Enterprise Sign On Engine (ESOE)

Bradley Beddoes - QuestNet 2007
http://www.esoeproject.org

What is an “ESOE”?

- It’s essentially a system that tackles some hard problems associated with electronic collaboration, to make it easier for end users to work, with some extra perks for administrators
- It’s middleware, so if you can’t see it, then it’s doing its job.
- It’s standards based, it’s open source, it’s the latest Java technologies and software development methodologies
- It takes care of Authentication, Single Sign On (there is a difference!) creation of authorization policies and enforcement of access control. It federates with business partners easily. Your electronic content is more accessible and more secure at the same time.

Why did we build it?

- There was a need for a new identity management and SSO system, nothing in the market place could really solve our needs and we looked around a lot for something to solve our problems.
- We want to make access to electronic resources easy for our students and staff, less to remember and easier to use at QUT and externally, otherwise you end up with....
- We're into Shibboleth, OpenID and others, we wanted to use the technologies widely but reduce the complexity of getting them running, while at the same time being able to extend to new technologies in the future

What is the design based on?

- **OPEN STANDARDS** of course!

- **SAML 2.0 -** Security Assertion Markup Language - an XML-based framework for ensuring that transmitted communications are secure. SAML defines mechanisms to exchange authentication, authorization and nonrepudiation information, allowing single signon capabilities for Web services.

We have SAML 2.0 libraries in Java and C++ you can use today if your working in the field

- **XACML 2.0 -** eXtensible Access Control Markup Language - an XML-based framework that expresses security policies and access rights to information for Web services, digital rights management (DRM), and enterprise security applications. We didn't quite implement the full XACML 2.0 spec because some of it simply isn't relevant, so to avoid confusion we term our implementation LXACML.

What authentication sources does ESOE support?

- The short answer is anything you like.
- The longer answer is we've written implementations for LDAP and Active Directory but with the plugin architecture we have defined you could write a module to authenticate from any source, even a flat file if you so desire (and yes we did it in testing).
- The pipeline can also allow you to do multiple authentications over time, increasing for example levels in multi-factor scenario which can then be used as a basis for access control.
- ESOE is able to support multiple authentication sources at the same time with its pipelined authentication design if some users have an account in one LDAP server and others in a database for example both can be used as sources of authentication at the same time.

What identity resolution sources does ESOE support?

- Just as for authentication basically anything you care to write a plugin for.

- Again we have supplied a generic LDAP connector out of the box

- Identity can actually be sourced from multiple repositories, cleaned and given to applications as a single representation. e.g. Your student system database might have details about the users roles as an attribute called “membership” while a faculty they belong to might store additional data in LDAP as an attribute called “facultyAccess”. ESOE can aggregate these two sources of identity, creating an institution wide accepted attribute called “roles”, applications can then consume this data without any special configuration.

How does ESOE do access control?

- Everything is based on LXACML policies which are very flexible and allow fine grained control over who can access what. This is very important when you consider that ESOE allows people to come into the system from basically anywhere

- Each application deploys a piece of software called a Service Provider Enforcement Point “SPEP”. This software does a number of things to communicate with the ESOE but the most important job it does is access control.

- For each resource request (lets say /secure/index.jsp) the SPEP will send a message to the ESOE to ask if this is allowed. The ESOE, based on all the knowledge it has about the principals identity and in combination with the LXACML policy will return a Permit or Deny statement. The SPEP then acts accordingly allowing the request to pass or denying it.

- Our SPEP software also does some interesting things with caching though the algorithm and math is complex and I won’t bore you all to death with specifics, needless to say its designed to prevent “ESOE death”.

- Additionally ESOE authorization can work hand in hand with existing access control features already in applications.

How does ESOE support Federation?

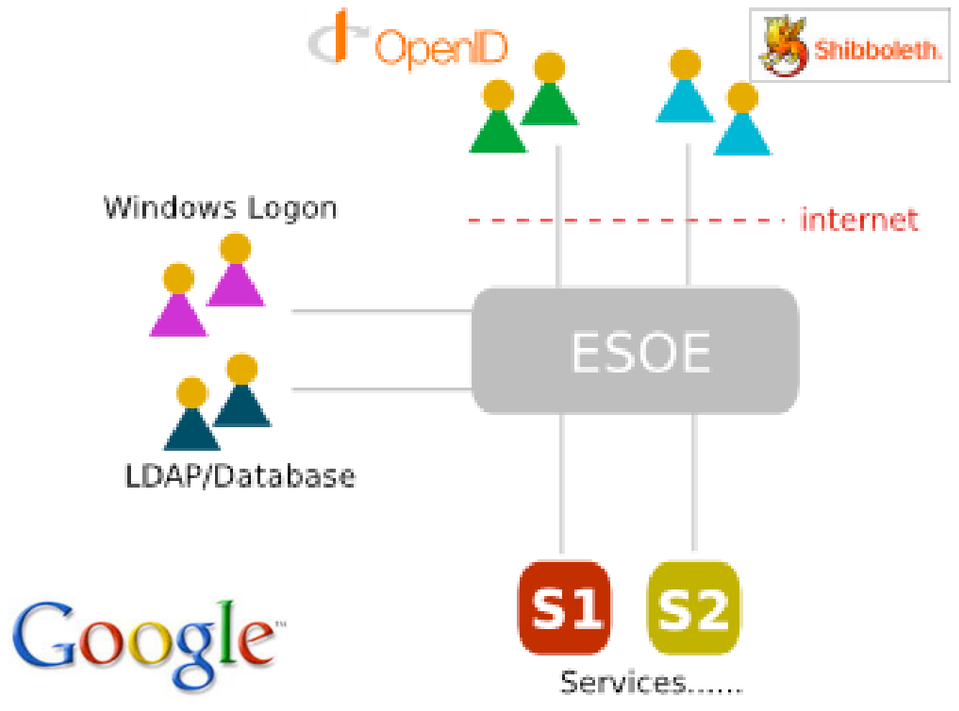
- ESOE has a centralized design, every SPEP in the authentication network is connected to it at all times.

- If ESOE has modules written to understand it then applications automatically understand it.

- Currently ESOE is able to communicate with users authenticating via:
  Shibboleth 1.3.x
  OpenID 1.x and 2.x

- In the future we may expand to support Yahoo BBAuth, WS-Federation and others.

- We're working with Google on some interesting code that may prove very useful in the federated application space.



A few demos to help show what ESOE does

- ESOE true single sign on support
- ESOE non windows and external client support
- ESOE Shibboleth support - Using Australian Access Federation
- ESOE OpenID support
- Time allowing demonstration of real time policy modification

Development Team

- Bradley Beddoes - Lead Architect
  b.beddoes@qut.edu.au

- Andre Zitelli - Senior Programmer
  a.zitelli@qut.edu.au

- Shaun Mangelsdorf - Programmer
  s.mangelsdorf@qut.edu.au

- Terence Smith - Project Manager
  t.smith@qut.edu.au

ESOE Community - Please feel free to join!

- **http://www.esoeproject.org**

- Mailing lists
- Online wiki
- Jira Issue Tracking
- Fisheye
- More on the way.

http://www.esoeproject.org