Digital Identity from LDAP to SAML and beyond

November 8, 2007

Pat Patterson
Federation Architect
pat.patterson@sun.com
blogs.sun.com/superpat

Agenda

- foreach (LDAP, SAML, Web Services, OpenID, Cardspace, OAuth, Concordia)
  - Background
  - Protocol
  - Use
- Goal is for you to have a superficial understanding of the range of options out there and some basis for selecting one of them for your next identity-related project

Setting the Scene - the 1990s

- Identity in silos
- X.500
  - Directory Access Protocol (DAP) over OSI stack
- Early NOS directories (Novell NetWare)
- Emergence of email
- LDAP
  - 1993
  - Tim Howes (UMich), Steve Kille (ISODE), Wengyik Yeong (Perf Sys Intl)

What is LDAP?

- Evolved from X.500
- Lightweight Directory Access *Protocol*
  - ASN.1/BER via TCP/IP on port 389
- LDAPv3 - RFC 2251 - published 1997
- Hierarchical database model
  - dc=example,dc=com
    - ou=People
      - uid=patp
        - cn: Pat Patterson
        - mail: pat.patterson@example.com
        - objectClass: inetOrgPerson
        - objectClass: organizationalPerson
        - ...

LDAP 10 years ago

- Email address book
- White pages for Enterprises
- Mostly Read Access
  - Fast
  - Thousands read requests per seconds
- Small data sets
  - 100,000 user entries was BIG
  - 20 attributes was a lot
- Very infrequent changes
  - Less than10% writes

LDAP Now

- Authentication source
  - Username/password
  - Certificates
- Role-Based Access Control
- Configuration store
- NOS, extranet, telco...
- Huge data sets
  - 10s of millions of entries is not unusual
- Access pattern closer to RDBMS

LDAP Basics

- Mozilla LDAP C, Java, Perl SDKs, JNDI, command line

```
$ ldapsearch -h localhost -p 1389 -s sub -b "dc=example,dc=com" -x
-LLL "(uid=patp)"
dn: uid=patp,ou=People,dc=example,dc=com
objectClass: person
objectClass: inetorgperson
objectClass: top
objectClass: organizationalperson
mobile: +1 680 734 6300
mail: patp@example.com
employeeNumber: 1
pager: +1 850 883 8888
sn: Patterson
postalCode: 93694
l: San Jose
cn: Pat Patterson
telephoneNumber: +1 390 103 6917
st: CA
uid: patp
givenName: Pat
homePhone: +1 280 375 4325
```

Typical LDAP Usage

- Authenticate user, retrieve profile
  - BIND as anonymous or admin user
  - SEARCH for user ID
    - Get Distinguished Name (DN) for user
    - May also get user attributes
  - BIND as user with DN
    - Can be simple plaintext password
    - SASL
      - Kerberos
      - Client certificate
      - etc

LDAP Directory Servers

- Sun Java System Directory Server
  - OpenDS
- Red Hat Directory Server
  - Fedora Directory Server
- OpenLDAP
- Novell eDirectory
- Microsoft Active Directory
- IBM, Oracle etc

LDAP Success Factors

- Standard Protocol
- Flexibility of the Information Model
  - Standard Schema
  - Extensibility
- Performance
- High Availability built in
- Simplicity

But It's Not All Goodness

- Many applications factor out authentication and even authorization via LDAP, but...
- One credential set means that users must still repeatedly present that credential
- The dream of a single directory per enterprise never came to pass
  - Regulatory concerns
  - Practical concerns
- Reality is multiple directories, many apps still maintain their own user repositories

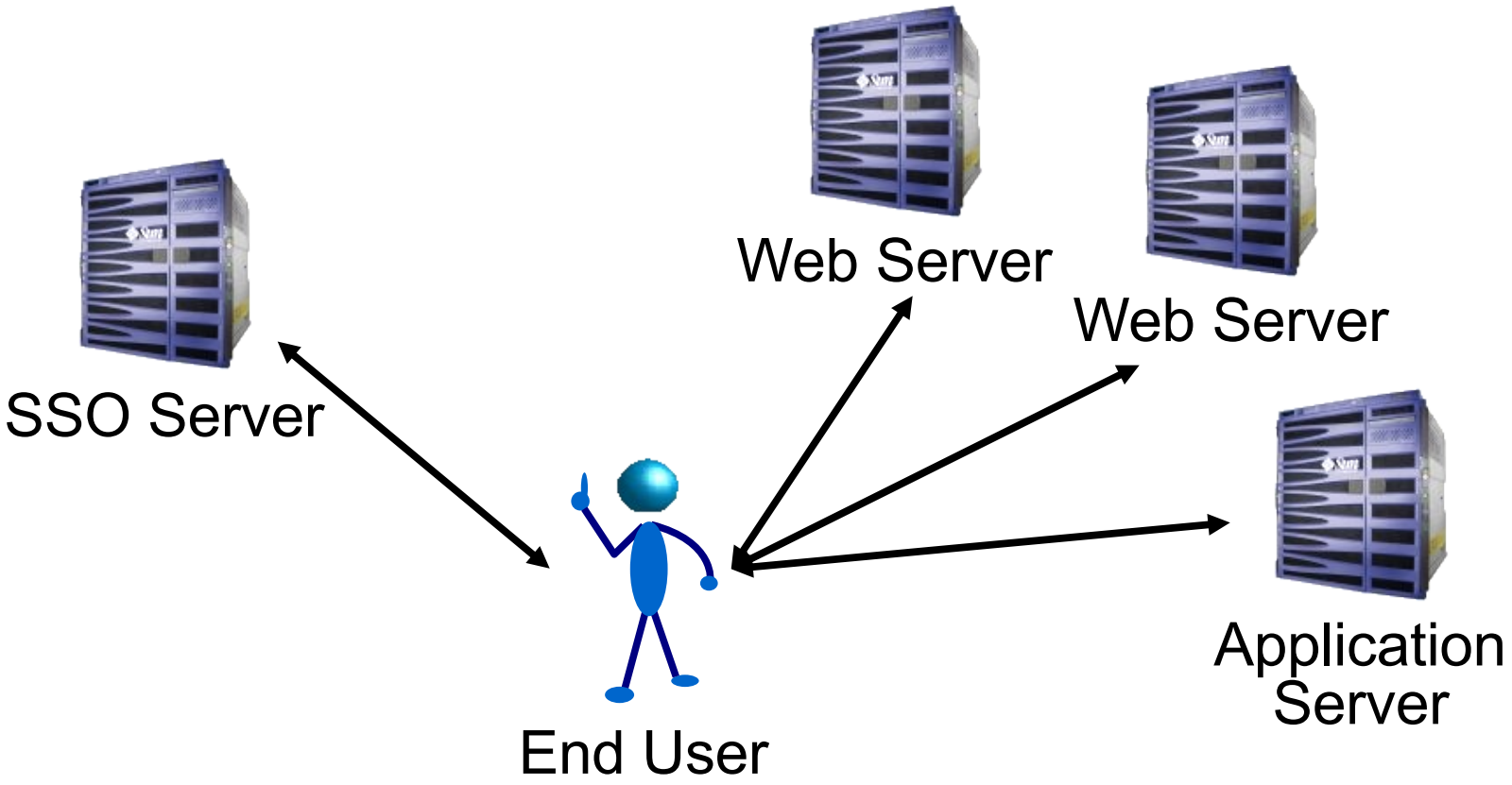
Enterprise Problems

- “Every application wants me to log in!”
- “I have too many passwords – my monitor is covered in Post-its!”
- “We're implementing Sarbanes-Oxley – we need to control access to applications!”
- “We need to access outsourced functions!”
- “Our partners need to access our applications!”

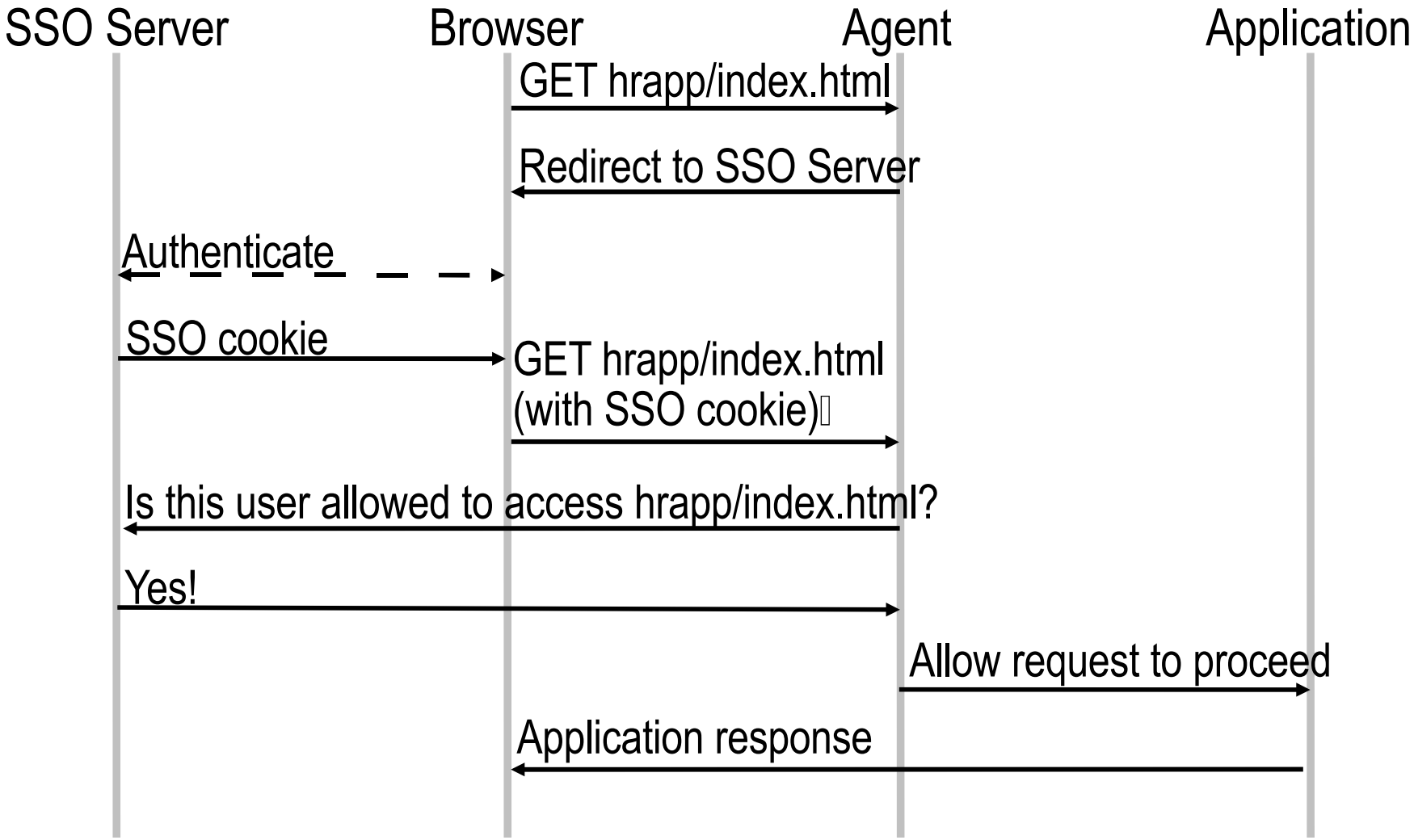
Web Access Management

- Simplest scenario is intra-enterprise
- Factor authentication and authorization out of web applications into web access management (WAM) solution
- Can use browser cookies within a DNS domain
- Proxy or Agent architecture implements role-based access control (RBAC)
- Users get single sign-on, IT gets control

SSO Within an Enterprise



How It Works


SSO Server
Browser
Agent
Application
GET hrapp/index.html
Redirect to SSO Server
Authenticate
SSO cookie
GET hrapp/index.html
(with SSO cookie)
Is this user allowed to access hrapp/index.html?
Yes!
Allow request to proceed
Application response

Web Access Management Products

- Sun Java System Access Manager
  - OpenSSO
- CA (Netegrity) SiteMinder Access Manager
- IBM Tivoli Access Manager
- Oracle (Oblix) Access Manager
- Novell Access Maneger
- JA-SIG CAS
- JOSSO

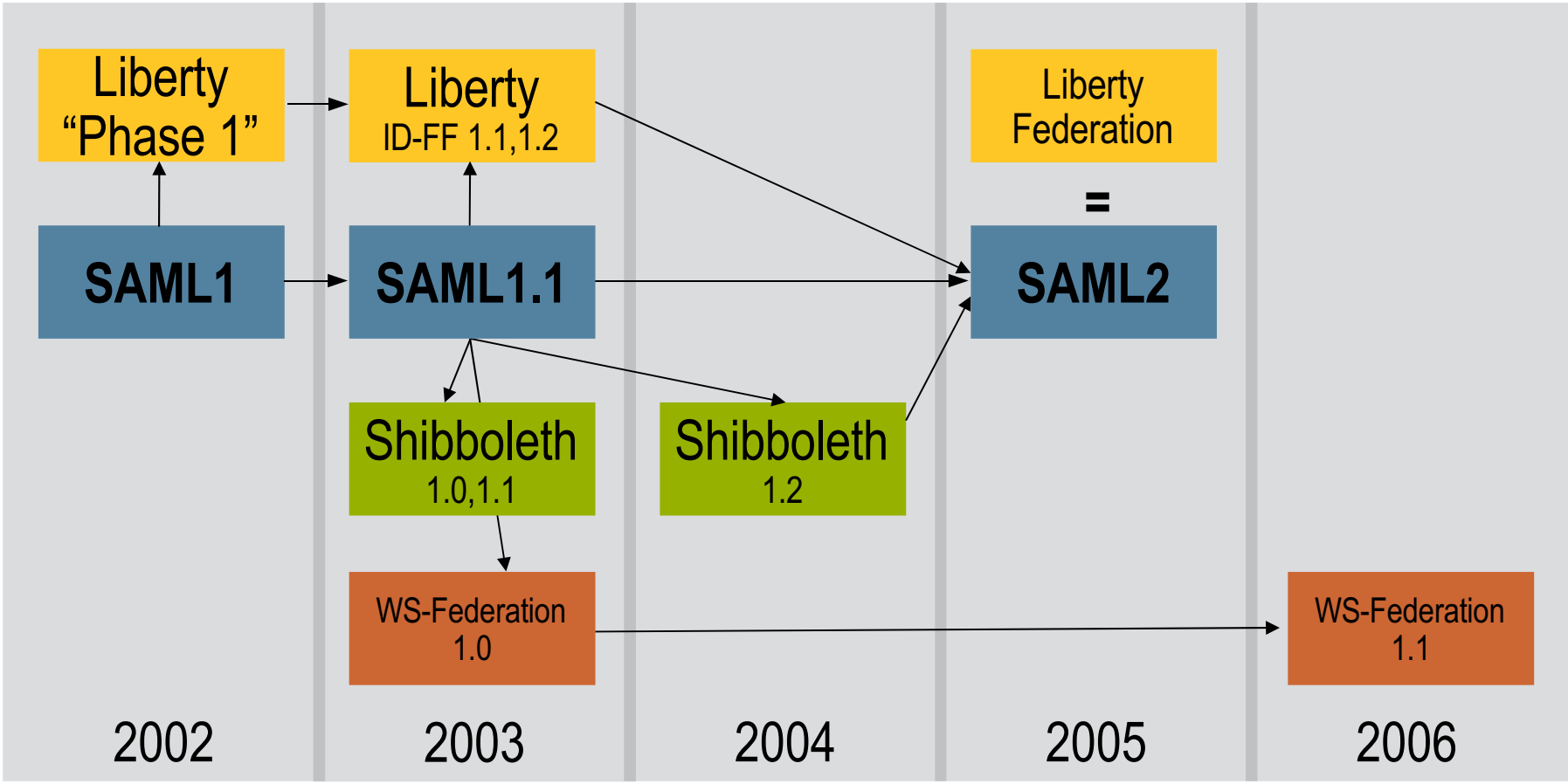
Enterprise Problems

- “Every application wants me to log in!”
- “I have too many passwords – my monitor is covered in Post-its!”
- “We're implementing Sarbanes-Oxley – we need to control access to applications!”
- “We need to access outsourced functions!”
- “Our partners need to access our applications!”

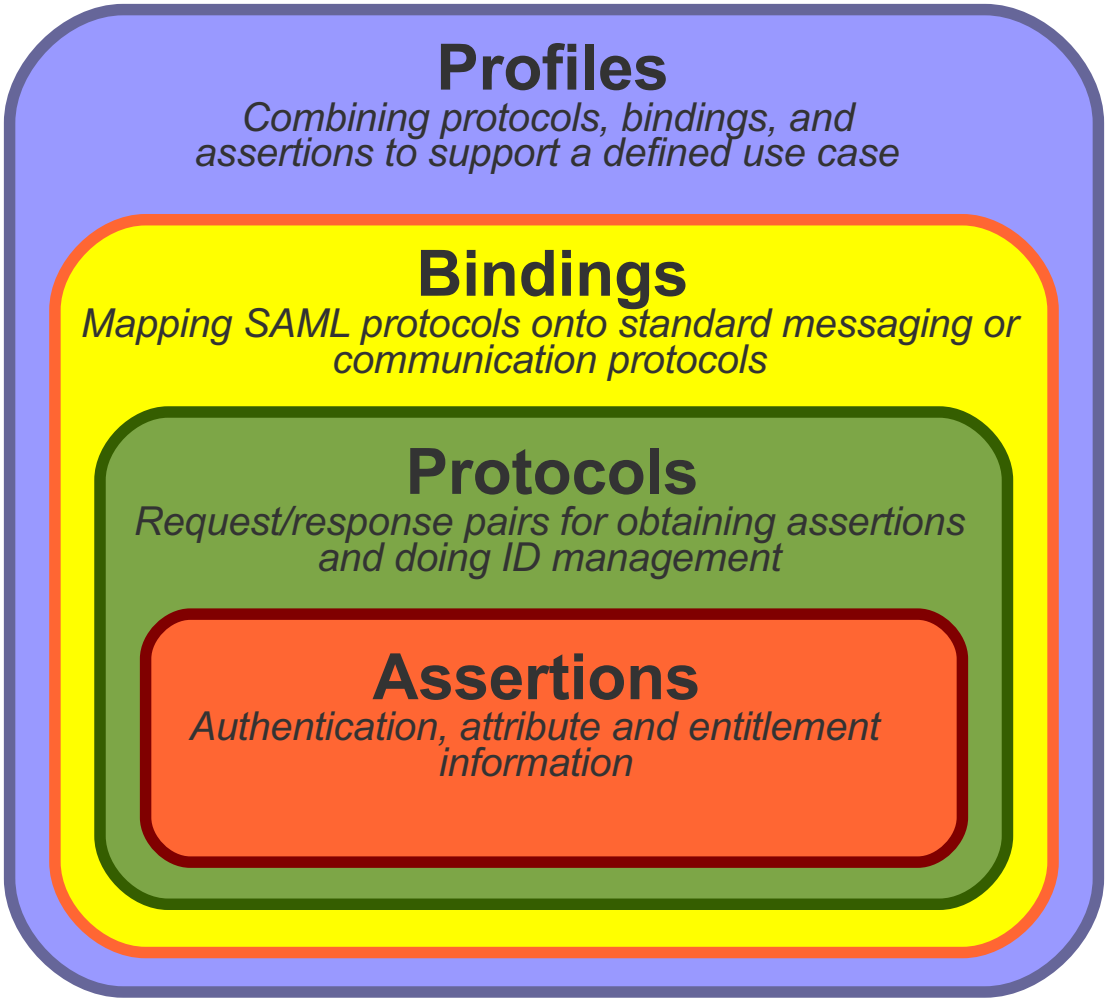
Single Sign-on *between* Enterprises

- Cookies no longer work
  - Need a more sophisticated protocol
- Can't mandate single vendor solution
  - Need standards for interoperability

Single Sign-On Standards



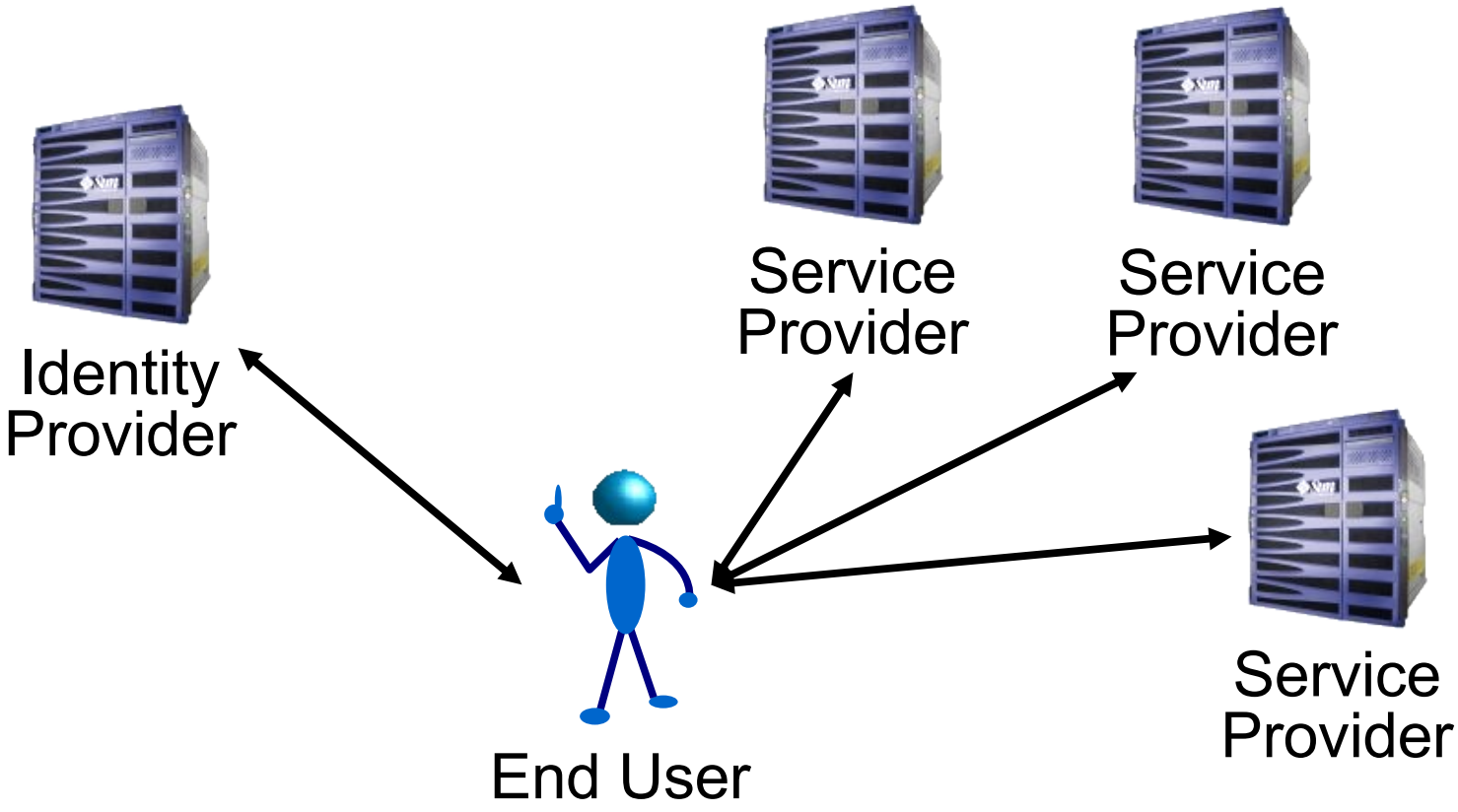
SAML 2.0 Concepts



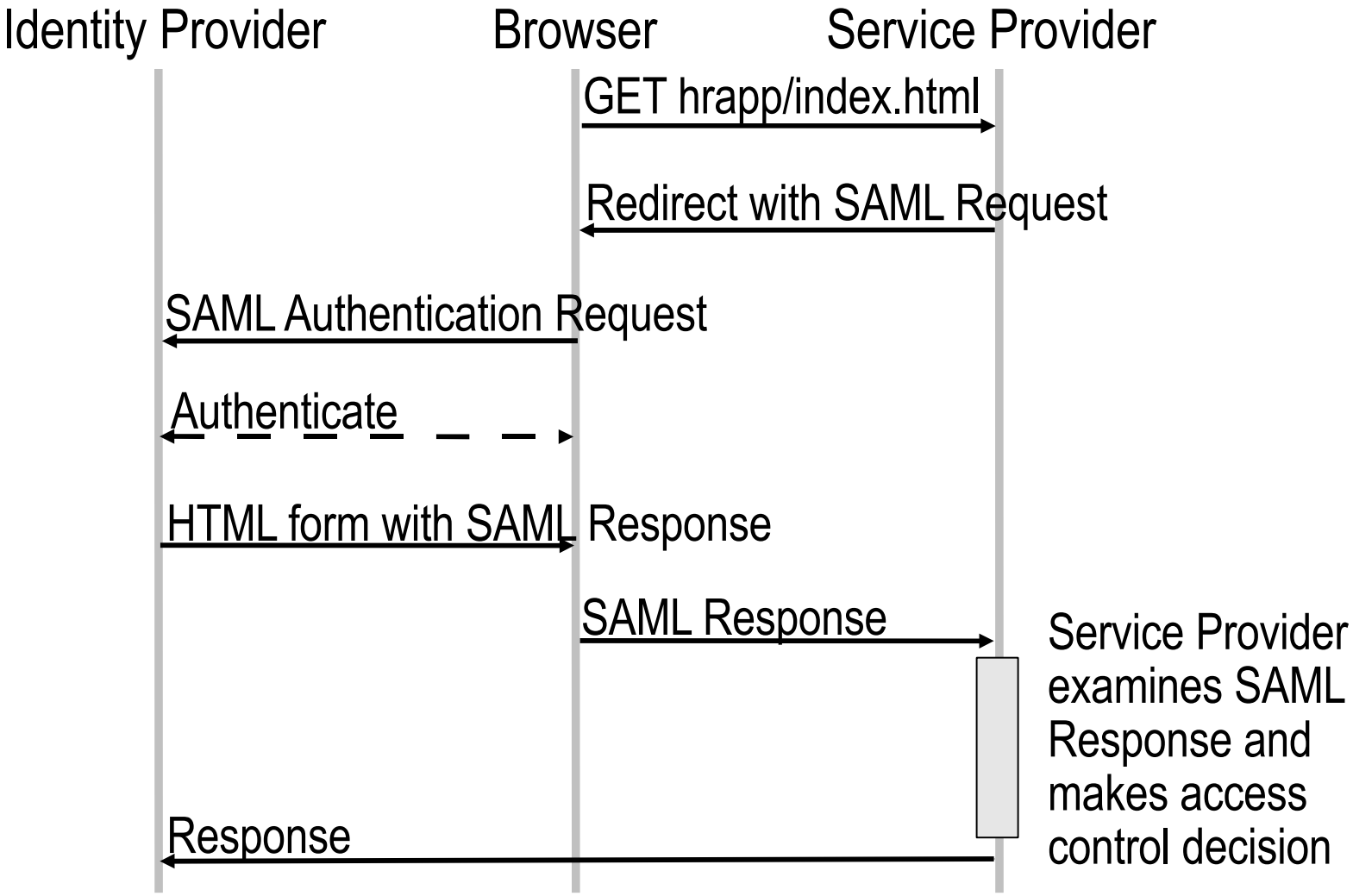
Authentication Context
Detailed data on types and strengths of authentication

Metadata
IdP and SP configuration data

SSO Across Enterprises



SAML 2.0 SSO Basics



SAML 2.0 Assertion

(Abbreviated!)

```
<Assertion Version="2.0" ID="..." IssueInstant="2007-11-06T16:42:28Z">
    <Issuer>https://pat-pattersons-computer.local:8181/</Issuer>
    <Signature>...</Signature>
    <saml:Subject>
        <saml:NameID Format="urn:oasis:...:persistent" ...>
            ZG0OZ3JWP9yduIQ1zFJbVVGHlQ9M
        </saml:NameID>
        <saml:SubjectConfirmation Method="urn:oasis:...:bearer">
            <saml:SubjectConfirmationData .../>
        </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions
      NotBefore="2007-11-06T16:42:28Z"
NotOnOrAfter="2007-11-06T16:52:28Z">
        <saml:AudienceRestriction>
            <saml:Audience>
                https://pat-pattersons-computer.local/example-pat/
            </saml:Audience>
        </saml:AudienceRestriction>
    </saml:Conditions>
    <saml:AuthnStatement AuthnInstant="2007-11-06T16:42:28Z" ...>
        <saml:AuthnContext>
            <saml:AuthnContextClassRef>
                urn:oasis:...:PasswordProtectedTransport
            </saml:AuthnContextClassRef>
        </saml:AuthnContext>
    </saml:AuthnStatement>
</saml:Assertion>
```

SAML 2.0 Adoption

- Sun, IBM, CA – all the usual suspects, except Microsoft
- OpenSAML (Internet2)
  - Java, C++
- OpenSSO (Sun)
  - Java, PHP, Ruby
- SimpleSAMLphp (Feide)
- LASSO (Entr'ouvert)
  - C/SWIG
- ZXID (Symlabs)
  - C/SWIG

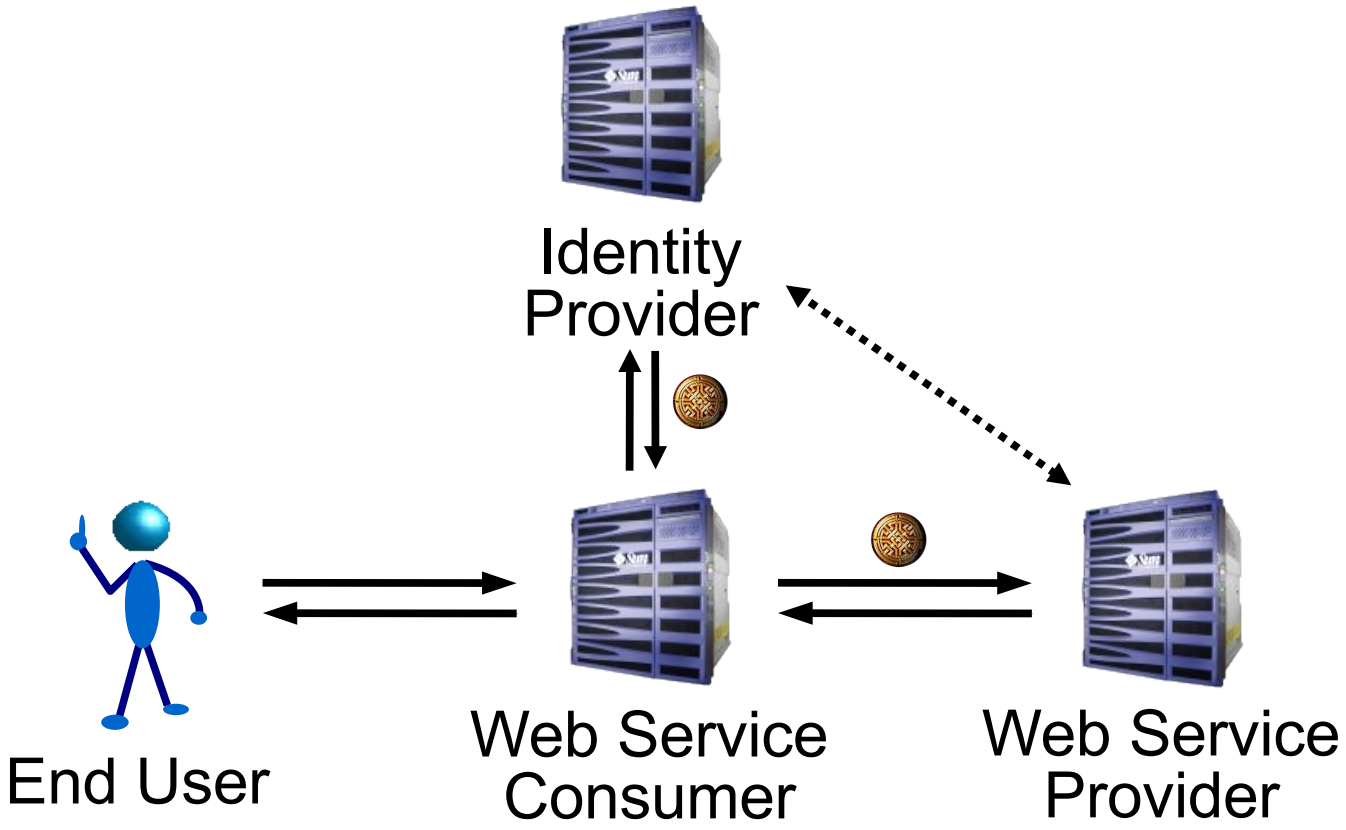
What About Web Services?

Transport Level Security

Transport Level Security != Identity

- Difficult choice between
  - No client authentication
  - Client authentication via certificates
- Scope of protection is limited to individual 'hops'
- Even with client authentication, no real non-repudiation due to difficulty of archiving and verifying message flow
- TLS/SSL is still essential for confidentiality and integrity at the transport level, but is not enough – we need a solution at the message level

Basic Web Services Security



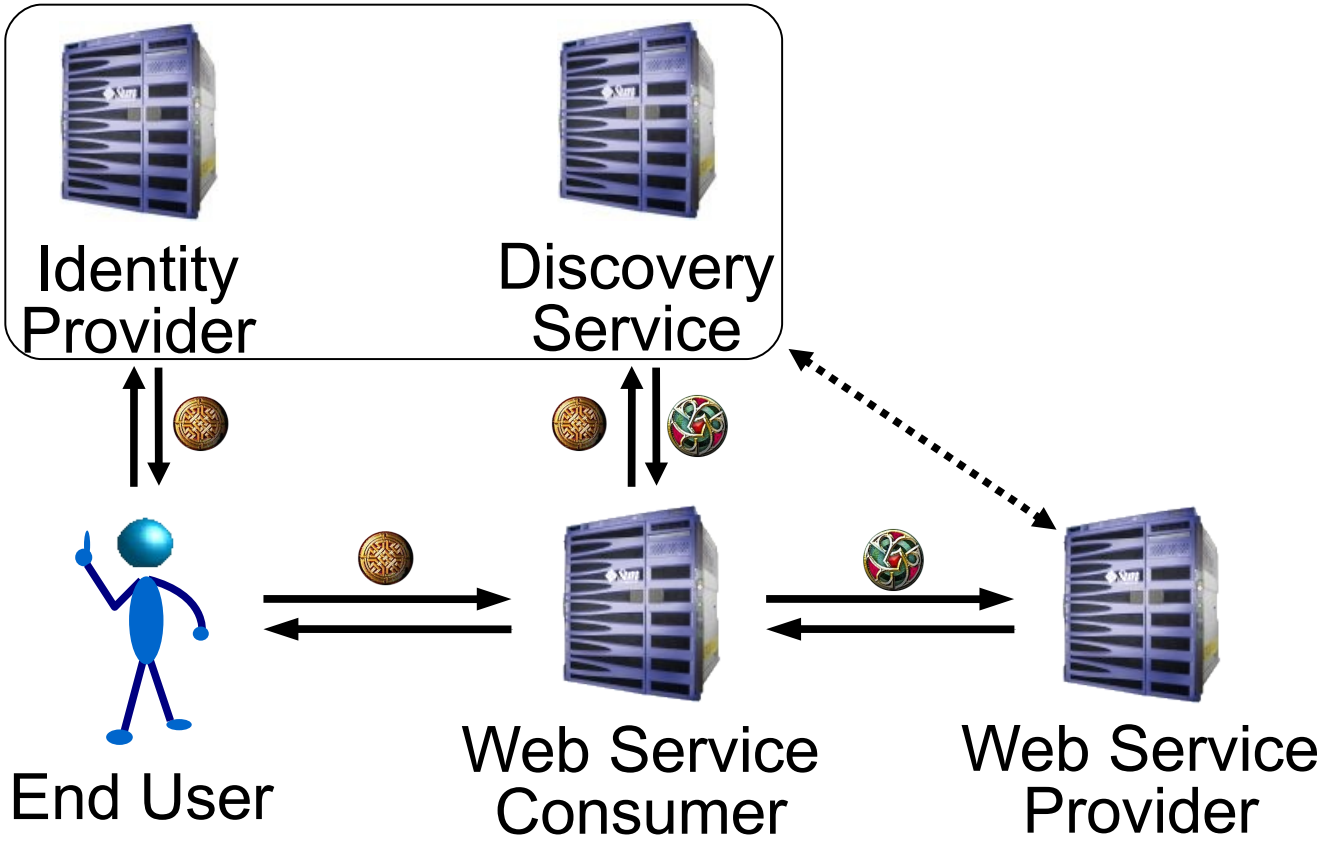
Message Level Security – Getting There

- Identity token carried in SOAP header
  - WS-Security, WS-I Basic Security Profile
  - Industry has converged on SAML Assertion as the token
- SAML allows for bearer tokens, holder-of-key tokens, audience restrictions etc
- Token can be archived with message
- But... restricting the audience to the immediate recipient leaves us with similarly limited scope of protection – one hop

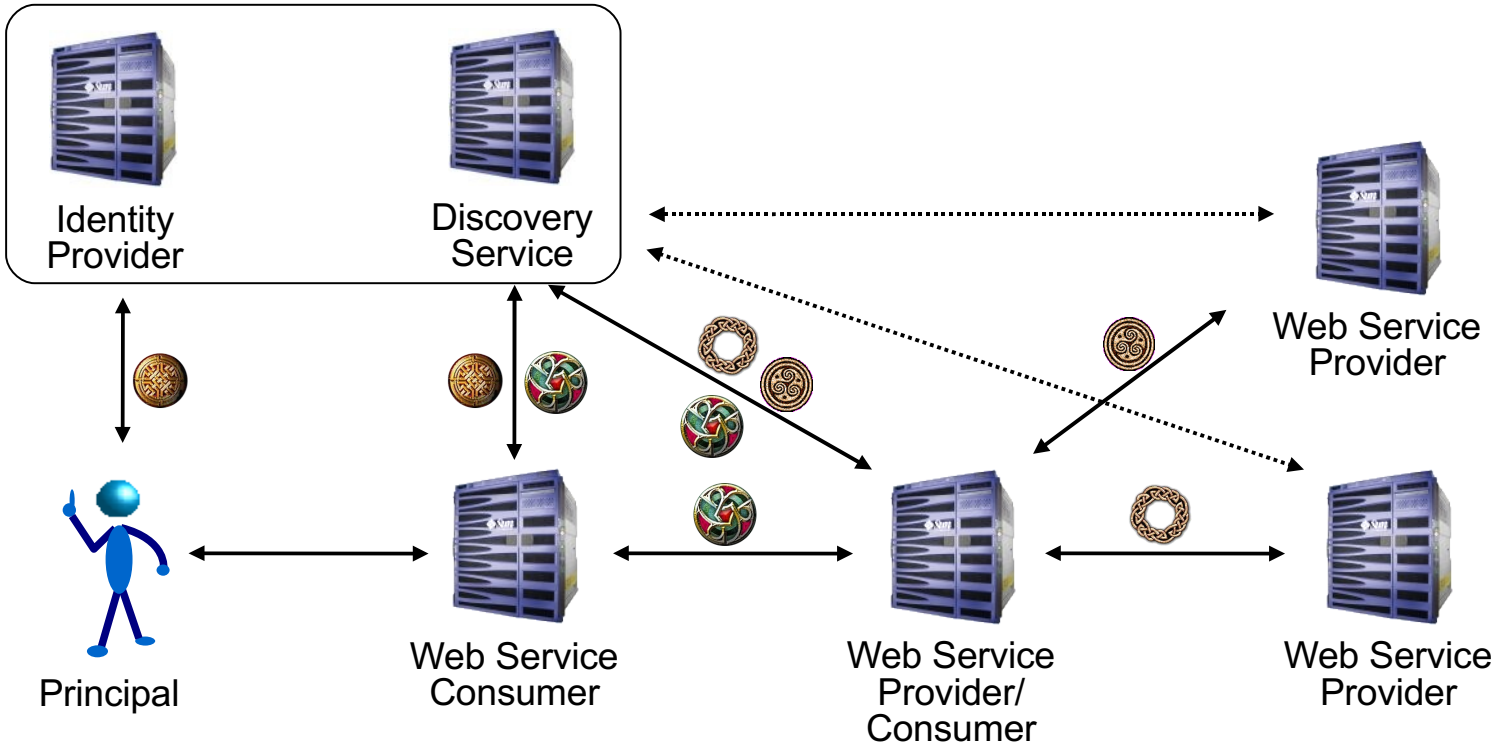
Requirements for Web Service *Identity*

- Identify the end user
- Locate the service
- Preserve identity
  - Across multiple 'hops'
  - Across domain boundaries
  - Across vendors' products
- Using existing technologies and idioms
- Maintaining privacy

Identity Web Services



Scaling Out


Identity Provider
Discovery Service
Web Service Provider
Principal
Web Service Consumer
Web Service Provider/ Consumer
Web Service Provider

Liberty Identity Web Services Framework (ID-WSF)

- Dynamic service discovery and addressing
- Common web services transport mechanisms to apply identity-aware message security
- Abstractions and optimizations to allow anything – including client devices – to host identity services
- Unified data access/management model for developers
- Flexibility to develop arbitrary new services
- User privacy through use of pseudonyms

ID-WSF 2.0

- February 2005 – October 2006
- SAML 2.0
  - Bootstrap from SAML 2.0 single sign-on
  - SAML 2.0 tokens
- People Service
  - End user group, role management
  - Cross-provider principal references
- Subscription, notification
  - Building on Data Services Template (DST) specification

People Service Use Case

- Alice and Bob have accounts at identity providers
- Alice's identity provider has deployed a People Service
- Alice has an account at photos.example.com, linked to her identity provider account
- Alice wants to share some photos with Bob, who has no photos.example.com account and doesn't want one
- http://www.projectliberty.org/liberty/content/download/387/2720/file/Liberty_Federated_Social_Identity.pdf

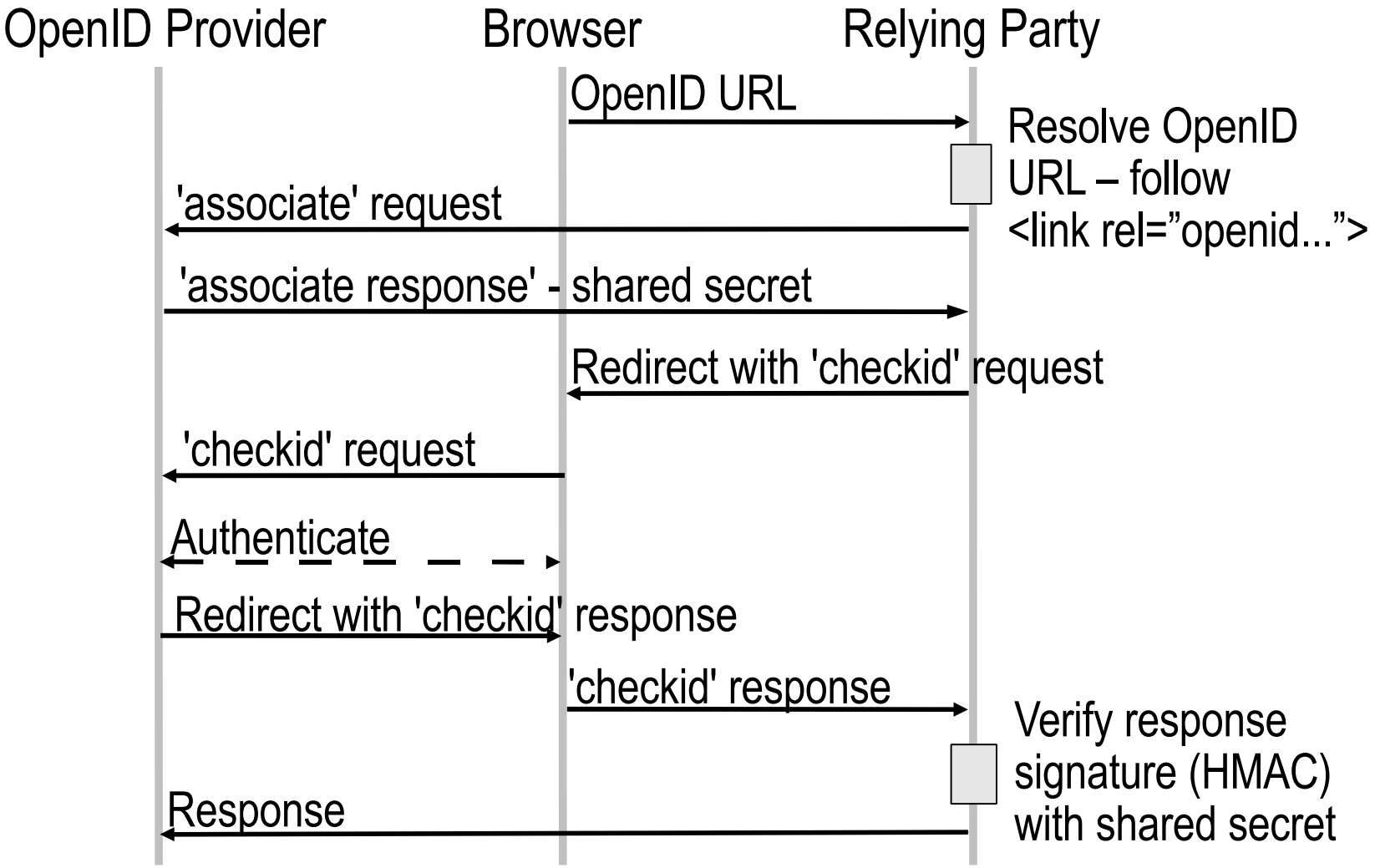
OpenID

- Simple decentralized authentication system
- No prior relationship assumed between OpenID Providers and Relying Parties
- Name-value pairs, rather than XML
- Assigns URLs or i-names to end users
  - Solves identity provider discovery problem, but...
  - In the absence of strong authentication, phishing is a real problem
  - End user acceptance of URL as an identifier is still in doubt

OpenID 1.x

- OpenID 1.0
  - 2005
  - Brad Fitzpatrick (LiveJournal/Six Apart)
- OpenID 1.1
  - 2006
  - David Recordon (Six Apart/Verisign/Six Apart)
- Simple Registration Extension
  - Common attribute request/response
    - Piggybacks on authentication request/response
    - Nickname, email address, full name etc

OpenID 1.1 Protocol



OpenID 2.0

- 'Real soon now' ☺
- Formalizes extension mechanism
  - OpenID Simple Registration Extension 1.1
  - OpenID Data Transport Protocol
    - Service Key Discovery
    - Messages
  - OpenID Attribute Exchange
  - OpenID Provider Authentication Policy Extension
- XRI i-names
- Yadis (Yet Another Distributed Identity System)
  - XRDS

XRDS

```
<XRDS ref="xri://=pat.patterson">
  <XRD>
    <CanonicalIDpriority="10">=!2A54.EB46.ED51.23F1</CanonicalID>
    <Service priority="10">
      <Typeselect="true">http://openid.net/signon/1.0</Type>
      <URI append="qxri" priority="2">http://2idi.com/openid/</URI>
      <URI append="qxri" priority="1">https://2idi.com/openid/</URI>
    </Service>
    <Service priority="5">
      <Typeselect="true">xri://+i-service*(+authn)*(+saml)*($v*1.0)</Type>
      <URI append="none" priority="10">http://amfm.example.com/</URI>
    </Service>
    <Service priority="10">
      <Type match="default"/>
      <Typeselect="true">xri://+i-service*(+contact)*($v*1.0)</Type>
      <Path match="null"/>
      <Path select="true">(+contact)</Path>
      <URI append="qxri" priority="1">http://2idi.com/contact/</URI>
    </Service>
  </XRD>
</XRDS>
```

OpenID Adoption

- OpenID Providers
  - Verisign PIP
  - AOL
  - Orange/France Telecom
  - Sun Microsystems :-)
- Relying Parties
  - Dopplr
  - Zooomr
  - Ma.gnolia

OpenID Adoption

- Applications
  - Drupal
  - Plone
  - DotNetNuke
  - Wordpress
- Libraries
  - Java, PHP, Perl, Python, Ruby, C++ etc etc etc
  - OpenSSO Extension
    - Java OpenID Provider

Cardspace

- AKA Infocard
- Microsoft Cardspace 1.0
  - Internet Explorer 7.0 - October 2006
  - Windows Vista - January 2007
- Smart client – the 'Identity Selector' – moves away from previous browser-centric models
- Based on WS-* stack, particularly WS-Trust
- Third-party implementations encouraged

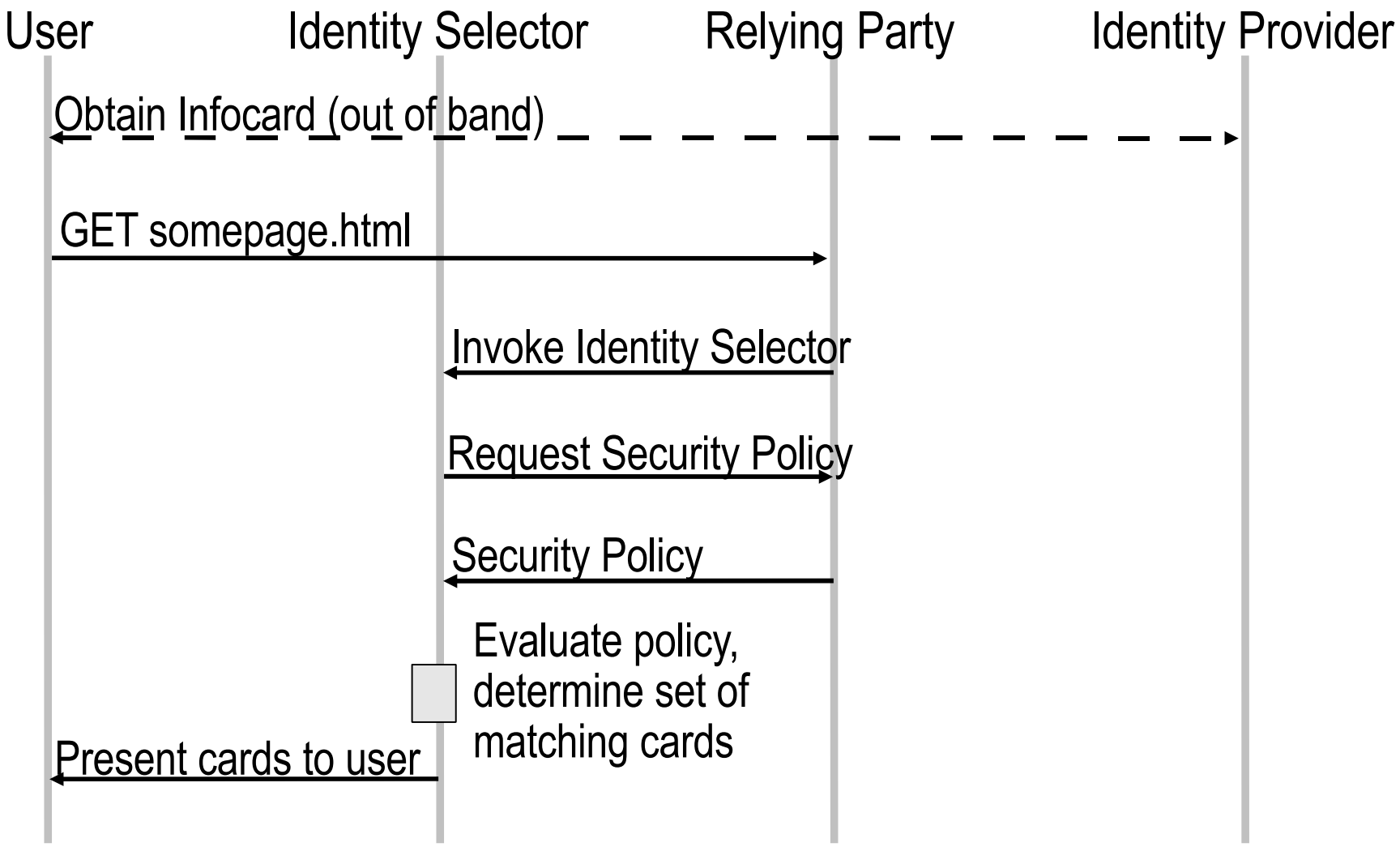
Cardspace Third-Party Implementations

- Identity Selectors
  - Higgins
    - Web-based, Client-based (DigitalMe), Eclipse-based
  - XMLDAP openinfocard
- Identity Provider
  - OpenSSO
  - Higgins
  - Verisign
  - XMLDAP
  - Bandit
  - Shibboleth
  - IBM, etc...

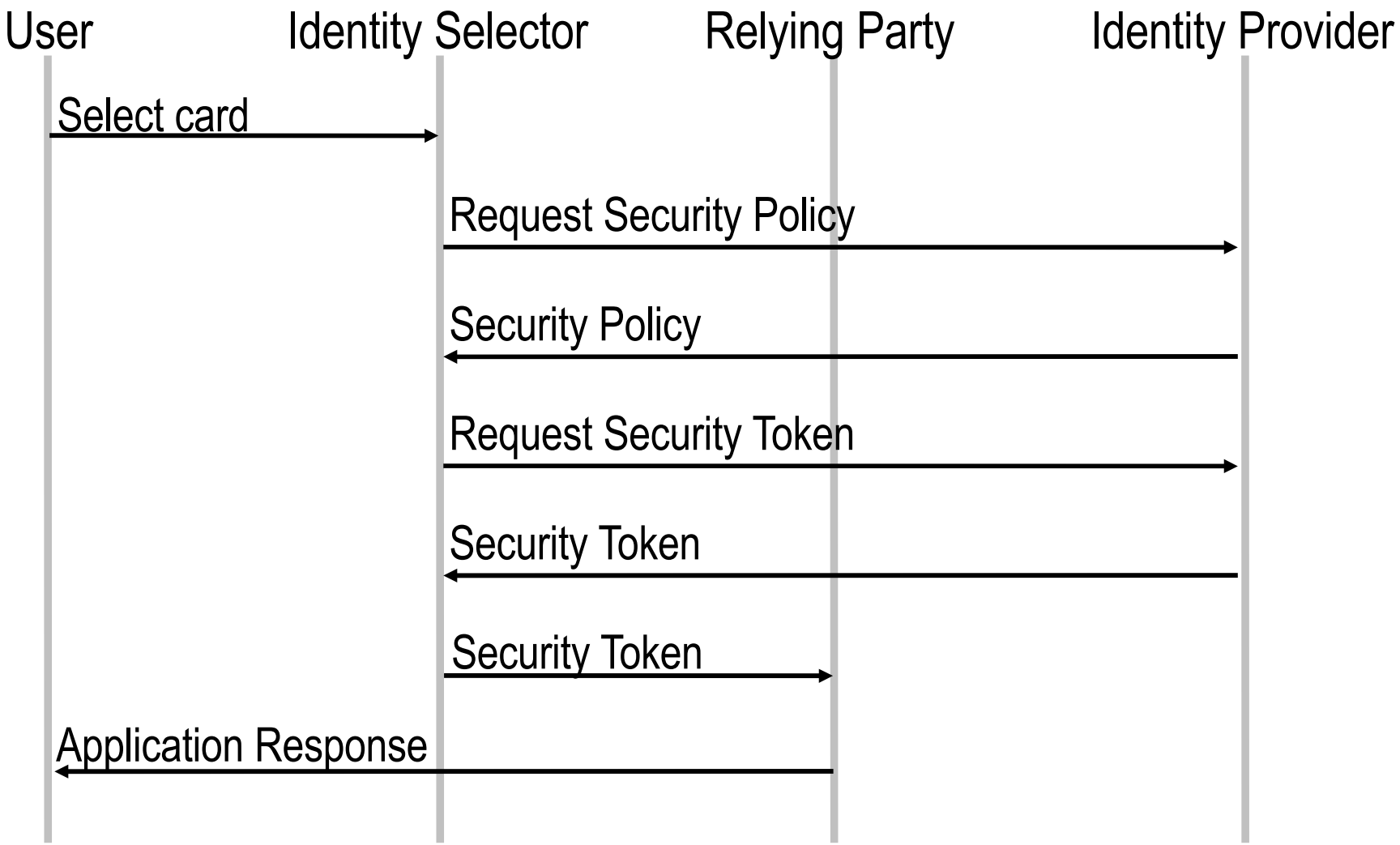
Cardspace Third-Party Implementations

- Relying Parties
  - Higgins
  - XMLDAP
  - Shibboleth
  - Pamela Project
  - Bandit
  - Ping Identity
  - Oracle
  - IBM
  - etc...

Cardspace Protocol



Cardspace Protocol (Continued!)



Cardspace Uptake

- Disappointing...
  - Kim Cameron's Blog (http://www.identityblog.com)
  - Microsoft Windows LiveID
  - ...?

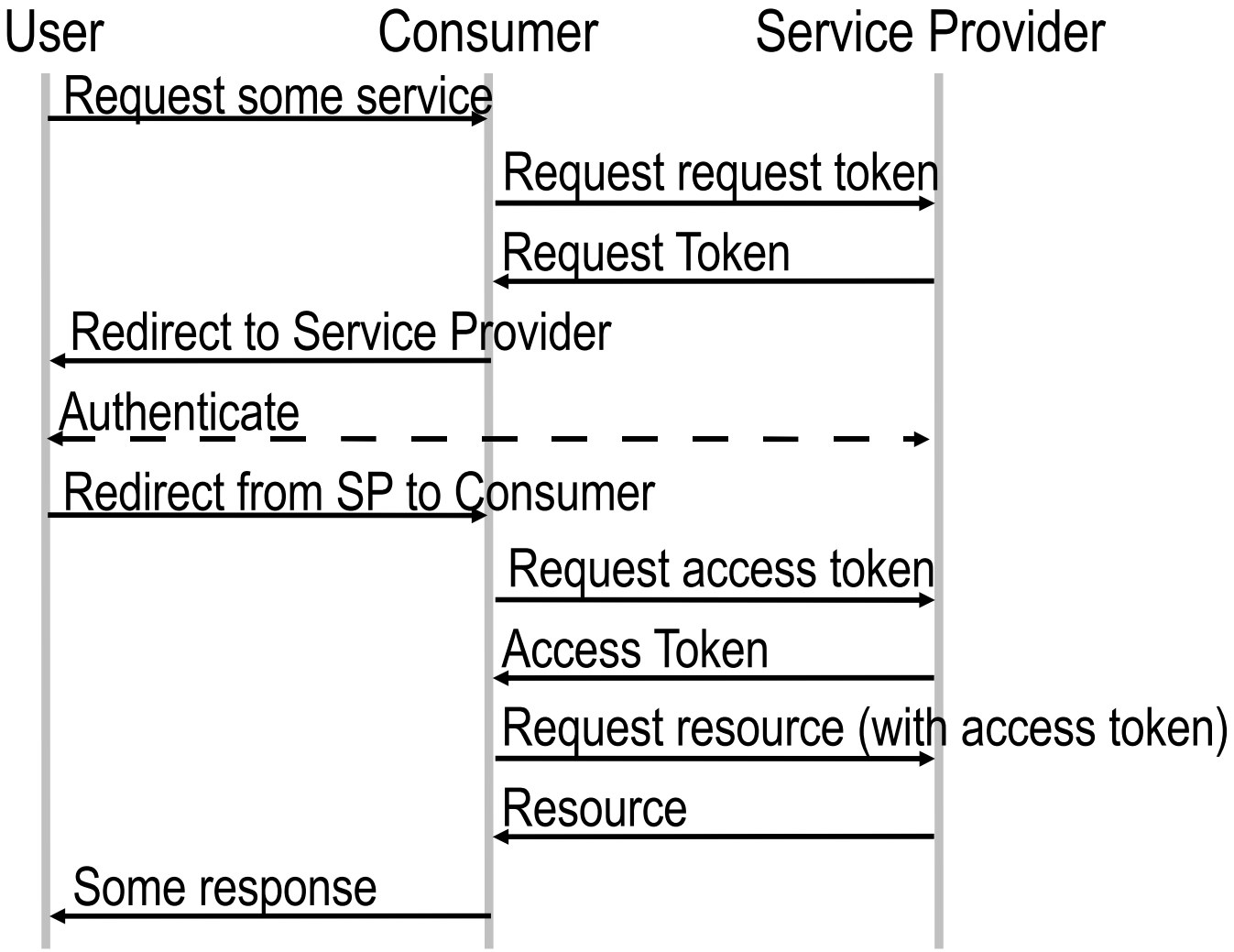
OAuth

- Version 1.0 'real soon now'
- Focuses on *authorization* rather than *authentication*
- Based on a raft of proprietary specs
  - Yahoo BBAuth
  - Google AuthSub
  - AOL OpenAuth
  - Flickr Auth API
- Wide participation
  - Twitter, Google, Pownce, Flickr, Ma.gnolia, Six Apart, Jaiku etc

OAuth Use Case

- “How do I authorize third-party services to access resources at my provider?”
  - Twitter mashups
  - Flickr photo services
  - Etc
- Least common denominator
  - OAuth can use HTTP headers, GET parameters, POST
  - PHP3 apps should be able to play!

OAuth Protocol



OAuth Adoption

- Early days (pre 1.0!), but...
- Test endpoints online
  - Twitter
  - Ma.gnolia
- Can expect spec participants to deploy

Concordia

- Not a protocol or even an organization as such
- More of a banner to rally beneath
  - Liberty Alliance
  - OpenID participants
  - Microsoft
- Customer-focused - “How do we get this stuff all working in the real world”
- Regular meetings colocated with identity events
- http://www.projectconcordia.org

Digital Identity from LDAP to SAML and beyond

November 8, 2007

Pat Patterson
Federation Architect
pat.patterson@sun.com
blogs.sun.com/superpat