

Portal Framework + Standards = Functionality Freedom

Francisco Pinto

Research Technologies Service/Humbul Humanities
Hub/Subject Portals Project, Oxford University, UK
francisco.pinto@computing-services.oxford.ac.uk

Michael Fraser

Research Technologies Service/Humbul Humanities
Hub, Oxford University, UK
mike.fraser@computing-services.oxford.ac.uk

Overview

- Portal Frameworks and Portlets
 - General & SPP Perspectives
- Portal Related Standards
 - Local & Remote Portlets
- SPP Portal Framework
 - Identified Functionality, Structure & Architecture
- Access Management
 - Local & Athens AMSs
 - AM Functionality
- Conclusion
 - Where We Are Now
 - SPP
 - Access Management
 - Functionality Freedom?

What is a Portal in General?

- Infrastructure for Services
 - Similar to a Windows-based Operating System
 - Basic Services
 - Windows Management
 - User Management
 - User Preferences
 - Security
 - Registry
 - ...
- Services
 - Similar to a Window
 - GUI Presentation
 - Provide Access to the Applications (Content)

What is a Portal for SPP?

- Portal Framework
 - Infrastructure for Services
 - Supporting Software to Provide Access to the Content
 - Independent of the User's and Content Software
 - Users Access
 - Anytime: 24x7x365
 - Anywhere: Via Preferred Web Browser
 - Seamless Access to Resources
 - Built-in Functionality
 - Security
 - Access Management (Authentication & Authorisation)
 - Personalisation
 - User/Group Profiles
 - Presentation
 - Variant of Model View Control (MVC) Paradigm (e.g. JSP, XSP, Velocity , XML/XSLT)

What is a Portal for SPP?

- Portlets
 - Services
 - Web Components
 - Containers of Functionality (e.g. Servlets)
 - Abstractions or Metaphors
 - Channels, Modules, Etc.
 - Pluggable Functionality
 - Common Integration Layer
 - Between the Content and the Presentation
 - Standards to Access the Content
 - Z39.50, SOAP, XML/RSS, SMTP/POP3/IMAP...
 - Standard-based
 - JSR 168, WSRP, ...
 - » Ensure Interoperability Across Different Portal Frameworks
 - » Third Party Development Potentially Reduces Time and Cost Becoming Crucial for the Institution's Core Business Infrastructure

Best Portal Framework?

- Criteria?
 - Integration with Existing Functionality
 - Easy to Develop New Functionality
 - Programming Language Independence
 - Standards to Access Content
 - Standards for Interoperability and Portability

All are Incomplete or have Deficiencies

Standards is the Solution

Portal Related Standards

- Java Specification Request (JSR) 168
(<http://www.jcp.org/en/jsr/detail?id=168>)
 - Java Community Process (JCP)
 - Open Organisation of Java Developer Institutions with the remit to develop and revise specifications, reference implementations for the Java Platform
 - Portlet Abstraction
 - Java Portlet API
 - Interoperability and Portability between Portals and Portlets
 - Status
 - Community Review: April 2003
 - Public Review: June 2003
 - (Approved with comments from IBM, BEA, SAP and Oracle)
 - Release Version 1.0: August 2003

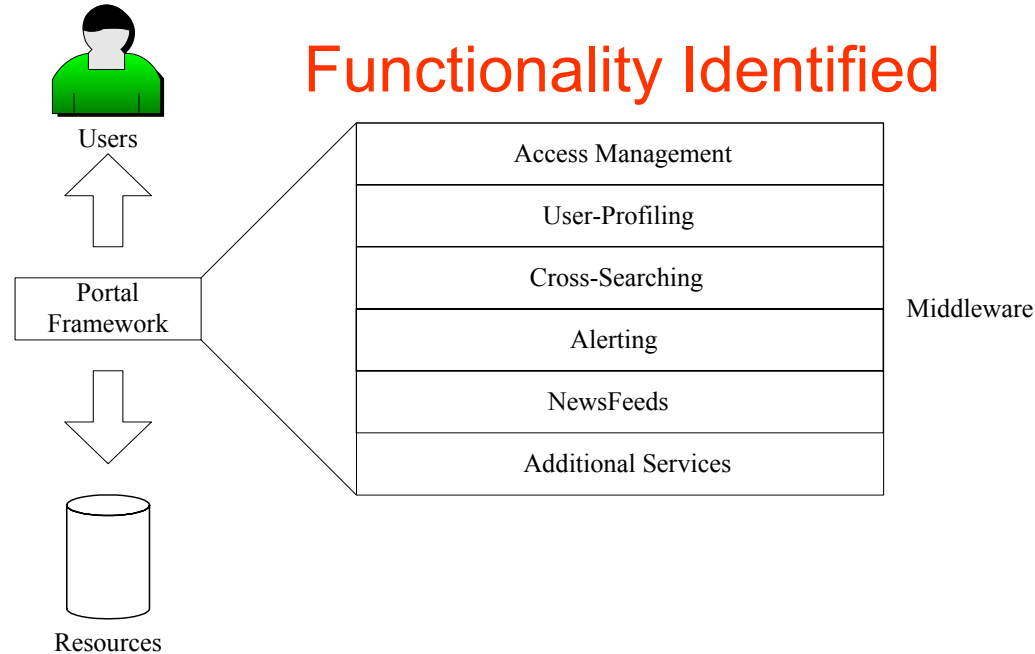
Portal Related Standards

- Web Services for Remote Portlets (WSRP)
(http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp)
 - Organization for the Advancement of Structured Information Standards (OASIS)
 - World-Wide Consortium that Drives the Development, Convergence and Adoption of e-Business Standards
 - Portlet Abstraction
 - XML and Web Services to Specify, Exchange and Plug Portlets in any WSRP-Compliant Portal Frameworks
 - Interoperability and Portability between Portals and Remote Portlets
 - Status
 - Approved Public Review: 31st May
 - IBM and Vignette have Draft Implementations

Portal Related Standards

- JSR 168 vs. WSRP
 - Work at Different Levels
 - JSR 168 Specifies the Interfaces for Local Portlets
 - WSRP Specifies the Interfaces for Exchanging Remote Portlets Across Portal Frameworks
 - Have to be Aligned
 - Same Notion of the Objects
 - Instantiate Locally the Portlets
 - Details of the Portlet API Have to be Exposed to WSRP in Order to Exchange Them

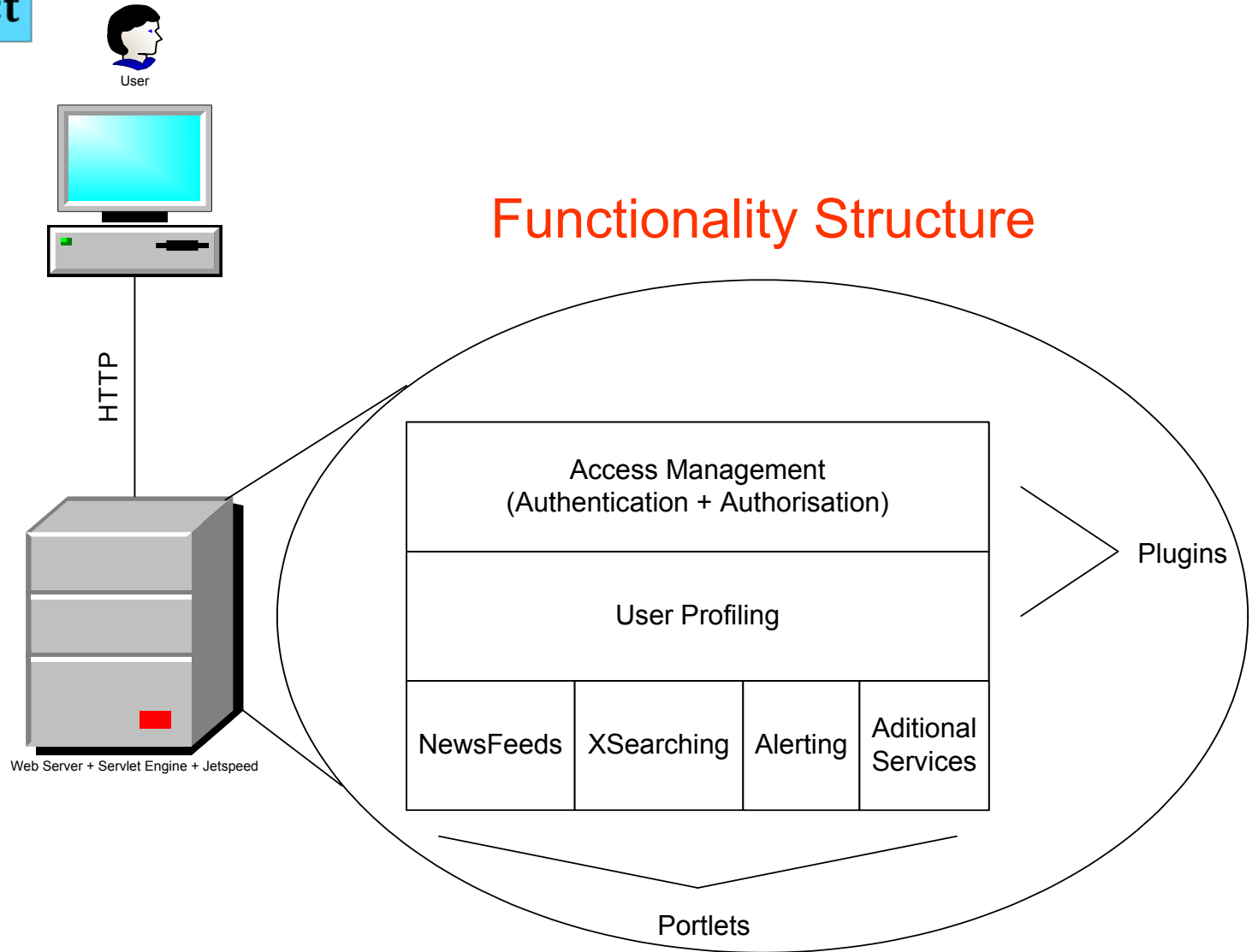
Portal Framework



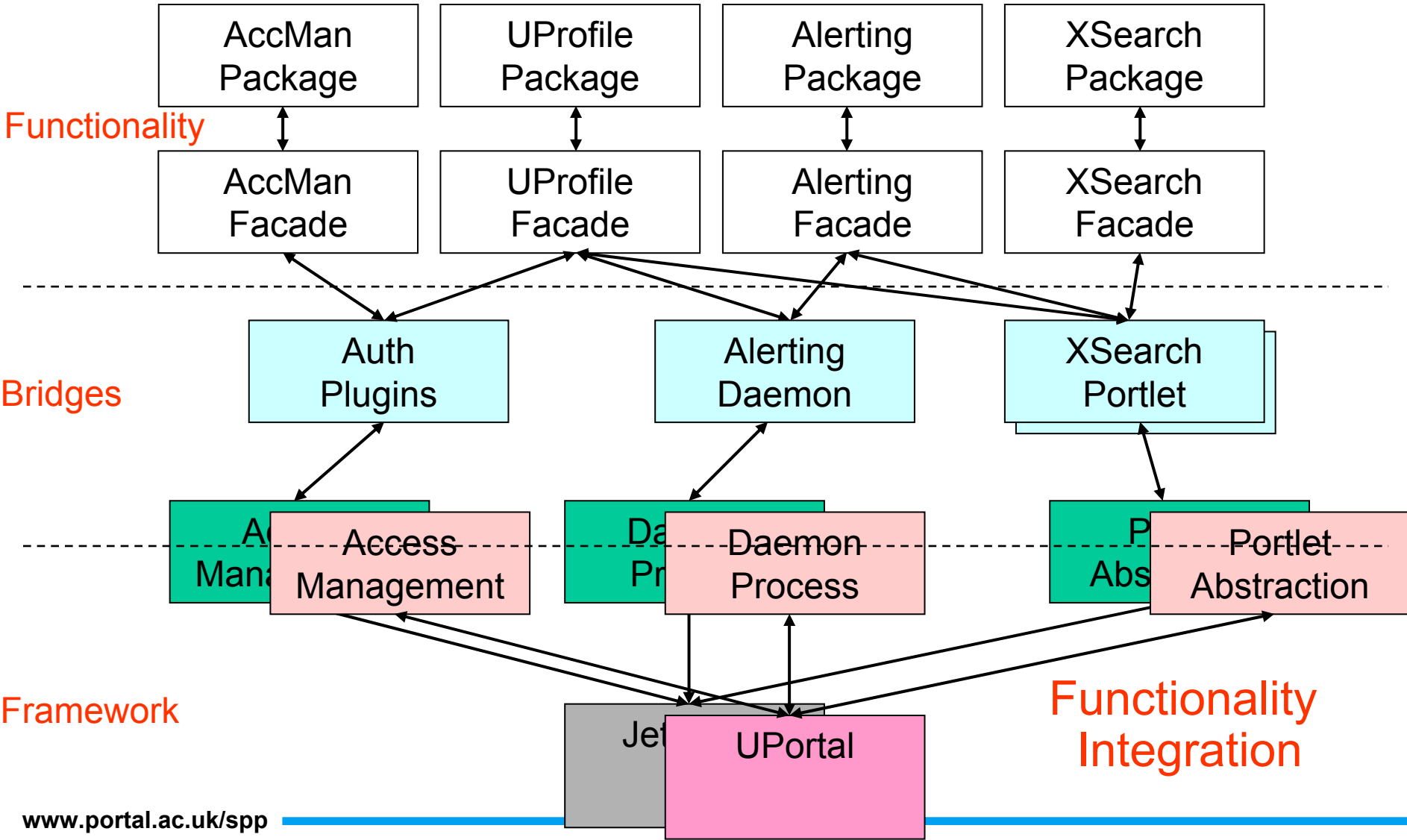
- **Jetspeed**
 - Portal Framework
 - Portlets Abstraction
 - Java Based
 - Open Source (Apache)
 - Open Standards
 - JSR 168, ?WSRP?

- **uPortal**
 - Portal Framework
 - Channel Abstraction
 - Java Based
 - Open Source (JA-SIG)
 - Open Standards
 - WSRP, ?JSR 168?

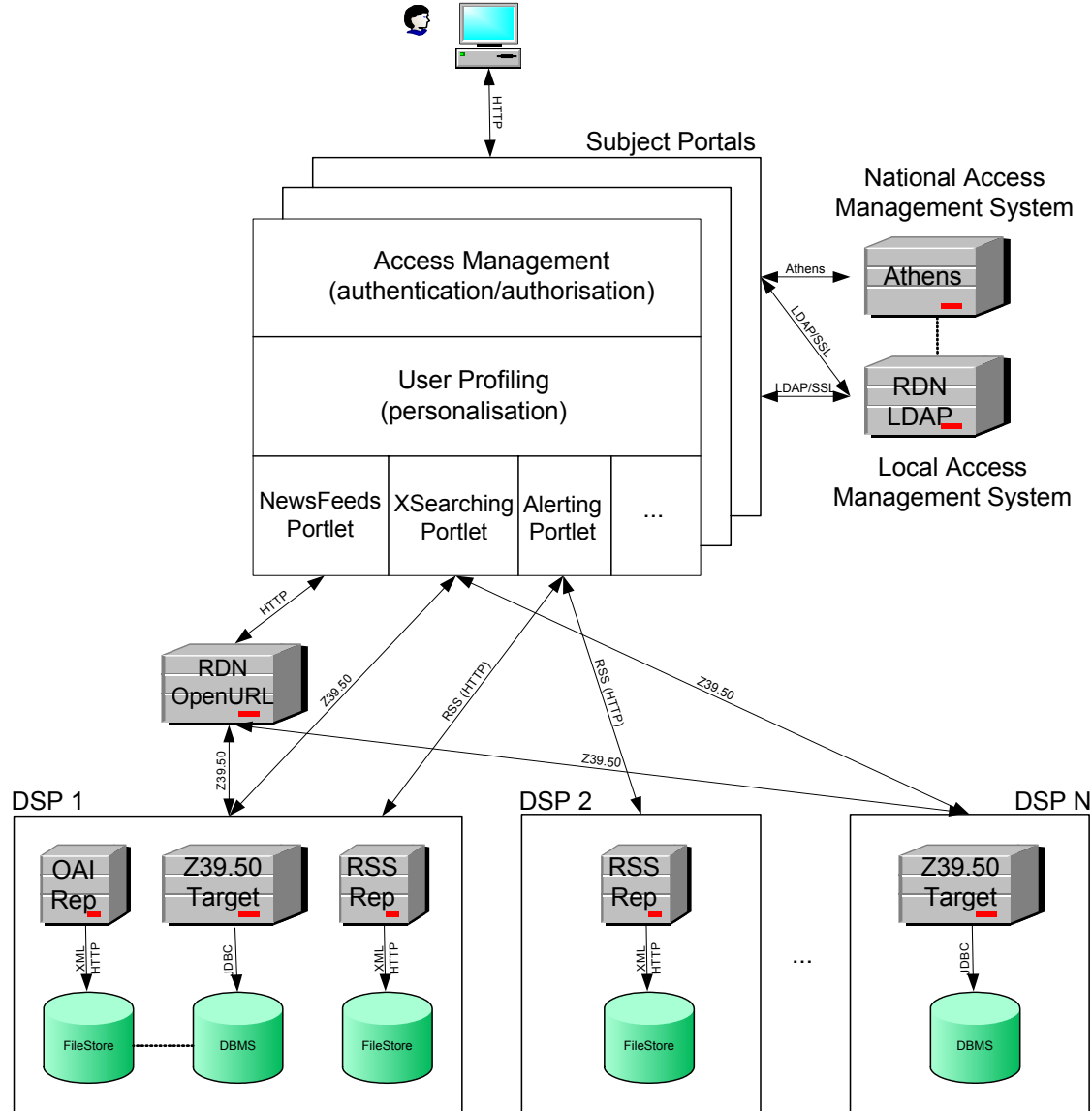
Portal Framework



Portal Framework



SPP Architecture



Local AMS

Not all SPP Users will be Athens Users

- Local AMS
 - RDN Context in-line with the Shared Services
 - LDAP Based and SSL Protected
 - Standard Schemas (e.g. eduPerson)
 - Identity
 - Essential for Personalisation Services
 - User Profiling, Saved Searches, Alerting Services
 - Requires Authentication and Authorisation

Athens AMS

- Classical API
 - Username/Password
- Athens Single-Sign On (AthensSSO)
 - Login Once at the Athens Authentication Point (AAP)
 - Authentication Protected Domain
 - Use a Ticket for Every Other Athens Protected Resources/Services (DSP)
 - Valid for a Browser Session or for 8h
 - User Just Exchanges Transfer Tokens
 - Valid for 60s
- Athens Single-Sign On++ (AthensSSO++)
 - Extend SSO to Access non-HTTP-SSO DSPs

Athens AMS

- Devolved Authentication (AthensDA)
 - Institution use their own “Trusted” AMS
 - Can be Implemented
 - LDAP-based Authentication
 - Digital Certificates within a PKI
 - Institution's VLE
 - Athens Just Provide Authorisation
 - Relies on AthensSSO++
 - Similar to Shibboleth (Internet2)

AM Functionality

- AM Interceptor
 - Accesses Directly the Local AMS

or

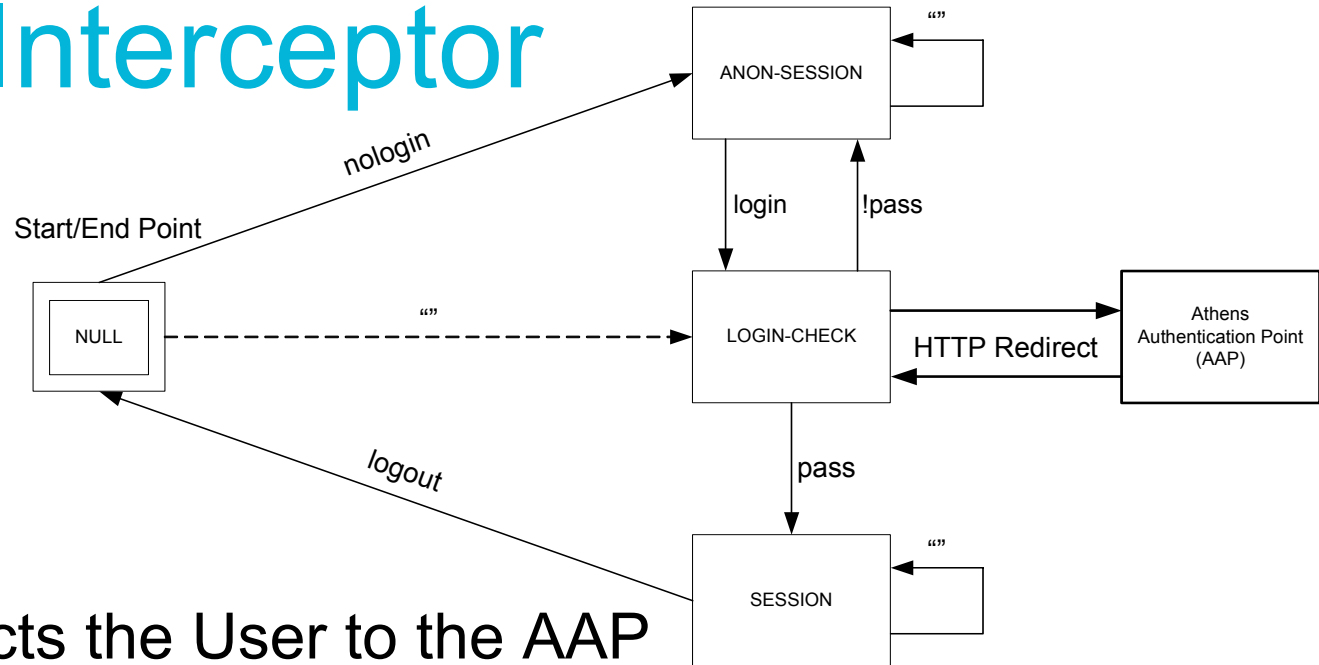
 - Redirects the user to the Athens AAP
 - Provides Transfer Tokens for Authentication
- AM Implementation
 - AM Plug-in
 - Directly Integrated in the Portal Framework
 - Pluggable Login Modules

or

 - Portal Service as a Web Service via SOAP
 - Pluggable Login Modules

AM Functionality: Interceptor

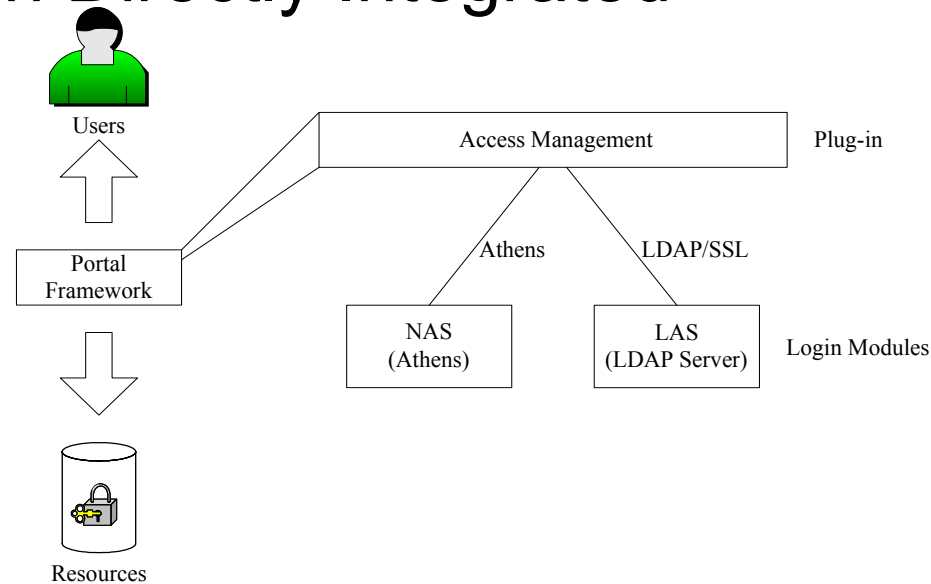
Finite
State
Machine



- Redirects the User to the AAP
- Verifies if the User is in an Athens SSO Session
 - If Not
 - Asks for Credentials
- Verifies if this is the User's Preferred AMS
 - If Yes
 - Creates a Portal Session
 - Loads the User Information from a Shared Service (LDAP/SSL)
 - Schemas: eduPerson (eduCause/Internet2) and SPP Private

AM Functionality: Implementation

- AM Plug-in Directly Integrated



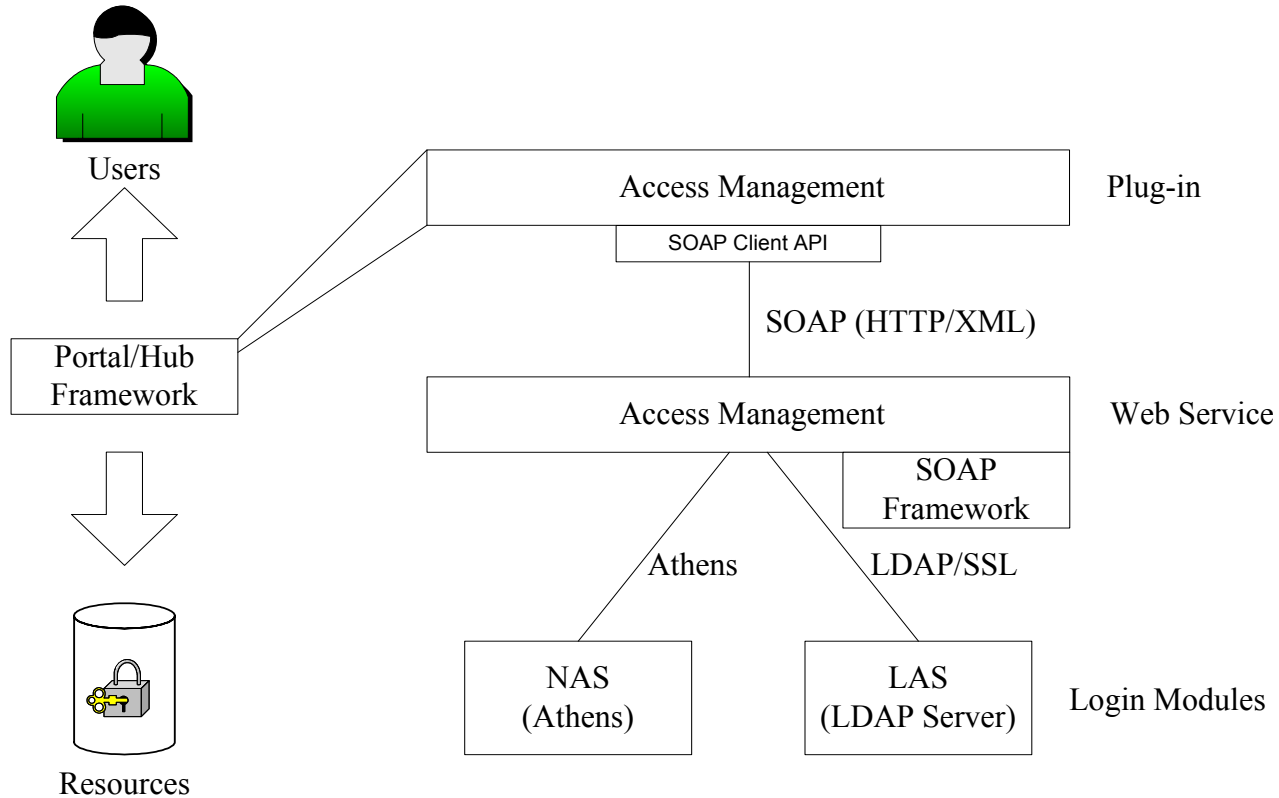
- JAAS Login Modules
 - Pluggable Authentication Modules (PAM)
 - Loaded in Run-Time According to the AMS
 - Athens: National AMS (NAS)
 - LDAP/SSL: Local AMS (LAS)

AM Functionality: Implementation

- AM Plug-in Directly Integrated
 - Pros
 - Simple Solution
 - Easy to Replace the Existing Functionality
 - Efficient
 - Runs in the Same Framework Environment as the Portal
 - Cons
 - Difficult to Integrate in Multiple Frameworks
 - No Standards at the Framework Level
 - AM is Developed in Java. What About Other Programming Language-based Frameworks?
 - Static Interoperability

AM Functionality: Implementation

- AM Plug-in Portal Service



AM Functionality: Implementation

- AM Plug-in Portal Service
 - Pros
 - Easier to Integrate in Multiple Frameworks
 - SOAP Client APIs for Most of the Programming Languages
 - » Access the AM Web Service
 - There are SOAP Frameworks for Many Programming Languages
 - » Enable to Run and Reuse Existing Functionality at the Hubs
 - Cons
 - Require an Additional Level of Communication
 - Complicates the Architecture
 - » SOAP Runs on Top of HTTP Using XML
 - » Initial Tests Shown: Average 1.4 Times Less Efficient
 - » What About Highly Interactive Functionality

Conclusion: Where We Are

- SPP
 - Currently
 - Middle of Round 3 Phase
 - Following
 - Prototype Subject Portals
 - Documentation
 - Dissemination
 - Exit Strategy
 - One Year (Starting in September 2003)

Conclusion: Where We Are

- Access Management
 - Package
 - Local and National AMSs
 - Direct and SOAP Based Integration
 - JAAS Login Modules
 - LDAP/SSL for LAS (JNDI and JLDAP)
 - AthensSSO++ for NAS
 - Approach Proved to be Reasonable
 - Flexible to Work with Multiple AMSs
 - New Login Modules Developed in Few Days
 - SPP is Working with X.509 Certificates
 - Prepare SPP for a Possible Future JISC AMS

Conclusion: Functionality Freedom?

- Portal Framework Independency?
 - Write New Bridges for Plug-ins ☹️
 - Integrate AM Interceptor ☹️
 - Reuse Existing Plug-ins 😊
- Portlets Independency?
 - Write a Bridge for the Abstract Portal ☹️
 - Reuse Existing Functionality 😊

OR...

Move to Standards and get Freedom 😊

I mean, emergent Standards ☹️