



Mississippi State
UNIVERSITY

J. A. “Drew” Hamilton, Jr., Ph.D.
Director, Distributed Analytics & Security Institute
Director, Center for Cyber Innovation
Professor, Computer Science & Engineering

CCI
Post Office Box 9627
Mississippi State, MS 39762

Voice: (662) 325-2294
Fax: (662) 325-7692
hamilton@cci.msstate.edu



Mississippi State University Center for Cyber Innovation

Domain 5 Identity and Access Management



Outline

(Controlling Access and Managing Identity) 13%

- **Physical and logical assets control**
- **Identification and authentication of people and devices**
- **Identity as a service (e.g. cloud identity)**
- **Third-party identity services (e.g. on-premise)**
- **Access control attacks**
- **Identity and access provisioning lifecycle (e.g. provisioning review)**



Access Controls

Reference: John Illg

Reference: Shon Harris

**Reference: Network Startup Resource
Center**



Basic Access Control Key Terms

- **Access controls protect against unauthorized access from any user or system to another user or system, regardless of whether this is physical or logical access.**
- **Access - the flow of information from a subject to an object.**
- **Objects - passive entities that contain data**
- **Subjects - request access to objects or data within objects.**
- **Authentication – second piece of credentials (password/key)**
- **Identification – prove you are who you claim to be (username/account number)**
- **Authorization – system check that you have correct rights/privileges to access requested information**
- **Accountability – system check that use of the requested information does not violate guidelines (audit logs, etc.)**
- **Logical Access Controls – tools used to ensure authentication, identification, authorization, and accountability**
- **Race Condition – occurs when processes carry out tasks on a shared resource in an incorrect order.**



Unix Users and Groups

- **Unix/Linux understands Users and Groups**
- **A user can belong to several groups**
- **A file can belong to only one user and one group at a time**
- **A particular user, the superuser “root” has extra privileges (uid = “0” in /etc/passwd)**
- **Only root can change the ownership of a file**



Program Execution

A program may be run by a user, when the system starts or by another process.

Before the program can execute the kernel inspects several things:

- Is the file containing the program accessible to the user or group of the process that wants to run it?**
- Does the file containing the program permit execution by that user or group (or anybody)?**
- In most cases, while executing, a program inherits the privileges of the user/process who started it.**



Unix File Permissions

- **In Unix, files and folders can be set up so that only specific users can view, modify, or run them.**
 - **For instance, you might wish to share an important file with other users, but do not want those users to be able to edit the file.**
- **Unix controls access to files on your computer through a system of “permissions.”**
 - **Permissions are settings configured to control exactly how files on your computer are accessed and used.**



File Permissions

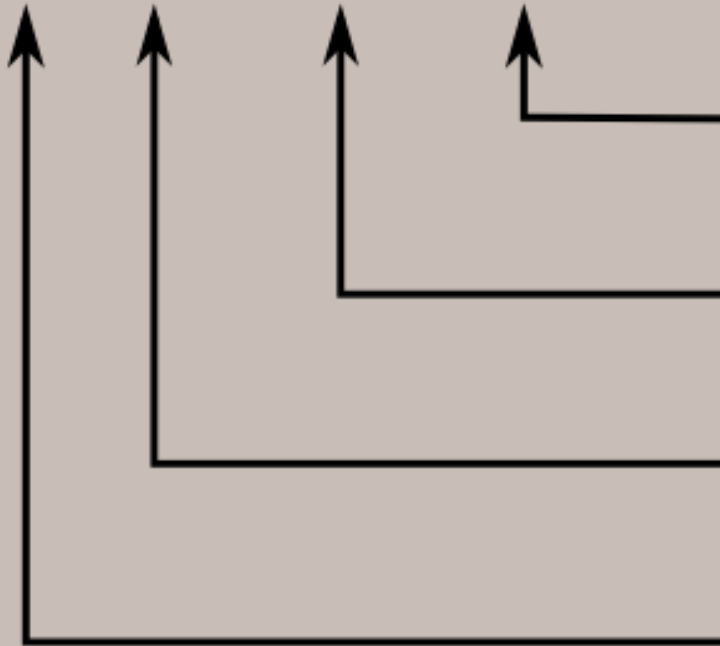
- On a Linux system, each file and directory is assigned access rights for the owner of the file, the members of a group of related users, and everybody else. Rights can be assigned to read a file, to write a file, and to execute a file (i.e., run the file as a program).
- To see the permission settings for a file, we can use the `ls -l` command. As an example, we will look at the `bash` program which is located in the `/bin` directory:

```
me@linuxbox me$ ls -l /bin/bash
-rwxr-xr-x 1 root root 316848 Feb 27 2000 /bin/bash
```
- Here we can see:
 - The file `/bin/bash` is owned by user `"root"`
 - The super user has the right to read, write, and execute this file
 - The file is owned by the group `"root"`
 - Members of the group `"root"` can also read and execute this file
 - Everybody else can read and execute this file



Typical file permissions

- rwx rwx rwx



Read, write, and execute permissions for all other users.

Read, write, and execute permissions for the group owner of the file.

Read, write, and execute permissions for the file owner.

File type:
- indicates regular file
d indicates directory



Absolute Mode File Permissions

Absolute mode:

We use octal (base eight) values represented like this:

<u>Letter</u>	<u>Permission</u>	<u>Value</u>
R	read	4
W	write	2
X	execute	1
-	none	0

For each column, User, Group or Other you can set values from 0 to 7. Here is what each means:

0= ---	1= --x	2= -w-	3= -wx
4= r--	5= r-x	6= rw-	7= rwx



Files common settings

Value	Meaning
777	(rwxrwxrwx) No restrictions on permissions. Anybody may do anything. Generally not a desirable setting.
755	(rwxr-xr-x) The file's owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.
700	(rwx-----) The file's owner may read, write, and execute the file. Nobody else has any rights. This setting is useful for programs that only the owner may use and must be kept private from others.
666	(rw-rw-rw-) All users may read and write the file.
644	(rw-r--r--) The owner may read and write a file, while all others may only read the file. A common setting for data files that everybody may read, but only the owner may change.
600	(rw-----) The owner may read and write a file. All others have no rights. A common setting for data files that the owner wants to keep private.

Directory Listing of Permissions

```
-r-xr-xr-x  1  root    root    68524  2011-12-19 07:18 /usr/bin/top
```

File Name
Modification Time/Date
Size (in bytes)
Group
Owner
"link count"
File Permissions

Group

The name of the group that has permissions in addition to the file's owner.

Owner

The name of the user who owns the file.

File Permissions

The first character is the type of file. A "-" indicates a regular (ordinary) file. A "d" indicate a directory. Second set of 3 characters represent the read, write, and execution rights of the file's owner. Next 3 represent the rights of the file's group, and the final 3 represent the rights granted to everybody else.

(Example modified from <http://www.linuxcommand.org/lts0030.php>)

Access Rights

- Files are owned by a *user* and a *group* (ownership)
- Files have permissions for the user, the group, and *other*
- “*other*” permission is often referred to as “world”
- The permissions are *Read, Write* and *Execute* (R, W, X)
- The user who owns a file is always allowed to change its permissions



Chmod – changing file permissions

- The chmod command is used to change the permissions of a file or directory. To use it, you specify the desired permission settings and the file or files that you wish to modify. There are two ways to specify the permissions. In this lesson we will focus on one of these, called the *octal notation* method.
- Here's how it works:
 - `rwX rwX rwX = 111 111 111`
 - `rw- rw- rw- = 110 110 110`
 - `rwX --- --- = 111 000 000`
- and so on... `rwX = 111 in binary = 7` `rw- = 110 in binary = 6` `r-x = 101 in binary = 5` `r-- = 100 in binary = 4`
- Now, if you represent each of the three sets of permissions (owner, group, and other) as a single digit, you have a pretty convenient way of expressing the possible permissions settings. For example, if we wanted to set `some_file` to have read and write permission for the owner, but wanted to keep the file private from others, we would:

```
me@linuxbox me$ chmod 600 some_file
```



CHMOD and File Permissions

There are two ways to set permissions when using the chmod command:

Symbolic mode:

testfile has permissions of `-r--r--r--`

U G O*

\$ `chmod g+x testfile` ==> `-r--r-xr--`

\$ `chmod u+wx testfile` ==> `-rwxr-xr--`

\$ `chmod ug-x testfile` ==> `-rw--r--r--`

U=user, G=group, O=other (world)



Numeric Mode File Permissions

Numeric mode cont:

Example index.html file with typical permission values:

```
$ chmod 755 index.html
```

```
$ ls -l index.html
```

```
-rwxr-xr-x  1 root  wheel  0 May 24 06:20 index.html
```

```
$ chmod 644 index.html
```

```
$ ls -l index.html
```

```
-rw-r--r--  1 root  wheel  0 May 24 06:20 index.html
```



Inherited Permissions

Two critical points:

- 1. The permissions of a directory affect whether someone can see its contents or add or remove files in it.**
- 2. The permissions on a file determine what a user can do to the data in the file.**

Example:

If you don't have write permission for a directory, then you can't delete a file in the directory. If you have write access to the file you can update the data in the file.



Directory Permissions

- The **chmod** command can also be used to control the access permissions for directories. Again, we can use the octal notation to set permissions, but the meaning of the **r**, **w**, and **x** attributes is different:
 - r** - Allows the contents of the directory to be listed if the **x** attribute is also set.
 - w** - Allows files within the directory to be created, deleted, or renamed if the **x** attribute is also set.
 - x** - Allows a directory to be entered (i.e. `cd dir`).



Unix Directory Listing Output

When looking at the output from “`ls -l`” in the first column you might see:

```
d = directory  
- = regular file  
l = symbolic link  
s = Unix domain socket  
p = named pipe  
c = character device file  
b = block device file
```



Managing ownership

- Anytime a user creates a new file or directory, his or her user account is assigned as that file or directory's "owner."
- For example, suppose the **ken** user logs in to her Linux system and creates a file named `linux_introduction.odt` using OpenOffice.org in home directory.
- Because she created this file, ken is automatically assigned ownership of `linux_introduction.odt`.



Chown - Changing File Ownership

- You can change the owner of a file by using the chown command. Here's an example: Suppose I wanted to change the owner of some_file from "me" to "you". I could:

```
me@linuxbox me$ su
```

```
Password:
```

```
root@linuxbox me# chown you some_file
```

```
root@linuxbox me# exit
```

```
me@linuxbox me$
```



How ownership works

- You can specify a different user and/or group as the owner of a given file or directory.
- To change the user who owns a file, you must be logged in as root.
- To change the group that owns a file, you must be logged in as root or as the user who currently owns the file.
 - ✓ Using `chown`
 - ✓ Using `chgrp`
 - ✓ You can also view file ownership from the command line using the `ls -l` command



Using chown

- The chown utility can be used to change the **user** or **group** that owns a file or **directory**.

Syntax `chown user.group file or directory.`

Example: If I wanted to change the file's owner to the **ken1** user, I would enter

```
chown ken1 /tmp/myfile.txt
```

–If I wanted to change this to the users group, of which **users** is a member, I would enter

```
chown .users /tmp/myfile.txt
```

Notice that I used a period (.) before the group name to tell chown that the entity specified is a group, not a user account.

Ex: `chown student.users /tmp/myfile.txt`

Note: You can use the **–R** option with chown to change ownership on many files at once recursively.



Using chgrp

- In addition to chown, you can also use **chgrp** to change the group that owns a file or directory.
- **Syntax:** `chgrp group file (or directory)`
- **Example:** `chgrp student /tmp/newfile.txt.`



Changing Group Ownership

- The group ownership of a file or directory may be changed with `chgrp`. This command is used like this:
 - `[me@linuxbox me]$ chgrp new_group some_file`
- In the example above, we changed the group ownership of `some_file` from its previous group to "new_group". You must be the owner of the file or directory to perform a `chgrp`.



Working with default permissions

- By default, Linux assigns **rw-rw-rw-** (**666**) permissions to every file whenever it is created in the file system.
- It also assigns **rwxrwxrwx** permissions to every directory created in the file system. It also assigns **rwxrwxrwx** permissions to every directory created in the file system.



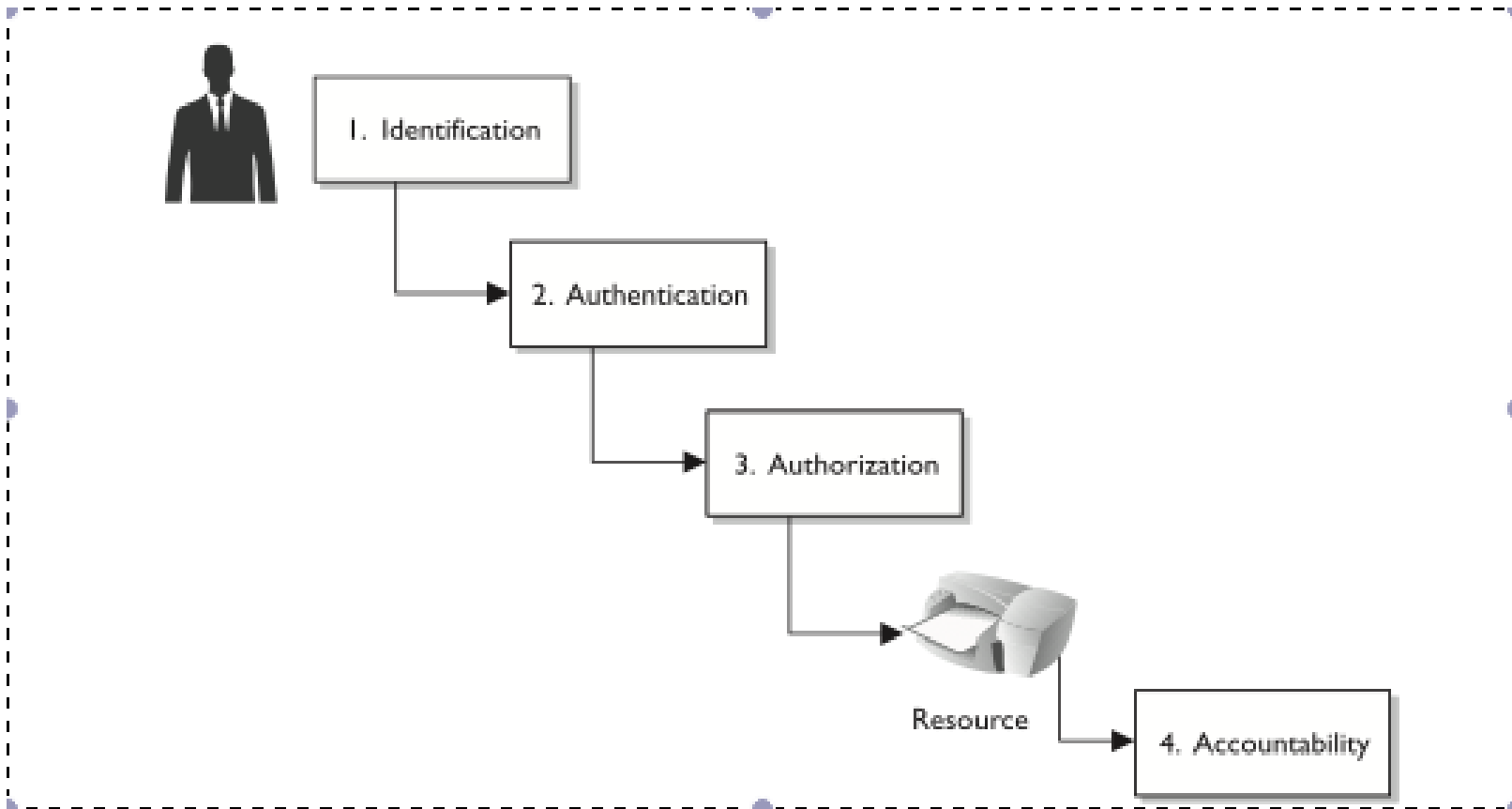
Working with default permissions

- To increase the overall security of the system, Linux uses a variable called **umask** to automatically remove permissions from the default mode whenever a file or directory is created in the file system. The value of umask is a three-digit number
- For most Linux distributions, the default value of umask is 022. Each digit represents a numeric permission value to be removed. **The first** digit references Owner, **the second** references Group, **the last** references Other.



Basic Access Control: How it Works

- Possible Test Question: Order of Operations is important!



Factors of Authentication

- Authentication can also be referred to as “Verification 1:1”. Both ask the same question: Are you who you say you are?
- How do we make sure someone is who they say they are? Test them and make sure that they know something that they should know (authentication by knowledge), have something that they should have (authentication by ownership), or have a characteristic that they should have (authentication by characteristic).
- Strong Authentication (or two – factor authentication) includes two of these factors.
- Possible test question → Biometrics alone cannot be a form of strong authentication: Only one factor of authentication checked.



Secure Identities and Identity Management (IdM)

- “Secure identities” have three key aspects: uniqueness, nondescriptness, and issuance.
- Uniqueness – identifier is specific to the individual and no two identifiers may be the same
- Nondescriptness – no piece of credentials should give away who owns the account
- Issuance – identities have been provided by an outside authority

Identity Management

- Identity Management (IdM) technologies help to identify, authenticate, and authorize activities
- High levels of IdM complexity are forcing out traditional IdM manual processes and replacing them with automated ones.
- Many IdM solutions, but these are important for CISSP: Directories, Web Access Management, Password Management, Legacy Single Sign – On, Account Management, and Profile Update

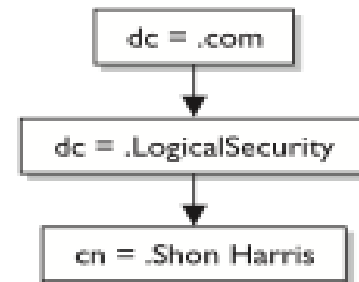


Directories

- Generally based on a combination of a database format (X.500, etc.) and a protocol that facilitates user interaction with the directory (LDAP, etc.)
- Objects are managed by a “directory service” that allows administrators to configure and manage their security settings
- How does directory service organize things? → namespaces
- LDAP Method: distinguished names (dn) that are composed of common names (cn) and domain components (dc).

Example:

```
dn: cn=Shon Harris,dc=LogicalSecurity,dc=com
cn: Shon Harris
```

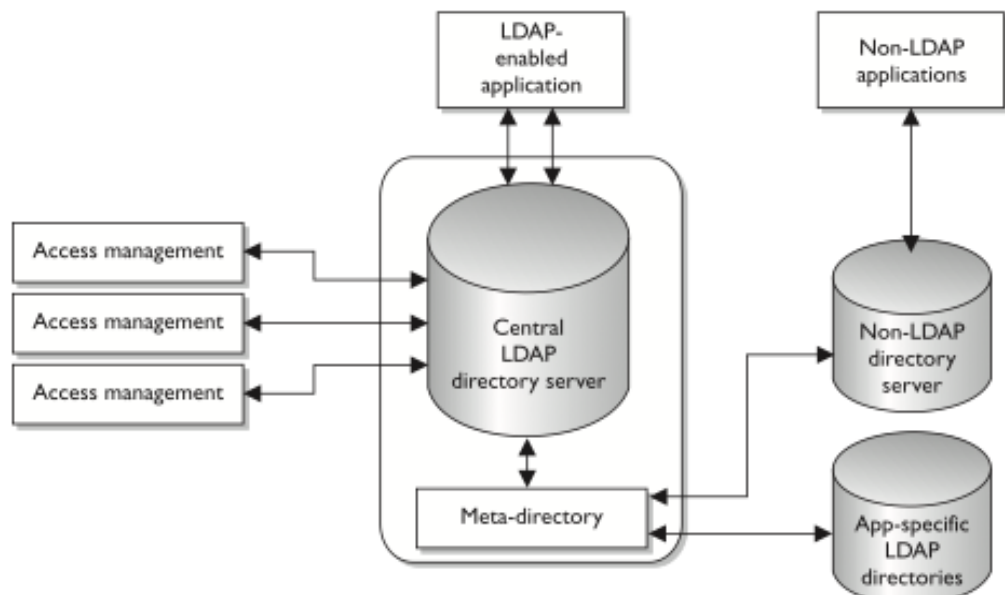


- Are there problems with directories? Yes → Legacy systems may not support current directory software.



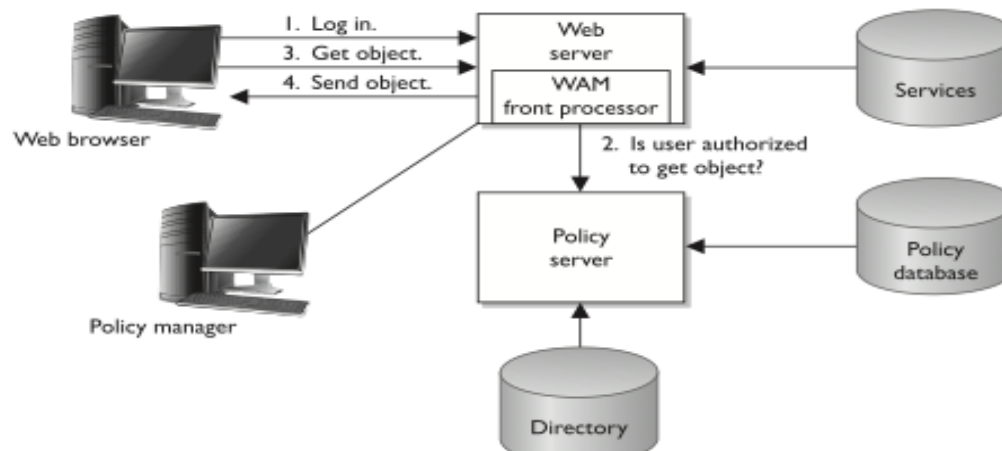
Directories and IdM

- Directories used for IdM are specialized for reading and searching.
- Benefits of Directories in IdM? Single point of contact → ask the Directory for anything
- Meta – Directories are used to help accomplish the end of accumulating user IdM info in one place. This info may be scattered, but the meta-directory gathers and stores it in one central location. Virtual Directories serve the same function as meta - directories. They however, do not serve as a secondary container for same data. Instead, they contain pointers to the data located elsewhere.
- What it looks like:



Web Access Management (WAM)

- Web access management control what users can access web – based assets with a web browser. WAM is experiencing a growth in use because of the increased importance of e – commerce.
- WAM follows the same basic principles for IdM as discussed earlier (identification, authentication, authorization) with the WAM server contacting external policy managers, servers, and databases as applicable in one's network to process authorization information. However, monitoring in web is different. It is done through the issuing of cookies (which include important user information like their access rights and personal preferences) after setting up encryption in order to protect cookie data. The WAM server sends the cookie to the client and requires the client to present the cookie whenever the client attempts to access more data or after certain time intervals have passed.
- What it looks like:



Password Management

- **Big Problem – Users forget passwords and require password to be reset. There are three major automated solutions for this that help reduce the need for dedicated human workforce: Password Synchronization, Self – Service Password Reset, and Assisted Password Reset**
- **Password Synchronization → force user to maintain just one complex password that updates all of his other passwords automatically (has obvious problems and obvious benefits)**
- **Self – Service Password Reset → password resets are performed using already authenticated external accounts (links sent via e – mail, etc.) or through authorization questions. If the test is passed, the user can reset his password.**
- **Assisted Password Reset → aid help – desk employees in performing password resets by providing a platform to authenticate users prior to their interaction with the help – desk (usually via personal questions) and forcing the user to change their password after the reset so that the help – desk employee will not know what the password is**



Legacy Single Sign – On And Account Management

- **Single Sign On (SSO technologies) authenticate one user at a time with no need for re-authentication. SSO technologies are different from password synchronization because a password is sent to ONE authentication system which then communicates with the other authentication systems across the network. In password synchronization, you must login to each different authentication system within the network separately (even though this log-in will be with the same password each time since updating one password updates all of the rest)**
- **Possible test question → Cons of SSO? Expensive and provides a single point of failure. Shut down the SSO, and everything goes down.**

Account Management

- **Account management deals with the creation and deletion of user accounts along with the modification of the privileges of those accounts. Often, this is done manually, which is not ideal. Administrators may provide too much access and become bogged down with the workload from changing user accounts across multiple systems. Software helps alleviate both problems by changing user accounts across multiple systems and providing a access request framework.**



Provisioning And Profile Updates

- **How does everything discussed tie together?**
 1. Information is pushed from an HR database to a directory (the Identity Repository). Related parties (bosses, etc.) will be notified if necessary.
 2. Attributes for different identities will accumulate in the identity repository as the user gains access to more and more information.
 3. These attributes will be accessed by IdM solutions in order to test user authorization.

Profile Updates

- Other information about the user may be stored in addition to authorization information. (Date of birth, home address, etc.) When this info is associated with an identity, it is called a Profile. Customer Relationship Management Systems (CRMs) allow a user to modify those parts of the profile that they should be able to view (this is called self - service).



Federations and Federated Identities

- **Companies now work together in order to provide complementary services to users (travel sites in particular apply here).**
- **Because of this, user identities may be shared between companies. Federated Identities are especially useful, as they are linked to from multiple companies – no need for the synchronization of directories.**



Authentication Methods

Markup Languages and IdM

- **Service Provisioning Markup Language (SPML)** was built on XML to aid with authentication. SPML makes sure that requests are authenticated by the sender before allowing access to a service.
- **eXtensible Access Control Markup Language (XACML)** is used to ensure that both sides follow the same security policies.

Biometrics Intro

- Analyzes a (supposedly) unique behavior or attribute to verify a user.
- Biometrics are more generally more expensive, sophisticated, and complex than other authentication solutions.
- Can be broken up into two different categories: physiological (“what you are”) and behavioral (“what you do”)



Biometrics and Error Rates

- **Type I Errors: authorized individual is rejected (also called the false rejection rate)**
- **Type II Errors: impostor is authorized (also called the false acceptance rate)**
- **Most important metric regarding biometrics is the Crossover Error Rate (or CER). CER is calculated as a percentage. It represents the point where the proportion of Type I Errors is equivalent to that of Type II Errors as you up the sensitivity of the system. Lower is better when you have CERs. (A CER of 3 is more accurate than a CER of 4).**
- **Obviously, there are situations where Type II Errors are more critical (military operations) and Type I Errors are more critical (more loose commercial settings).**
- **Because of Type I errors, Biometrics can provide an annoyance to the user, and therefore may not be excepted in case of extreme need.**



Biometric Methods

- The well known have no need for explanation: fingerprinting and retina scans.
- Others are similar to these: Palm Scans look at the placement and shape ridges, creases, and grooves of the palm. Hand Geometry measures the shape, length, width of the hands along with various amalgamate ratios. Hand Topography looks at the peaks and valleys of a hand...its curvature, like one would approach regular topography. Iris scans analyze unique colors and rings within the iris.
- Also used are behavioral based biometrics. Possible test question → Signature Dynamics doesn't just match a signature to one stored in a database; it also matches up the tempo and pressure of the electronic pen stroke actively.
- Keystroke Dynamics uses this same tempo and rhythm principle to authenticate based on a typing sample. Voice Print checks subtle speech patterns.
- Problems? What if your biometrics change? A cold could change your voice, you could have a wrist injury, etc.



Password Policy

- Passwords are commonly used, but have an obvious weakness: the users creating and maintaining them.
- Common attacks on passwords include electronic monitoring, password file retrieval, brute force attacks, dictionary attacks, social engineering, and rainbow tables. (These should all be self explanatory)
- Password policies must force users to address major concerns raised by all of the above.
- Education is the key: make users allies along the security front.

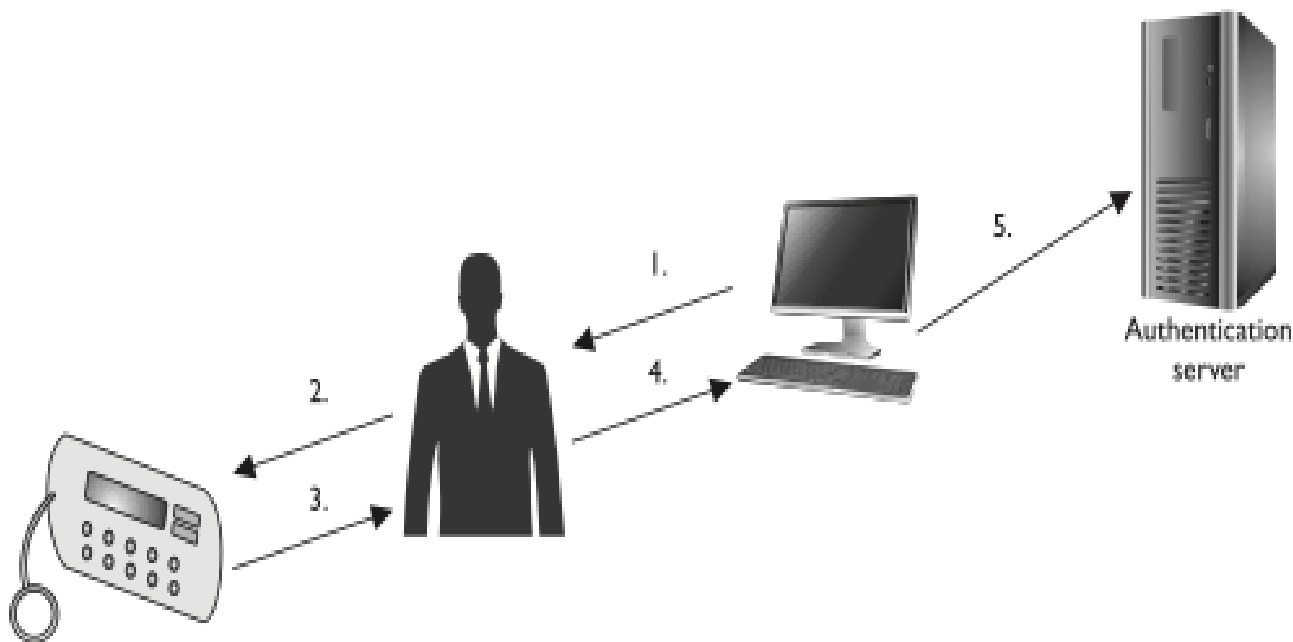


Parts of a Successful Password Policy

- **Password Checkers** – test the viability of the password using hackers own cracking methods in order to prevent against brute force attacks, dictionary attacks, and rainbow tables.
- **Password Hashing** – help prevent against replay attacks and password file retrieval by using a hashed password in place of the real password. Salts may be used in order to add an additional layer of randomness that protects against the brute forcing of the hashed password.
- **Password Aging** – the password expires after a certain date in order to prevent against the continued malicious use of a password and against really involved brute force attacks. Password history may also be kept in order to prevent the user from reusing passwords.
- **Limited Login Attempts** – limit the number of times the user can attempt to login to help prevent dictionary or brute force style attacks
- **Cognitive Passwords** – questions asked of the user
- **One – Time Passwords** – good only once → extra protection
- **Token Device** – external provider of one-time passwords. Can be **Synchronous** – where time is used as the seed for the one time password or **Nonsynchronous** where the user is sent a random Nonce value by the authentication server, types it into their token device, and the device sends the encrypted nonce value back to the authentication server for authentication.



Asynchronous Token Device: How it Works



1. Challenge value displayed on workstation.
2. User enters challenge value and PIN into token device.
3. Token device presents a different value to the user.
4. User enters new value into the workstation.
5. Value sent to authentication service on server.
6. Authentication service is expecting a specific value.
7. User is authenticated and allowed access to workstation.



Password Policy Parts (cont.)

- **Digital Signatures** – Uses private/public key cryptography. Private keys are unique to each user. A digital signature is a hash value encrypted with a users private key.
- **Passphrases** – Sequence of characters longer than a password used for authentication. Can be turned into a Virtual Password (passphrase is modified into a cryptographic key) if entered into an application.
- **Memory Cards** – Store authentication information, but do not process it
- **Smart Cards** – Stores authentication information and has a microprocessor to perform authentication calculations. Contact Smart Cards have a contact that allows the outside to provide the power for the smart card, while Contactless smart cards use antennas to receive the power wirelessly without a contact. Smart cards do not become active until the proper PIN is entered. Thus, they can be a form of two – factor (strong) authentication.

Smart Card Attacks

- **Fault Generation** – the manipulation of physical inputs (voltage, etc.) to the smart card in order to monitor changes in the encryption function in preparation for an attack on the keys themselves.
- **Side – Channel Attacks** – Watch how the smart card works using differential power analysis (watch power emissions) and electromagnetic analysis (watch frequencies emitted) and timing. If information about the cards themselves are known (how encryption algorithm is implemented, processing power of the card), side – channel attacks can lead the attacker to the key.
- **Smart Card interoperability** can also be a major problem. ISO/IEC 14443 was used to help solve the problem



Authorization

- Up until this point, we have discussed mostly identification and authentication. Authorization is a distinct field of its own.
- Major question: → Who should be able to access what data?
- Two main factors are involved: trust and need to know.
- 5 Systematic Criteria can help aid in building an authorization policy – roles (authorize based on job assignment), groups (authorize by placing them in a general group that has similar need to know/trust characteristics as they do), location (authorize by restricting access to only those with the correct physical/logical location), time of day (only authorize users during standard hours), and transaction type (only authorize access to data that has certain characteristics → money less than \$2,000, etc.)
- Always default to not authorizing anyone if you aren't sure.
- Need to know is important; don't allow anyone who doesn't need the data to access it.



Single Sign On Technologies

- **Discussed already: one login allows you access to everything.**
- **Not used universally, but still chunks of single sign-on area can be found within a corporate environment.**
- **Thin Clients don't have many (if any) peripherals and hardware, instead only existing to request resources, services, etc., from the central server mainframe.**
- **Kerberos and SESAME are the single sign – on technologies tested by the CISSP exam.**



Kerberos

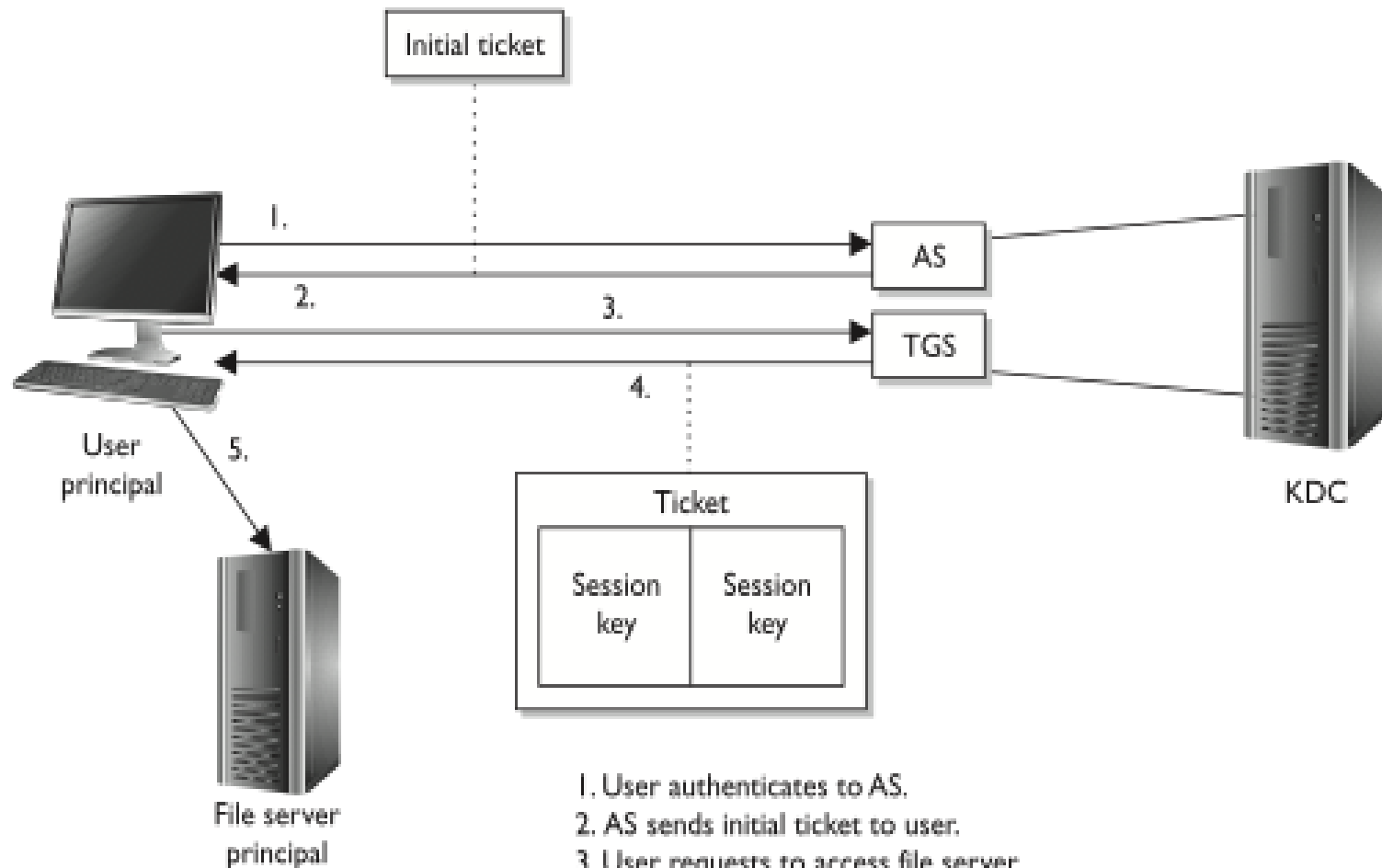
- **Key Distribution Center (KDC)** holds all key information for all users and services.
- The KDC provides a security service to Principals who can be users, applications, or network services. The KDC must have account for and share a secret key with each principal. Users have their passwords transformed into keys. A set of principals on the KDC is called a Realm. There can be 1:* realms on a KDC.
- A ticket is generated by the Ticket Granting Service (TGS) on the KDC and given to a principal when that principal needs to authenticate to another principal.

Kerberos Authentication Process

- 1: User authenticates to the authentication service on the KDC. 2: User is sent a ticket granting ticket (TGT) which signifies that authentication has occurred. 3: User requests access to another principal. 4: TGT sent to the TGS. 5: TGS creates a new ticket for the user which includes session keys. 6: User sends ticket to other principal and grabs session keys. 7: Other principal grants access.



Kerberos: What it Looks Like



1. User authenticates to AS.
2. AS sends initial ticket to user.
3. User requests to access file server.
4. TGS creates new ticket with session keys.
5. User extracts one session key and sends ticket to file server.



Weaknesses of Kerberos

- **KDC is single point of failure and a single choke point for requests which presents scalability concerns.**
- **Secret and Session keys are stored on user workstations and can be captured.**
- **Vulnerable to dictionary attacks and other exhaustive measures.**
- **Network traffic is not protected if encryption is not enabled.**
- **Short keys can be brute forced (duh!)**
- **All involved clocks must be synchronized.**

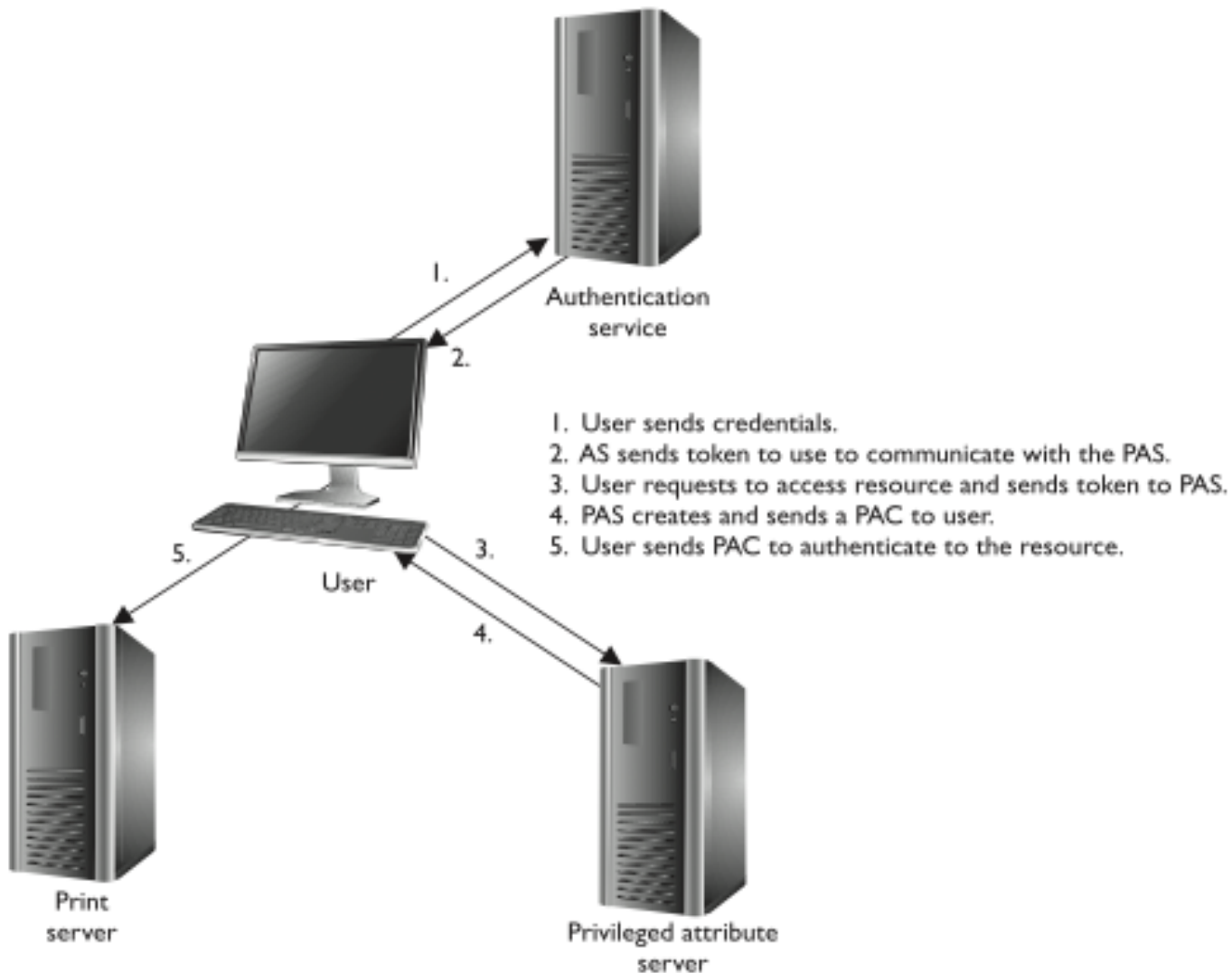


SESAME

- **Full Name: Secure European System for Applications in a Multi-vendor Environment**
- **Idea: extend functionality of Kerberos to protect against its weaknesses by using both symmetric and asymmetric key encryption (Kerberos is symmetric only).**
- **Terminology Changes: Privileged Attribute Certificates (PACs) authenticate subjects to objects in place of tickets. Privileged Attribute Servers (PASs) deal with the issuing of PACs instead of KDCs issuing tickets. KDC responsibility is broken up between an AS and the PAS instead of centrally located as with the KDC**

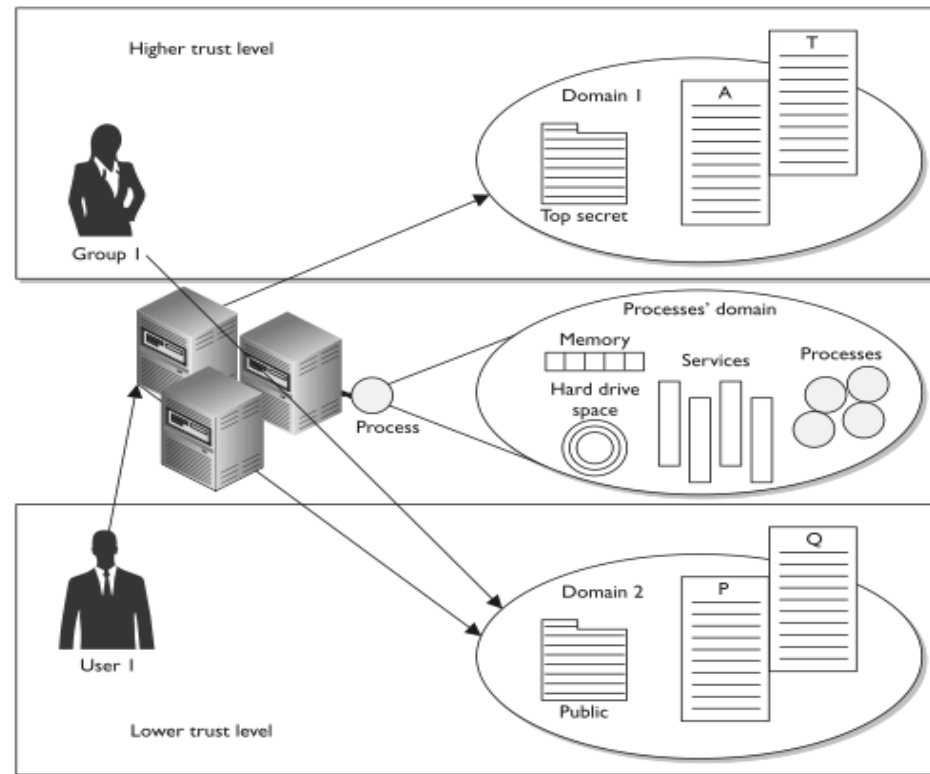


SESAME: What it Looks Like



Security Domains

- Domain – the set of possible resources available to a subject.
- Security Domains – group resources that have the same general trust level together should be grouped in the same logical domain. Security domains can have a tree structure with one domain located within another domain.
- What it looks like:



Access Control Models

- **Access Control Models** are frameworks that determine **HOW** subjects are allowed to access objects.
- There are 3 main kinds: **Discretionary Access Control (DAC)**, **Mandatory Access Control (MAC)**, and **Nondiscretionary Access Control** (or role based access controls → **RBAC**).
- Generally built into OS kernels.
- **DAC** – Creator of the file is its owner. He/she determines the level of access of other users.
- **MAC** – Users/Owners do not have as much control, instead the OS is the final authority. If user/data clearances do not match in the context of the global security policy, it doesn't matter what anyone else thinks, the OS will not allow access. Security labels (or Sensitivity Labels) attached to each object allow for this clearance check to be performed. Sensitivity labels contain both a classification (clearance info) and categories (need to know specifications).



Role – Based Access Controls (RBACs)

- In an RBAC, a user is assigned a Role related to their job. Role determines what resources they can access. Roles are given more/less access depending on need to know bases. RBACs are more nimble than DACs, as roles are easy to assign.
- Session – a mapping between a user and a group of roles.
- Hierarchical RBACs create a chain of command where roles inherit privileges from roles higher up in the hierarchy.
- In Hierarchical RBACs, role access is an accumulation of rights: a doctor role may inherit from a nurse role (Book example!), etc.
- Two different ways to separate duties: Static Separation of Duty (SSD) through RBAC deters fraud by keeping certain privileges from being combined PERIOD. Dynamic Separation of Duties (DSD) through RBAC deters fraud by keeping certain privileges from being combined AT THE SAME TIME.



Access Control Techniques

- **Access Control Techniques and Technologies help support access control models.**
- **Rule – Based Access Controls (RBACs? No, dammit! I think acronyms should suffer an ignominious death) are identity INDEPENDENT ways to define how subjects and access objects. (no access between 2 and 5 pm, no matter the user, etc.)**
- **Constrained User Interfaces – restrict user access by limiting the commands users can perform or buttons they can press.**
- **Access Control Matrix – a table of subjects and objects indicating what each subject can do to each object**
- **Capability table – shows what access a single subject can have on a number of different objects.**
- **Access Control Lists – Lists of subjects that are authorized to use a specific subject. They map from the access control matrix to the object in question.**



Access Control Matrix in Action

Access Control Matrix

	Subject	File 1	File 2	File 3	File 4
	Larry	Read	Read, write	Read	Read, write
Capability	Curly	Full control	No access	Full control	Read
	Mo	Read, write	No access	Read	Full control
	Bob	Full control	Full control	No access	No access

ACL

Capability = row in matrix

ACL = column in matrix



Content (or Context?) Dependent Access Control

- **Content dependent access controls determine what access a subject can have on an object by what information that object contains.**
- **Context dependent access controls reviews state-based information before allowing access (the primary example is TCP communication)**



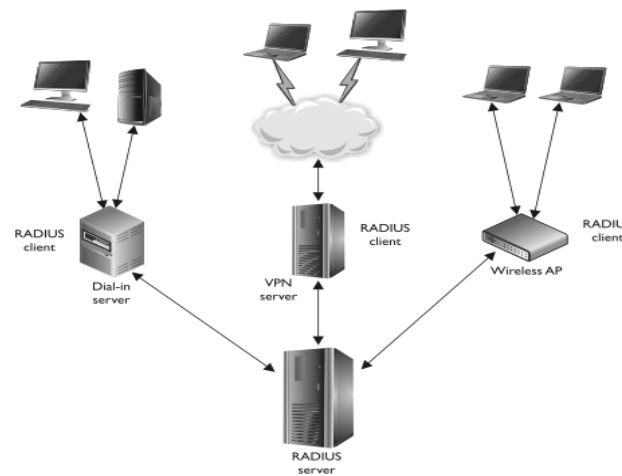
Access Control Administration

- **Main question: Now that I have all of this access control stuff, how do keep track of it all? Will I keep all of the related information in a central location or store it in a number of different locations?**
- **Central Access Control Administration → one entity is responsible for overseeing access to all corporate resources. 3 are covered in the book: RADIUS, TACACS, and Diameter (Ha!).**



RADIUS (Remote Authentication Dial – In Service)

- Used by most ISPs.
- Communication over PPP (point to point protocol)
- Can be used in a corporate environment for the authentication of remote users
- Authentication takes place like so:
 1. Handshake occurs and authentication protocol is agreed upon (PAP, CHAP, or EAP)
 2. Customer provides username/password
 3. if authenticated, customer given IP address, connection information, and allowed to access the internet
- What does it look like?: ->



TACACS+

- **TACACS+ - Basically RADIUS on TCP instead of UDP with the addition of full session encryption instead of username/password encryption**
- **TACACS+ also allows for closer control over what users can and cannot do.**
- **Rule of thumb: Use RADIUS for simple solutions and TACACS+ for larger, more complex environments.**



Diameter

- **Better than RADIUS; it has its own base protocol with extensions that allow for communication can work over a wide variety of protocols (VoIP, etc.), instead of just PPP/SLIP.**
- **Diameter is also peer-based, instead of a client/server architecture and can be made backwards compatible with RADIUS.**



Diameter

- **Decentralized Access Controls store access information in a number of places.**
- **This allows people closer to the situation to directly handle access control information.**
- **However, it can also result in conflicts of interests and other nightmares: one localized group fighting for access rights with another group; one user being deleted in one location and not another location.**



Access Control Layers

- **Access Control Layers give one more assurance that users do not get improper access, since they have to go through more than one layer of security.**
- **There are three main access control layers: Administrative Controls, Physical Controls, and Technical Controls**



Kinds of Administrative Controls

- **Administrative controls are created by administrators and form the top layer of access control; they are a skeleton that other controls build on. They are a sort of company policy.**
- **Personnel Controls – indicate how employee hirings, terminations, suspensions, etc. should be handled.**
- **Supervisory Structure – give each employee a supervisor who will be responsible for their actions; forces employers to have active interest in employee activities.**
- **Security – Awareness Training – make employees less stupid; less stupid is generally good.**
- **Testing – Test all security items regularly in order to ensure that management goals are achieved.**



Kinds of Physical Controls

- **Even in the internet age, restricting physical access is still important.**
- **Network Segregation – keep unrelated network equipment from being in the same room from each other.**
- **Perimeter Security – fences (electrified is good!), security guards, cameras. Get as paranoid as you please!**
- **Computer Controls – Hot glue the USB ports. (Conficker!)**
- **Work Area Separation – Keep accountants away from labs, etc.**
- **Cabling – Keep all of your miles of snaking cables out from underfoot.**
- **Control Zone – Keep different security zones physically as well as logically separated.**



Kinds of Technical Controls

- **Technical Controls are part of OSs, software security packages, etc.**
- **System Controls – RADIUS, Kerberos, TACACS+, etc.**
- **Network Architecture – Use DMZs and other internal network segregation to slow down network infiltration.**
- **Network Access – Firewall rules, etc.**
- **Encryption and Protocols**
- **Auditing tools – monitor user activity**



Access Control Functions

- Each of these control types just mentioned perform a specific function: deterrence, prevention, corrective, recovery, detective, compensating, or directive.
- Preventative measures form your backbone with other functions supporting the preventative measures in case of an attack.



Accountability

- **Main idea: Make sure users are responsible for what they do by using audit tools.**
- **These tools must: 1. Store their logs securely. 2. Keep the size of their logs down. 3. Protect their logs from unauthorized access. 4. Be reviewed by competent people. 5. Only allow administrators to delete logs. 6. Contain activities of all high – privileged accounts like root.**
- **A number of events can be a part of audit logs. These events (and the degree to which they are monitored) must be chosen carefully in order to try and support the above. The Clipping Level determines how often an event (failed login,etc.) must occur before it triggers an entry in the log.**
- **Reviewing audit information can be very daunting if one doesn't have the tools. Audit Reduction Tools parse out events that are unrelated to the attack that you are investigating.**
- **Keystroke Monitoring should only be done in extreme cases: usually if one is suspicious of an individual.**
- **Protect the Logs! - if an attacker gets access to your logs, they can Scrub the evidence of their activities from them. Certificates and Write only Media can help with this.**



Unauthorized Disclosure

- **Security not only requires one to keep outsiders from getting at your sensitive data; it also requires you to keep insiders from giving it away.**
- **Object reuse – Wipe, degauss, or incinerate your old hard drives to remove possible residual data.**
- **Emanation Security – Hackers can use electric signals emitted as a byproduct of your system to determine qualities of that system.**
- **Solution? → TEMPEST (use Faraday cages to cut down on emissions) or white noise (can't tell the real emissions from the randomness)**



Access Control Monitoring

- **Key Question: Who attempted to access what resources?**
- **IDSes (Intrusion Detection Systems) – attempt to spot malicious access and sound the alarm and come in two flavors: network based IDSes (NIDS) and host – based IDSes (HIDS).**
- **NIDS use NICs in promiscuous mode in order to see all network traffic and notice anomalies.**
- **HIDS monitor activity on the computer itself for anomalous activities.**
- **Use? → NIDS more generally, HIDS only on critical servers.**



Types of IDS

- Can be either signature (look for attacks similar to previous attacks) or anomaly (look for strange activity) based.
- Signature based IDSs have problems with zero – day exploits for obvious reasons, but cut down on false positives.
- State based IDSs are a kind of signature IDS where state transition sequences form the signature of an attack.



Anomaly IDSs

- **Anomaly IDSs monitor anomalies of a number of different flavors: statistical, protocol, traffic, and rule/heuristic.**
- **Statistical – Use a sequence of “normal” activities as a baseline for testing future activity. Can detect 0 day exploits at a cost: tons of false positives that may overwhelm analysts and hide true positives.**
- **Protocol – Check each protocol on the protocol stack for out of the ordinary happenings.**
- **Traffic – Detect changes in traffic patterns (like DoSes)**
- **Rule/Heuristic – Uses an inference engine to apply rules to facts in order provide more sophisticated, “smarter” anomaly checking that cuts down on false positives.**



IDS Architecture

- **IDSs use sensors to detect attacks. When an attack is sensed, the sensor then informs the monitor in whatever way is applicable for that level of attack.**
- **IDS sensor placement is therefore important. Placing a sensor on both the interior and exterior of a firewall is a good idea to get a sense of attack attempts and actual intrusions.**



IDS Support/Alternatives

- **Intrusion Prevention Systems (IPSs) → Idea: Try and cut off an attack before it happens.**
- **Honeypots → Idea: Sacrifice a computer in order to know what kinds of attacks are currently in favor with attackers.**
- **Network Sniffers → Idea: Examine live network traffic as it happens**



Phishing and Pharming

- **Phishing – same as always, but with JavaScript replacing malicious URLs with fake real URLs and with pop – up forms present on legitimate web sites.**
- **Pharming - similar to phishing, except uses DNS poisoning to redirect legitimate web traffic to their information gathering malicious web site.**



Physical and logical assets control

Source Unknown
Reference: DHS PACS



Physical and Infrastructure Security

- **Logical security:** Protects computer-based data from software-based and communication-based threats
- **Physical security** (also called infrastructure security)
 - Protects the information systems that contain data and the people who use, operate, and maintain the systems
 - Must prevent any type of physical access or intrusion that can compromise logical security
- **Premises security** (also known as corporate or facilities security)
 - Protects the people and property within an entire area, facility, or building(s), and is usually required by laws, regulations, and fiduciary obligations
 - Provides perimeter security, access control, smoke and fire detection, fire suppression, some environmental protection, and usually surveillance systems, alarms, and guards



DHS Physical Access Control Systems

- **A) Identification** – PACS requires an individual’s PII so it can authorize physical access to DHS facilities.
 - PACS sensors read the information on an individual’s Personal Identity Verification (PIV)² card to verify if the individual is authorized access.
- **B) Visitor Management** – Visitors and construction and service contractors who have not been issued a PIV card must be identified before being granted access.
 - This is accomplished by having the individual provide the information requested on DHS Form 11000-13 “Visitor Process Information.”
 - OCSO personnel enter the information on the form into the PACS visitor management function.
 - This information is then used to conduct a search of the National Crime Information Center (NCIC) to determine if there are any criminal records or outstanding arrest warrants for the individual.
 - The results of the NCIC check are entered into PACS. If there is no disqualifying information, such as an outstanding arrest warrant, the visitor is cleared for access.
 - Access requests by foreign visitors⁴ (non-U.S. citizens and non-Legal Permanent Residents) are processed through the DHS Foreign National Visitor Management System (FNVMS).



DHS Physical Access Control Systems

- ***C) Parking Permit Management*** – The Office of the Chief Administrative Officer (OCAO) uses PACS to issue and track parking permits for the NAC.
 - OCAO personnel access PACS to determine if an individual is eligible to receive a parking permit.
 - Once determined to be eligible, the individual must submit General Services Administration (GSA) Parking Application, Form 2941.
 - Upon issuance of the parking permit, OCAO personnel enter into PACS the name and e-mail address of the permit holder, the permit number and type, issue date, and expiration date.
- ***D) Alarm Monitoring and Intrusion Detection*** – The PACS alarm monitoring application allows OCSO personnel to monitor the Intrusion Detection System (IDS).
 - A record is created in PACS of all IDS alarm activations or other issues, such as communication and power failures for example.
 - The IDS in PACS consists of sensors, lights, and other mechanisms through which OCSO can detect the unauthorized intrusion of persons or devices.
 - The only PII collected by the PACS IDS suite is the first and last name of the individual authorized to turn the alarm system on and off and the corresponding PIN number which the individual inputs into the alarm keypad to activate or deactivate the alarm.

Human-Caused Threats

- **Less predictable, may be targeted, harder to deal with**
- **Include:**
 - **Unauthorized physical access**
 - **leading to other threats**
 - **Theft of equipment / data**
 - **Vandalism of equipment/data**
 - **Misuse of resources**



Mitigation Measures Environmental Threats

- **Inappropriate temperature and humidity**
 - Environmental control equipment, power
- **Fire and smoke**
 - Alarms, preventative measures, fire mitigation
 - Smoke detectors, no smoking
- **Water**
 - Manage lines, equipment location, cutoff sensors
- **Other threats: limit dust entry, pest control**



Mitigation Measures Human-Caused Threats

- **Physical access control**
 - IT equipment, wiring, power, comms, media
- **Have a spectrum of approaches**
 - Restrict building access, locked area, secured, power switch secured, tracking device
- **Also need intruder sensors/alarms**



Recovery from Physical Security Breaches

- **Redundancy**
 - To provide recovery from loss of data
 - Ideally off-site, updated as often as feasible
 - Can use batch encrypted remote backup
 - Extreme: remote hot-site with live data
- **Physical equipment damage recovery**
 - Depends on nature of damage and cleanup
 - May need disaster recovery specialists



Disaster Recovery: Backup facilities

- **Hot sites**
 - ready to run
 - readiness at high cost
- **Cold sites**
 - Building facilities, power, communications
 - No computing resources
- **Site sharing**
 - Sharing among firms
 - Computing incompatibility
- **Need backup tapes/resources at remote site**



KU Example Policy

5. Physical and Environmental security

5.1. Secure Areas

- 5.1.1. **Physical Security Perimeter** - Company shall use security perimeters to protect all non-public areas, commensurate with the value of the assets therein. Business critical information processing facilities located in unattended buildings shall also be alarmed to a permanently manned remote alarm monitoring station.
- 5.1.2. **Physical Entry Controls** - Secure areas shall be segregated and protected by appropriate entry controls to ensure that only authorised personnel are allowed access. Similar controls are also required where the building is shared with, or accessed by, non-Company staff and organisations not acting on behalf of Company.
- 5.1.3. **Securing Offices, Rooms and Facilities** - Secure areas shall be created in order to protect office, rooms and facilities with special security requirements.
- 5.1.4. **Working in Secure Areas** - Additional controls and guidelines for working in secure areas shall be used to enhance the security provided by the physical control protecting the secure areas.

Employees of Company should be aware that additional controls and guidelines for working in secure areas to enhance the security provided by the physical control protecting the secure areas might be in force. For further clarification they should contact their Line Manager.

- 5.1.5. **Isolated Access Points** - Isolated access points, additional to building main entrances (e.g. Delivery and Loading areas) shall be controlled and, if possible, isolated from secure areas to avoid unauthorised access.
- 5.1.6. **Sign Posting Of Computer Installations** - Business critical computer installations sited within a building must not be identified by the use of descriptive sign posts or other displays. Where such sign posts or other displays are used they must be worded in such a way so as not to highlight the business critical nature of the activity taking place within the building.



Physical/Logical Security Integration

- Have many detection / prevention devices
- More effective if have central control
- Hence desire to integrate physical and logical security, especially access control
- Need standards in this area
 - FIPS 201-1 “*Personal Identity Verification (PIV) of Federal Employees and Contractors*”



Personal Identity Verification (PIV)

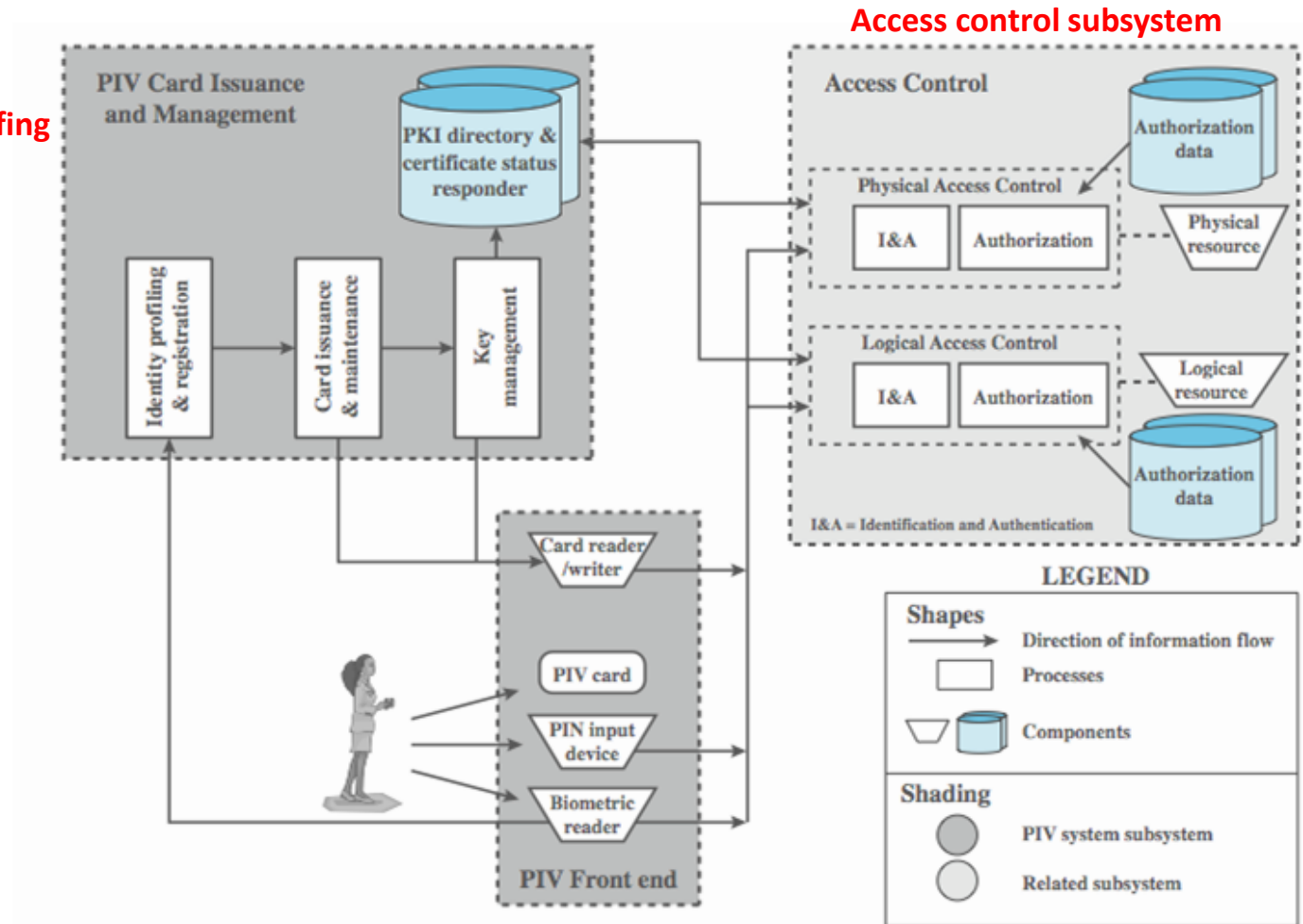
Identity proofing

Three assurance levels:

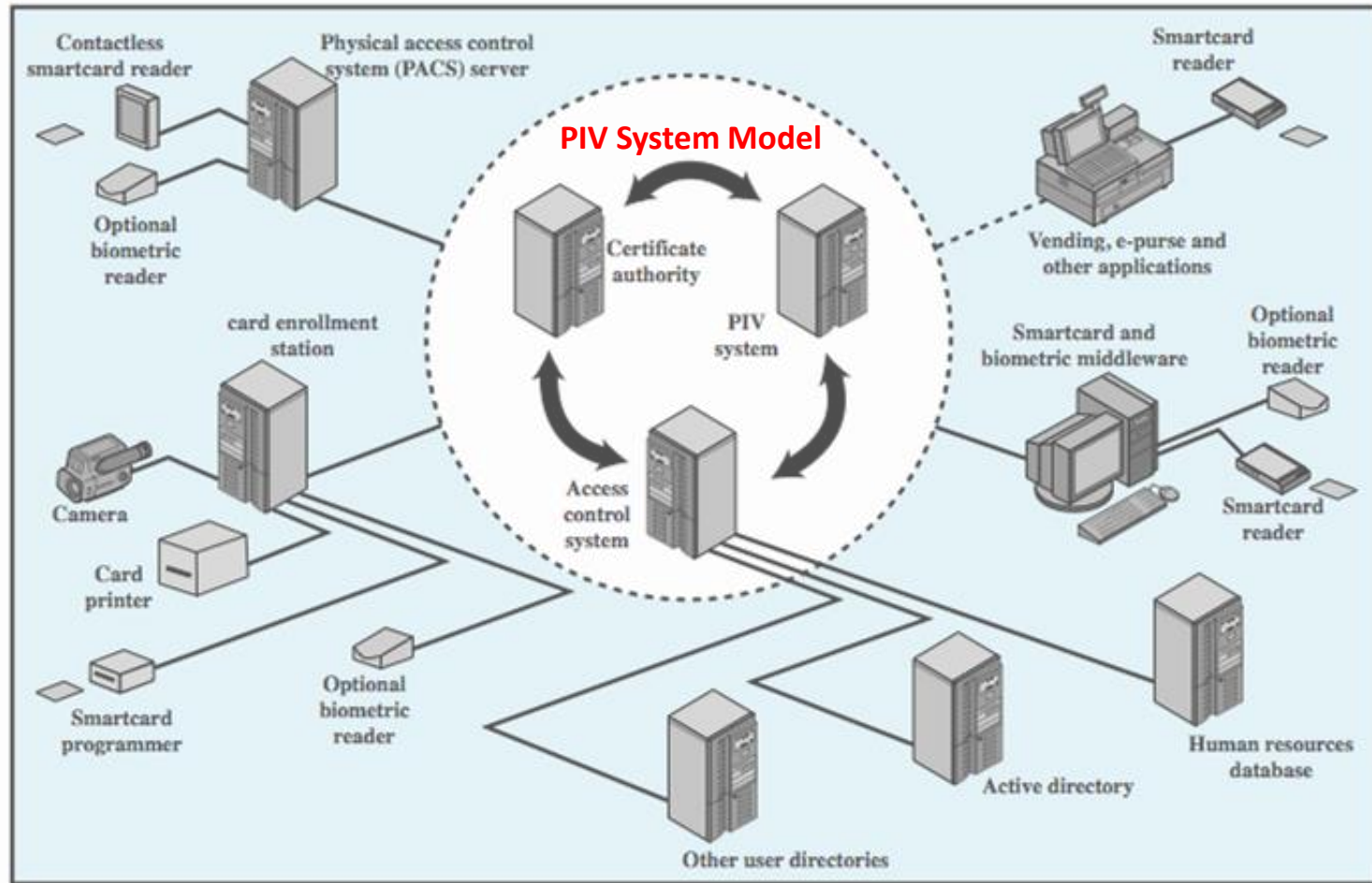
(1) Some confidence (use of smart cards/PIN)

(2) High confidence (plus use of biometrics)

(3) Very high (at the presence of an official observer)

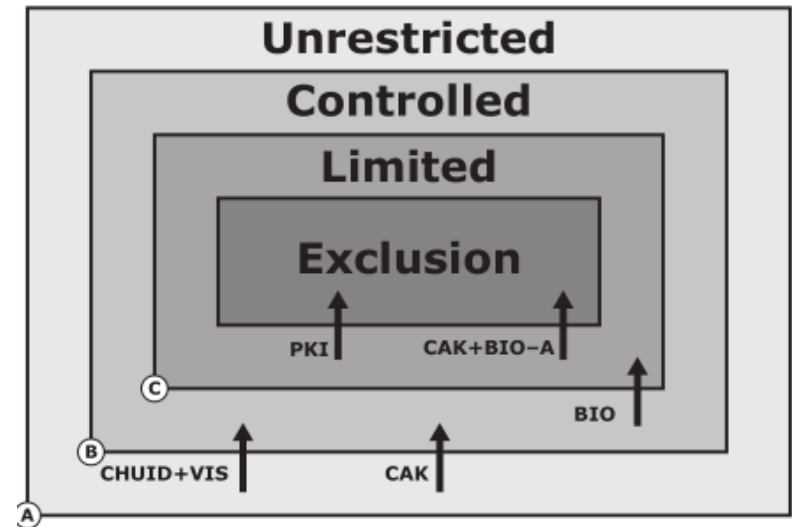


PIV (Physical/Logical) Convergence

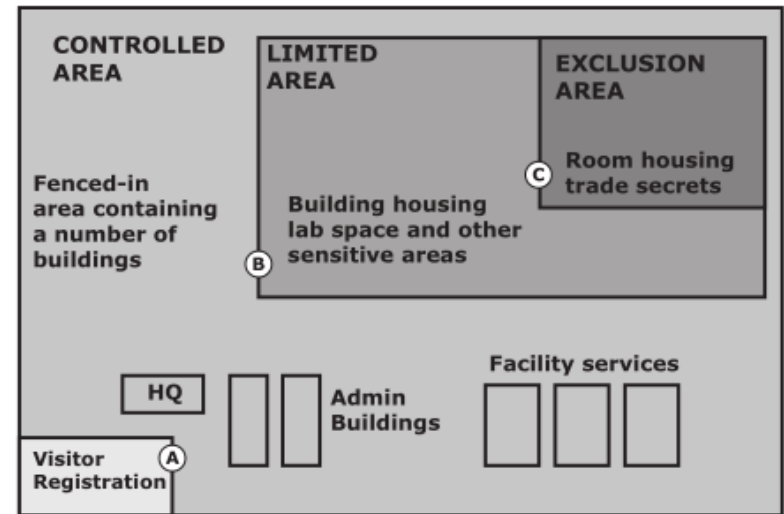


FIPS 201 SP 800-116

- **Alternative authentication mechanisms that be used for access to a specific area**
 - **CHUID:** card holder unique identification identifier
 - **CAK:** card authentication key

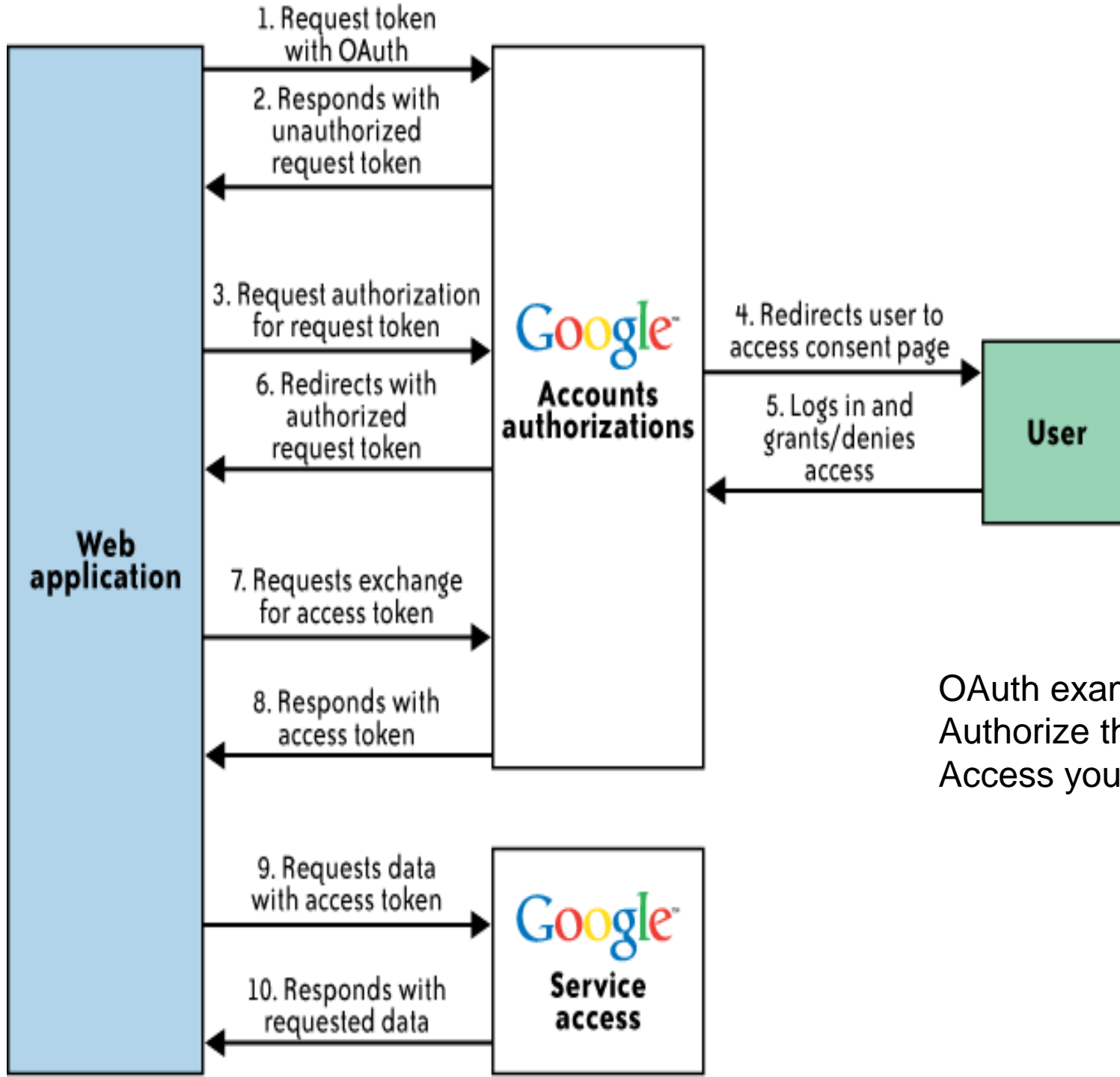


(a) Access Control Model



(b) Example Use





OAuth example:
 Authorize the third party to
 Access your data/credential

Identification and authentication of people and devices

Reference: Guide to Computer Network Security

Reference: Dustin Puryear



Authentication

- **Authentication is the process of validating the identity of someone or something.**
- **Generally authentication requires the presentation of credentials or items of value to really prove the claim of who you are.**
- **The items of value or credential are based on several unique factors that show something you know, something you have, or something you are**



Authentication

- ***Something you know:*** This may be something you mentally possess. This could be a password, a secret word known by the user and the authenticator.
- ***Something you have:*** This may be any form of issued or acquired self identification such as:
 - **SecurID**
 - **CryptoCard**
 - **Activcard**
 - **SafeWord**
 - **and many other forms of cards and tags.**
- ***Something you are:*** This being a naturally acquired physical characteristic such as voice, fingerprint, iris pattern and other biometrics.
- In addition to the top three factors, another factor, though indirect, also plays a part in authentication.
 - ***Somewhere you are:*** This usually is based on either physical or logical location of the user. The use, for example, may be on a terminal that can be used to access certain resources.



Authentication

- In general authentication takes one of the following three forms:
 - **Basic authentication**, involving a server. The server maintains a user file of either passwords and user names or some other useful piece of authenticating information. This information is always examined before authorization is granted.
 - **Challenge-response**, in which the server or any other authenticating system generates a challenge to the host requesting for authentication and expects a response.
 - **Centralized authentication**, in which a central server authenticates users on the network and in addition also authorizes and audits them.



Multiple Factors and Effectiveness of Authentication

- To increase authentication effectiveness, a scheme with multiple methods is used. Systems using a scheme with two or more methods can result in greater system security
- The popular technique, referred to as *multi-factor* authentication, overcome the limitations of a specific authentication.



Authentication Elements

- An authentication process as is based on the following five elements:
 - **Person or Group Seeking Authentication** - usually users who seek access to a system either individually or as a group. If individually, they must be prepared to present to the authenticator evidence to support the claim that they are actually authorized to use the requested system resource.
 - **Distinguishing Characteristics for Authentication** - User characteristics are grouped into four factors that include: something you know, something you have, something you are, and a weaker one somewhere you are. In each of these factors, there are items that a user can present to the authenticator for authorization to use the system.



Authentication Elements (cont)

- **The Authenticator** - to positively and sometimes automatically identify the user and indicate whether that user is authorized to access the requested system resource.
- **The Authentication Mechanism** - consists of three parts that work together to verify the presence of the authenticating characteristics provided by the user.
 - the input,
 - the transportation system,
 - and the verifier.
- **Access Control Mechanism** - User identifying and authenticating information is passed to access control from the transport component. That information is validated against the information in its database residing on a dedicated authentication server, if the system operates in a network, or stored in a file on a local medium.



Types of Authentication

- There are two basic types of authentication. non-repudiable and repudiable. Other types of authentication include user, client, and session authentication.
 - **Non-repudiable Authentication** – involves characteristics whose proof of origin cannot be denied. Such characteristics include biometrics like iris patterns, retinal images, and hand geometry and they positively verify the identity of the individual.
 - **Repudiable Authentication** – involves factors, “what you know” and “what you have,” that can present problems to the authenticator because the information presented can be unreliable because such factors suffer from several well-known problems including the fact that possessions can be lost, forged, or easily duplicated.



Authentication Methods

- There are several authentication methods including: password, public-key, anonymous, remote and certificate-based authentication.
 - **Password authentication** - the oldest and the easiest to implement. It includes reusable passwords, one-time passwords, challenge response passwords, and combined approach passwords.
 - **Public Key Authentication** – This requires each user of the scheme to first generate a pair of keys and store each in a file. Each key is usually between 1024 and 2048 bits in length. Public-private keys pairs are typically created using a key generation utility. The server knows the user's public key because it is published widely. However, only the user has the private key.



Authentication Methods

- **Anonymous Authentication - Clients who do not intend to modify entries or access protected attributes or entries on a system typically use anonymous authentication.**
 - **Mostly these users are not indigenous users in a sense that they do not have membership to the system they want access to. They access the system via a special “anonymous” account.**
- **Digital Signatures-Based Authentication – is an authentication technique that does not require passwords and user names. It consists of an electronic signature that uses public key infrastructure (PKI) to verify the identity of the sender of a message or of the signer of a document.**
 - **The scheme may include a number of algorithms and functions including the Digital Signature Algorithm (DSA), Elliptic Curve Digital Signature and Algorithm (ECDSA), account authority digital signature, authentication function, and signing function.**



Authentication Methods

- **Wireless Authentication**

- This is an IEEE's 802.1X, Extensible Authentication Protocol (EAP) scheme that authenticates mobile devices as they connect to fixed network as well as mobile networks.
- This authentication requires Wi-Fi mobile units to authenticate with network operating systems.



Developing an Authentication Policy

- In many organizations the type of authentication used is not part of the security policy, therefore, few have a say in what authentication policy is used.
 - It is becoming increasingly popular to involve as wide a spectrum of users in the development of the authentication policy.
 - Sometimes it even requires input from business and IT representative communities that do business with the organization.
- This is sometimes key to ensuring acceptance and compliance by those communities.
- Several steps are necessary for a good authentication policy:



Developing an Authentication Policy

- List and categorize the resources that need to be accessed, whether these resources are data or systems.
 - Categorize them by their business sensitivity and criticality.
- Define the requirements for access to each of the above categories taking into account both the value of the resource in the category as well as the method of access.
- Set requirements for passwords and IDs.
- Create and implement processes for the management of authentication systems.
- Communicate policies and procedures to all concerned in the organizations and outside it.

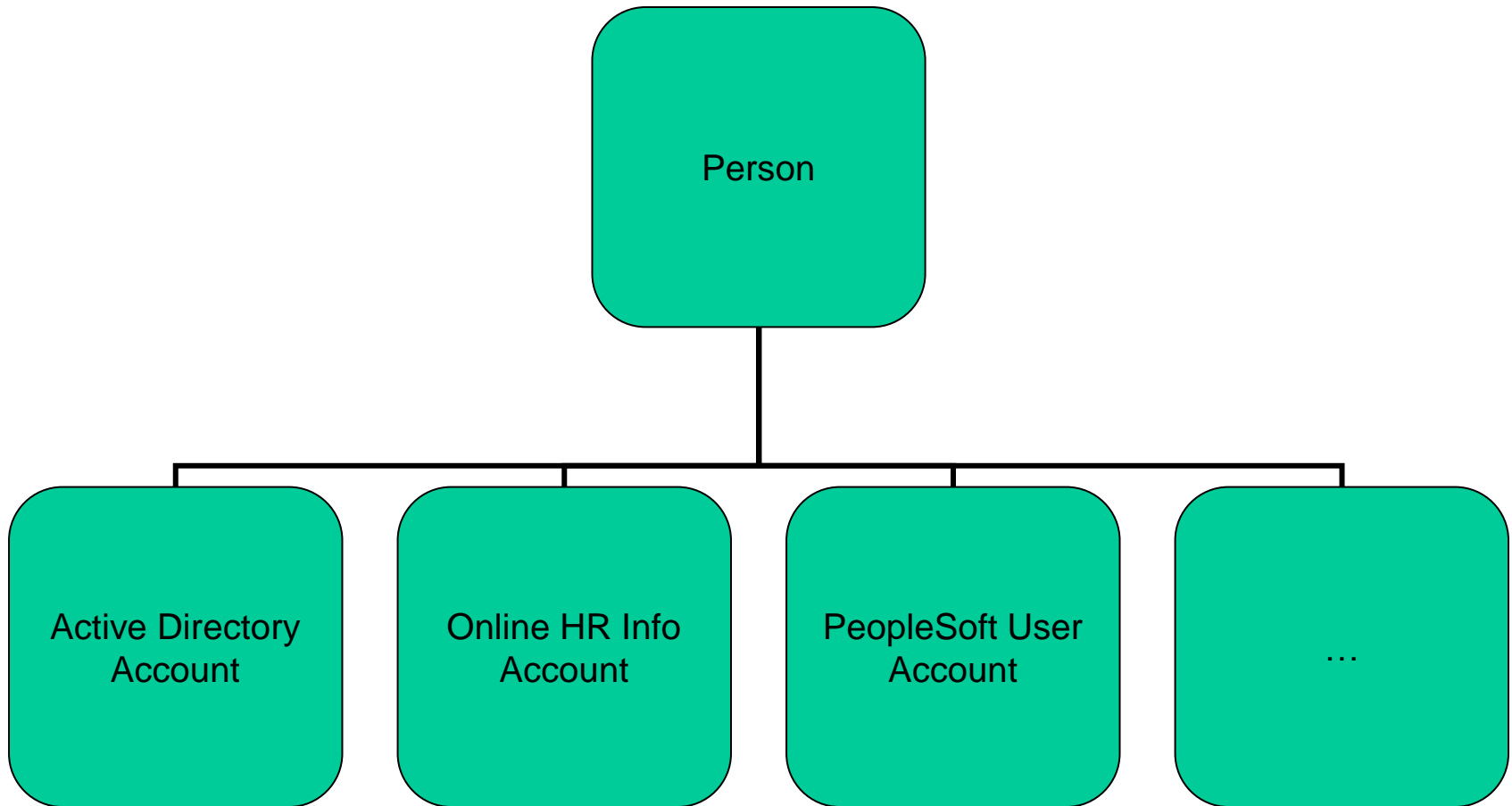


Identity and Access Management (IAM)

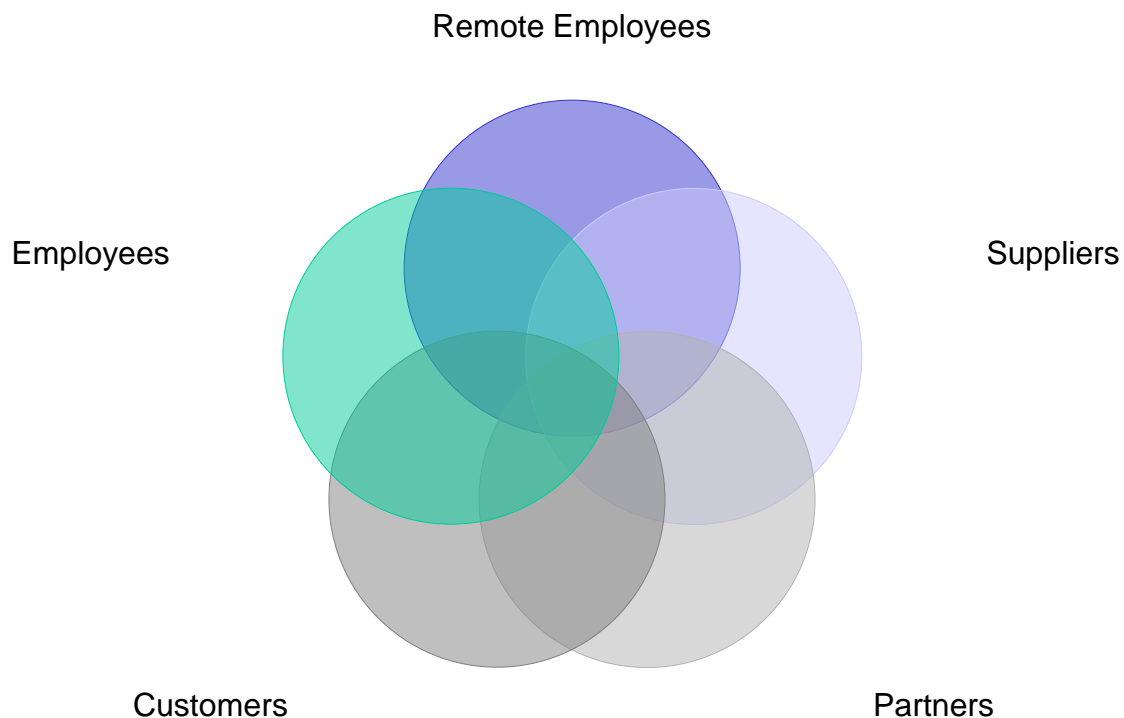
- **Networks use multiple identity systems**
- **The Internet is no better**
- **Users get confused with all of these IDs**
- **Management and audit has difficulty keeping track of all these IDs**
- **The bad guys are quite happy**



Many Accounts and IDs



Multiple Identity Contexts



Trends

- ***Regulation and Compliance***
 - SOX, HIPAA, GLB
- ***Increasing Threats***
 - Identity theft
 - Exposure of confidential info
- ***Maintenance Costs***
 - The average employee needs access to 16 applications
 - Companies spend an estimated \$20-30 user/year for password resets

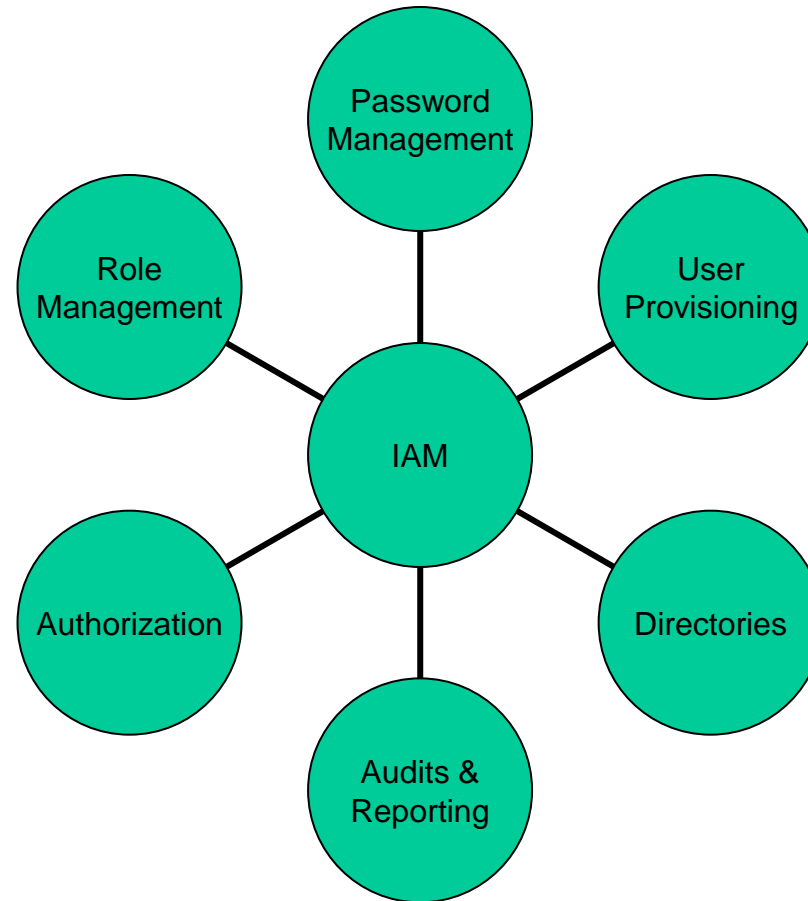


Multiple Identities

End-users	<ul style="list-style-type: none">◆ Too many IDs◆ Too many passwords◆ Must wait for access to applications
Administrators	<ul style="list-style-type: none">◆ Too many IDs◆ Too many end-user requests◆ Difficult or unreliable ways to syncs all the accounts
Audit/Compliance	<ul style="list-style-type: none">◆ Orphaned accounts◆ Limited or no audit capability◆ Where are the audit trails?



Identity and Access Management



IAM Terms

- **Authentication (AuthN)**
 - Verify that a person is who they claim to be
 - This is where multi-factor authentication comes into play
 - Identification and authentication are related but not the same
- **Authorization (AuthZ)**
 - Deciding what resources can be accessed/used by a user
- **Accounting**
 - Charges you for what you do



IAM Organization

Identity Management	<ul style="list-style-type: none">◆ Account Provisioning & Deprovisioning◆ Synchronisation
Administration	<ul style="list-style-type: none">◆ User Management◆ Password Management◆ Workflow◆ Delegation◆ Audit and Reporting
Access Management	<ul style="list-style-type: none">◆ AuthN◆ AuthZ



ISC2 Identification Methods

- **ID Badges**
- **UserID**
- **Account number/PIN**
- **MAC Address**
- **IP Address**
- **RFID**
- **Email Address**



Identity Management Implementation

- **Password Management**
- **Account Management**
- **Profile Management**
- **Directory Management**
- **Single sign-on**



Identity as a service (e.g. cloud identity)

**Lauren Clarke, Manoj Muniswamaiah,
Shawn Cicoria, John Sherlock
Pace University**



Research Problem / Question...

Is there an approach to Federated Identity Management (FIM) or IDaaS (Identity as a Service) that both providers of capabilities and consumers trust enough?

i.e. will a bank (or a Government agency) ever use an external identity provider to allow consumers to interact, transact, or bank online?

- **Acceptance test - "I can use my Facebook ID to logon to my bank"**
- **Sub Problems**
 - **How does this impact the "Internet of things/everything?"**
 - **Trust and concerns**
 - **Commercial**
 - **Consumer**
 - **Government**



Challenges with Federated Identity Adoption Among Users and Organizations

IDaaS (aka ‘Federated Identity’) has been developed using proven solutions based on technologically sound methods and components

- A number of studies have demonstrated that technology is sound
- Efforts toward far-reaching adaption have been well funded by governments
- Government and industry have expressed a desire and a willingness to integrate IDaaS solutions into their offerings

The problem, and therefore, the research opportunity:

- Consumers are slow to adapt IDaaS services into their online/cloud usage
- There appears to be a low level of trust in the IdP’s (Identity Providers)
- Lack of truly ubiquitous solutions appears to be contributing to slow adaption
- What is causing this slow adaption of IDaaS globally?



Defining Identity Management (the Terms)

- **IAM (Identity and Access Management), i.e. Identity Provisioning**
 - Identification
 - Authentication
- **FIM: Federated Identity Management**
- **IDaaS (*)**
 - Web-centric SaaS (for use within cloud-based applications), typically for B2B apps
 - Cloud-delivered legacy IDM services, typically for B2C apps
- **eID and eIDM: the European IDaaS Initiative**
- **IdP: Identity Provider (aka Asserting/Assuring Party)**
- **Open Standards**
 - SAML
 - OAuth

(*) “Understanding IDaaS: Benefits and Risks”

<http://searchcloudsecurity.techtarget.com/feature/Understanding-IDaaS-The-benefits-and-risks-of-Identity-as-a-Service>



FIM/IDaaS: The Technology

- **The Protocols**

- SAML (Security Assertion Markup Language)
<http://en.wikipedia.org/wiki/SAML>
- WS-Federation ('Web Services' -
<http://en.wikipedia.org/wiki/WS-Federation>)
- OAuth 1.0 & 2.0 - Open Auth
<http://en.wikipedia.org/wiki/OAuth>
- Kerberos - MIT (used in Enterprise applications - rarely done cross-commercial organization)

- **Products Available**

- <http://shibboleth.net/>
- AD FS (Microsoft) & Windows Azure AD (Cloud based)
- Ping Federate
- Oracle (Sun)
- IBM
- CA (Siteminder)



Federated Logon Example

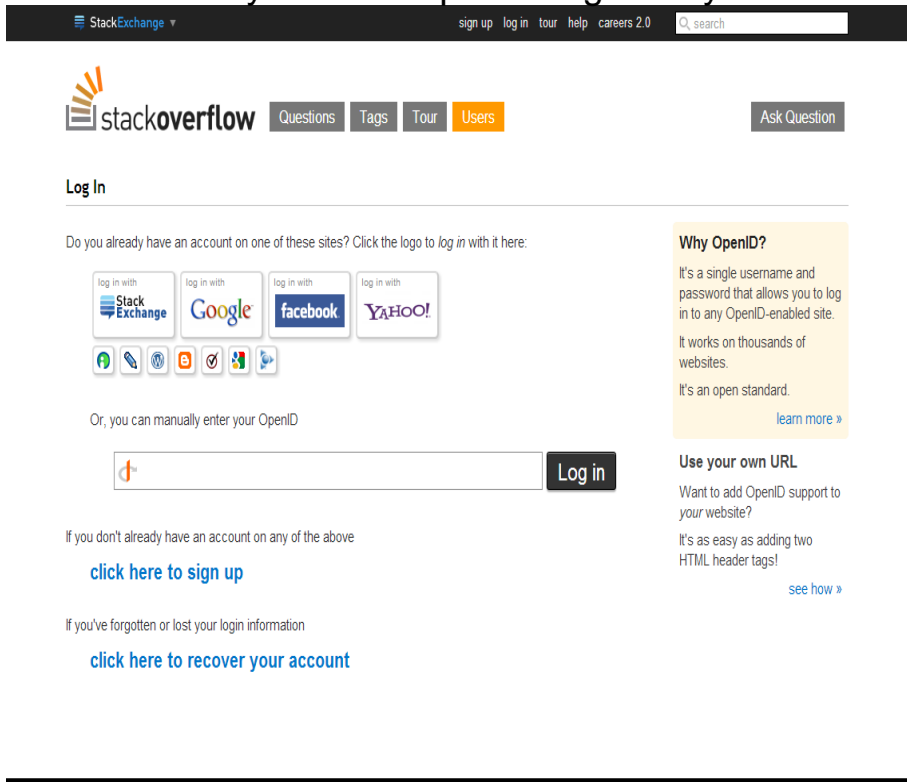
Bearer Token Based

- The OAuth 2.0 Authorization Framework: Bearer Token Usage

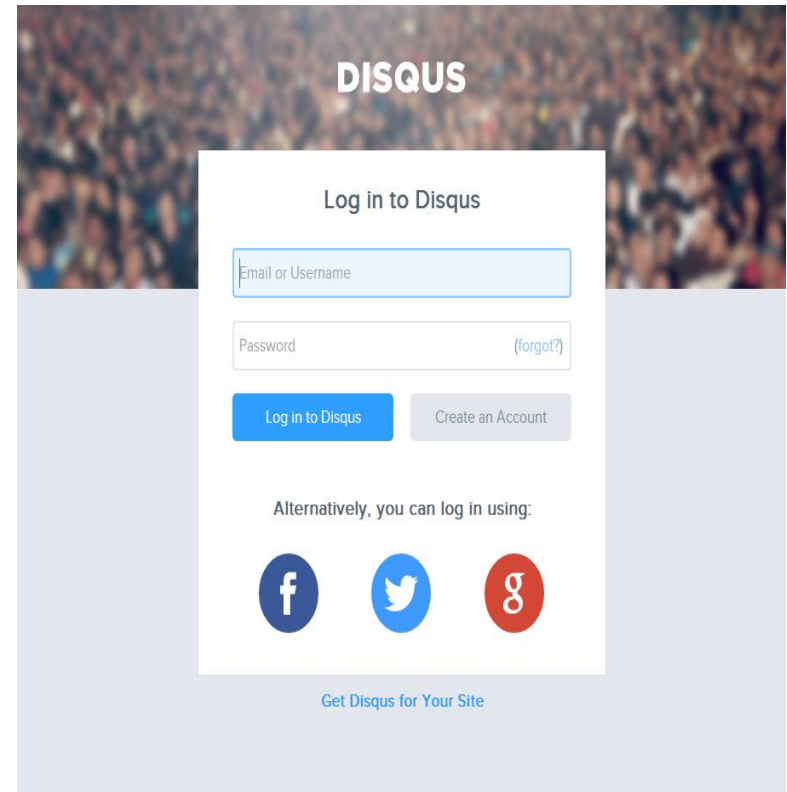
<https://tools.ietf.org/html/rfc6750>

- Bearer Token :

<http://pic.dhe.ibm.com/infocenter/wdpxc/v2r0/index.jsp?topic=%2Fcom.ibm.websphere.help.glossary.doc%2Ftopics%2Fglossary.html>



The screenshot shows the Stack Overflow website's login page. At the top, there's a navigation bar with 'StackExchange' and links for 'sign up', 'log in', 'tour', 'help', and 'careers 2.0'. Below the navigation bar, the 'stackoverflow' logo is followed by tabs for 'Questions', 'Tags', 'Tour', and 'Users', and an 'Ask Question' button. The 'Log In' section is prominent, with a heading 'Log In' and a sub-heading 'Do you already have an account on one of these sites? Click the logo to log in with it here:'. There are four 'log in with' buttons for Stack Exchange, Google, Facebook, and Yahoo!. Below these are social media icons for GitHub, LinkedIn, and others. A text input field for 'Or, you can manually enter your OpenID' is followed by a 'Log in' button. At the bottom, there are links for 'click here to sign up' and 'click here to recover your account'. On the right side, there's a 'Why OpenID?' section explaining that it's a single username and password that works on thousands of websites. Below that is a 'Use your own URL' section with a link to 'see how'.



The screenshot shows the Disqus login page. The background is a blurred image of a crowd. The 'DISQUS' logo is at the top. Below it, the heading 'Log in to Disqus' is centered. There are two input fields: 'Email or Username' and 'Password (forgot?)'. Below the input fields are two buttons: 'Log in to Disqus' and 'Create an Account'. Underneath, there's a section 'Alternatively, you can log in using:' with three social media icons: Facebook, Twitter, and Google+. At the bottom, there's a link 'Get Disqus for Your Site'.



OAuth (Open Authorization) example

The screenshot shows a Facebook 'Request for Permission' dialog for 'The New York Times' app. The dialog lists the following permissions:

- Access my basic information**: Includes name, profile picture, gender, networks, user ID, list of friends, and any other information I've shared with everyone.
- Send me email**: The New York Times may email me directly at tmoores@sees.harvard.edu - Change
- Access my data any time**: The New York Times may access my data when I'm not using the application.
- Access my profile information**: Likes, Music, TV, Movies, Books, Quotes, About Me, Interests, Groups, Birthday, Hometown, Current City, Website, Education History and Work History.

At the bottom of the dialog, it states: 'Use of this data is subject to the [The New York Times Privacy Policy](#) - Report App'.

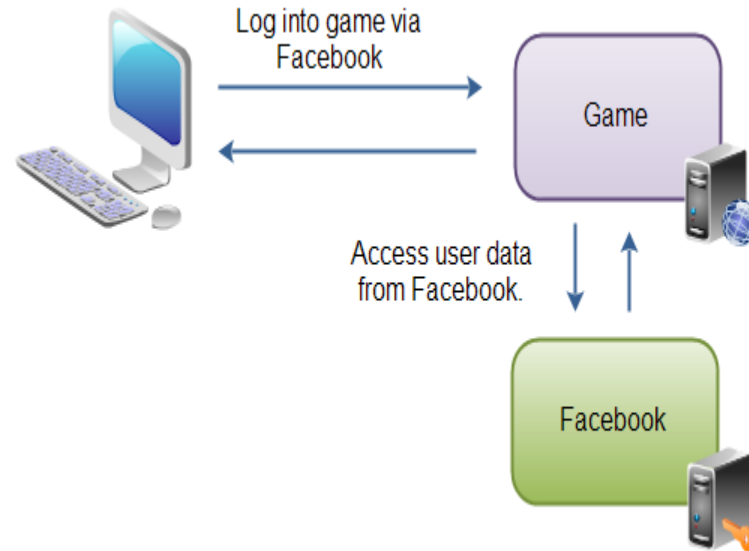
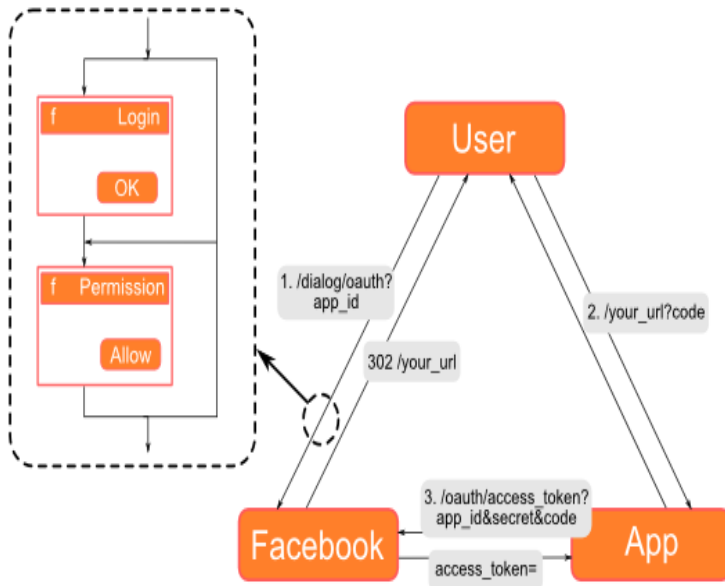
On the left side of the image, a Google accounts sign-in page is visible, showing the text: 'Stackoverflow.com is asking for some information from your Google Account twmoore@gmail.com' and 'Email address: twmoore@gmail.com'. There are 'Allow' and 'No thanks' buttons, and a checked box for 'Remember this approval'.



OAuth Flow

Facebook OAuth Authentication

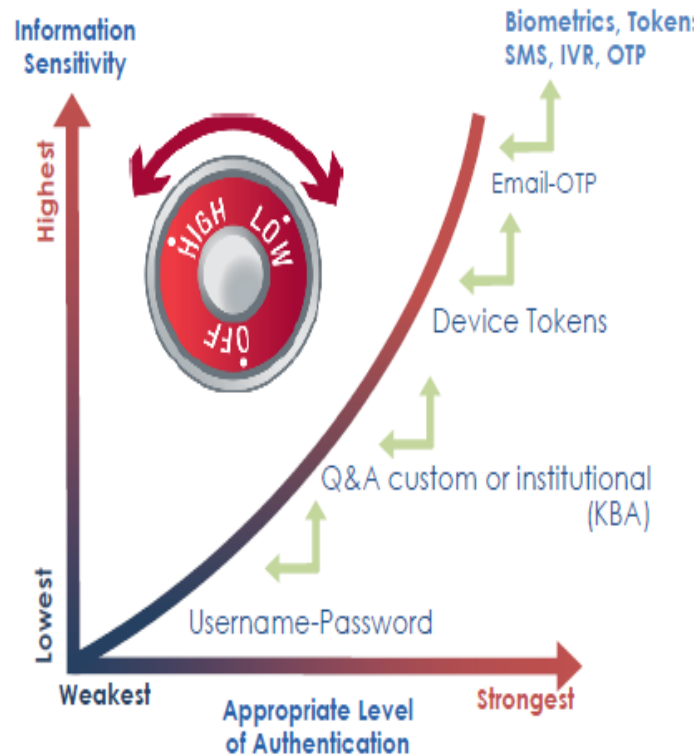
<http://tutorials.jenkov.com/oauth2/index.html>



http://tungwaiyip.info/blog/2011/02/19/facebook_oauth_authentication_flow



Progressive Authentication



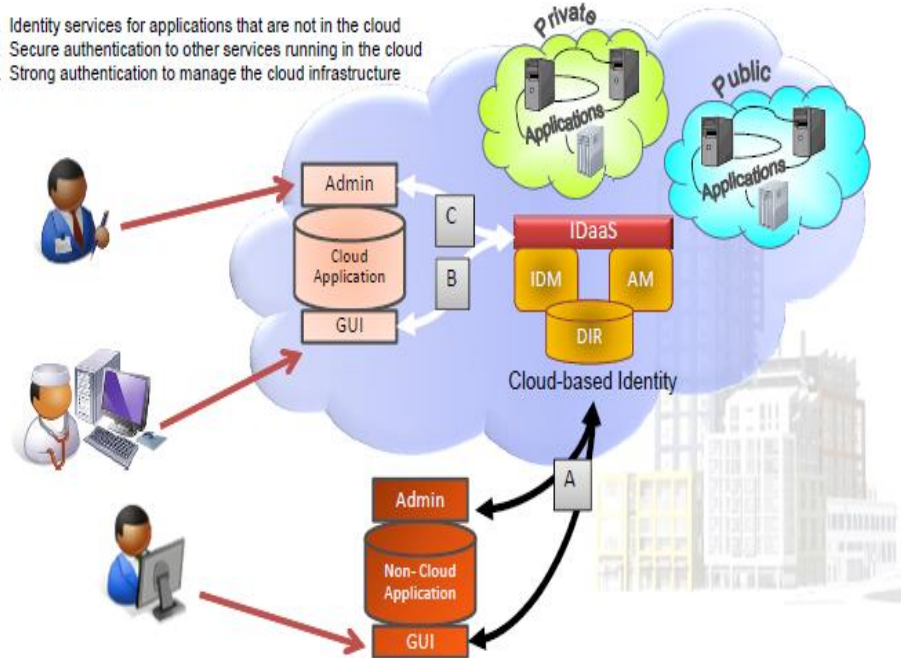
http://www.equifax.com/technology_analytics/anakam/documentation/anakam_progressive_authentication_brochure.pdf



Identity in the Cloud

Taking Identity to the Cloud

- A. Identity services for applications that are not in the cloud
- B. Secure authentication to other services running in the cloud
- C. Strong authentication to manage the cloud infrastructure



<http://www.himssanalytics.org/uploads/product/whitepaper/63D0712307954ED38B75FE1F5FB84BB5.pdf>

Application ID Request (Service Provider vs. Identity Provider) (*)

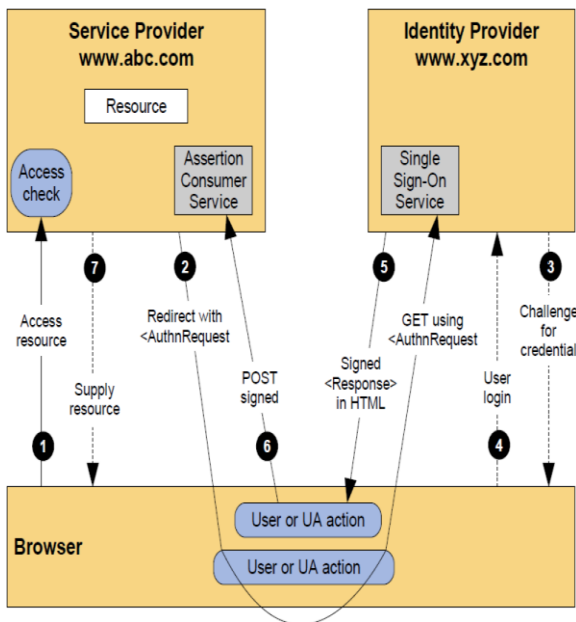


Figure 3.2-1. Service Provider-Initiated Access to an Application in a Federation

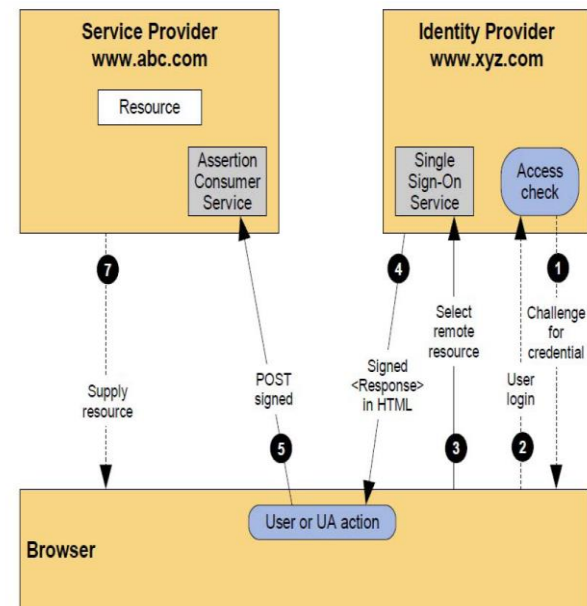


Figure 3.2-2. Identity Provider-Initiated Access to an Application in a Federation

(*) "CSC White Paper: "Identity Federation Concepts"

http://assets1.csc.com/cybersecurity/downloads/FIM_White_Paper_Identity_Federation_Concepts.pdf



Single Sign On

(Traditional vs. Federated) (*)

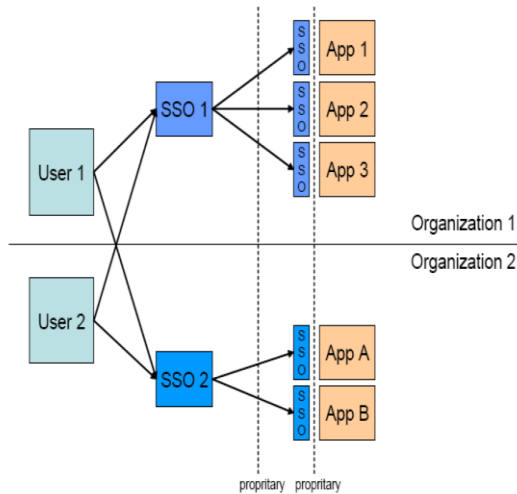


Figure 3.13-1. Traditional SSO: Using Multiple SSOs to Manage Application Access

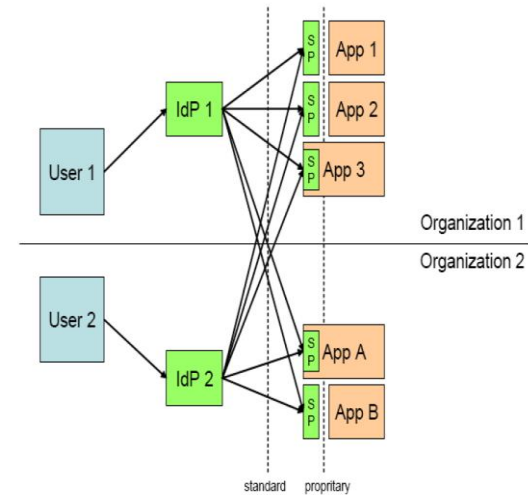


Figure 3.14-1. Typical Federation SSO

(*) “CSC White Paper: “Identity Federation Concepts”

http://assets1.csc.com/cybersecurity/downloads/FIM_White_Paper_Identity_Federation_Concepts.pdf



**SAML 2.0 and WS-Federation:
Many Commonalities**

- Same base principles — an identity provider and a service provider
- Both support all the features discussed
- Both address the issue of passive and active clients
- Both can tunnel their communication through the browser when required
- Both have SOAP as a communication channel for active clients

SAML 2.0 vs. WS-Federation: Key Differences		
Attribute query and assertion	SAML precisely defines messages for exchanging queries and assertions.	WS-Federation does not define any specific messages but suggests using a Web service for this purpose..
How services are defined	The SAML standard defines a transport-neutral protocol that can be used with various bindings, SOAP being one of them.	The WS-Federation specification defines its services using SOAP and then builds facilities for tunneling SOAP through the browser.
Assertions and security tokens	The SAML standard defines identity information in the form of assertions. Large parts of the standard go into defining the assertions and attribute profiles.	No definition of what the security token looks like; the token is opaque, with one possible security token being SAML assertions as defined in the SAML standard. Thereby WS-Federation can leverage part of SAML, which is well in line with the compositability concept normally used in WS-* specifications.
Handling of single logout	Logout information will always be sent to as many receivers as possible. No notion of session timeout.	An application has to register to receive logout information. No notion of session timeout.

(*) "CSC White Paper: "Identity Federation Concepts"
http://assets1.csc.com/cybersecurity/downloads/FIM_White_Paper_Identity_Federation_Concepts.pdf



Benefits

- **Single Sign-On (SSO)** – a user is prompted to authenticate once and allowed access to multiple systems for which he/she is authorized. This may be internally as well as across organization domains.
- **Cost Reduction**
 - **Aetna in conjunction with NaviMedix function as IdPs in a federated system to manage the billing of medical practices (*)**
 - **The elimination of redundant user administration systems**
- **Effective use of resources**
 - **InCommon and NIH collaborate as IdPs and SPs to allow access for information sharing between the medical research laboratories and medical schools(*)**

(*)S. Landau and T. Moore, “Economic tussles in federated identity management,” *First Monday*, vol. 17, no. 10, pp. 1–20, 2012



Risks

- **Trust (*)**
 - **trustworthiness of the user**
 - **service provider and identity provider's inability to prevent Man in the middle attacks which lead to theft of user's identity**
 - **trustworthiness of the IDPs and SPs**
- **Security (*)**
 - **logic flaws from browser relayed messages (browser)**
 - **insecure channel (network)**
- **Liability(**)**
 - **Who is responsible when something goes wrong?**

(*) E. Ghazizadeh, M. Zamani, A. Pashang and J. A. Manan, "A survey on security issues of federated identity in the cloud computing, 2012 IEEE, pp. 562–565, 2012

(**) S. Landau and T. Moore, "Economic tussles in federated identity management," *First Monday*, vol. 17, no. 10, pp. 1–20, 2012



FIM/IDaaS: Current Status

- **The Actors**
 - The Government
 - Private Industry
- **Who are the leaders, charting the way?**
 - Microsoft
 - Ping Federate
 - Oracle (Sun)
 - IBM
 - CA
- **How much should the government be involved?**
 - In the design
 - In the implementation
 - In the control and monitoring



Europe's Efforts in the eIDM Arena

- The EU has had a number of eIDM initiatives since 1995
- Cross-border requirements are incremental challenges for the EU
- To date, the EU eIDM framework has been based on the interoperability of eSignatures
- Lack of EU-wide legal framework resulted in difficulty in defining actor responsibilities and liabilities. (*)
 - This has results in a degree of “subsidiarity” amongst the member nations: each member nation maintaining its autonomy and responsibility. (**)
- There was a reasonable level of trust in the overall solution in the UK, with ease of use scoring the highest as a key user requirement (***) ... but
 - 50% left the site once reaching the hub page (the sign-up page)
 - 25% left the site once reaching the consent page
 - 34% of test users felt threatened rather than reassured by the privacy statement

(*) “eID for Europe: Moving from Problems to Solutions”
<http://www.jiclt.com/index.php/jiclt/article/viewFile/179/177>

(**) “Federated Identity to Access eGovernment Services – Are Citizens Ready for This? DIM’13 11/8/13

(***) “Is Europe Ready for a Pan-European IDM?” IEEE July/August 2012



Issues Holding Back FIM/IDaaS

- What exactly constitutes one's identity?
- Who creates a person's eID and managed it?
- Who will manage/regulate the links between the identities?
- How is the integrity/reliability/privacy of the identity ensured? Who will stand behind this guarantee?

Globally, there is a consensus that new research would be needed to coordinate existing knowledge and know-how (which is already available to a significant extent) into a coherent vision. (*)

(*) "eID Infrastructure for Trustworthy Services in eGovernment and eCommerce" 2012
<http://world-comp.org/p2012/SAM9717.pdf>



Research Extensions

- **There already exist many conceptual models which ‘prove’ that:**
 - **There is a need for a unified solution to these IDM problems**
 - **Available technological solutions would work in today’s eCommerce environments**
- **Questions which still need to be answered:**
 - **Who should ‘own’ (i.e. regulate) the solution?**
 - **What would it take to raise consumers’ confidence in any solution (win them over)?**
 - **What is the model for user recourse if the solution breaks (who is liable)?**



Third-party identity services (e.g. on-premise)

Reference: Jens Haeusser, UBC



Identity Management

- **Today: Centralized Identity Management**
 - **Overview, Best Practices, and Lessons Learned**
 - **“Identity 1.0”**
- **Tomorrow: Federated ID**
 - **Shibboleth and eduroam**
 - **“Identity 1.5”**
- **What’s Next: Distributed / User Centric ID**
 - **Open ID, Cardspace, and Claims**
 - **“Identity 2.0”**



What is Identity Management?

- **Lifecycle maintenance of electronic accounts**
- **Provisioning**
 - **Account creation**
 - **Account updates**
 - **Role maintenance**
 - **Account removal**
- **Authentication & Authorization**
- **Access Control**



Why is it Important?



*“Your identity is
your most valuable
possession.*

Protect it.

*And if anything
goes wrong, use
your powers!”*

- Elastigirl



Kim Cameron's Identity Weblog



Mississippi State University Center for Cyber Innovation

Domain 5 Identity and Access Management



138

Today's Challenges

- **Complex and fractured identity landscape**
 - **Many systems of records**
 - **Many applications**
 - **Many passwords**
 - **Many overlapping roles**
- **Make life easier for faculty, staff and students**
 - **Enable access to resources**
 - **Enforce privacy and security**
 - **Create a sense of a unified University**

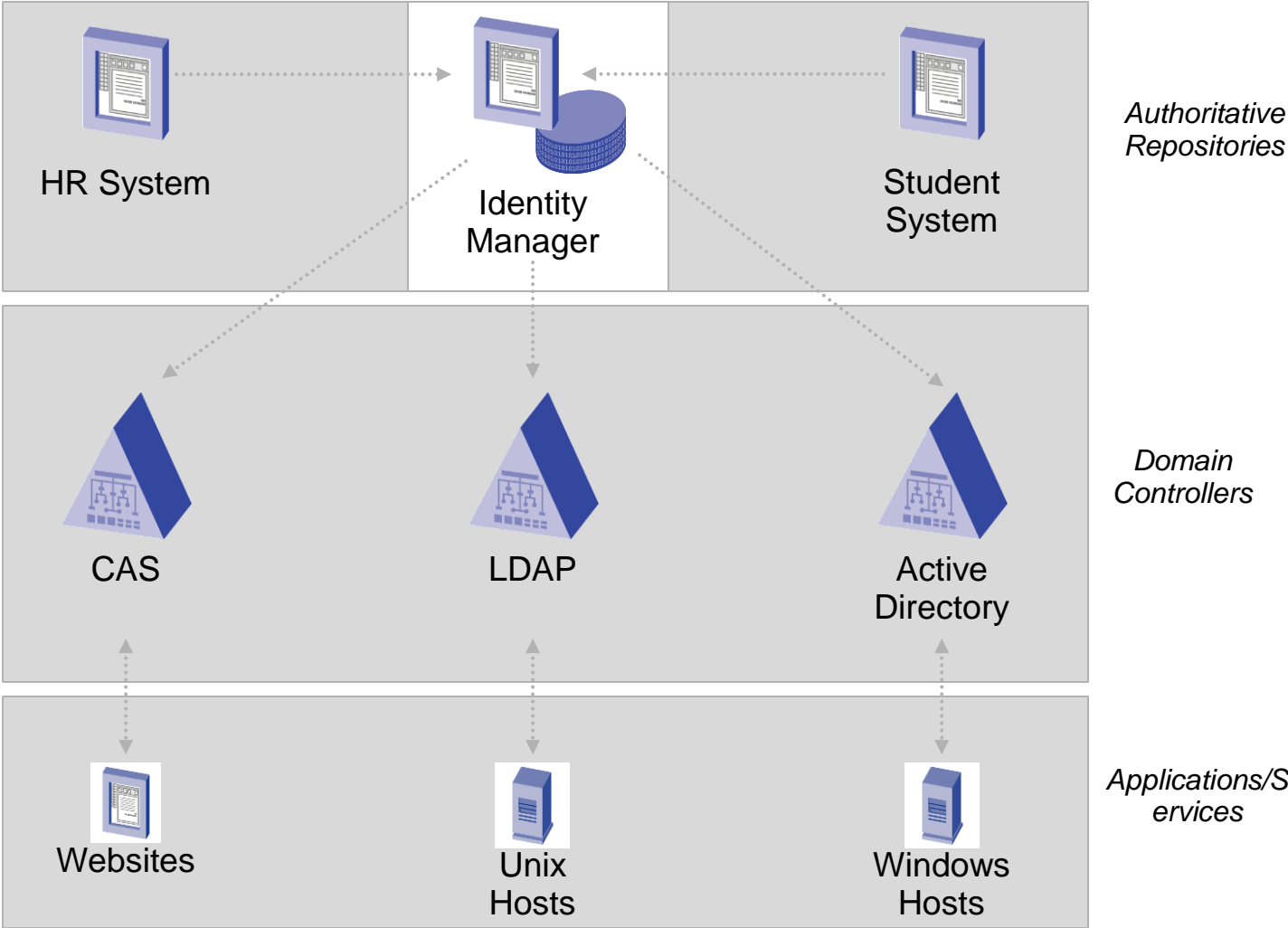


Today's Solutions

- **Consolidated directories**
- **Integrated and automated provisioning**
- **Multiple managed domain controllers**
- **Separation of Authentication and Authorization**
- **Role-based access control**
- **Virtual organizations**
- **Distributed and delegated administration**
- **Initial/reduced/single sign-on**



A Provisioning Example



Lessons Learned

- **It's all about relationships**
 - **Let people engage, cradle to grave**
 - **Multiple, overlapping, ever changing**
- **Embrace multiple authoritative sources**
 - **Authoritative for attributes, not people**
- **Account names should be ephemeral**
 - **Users should be free to select and change**
 - **Applications should record account ID, not name**
- **Dynamic rules, not static roles**



Tomorrow: Federation

- **Today's solutions are institution centric**
 - **Institution as walled garden**
 - **Centralized Identity - “Identity 1.0”**
- **Tomorrow's solutions move beyond the institution**
 - **Broadcast identity from one institution to another**
 - **Trust model controlled by institution, not user**
 - **Federated Identity - “Identity 1.5”**



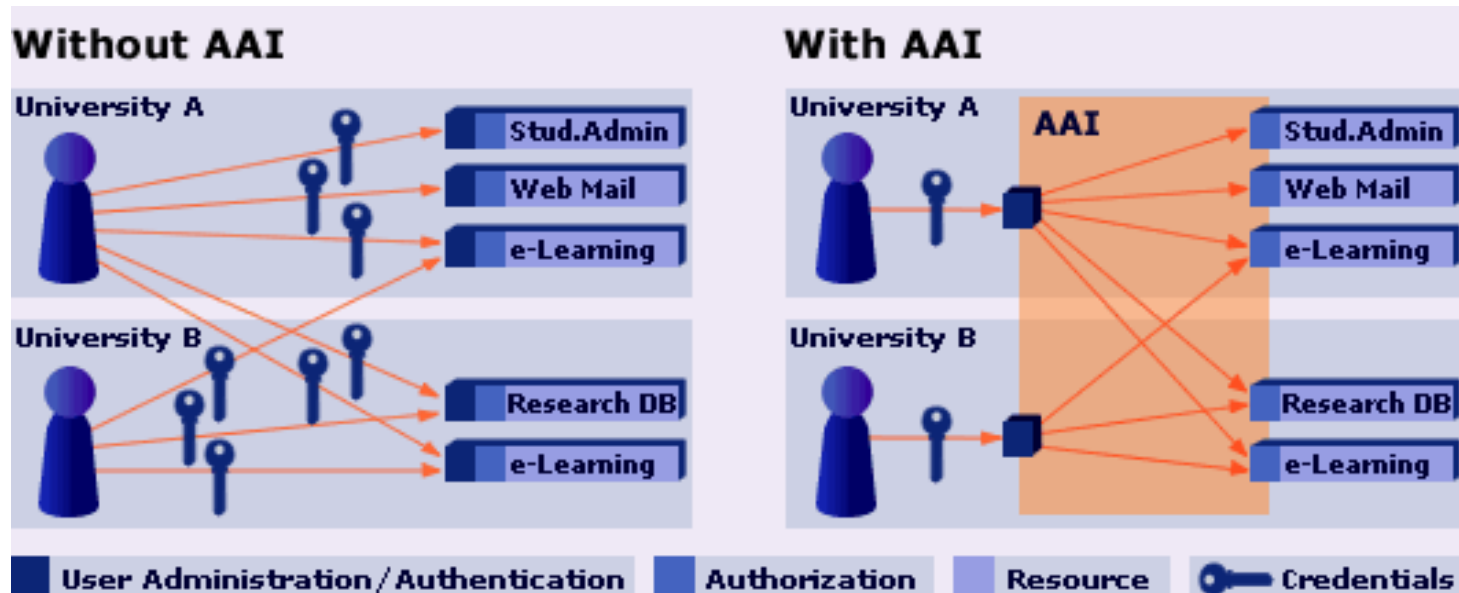
What are Federations?

- **Group of organizations sharing a set of agreed policies and rules for access to online resources**
 - **enable members to establish trust and shared understanding of language or terminology**
 - **provide a structure / legal framework that enables authentication and authorization**
- **Enables people to use their home credentials to connect to remote sites**
 - **Without revealing their credentials (pseudonymity)**
 - **Without releasing unnecessary private information**



A Federation Example

Authentication and Authorization



SWITCH

The Swiss Education & Research Network



Mississippi State University Center for Cyber Innovation

Domain 5 Identity and Access Management



145

What is?



Shibboleth®

- **An open source project supporting inter-institutional sharing of web resources subject to access controls.**
- **Streamlines sharing secured online services**
- **Leverages campus identity and access management infrastructures**
 - **sends information about users to resource site**
 - **enables resource provider to make authorization decisions**
- **Ideal for lightweight web authentication**
 - **digital libraries**
 - **learning object repositories**



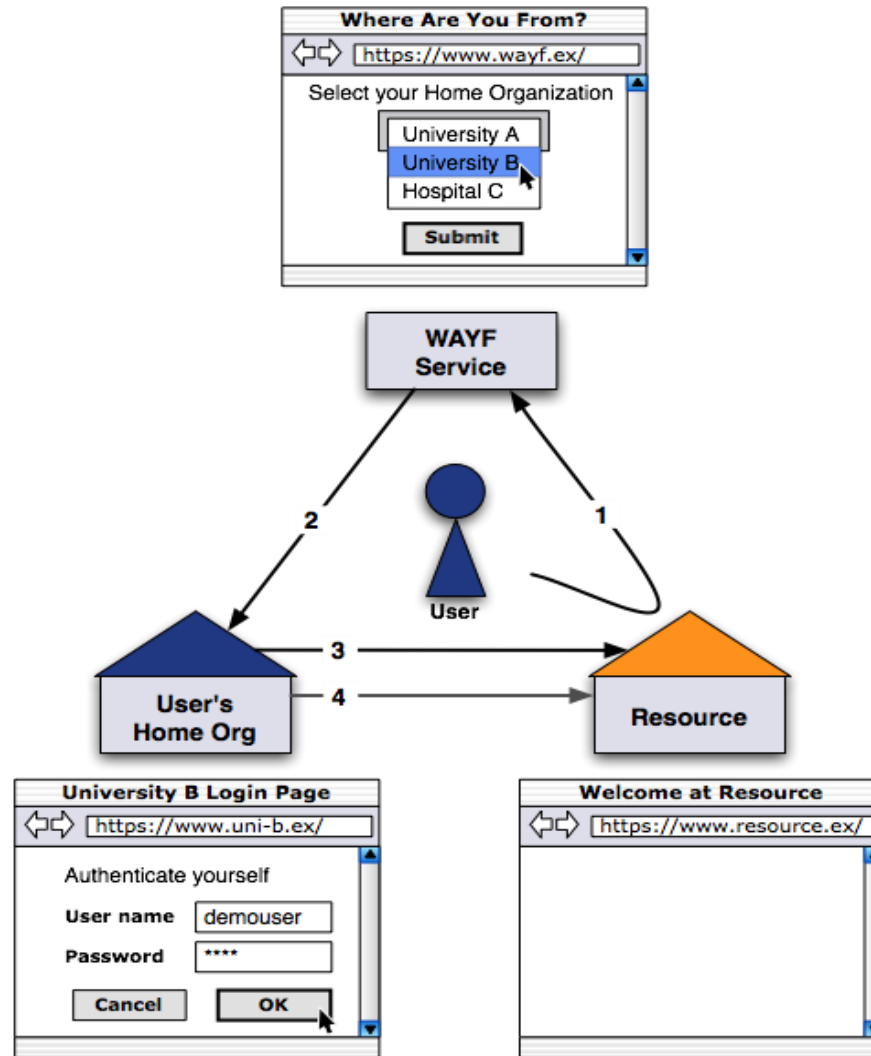
Mississippi State University Center for Cyber Innovation

Domain 5 Identity and Access Management



146

How Does it Work?



SWITCH



Where is it Used?

Information Providers:

Bodington

EBSCO Publishing

Elsevier ScienceDirect

ExLibris - SFX

JSTOR

National Digital Science Library
(NSDL)

Project MUSE

TurnItIn

Products:

Blackboard

Confluence

EZProzy

iTunesU

Moodle

Twiki

Sakai

Sympa

WebCT



What is?

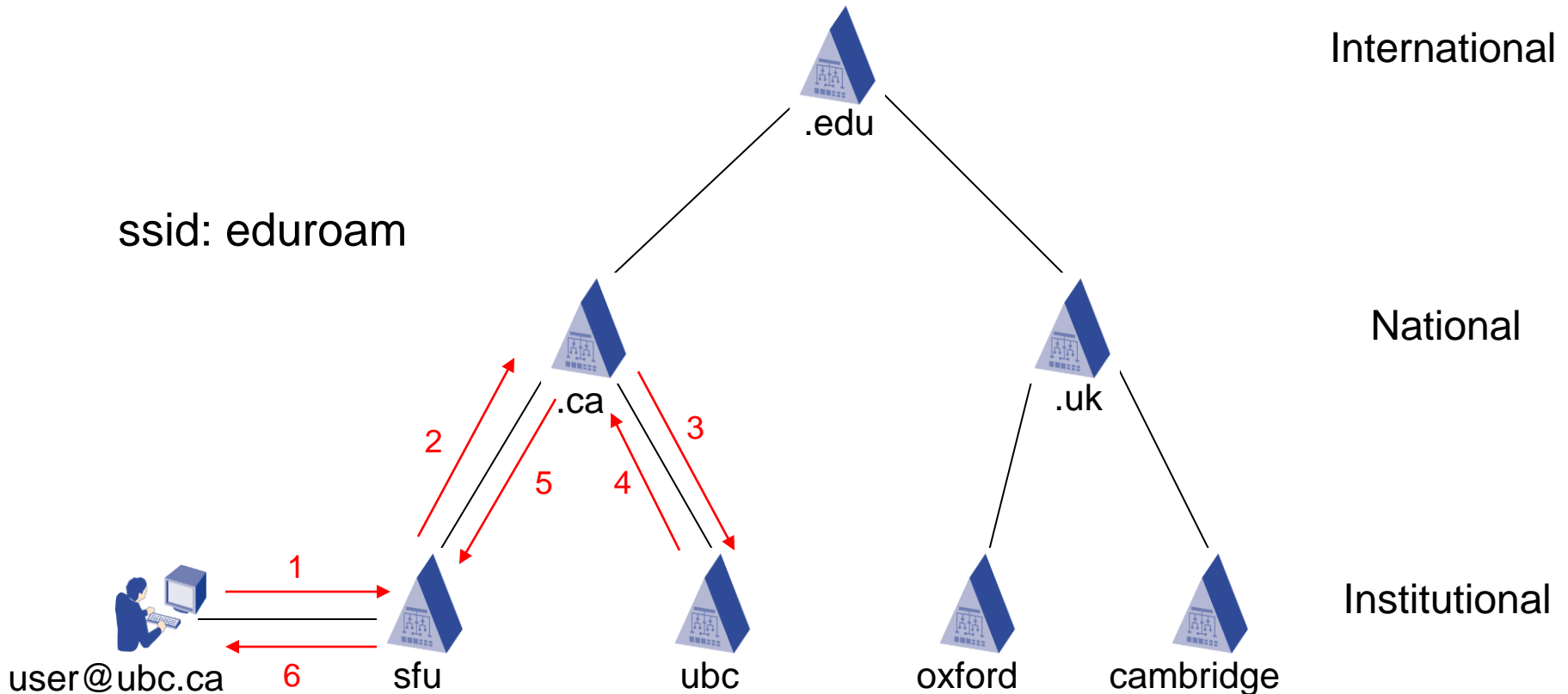


Network roaming for higher education and research

- **eduroam stands for Education Roaming**
- **Originally a European initiative**
- **Launched in 2003 to deal with the “Roaming Scholar problem”**
- **RADIUS-based infrastructure**
- **Uses 802.1X to allow inter-institutional roaming**
- **Allows users visiting other eduroam institutions to access WLAN using home credentials**



How Does it Work?



Higher Education Federations

- **Shibboleth**
 - **InCommon (US)**
 - **UK Access Management Federation**
- **eduroam**
 - **JANET (UK)**
 - **TERANA**
- **Policy Based**
 - **CIMF (Canada)**
 - **SWITCH (Switzerland)**



What Comes Next?

- Move control from the institution to the individual
- Complex interactions with many institutions
- Greater control over identity data
 - User chooses which attributes (claims) to release, and where to get those claims
- User Centric Identity - “Identity 2.0”

“Of course I have a secret identity. I mean, do you see me at the supermarket wearing... this? Who wants to go shopping as Elastigirl, know what I'm saying?”

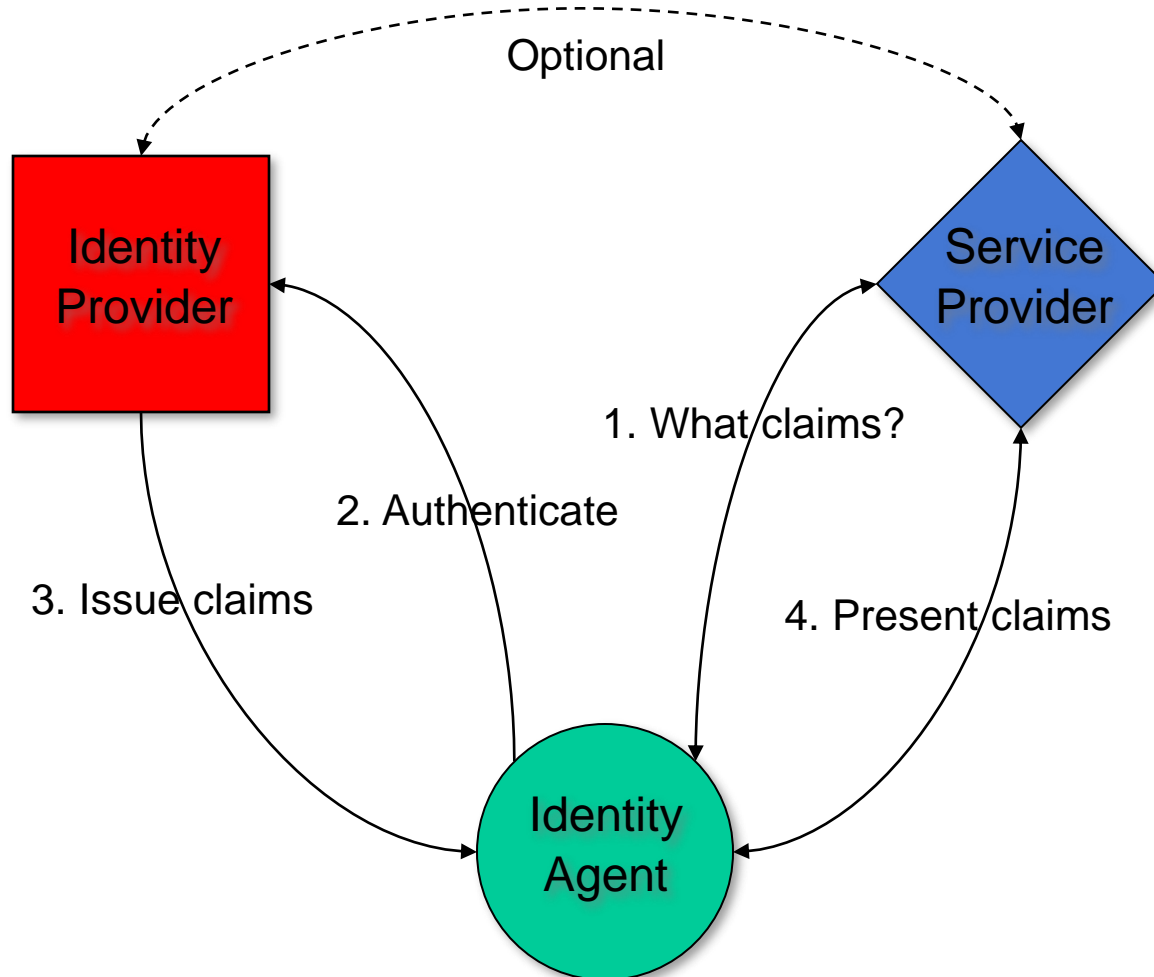


What are Claims?

- **An assertion, made by the user, of identity data**
 - **Identifier (account name)**
 - **Personal information (name, address, birthday)**
 - **Group membership (over 21, University student)**
- **Multiple types**
 - **Directly validated (password)**
 - **User-asserted (self signed)**
 - **Third party validated (trusted public key)**



How Does it Work?



What is OpenID?

- Open source, distributed authentication system
- Simple and lightweight: identity is a URL
- Fully decentralized and open platform
- I want to log into example.com:
 1. I type my OpenID URL into the login form on example.com
 2. example.com redirects me (via my web-browser) to myopenid.com
 3. I tell myopenid.com whether or not I trust example.com with my identity
 4. I am redirected back to example.com and am automatically logged in



What is CardSpace?

- **Windows client software- part of Microsoft's "Identity Metasystem"**
- **Stores "Identity Cards"**
 - **Bundles of claims**
 - **Managed or self-issued cards**
- **Presents user with choice of valid cards**
- **Token Agnostic**
 - **Can use SAML, Shibboleth, OpenID, WS-*, ...**



The Coming Convergence

- **Still early days, and rapid development, but...**
- **Active, open conversation between developers, creating the Internet Identity Layer**
- **Open Source Infocard clients and servers emerging**
- **Microsoft sponsored Shibboleth-Cardspace integration**
- **CAS 3.1 supports OpenID and SAML**



Conclusion

- Identity practice undergoing dramatic changes
- Users will expect to engage with us in new ways
 - Bring identity information when they join
 - Gradual migration to claim based access
- Prepare by continuing to strengthen and consolidate internal Identity Management
- Target low hanging fruit for Federation
- Keep abreast of user-centric identity management





"Your identity is your most valuable possession. Protect it. And if anything goes wrong, use your powers!" - Elastigirl

Kim Cameron's Identity Weblog



Comments

LEAVING A COMMENT

How to use an InfoCard to leave comments without a password

White Papers

The LAWS OF IDENTITY

The key to this site: an introduction to Digital Identity - the missing layer of the Internet.

A PRIVACY-COMPLIANT IDENTITY METASYSTEM

The Identity Metasystem: Towards a Privacy-Compliant Solution to the Challenges of Digital Identity

PRIVACY IN THE LAWS OF IDENTITY

The Case for Privacy-Embedded Laws of Identity in the Digital Age by Commissioner Ann Cavoukian

INTRODUCING INFOCARD

An excellent introduction to InfoCard technology by David Chappell

The IDENTITY METASYSTEM

A proposal for building an identity

Roland Dobbins on DDoS attacks and mitigations

Most DDoS attacks these days aren't spoofed, because there's no need, given the zillions of botted computers out there...

Posted on Monday 28 May 2007

Roland Dobbins has written to point out that the recent [Russian cyber-attacks](#) on Estonia are not the first launched by one state against another (he cites incidents during the Balkan conflict, as well as China versus Japan).

Then he gives us an overview of DDoS attacks and mitigations:

DoS attacks are easy to trace as long as Service Providers (SPs) have the proper instrumentation and telemetry enabled on their routers - NetFlow is the most common way of doing this, along with various open-source and commercial tools (nfdump/nfsen, Panoptis, Arbor, Lanclope, Narus, Q1).


Most DDoS attacks these days aren't spoofed, because a) there's no need, given the

Login to

Kim Cameron's Identity Weblog



[With an Information Card](#)

 Selector Installed

[What is an Information Card?](#) [Lost your Card?](#)

Or use a one-time [Site Pass](#)

Email:

First name:

Last name:




Code:

[« Back to blog](#)

Windows CardSpace

Do you want to send a card to this site?

Review the following site information and privacy statement to decide if you want to send a card to this site.

 • This site does not meet Windows CardSpace requirements for a bank or major Internet business. To learn more, click [Why is this important?](#)

Site information: _____

www.identityblog.com
Organization name not verified
Location not verified
[View privacy statement](#)

Cards that are sent to this site may be sent to the site's designated agents.

Site information verified by: _____
Starfield Secure Certification Authority

[➔ Yes, choose a card to send](#)

[➔ No, return to the site](#)

Tasks

- [View certificate details](#)
- [View privacy statement](#)
- [Disable Windows CardSpace](#)
- [Why is this important?](#)
- [Help](#)


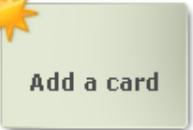
[« Back to blog](#)

Windows CardSpace

Choose a card to send to: www.identityblog.com

To see or edit card data before you send it, select a card, and then click Preview. To create a new card, click Add a card and then click Add.

Your cards:

-  Demo Card
-  Add a card

Tasks

- Duplicate card
- Delete card
- Add a card
- Back up cards
- Restore cards
- Preferences
- Delete all cards
- Disable Windows CardSpace
- Which card should I send?
- Help
- Learn more about this site


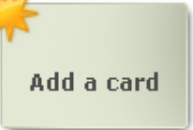
[« Back to blog](#)

Windows CardSpace

Choose a card to send to: www.identityblog.com

To see or edit card data before you send it, select a card, and then click Preview. To create a new card, click Add a card and then click Add.

Your cards:

Demo Card

You have not sent this card to the site. You can review the card before you send it. To review the card, click Send or Preview

[Send](#) [Preview](#)

Tasks

- [Duplicate card](#)
- [Delete card](#)
- [Add a card](#)
- [Back up cards](#)
- [Restore cards](#)
- [Preferences](#)
- [Delete all cards](#)
- [Disable Windows CardSpace](#)
- [Which card should I send?](#)
- [Help](#)
- [Learn more about this site](#)


[« Back to blog](#)


Windows CardSpace

Do you want to send this card to: www.identityblog.com

Review the data that this site is requesting. To edit the data, name, and picture for this card click Edit.

Tasks
[Edit card](#)
[View card history](#)
[Lock card](#)
[What data will be sent?](#)
[Help](#)

 You have not sent this card to the site. Review the card before you send it.


Demo Card

Personal Card

Card data that will be sent to this site: _____

Fields marked with an asterisk (*) are required

* First Name: Jens
* Last Name: Haeusser
* Email Address: jens.haeusser@ubc.ca
* Site-specific card ID: 6MW-HLUG-SRB

Recent card history (not sent): _____

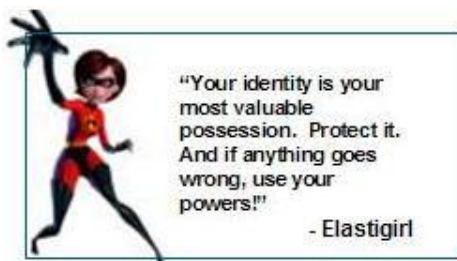
This card has not been used before.

Additional card details (not sent): _____

Created On: 5/29/2007

Include optional data

[« Back to blog](#)



Kim Cameron's Identity Weblog



Comments

[LEAVING A COMMENT](#)
How to use an InfoCard to leave comments without a password

White Papers

[The LAWS OF IDENTITY](#)
The key to this site: an introduction to Digital Identity - the missing layer of the Internet.

[A PRIVACY-COMPLIANT IDENTITY METASYSTEM](#)
The Identity Metasystem: Towards a Privacy-Compliant Solution to the Challenges of Digital Identity

[PRIVACY IN THE LAWS OF IDENTITY](#)
The Case for Privacy-Embedded Laws of Identity in the Digital Age by Commissioner Ann Cavoukian

[INTRODUCING INFOCARD](#)
An excellent introduction to InfoCard technology by David Chappell

[The IDENTITY METASYSTEM](#)
A proposal for building an identity

Roland Dobbins on DDoS attacks and mitigations

Most DDoS attacks these days aren't spoofed, because there's no need, given the zillions of botted computers out there...

Posted on Monday 28 May 2007

Roland Dobbins has written to point out that the recent [Russian cyber-attacks](#) on Estonia are not the first launched by one state against another (he cites incidents during the Balkan conflict, as well as China versus Japan).

Then he gives us an overview of DDoS attacks and mitigations:

DoS attacks are easy to trace as long as Service Providers (SPs) have the proper instrumentation and telemetry enabled on their routers - NetFlow is the most common way of doing this, along with various open-source and commercial tools (nfdump/nfsen, Panoptis, Arbor, Lanclope, Narus, Q1).

Most DDoS attacks these days aren't spoofed, because a) there's no need, given the

Access control attacks

Reference: Jorina Van Malsen



Outline

- **Access Controls**
- **Access Control Vulnerabilities**
- **Securing Access Controls**
- **Attacking Access Controls**



Access Controls

- **A system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system.**
- **Access controls are a critical defense mechanism within the application because they are responsible for making the decision of whether it should permit a given request to perform its attempted action of access the resources that it is requesting.**
- **When they are defective, an attacker can often compromise the entire application, taking control of administrative functionality and accessing sensitive data belonging to every other user.**
- **Are among the most commonly encountered categories of web application vulnerability.**



Common Categories of Vulnerabilities

1. Broken Authentication

- Encompasses various defects within the application's login mechanism

2. Broken Access Controls

- Application fails to properly protect access to data and its functionality

3. SQL Injection

- Enables an attacker to submit crafted input to interfere with the application's interaction with back-end databases.

4. Cross-Site Scripting

- Enables an attacker to target other users of the application

5. Information Leakage

- An application divulges sensitive information that is of use to an attacker in developing an assault against the application, through defective error handling or other behavior



Vertical vs Horizontal Access Controls

- **Vertical Access Controls:**
Allow different types of users to access different parts of the application's functionality
→ Division between ordinary users and administrators
- **Horizontal Access Controls:**
Allow users to access a certain subset of a wider range of resources of the same type
→ Web mail application may allow you to read your email but not one else's; you can only see your own details



Access Control Vulnerabilities

- **Access controls are broken if any user is able to access functionality or resources for which he is not authorized**
- **Among the most commonly encountered categories of web application vulnerabilities**
- **Two main types of attack against access controls**
 - **Vertical privilege escalation**
 - **When a user can perform functions that their assigned role does not permit them to do**
 - **Horizontal privilege escalation**
 - **When a user can view or modify resources to which he is not entitled**



Access Control Security and its Weaknesses

1. Completely Unprotected Functionality
2. Identifier-Based Functions
3. Multistage Functions
4. Static Files



Completely Unprotected Functionality

In many cases of broken access controls, sensitive functionality and resources can be accessed by anyone who knows the relevant URL

→ E.g. when <https://wahh-app.com/admin/> allows user to enter certain user interface.

→ Weaknesses:

1. URL can be guessed (especially by insider)
2. Link appears in browser histories and the logs of web servers and proxy servers
3. Users may write them down, bookmark them or email them around
4. They are not normally changed periodically, as passwords should be
5. When users change job roles, and their access to administrative functionality needs to be withdrawn, there is no way to delete their knowledge of a particular URL.



Identifier-Based Functions

When a function of an application is used to gain access to a specific resource, it is very common to see an identifier for the requested resource being passed to the server in a request parameter, either within the URL query string or the body of a post request

→ When the user who owns the document is logged in, a link to this URL is displayed on the user's My Documents page. Other users do not see this link. In order to be able to open the link/application an attacker needs to know the name of the application page and the identifier of the document he wishes to view.

→ Weaknesses:

1. Passwords often easy to guess
2. Lots of people write down resources identifiers or save them on their computer, so easy to find



Multistage Functions

- Involves capturing different items of data from the user at each stage.
- This data is strictly checked when first submitted and then is usually passed to each subsequent stage, using hidden fields in an HTML form.

Main Weaknesses:

1. Often assumed by the developers is that any user who reaches the later stages of the process must have the relevant privileges because this was verified at the earlier stages
2. Also often assumed is that people will access application pages in the intended sequence; by taking “other path” people could avoid user identification



Static Files

In some cases, requests for protected resources are made directly to the static resources themselves, which are located within the web root of the server.

- e.g. an online publisher may allow users to browse its book catalog and purchase ebooks for download. Once the payment has been made, the user is directed to a download URL.

As this is a completely static resource, it does not execute on the server, and its contents are simply returned directly by the web server. Hence, the resource itself cannot implement any logic to verify that the requesting user has the privileges.

When static resources are accessed in this way, it is highly likely that there are no effective access controls protecting them and that anyone who knows the URL naming scheme can exploit this to access any resources they desire.



Securing Access Controls: Pitfalls

Access controls are one of the easiest areas of web application security, though, there are several obvious pitfalls to avoid:

- Usually arise from ignorance about the essential requirements of effective access control or flawed assumptions about the kinds of requests that users will make and against which the application needs to defend itself
 - Web application developers often implement access control functions on a piecemeal basis, adding code to individual pages in cases where they register that some access control is required, and often cutting and pasting the same code between pages to implement similar requirements.
1. Do not trust any user-submitted parameters to signify access rights (such as admin = true)
 2. Do not assume that users will access application pages in the intended sequence (make sure people will also not be able to avoid access controls by taking a different “path”)
 3. Do not trust the user not to tamper with any data that is transmitted via the client. → If some user-submitted data has been validated and is then transmitted via the client, do not rely upon the retransmitted value without revalidation.



Implementing Effective Access Controls within Web Applications (1)

- **Explicitly evaluate and document the access control requirements for every unit of application functionality.**
- **This needs to include both those who can legitimately use the function and what resources individual users may access via the function.**
- **Remove all access control decisions from the user's session**



Implementing Effective Access Controls within Web Applications (2)

Use a central application component to check access controls

→ Advantages:

1. Increases the clarity of access controls within the application, enabling different developers to quickly understand the controls implemented by others
2. Maintenance more efficient and reliable. Most changes will only need to be applied once, to a single shared component, and will not need to be cut and pasted to multiple locations.
3. It improves adaptability. Where new access control requirements arise, these can be easily reflected within an existing API implemented by each application page
4. In results in fewer mistakes and omissions than if access control code is implemented piecemeal throughout the application



Implementing Effective Access Controls within Web Applications (3)

- Process every single client request via this component to validate that the user making the request is permitted to access the functionality and resources being requested
- Use programmatic techniques to ensure that there are no exceptions to the previous point.
 - An effective approach is to mandate that every application page must implement an interface that is queried by the central access control mechanism.
 - By forcing developers to explicitly code access control logic into every page, there can be no excuse for omissions
- For particularly sensitive functionality, such as administrative pages, you can further restrict access by IP address to ensure that only users from a specific network range are able to access the functionality, regardless of their login status.



Implementing Effective Access Controls within Web Applications (4)

- **If static content needs to be protected, there are two methods of providing access control:**
 - **Static files can be accessed indirectly by passing a file name to a dynamic server-side page which implements relevant access control logic**
 - **Direct access to static files can be controlled using HTTP authentication or other features of the application server to wrap the incoming request and check the permissions for the resource before granting access.**
- **Identifiers specifying which resource a user wishes to access are vulnerable to tampering whenever they are transmitted via the client.**
- **The server should trust only the integrity of server-side data. Any time these identifiers are transmitted via the client, they need to be revalidated to ensure the user is authorized to access the requested resource.**



Implementing Effective Access Controls within Web Applications (5)

- For security-critical application functions such as the creation of a new bill payee in a banking application, consider implementing per-transaction reauthentication and dual authorization to provide additional assurance that the function is not being used by an unauthorized party.
- This will also mitigate the consequences of other possible attacks, such as session hijacking.
- Log every event where sensitive data is accessed or a sensitive action is performed.
- These logs will enable potential access control breaches to be detected and investigated



A Multi-Layered Privilege Model

- **Issues relating to access apply not only to the web application itself but also to the other infrastructure ties which lie beneath it**
- **In this case, these access controls could be a good alternative:**
 1. **Programmatic Control**
 2. **Discretionary Access Control (DAC)**
 3. **Role-Based Access Control (RBAC)**
 4. **Declarative Control**



Programmatic Control

- **The matrix of individual database privileges is stored in a table within the database, and applied programmatically to enforce access control decisions.**
- **The classification of user roles provides a shortcut for applying certain access control checks, and this is also applied programmatically**
- **Advantages:**
 - **It can be extremely fine-grained**
 - **It can build in arbitrarily complex logic into the process of carrying out access control decisions within the application**



Discretionary Access Control (DAC)

Various application users have privileges to create user accounts

Closed DAC Model

Access denied unless explicitly granted

Open DAC Model

Access is permitted unless explicitly with-drawn



Role-Based Access Control (RBAC)

- **Named roles which contain different sets of specific privileges. Each user is assigned to one of these roles.**
- **Enables many unauthorized requests to be quickly rejected with a minimum amount of processing being performed**
- **Number of roles should be balanced**
 - Too many roles → Difficult to manage accurately
 - Too few roles → Resulting roles will be assigned privileges that are not strictly necessary for performance of their function



Declarative Control

- **Uses restricted database accounts when accessing the database**
- **Employs different accounts for different groups of users with each account having the least level of privilege necessary for carrying out the actions which that group is permitted to perform**
- **Advantage: Even if a user finds a means of breaching the access controls implemented within the application tier, so as to perform a sensitive action such as adding a new user, they will be prevented from doing so because the database account that they are using does not have the required privileges within the database**



Attacking Access Controls

Finding a break in access controls is almost trivial

- **Request a common administrative URL and gain direct access to the functionality.**
- **In other cases, it may be very hard, and subtle defects may lurk deep within application logic, particularly in complex, high-security applications.**
- **The most important lesson when attacking access controls is to look everywhere. If you are struggling to make progress, be patient and test every single step of every application function. A bug that allows you to own the entire application may be just around the corner.**



Identity and access provisioning lifecycle (e.g. provisioning review)

Reference: Brian Brekkan, Microsoft



Agenda

- Business and IT Challenges
- Business Ready Security
- Identity and Access Management
- The Road Ahead
- Summary



Business Needs and IT Challenges

Provide secure access to applications from anywhere

Simplify user experience for collaboration

Provide seamless movement between applications

Reduce cost of account management

Multiple locations and devices

Difficulty in extending business resources

Disparate systems to manage

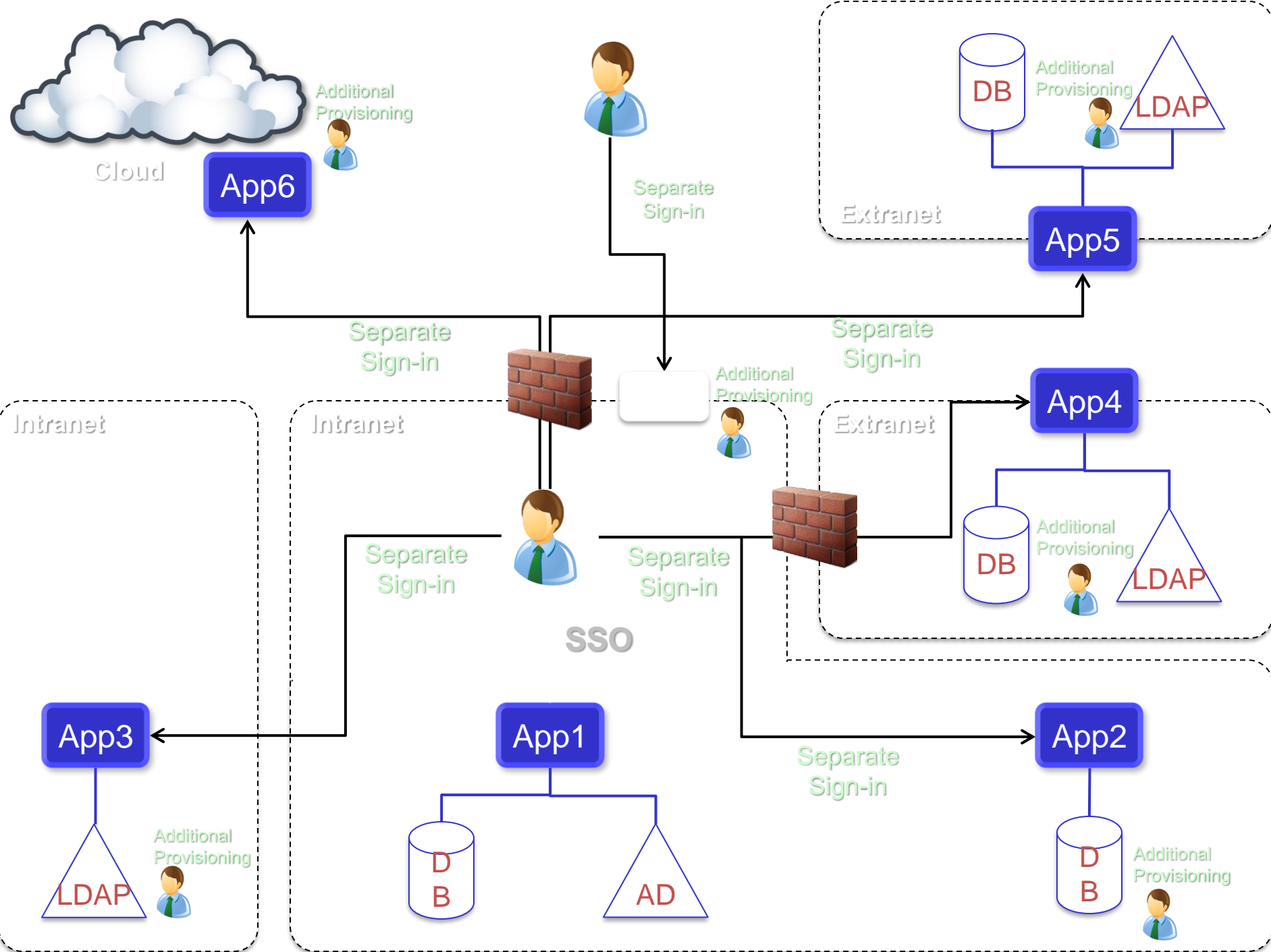
Complex account lifecycle management



BUSINESS Needs
Agility and **Flexibility**



IT Needs
Control



Business Ready Security

Help securely enable business by managing risk and empowering people

Protect everywhere,
access anywhere



Simplify the security
experience,
manage compliance

Integrate and extend
security across the
enterprise

<i>from:</i>	<i>to:</i>
Block	Enable
Cost	Value
Siloed	Seamless



Business Ready Security Solutions

Secure Messaging

Secure Collaboration

Secure Endpoint

Information Protection

Identity and Access Management

 Microsoft®
Forefront™
Identity Manager

 Microsoft®
Forefront™
Unified Access Gateway

 Windows 7

 Windows Server® 2008 R2

 Windows Server® 2008 R2
Active Directory® Federation Services



Mississippi State University Center for Cyber Innovation

Domain 5 Identity and Access Management



194

The Products

Forefront Identity Manager

Unified Access Gateway

AD Federation
Services

AD Domain
Services

AD Certificate
Services

AD Lightweight
Directory Services

Active Directory

Windows Identity
Foundation

.Net Framework

Windows CardSpace

Windows Server and Windows Client
Identity and Access Management Solution

Partner and Custom Solutions

Identity and Access Management

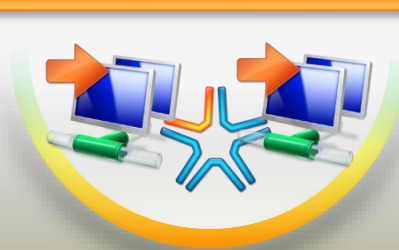
Enable more secure, identity-based access to applications on-premises and in the cloud from virtually any location or device

PROTECT everywhere
ACCESS anywhere



- Provide more secure, always-on access
- Enable access from virtually any device

INTEGRATE and
EXTEND security



- Control access across organizations
- Provide standards-based interoperability

SIMPLIFY security,
MANAGE compliance



- Extend powerful self-service capabilities to users
- Automate and simplify management tasks



UAG and DirectAccess better together:

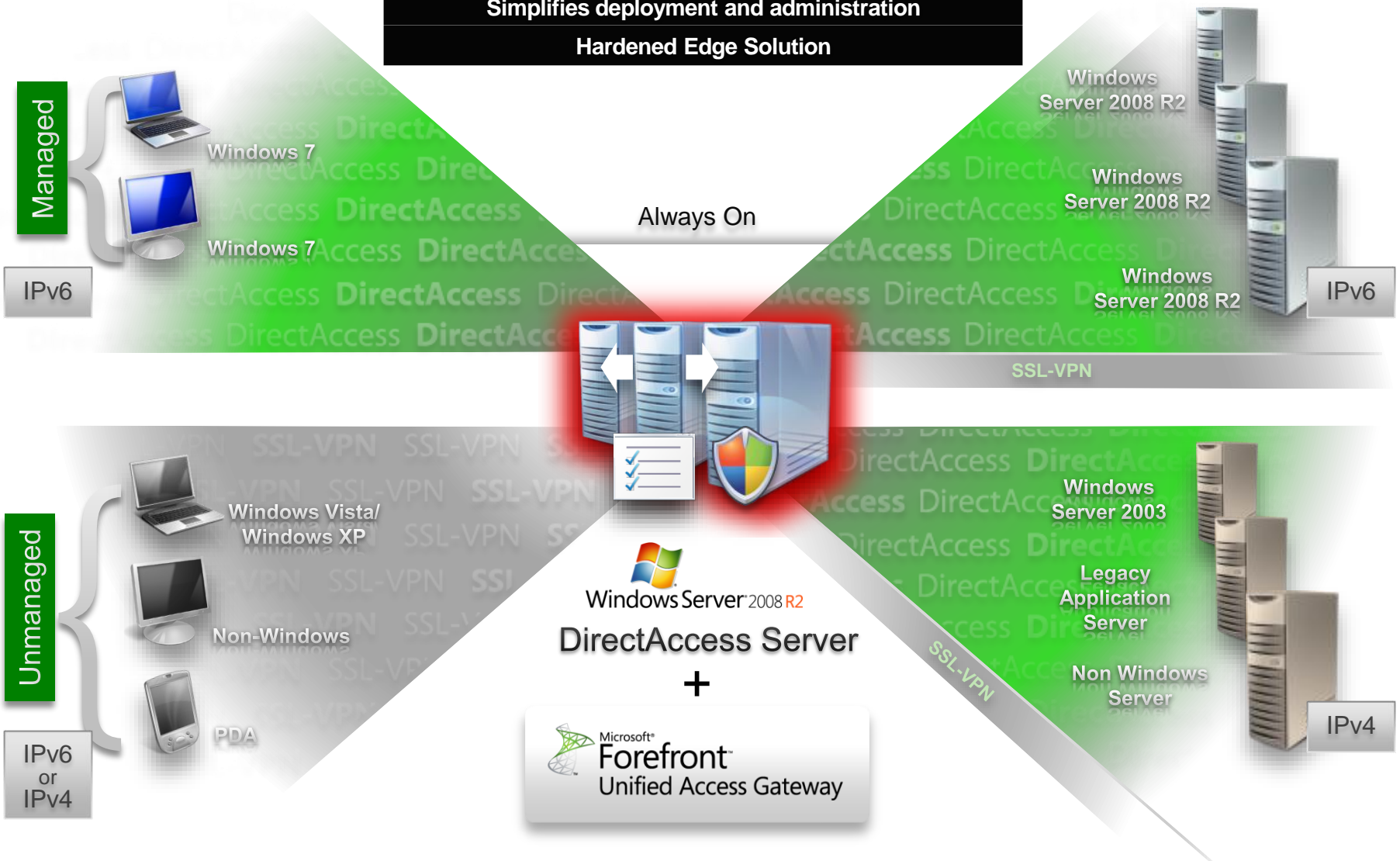
Extends access to line of business servers with IPv6 support

Access for down level and non Windows clients

Enhances scalability and management

Simplifies deployment and administration

Hardened Edge Solution



Identity and Access Management

Enable more secure, identity-based access to applications on-premises and in the cloud from virtually any location or device

PROTECT everywhere
ACCESS anywhere



- Provide more secure, always-on access
- Enable access from virtually any device

INTEGRATE and
EXTEND security



- Control access across organizations
- Provide standards-based interoperability

SIMPLIFY security,
MANAGE compliance



- Extend powerful self-service capabilities to users
- Automate and simplify management tasks

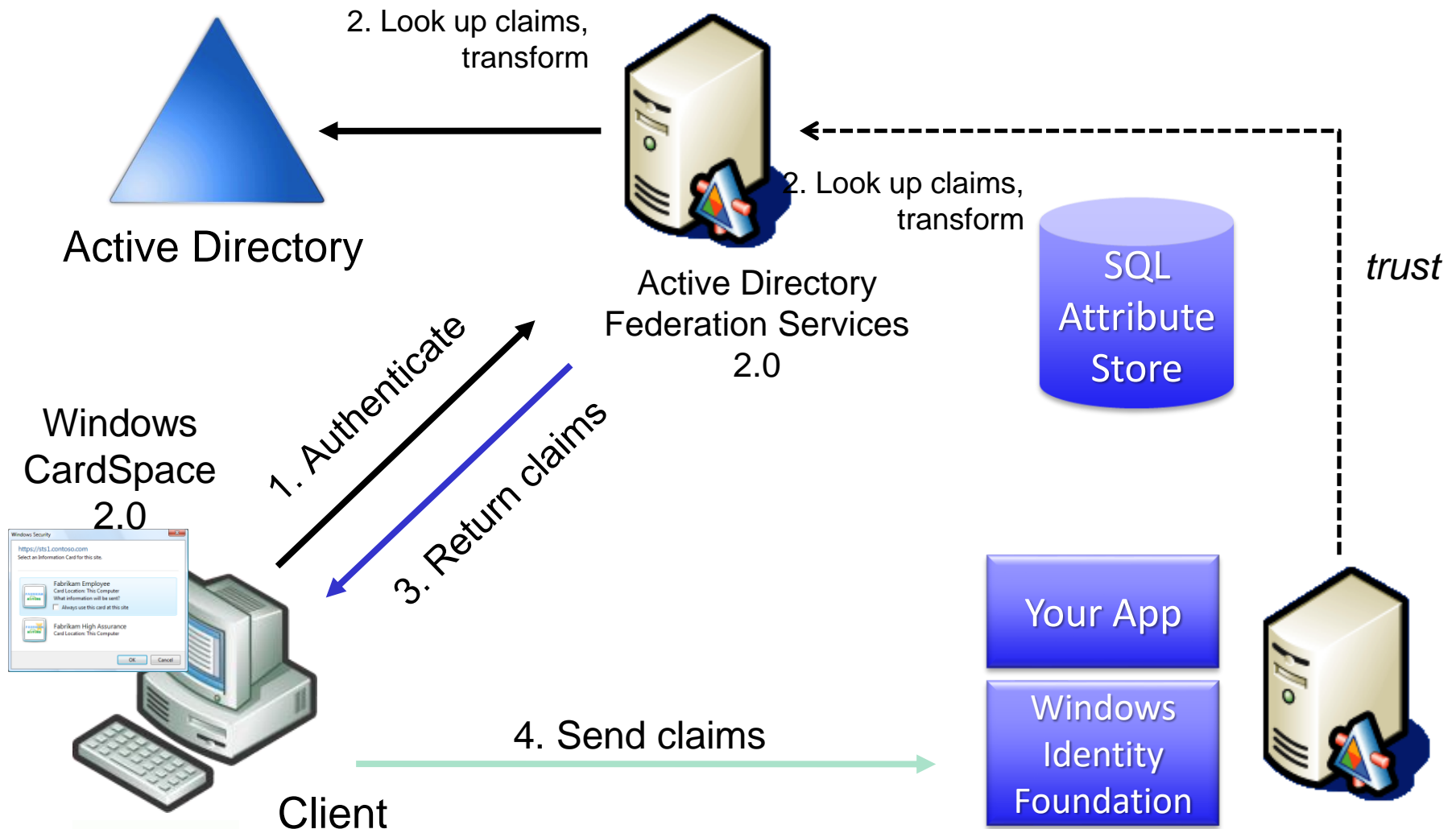


Authentication Problem Statement

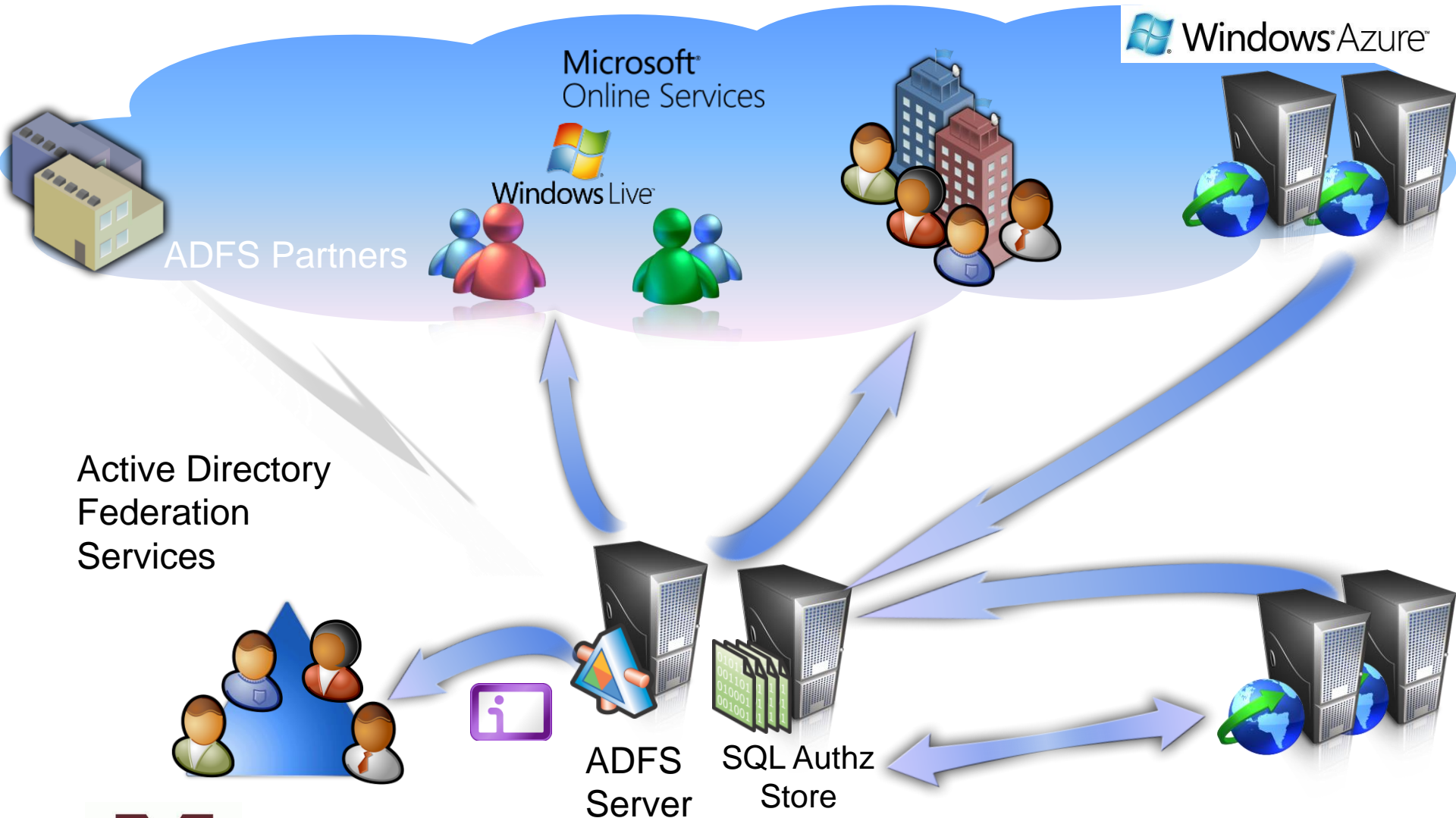
- **Every connected app must handle two functions**
 - Authenticate user
 - Get information about user to drive app behavior
- **Many different technologies to do this**
 - Name/password, X.509, Kerberos, SAML, LDAP, ...
 - Scenario drives technology choice
 - App becomes bound to constraints of technology
- **Solution: claims-based identity**
 - Abstraction layer hides detail of authenticating user, getting information about user
 - Application logic exposed to *claims* only; claims = information about the user
 - Change details after deployment without changing application code



What is claims based access?



How ADFS is Changing the Game



Forefront Identity Manager – Features



Policy Management

- SharePoint-based console for policy authoring, enforcement & auditing
- Extensible WS- * APIs and Windows Workflow Foundation workflows
- Heterogeneous identity synchronization and consistency



Credential Management

- Heterogeneous certificate management with 3rd party CAs
- Management of multiple credential types
- Self-service password reset integrated with Windows logon



User Management

- Integrated provisioning of identities, credentials, and resources
- Automated, codeless user provisioning and de-provisioning
- Self-service profile management



Group Management

- Rich Office-based self-service group management tools
- Offline approvals through Office
- Automated group and distribution list updates



Summary

Enable more secure, identity-based access to applications on-premises and in the cloud from virtually any location or device

PROTECT everywhere
ACCESS anywhere



- Provide more secure, always-on access
- Enable access from virtually any device

INTEGRATE and
EXTEND security



- Control access across organizations
- Provide standards-based interoperability

SIMPLIFY security,
MANAGE compliance



- Extend powerful self-service capabilities to users
- Automate and simplify management tasks

Learn more at: www.microsoft.com/forefront



Mississippi State University Center for Cyber Innovation

Domain 5 Identity and Access Management



203

Resources

Microsoft®
tech.ed
Online

www.microsoft.com/teched

Sessions On-Demand &
Community

Microsoft® **TechNet**

<http://microsoft.com/technet>

Resources for IT Professionals

Microsoft® | Learning

www.microsoft.com/learning

Microsoft Certification & Training
Resources

msdn®

<http://microsoft.com/msdn>

Resources for Developers



Summary

- **Physical and logical assets control**
- **Identification and authentication of people and devices**
- **Identity as a service (e.g. cloud identity)**
- **Third-party identity services (e.g. on-premise)**
- **Access control attacks**
- **Identity and access provisioning lifecycle (e.g. provisioning review)**

