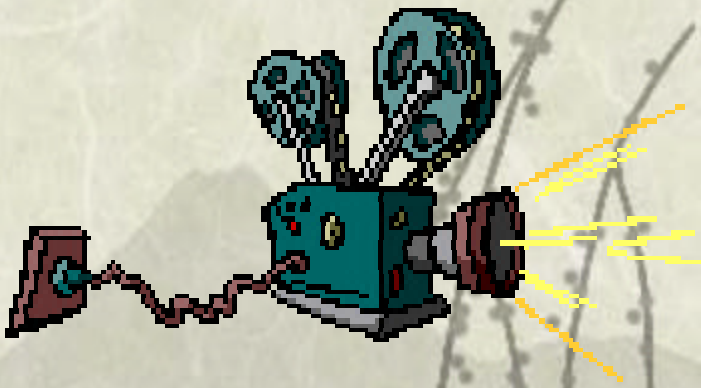
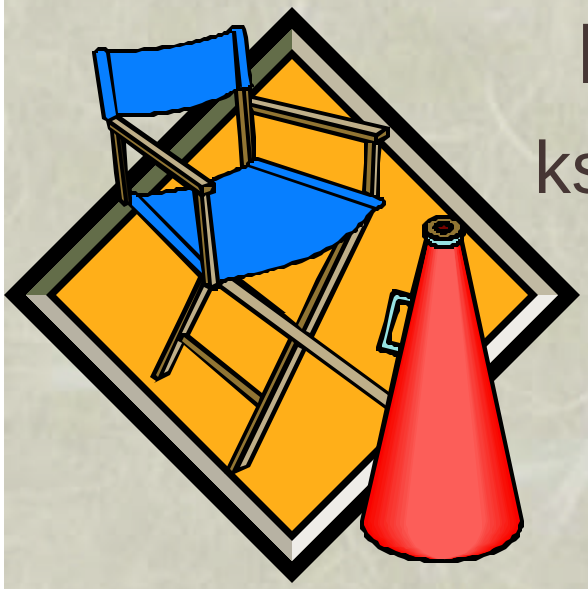


Next Generation Security Substrate Architecture - A Perspective

Krishna Sankar

ksankar@cisco.com

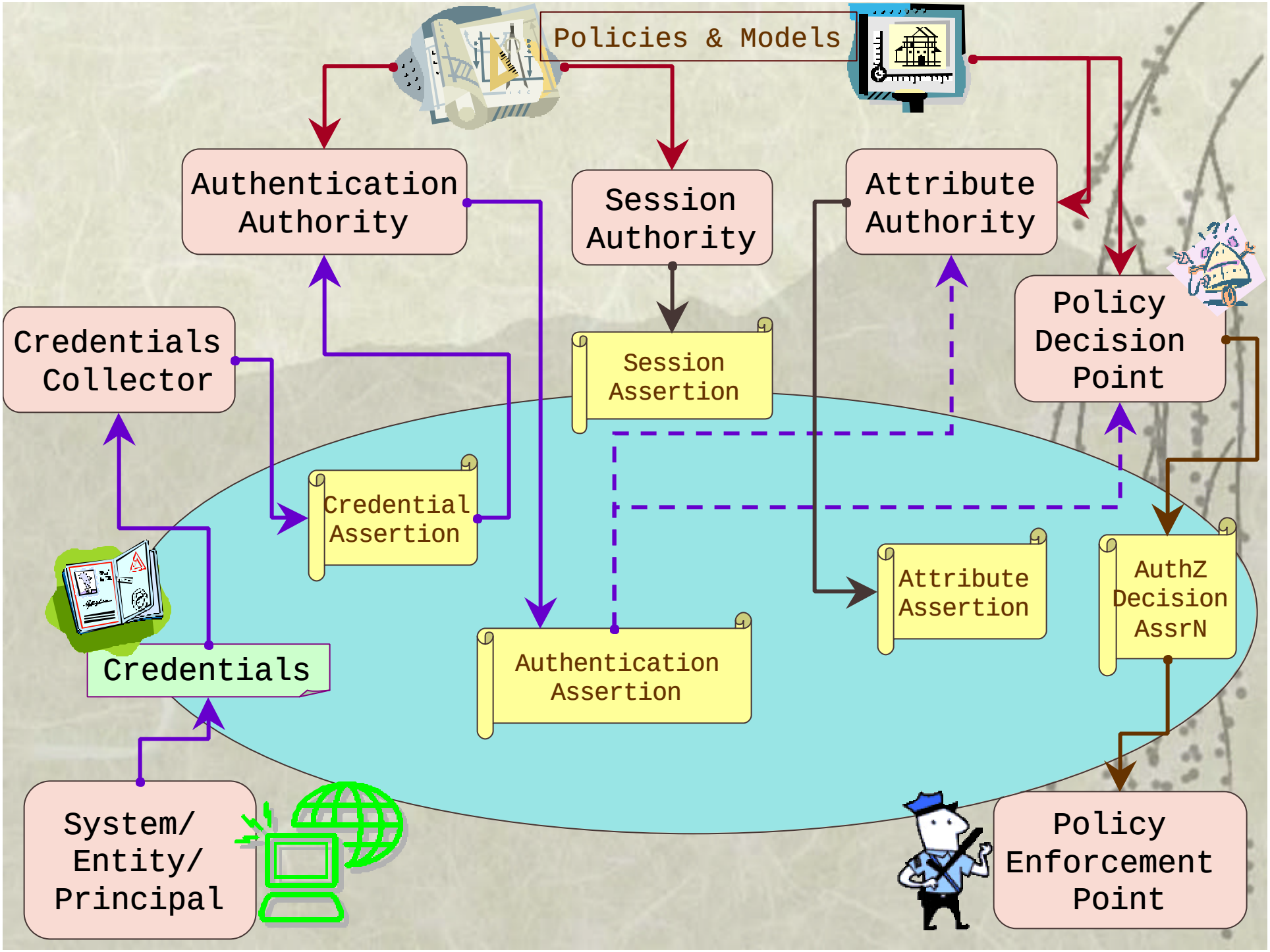


What is SAML ?

- XML based Framework
- A set of XML vocabularies for :
  - Authentication Assertion
  - Attribute Assertion
  - Session Assertion (Future)
  - Credential Assertion (Future)
  - So that data traveling on the wire is standardized
- A standard message exchange protocol
  - Clarity in orchestrating how you ask for and get the information you need
- Rules for how the messages ride “on” and “in” transport protocols
  - For better interoperability

What problems does SAML try to solve?

- Permissions management data is shared in mostly proprietary ways
  - Integrating new security features may require developing a lot of new code
  - The different systems that generate and use security data are very tightly coupled
- Web-based applications show the need for more federation
  - We need to cross domains more easily


Policies & Models
Authentication Authority
Session Authority
Attribute Authority
Policy Decision Point
Credentials Collector
Session Assertion
Credential Assertion
Attribute Assertion
AuthZ Decision AssrN
Credentials
Authentication Assertion
System/ Entity/ Principal
Policy Enforcement Point

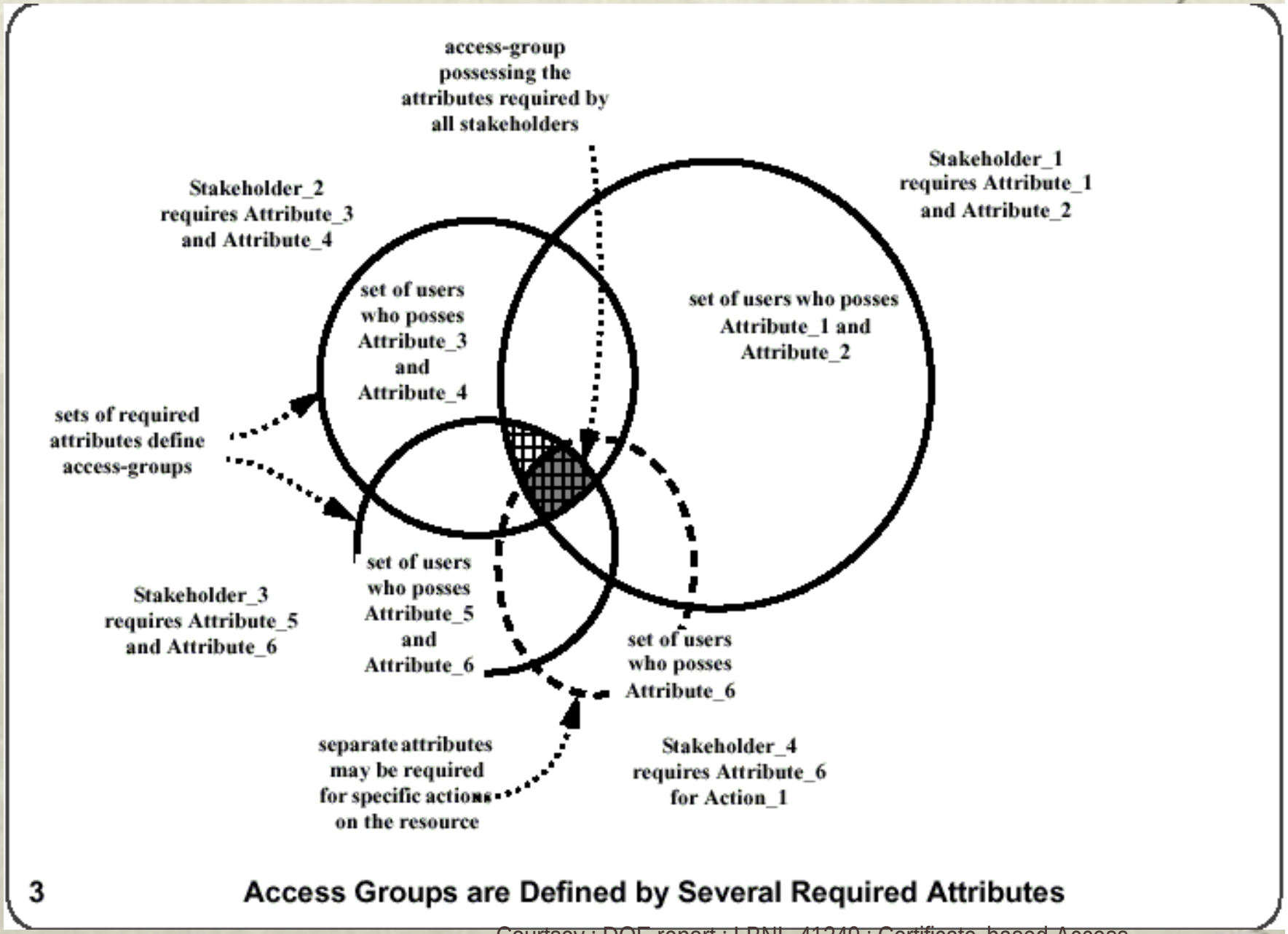
This model is conceptual only

- In practice, multiple kinds of authorities may reside in a single software system
  - SAML allows, but doesn't require, total federation of these jobs
- Also, the arrows may not reflect information flow in real life
  - Information can be pulled or pushed
  - Not all assertions are always produced
  - Not all potential consumers (clients) are shown

Policy (& Role) Based Visibility

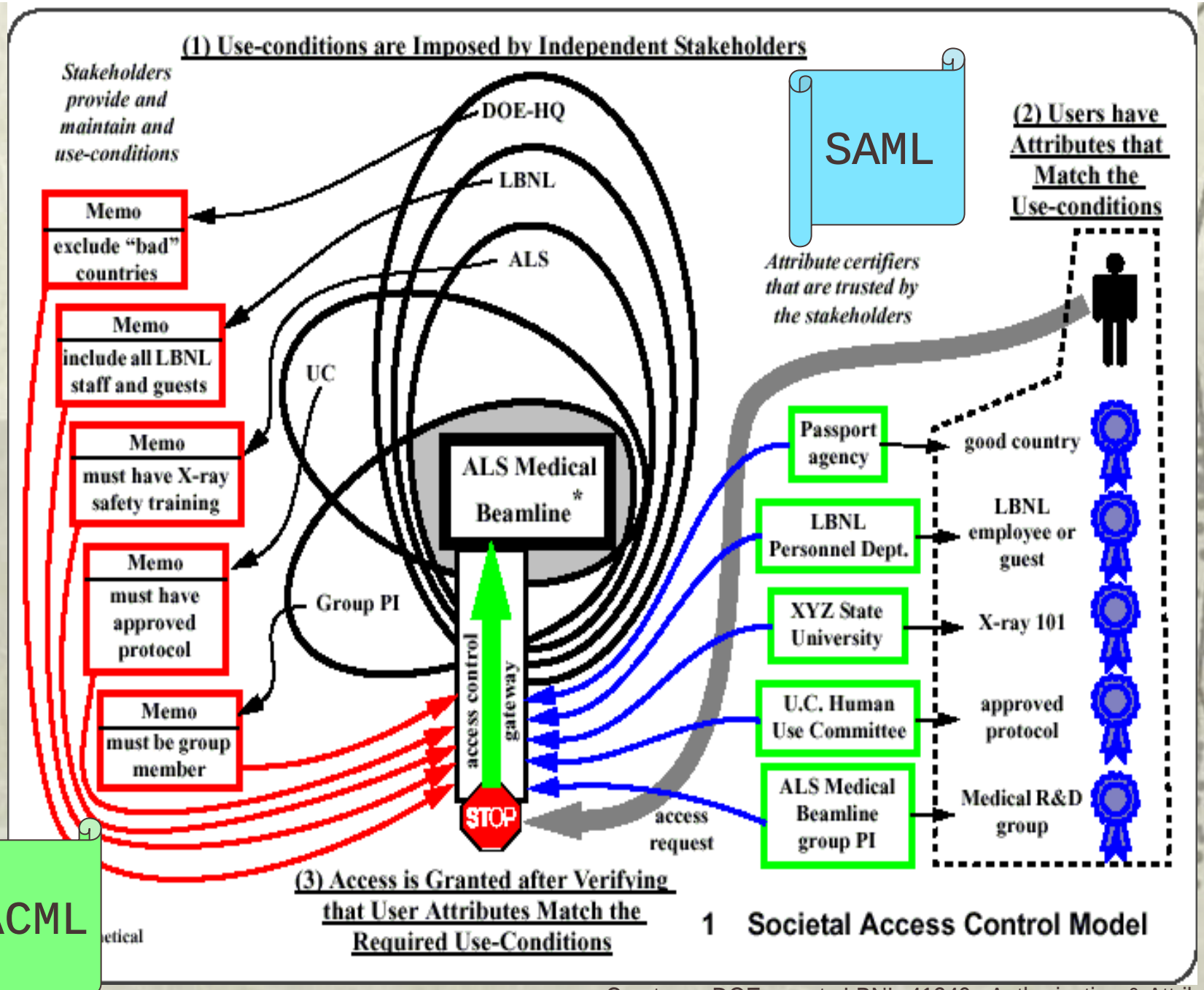
- Collaboration across multiple, independent and geographically dispersed stakeholders
- Stakeholders able to enforce policies even when controlled by different administrative domains
  - Traditional : ACLs
    - Cannot scale. Cause too many errors
- Multiple layers of management would impose restrictions

Courtsey : DOE report : LBNL-42928 : Certificate-based Access Control for widely distributed resources



3 **Access Groups are Defined by Several Required Attributes**

Courtsey : DOE report : LBNL-41349 : Certificate-based Access Control for widely distributed resources



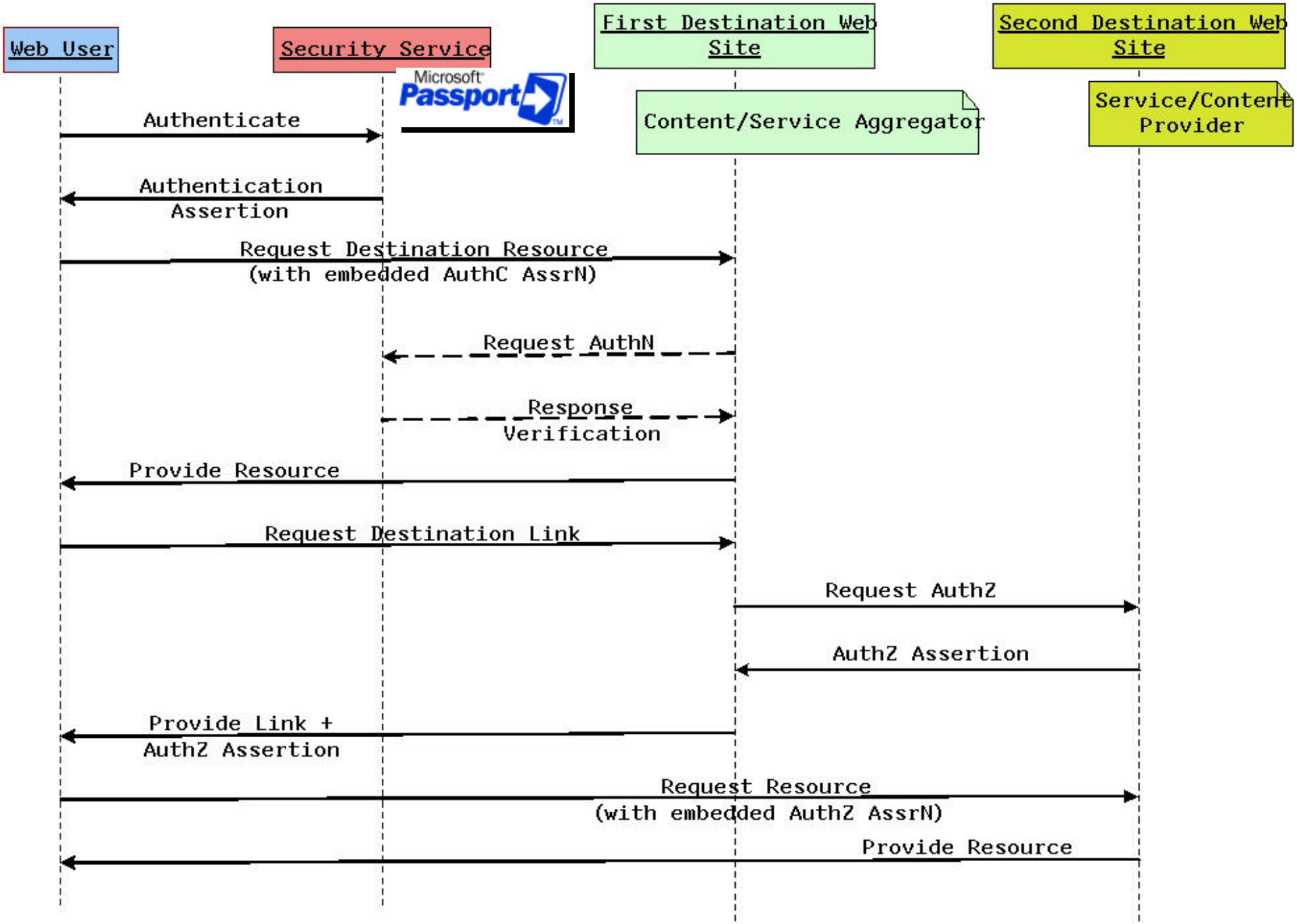
Courtesy : DOE report : LBNL-41349 : Authorization & Attribute Certificates for Widely Distributed Access Control

Types Of Security Mechanisms

- Stored Password
  - Good for internal systems
- Credential-pass-through
  - More secure (in the sense that the authC is known at all times)
- Role based Access Control
  - Good mechanism.
  - Requires business process to define roles

Single Sign-on, Third-Party Security Service

- Browser based single-sign-on
- Applications :
  - Eco Systems
  - Consortiums
  - AOL/Passport/LA based applications


Web User
Security Service
Microsoft Passport
First Destination Web Site
Content/Service Aggregator
Second Destination Web Site
Service/Content Provider
Authenticate
Authentication Assertion
Request Destination Resource (with embedded AuthC AssrN)
Request AuthN
Response Verification
Provide Resource
Request Destination Link
Request AuthZ
AuthZ Assertion
Provide Link + AuthZ Assertion
Request Resource (with embedded AuthZ AssrN)
Provide Resource

Article in

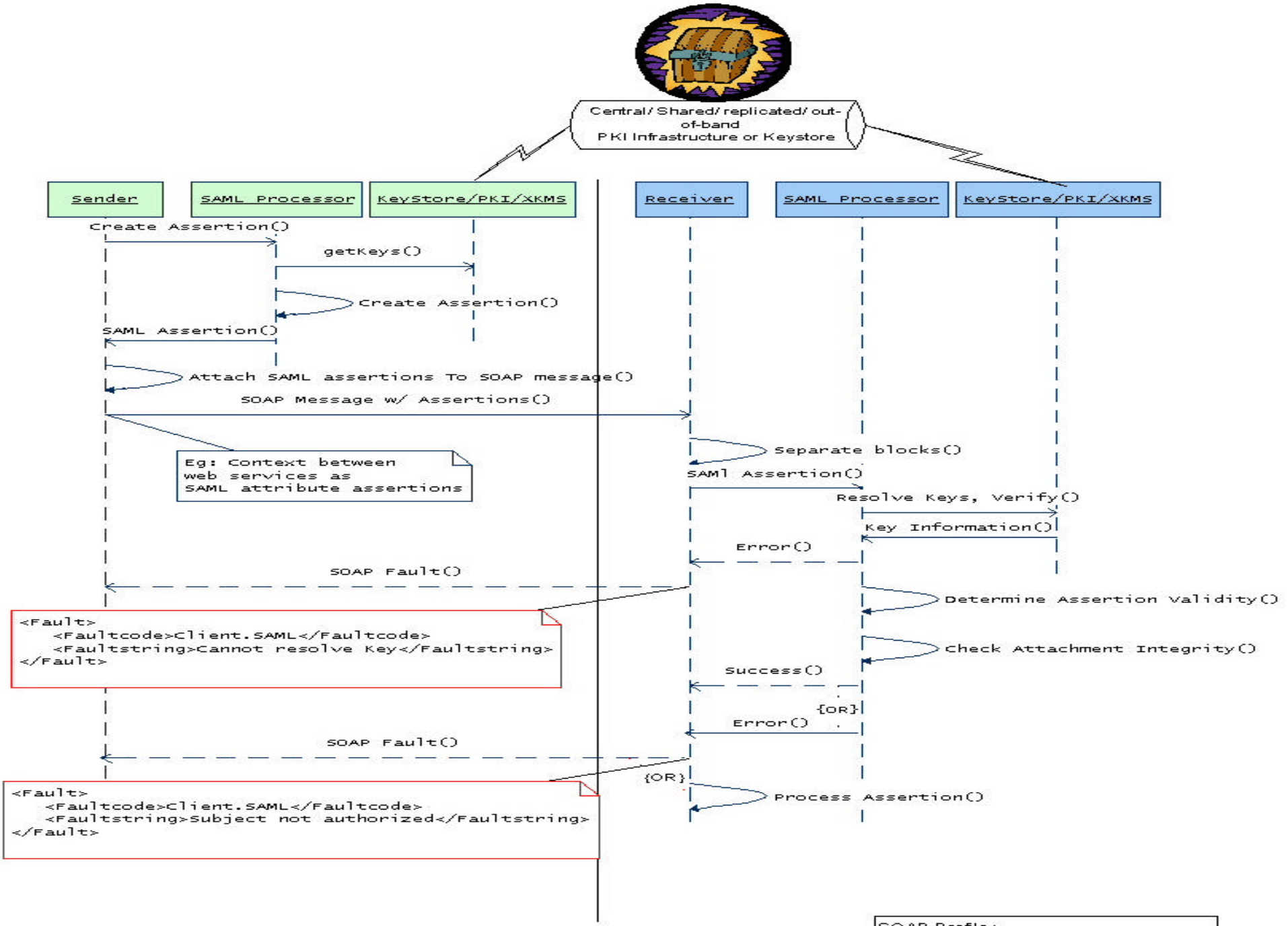
"While PKI will authenticate and control user access to the front-end application, its capabilities do not extend to the connection between the front-end application and the back-end system, Foody said.

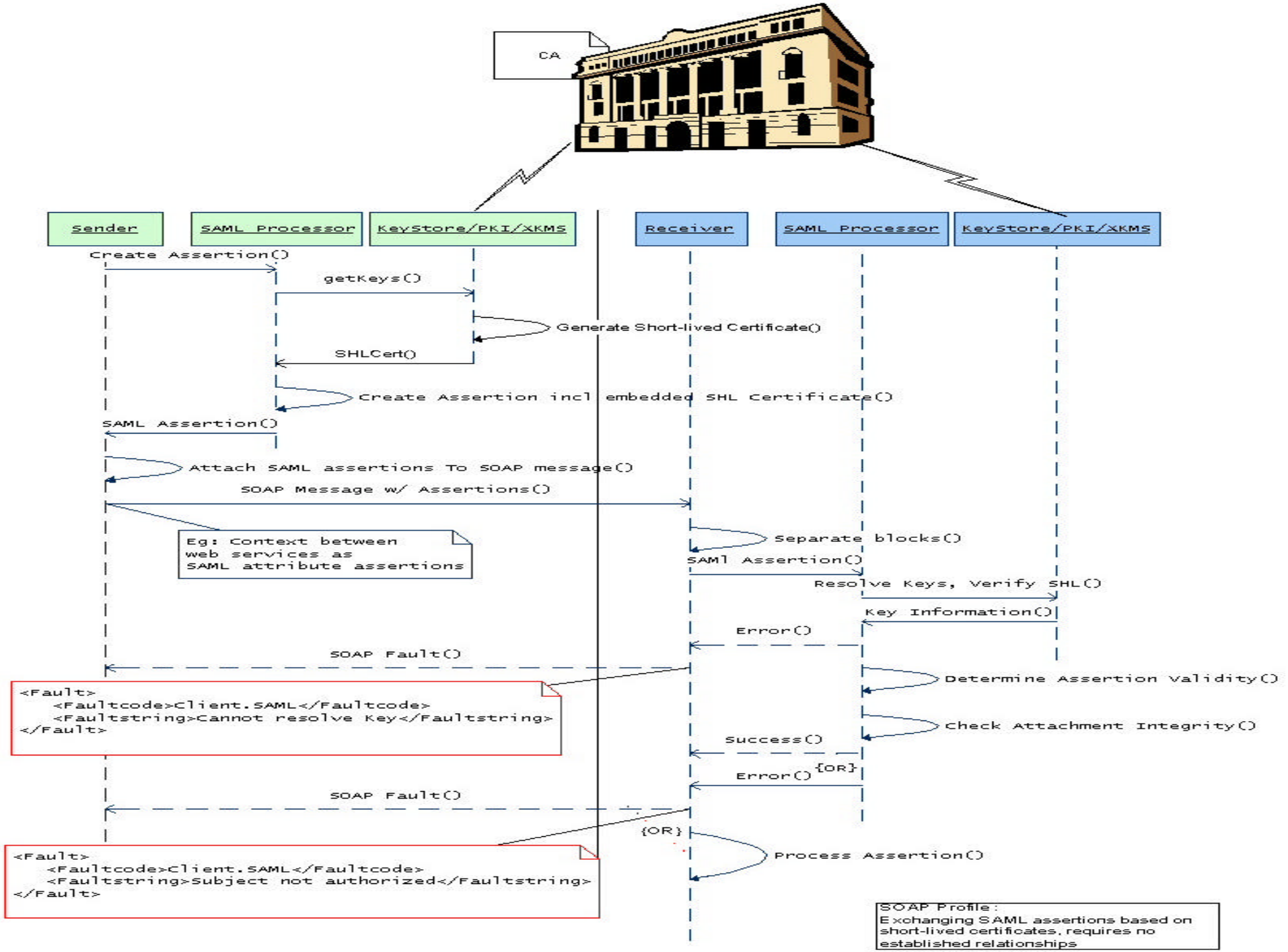
Security is a challenge for integrated e-business systems that are multi-tiered & link multiple applications."

Courtsey : http://www.techrepublic.com/printerfriendly.jhtml;jsessionid=L3MAO24ZPS2TGCTEAAKSFFA?id=r00820010810swm01.htm

Assertion Exchange

- Session handling
- Context for web services
- Applications :
  - B2B Eco Systems
  - Web Services


Central/ Shared/ replicated/ out-of-band
PKI Infrastructure or Keystore
Sender
SAML Processor
KeyStore/PKI/XKMS
Receiver
SAML Processor
KeyStore/PKI/XKMS
Create Assertion()
getKeys()
Create Assertion()
SAML Assertion()
Attach SAML assertions To SOAP message()
SOAP Message w/ Assertions()
Eg: Context between web services as SAML attribute assertions
Separate blocks()
SAMl Assertion()
Resolve Keys, Verify()
Key Information()
Error()
SOAP Fault()
Determine Assertion Validity()
<Fault>
<Faultcode>Client.SAML</Faultcode>
<Faultstring>Cannot resolve Key</Faultstring>
</Fault>
Check Attachment Integrity()
Success()
{OR}
Error()
SOAP Fault()
{OR}
<Fault>
<Faultcode>Client.SAML</Faultcode>
<Faultstring>Subject not authorized</Faultstring>
</Fault>
Process Assertion()
SOAP Profile :
Exchanging SAML assertions with pre-established relationship


CA
Sender
SAML Processor
KeyStore/PKI/XKMS
Receiver
SAML Processor
KeyStore/PKI/XKMS
Create Assertion()
getKeys()
Generate Short-lived Certificate()
SHLCert()
Create Assertion incl embedded SHL Certificate()
SAML Assertion()
Attach SAML assertions To SOAP message()
SOAP Message w/ Assertions()
Eg: Context between
web services as
SAML attribute assertions
Separate blocks()
SAml Assertion()
Resolve Keys, Verify SHL()
Key Information()
Error()
SOAP Fault()
Determine Assertion Validity()
<Fault>
<Faultcode>Client.SAML</Faultcode>
<Faultstring>Cannot resolve Key</Faultstring>
</Fault>
Check Attachment Integrity()
Success()
Error()
{OR}
SOAP Fault()
{OR}
Process Assertion()
<Fault>
<Faultcode>Client.SAML</Faultcode>
<Faultstring>Subject not authorized</Faultstring>
</Fault>
SOAP Profile :
Exchanging SAML assertions based on
short-lived certificates, requires no
established relationships