



Red IRIS

PAPI 1.2

A usage-driven release

1st TERENA AA Workshop
Stockholm, November 2002

Goals

- Move towards interoperability
 - Attribute-based authorization decisions
- Incorporate new functionality
 - Portability, configuration,...
- Enhance privacy preservation
 - Control on attribute release
- Taking advantage of the growing user community
 - Reviews and contributions
- Keeping in mind the current user community
 - Seamless evolution
 - Requests on more user-centric features
 - Personalization
 - Proxy mode

PAPI 1.2 and PAPI 2

- Parallel developments
- Coverage of current user requests
- Re-design to fulfill
 - Interoperability
 - Better architecture scalability
 - Portability
 - More flexible trust model
- Both branches influence each other
 - Make current users seamlessly converge to new versions
 - Incorporate new user communities

PAPI 1.2 - An overview

- Still based in Perl
 - And Perl-ish configuration and features
- Support for attribute-based authorization
 - Assertions sent by the AS can be individualized
 - PoAs can specify richer authz filters on these assertions
- Better personalization mechanisms
 - Individual accept/reject objects
 - Automatic redirection at the AS
- Extended proxy mode
 - Applicable to a whole domain
 - Support for HTTP authentication

Support for attribute-based authz

- It was perceived that authorization was performed at the AS, not the PoA
 - Even when GPoAs are in use and PoAs can define specific filters
- An AS always sent the same information about a user to all (G)PoAs
 - This made difficult attribute-based local authz
 - Too many data for one, too few for other
 - And raised concerns on privacy preservation
 - Applicable to very uniform user communities
 - And flat set of services

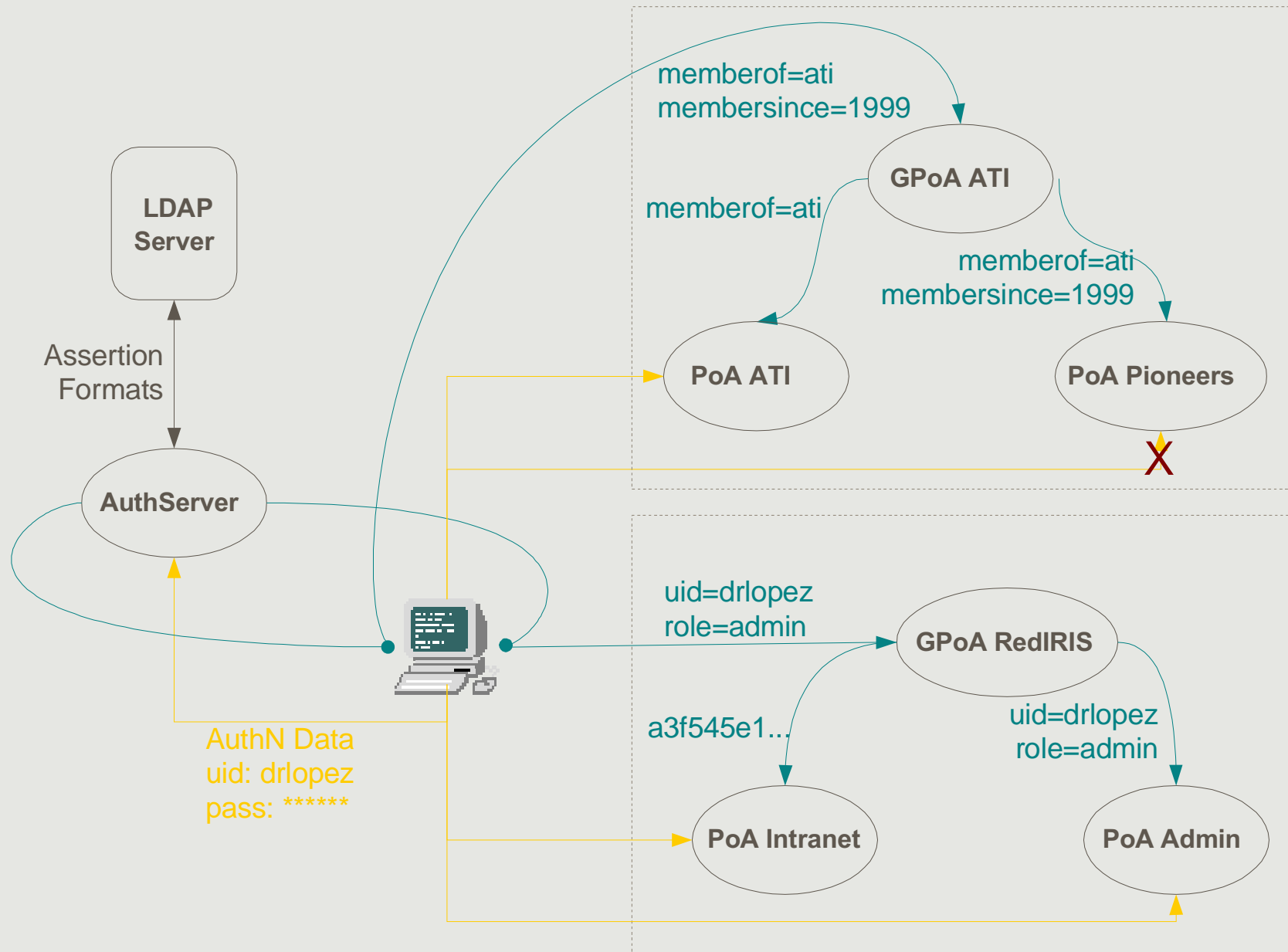
Support for attribute-based authz

- For each (G)PoA an AS is going to contact an assertion format string is derived from:
 - User and group data
 - The (G)PoA definition
 - The AS defaults
- Inside the assertion format string, the AS can substitute
 - Connection variables
 - Username (or a hash of it), a nonce, anything else passed through the HTML forms or the configuration
 - Attributes of the user entry
 - Based on LDAP although other sources are possible
- A Perl-ish way to ARPs

Support for attribute-based authz

- When a (G)PoA receives a request for tokens can apply filters
 - Even (and specially) when it comes from a parent GPoA
 - `PAPI_Filter RegExp => [accept|reject]`
- A GPoA is the only element receiving a priori data about the user
 - It can control which data is passed down the hierarchy
 - `Hash_User_Data [0|1]` (released)
 - `GPoA_Rewrite PoARegExp AssertionRegExp => RewriteExp` (under test)
 - Some experiments on using the AS as a top-level GPoA
- Immediate filters can be applied on token values through `Cookie_Reject`

Support for attribute-based authz



Better personalization

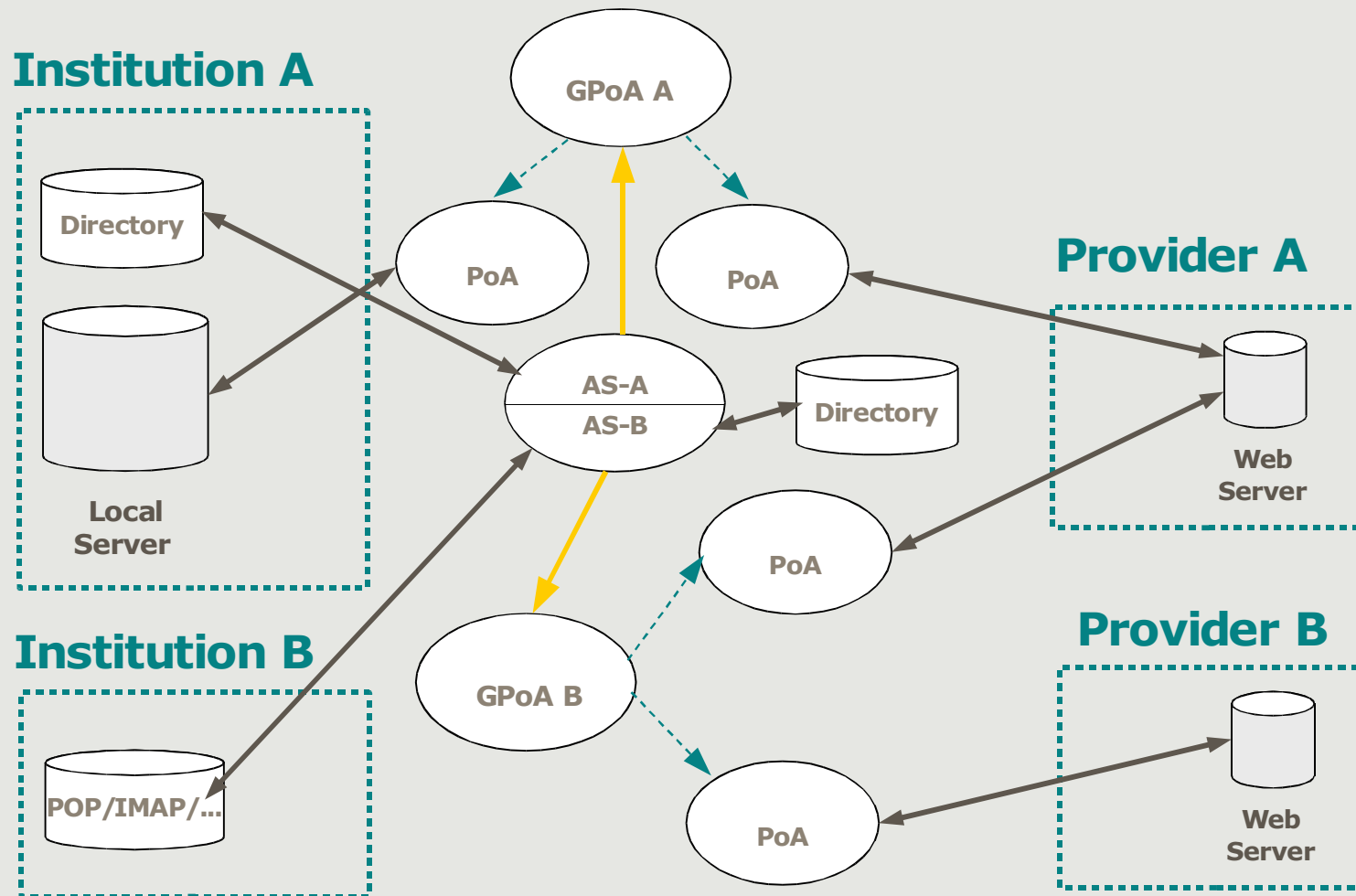
- One of the more recurrent requests
 - We are dealing with *digital IDs*
 - Portals offering access to internal and external services
 - Compatible with privacy
- New personalization features
 - The objects to be requested from PoAs at the initial connection can be defined on a per-site basis at the AS
 - Many possibilities to be explored (JavaScript, CSS, ...)
 - Assertion formats can be used to convey data to special on-site applications, using `Hcook_Handler`
 - The AS is able to work with the referring URL
- Much has to be experimented and documented
 - New scenarios, bigger and wider communities,...

Enhanced proxy functionality

- It is a common-sense answer to a very common problem dealing with eggs and chickens:
 - No providers are willing to invest in a new and not widespread technology
 - No adpoters if there is no support from providers
- Proxy enhancements are motivated by experiences with real-life situations
 - Support for HTTP Authentication (Basic & Digest)
 - Plan to use `HttpCookie` contents => personalization
 - Experiments with form-based authentication
 - Proxy mode for a entire domain
- Proxy mode enables PAPI@RedIRIS
 - A way of getting experienced with PAPI by means of full proxying from the RedIRIS servers

PAPI@RedIRIS

- Simplify the adoption of the PAPI technology



PAPI@RedIRIS

- Provides
 - As many GPoA/PoA as necessary
 - Inside the RedIRIS or the institution domain
 - Local control of the user interface elements
 - If the RedIRIS directory is used, a Web interface for entry management
 - Usage statistics
 - Both of them, protected by PAPI
- Currently in production
 - >400 users, 19 institutions, 6 providers (>800 electronic resources)
- Pilots and evaluations for other institutions

Widening user communities

■ Inside the academic world

- Internet interface to control and collecting data from experiments with the Stellerator fusion reactor.
 - Collaboration with CIEMAT, partially funded by EURATOM
- The GLAM project
 - At the University of London Library, funded by JISC
- Bibliotecnica
 - A library portal at the UPC

■ And a little beyond

- A call for tenders for AA systems at one of the Spanish state governments includes PAPI support
- Contacts for technology transfer with a couple of companies
 - Internal usage, customer services and consultancy

Improving the technology

- Discussing interoperability with the Shibboleth team
 - Definition of scenarios
 - Trust model
- Make bigger the development team
 - Enhance user friendliness
 - Both for final users and for administrators
 - Incorporate PMI technologies within PAPI
 - DRM issues
- And take advantage of TF-AACE/TF-LSD
 - External engines
 - Alignment with common specifications/practices
 - Common attributes and attribute semantics