

The role of directories in Single Sign on Systems

Victoriano Giralt

Central Computing Facility
University of Malaga

TERENA EuroCAMP
Ljubljana
April 3rd 2006



The Dark Ages
The Enlightenment
The Industrial Revolution
The XXI century
Summary

Outline

1 The Dark Ages



Outline

- 1 The Dark Ages
- 2 The Enlightenment



Outline

- 1 The Dark Ages
- 2 The Enlightenment
- 3 The Industrial Revolution



Outline

- 1 The Dark Ages
- 2 The Enlightenment
- 3 The Industrial Revolution
- 4 The XXI century



The Dark Ages of Authentication

there were no directories

- There is no central credential repository



The Dark Ages of Authentication

there were no directories

- There is no central credential repository
- Each and every application has its own credential repository



The Dark Ages of Authentication

there were no directories

- There is no central credential repository
- Each and every application has its own credential repository
- Users are in the midst of their worst nightmare



The Enlightenment of Authentication

there IS a directory

- Directories appear



The Enlightenment of Authentication

there IS a directory

- Directories appear
- We have a *centralised* credential repository



The Enlightenment of Authentication

there IS a directory

- Directories appear
- We have a *centralised* credential repository
- We don't really know what to do with it



The Enlightenment of Authentication

there IS a directory

- Directories appear
- We have a *centralised* credential repository
- We don't really know what to do with it
- Every application does its own authentication



The Enlightenment of Authentication

there IS a directory

- Directories appear
- We have a *centralised* credential repository
- We don't really know what to do with it
- Every application does its own authentication
- Fortunately, users only have to remember one set of credentials



The Revolution in Authentication

Single Sign On

- The directory disappears into the back stage



The Revolution in Authentication

Single Sign On

- The directory disappears into the back stage
- Kerberos can use the directory for AuthN



The Revolution in Authentication

Single Sign On

- The directory disappears into the back stage
- Kerberos can use the directory for AuthN
- Web Single Sign On systems also use the directory for AuthN



The Revolution in Authentication

Single Sign On

- The directory disappears into the back stage
- Kerberos can use the directory for AuthN
- Web Single Sign On systems also use the directory for AuthN
- Polite applications know how to do AuthN against a directory



The Revolution in Authentication

Single Sign On

- The directory disappears into the back stage
- Kerberos can use the directory for AuthN
- Web Single Sign On systems also use the directory for AuthN
- Polite applications know how to do AuthN against a directory
- There are some applications left that have *an attitude* and we must find a way to provision them



XXI century directories do AuthZ

Storing privileges in the directory

- The directory is used as an unique point for AuthoriZation



XXI century directories do AuthZ

Storing privileges in the directory

- The directory is used as an unique point for AuthoriZation
- A sole authorization model



XXI century directories do AuthZ

Storing privileges in the directory

- The directory is used as an unique point for AuthoriZation
- A sole authorization model
- Agent-Function-Qualifier



Getting AuthZ to applications out of the directory

- Direct directory search



Getting AuthZ to applications out of the directory

- Direct directory search
- Web services



Getting AuthZ to applications out of the directory

- Direct directory search
- Web services
- Authorization assertions for Web SSO systems



Getting AuthZ to applications out of the directory

- Direct directory search
- Web services
- Authorization assertions for Web SSO systems
- Provisioning for applications with *an attitude*



Summary

- No one is using multiple credentials anymore



Summary

- No one is using multiple credentials anymore, true?



Summary

- No one is using multiple credentials anymore, true?
- A single set of credentials



Summary

- No one is using multiple credentials anymore, true?
- A single set of credentials
- Central AuthN/AuthZ management



Summary

- No one is using multiple credentials anymore, true?
- A single set of credentials
- Central AuthN/AuthZ management
- Fast provision



Summary

- No one is using multiple credentials anymore, true?
- A single set of credentials
- Central AuthN/AuthZ management
- Fast provision and deprovision



Introduction to technical matters

How we can achieve XXI century directory based AuthN/Z

We have different options depending on



Introduction to technical matters

How we can achieve XXI century directory based AuthN/Z

We have different options depending on

- the kind of applications we want to integrate



Introduction to technical matters

How we can achieve XXI century directory based AuthN/Z

We have different options depending on

- the kind of applications we want to integrate
- the kind of infrastructure we are using



Introduction to technical matters

How we can achieve XXI century directory based AuthN/Z

We have different options depending on

- the kind of applications we want to integrate
- the kind of infrastructure we are using
- the desired level of interoperability



Introduction to technical matters

How we can achieve XXI century directory based AuthN/Z

We have different options depending on

- the kind of applications we want to integrate
- the kind of infrastructure we are using
- the desired level of interoperability

Web applications, easier to integrate into the SSO picture



Introduction to technical matters

How we can achieve XXI century directory based AuthN/Z

We have different options depending on

- the kind of applications we want to integrate
- the kind of infrastructure we are using
- the desired level of interoperability

Web applications, easier to integrate into the SSO picture

Traditional applications are a much different issue



Active Directory Server

The MS way of things

- Designed for a Microsoft centric environment



Active Directory Server

The MS way of things

- Designed for a Microsoft centric environment
- Works for web and non web applications



Active Directory Server

The MS way of things

- Designed for a Microsoft centric environment
- Works for web and non web applications as long as they are on MS-Windows



Active Directory Server

The MS way of things

- Designed for a Microsoft centric environment
- Works for web and non web applications as long as they are on MS-Windows
- Not much interoperable



Active Directory Server

The MS way of things

- Designed for a Microsoft centric environment
- Works for web and non web applications as long as they are on MS-Windows
- Not much interoperable
- Can do AuthN to another LDAP using ADAM



Active Directory Server

The MS way of things

- Designed for a Microsoft centric environment
- Works for web and non web applications as long as they are on MS-Windows
- Not much interoperable
- Can do AuthN to another LDAP using ADAM
- Can be tamed with the help of Kerberos



Active Directory *Federation* Services

The *F* word comes into play

- Designed for interoperating with non Microsoft environments



Active Directory *Federation* Services

The *F* word comes into play

- Designed for interoperating with non Microsoft environments
- Works with LDAP for AuthN thanks to ADAM



Active Directory *Federation* Services

The *F* word comes into play

- Designed for interoperating with non Microsoft environments
- Works with LDAP for AuthN thanks to ADAM
- Uses SAML for federating



Active Directory *Federation* Services

The *F* word comes into play

- Designed for interoperating with non Microsoft environments
- Works with LDAP for AuthN thanks to ADAM
- Uses SAML for federating
- Only for Web SSO



Active Directory *Federation* Services

The *F* word comes into play

- Designed for interoperating with non Microsoft environments
- Works with LDAP for AuthN thanks to ADAM
- Uses SAML for federating
- Only for Web SSO
- Available on Windows 2003 Server R2



Oracle SSO

You play by our rules

- Centred around iAS and OID



Oracle SSO

You play by our rules

- Centred around iAS and OID
- Mainly Web SSO



Oracle SSO

You play by our rules

- Centred around iAS and OID
- Mainly Web SSO
- It is not clear it can integrate non web apps



Oracle SSO

You play by our rules

- Centred around iAS and OID
- Mainly Web SSO
- It is not clear it can integrate non web apps
- Can integrate external web apps via Apache module (mod_osso) or SDK



Oracle SSO

You play by our rules

- Centred around iAS and OID
- Mainly Web SSO
- It is not clear it can integrate non web apps
- Can integrate external web apps via Apache module (mod_osso) or SDK
- Has an API for interoperation but needs user synchronisation



PERMIS

X.509 based AuthZ policies

Permis is an AAI with a Privilege Management Infrastructure



PERMIS

X.509 based AuthZ policies

Permis is an AAI with a Privilege Management Infrastructure

- Based on X.509 Attribute Certificates



PERMIS

X.509 based AuthZ policies

Permis is an AAI with a Privilege Management Infrastructure

- Based on X.509 Attribute Certificates
- The certificates do a strong bind between holder and granted privilege



PERMIS

X.509 based AuthZ policies

Permis is an AAI with a Privilege Management Infrastructure

- Based on X.509 Attribute Certificates
- The certificates do a strong bind between holder and granted privilege
- Privileges granted can range from University degrees through file access



PERMIS

X.509 based AuthZ policies

Permis is an AAI with a Privilege Management Infrastructure

- Based on X.509 Attribute Certificates
- The certificates do a strong bind between holder and granted privilege
- Privileges granted can range from University degrees through file access
- Attribute certificates are stored in the holder's entry in the directory



PERMIS

X.509 based AuthZ policies

Permis is an AAI with a Privilege Management Infrastructure

- Based on X.509 Attribute Certificates
- The certificates do a strong bind between holder and granted privilege
- Privileges granted can range from University degrees through file access
- Attribute certificates are stored in the holder's entry in the directory
- Digitally signed policies are also stored in the owner's entry



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- **PMI**

Privilege Management Infrastructure

Strong authorization infrastructure that extends X.509 PKIs.

It is based upon the same cryptographic principles.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- PMI
- AC

Attribute Certificate

Strong binding of owner and attribute, based on digital signatures.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- PMI
- AC
- AA

Attribute Authority

The entity that grants the privileges by issuing the Attribute Certificate to the holder.

E.g: a University, the owner of a file or a manager.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- PMI
- AC
- AA
- **Owner**

Certificate owner

The entity to which the privileges have been granted.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- PMI
- AC
- AA
- Owner
- **Attributes**

Attributes

Part of the certificate that is signed, like the public key in a PKI.

Can be used to store privileges and policies.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- **SOA**

Source Of Authority

Equivalent to a PKI's Root CA. It is the root of trust. A resource access control system implicitly trusts the SOA for granting access rights and privileges to it.

The SOA issues ACs to AAs and end users.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- SOA
- PDP

Policy Decision Point

Entity where AuthZ policies are stored, in a signed AC, and AuthZ decisions are taken based on such policies.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- SOA
- PDP
- **PEP**

Policy Enforcement Point

The entity that protects access to a resource and acts based on queries to the PDP.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- SOA
- PDP
- PEP
- **DIS**

Delegation Issuing Service

An entity used by the AAs to issue delegation ACs.

It allows for better control, auditing and logging of the delegation of privileges.

It can also reduce complexity of the privilege issuing chain.



PERMIS

X.509 based AuthZ policies

Some PERMIS key terms

- SOA
- PDP
- PEP
- DIS
- **Non
assertion**

Preventing privilege abuse

The owner of an AC marked as *no assertion* can grant the indicated privileges but cannot use them. Useful for the DIS.



Kerberos

taming the beasts

Kerberos, though old, will take us into the XXI century of SSO.



Kerberos

taming the beasts

Kerberos, though old, will take us into the XXI century of SSO.

- It can use the directory for AuthN



Kerberos

taming the beasts

Kerberos, though old, will take us into the XXI century of SSO.

- It can use the directory for AuthN
- It can control access to many kinds of applications



Kerberos

taming the beasts

Kerberos, though old, will take us into the XXI century of SSO.

- It can use the directory for AuthN
- It can control access to many kinds of applications
- ADS can use Kerberos for AuthN



Kerberos

taming the beasts

Kerberos, though old, will take us into the XXI century of SSO.

- It can use the directory for AuthN
- It can control access to many kinds of applications
- ADS can use Kerberos for AuthN
- There are rumours of someone having used it for AuthN with OID



Kerberos

taming the beasts

Kerberos, though old, will take us into the XXI century of SSO.

- It can use the directory for AuthN
- It can control access to many kinds of applications
- ADS can use Kerberos for AuthN
- There are rumours of someone having used it for AuthN with OID
- Unfortunately it cannot be used for AuthN to web apps from non Windows clients



eduPermission and eduPermissionGroup

a work in progress

It is a discussion in progress in MACE about ways of storing permissions in the directory.



eduPermission and eduPermissionGroup

a work in progress

It is a discussion in progress in MACE about ways of storing permissions in the directory.

- **Subentries**

eduPermission: as objects subentries (Tom Barton's)

The permissions objects are stored as subentries of the holder's entry.

It might have scaling problems if holders are persons, as the numbers may explode.

eduPermission and eduPermissionGroup

a work in progress

It is a discussion in progress in MACE about ways of storing permissions in the directory.

- Subentries
- **Groups**

eduPermissionGroup (Brendan Belina's)

Permissions are described as group entries in the directory and are granted to persons by way of inclusion in the group, using standard membership mechanisms.



eduPermission and eduPermissionGroup

a work in progress

It is a discussion in progress in MACE about ways of storing permissions in the directory.

- Subentries
- Groups
- **Objects**

Permissions are objects

Both approaches share the way they describe permissions, as objects with multiple attributes for storing their properties such as the application to which they are applied.



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

Assigns access rights to the designated application:



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

Assigns access rights to the designated application:

- **Function**

entitlement

the URN describes a right for a user or role



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:**appAccess**:SolicitudGasto:*LEVEL*

Assigns access rights to the designated application:

- **Function**

appAccess

kind of right, access to an application in this case.



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:**SolicitudGasto**:*LEVEL*

Assigns access rights to the designated application:

- **Function**

SolicitudGasto

application the right is granted on.



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:**LEVEL**

Assigns access rights to the designated application:

- **Function**

LEVEL

granted access level, application specific:
RUG, ROU, RGE



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage

LDAP search

The application does a standard directory search to find out if the user that has been authenticated has the right to use it and the access level that has been granted to her.

URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage

Query via web service

The application queries a web service with user and application identifier as inputs and obtains the access level or the absence of the right to use.



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage

WebSSO AuthZ assertion

The authentication server has information about the accessed resource, once the user is AuthN'd, retrieves application specific AuthZ information from the entitlements in the user's entry in the directory, and passes them onto the resource



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage
- **Advantages**

Unique authorization point

All of an object's authorisations, both explicit and implicit, are centrally kept in a directory entry.



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage
- **Advantages**

A sole authorization model

URNs allow us to express all authorization in a common form, with application specific semantics.



URNs in Entitlements for AuthZ

as it is in use at UMA (by example)

irisUserEntitlement = urn:mace:rediris.es:uma.es:
entitlement:applAccess:SolicitudGasto:*LEVEL*

- Function
- Usage
- **Advantages**

Agent-Function-Qualifier

Who can do What on Which object



SPOCP

multiple source AuthZ policies

With regard to AuthZ, we can consider SPOCP as



SPOCP

multiple source AuthZ policies

With regard to AuthZ, we can consider SPOCP as

- Engine

AuthZ policy engine

AuthZ policies can be described and applied to resources using SPOCP



SPOCP

multiple source AuthZ policies

With regard to AuthZ, we can consider SPOCP as

- Engine
- **Service**

AuthZ policy service

SPOCP is implemented as a service that resources query for taking AuthZ decisions based on the policy engine.



SPOCP

multiple source AuthZ policies

With regard to AuthZ, we can consider SPOCP as

- Engine
- Service
- **Aggregator**

AuthZ source aggregator

We can use SPOCP for aggregating information on which AuthZ decisions can be based, through the use of boundary conditions.



SPOCP

multiple source AuthZ policies

With regard to AuthZ, we can consider SPOCP as

- Engine
- Service
- Aggregator

Most important, it allows us to use most of the presented methods of AuthZ, and then some.



SPOCP

multiple source AuthZ policies

Some SPOCP key terms



SPOCP

multiple source AuthZ policies

Some SPOCP key terms

- **S-Expression**

Policy language

The access policies to resources are described using expressions like:
(spocp (resource etc passwd)(action write)(subject (uid 0)))



SPOCP

multiple source AuthZ policies

Some SPOCP key terms

- S-Expression
- **Policy engine**

the Less-Permissive function

By applying this function, the engine guarantees that the querying party will receive a formally correct answer, thus assuring the right AuthZ decision.



SPOCP

multiple source AuthZ policies

Some SPOCP key terms

- S-Expression
- Policy engine
- **AuthZ server**

Answer to resource AuthZ queries

SPOCP can be implemented as a server that listens on a socket and resolves application queries for AuthZ decisions.



SPOCP

multiple source AuthZ policies

Some SPOCP key terms

- S-Expression
- Policy engine
- AuthZ server
- **Boundary condition**

Attributes from many sources

SPOCP can use different sources for getting attributes on which to base the AuthZ decision. One of this sources is the directory, as well as relational databases, network information or whatever anyone wants to program for.

SPOCP

multiple source AuthZ policies

Some SPOCP key terms

- S-Expression
- Policy engine
- AuthZ server
- Boundary condition
- **Plug-in**

The way to reach the attribute sources

This is the basis for boundary conditions. It is based on a modular approach like Apache's.

Any needed boundary condition can be implemented in a module that will be loaded at runtime.



A glimpse of the future

A passwordless world

- InfoCard



A glimpse of the future

A passwordless world

- InfoCard
- Higgins, the OpenSource response

