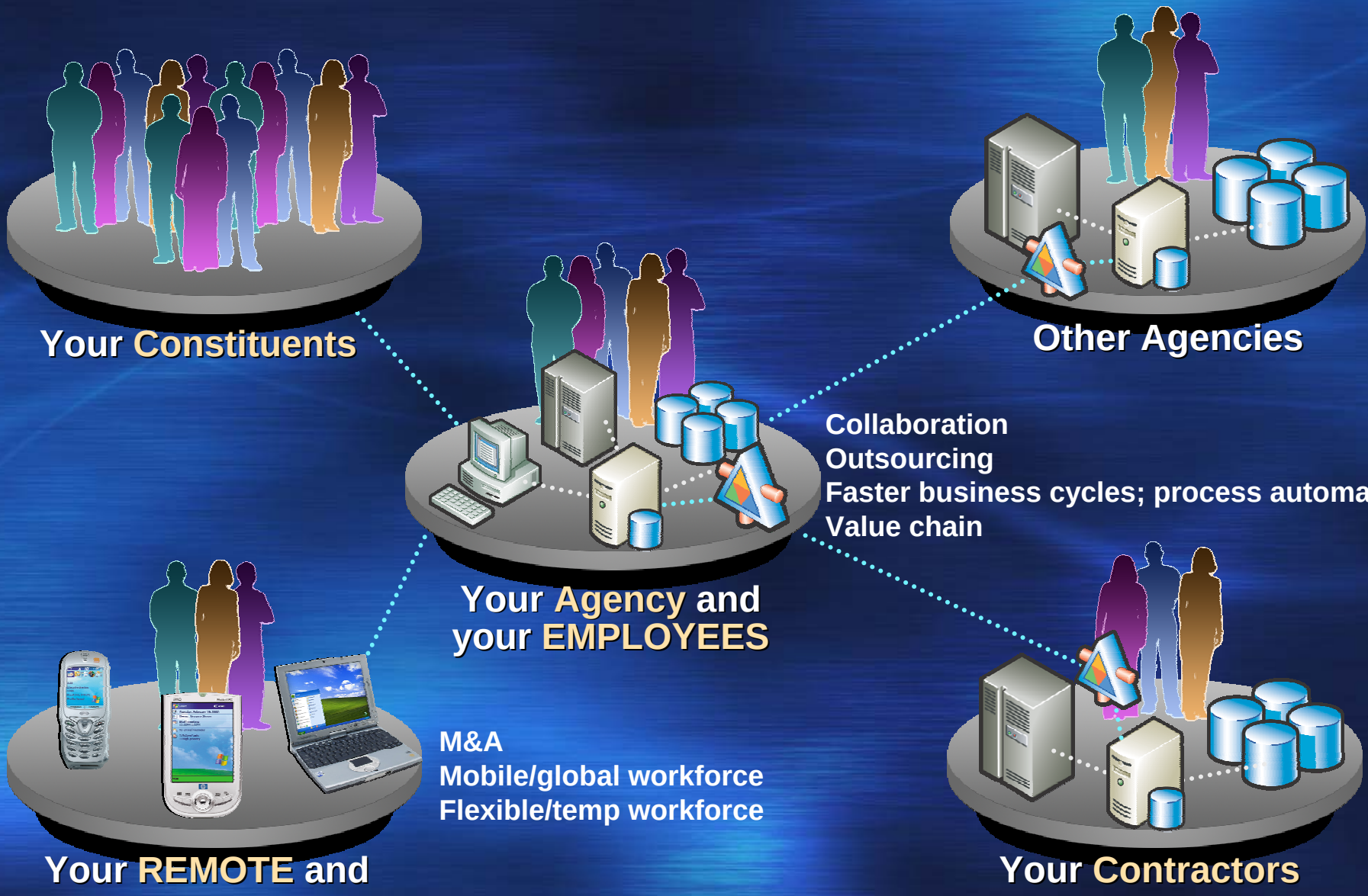
Bridging Trust Between Enclaves

Jon R. Wall
Security / IA
Microsoft Corporation

Agenda

- Level set
  - Distributed IAM Problems
  - Federated IAM Solution
- Active Directory Federation Services
  - Architecture & Components
  - Managing Access with Claims (User Attributes)
  - Demo
- ADFS WS-* Specifications Heritage
  - Multi-vendor Interoperability

eBusiness Extends your Network



Solution: Federated Identity and Access Management

Industry Definition

- Standards-based technology & IT processes …
- Distributed identification, authentication & authorization …
- Across boundaries (security, departmental, organizational or platform boundaries) …

ADFS Vision

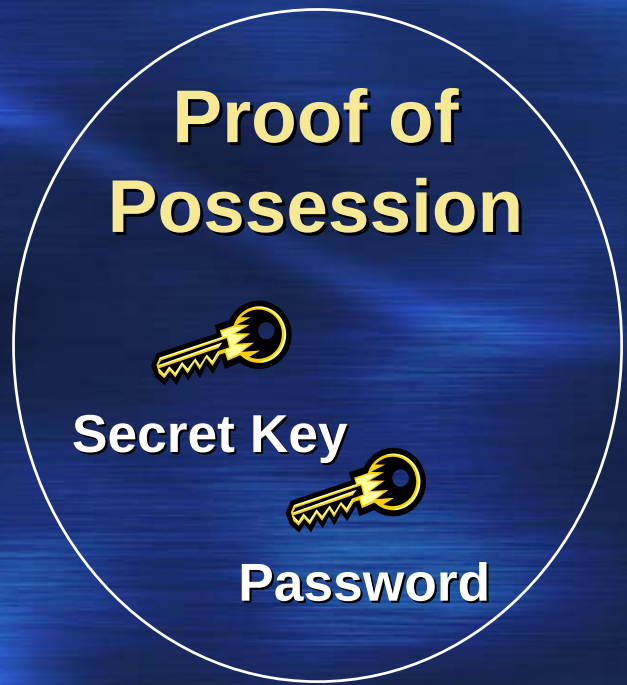
- Log on once, **secure** access to everything
- Leverage Windows identity and services as broadly as possible

Security Tokens & Claims

Distributed authentication/authorization

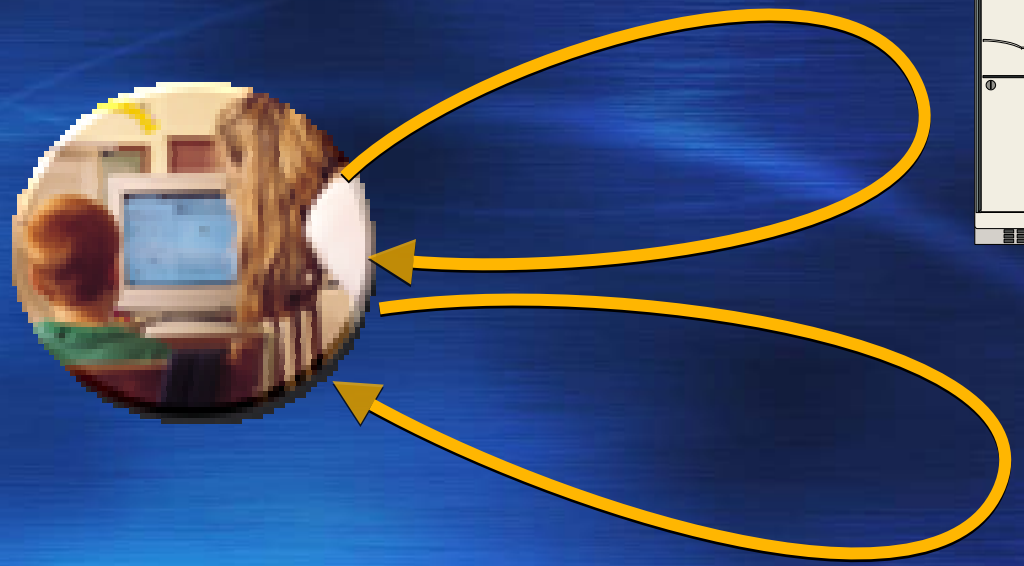
Security tokens assert claims

Claims – Statements authorities make about security principals (name, identity, key, group, privilege, capability, etc).



Security Token Service

A security token service issues security tokens



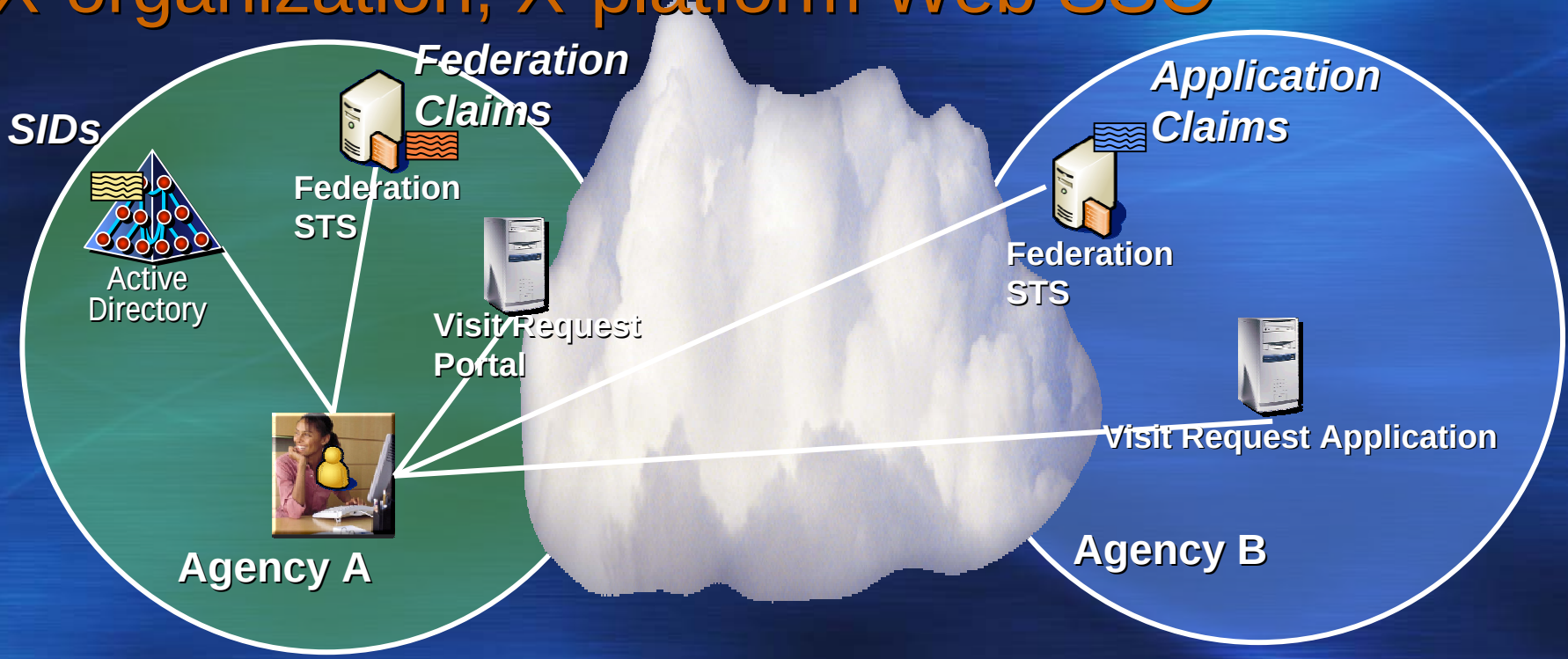
Key Distribution Center

Security Token Service

STS's can "swap" tokens as a request crosses security domain boundaries

Federated IAM in Action

X-organization, X-platform Web SSO



1. User clicks Agency A portal link to Agency B Visit Request application
2. User redirected to Agency A STS
   - Seamlessly authenticated via Kerberos (Windows integrated AuthN & AD)
3. User obtains SAML security token from Agency A STS for Agency B STS
   - Federation claims per business agreement
4. User obtains SAML security token from Agency B STS for application
   - Federation + application-specific claims
5. User accesses Agency B Visit Request application

Active Directory Federation Services

ADFS Architecture

Active Directory

- Authenticates users
- Manages attributes used to populate claims

Federation Service (FS)

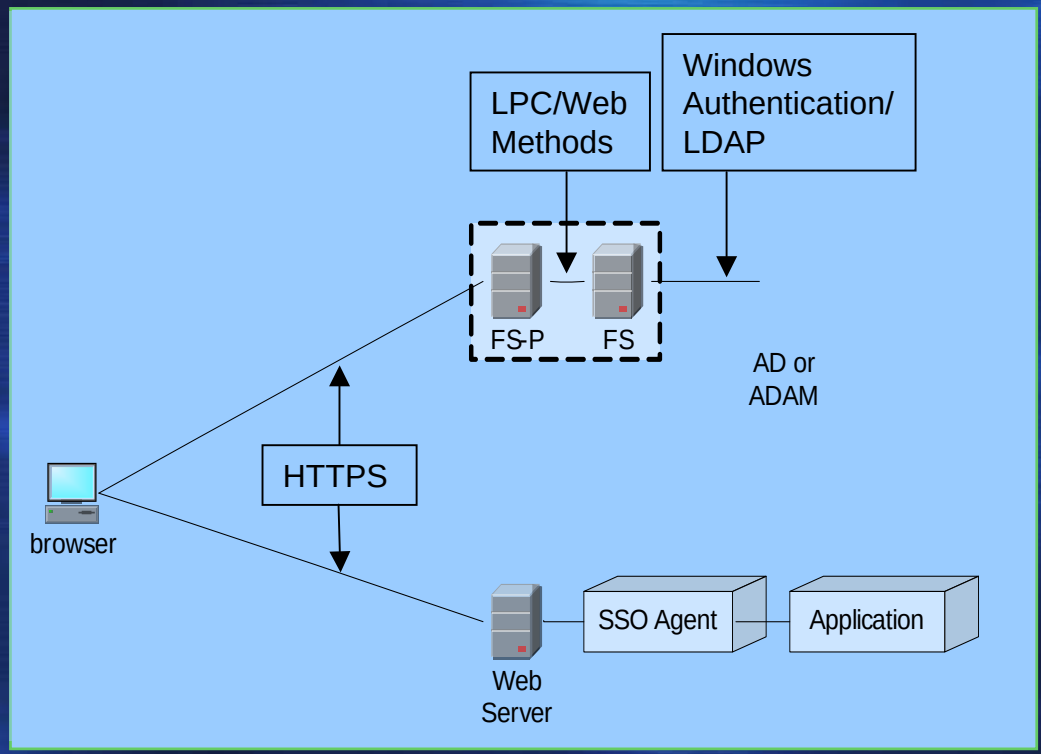
- STS Issues security tokens
- Manages federation trust policy

FS Proxy (FS-P)

- Client proxy for token requests
- Provides UI for browser clients

Web Server SSO Agent

- Enforces user authentication
- Creates user authorization context



Note:

ADFS supports both W2K & W2K3 forests

FS & FS-P co-located by default, Can be separate boxes

FS, FS-P & SSO agent require IISv6 W2K03 R2

Federation Service

ASP.NET-hosted service running on IISv6 - W2K3 Server R2

Federation Policy management

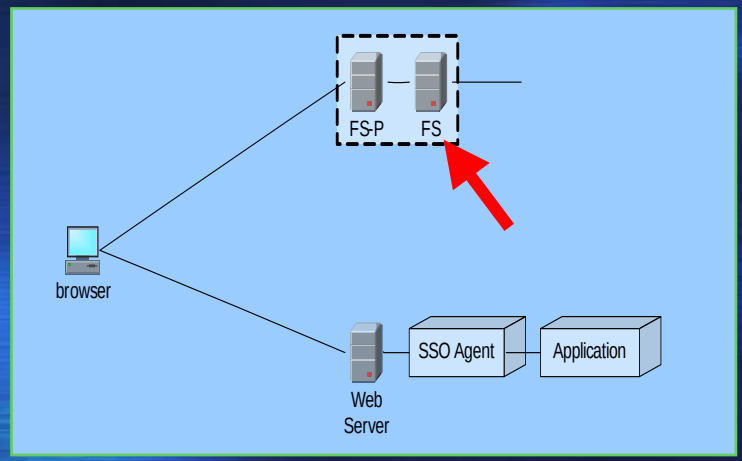
- Establishes trust for signed security tokens by certificate-based key distribution
- Defines token/claim types & shared namespace for Federated security realms

Security token generation

- Retrieves user attributes for claim generation from AD (or ADAM) via LDAP
- Transforms claims (if required) between internal & federation namespaces
- Builds signed SAML security token & sends to LS
- Builds “User SSO” cookie contents & sends to LS

User authentication

- Validates ID/Password via LDAP Bind for Forms-based authentication



Federation Service Proxy

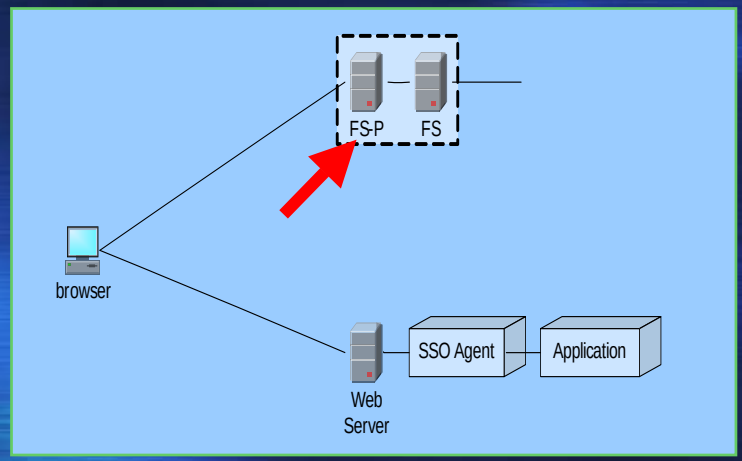
ASP.NET-hosted service running on IISv6 - W2K03 Sever R2

User authentication

- Provides UI for Home Realm Discovery & Forms-based Logon
- Authenticates users for Windows Integrated & Client SSL authentication
- Writes “User SSO” cookie to Browser (similar to Kerberos TGT)

Security token processing

- Requests security token for client from FS
- Routes token to web server via “POST redirect” through Browser



Web Server SSO Agent

ISAPI extension for IISv6 - W2K3 Server R2

User authentication

- Intercepts URL GET requests & Redirects un-authenticated clients to LS
- Writes “Web Server SSO” cookie to Browser (similar to Kerberos service ticket)

Windows Service

User authorization

- Creates NT Token for impersonation (AD users only)

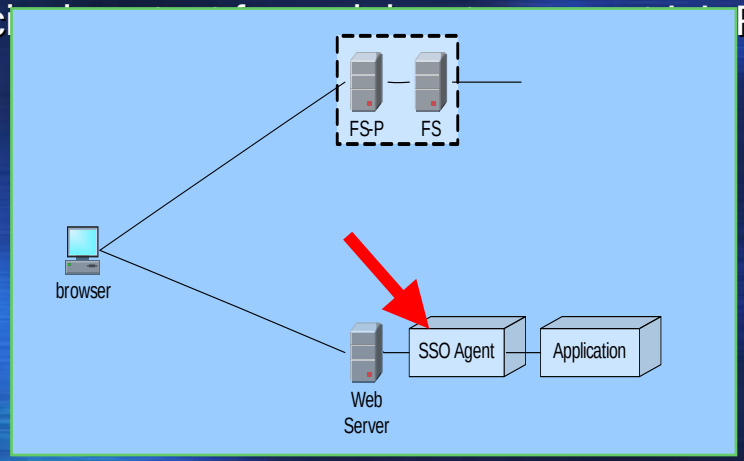
Managed Web Module

Security token processing

- Validates user’s security token and parses claims in token

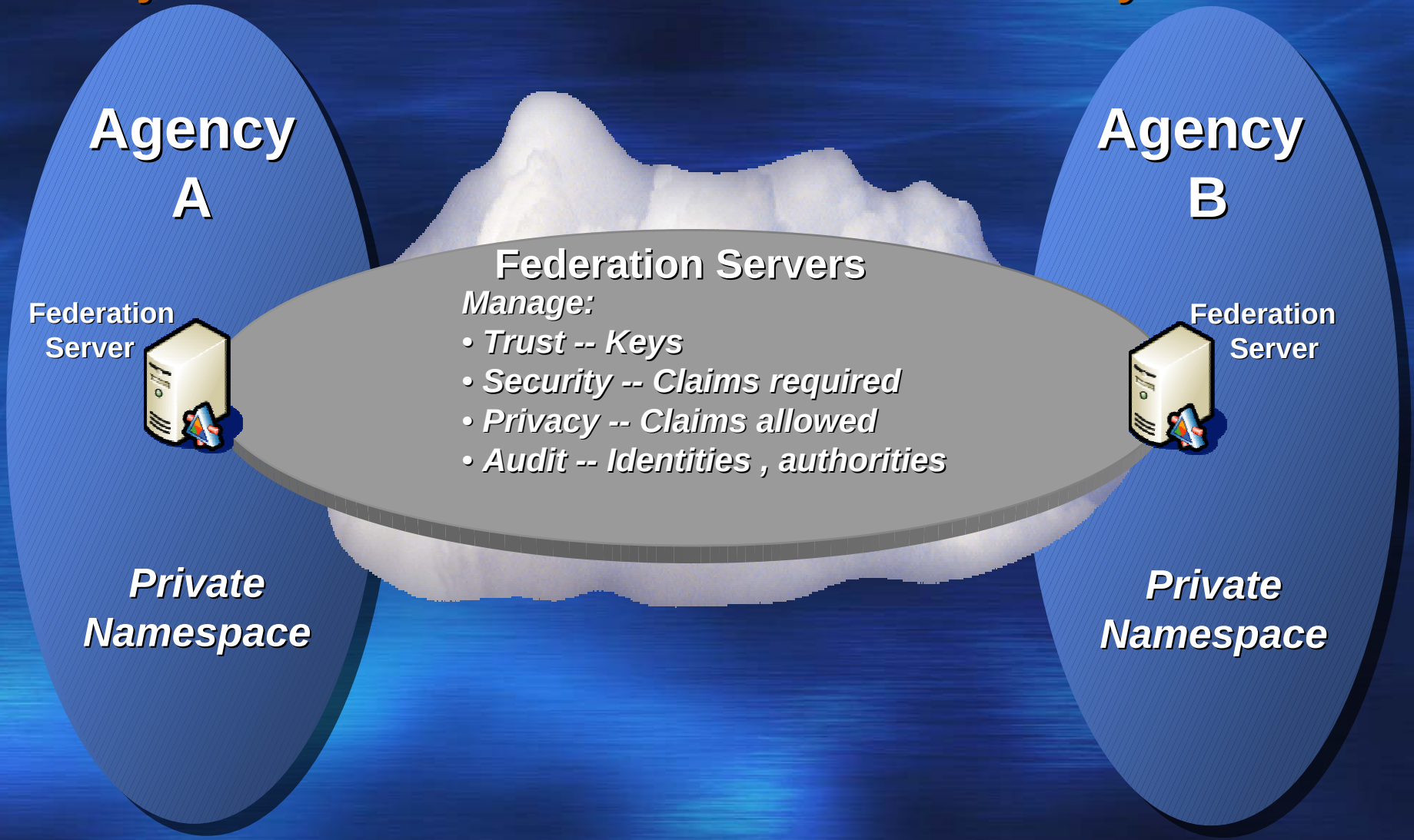
User authorization

- Populates ASP.NET GenericPrinci Role()
- Provides raw claims to app

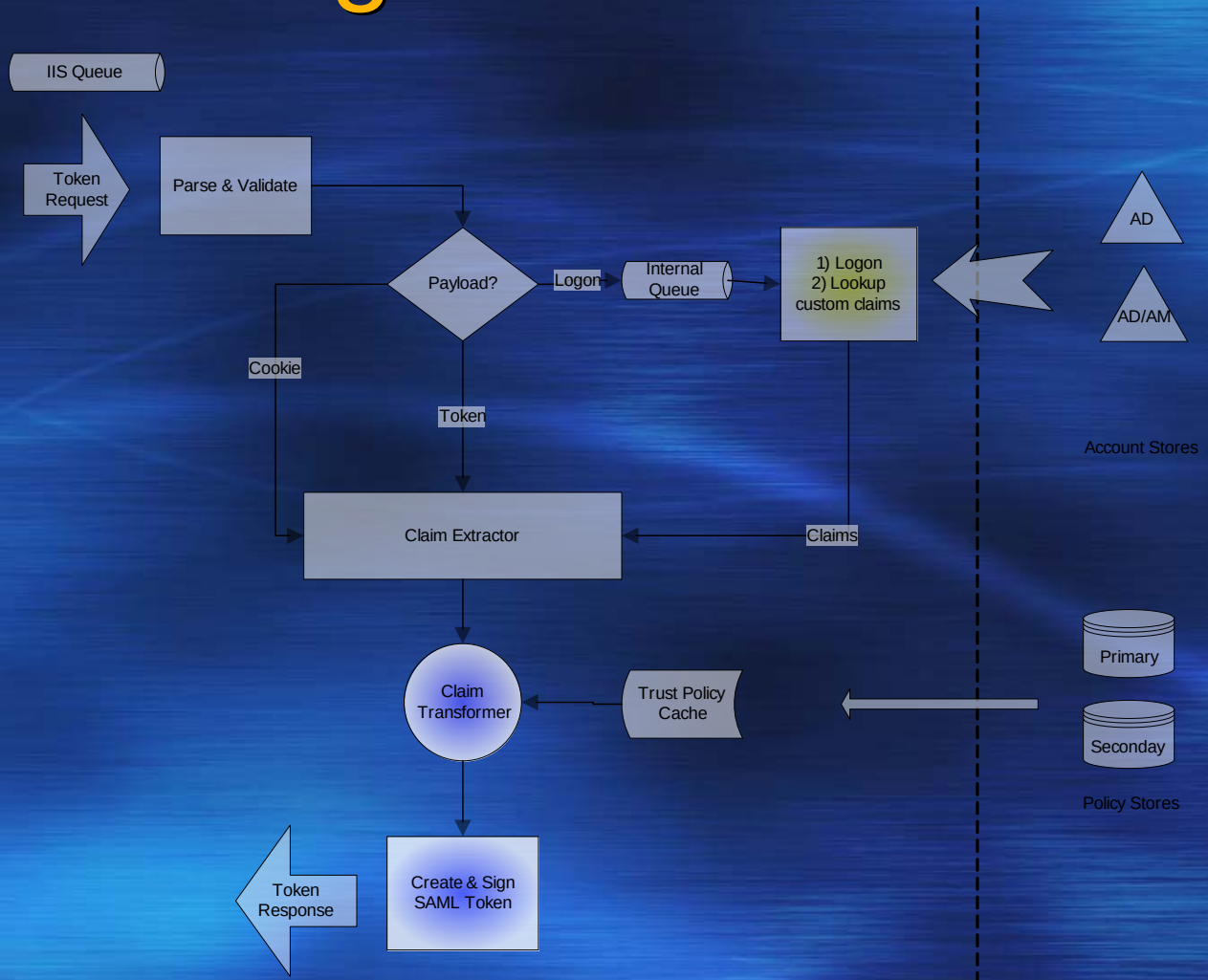


ADFS Identity Federation for IAM

Projects AD Identities to other security realms



ADFS: Claim & Token Processing



ADFS: Supported Security Tokens

- Currently only issue SAML tokens
- Tokens are not encrypted
  - All messages are over HTTPS
- Tokens are signed
  - (default) Signed with RSA Private key and signature verified with public key from X.509 certificate
  - (optional) Can be signed with Kerberos session key
    - FS-R tokens for Web server SSO Agent
      - NT service component of Web server SSO Agent must run as a domain service account and must have an SPN configure

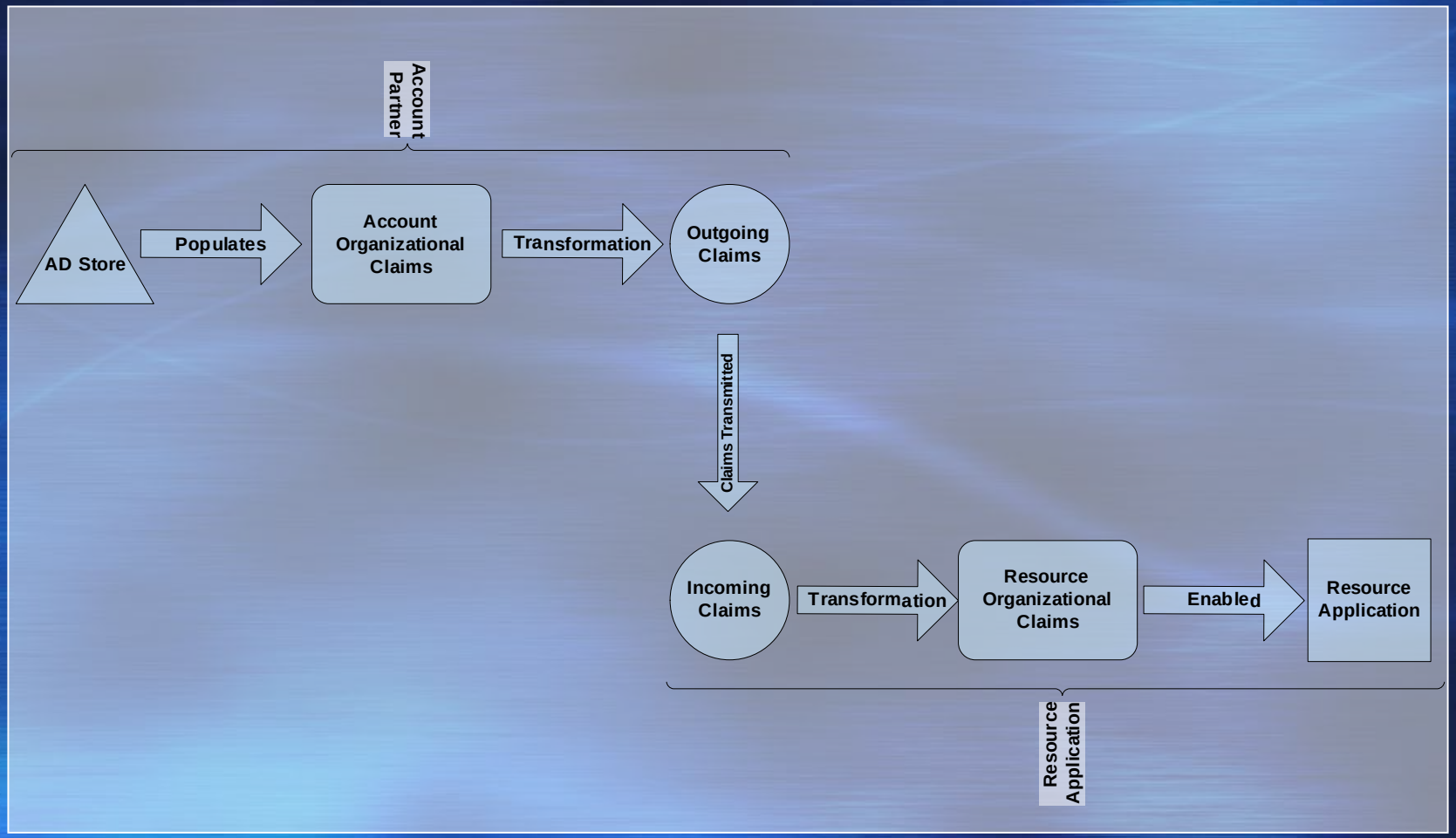
ADFS: Supported Claim Types

- WS-Federation interoperable claim types
  - Identity
    - User Principal Name (UPN)
    - Email Address
    - Common Name (any string value)
  - Group
  - Custom
    - name/value pair (eg SSN / 123-45-6789)
- ADFS-to-ADFS only authZ data
  - SIDs
    - Sent to avoid shadow accounts (for employees) in extranet DMZ
    - Sent in SAML token Advice element (not a standard claim type)
- Organizational claims
  - Common set of claims across account stores and partners
  - Mark organizational claims as sensitive (not audited/logged)

ADFS: Claims Processing Extensibility

- Interface allows plug-in modules to be developed for Custom Claim Transformation
  - FS supports one claim transform module, Not a pipeline for multiple modules
  - Further lookups to a LDAP or SQL store
  - Complex claim transformations requiring computation

ADFS Federation Claims Flow

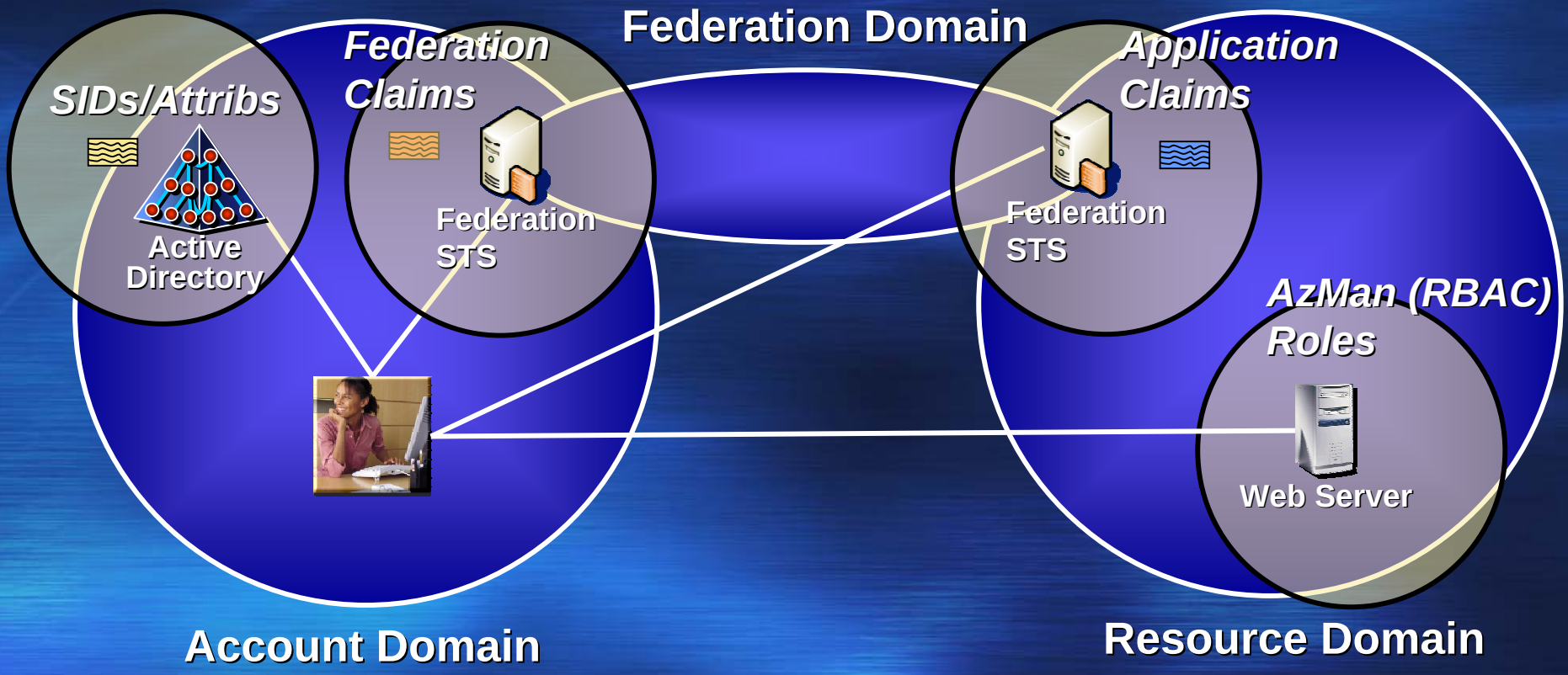


Supply Chain/Purchasing Application

Demo

Federated IAM via Claims & RBAC

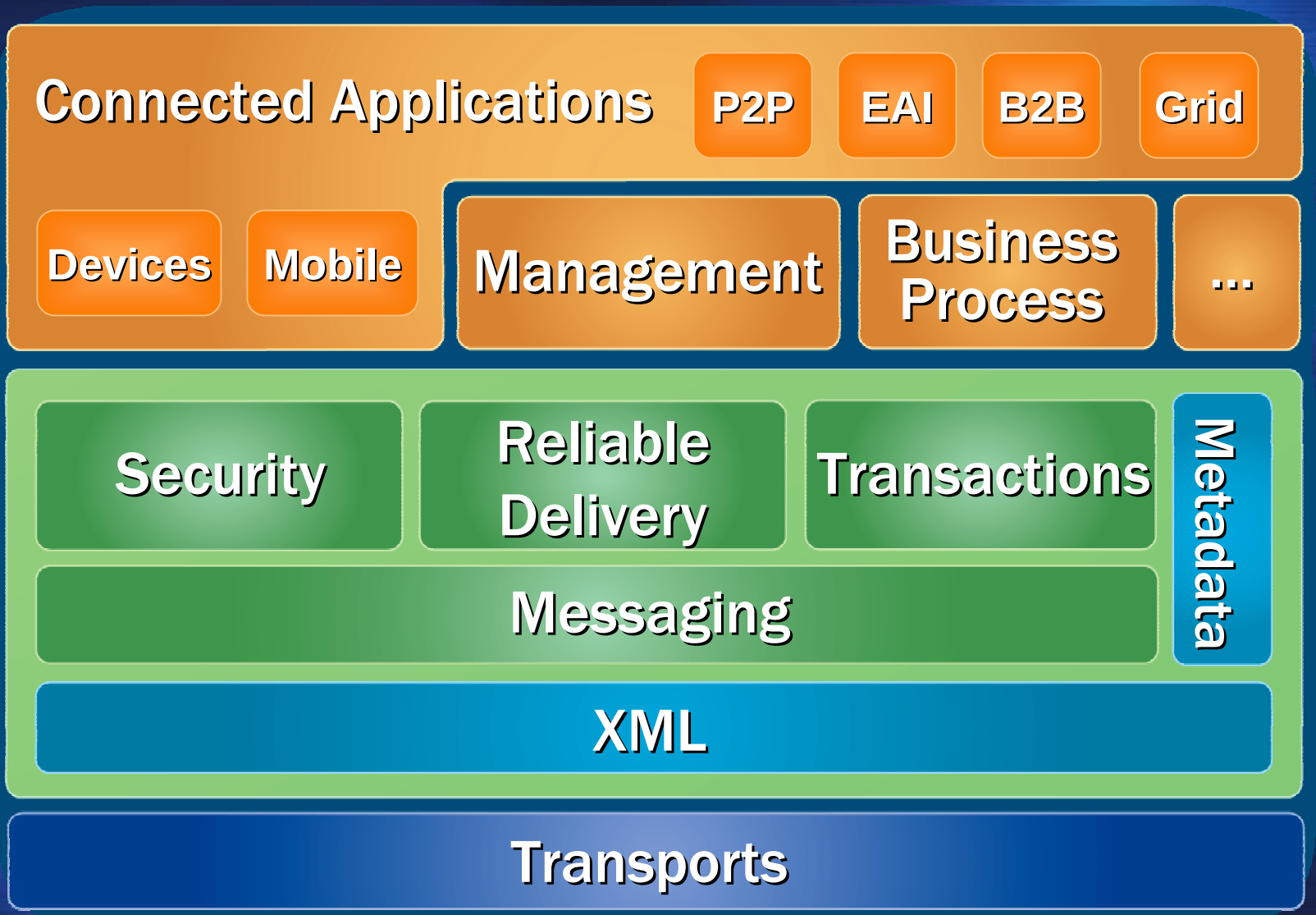
ADFS & Authorization Manager integration



ADFS Web Services Specifications Heritage

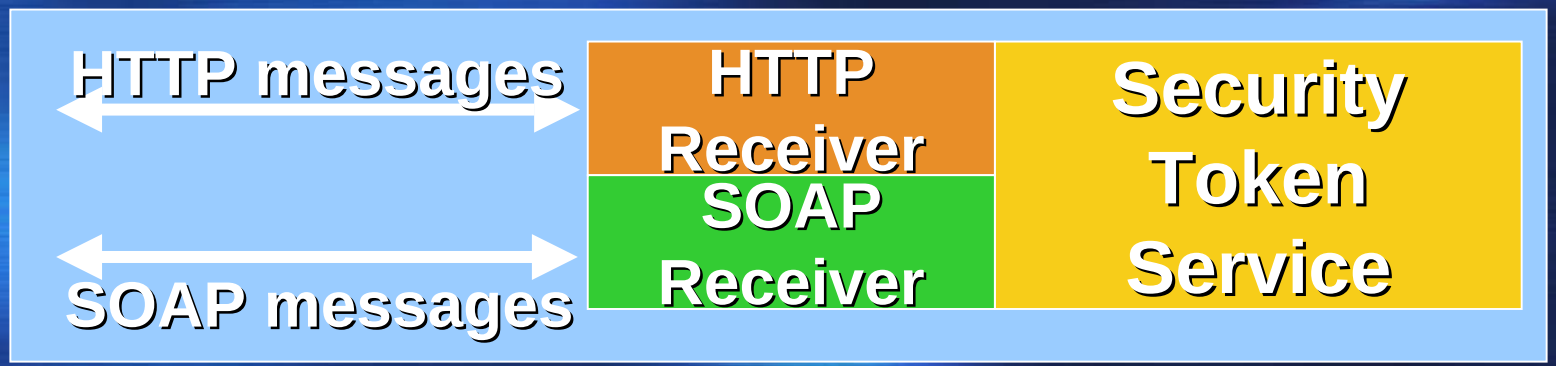
* Interoperability
* Extensibility

Web Services Specifications



WS-Federation

- Web Services Federation Language
  - Defines messages to enable security realms to federate & exchange security tokens
- BEA, IBM, Microsoft, RSA, VeriSign
- Two “profiles” of the model defined
  - Passive (Browser) clients – HTTP/S
  - Active (Smart) clients – SOAP



Passive Requestor Profile

- Binding of WS-Federation & WS-Trust for browser clients
  - Authentication Requires secure transport (HTTPS)
    - Passive (dumb) clients
      - Adhere to policy by following redirects
      - Indirectly acquire tokens via HTTP msgs
      - Cannot provide “proof of possession” for tokens
  - Limited (time based) token caching
    - Tokens can be replayed

WS-Federation Interoperability

- WS-* public workshops/mailing list prepare specs for submission to standards bodies
  - http://groups.yahoo.com/group/WS-Security-Workshops/
- WS-Federation vendor workshop (3/29/04)
  - Passive Requestor Profile & SAML token
  - Microsoft, IBM, RSA, Oblix, PingID, Open Network, Netegrity
  - 100% interop achieved by all participants
- WS-Federation product previews at TechEd
  - Interop pavilion & Vendor panel

Active Requestor Profile

- A binding of WS-Federation & WS-Trust for SOAP clients
  - Determine token needs from policy
  - Actively request tokens via SOAP msgs
- Strong authentication of all requests
  - Client can provide “proof of possession” for security tokens
- Supports delegation
  - Client can provide token for web service to use on its behalf
- Allows rich token caching at client
  - Improved user experience & performance


Microsoft®