INF3510 Information Security
University of Oslo
Spring 2012

Lecture 8

Identity and Access Management

Audun Jøsang

Outline

- Identity and access management concepts
- Identity management models
- Access control models (security models)

Identity & access management

Identity

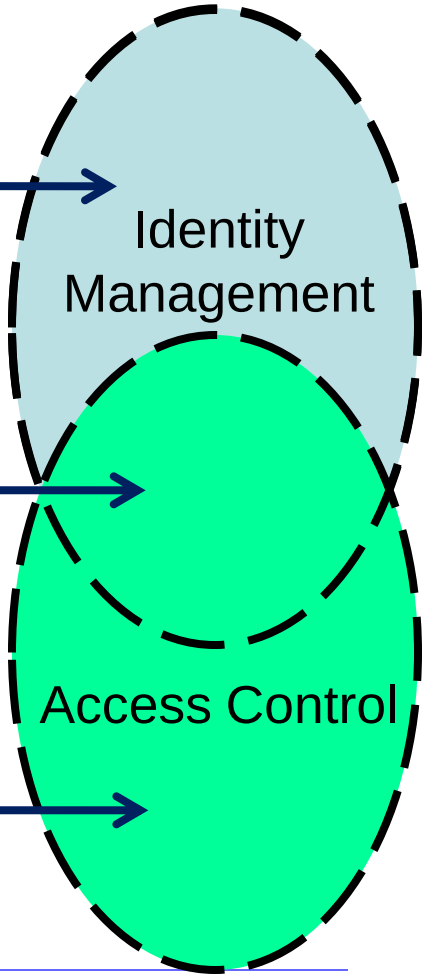
- Representing and entities as digital identities
- Managing name spaces of unique identifiers
- Mapping identities between domains

Authentication

- Registration
- Provisioning
- Authentication

Access

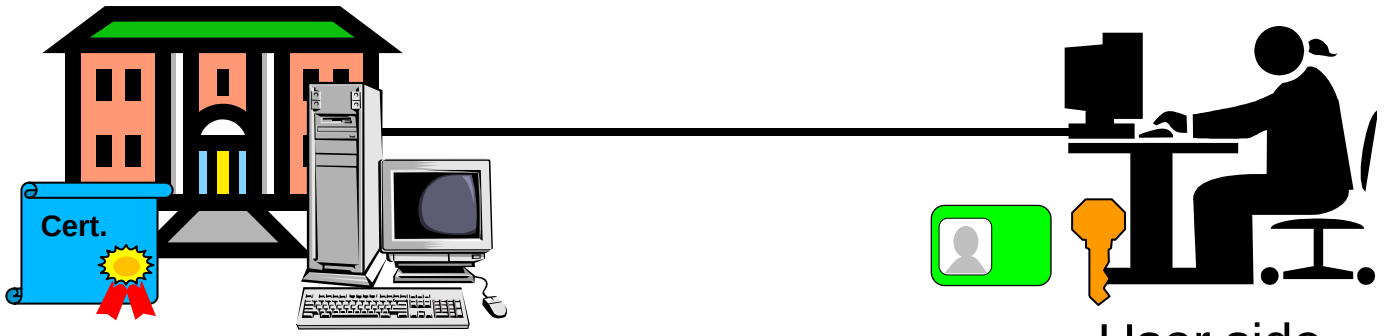
- Authorization
- Access approval
- Accounting



AAA:
"Authentication, Authorization & Accounting"

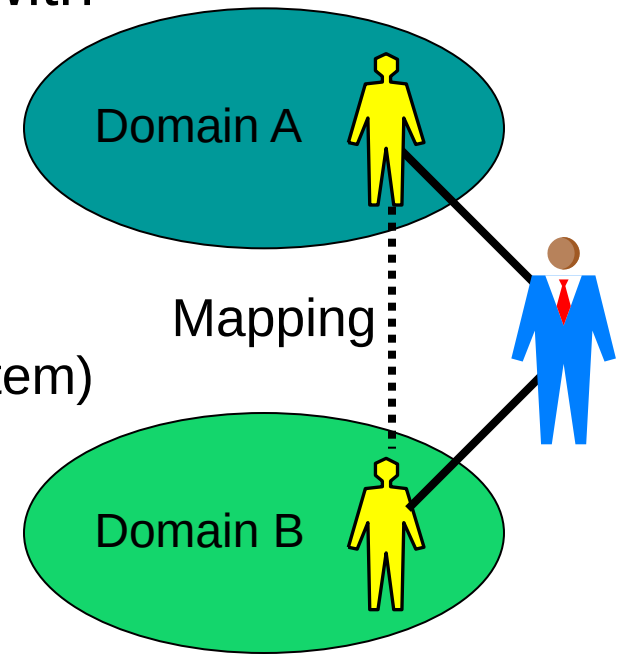
Identity management types

(1) Mgmt of user IDs and credentials on SP side	(2) Mgmt of user IDs and credentials on user side
(3) Mgmt of SP IDs and credentials on SP side	(4) Mgmt of SP IDs and credentials on user side

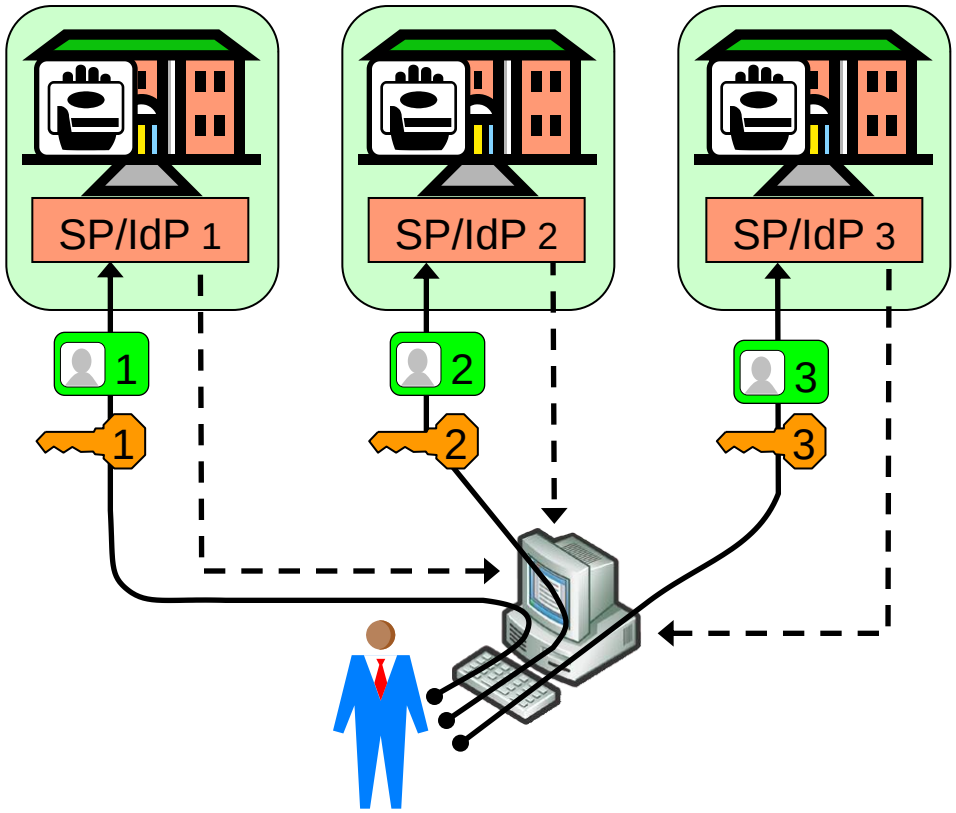


Identity Domains

- An identity domain is a network realm with a name space of unique names
- Management structures:
  - Single authority, e.g. User Ids in company network
  - Hierarchical: e.g. DNS (Domain Name System)
- A single policy is normally applied in a domain
- Integration/federation of domains
  - Requires mapping of identities of same entity
  - Requires alignment of policies



Silo domain model



Legend:

SP

IdP

Identity domain

User identifier managed by IdP #

Authentication token managed by IdP #

Service logon

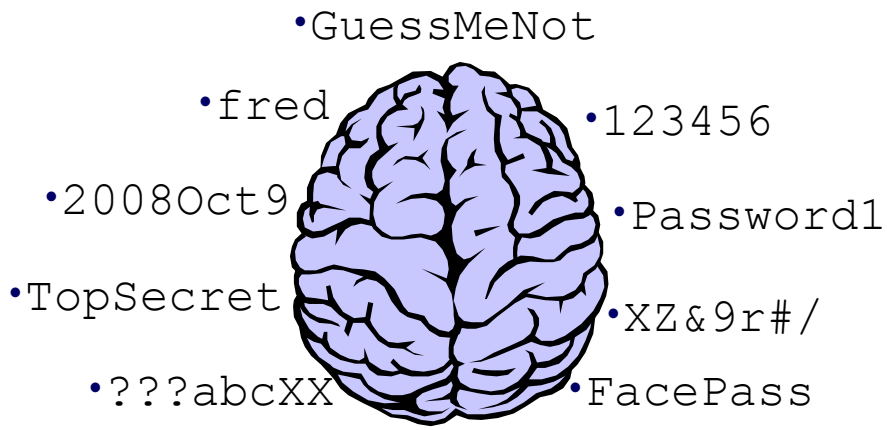
Service provision

Silo user-identity domains

- SP = IdP: defines name space and provides access credentials
- Unique identifier assigned to each entity
- Advantages
  - Simple to deploy, low cost for SPs
- Disadvantages
  - Identity overload for users, poor usability

Tragedy of the Commons

Common grazing field



Common brain

Common pockets

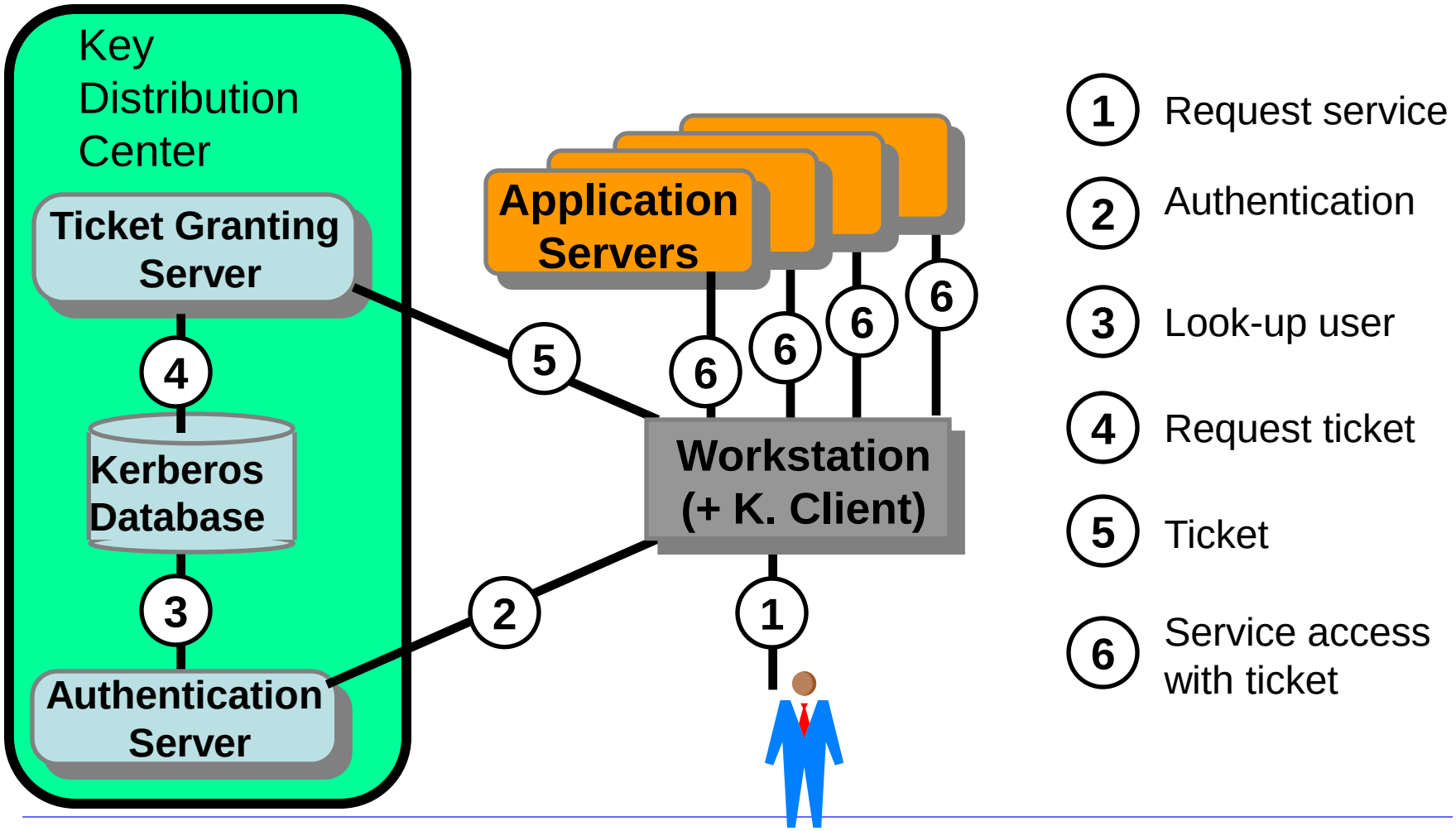
Push towards SSO (Single Sign-On)

- Users don’t want more identifiers
- Low acceptance of new services that require separate user authentication
- Silo model requires users to provide same information to many service providers
- Silo model makes it difficult to offer bundled services, i.e. from different service providers
- Service providers want better quality user information

Kerberos SSO

- Part of project Athena (MIT) in 1983.
- User must identify itself once at the beginning of a workstation session (login session).
- Server authenticates Kerberos client on user's workstation instead of authenticating user
  - So user does not need to enter password every time a service is requested!
- Every user shares a password with the AS (Authentication Server)
- Every SP (service provider) shares a secret key with the TGS (Ticket Granting Server)
- Tickets are sealed (encrypted) by TGS proves to SPs that the user has been authenticated

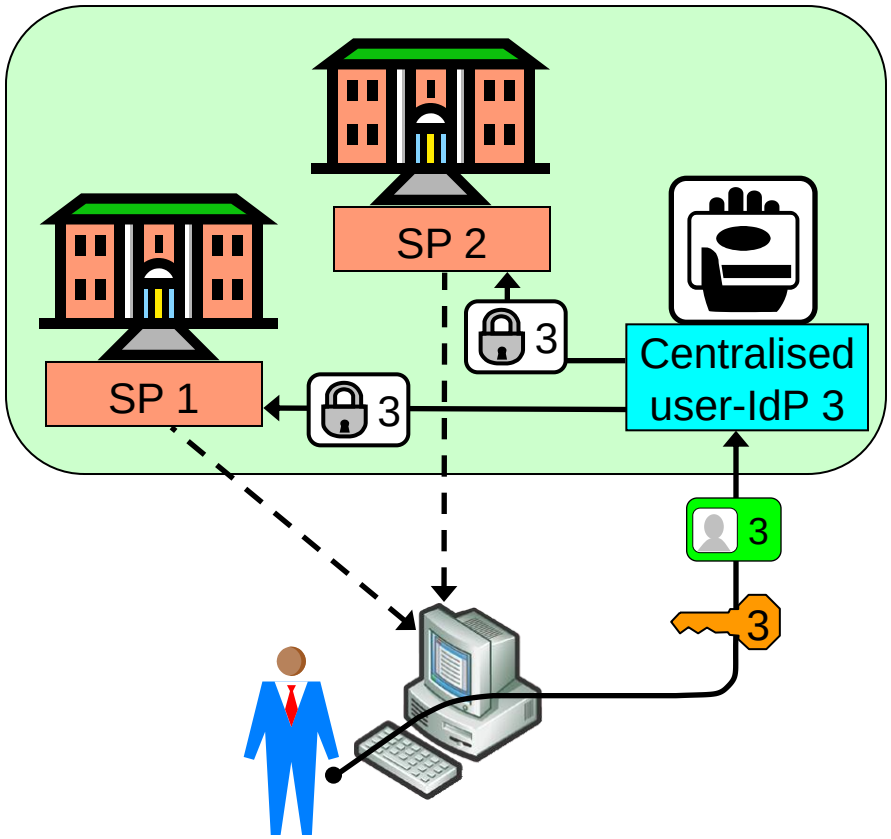
Kerberos – simplified protocol



Kerberos – Advantages and limitations

- First practical SSO solution
- Centralized TTP (Trusted Third Party) model
- Uses only symmetric cryptography
- Requires Kerberos clients and servers + KDC
- Only suitable for organisations under common management (single domain)
- Does not scale to very large domains
- Not suitable for open environments (Internet)

Single Domain SSO



Examples: Kerberos, .net Passport

Legend:

SP

IdP

Identity domain

User identifier issued by IdP #

Security assertion sent by IdP #

Authentication token managed by IdP #

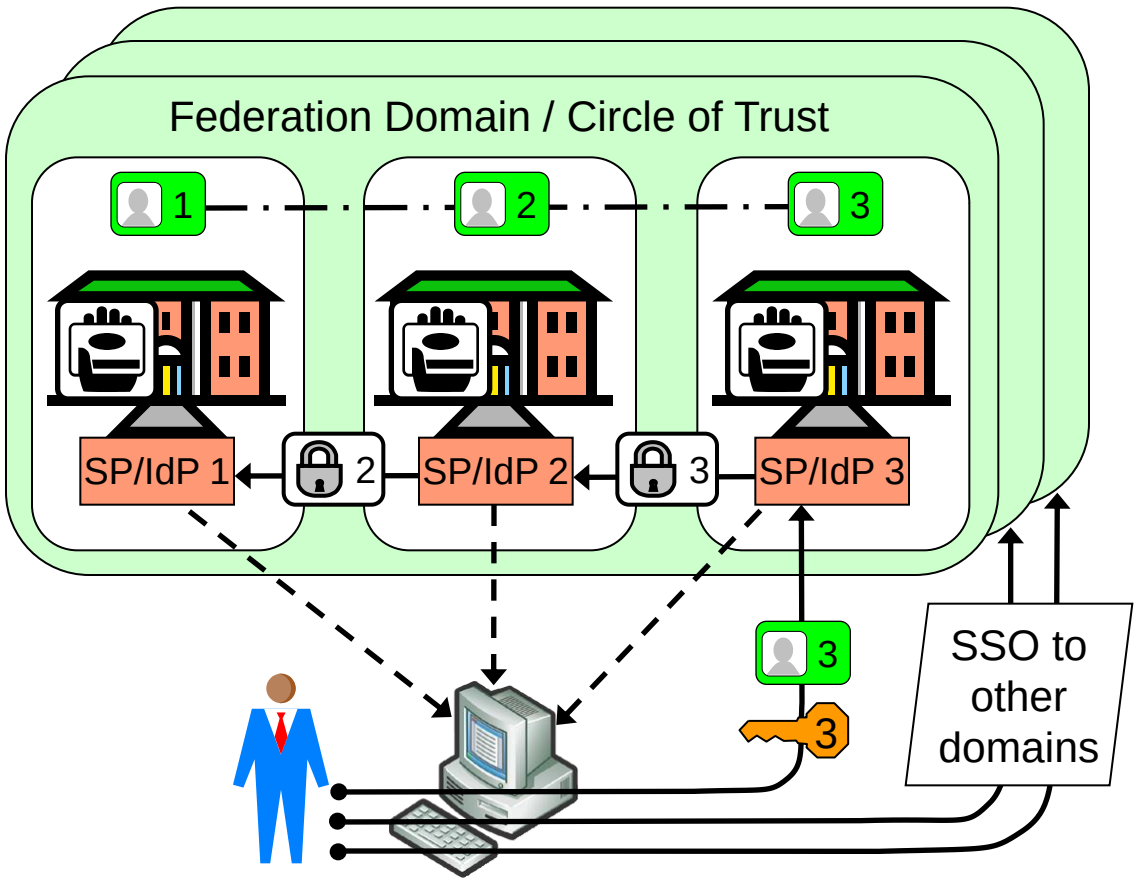
Service logon

Service provision

Single Domain SSO

- Single authority that acts as IdP (identity provider) and credentials provider
- Single authority authenticates users
- Advantages
  - Well suited for servers (SPs) under single management, e.g. within large private and government organisations
  - Good usability
- Disadvantages
  - Politically and technically difficult to implement in open environments.
  - Who trusts authentication by other organisations?
  - Requires Kerberos client on all user workstations

Federated SSO model

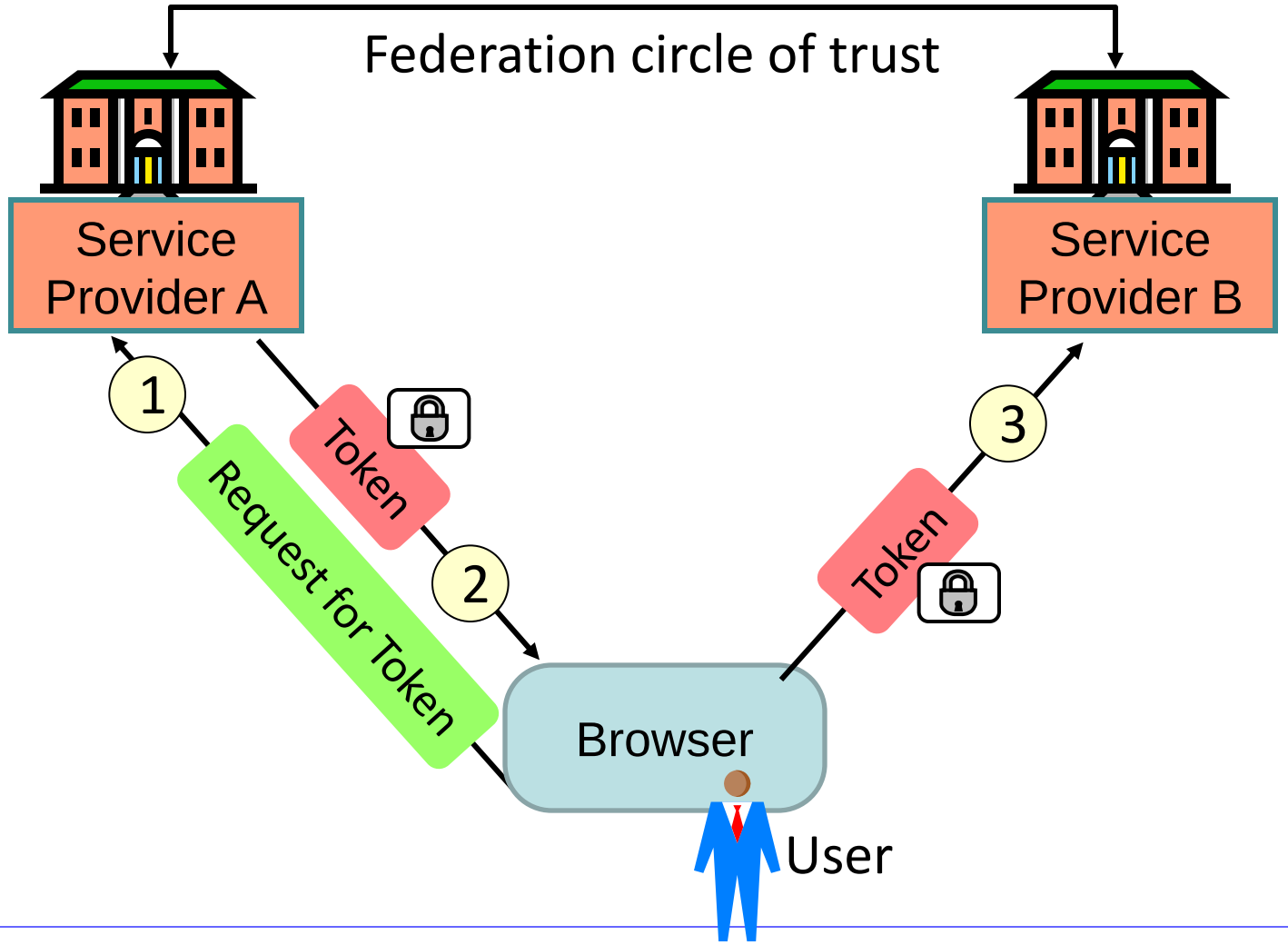


Legend :

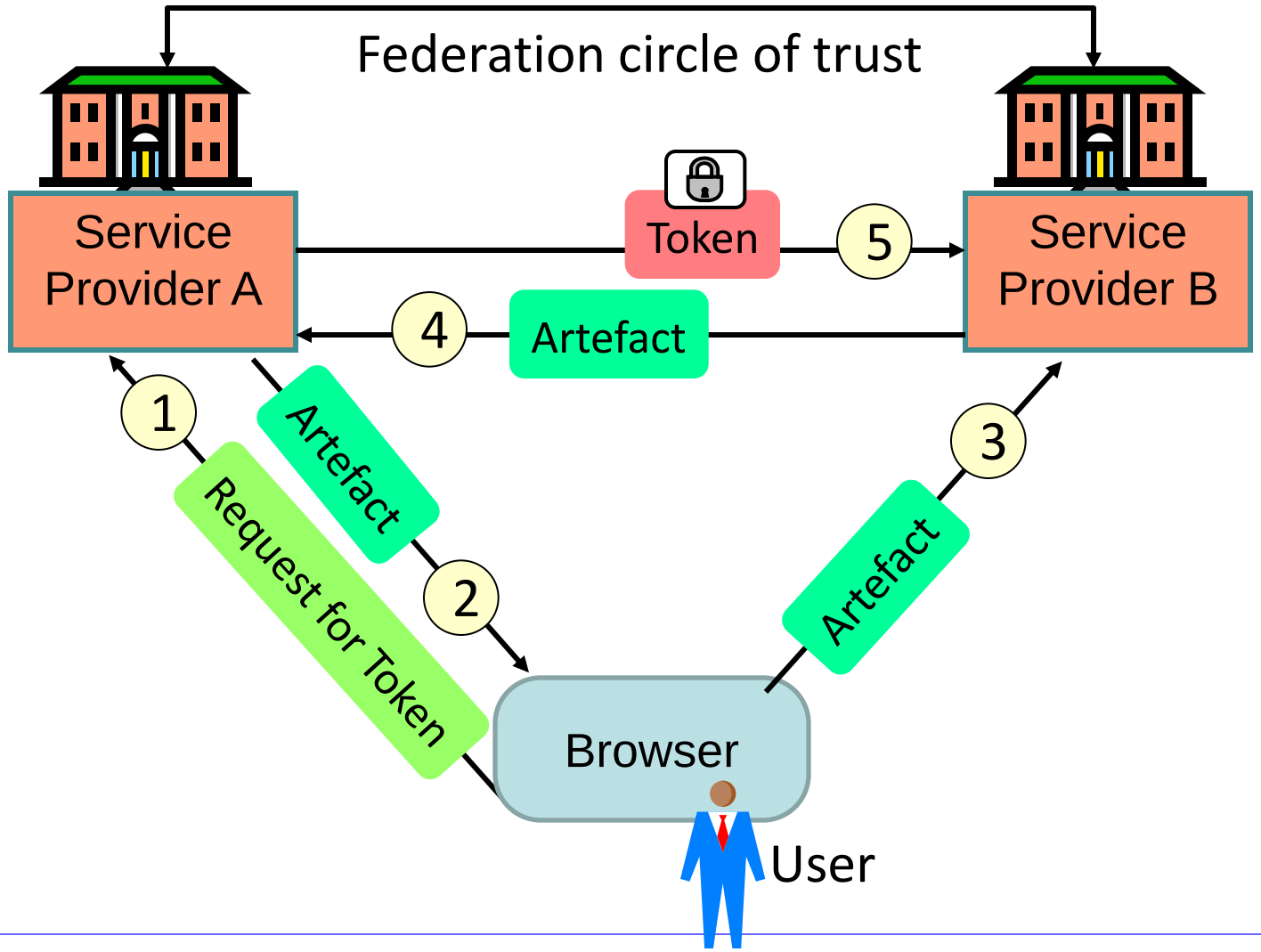
- SP
- IdP
- Identity domain
- # — User identifier issued by IdP #
- # — Authentication token managed by IdP #
- # — Security assertion sent by IdP #
- Service logon
- Service provision
- Identifier mapping

Examples: Liberty Alliance, SAML2.0, WS-Federation, Shibboleth

SAML protocol profile: Browser Post Security token via front-end



SAML protocol profile: Browser Artefact Security token via back-end



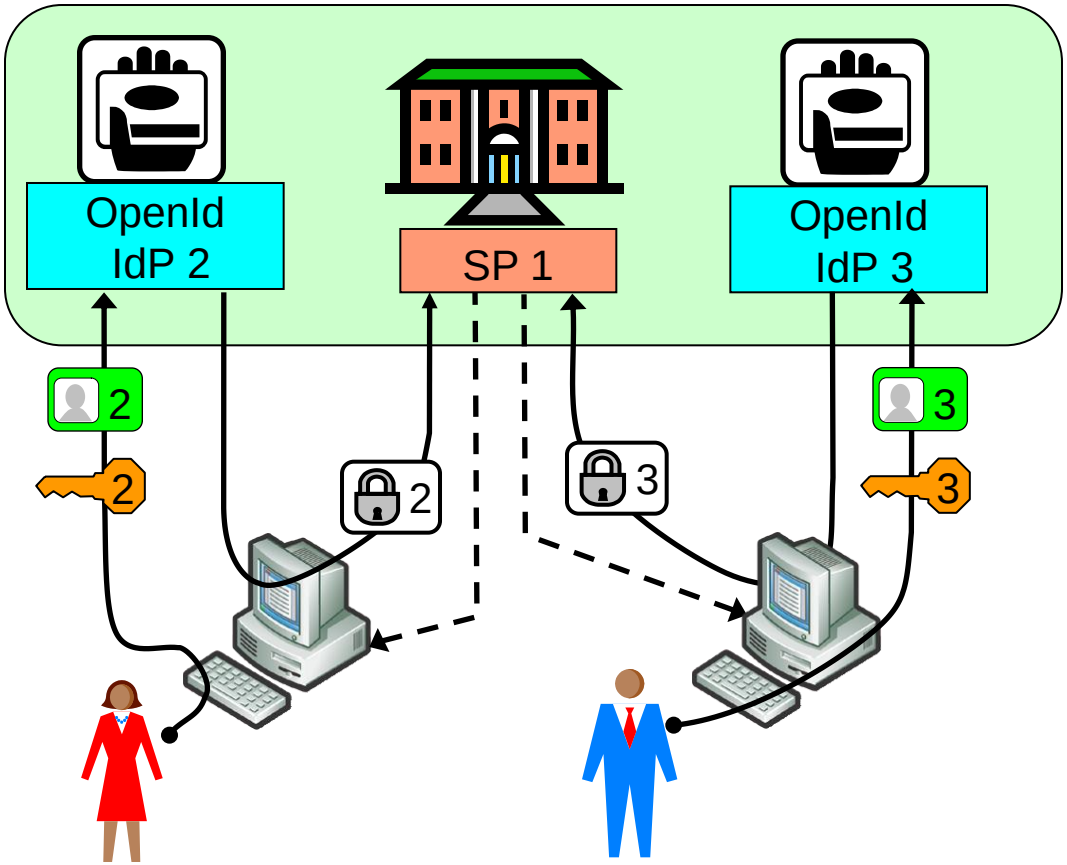
Federated SSO

- Identity Federation
  - A set of agreements, standards and technologies that enable a group of SPs to recognise user identities and entitlements from other SPs
  - Identifier (and credential) issuance as for the silo model
  - **Mapping** between a user's different unique identifiers
  - Authentication by one SP, communicated as security assertions to other SPs
  - Provides SSO in open environments

Federated SSO

- Advantages
  - Improved usability (theoretically)
  - Compatible with silo user-identity domains
  - Allows SPs to bundle services and collect user info
- Disadvantages
  - High technical and legal complexity
  - High trust requirements
    - E.g. SP1 is technically able to access SP2 on user's behalf
  - Privacy issues
  - Unimaginable for all SPs to federate,
    - multiple federated SSOs not much better than silo model

Open identity model



Legend :

SP

IdP

Common identity domain

User identifier managed by IdP #

Authentication token managed by IdP #

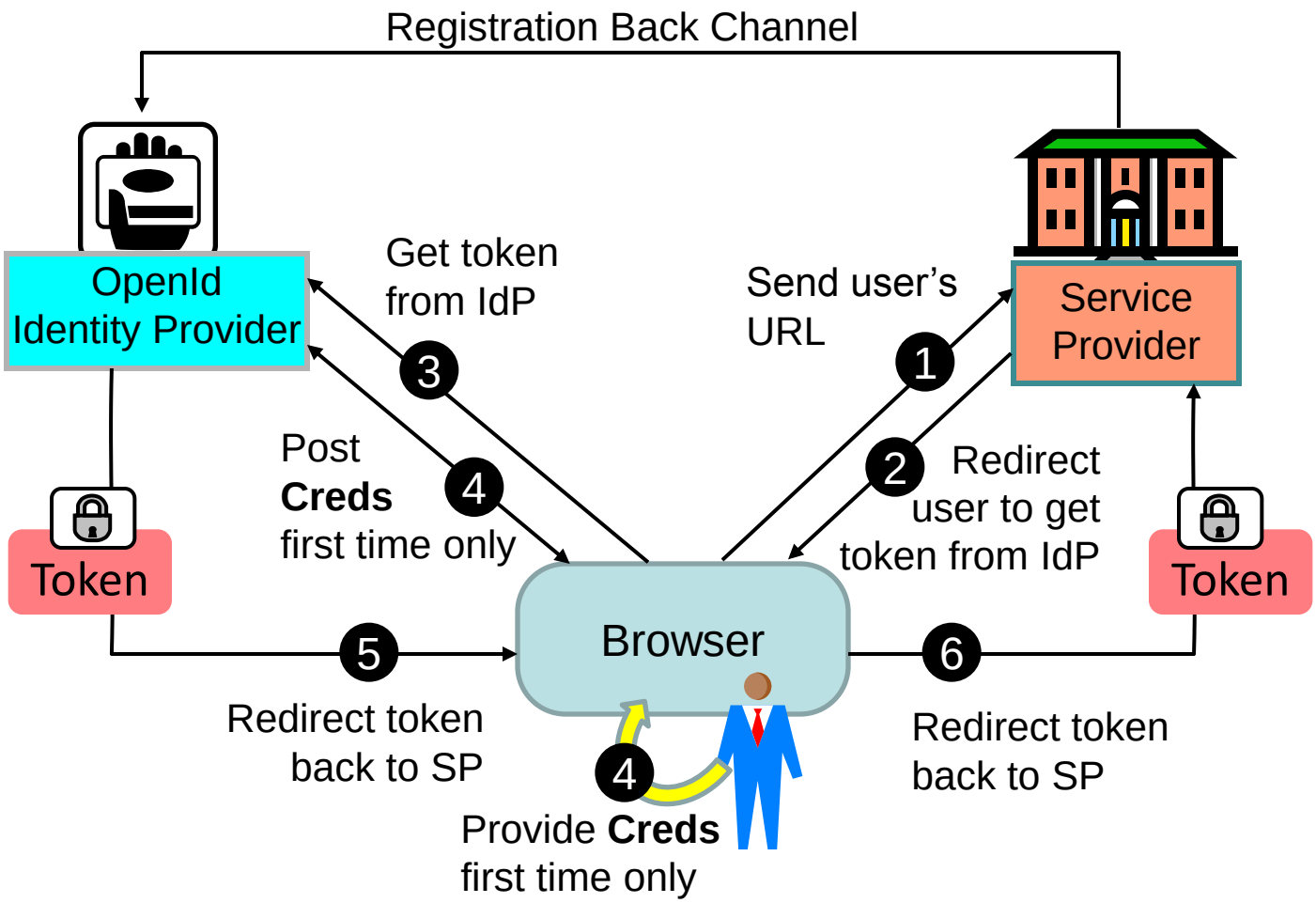
Security assertion issued by IdP #

Service logon

Service provision

Example: OpenID

OpenId authentication protocol - details

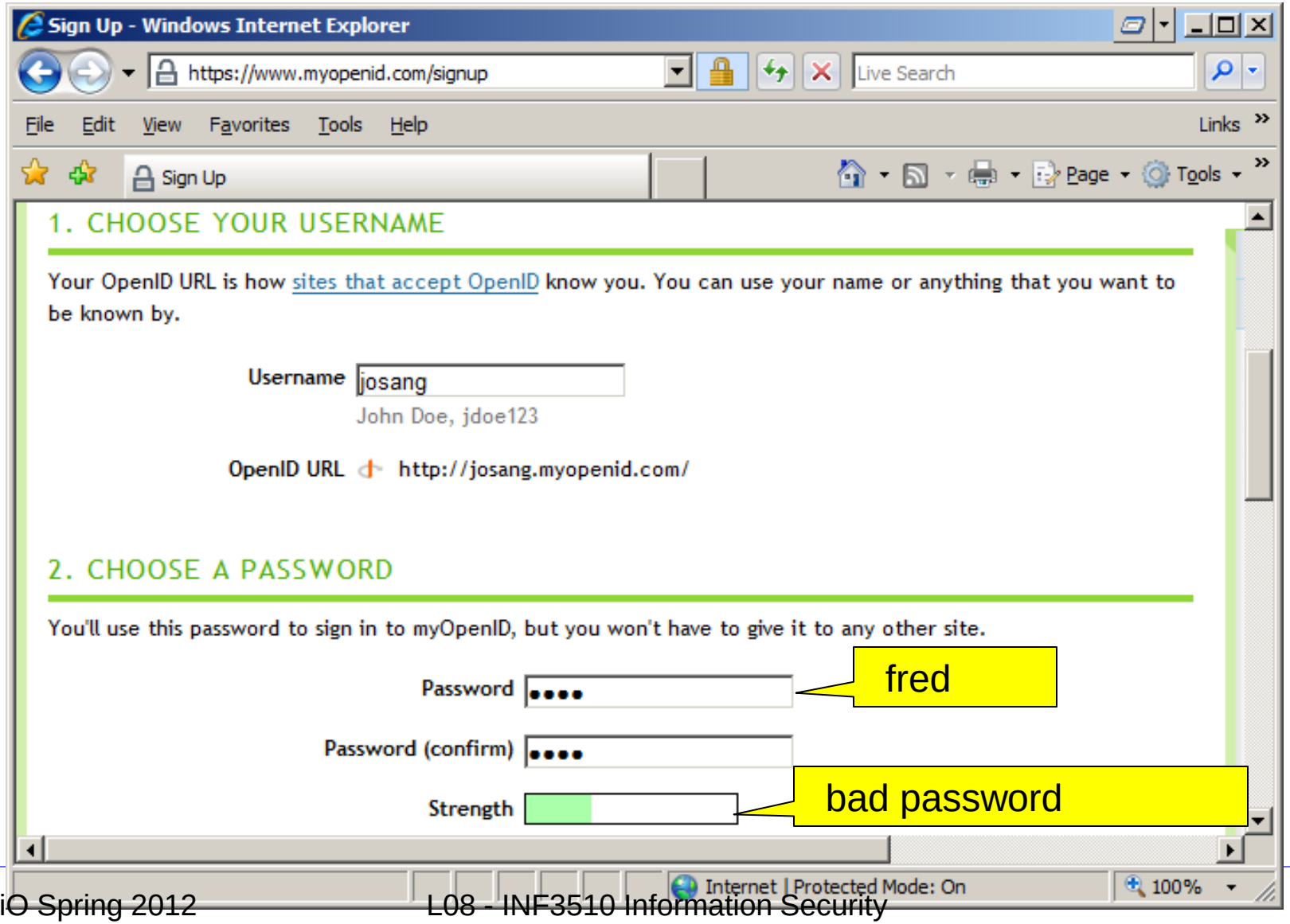


Open identity model

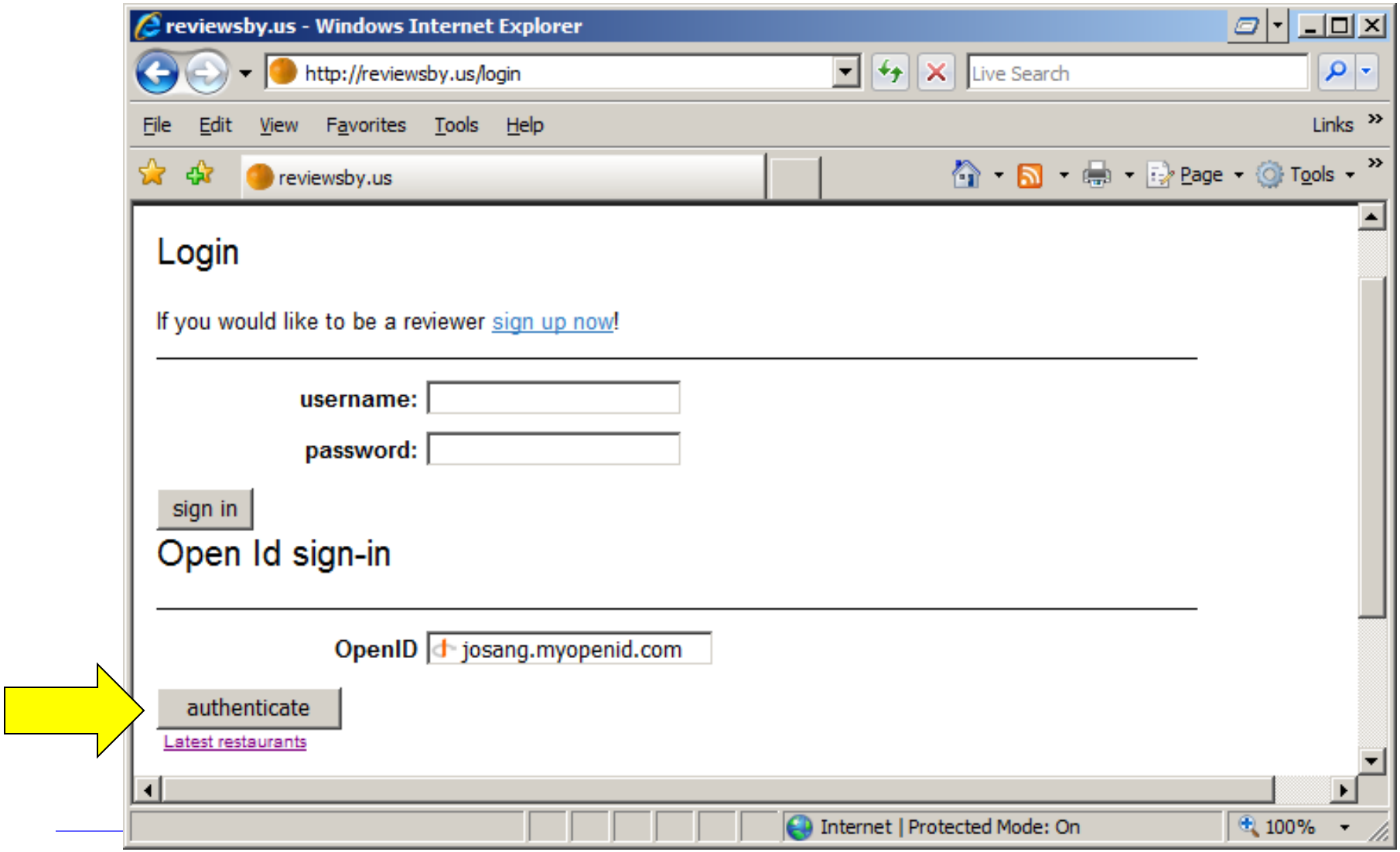
- Single common identifier name space
  - E.g. based on URIs or XRis
- Multiple Identity Providers
  - Each IdP controls its own domain name
  - Registers users under own domain name
- Whoever controls a domain name can be IdP

- IdPs are involved in every service access
  - Collect info about service access

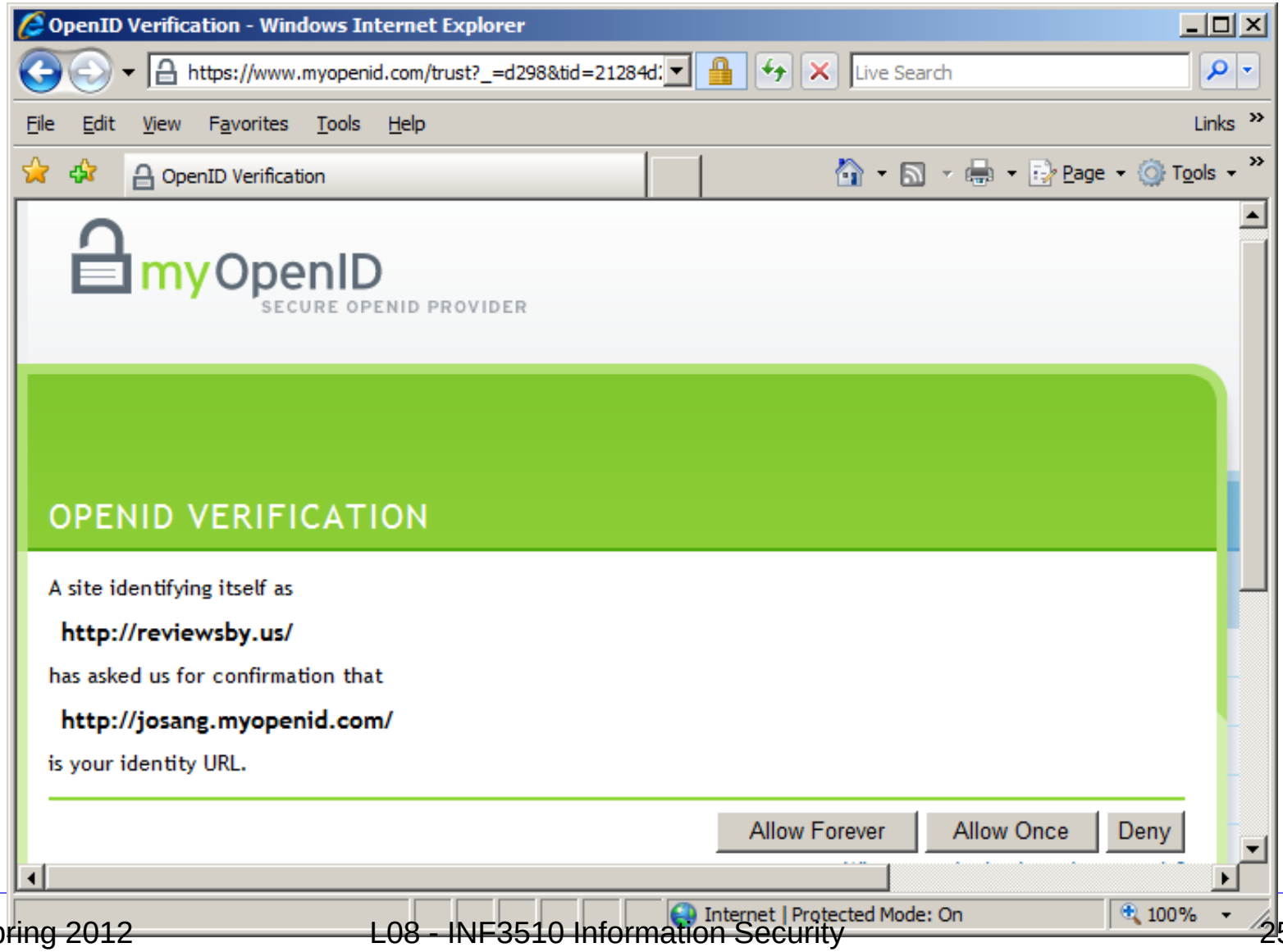
OpenID self registration



Service Access Without Password



First Time Sevice Access



OpenID Characteristics

- Self registration
- Anybody can be IdProvider and Server, also you
- Not all IdProviders are recognised as ”authorities”
- A SP can specify which IdPs it accepts
- Not suitable for sensitive services
- Typically targets online services with AAL-1
- Vulnerable to multiple forms of abuse

OpenID Business Model

- For ID Providers
  - Collection of market data
  - Knows who uses which service
  - Fragmentation of ID Provider market is a threat
- For Service Providers (Relying Party)
  - Potentially more traffic and business
- For users
  - Avoid multiple identities
  - Avoids typing passwords
  - (Must still type OpenID identifier)

Four national identity federations

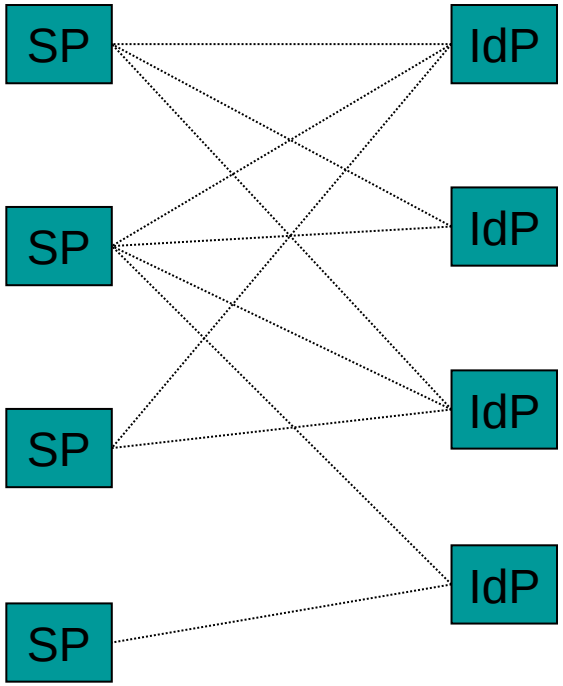
Haka (Finland): Operational (Shibboleth)

FEIDE (Norway): Operational (Moria, SAML2.0)

DK-AAI (Denmark): Piloting (A-Select)

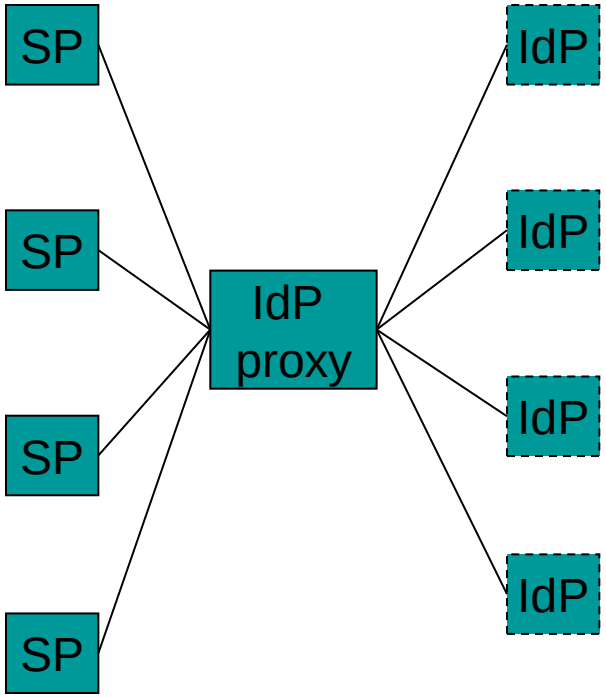
SWAMID (Sweden): Piloting (Shibboleth)

Technical shape of a federation: Distributed



- Model deployed by Haka (.fi), SWAMID (.se) and several other federations
- Pros
  - No single point of failure in the message flow
  - Costs of federation management low
- Cons
  - Hard to track errors and
  - Not well supported by commercial products

Technical shape of a federation: Centralized



- Model deployed by FEIDE (.no) and WAYF (.dk)
- Pros
  - A single point where to locate problems and introduce new features
  - Economics of scale
- Cons
  - A single point of failure
  - Everyone needs to trust the IdP in the middle

FEIDE (Felles Elektronisk Identitet)

- FEIDE is a system for Id management within the Norwegian national education sector.
- Users have only one username and password
- Users access web-services via a central log-in service
- Services are given what they need to know about the user
- Services are not given the users password/credential, only information about the user

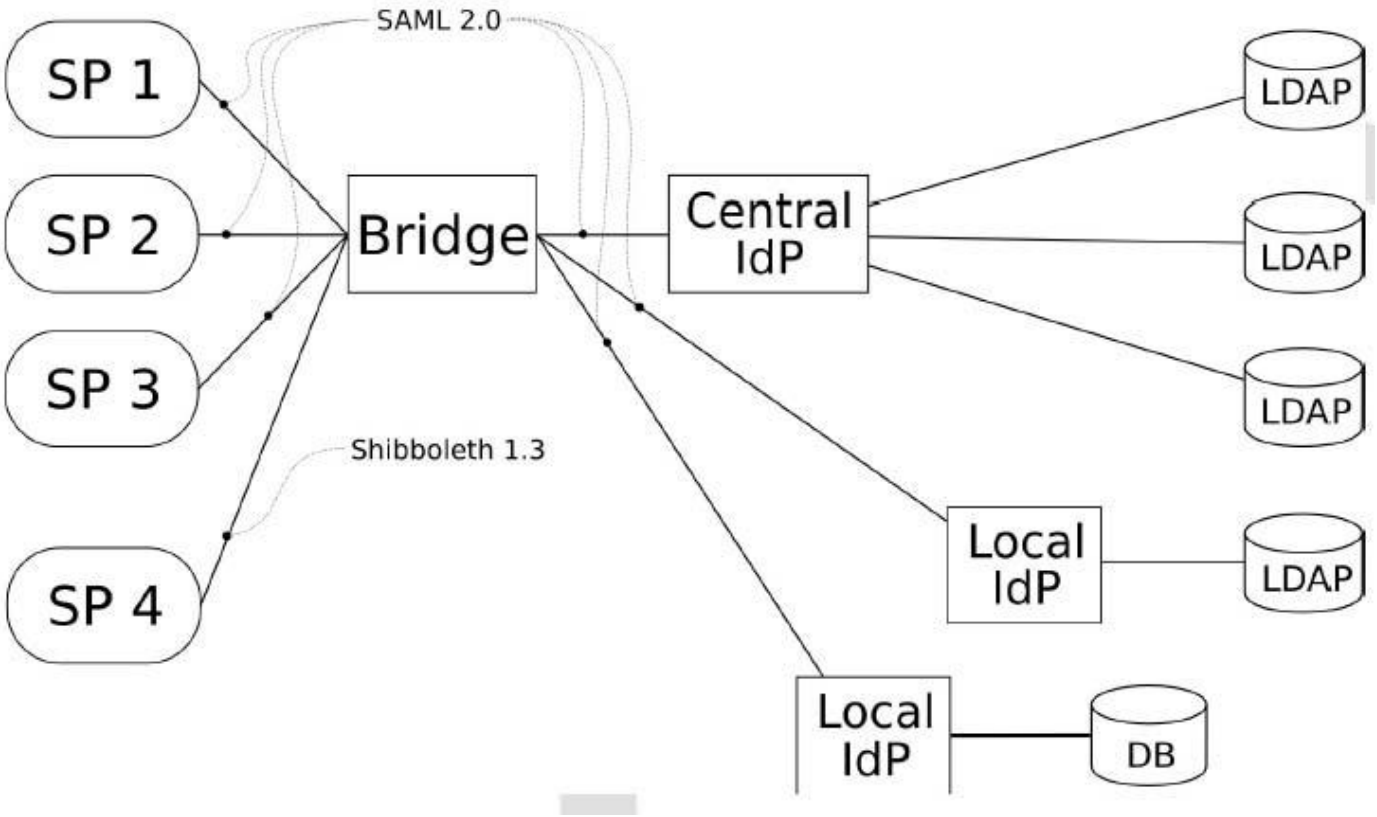
FEIDE (continued)

- FEIDE have formal agreements with the schools before they are connected
- The home organizations (schools) are responsible for the data about the users (correct and up-to-date)
- Home organizations decide themselves what services their users should be able to access via the central log-in service

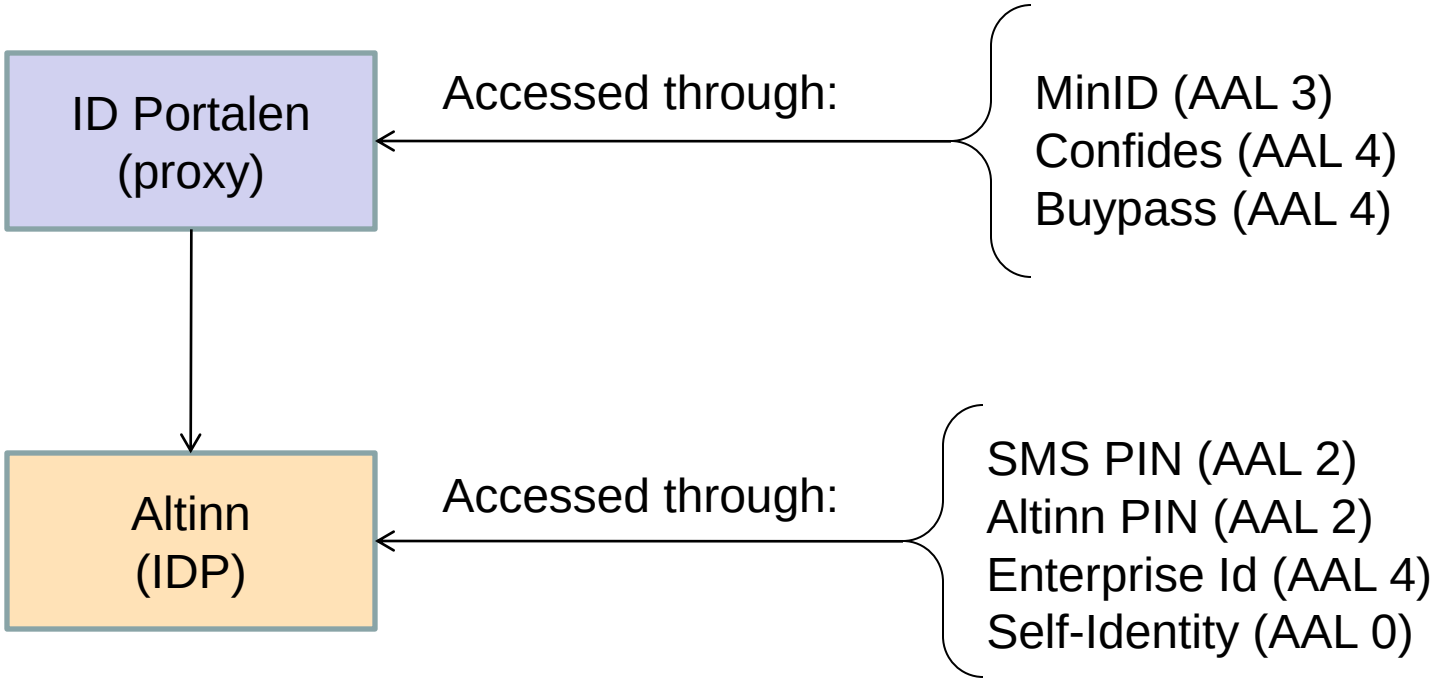
FEIDE Technical Aspects

- Based on SAML 2.0
- Backend authenticate users by using LDAP
- One central identity provider (IdP) where service providers (SPs) are connected
- Single Sign On when going between services
- Single Log Out when logging out from a service

FEIDE Architecture



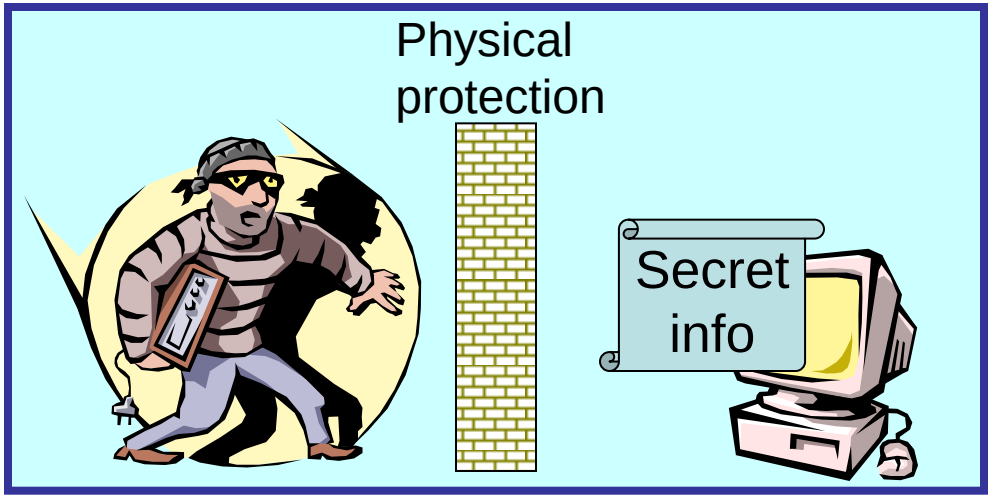
Government Id Management



Introduction to access control

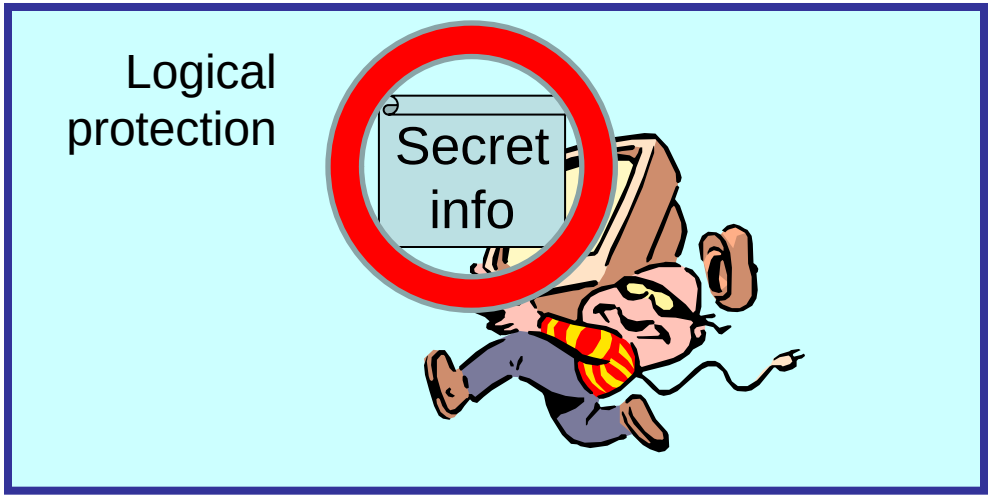
Physical Access Control:

(not the theme today)



Logical Access Control:

(this presentation)



Introduction to access control

- Access Control
  - controls how users and systems access other systems and resources
  - prevents unauthorized users access to resources

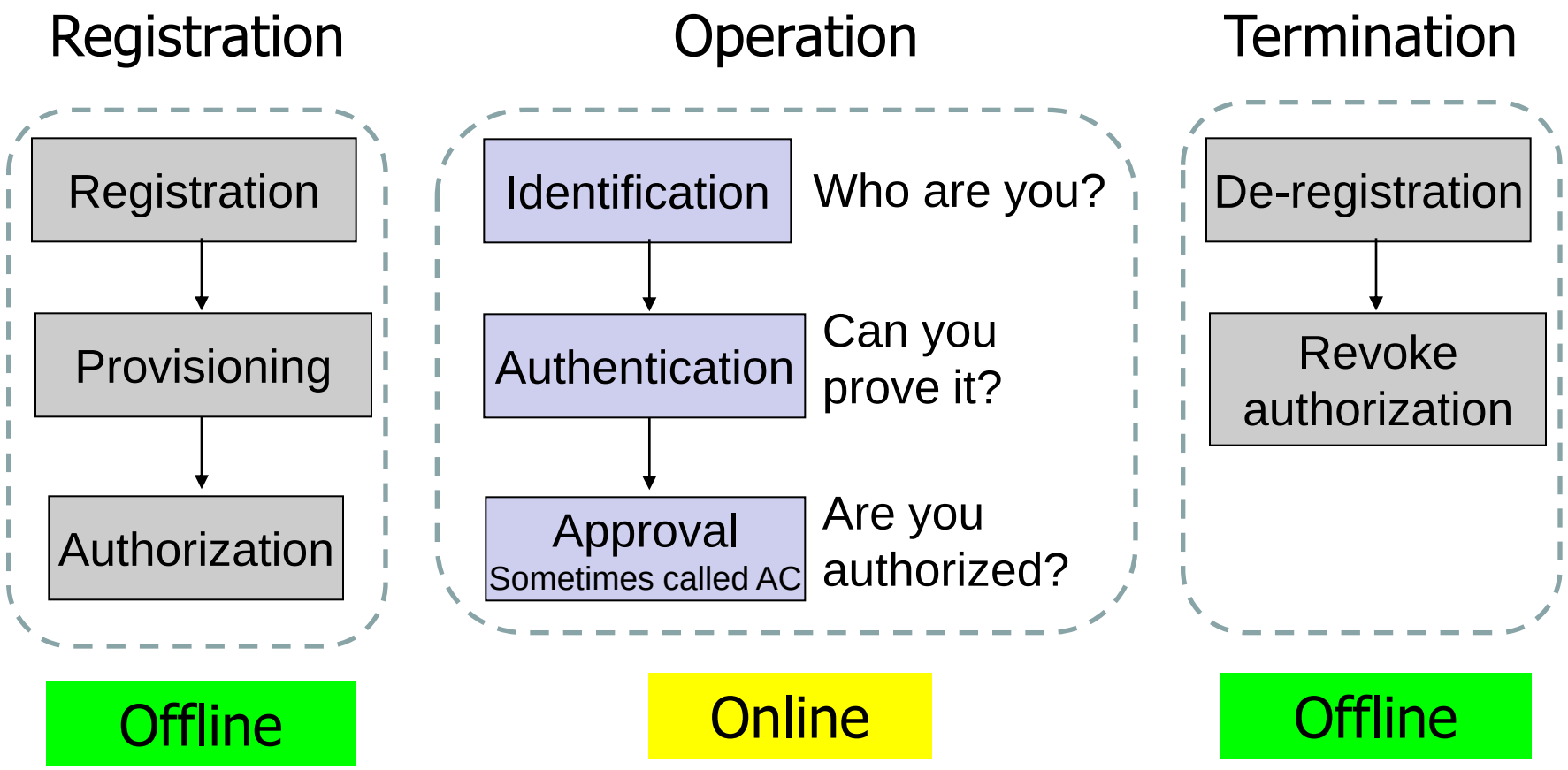
- Unauthorized access could compromise
  - Confidentiality
  - Integrity
  - Availability

  of information assets

Authorization and Access Control

- To authorize is to specify access permissions for roles, individuals, entities or processes
  - Authorization policy normally defined by humans
  - Assumes the existence of an authority
- Authority can be delegated
  - Company Board → Manager → Sys.Admin. → User
  - Delegation can be automated between IT processes
- Authorization policy is Implemented in IT systems in the form of access rules
- Systems approve/reject access based on access rules

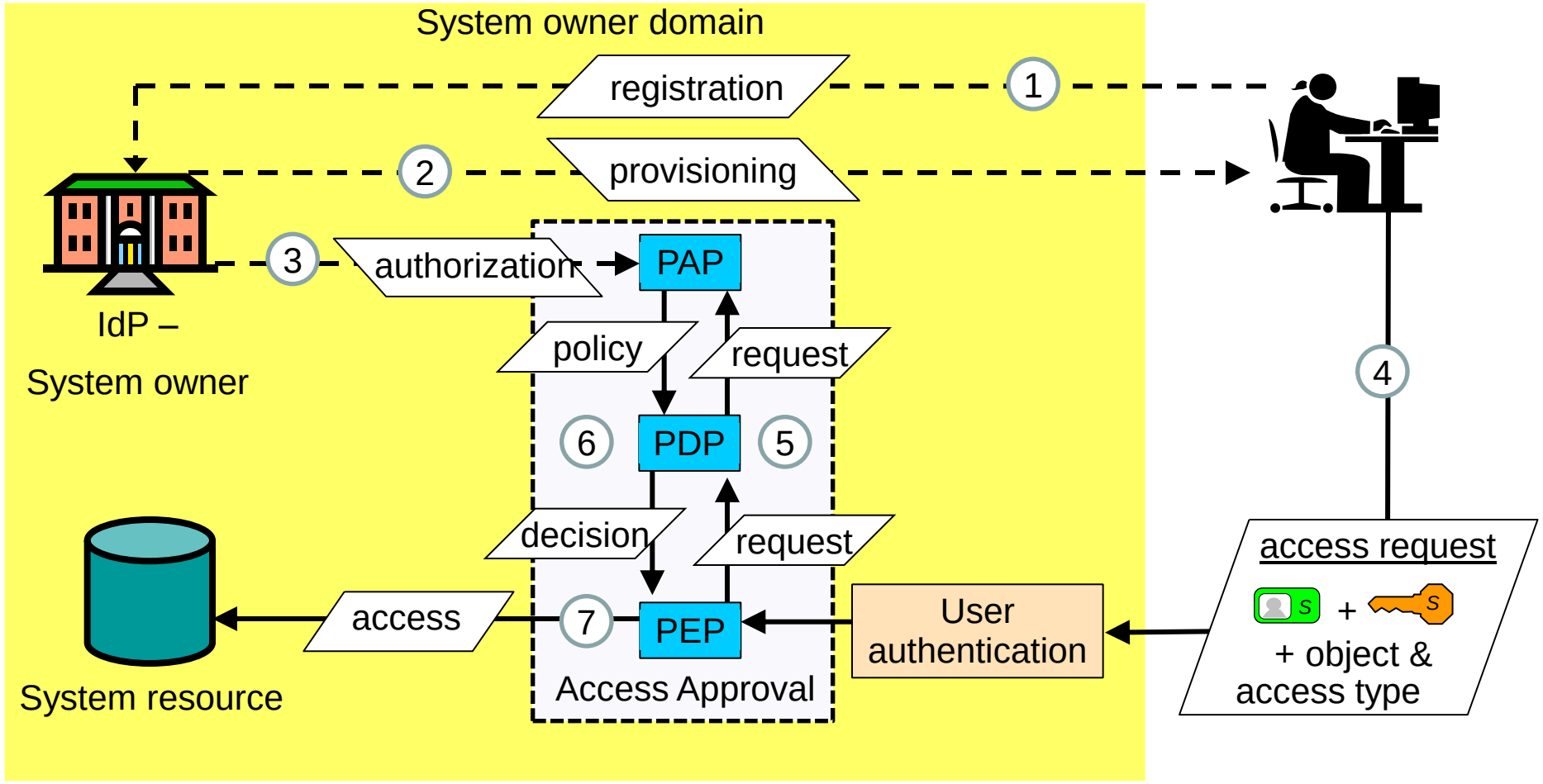
Access Control Phases



Confusion around authorisation

- The term “authorisation” is often wrongly used in the sense of access approval
  - Example Harris
- Wrong use of term “authorisation” causes confusion:
  - Attacker with stolen password would be authorized.
  - Information leaked by attacker with stolen password would not represent breach of confidentiality
  - When accused of illegal access with stolen password, an attacker could claim that he was authorized according to standard text books on IT security
- Beware or both correct and wrong interpretations of “authorisation”

Access control scenario
(http://www.oasis-open.org/specs/index.php)



Access modes

- Modes of access:
  - ***What access permissions (authorizations) should subjects have?***
- If you are authorized to access a resource, what are you allowed to do to the resource?
  - Example: possible access permissions include
    - read - observe
    - write – observe and alter
    - execute – neither observe nor alter
    - append - alter

Basic concepts

- Access control approaches:
  - *How do you define which subjects can access which objects?*
- Three main approaches
  - Discretionary access control (DAC)
  - Mandatory access control (MAC)
  - Role-based access control (RBAC)

Basic concepts: DAC

- Discretionary access control: 2 interpretations:
  1. Access rights to an object or resource are granted at the discretion of the owner of the object
     - e.g. security administrator, the owner of the resource, or the person who created the asset
     - DAC is discretionary in the sense that a subject with a certain access authorization is capable of passing that authorization (directly or indirectly) to any other subject.
  2. According to the Orange Book (TCSEC) DAC is implemented as a an ACL (Access Control List)
- Popular operating systems use DAC
  - Windows and Linux

Basic concepts: ACL

- Access Control Lists (ACL)
  - Attached to an object
  - Provides an access rule for a list of subjects
  - Simple means of enforcing policy
  - Does not scale well
- ACLs can be combined into an access control matrix covering access rules for a set of objects

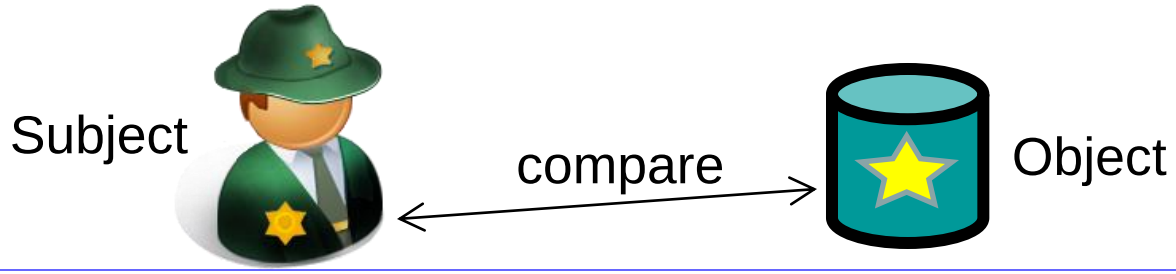
		Objects			
		O1	O2	O3	O4
Subjects	S1	rw	-	x	r
	S2	r	-	r	rw
	S3	-	x	-	-
	S4	rw	x	x	x

Basic concepts: MAC

- Mandatory access control: 2 interpretations:
  1. A central authority assigns access privileges
     - MAC is mandatory in the sense that users do not control access to the resources they create.
  2. According to the Orange Book (TCSEC) MAC is implemented with security labels
     - Example: Security clearance and classification levels
- A system-wide **set of rules** for objects and subjects specify permitted modes of access
  - (SE)Linux includes MAC

Basic concepts: Labels

- Security Labels can be assigned to subjects and objects
  - Represents a specific security level, e.g. “Confidential” or “Secret”
- Object labels are assigned according to sensitivity
- Subject labels are determined by the authorization policy
- Access control decisions are made by comparing the subject label with the object label according to rules
- The set of decision rules is a security model
  - Used e.g. in the Bell-LaPadula and Biba models (see later)



Basic concepts: Combined MAC & DAC

- Combining access control approaches:
  - A combination of mandatory and discretionary access control approaches is often used
    - Mandatory access control is applied first:
    - If access is granted by the mandatory access control rules,
      - then the discretionary system is invoked
    - Access granted only if both approaches permit
  - This combination ensures that
    - no owner can make sensitive information available to unauthorized users, and
    - ‘need to know’ can be applied to limit access that would otherwise be granted under mandatory rules

Basic concepts: RBAC

- Role based access control
  - Access rights are based on the role of the subject, rather than the identity
    - A role is a collection of procedures or jobs that the subject performs
    - Examples: user, administrator, student, etc
    - A subject could have more than one role, and more than one subject could have the same role
  - RBAC can be combined with DAC and MAC

Example security models

- In order to describe an access policy, it is necessary to describe the entities that the access policy applies to and the rules that govern their behaviour.
- A security model provides this type of description.
- Security models have been developed to describe access policies concerned with:
  - Confidentiality (Bell-LaPadula,Clark-Wilson,Brewer-Nash,RBAC)
  - Integrity (Biba, Clark-Wilson, RBAC)
  - Prevent conflict of interest (Brewer-Nash, RBAC)

Bell-LaPadula Model:

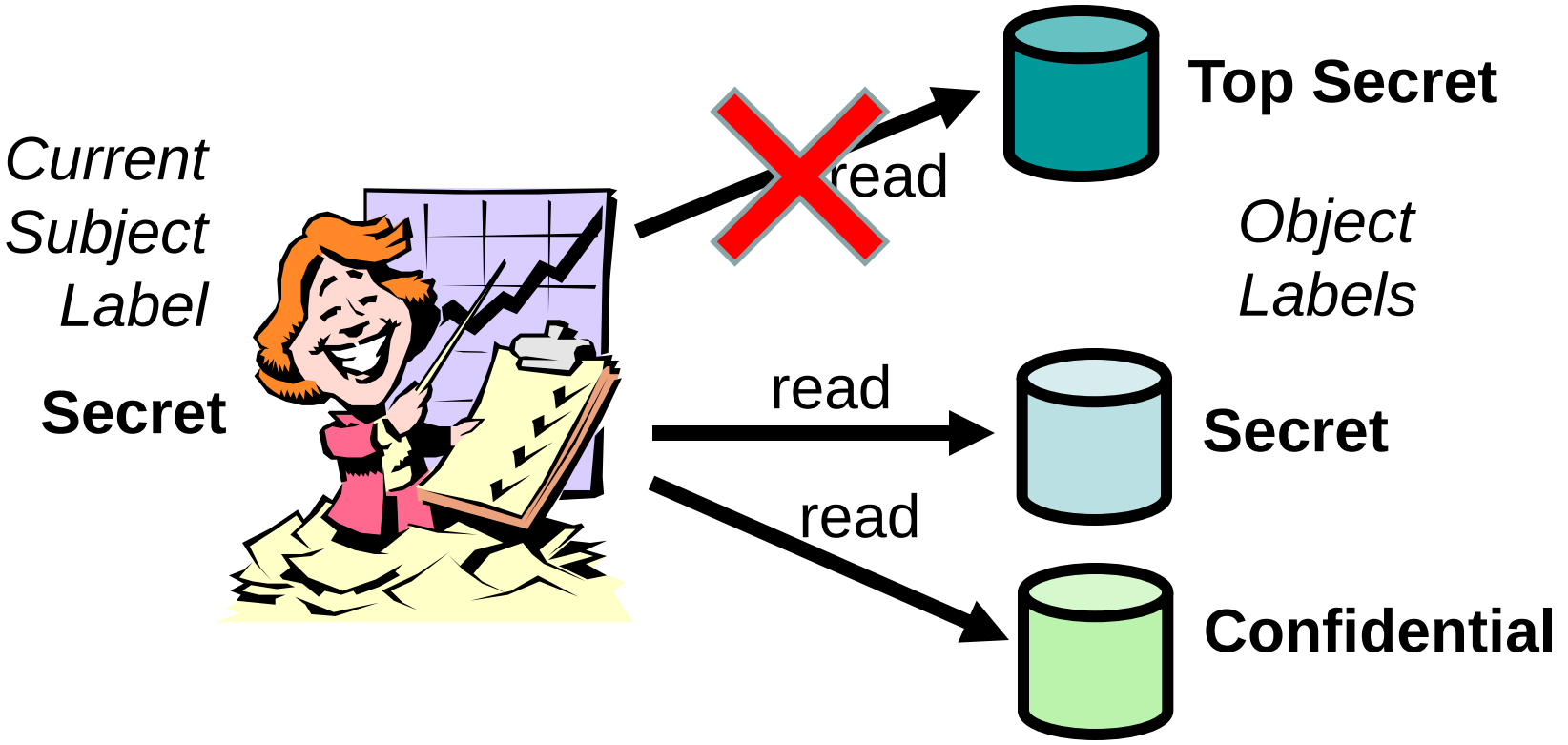
SS-property (Simple Security): No Read Up

- You should not be able to read files with a higher label than your label, because it could cause disclosure of sensitive information.

***-Property (Star Property): No Write Down**

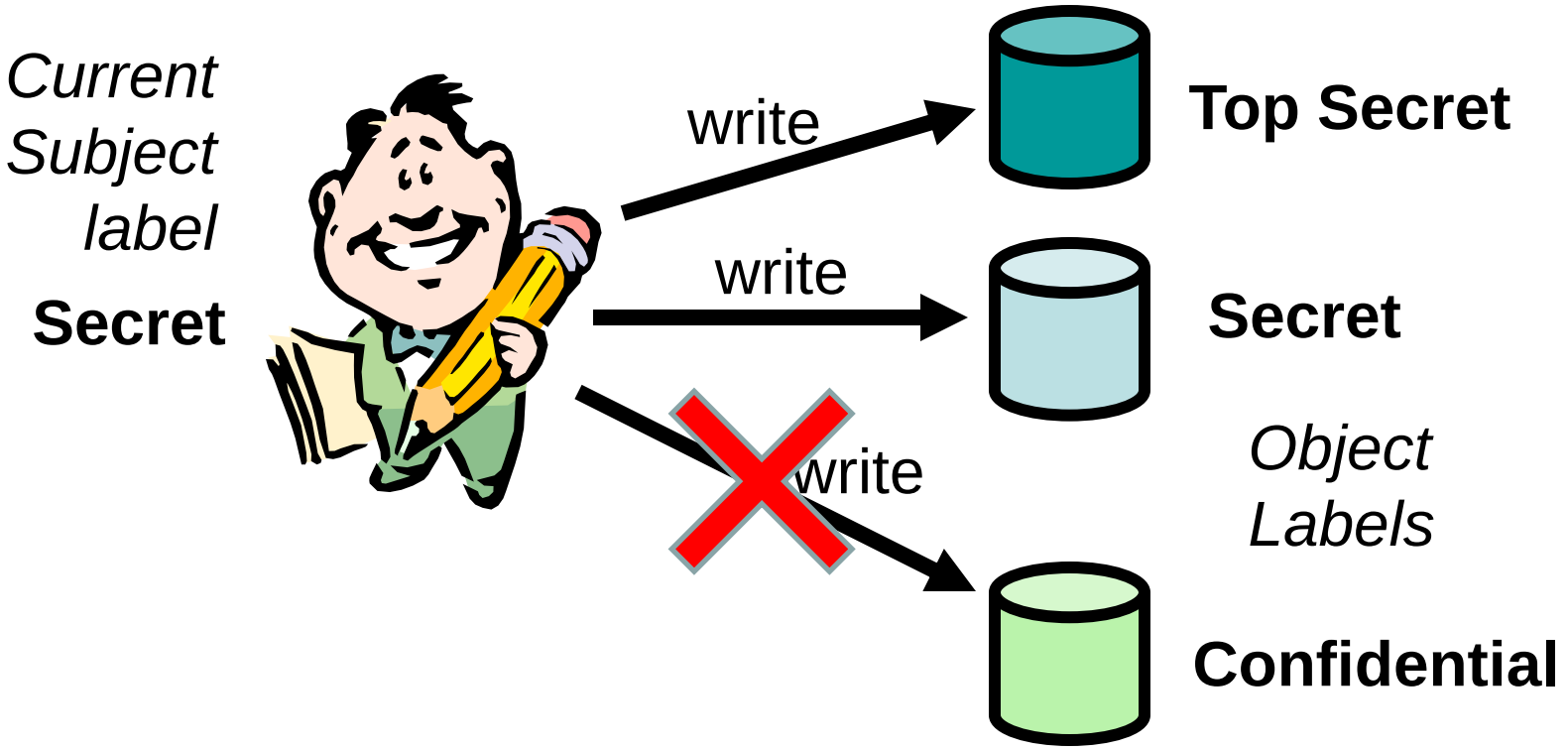
- Subjects working on information/tasks at a given label should not be allowed to write to a lower level because this could leak sensitive information.
- For example, you should only be able write to files with the same label, or higher, as your security clearance level.

Bell-LaPadula (MAC model)
SS-Property: No Read Up



Diagram

Bell-LaPadula (MAC model)
*-Property: No Write Down

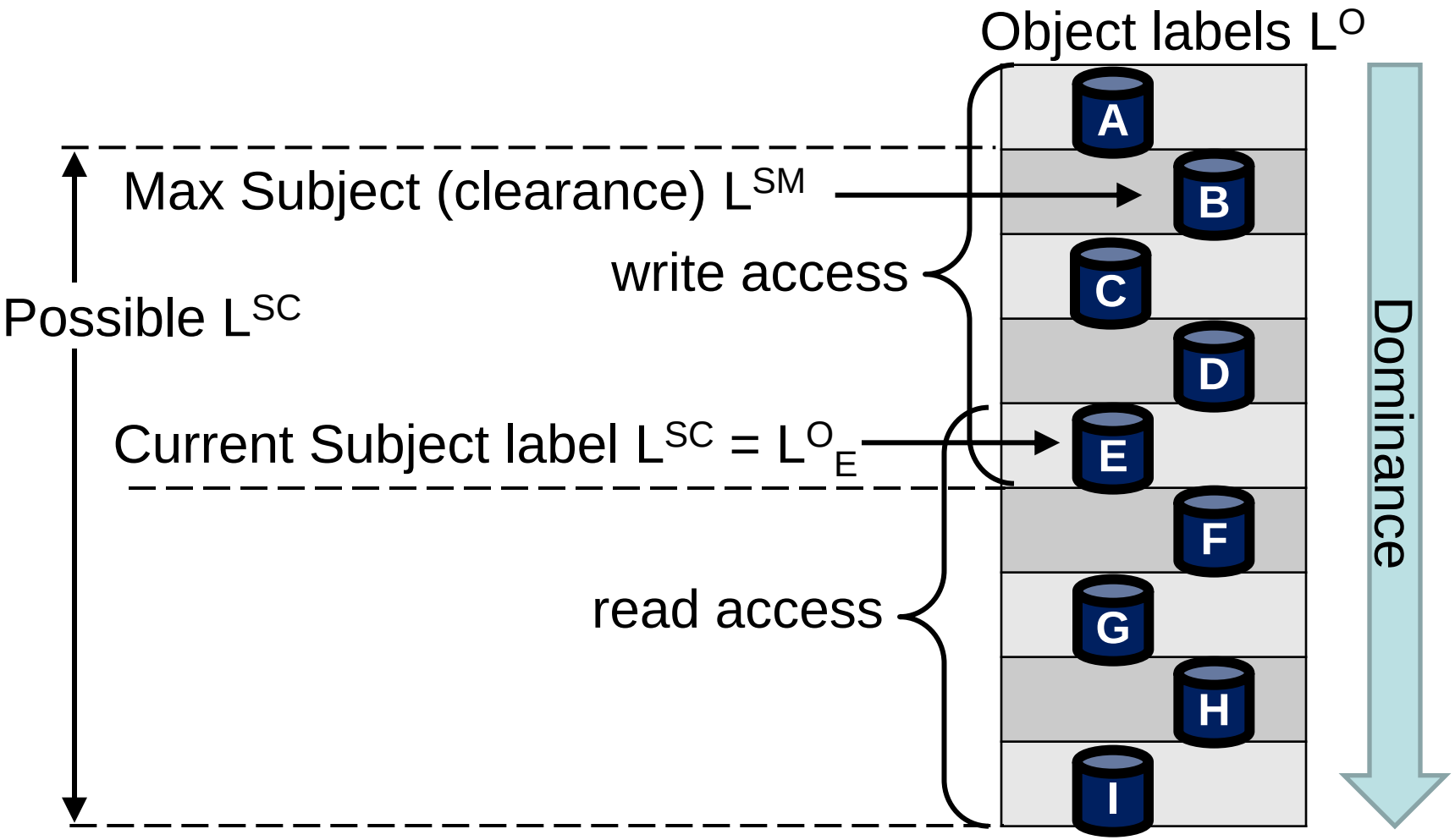


Labels in Bell La Padula

- Users have a clearance level L^{SM} (Subject Max level)
- Users log on with a current clearance level L^{SC} (Subject Current level) where $L^{SC} \leq L^{SM}$
- Objects have a sensitivity level L^{O} (Object)

- SS-property allows read access when $L^{SC} \geq L^{O}$
- *-property allows write access when $L^{SC} \leq L^{O}$

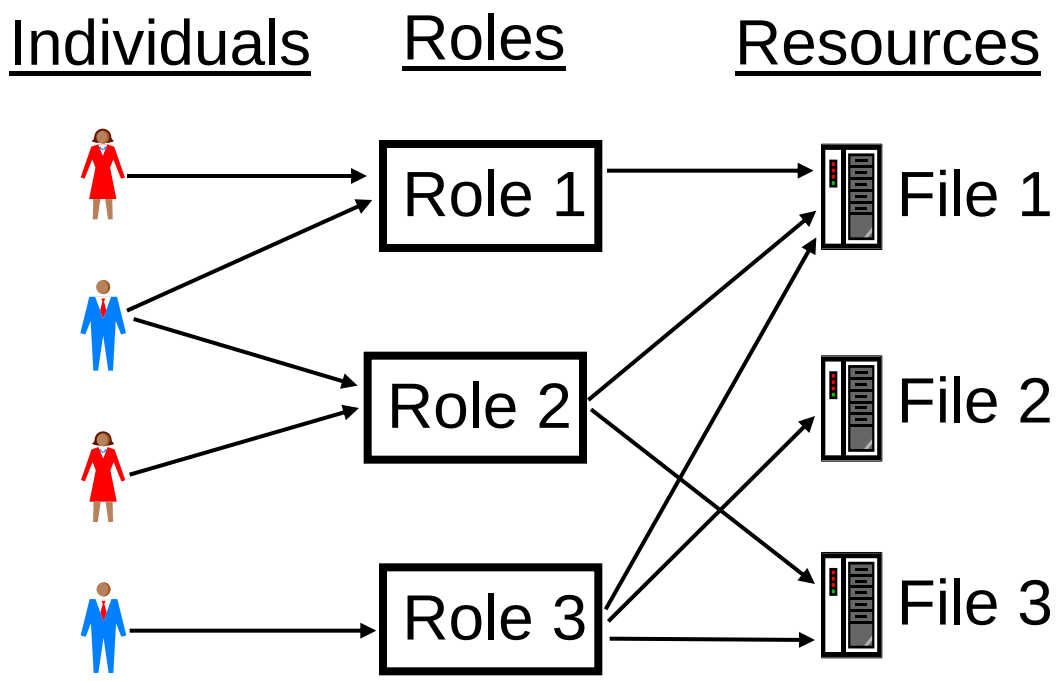
Bell-LaPadula label relationships



RBAC: Role Based Access Control

- A user has access to an object based on the assigned role.
- Roles are defined based on job functions.
- Permissions are defined based on job authority and responsibilities within a job function.
- Operations on an object are invocated based on the permissions.
- The object is concerned with the user's role and not the user.

RBAC Flexibility



User's change frequently, roles don't

- RBAC can be configured to do MAC and/or DAC

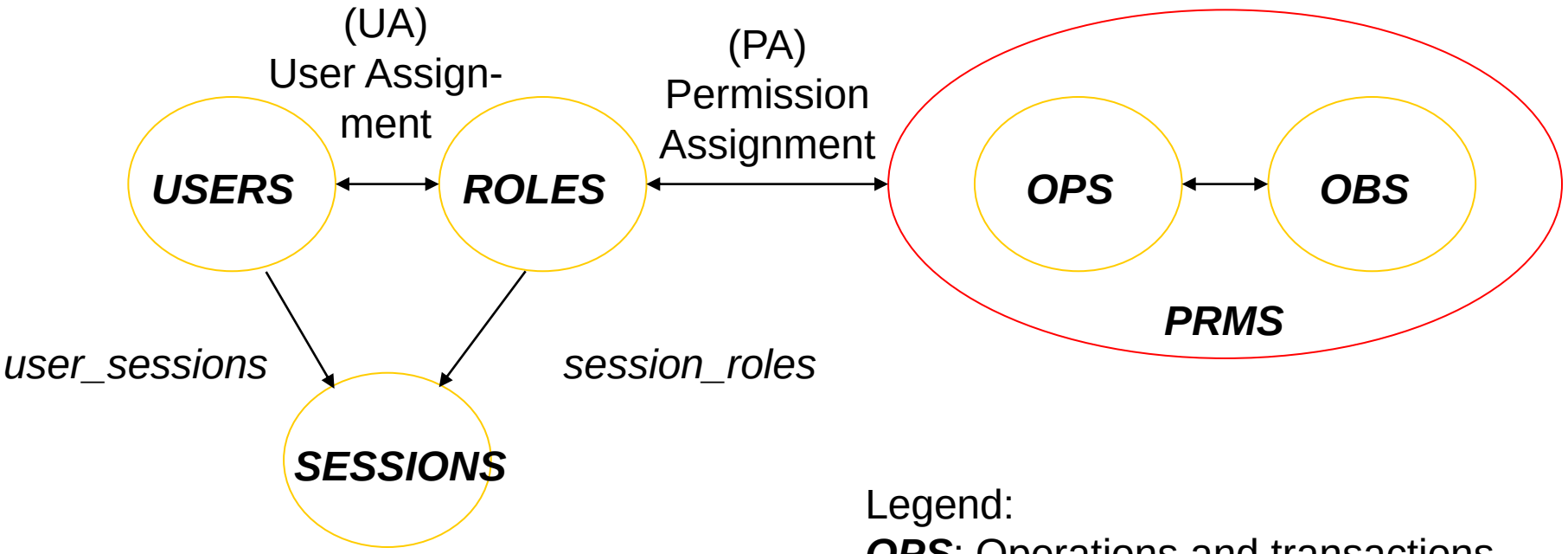
RBAC Privilege Principles

- Roles are engineered based on the principle of least privilege .
- A role contains the minimum amount of permissions to instantiate an object.
- A user is assigned to a role that allows her to perform only what's required for that role.
- All users with the same role have the same permissions.

RBAC Core Components

- Defines:
  - USERS
  - ROLES
  - OPERATIONS (*ops*)
  - OBJECTS (*obs*)
  - User Assignments (*ua*)
    - assigned_users
  - Permissions (*prms*)
    - Assigned Permissions
    - Object Permissions
    - Operation Permissions
  - Sessions
    - User Sessions
    - Available Session Permissions
    - Session Roles

Core RBAC



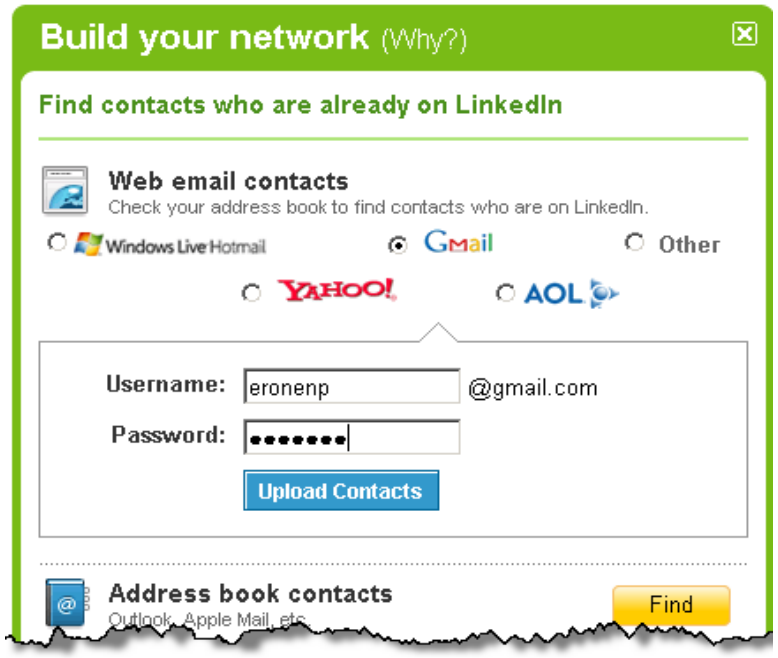
Legend:
OPS: Operations and transactions
OBS: Objects, databases, files
PRMS: Permissions

Web Access Delegation with OAuth

- OAuth: Open Authorization

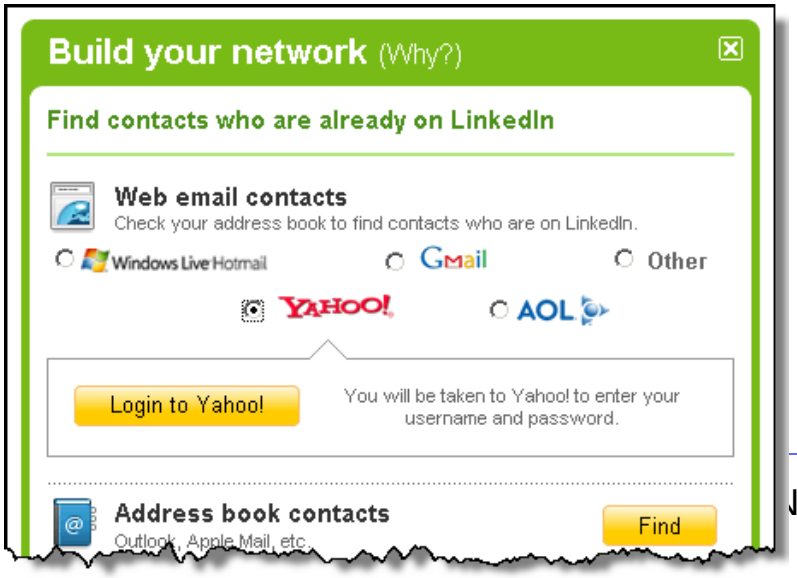
- *OAuth provides a way to grant access to your user data stored on specific website to a third party website, without needing to provide this third party website with your authentication credentials for accessing website.*

User authorizes access to own account



Without Oauth.
Password for user account on data resource website revealed to 3rd party Web application

BAD

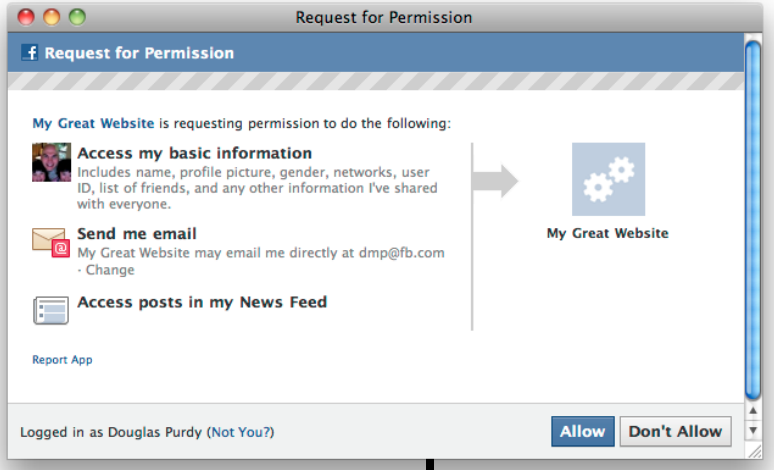


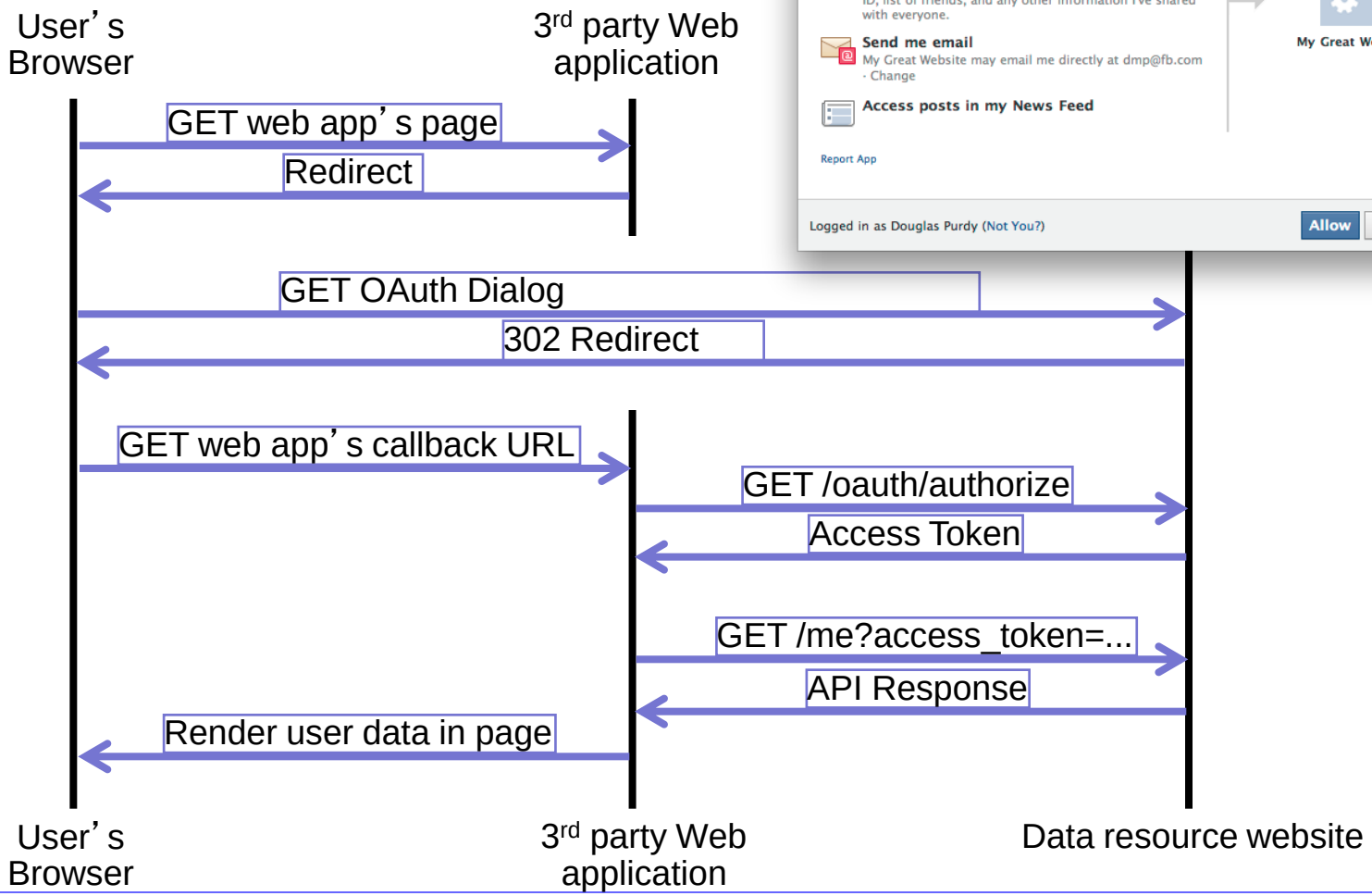
With Oauth.
No password sent to 3rd party Web application.

GOOD

- Problematic to reveal password of user account on website (e.g. Gmail) to 3rd party Web application (e.g. LinkedIn), because Web application could take control over user account on that website.
- OAuth provides a way to authorize 3rd party Web application to get limited access to user account on user's website.
- OAuth is used extensively in Web 2.0

OAuth Message Flow





OAuth remarks

- Open Web Authorization (OAuth) is developed within the IETF to provide delegated access authorization between Web-based applications.
  - Usage for non-Web based applications has been proposed as well.
- Work is in progress and re-chartering will expand the work to include new features and use cases as well as security.
- OAuth is a relatively recent technology which is rapidly evolving, and is therefore not well studied from a security perspective.

End of lecture