OSS clients use some form of digital signature to provide the requests to the agents on the nodes. The nodes must have the capability to verify such signatures. A message freshness mechanism is also required to prevent replay attacks.

Access control decisions are also made at the OSS level, meaning that the OSS needs to include a PDP. Furthermore the enforcement of these decisions is also handled by a PEP dedicated to the OSS. After enforcing the access control decision, the OSS signs the request and uses a dedicated *client (C)* to relay authorised user commands to the nodes. The commands are received by an *agent (A)* at the node which verifies the signature of the OSS and then unconditionally executes the request on the *Managed Objects (MO)*.

All access control management happens at the OSS level through a central PAP, whereas the nodes only need to be able to verify the signatures of the OSS clients in order to verify the integrity of the transferred commands. Replay protection must also be provided. Thus the OSS clients have root rights on the nodes. This architecture is illustrated in figure 2

New nodes are added to the system by configuring them with the public keys of the OSS-clients and registering the nodes with the OSS, so that they are known by the clients.

If a user accesses a node locally, this architecture would require the user to get his request authorised beforehand by the OSS and fetch the signed request. During the out-of-band access this request is then forwarded to the node's agent, which checks the signature of the OSS. We call this out-of-band variant of the

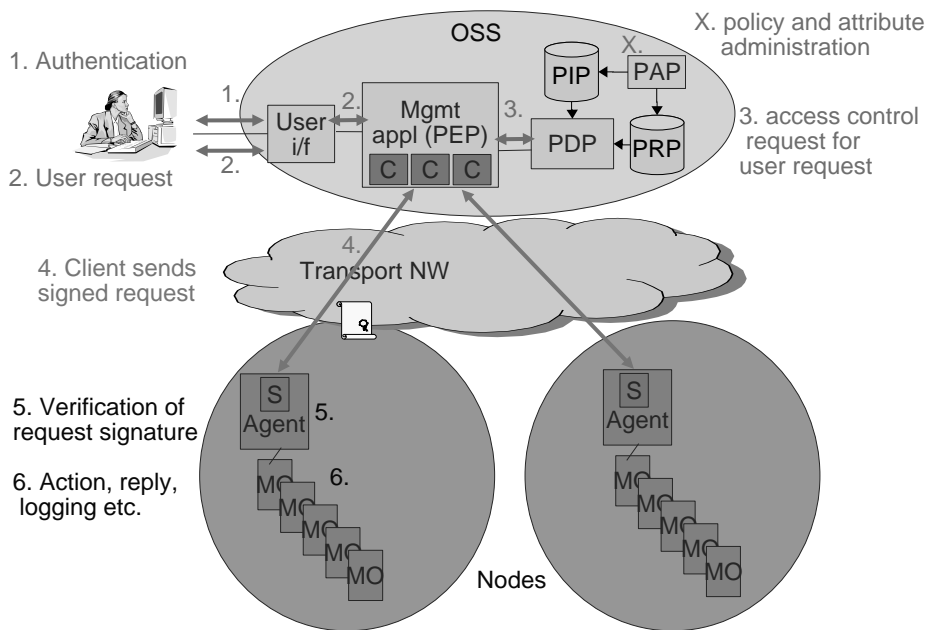


Figure 2: Centralised access control architecture

centralised authorisation enforcement the *authorisation push* scenario. Figure 3 shows this architecture.

The advantages of this architecture can be summarised as follows:

- The access control code on the agent side is simple.
- Inconsistencies in the access control policies will be easier to detect and remedy due to the centralised nature of the PRP.
- It will be easy to audit policies and other relevant access control information.
- The administration of policies is simple since they are all at the same place.

On the downside we can note the following:

- If the OSS breaks down, access control becomes unavailable, thus preventing management of the system.
- Since all access control happens at the OSS, it may become a bottleneck and operations may slow down.
- Since the OSS PRP will contain all policies, a lot of policies will get browsed for every request. Even though efficient filtering may alleviate this problem, it still requires some additional processing.
- In the case of out-of-band access the user needs to have authorised the request at the OSS beforehand.

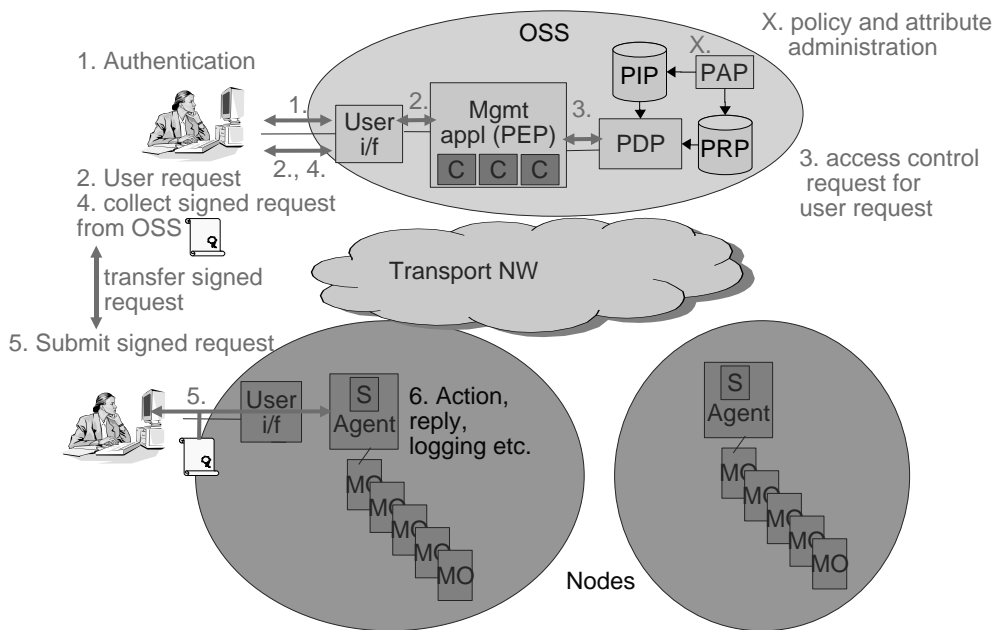


Figure 3: Out-of-band access for the centralised architecture using the authorisation push mechanism

An neutral observation is that message integrity between OSS-client and node-agent are the weakest link in the security of this architecture.

If the OSS is unavailable to issue signed requests, no access to the nodes is possible. This may be considered unacceptable and therefore a fall-back solution might be considered. However such a solution will always introduce a back-door into the system, which can then be abused to gain unauthorised access.

3.2 Decentralised authorisation enforcement

In this architecture all services providing authorisation data, decisions and enforcement (PRP, PIP, PDP and PEP) are distributed on the nodes. The OSS only works as Policy Administration Point by providing tools to manage policies (inspection, update, revocation/deletion) on multiple nodes simultaneously. The OSS also provides a mirror of the PRPs and PIPs on the nodes, so that a central overview of valid policies is available. Having a central storage for parameters enables automatic checking for trivial configuration errors before commands are executed.

Besides policy administration, the OSS is only used to forward user requests to the node agents. The role of the OSS can be somewhat extended by having it provide facilitating services or tools to do this.

The agent on the node takes the role of a PEP and uses a PDP on the node to check incoming requests. The node saves policies and other relevant access control information in its internal data store. Figure 4 illustrates this architecture.

New nodes are added to the system by configuring them with relevant access

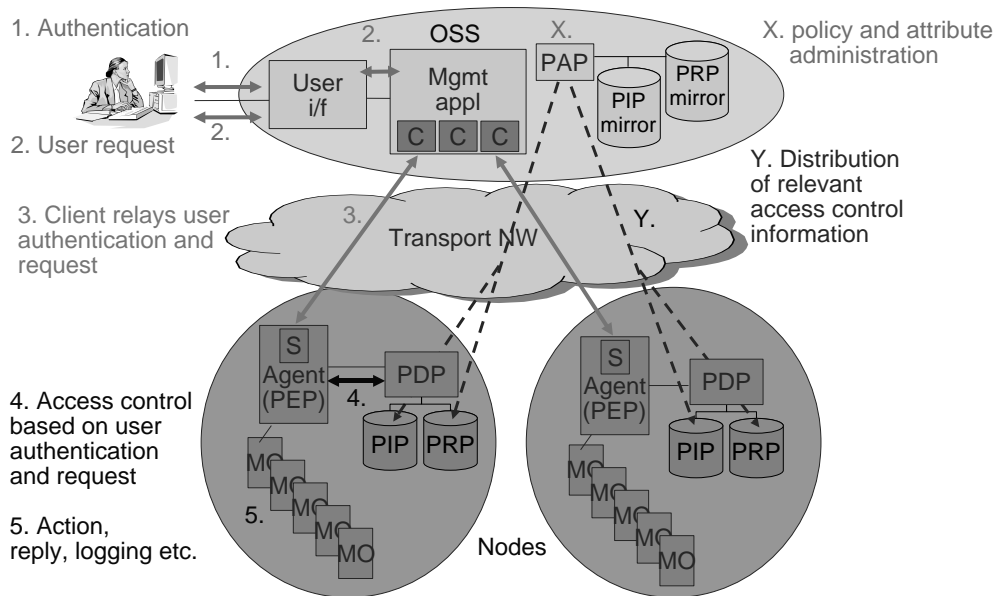


Figure 4: Decentralised access control architecture

control information from the OSS-mirror and registering the nodes with the OSS-PAP, so that they get the notified of access control information updates.

A user connecting to a node locally without passing through the OSS, is treated the same way as a user passing his request through the network, using the OSS, as far as access control is concerned. We call this out-of-band variant of the decentralised authorisation enforcement the *internal authorisation pull* scenario. Figure 5 shows this architecture.

This architecture has the following advantages:

- Failures of the OSS do not influence access control functions
- When access control is performed, no authorisation data needs to be transferred over the network, therefore more bandwidth is available for production data.
- Nodes have greater autonomy in cases of network failures, since access control can be performed independently.

The drawbacks of this architecture are:

- Global updates of authorisation information (i.e. for all or a large group of nodes) require a reliable broadcast mechanism.
- In order to avoid inconsistent policies, the OSS needs to have a mechanism to handle unreachable nodes during a global update of authorisation information.
- The code on the node agents will be more complex (i.e. require more computational resources, disk and memory space).

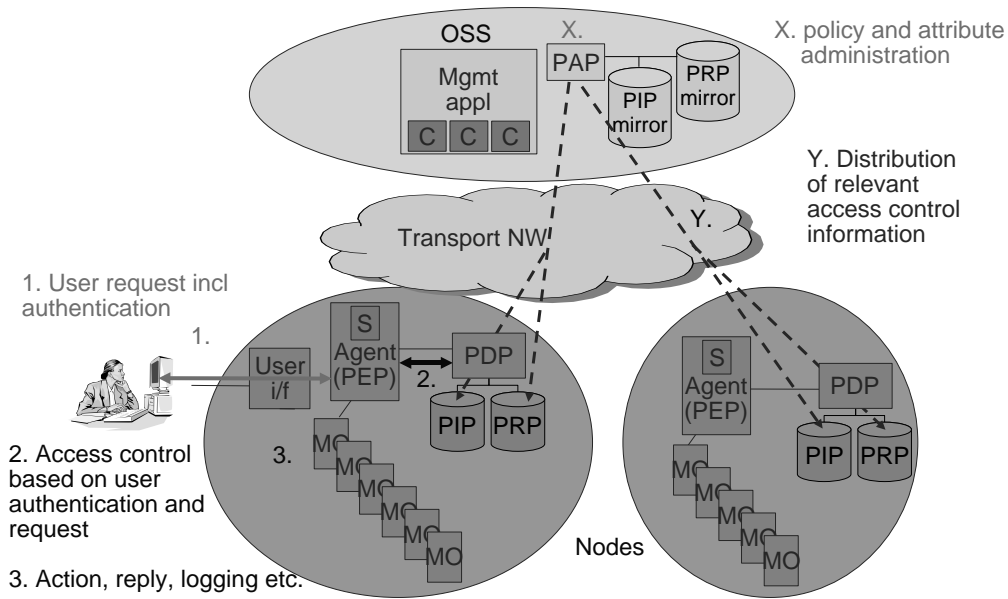


Figure 5: Out-of-band access for the decentralised architecture using the internal authorisation pull mechanism

- The nodes will need to have some reliable way to measure time in order to prevent backdating of operations and replay attacks.

A decision that has to be made in this architecture is whether a PAP at the node level for locally updating authorisation information (in the PIP and PRP) should be provided as well. This would further reduce the impact of failures in the OSS, but it would aggravate the problem of possible inconsistencies in the access control information.

3.3 Hybrid authorisation enforcement

This architecture divides up the authorisation data between the OSS and the nodes. The idea is to have frequently updated data centrally available and relatively static data distributed on the nodes. We thereby win some autonomy in case the OSS breaks down, while still having the advantages of centralised access control. In order to achieve this we set up a PIP within the OSS that provides information relevant to access control (e.g. user roles, resource groups, access profiles).

The nodes feature a PDP that does simple access control based on the attributes provided by the OSS. The policies stored on the node's PRP are designed to be stable enough not to require frequent updates. This architecture is shown in figure 6.

New nodes are added to the system by configuring them with relevant policies and registering the nodes with the OSS, so that they are known by the clients.

A user wanting to connect to a node locally without passing through the OSS, needs to acquire assertions describing his attributes first. These are then passed to the node agent, using the same protocol as the OSS client would.

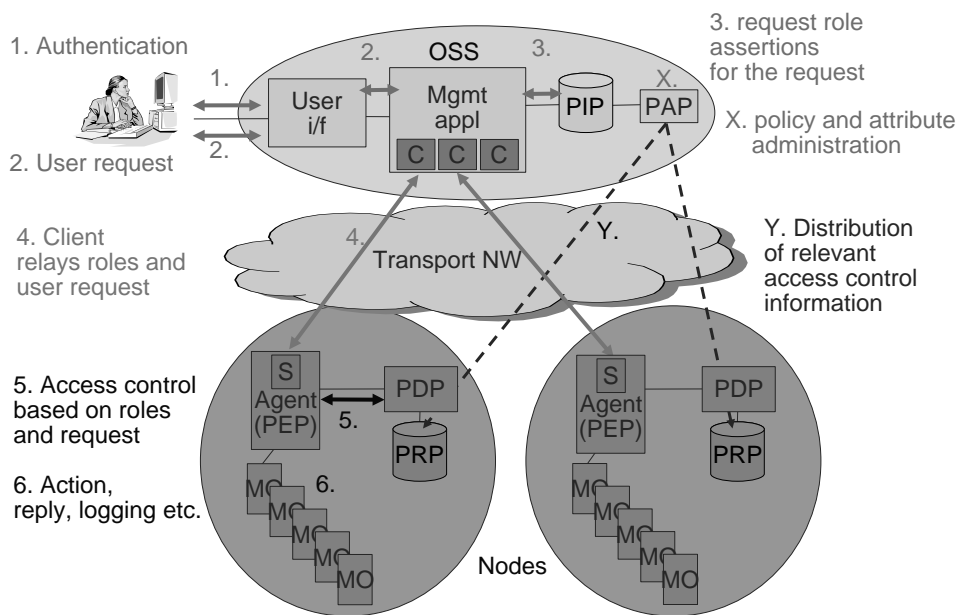


Figure 6: Hybrid access control architecture

The node then proceeds to perform access control as it would normally. We call this out-of-band variant of the hybrid authorisation enforcement architecture the *mixed authorisation pull* scenario, since information is pulled both locally from the node and externally from the user. Figure 7 shows this architecture.

We find the following advantages to this architecture:

- The impact of failures of the OSS can be reduced by issuing long lived attribute assertions.
- Frequently updated authorisation information is centrally available at the OSS and is therefore more easy to manage, debug and audit. In a role based access control scenario this would mean that the user-to-role assignments can be very dynamic.
- The access control code on the agent side relatively simple.
- In case of out-of-band access, full access control is still performed.

This architecture has the following drawbacks:

- Policies on the nodes need to be carefully designed to avoid the need for frequent updates. In case of role based access control this means that the role-to-permission assignment should be very static.
- The cost of creating/modifying policies based on new attributes is relatively high.
- If the lifetime of attribute assertions is long, we might need an assertion revocation mechanism. However the longer the assertion lifetime, the lower the impact of OSS failures. We call this the *assertion lifetime dilemma*.

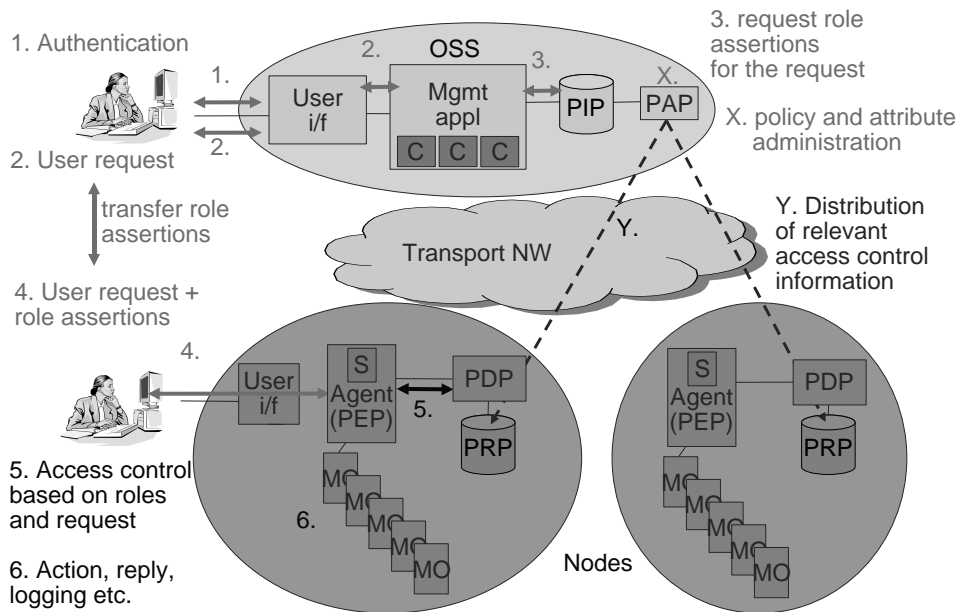


Figure 7: Out-of-band access for the hybrid architecture using the mixed authentication pull mechanism

4 Discussion

Having presented several architecture alternatives, we now proceed to discuss them in the light of the specific requirements in section 2.3.

High reliability:

The *centralised architecture* requires the OSS to be up and running for each access to a node, although some delay is possible between the access to the OSS and the access to the node, depending on how long a request assertion may be used. Introducing a fall-back access solution that does not require a request assertion increases the availability of the node in case of OSS failures, it does however also reduce the security of the system, since end-user privileges are not enforced for such accesses. Finally the OSS must also be available for all policy and attribute administration tasks, although those would be more rare, than common access requests.

In the *decentralised architecture* the OSS needs only to be available for policy and attribute administration tasks. Therefore full access control on the nodes is very reliably provided. The dependence on the OSS for administrative tasks can be further reduced by providing local administration tools. These do however increase the risk of inconsistent policies and should only be used in emergencies.

The *hybrid architecture* requires the OSS to be up and running for attribute assertion retrieval and for administration of attributes and policies. This dependence can be considerably reduced by increasing the lifetime of attribute assertions. Given those assertions, the access control is very reliable, since everything else happens at the node level.

Interdependencies w. external services:

All architectures require a functioning authentication system for users and services that access protected resources on the nodes. Furthermore all architectures require some kind of communication security protocol to ensure the integrity and freshness of messages exchanged over the network.

In addition to this, the nodes in the *centralised* and *hybrid architecture* require some method of signature and expiry checking for assertions (requests in the centralised architecture and attribute/role assertions in the hybrid architecture). Depending on assertion lifetime, the *hybrid architecture* may also require an assertion revocation mechanism to be deployed.

Central and local administration possible:

The *centralised architecture* does not allow for administration of access control at node level. The only thing that can be managed locally is the trust relation to the OSS clients.

When using the *decentralised architecture*, central and local administration are possible. Keeping mirrors of the access control information on the nodes at the OSS level increases the efficiency of central administration.

The *hybrid architecture* allows user attributes to be managed centrally, whereas policies can be managed both centrally and locally.

Consistent data about access control state:

The *centralised architecture* stores all access control data centrally. Therefore a consistent, up to date state information is always available.

In the *decentralised architecture* problems might arise, if access control data updates do not reach intended recipients. Therefore it is advisable to implement a database-like transaction system for access control data administration.

The same holds true for the *hybrid architecture*, although to a lesser extent, since only the policy data is distributed and we expect policy updates to happen more rarely than in the other architectures.

Low bandwidth use:

When accessing nodes over the network using the *centralised architecture*, signing requests only leads to a negligible increase of the bandwidth use. All attribute and policy administration does not use any bandwidth at all, since it is done locally at the OSS level.

The *decentralised architecture* does not increase the bandwidth use of transferring requests over the network. However all attribute and policy administration that is done at the OSS level requires some bandwidth to send the results to the affected nodes. Consequently changes affecting many nodes will use a considerable amount of bandwidth.

In the *hybrid architecture*, transferring the attribute assertions when accessing nodes over the network requires some additional bandwidth, as opposed to only transferring the request. This gives a strong incentive to use only the minimal set of attributes required to perform a given task.⁴ Attribute administration does not consume bandwidth, since it is done locally at the level of the OSS. However policy administration will consume bandwidth, as the results need to

⁴In access control this generally recognised as a very desirable property and known as the *least privilege principle* [6]

be transmitted to the nodes. A careful design of policies and attributes, that allows policies to remain relatively stable and uses attributes to handle dynamics helps to reduce this kind of bandwidth use.

End user privileges enforced in the Nodes:

The *centralised architecture* does not allow to enforce full end user privileges in the node, instead these privileges are enforced at the OSS level, by signing the user's request.

In the *decentralised architecture* does all privilege enforcement at the node level, including even administrative privileges.

In the *hybrid architecture*, end user privileges are enforced at the node level. However attributes necessary for using these privileges are asserted at the OSS level.

Simplicity:

The *centralised architecture* is simple to set up at large scale, since the required modifications at the node level are comparatively small. Most complexity lies at the OSS level, which is comparatively easy to change and configure. Administration of access control information is also relatively easy, since it is available centrally and therefore no complex distribution mechanisms are needed.

A *decentralised architecture* is difficult to set up at large scale, since big modifications of the code at the node level are required. All nodes need the full access control infrastructure. Administration of access control information is complex, since distribution mechanisms are required to broadcast updates to concerned nodes.

When using the *hybrid architecture* the difficulty of setting up both nodes and the OSS is moderate, since both need some infrastructure. Administration and updating of user attributes is simple, whereas policies are difficult both to administrate and update due to their distributed nature.

5 Conclusion and next steps

We have presented different architecture alternatives starting from the natural approaches of centralised versus decentralised designs. We have presented some telecom specific criteria and made a comparison and analysis of the security properties of these architectures, with respect to those criteria. The aim of this work is not to give a single recommendation for every possible telecom application, as such an approach would certainly prove to be inflexible. Instead we want to display the trade-offs between the different architectures and give a set of parameters that can be used to make a business decision on which architecture to use. In order to reach such a decisions, the advantages and disadvantages of the different architectures have to be weighted depending on the specific system and business requirements.

In the next steps of our work, we plan to implement a simulation of these different architectures in order to run stress tests and find unexpected effects that can influence the architecture decisions. We plan to build our access control solution on the OASIS standard XACML[1], especially on the upcoming version 3.0 of the standard [5], since it provides administrative policies, an important feature for decentralised access control. In order to encode attribute

assertions we plan to use the OASIS standard SAML[3], since it integrates easily with XACML. We have developed an open source implementation of a PDP for the XACML 3.0 draft⁵ and an assertion server⁶ for SAML based attribute administration.

In order to have a realistic test scenario we plan to explore XACML based access control for remote configuration of nodes using the IETF Netconf [4] protocol. This work will be covered in a future report.

References

- [1] S. Godik, T. Moses, and Eds. eXtensible Access Control Markup Language (XACML). Standard, Organization for the Advancement of Structured Information Standards (OASIS), February 2003. <http://www.oasis-open.org/committees/xacml>.
- [2] M. Kaeo. Operational Security Current Practices. Internet-Draft draft-ietf-opsec-current-practices-07, Internet Engineering Task Force (IETF), August 2006. <http://www.ietf.org/internet-drafts/draft-ietf-opsec-current-practices-07.txt>.
- [3] E. Maler, P. Mishra, R. Philpott, and Eds. The OASIS Security Assertion Markup Language (SAML) v1.1. Standard, Organization for the Advancement of Structured Information Standards (OASIS), September 2003. <http://www.oasis-open.org>.
- [4] Network Configuration Working Group. Network Configuration (netconf). Technical report, Internet Engineering Task Force (IETF), 2003. <http://www.ietf.org/html.charters/netconf-charter.html>.
- [5] E. Rissanen, H. Lockhart, T. Moses, and Eds. XACML v3.0 administrative policy. Standard, Organization for the Advancement of Structured Information Standards (OASIS), June 2006. <http://www.oasis-open.org/committees/xacml>.
- [6] J. Saltzer and M. Schroeder. The protection of information in computer systems. In *Proceedings of IEEE*, pages 1278–1308. IEEE Computer Society, September 1975.
- [7] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. AAA Authorization Framework. Request For Comments (RFC) 2904, Internet Engineering Task Force (IETF), August 2000. <http://www.ietf.org/rfc/rfc2904.txt>.

⁵<http://www.sics.se/spot/xacml.3.0.html>

⁶http://www.sics.se/spot/assertion_server.html