Building Secure Distributed Applications Using Windows Server 2003

Chris Crall

Program Manager
Microsoft Corporation

HP WORLD 2003
Solutions and Technology Conference & Expo

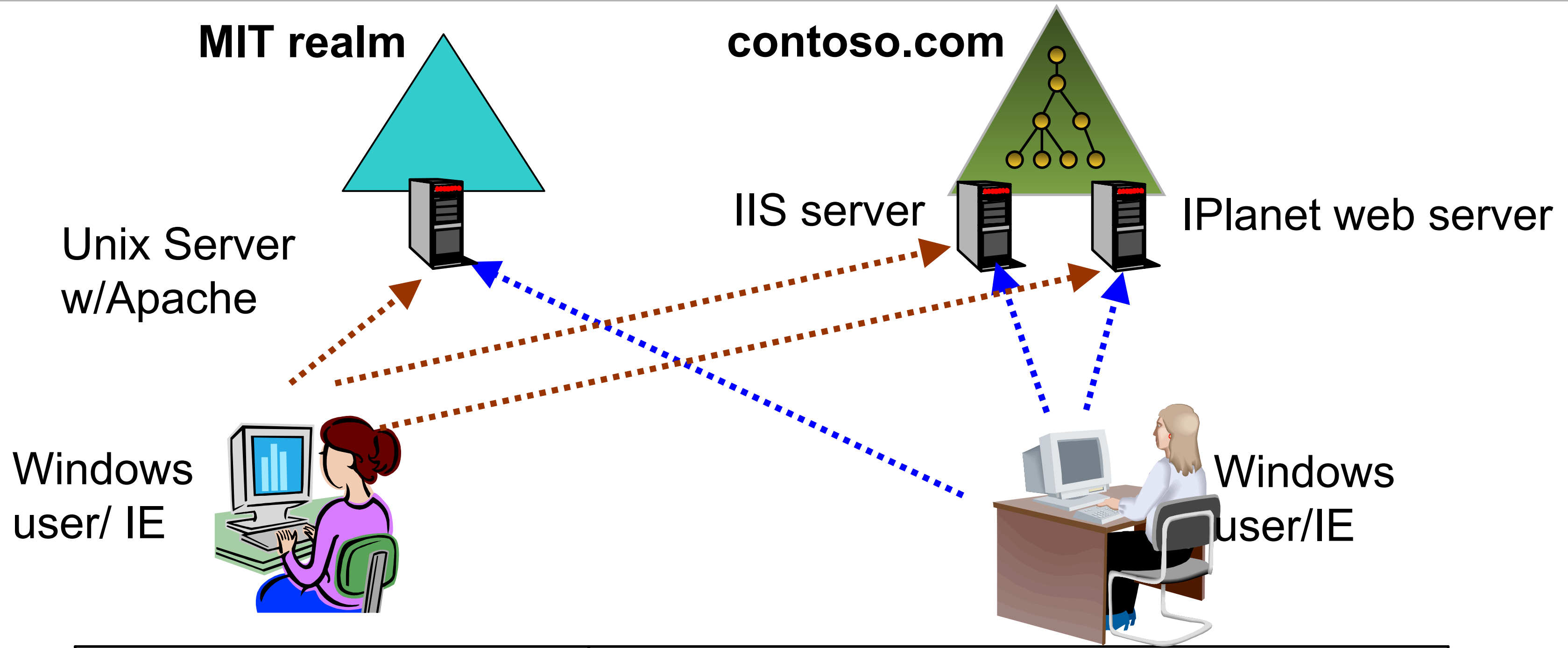
Agenda

I. Multi tier app scenarios
II. Authentication mechanisms
III. Authorization mechanisms
IV. Trust mechanisms
V. Putting it all together

I. Multi tier app scenarios

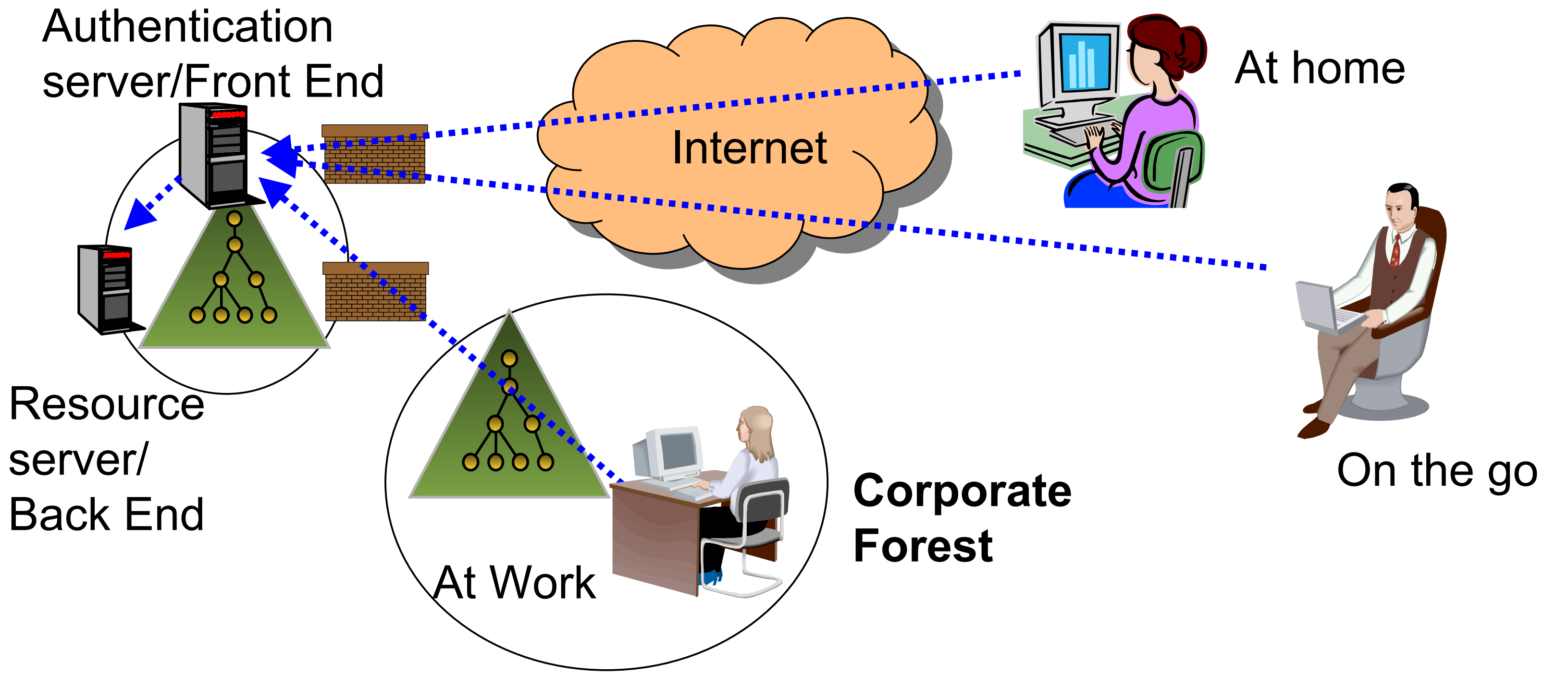
- Internet Access
- Employee Access
- Customer Access
- Partner Access

Intranet access scenario



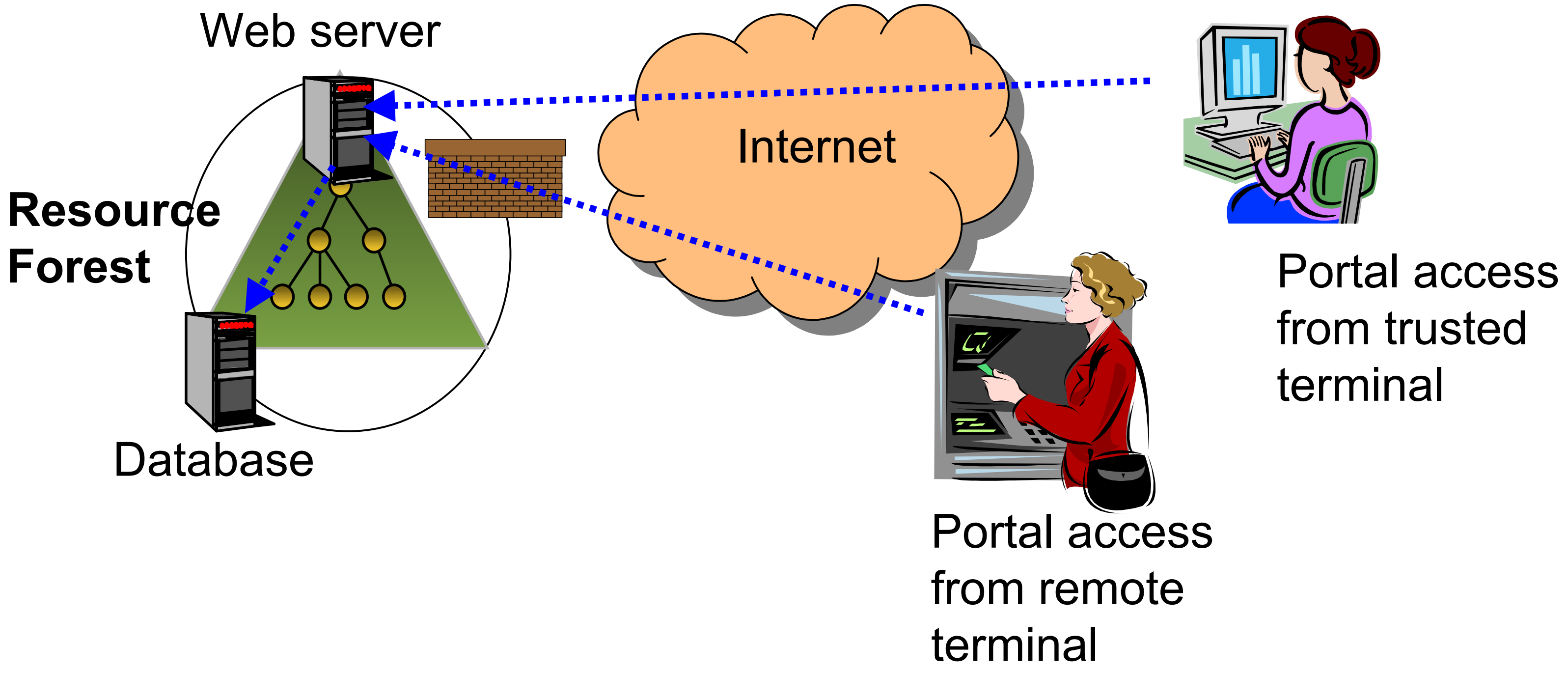
Clients	**IE, Netscape, Opera, …**
Authentication strength	**High**
Usability	**Medium**
Experience	**Provide true SSO**
Infrastructure	**Avoid multiple identity store?**

Extranet - employee access scenario



Clients	IE, Netscape, Opera, …
Authentication strength	High
Infrastructure	No duplication
Usability	Medium

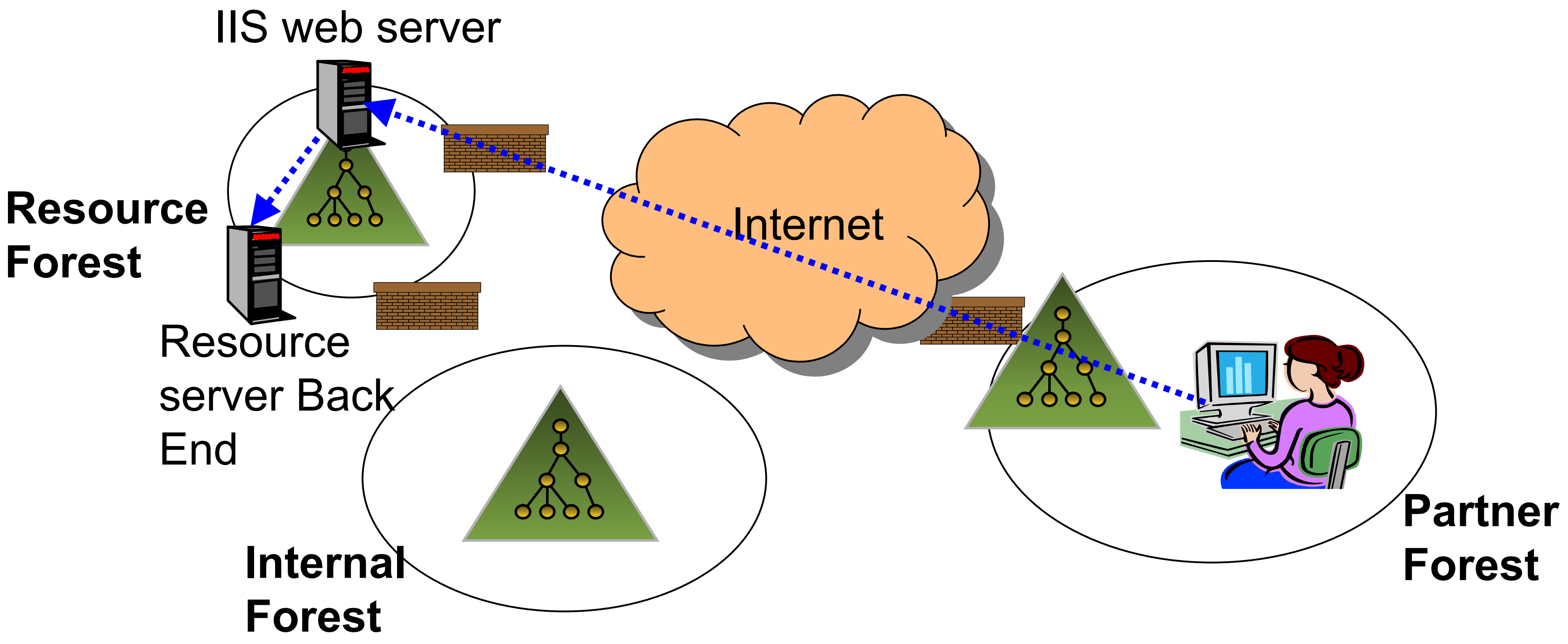
Extranet - Customer access scenario



Web server
Resource Forest
Database
Internet
Portal access from trusted terminal
Portal access from remote terminal

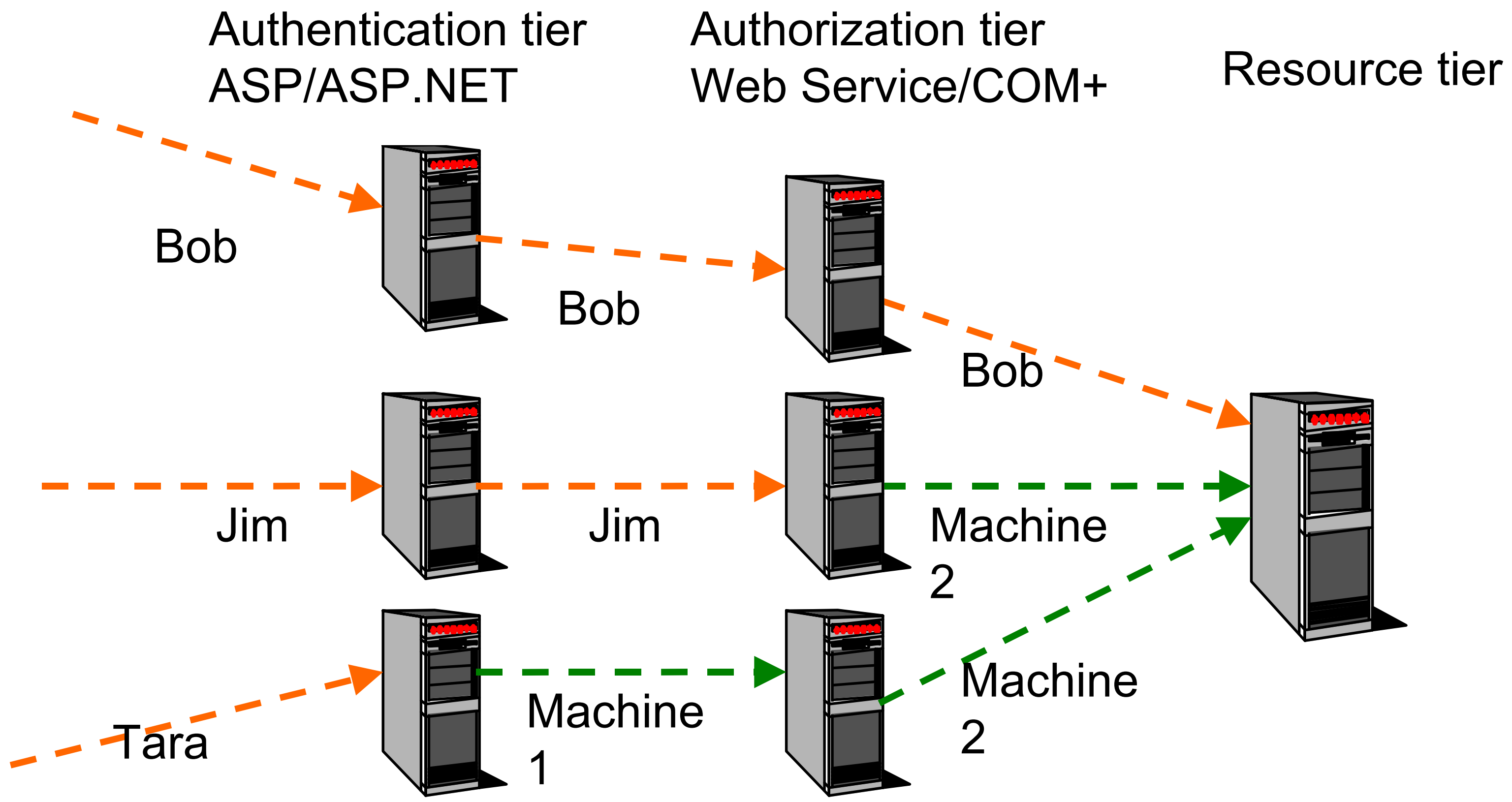
Clients	**IE, Netscape, PocketPC**
Usability	**High**
Authentication strength	**Depends on the application (low for most B2C)**

Extranet- Partner access



Clients	IE or Netscape
Authentication strength	High
Usability	Medium

Multi tier apps –Trusted Subsystem vs. Impersonation



II. Authentication Mechanisms

- Windows Authentication Protocols
  - Kerberos
  - NTLM
  - Negotiate
  - SSL
  - Digest
  - Forms
  - Passport
- Protocol Transitions

Windows Integrated

- Background
  - Uses Negotiate protocol (RFC2478)
  - Prefers Kerberos(RFC1510) but falls back to NTLM when not available
- Pros/Cons
  - Leverages existing AD and infrastructure
  - Simple to enable – checkbox in IIS
- Requirements
  - Requires IE 5.0 or higher on W2K or higher
  - Code required - none

SSL authentication

- SSL & TLS (RFC 2246)
- Background
  - Both Server and client authentication available
  - Widely used for server auth but also useful for client auth
- Pros/Cons
  - Requires widespread distribution and availability of certificates
  - Once certs are distributed easy to deploy
- Requirements
  - Supported by most browsers/web servers
  - Code required - none

Basic/Digest authentication

- Background
  - Basic and Digest are HTTP specific authentication methods
  - Basic sends clear passwords
- Pros/Cons
  - Easy to enable with IIS checkbox
  - SSL encryption advised for both to protect password data
- Requirements
  - Netscape only supports basic
  - Code required - none

Forms authentication

- Background
  - User types in name and password in app form
  - Cookie written back to browser
- Pros/Cons
  - Easy to develop with ASP.NET forms authentication
  - Users have to remember password for each site
- Requirements
  - Supports most browsers

Forms authentication

One line of code logs the user in

```
//
Bind to AD to verify user creds

FormsAuthentication.RedirectFromLoginPage(Us
ername, Persist);
```

Passport authentication

- Background
  - Similar to forms auth
  - No worries about managing user passwords
  - Other information can be stored locally
- Pros/Cons
  - Built in mapping to Windows users
  - SSO for users
- Requirements
  - Same browser requirements as forms
  - No code for authentication

Passport authentication

Write the passport user name (this will be the PUID)

Associate a passport identity with the current identity

To get the additional attributes we need to call the GetObject method

```
using System.Security.Principal;

PassportIdentity Passport = Context.User.Identity;
Response.Write(Passport.Name);
Response.Write(Passport.GetObject("MemberName"));
```

Signed messages

- Background
  - Integrity protection of messages without SSL burden
  - Uses CAPICOM (generates PKCS#7 standard messages)
  - Message verification implies message was sent by the corresponding sender
- Pros/Cons
  - Requires distribution of certs
  - No server authentication possible
- Requires IE

CAPICOM

Open the store

Select the certificate

Then the raw data needs to be signed.

```
 Set oStore = CreateObject("CAPICOM.Store")
 oStore.Open CAPICOM_CURRENT_USER_STORE,
CAPICOM_MY_STORE, CAPICOM_STORE_OPEN_READ_ONLY
Or CAPICOM_STORE_OPEN_EXISTING_ONLY

set oSelectedCerts = oCerts.Select()
  Set oSignerCert = oSelectedCerts (1)
Set oSigner = CreateObject("CAPICOM.Signer")
oSigner.Certificate = oSignerCert
      Set oSigned = CreateObject("CAPICOM.SignedData")
      oSigned.Content = strDataToSign
      SignedData = oSigned.Sign( oSigner )
```

Protocol Transition - Kerberos S4U2self extension

- Background
  - Service: authenticates via Kerberos
  - User: authenticates to service (however)
  - Service: makes S4U2self TGS-REQ
    - Gets service ticket to itself; PAC has user’s authorization data (user & groups SIDs)
- Requirements
  - LsaLogonUser(user_UPN)
    - No Password needed
    - Impersonation token (service has TCB)
    - Identification token (no TCB)

Protocol transition

Only one line required

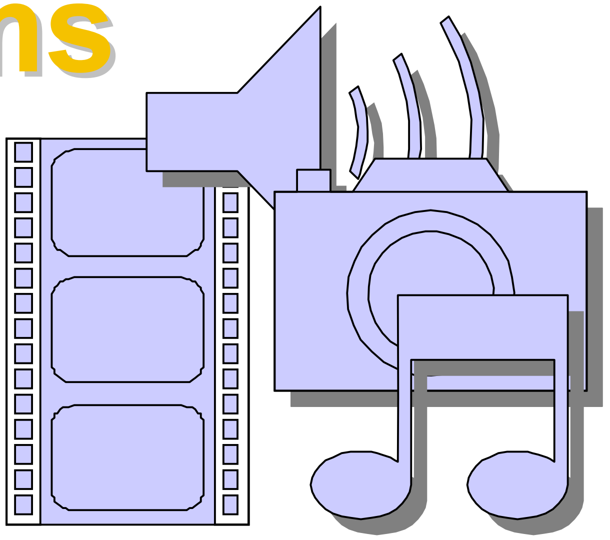
```
using System.Security.Principal;

WindowsIdentity Id = new
WindowsIdentity("TESTDOM\test")
```

Constrained Delegation Kerberos S4U2proxy extension

- Background
  - Service: gets service ticket to itself
    - From Kerberos client or via S4U2self
    - Service does not get user's TGT
  - Service: makes S4U2proxy TGS-REQ
    - Delegation evidence is ticket, not user's TGT
    - Gets delegated service ticket to target server; PAC has user's authorization data
- Requirements
  - InitializeSecurityContext(target_SPN)
    - Service needs impersonation token
    - Windows 2003 native mode

Demo

Authentication mechanisms



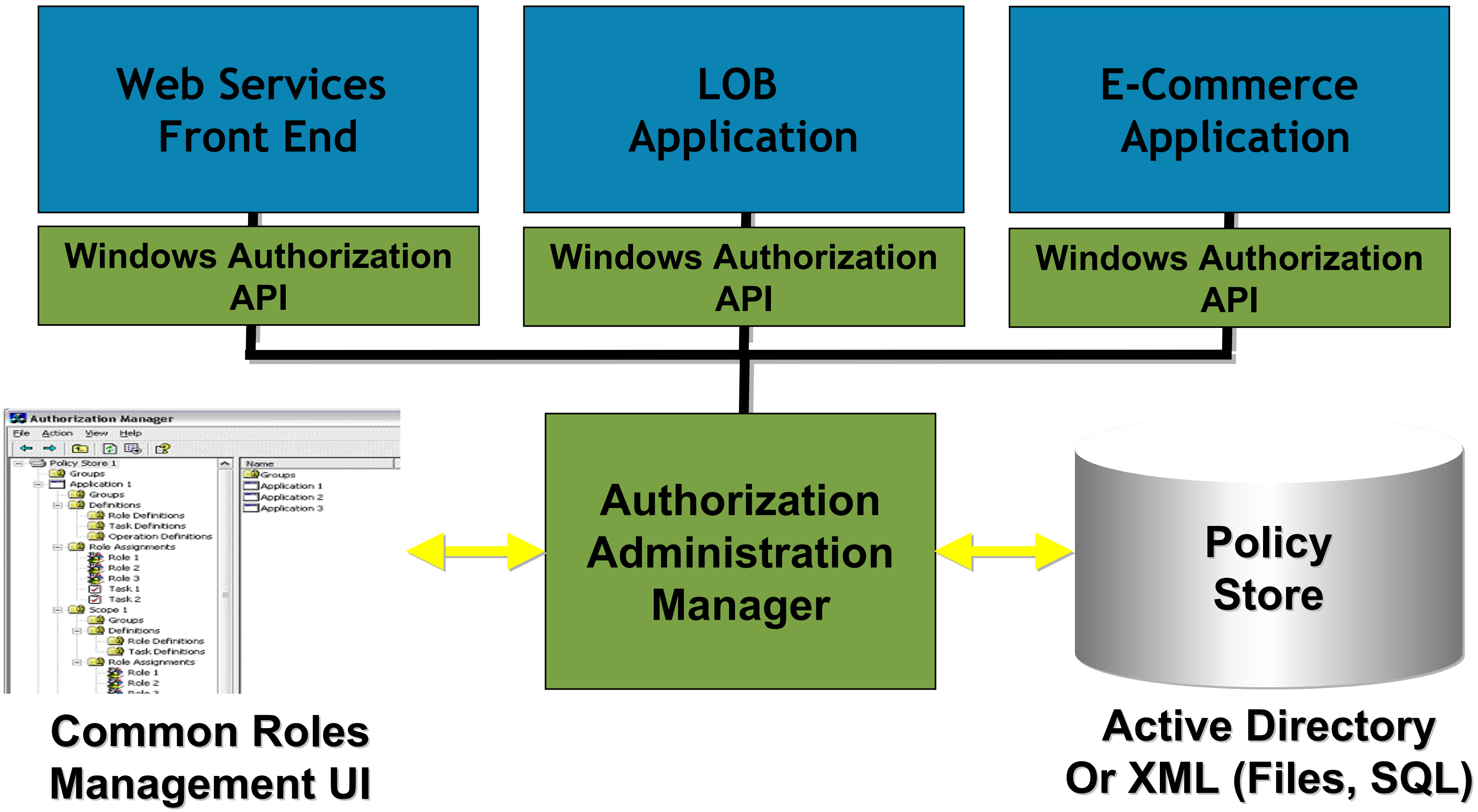
III. Authorization Mechanisms

- Role Based Access
- AZMan

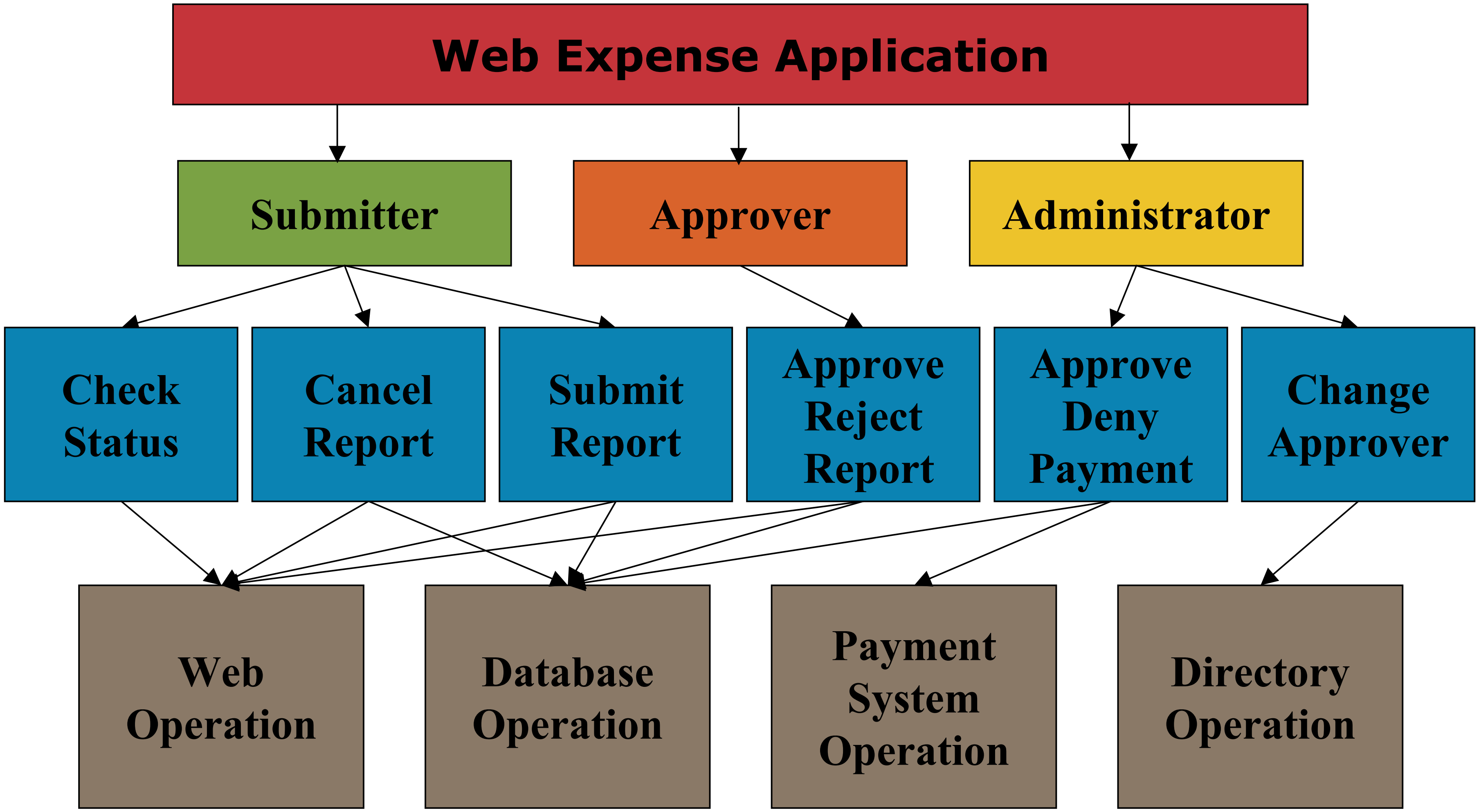
Why Role Based Authorization?

- Developers
  - App can manage its own groups
  - Know if user request can succeed
- Administrators
  - Manage roles, not object ACLs
    - Job description defines entitlements
    - No more ACE ordering & ACL inheritance surprises
  - Simplify entitlement reporting & auditing
- Resource owners
  - Query groups capture business dynamics

Windows.NET Authorization Manager



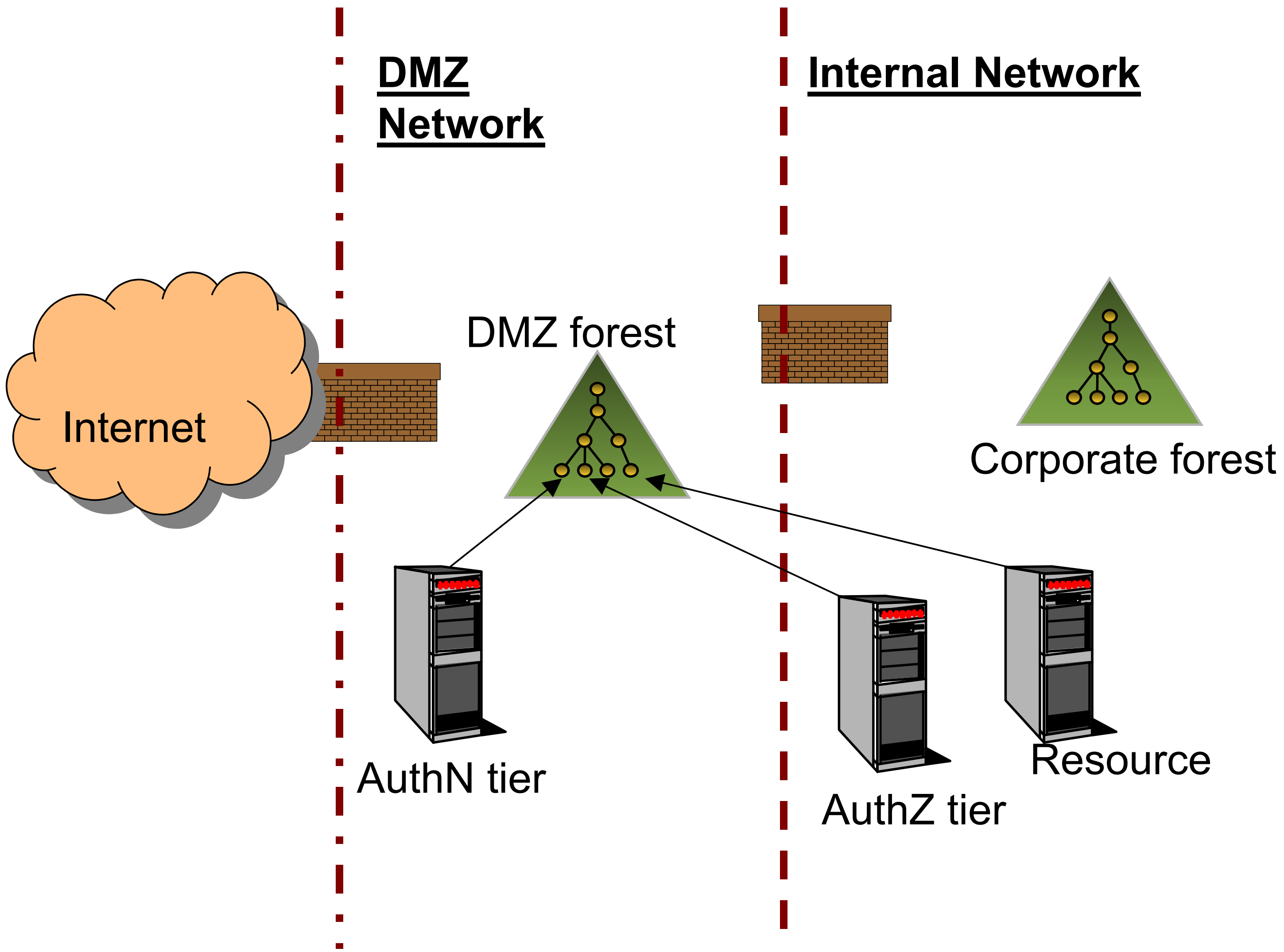
Role={Tasks}
Task={Operations}



Programming Model

- Development
  - Implement Operations, Tasks, BizRule scripts
- Installation
  - Declare Policy definitions
    - Operations, Tasks (w/ BizRules), Roles
- Runtime
  - Startup
    - AzInitializeAdminMgr, AzInitializeApplication
  - Client Connection
    - IAzInitializeContext (from NT token or UserName)
    - Render UI: GetRolesForUser
  - Operation Request
    - AzClientContext.AccessCheck(Scope, Operations, BizRule parameters (optional))
    - Biz Rules are automatically executed.

Deploying the application



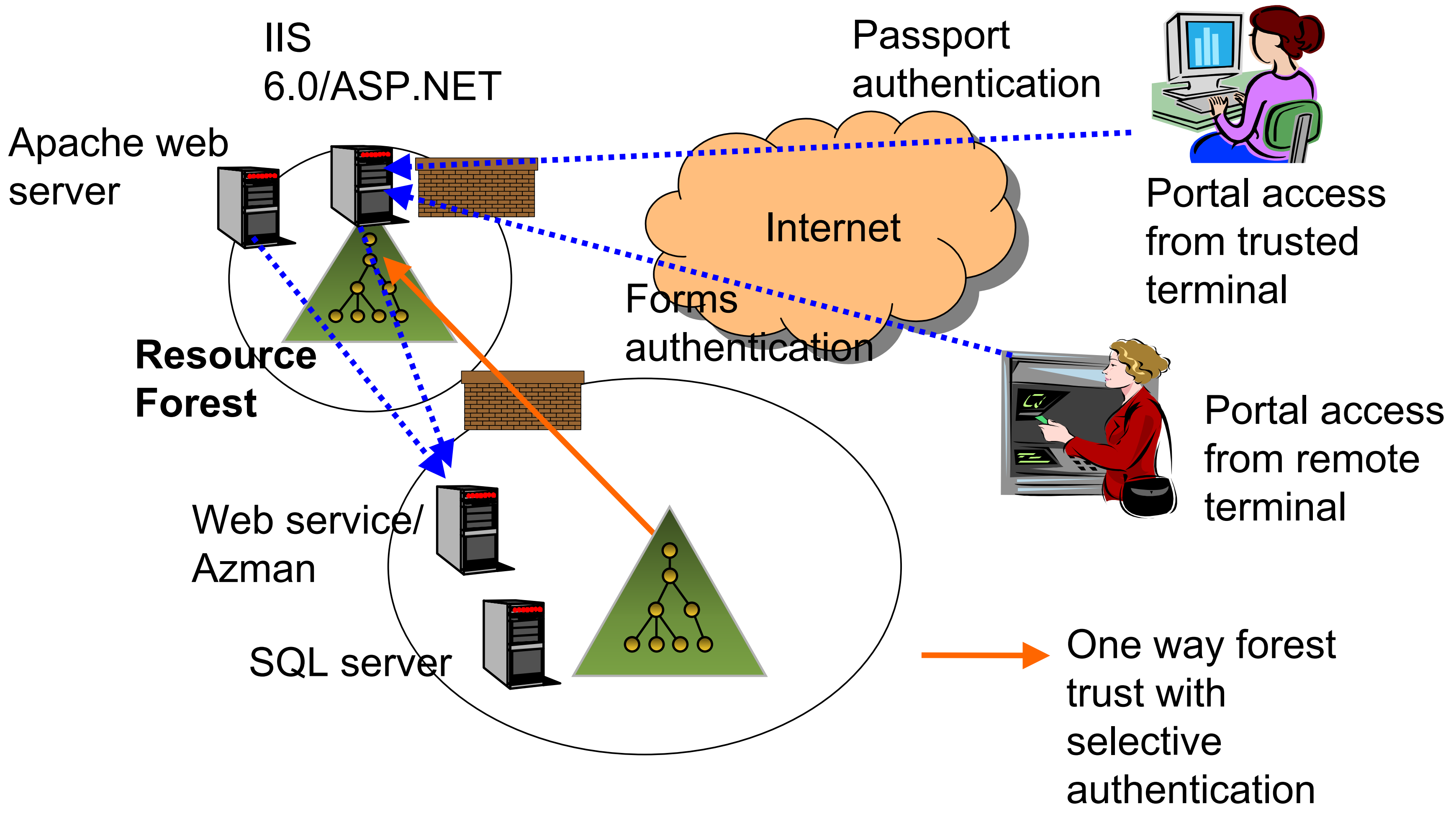
IV. Trust Mechanisms

- Forest Trust
  - Enables Kerberos Authn & Delegation
  - Easier trust management
- Selective Authentication
  - Restricts the scope of trust
- Trust Over Firewalls
  - Two new reg keys
    - Directory Services
    - Netlogin

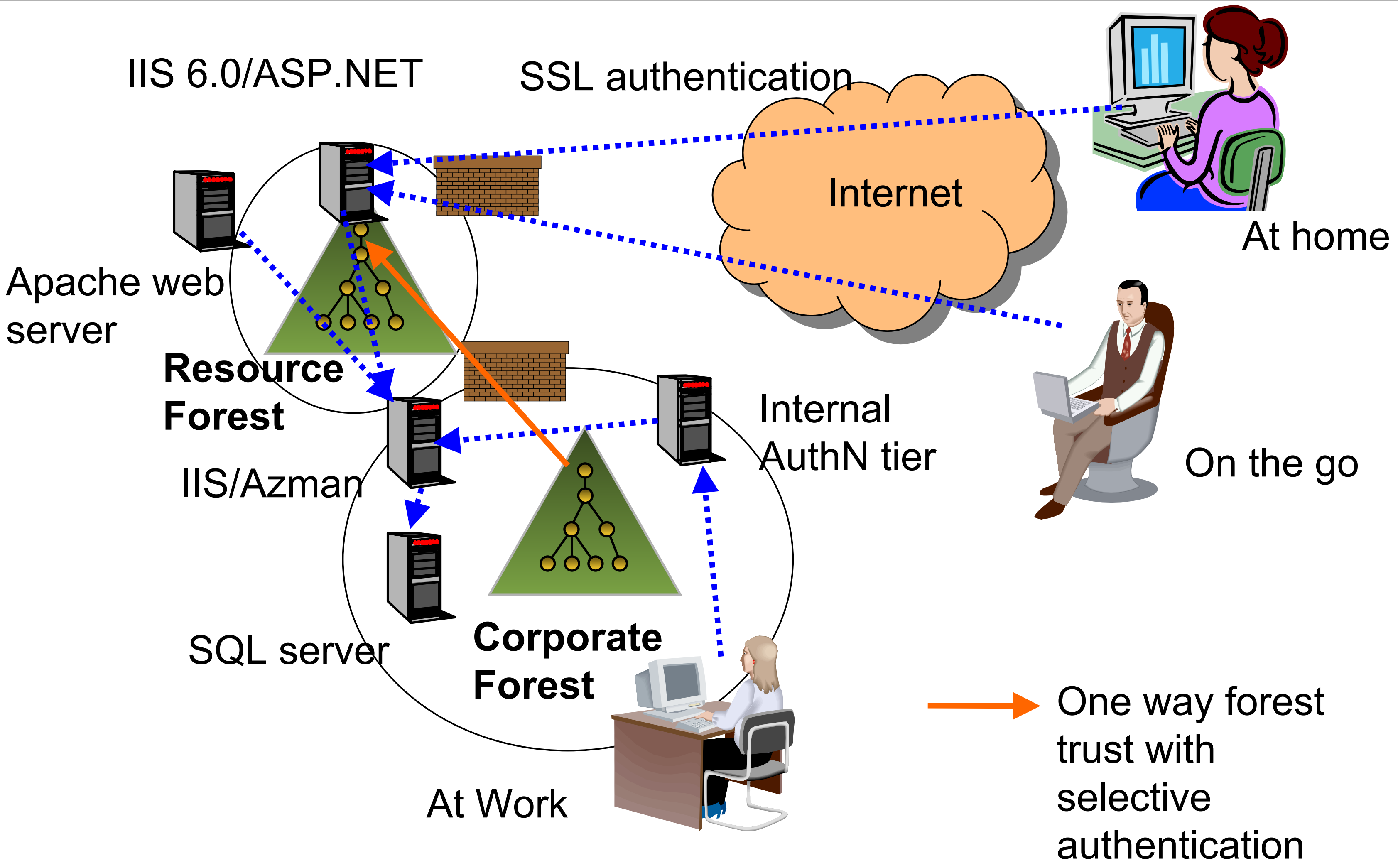
V. Putting it all together

- Customer Access
- Employee Access
- Partner Access
- Demo

Customer access

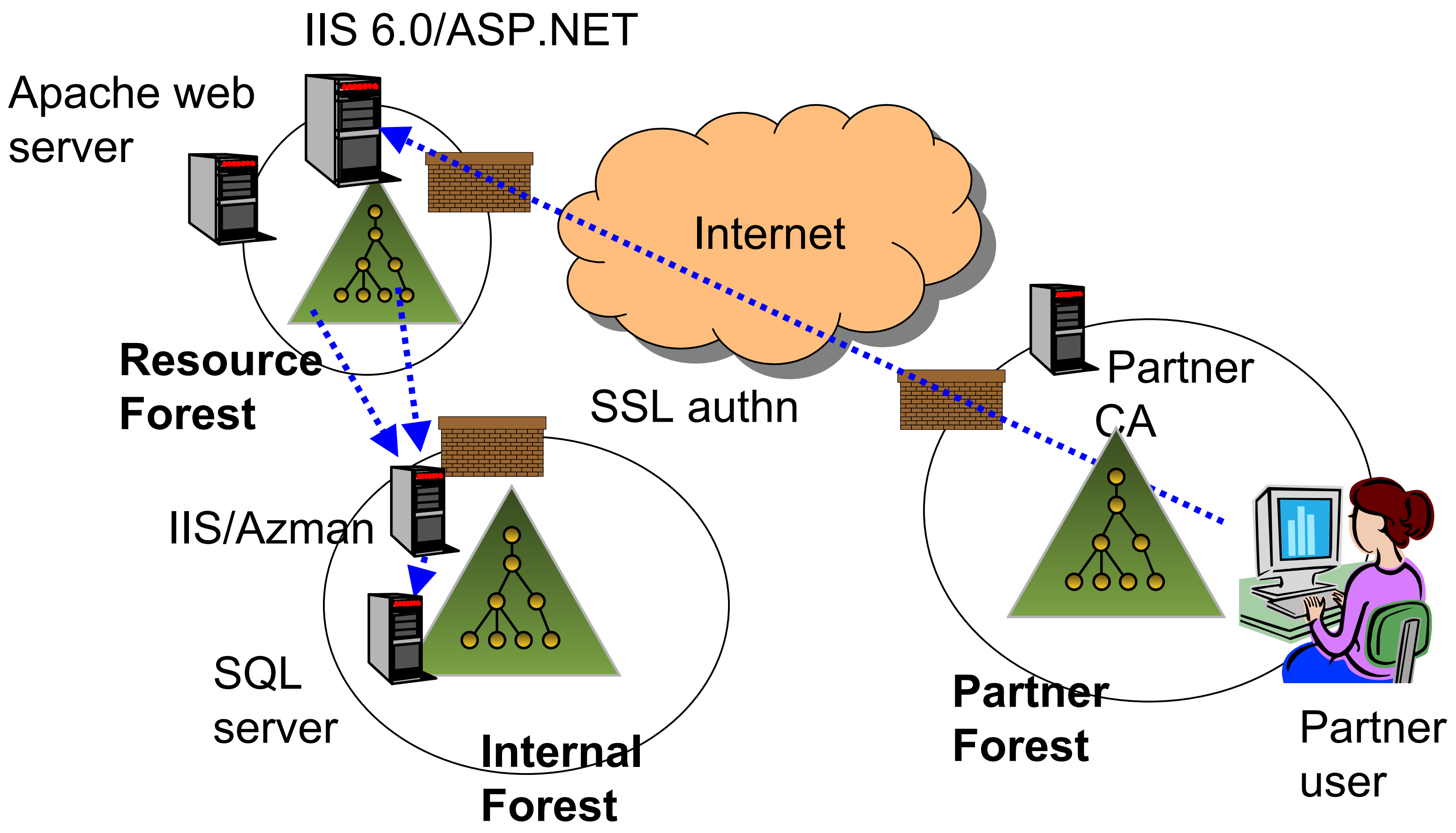


Employee access

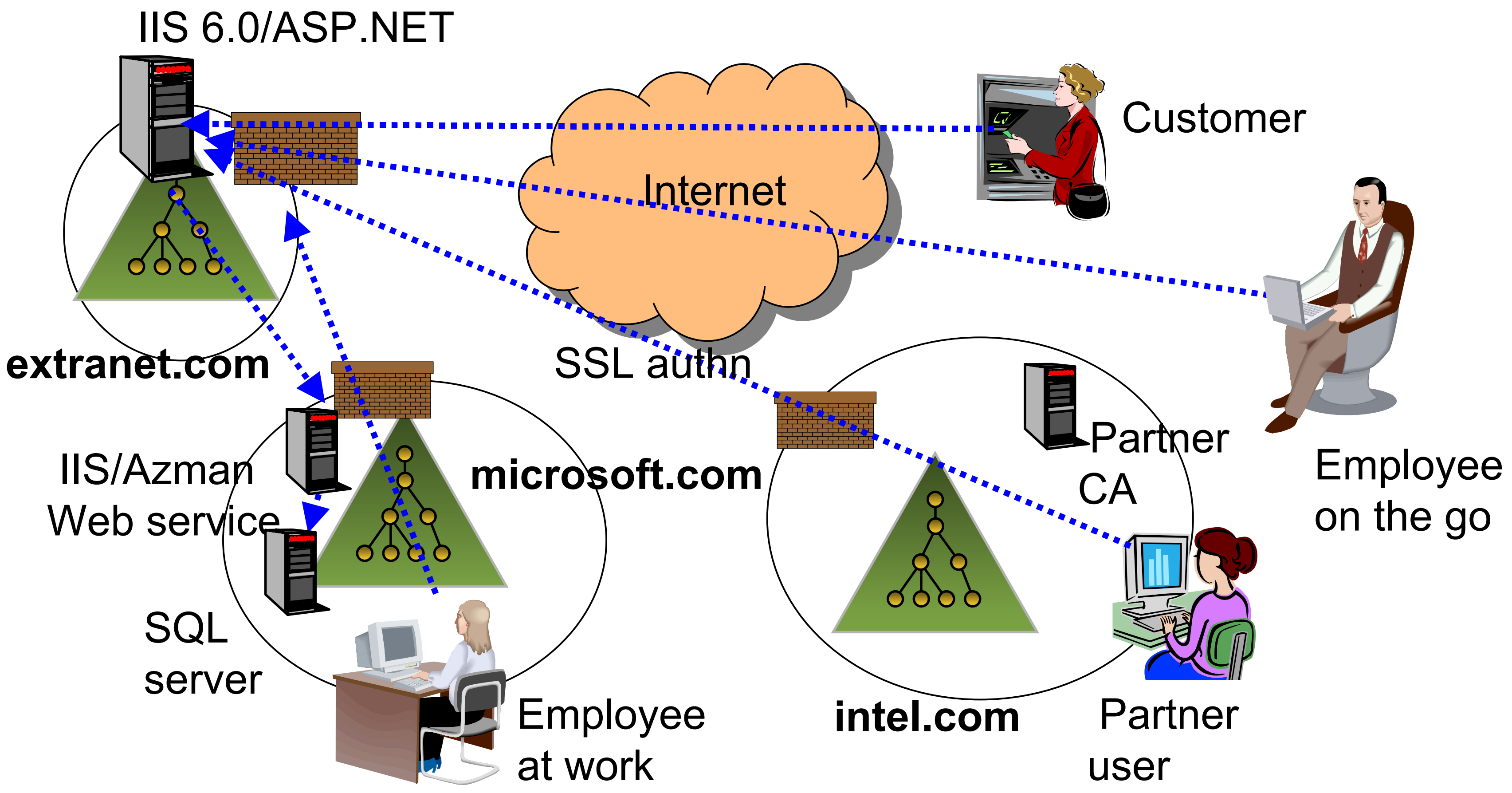


→ One way forest trust with selective authentication

Partner access



Demo


IIS 6.0/ASP.NET
Internet
Customer
extranet.com
SSL authn
IIS/Azman Web service
microsoft.com
Partner CA
Employee on the go
SQL server
Employee at work
intel.com
Partner user

Resources

- Windows Server 2003 Security Whitepapers
  http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/default.asp

- IETF RFCs – www.ietf.org
  - 1510 Kerberos
  - 2478 Negotiate
  - 2246 TLS

HP WORLD 2003

Solutions and Technology Conference & Expo

Interex, Encompass and HP bring you a powerful new HP World.

interex shared knowledge • shared power

encompass AN HP USER GROUP

hp invent