Security Assertion Markup Language (SAML)

Massimo Nardone, TKK, S-38.153 Security of Communication Protocols

Agenda

- Introducing SAML
- Status of SAML and related standards efforts
- SAML concepts
- SAML Use Cases
- SAML Architecture: Assertions – Protocol – Bindings and Profiles
- SAML Scenarios
- Conclusion and Resources

What SAML is for?

- Distributed Authorization
- Federated Identity Management
- Multi-vendor Portals
- Web Services Access Control

SAML is NOT…

- A new form of Authentication
- Existing security “translated” into XML
- An alternative to WS-Security
- Limited to legacy applications
- Limited to Web Browser applications
- Limited to Web Services security

SAML Value Proposition

- XML Framework for exchanging security information over the internet. E.g. Assertions.
- These assertions about authentication and authorization are expressed as XML documents
- Standardization efforts carried out within Security Services Technical Committee at OASIS
- Based on merger of two competing security efforts viz. S2ML and AuthML
- Enables universal sharing of Authentication and Authorization information
- Platform neutral solution
- Security framework independent of Vendor implementation and architecture

SAML in the Security Puzzle

SAML is part of the XML-based security standards family

- **XML Encryption**: represent the encrypted content of XML data and the information that enables a recipient to decrypt it
- **XML Signature**: provide integrity, signature assurance, and non-repudiation for Web data
- **XKMS**: specify protocols for distributing and registering public keys (used in conjunction with XML Signature)
- **XACML**: provide a specification for policies to access XML documents, based on objects (elements to be accessed in the XML document), subject (the user), action (read, write, create, delete)
- **SPML**: Service Provisioning Markup Language for exchanging user, resource, and service provisioning information

SAML: Industry Traction

Used in security services implementation of Internet2

- Sun (Network Identity/iPlanet DSAME)
- Entrust (GetAccess portal)
- Systinet (WASP Secure Identity)
- Securant (RSA Cleartrust)
- Entegrity (AssureAccess)
- Netegrity (AffiliateMinder)

SAML Use Cases

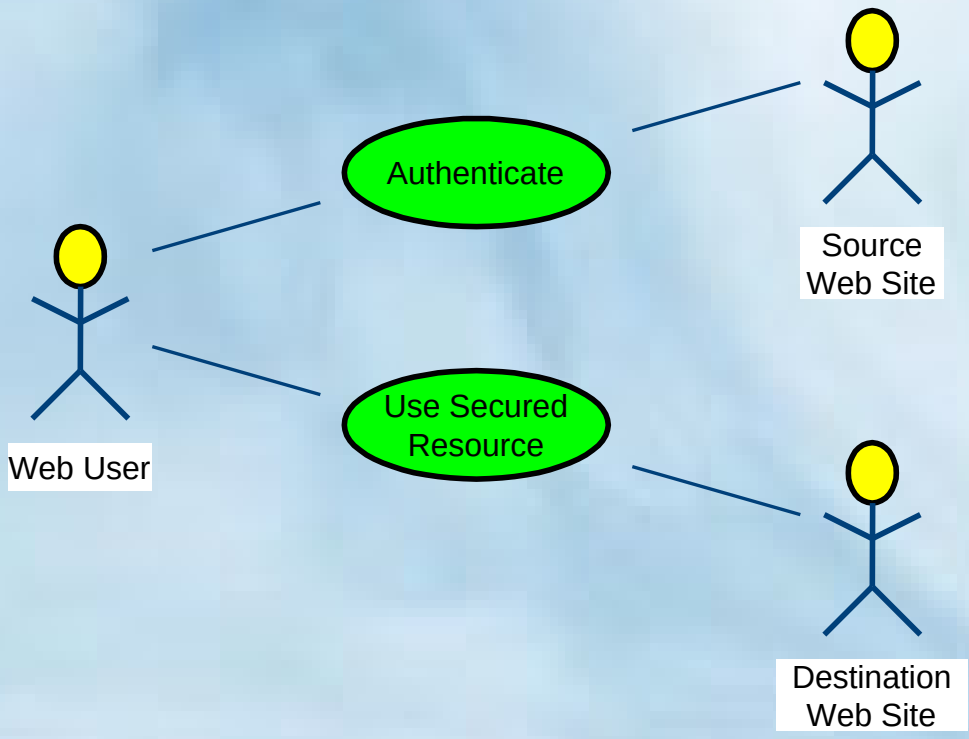
- SAML developed three “use cases” to drive its requirements and design:

  1. Single sign-on (SSO)
  2. Distributed transaction
  3. Authorization service

- Each use case has one or more “scenarios” that provide a more detailed roadmap of interaction

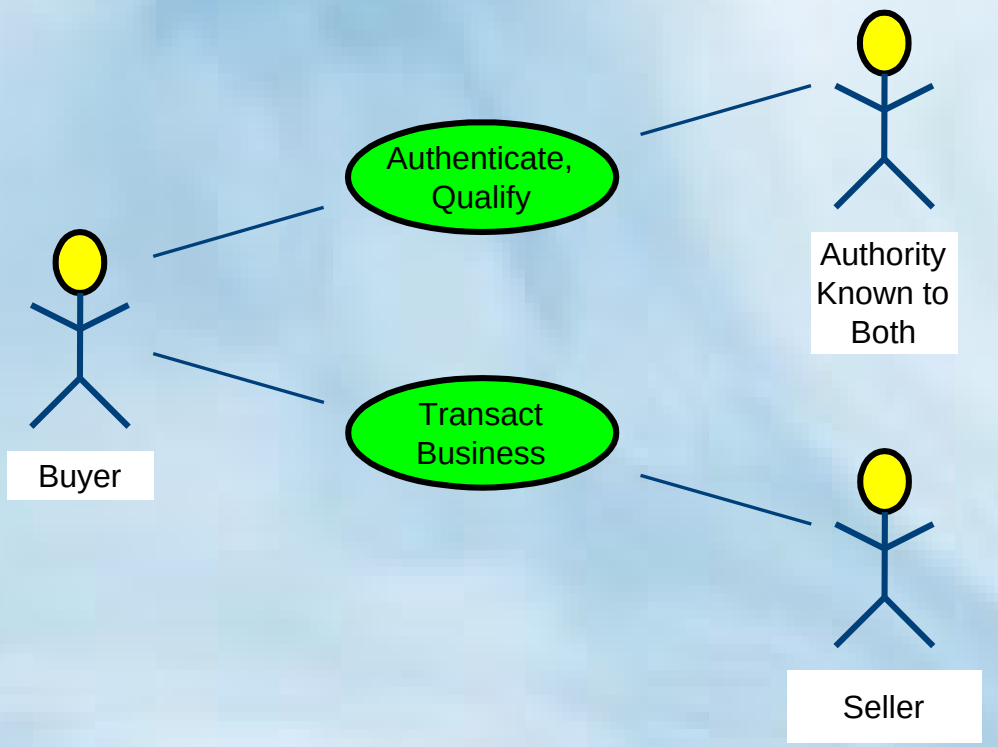
#1: Single sign-on (SSO)

- Logged-in users of site1 are allowed access to sister site2



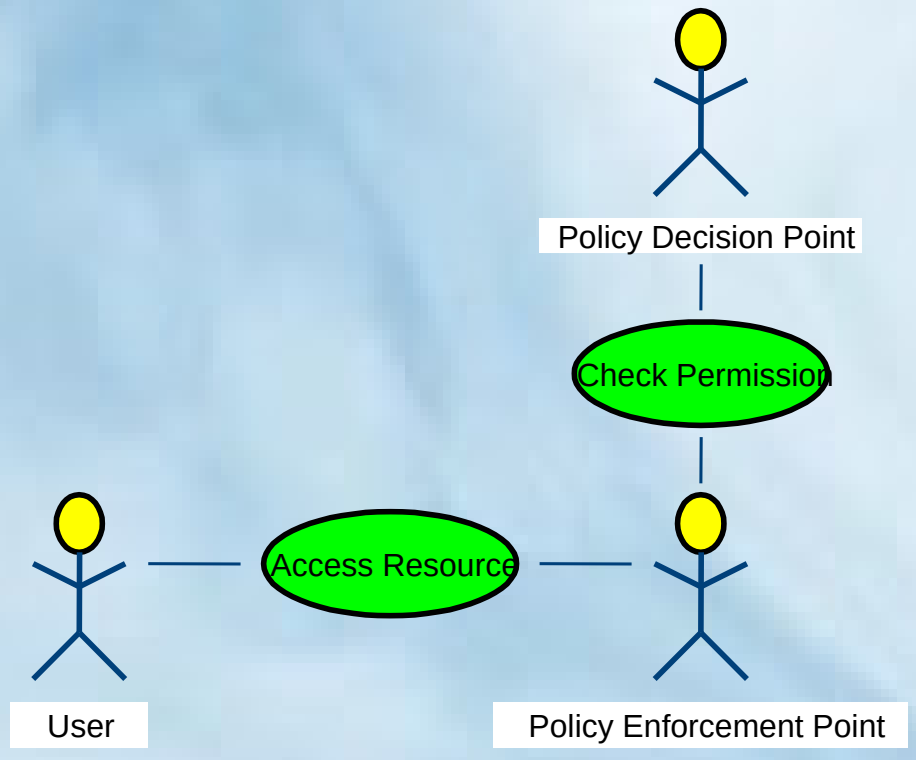
#2: Distributed transaction

- Employees at site1 are allowed to order goods from site2 if they are authorized to spend enough



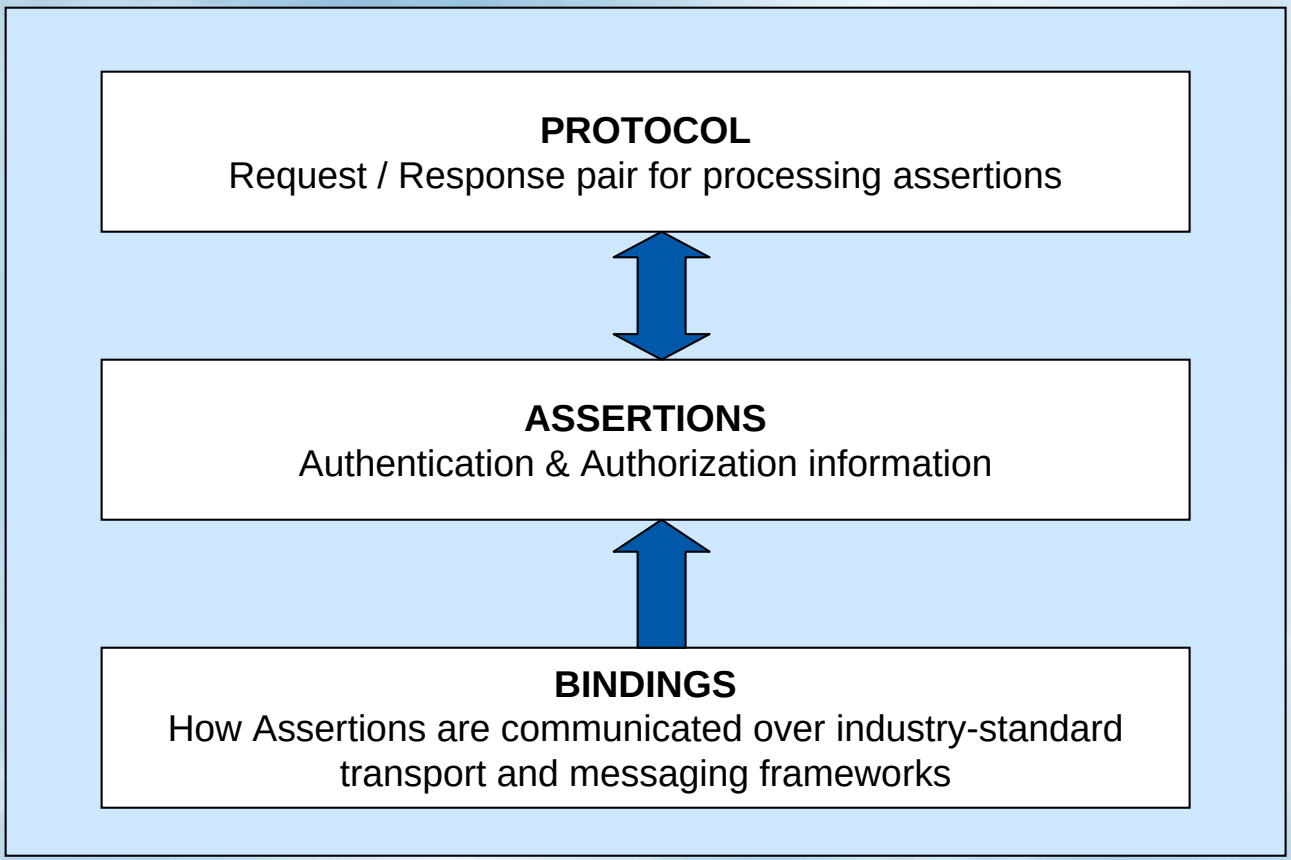
#3: Authorization service

- Employees at site1 order goods directly from site2, which performs its own authorization

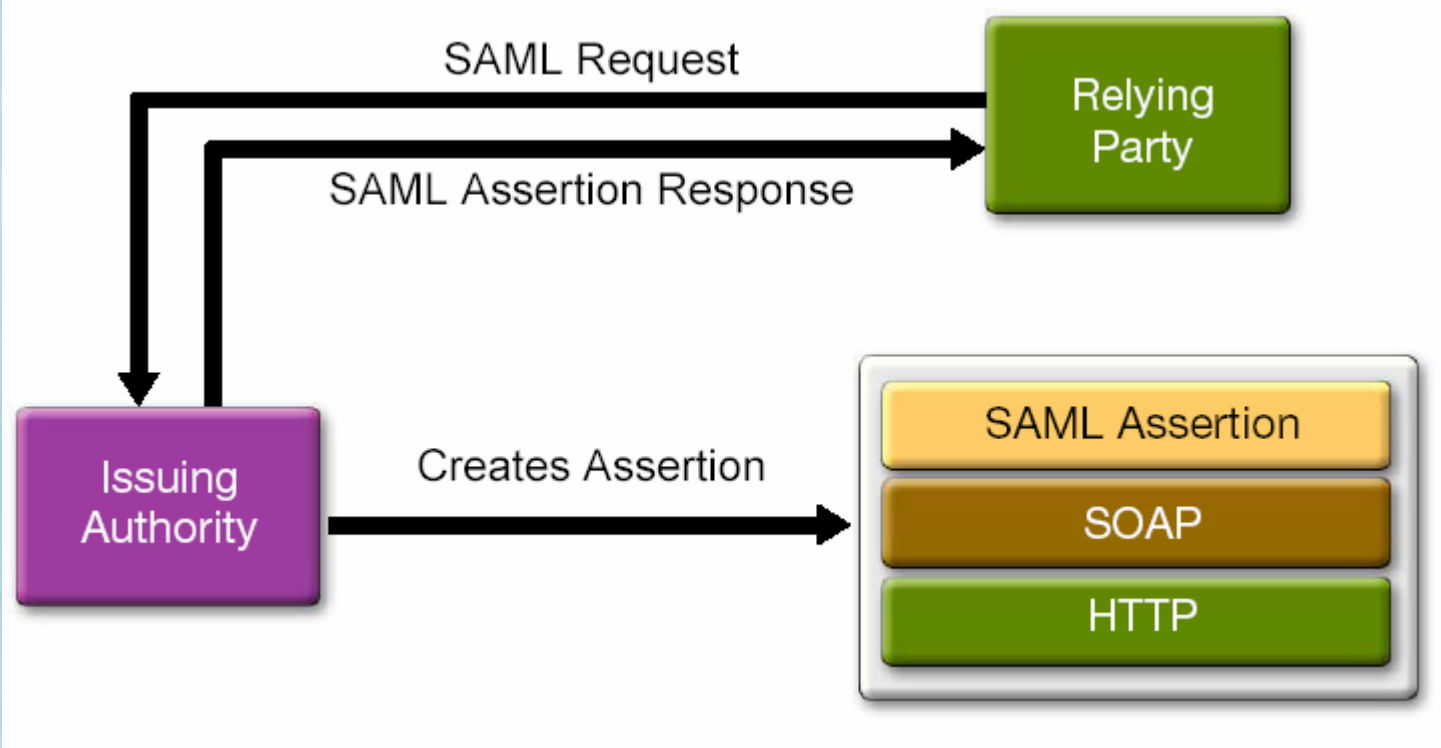


SAML – Architecture 1/2

The SAML specification includes three distinct parts: assertions, protocol, and bindings.



SAML Architecture 2/2



SAML: Possible Issuing Authorities

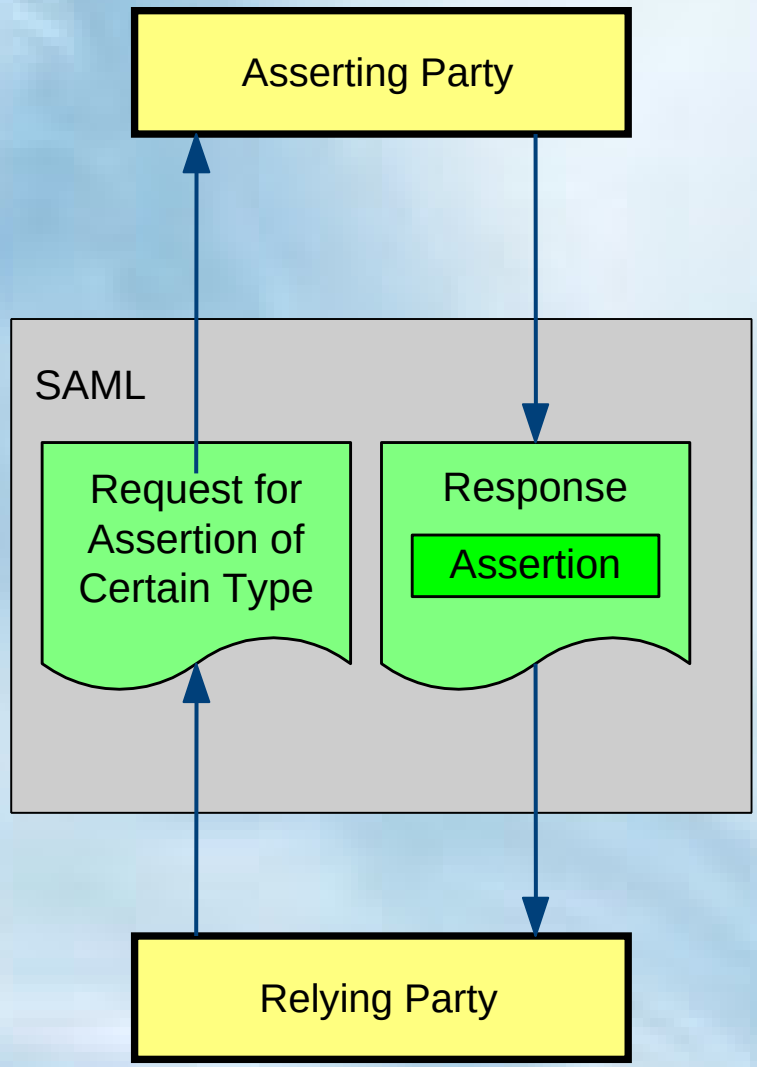
Third-party Security Services Providers:

- Microsoft for its Passport initiative
- XNSORG for its Web Identity Platform
- DotGNU for its Virtual Identity Platform

SAML assertions

- Assertions are declarations of fact, according to someone
- SAML assertions are compounds of one or more of three kinds of “statement” about “subject” (human or program):
  - Authentication
  - Attribute
  - Authorization decision
- You can extend SAML to make your own kinds of assertions and statements
- Assertions can be digitally signed (XML-DSIG)

SAML protocol for getting assertions



SAML assertions: type of information

- Basic information
- Claims
- Conditions
- Advice

SAML assertions: type of information

At a minimum, the information consists of the following elements:

- Issuer and issuance timestamp
- Assertion ID
- Subject
  - Name plus security domain
  - Optional subject confirmation (e.g., public key)
- Conditions (under which an assertion is valid)
  - Assertion Validity Period
    - NotBefore and NotOnOrAfter
  - Audience restrictions
  - Target restrictions
  - Application-specific conditions
- Advice (Additional information on how an assertion was made provided by issuing authority)

SAML assertions: an Example

```
<saml:AssertionList xmlns:saml="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-assertion-16.xsd"
  MajorVersion="1" MinorVersion="0" AssertionID="10.20.30.40.1234567890" Issuer="TKK" IssueInstant="2002-08-05T13:14:15Z"
  >

  <saml:Conditions NotBefore="2002-08-05T13:09:15Z" NotOnOrAfter="2002-08-05T13:24:15Z" >
  <saml:Audience> TKKPartnerShipAgreementOf2002</saml:Audience>
  </saml:Conditions>

  <saml:AuthenticationStatement AuthenticationMethod="Password" AuthenticationInstant="2002-08-05T13:14:15Z" >
      <saml:Subject><saml:NameIdentifier SecurityDomain="www.tkk.com" Name="mnardone" /></saml:Subject>
      <saml:AuthenticationLocality IPAddress="10.20.30.40" DNSAddress="tkk12345" />

  </saml:AuthenticationStatement>

  <saml:AttributeStatement>
            <saml:Subject>
                <saml:NameIdentifier SecurityDomain="www.tkk.com" Name="mnardone" />
            </saml:Subject>
        <saml:Attribute attributeName="PartnerBlock"AttributeNamespace="http://www.pt.com/ns" >
          <saml:AttributeValue>
            <Partner><Company>TKK</Company>
            <SubscriptionCategory>Premium</SubscriptionCategory><AccessLevel>10</AccessLevel>
            </Partner>
          </saml:AttributeValue>

        </saml:Attribute>

   </saml:AttributeStatement>

</saml:AssertionList>
```

SAML Bindings and Profiles

- This is where SAML itself gets made secure
- A “binding” is a way to transport SAML requests and responses
  - SOAP-over-HTTP binding is a baseline
  - Other bindings will follow, e.g., raw HTTP
- A “profile” is a pattern for how to make assertions about other information
  - Two browser profiles for SSO: artifact and POST
  - WS-Security profile for securing SOAP payloads

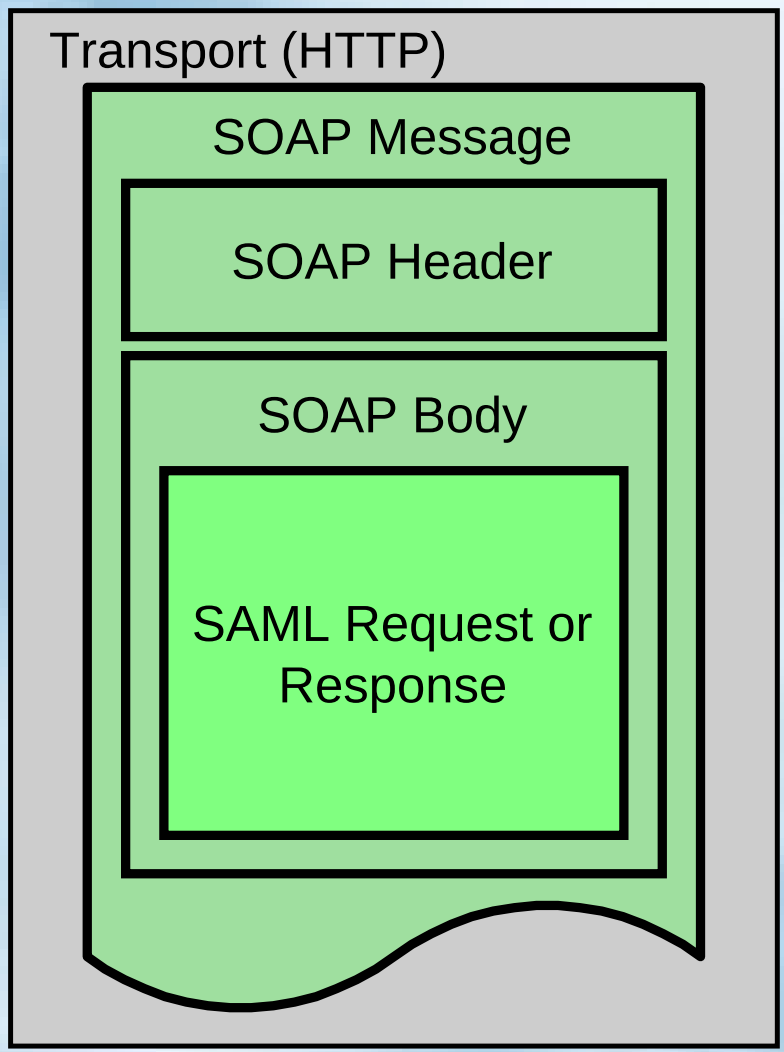
SAML Bindings

SAML protocol bindings specify how SAML request/response message exchanges are mapped to standard messaging protocols.

The currently available bindings are:

- SAML HyperText Transport Protocol (HTTP) Binding:
  Describes how SAML request/response message exchanges are mapped into HTTP message exchanges.
- SAML Simple Object Access Protocol (SOAP) Binding
  Describes how SAML request/response message exchanges are mapped into SOAP message exchanges.

The SOAP-over-HTTP binding



SAML Profiles

SAML Profiles specify how SAML assertions are inserted in and extracted from a message framework or protocol.

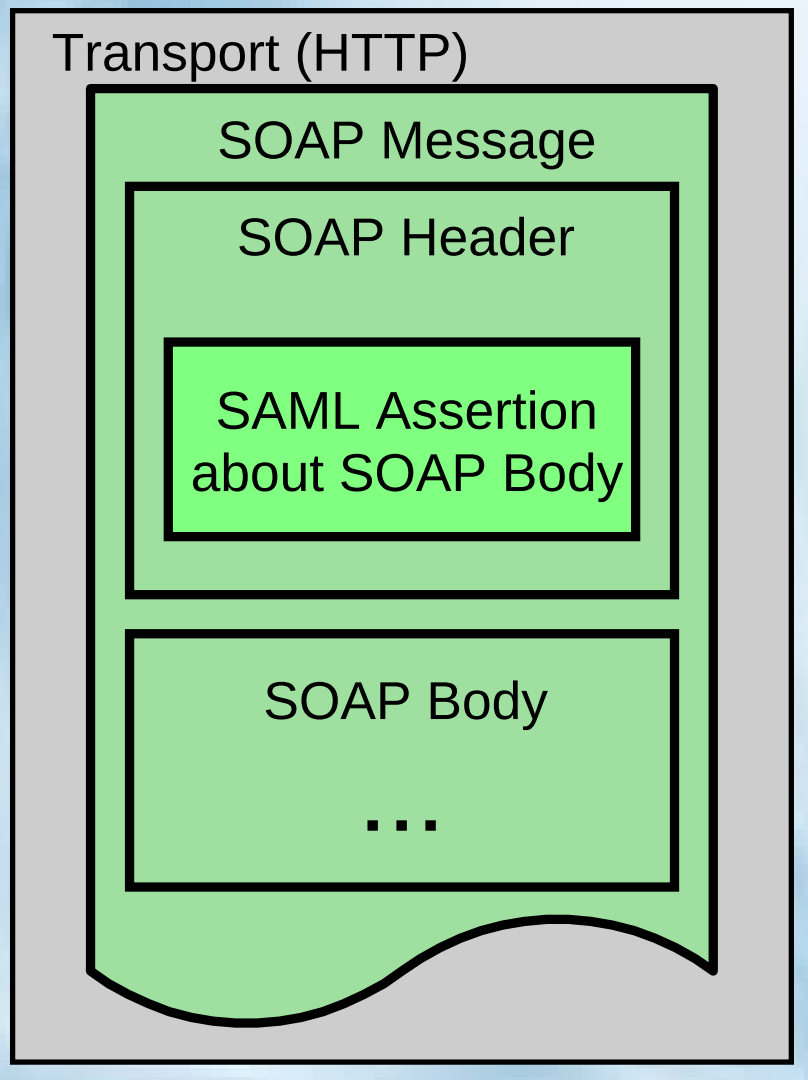
The currently available profiles are:

- Web Browser Profiles of SAML
- SOAP Profile of SAML

Web Browser Profiles of SAML

- **Pull Profile**
  - SAML artifacts are passed between sites by the browser on URL query strings
  - Assertions are “pulled” by destination site from source site
- **Push Profile**
  - HTML forms include SAML assertions “pushed” to destination site

SAML Web Services Profile

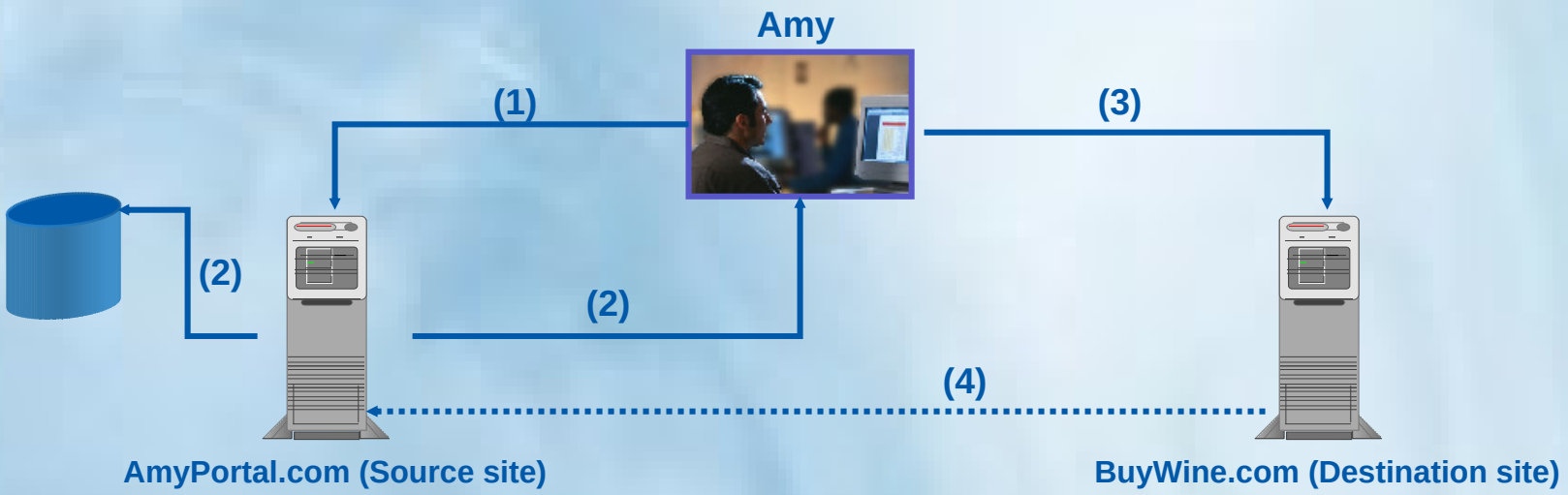


SAML Scenarios

SAML scenarios are based on the 3 SAML use cases described earlier

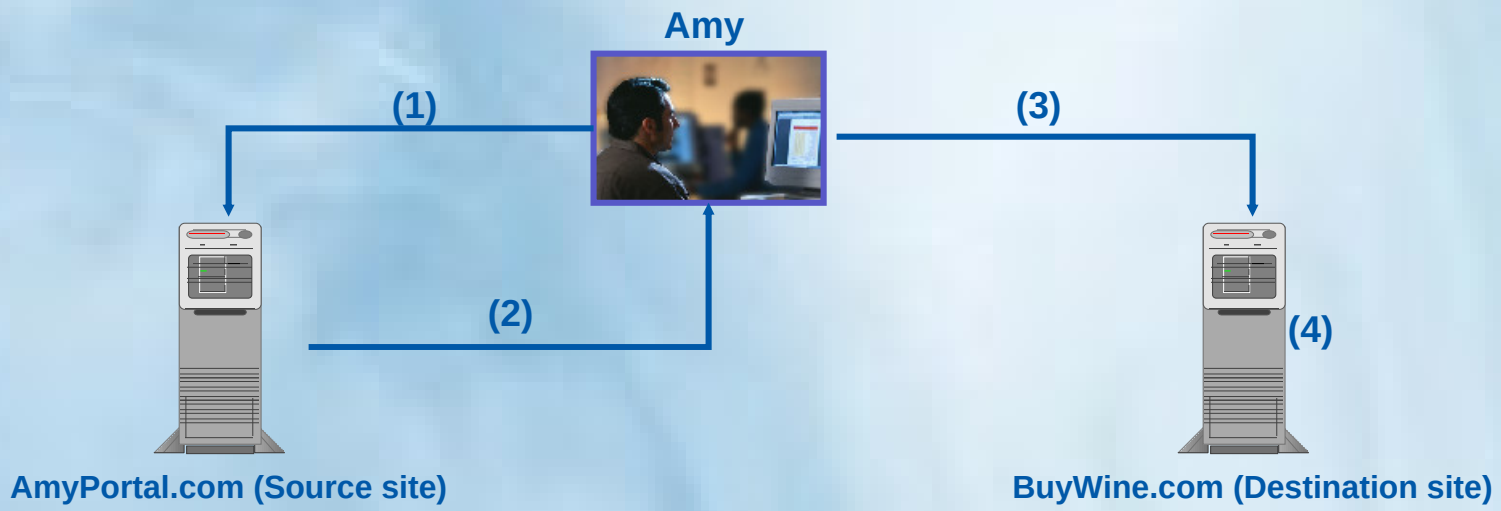
- Browser-Driven Interaction (SSO)
  - Pull
  - Push
- Remote Authorization
- XML Message Transfer (Back office transaction)

SAML Scenarios: Browser-Driven Interaction (Pull)



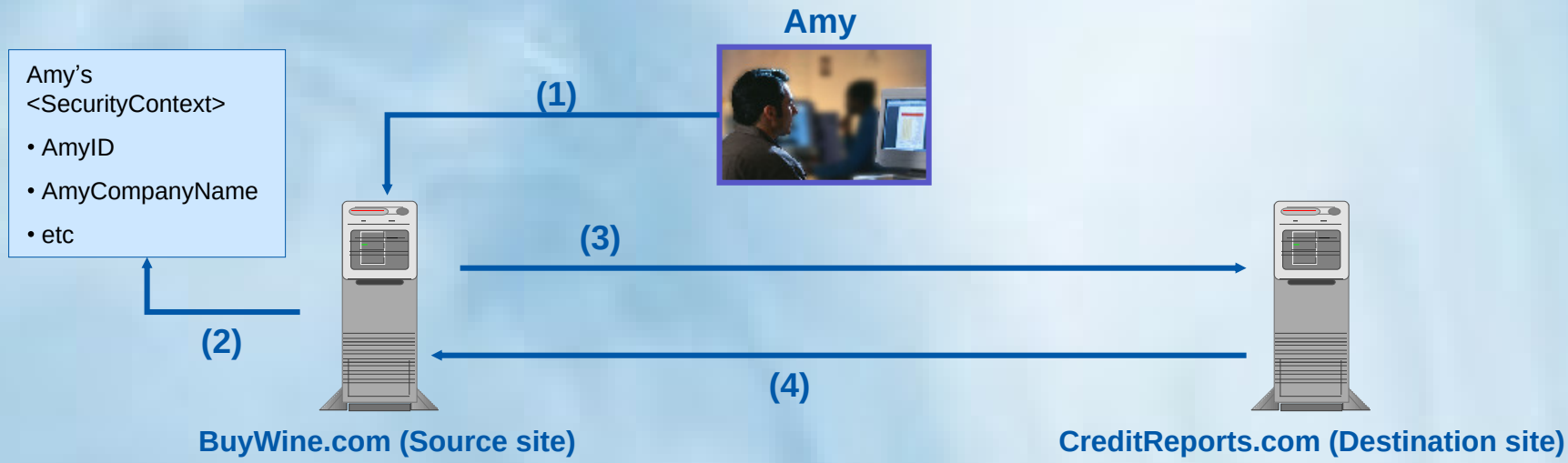
1. Amy authenticates at AmyPortal.com (Source site).
2. AmyPortal.com creates a security context (SAML assertions) for Amy, stores it (stateful server) and produces a small, fixed-size (8-byte) SAML artifact for these assertions.
3. Amy visits BuyWine.com (Destination site). The URL line includes the destination site, the SAML artifact, and the target resource (via HTTP over server-side SSL), e.g., https://buywine.com?SAMLartifact=8B-hexNumber&target=buy-wine/mywine.html
4. BuyWine.com uses the SAML artifact information to “pull” the full assertions describing Amy (AuthN and AuthZ) from AmyPortal.com and makes the access decision. If authorized, Amy is not challenged again to access resources at BuyWine.com (SSO)

SAML Scenarios: Browser-Driven Interaction (Push)



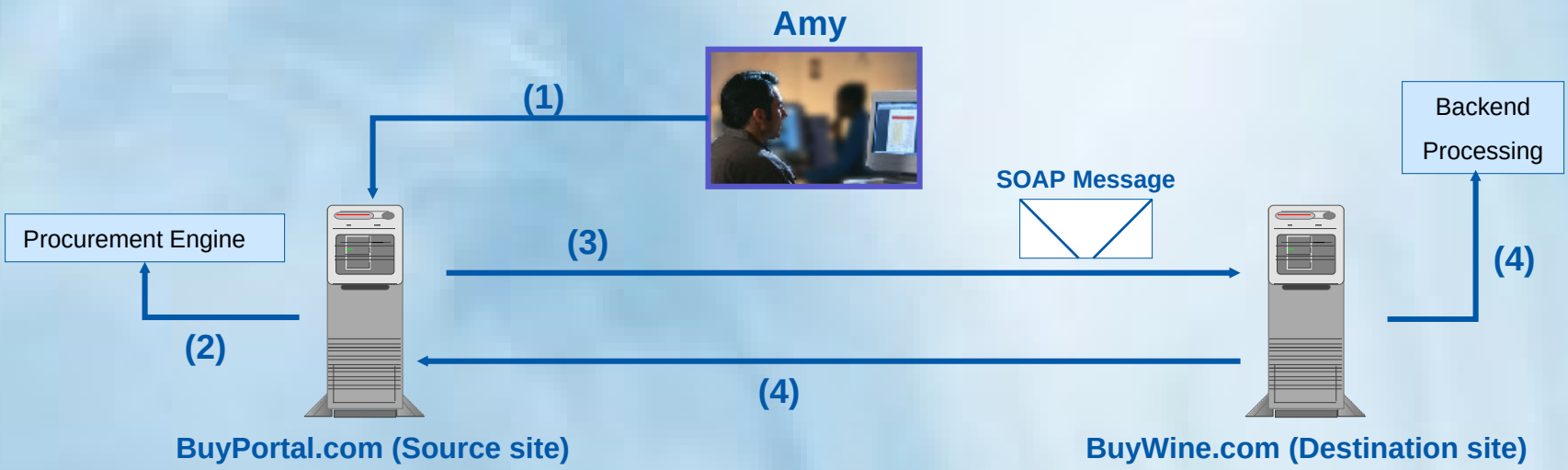
1. Amy authenticates at AmyPortal.com (Source site).
2. AmyPortal.com creates a security context (I.e. generates a digital-signed SAML assertions) for Amy, and includes the security context together with target information in an HTML form.
3. Amy submits the HTML form, thus “pushing” the security context to BuyWine.com (HTTP POST).
4. BuyWine.com reads the POSTed security context and target information, and then makes an access decision.

SAML Scenarios: Remote Authorization



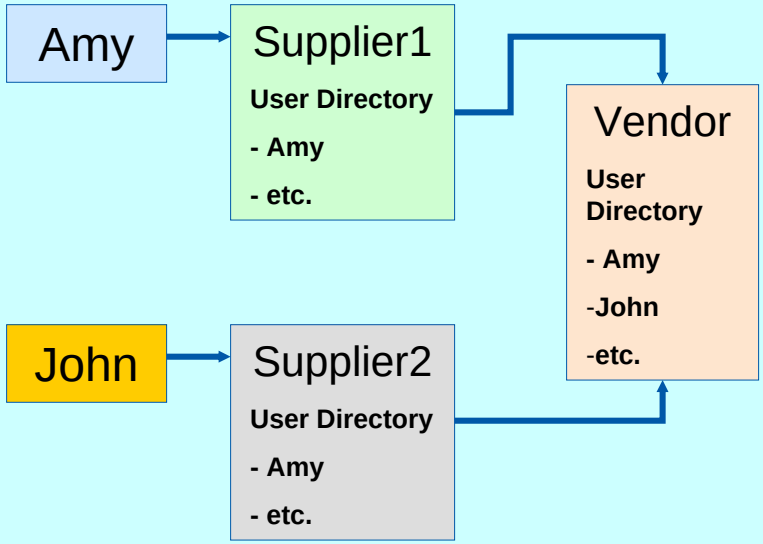
1. Amy wants to buy wine from BuyWine.com
2. BuyWine.com creates a SAML security context for Amy.
3. BuyWine.com wants to know Amy's company's financial status, risk of non-payment, etc. BuyWine.com sends a request to CreditReports.com to know whether Amy and the company he works for have the appropriate rating. BuyWine.com's request includes the SAML security context for Amy and his company, plus a SAML query for rating.
4. CreditReports.com sends a SAML response to BuyWine.com

SAML Scenarios: XML Message Transfer

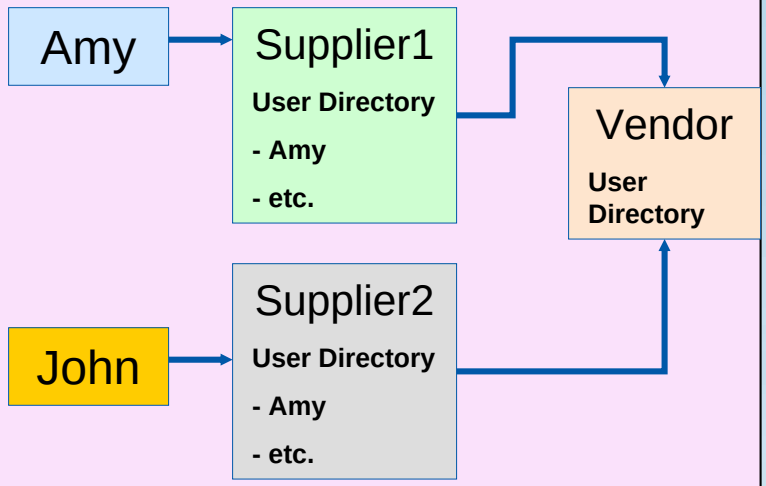


1. Amy wants to buy wine from BuyWine (Destination site) through the BuyPortal(Soruce site). Amy authenticates (out of band) at BuyPortal.com and invokes a procurement engine to fill out a purchase order (PO).
2. BuyPortal.com creates the security context for the PO by inserting SAML assertions in the SOAP header of the SOAP message that carries the PO as payload.
3. BuyPortal.com sends the document to BuyWine.com.
4. BuyWine.com opens the incoming SOAP document, reads the SAML assertions included in the SOAP header, processes the PO, and sends a PO response (out of band).

SAML Benefits



In this model, Suppliers must synchronize their user directories with the Vendor's because user information must be maintained at each site involved in the interaction. This means costly, often-repeated duplication processes.



In the SAML model, no directory duplication or synchronization is necessary. Security information "travels" with Amy and John. Neither Amy nor John is known by the Vendor. The Vendor uses SAML Assertions generated for Amy and John by their respective companies. User information does not need to be maintained at each site involved in the interaction.

For more information…

- **Securing Web Services Whitepaper: http://members.netegrity.com/access/files/TransactionMinder.pdf**
- **SAML: http://www.oasis-open.org/committees/security**
- **JSAML Toolkit: http://www.netegrity.com/products**
- **JSAML Whitepaper: http://www.netegrity.com/files/JSAMLwhitepaper.pdf**
- **XML Sig: http://www.w3.org/Signature**
- **XML Encryption: http://www.w3.org/Encryption/2001**
- **XKMS: http://www.w3.org/2001/XKMS**
- **Microsoft Passport: http://www.passport.com**
- **Liberty Alliance project: http://www.projectliberty.org**

Questions?