



Identity management and access control: State of art and research perspectives

Audun Jøsang, Ijlal Loutfi
University of Oslo
IFI Department of Informatics

FRISC Winter School, Finse
May, 9th 2015

Agenda

- Why is Identity Management important ?
- Identity management phases.
- Authentication mechanisms.
- Identity Federations.
- Fido.
- Access Control.
- Research perspectives and conclusions.

Why is Identity Management important?

What has changed?



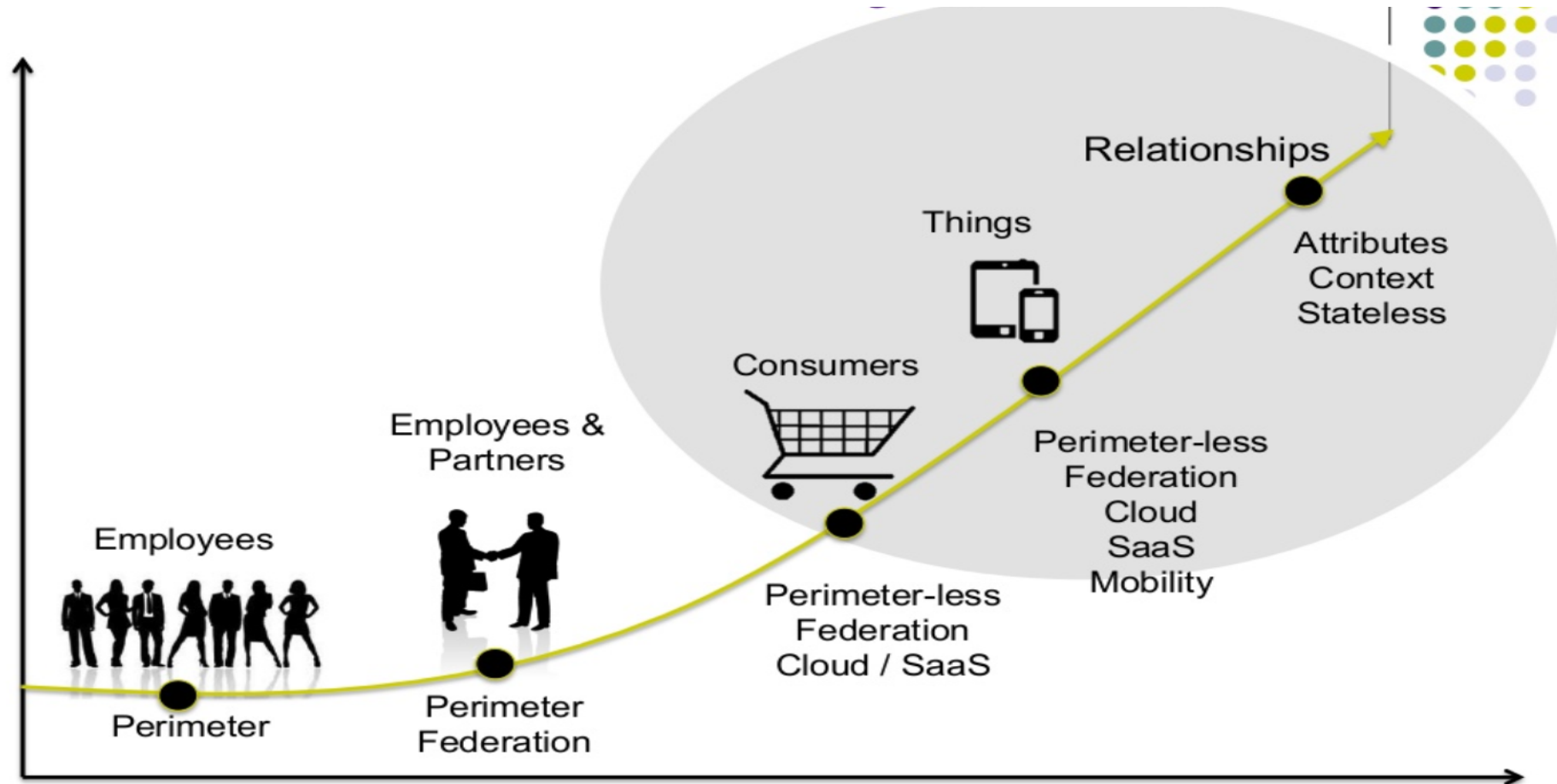
Online managed health services



Online education



How did we get there?

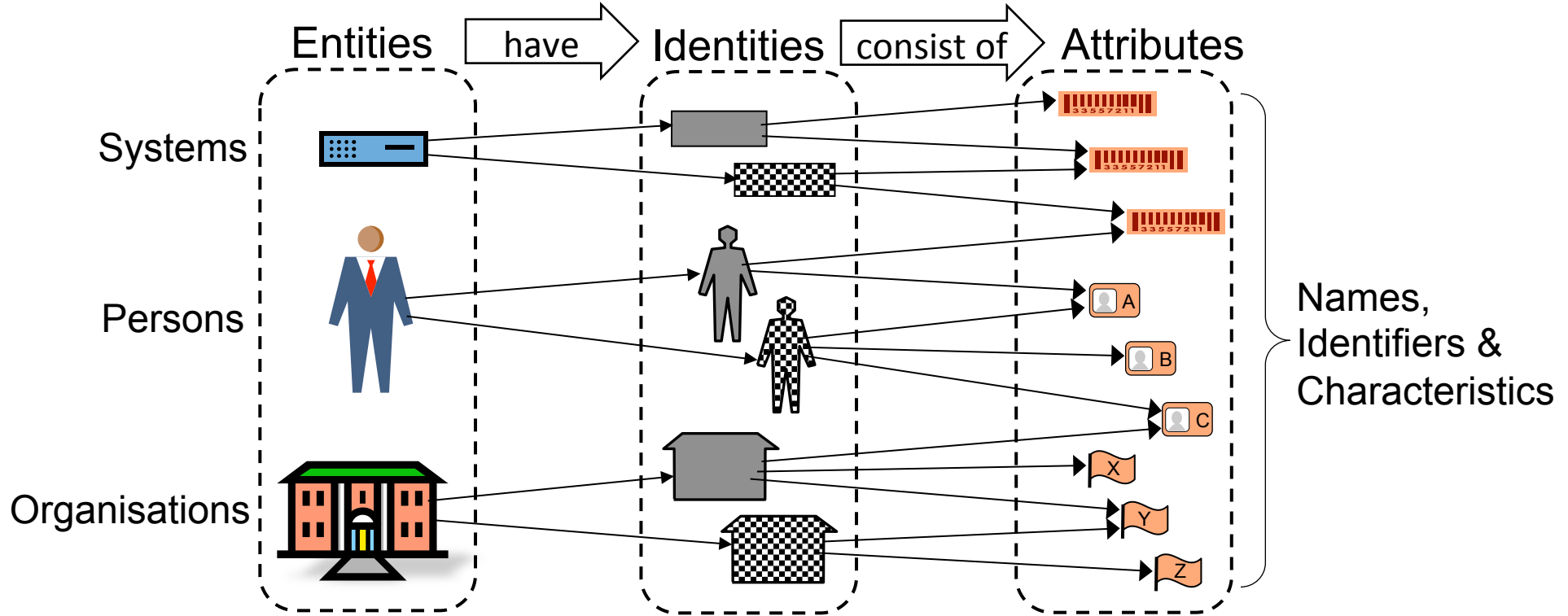


How do we consume ALL of these services?

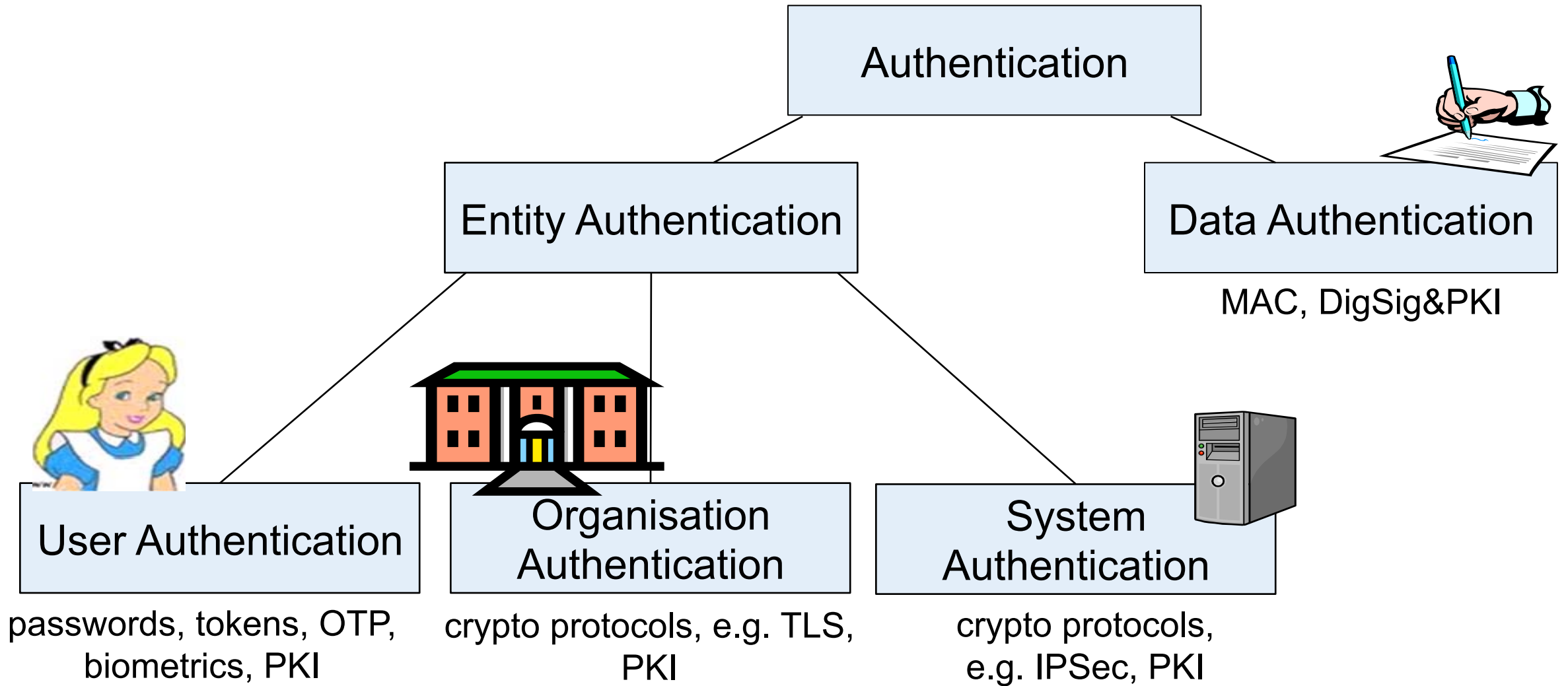
- Devices?
- Connectivity?
- Cognitive power?

→ **IDENTITY**



The concept of identity



Taxonomy of Authentication



Identity management processes

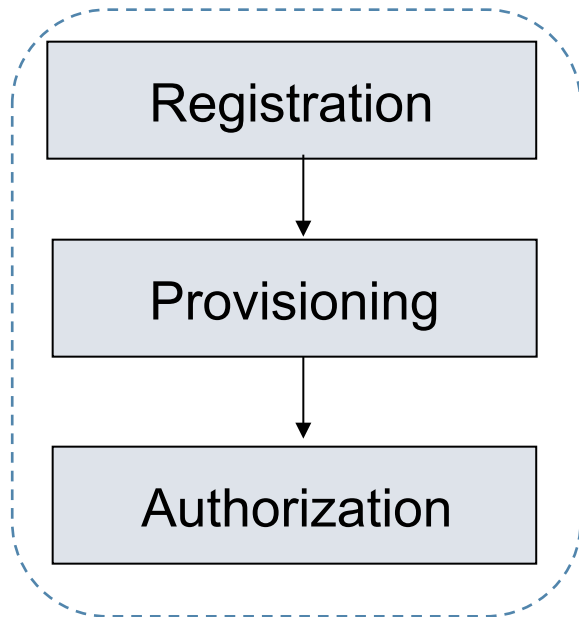
	User Side	Service Provider Side
User Identity Management 	IdMan processes for user Ids & credentials on user side	IdMan processes for user Ids & credentials on SP side
SP Identity Management 	IdMan processes for SP Ids & credentials on user side	IdMan processes for SP Ids & credentials on SP side

- This lecture focuses on user identities, not SP identities

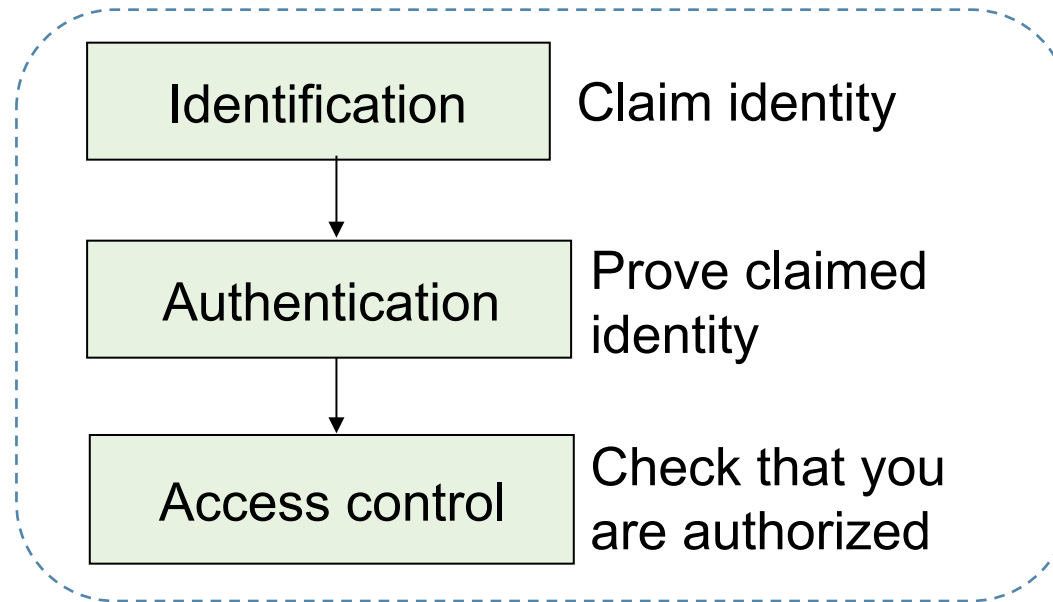
How do we manage user identities
throughout their lifecycle?

Identity and Access Management (IAM) Phases

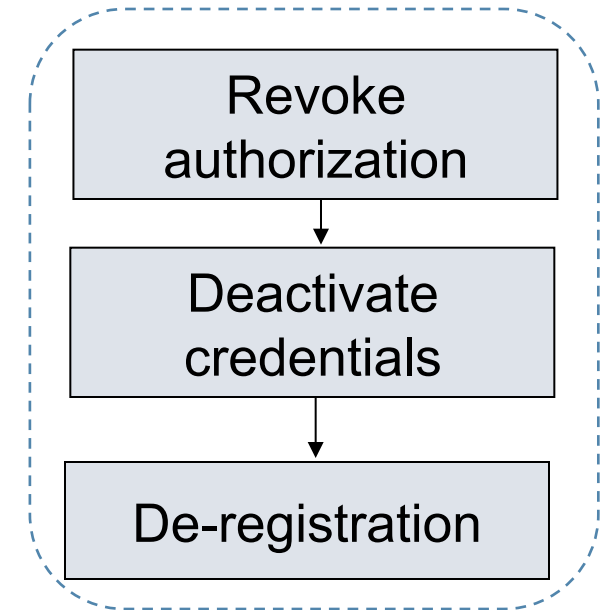
Configuration phase



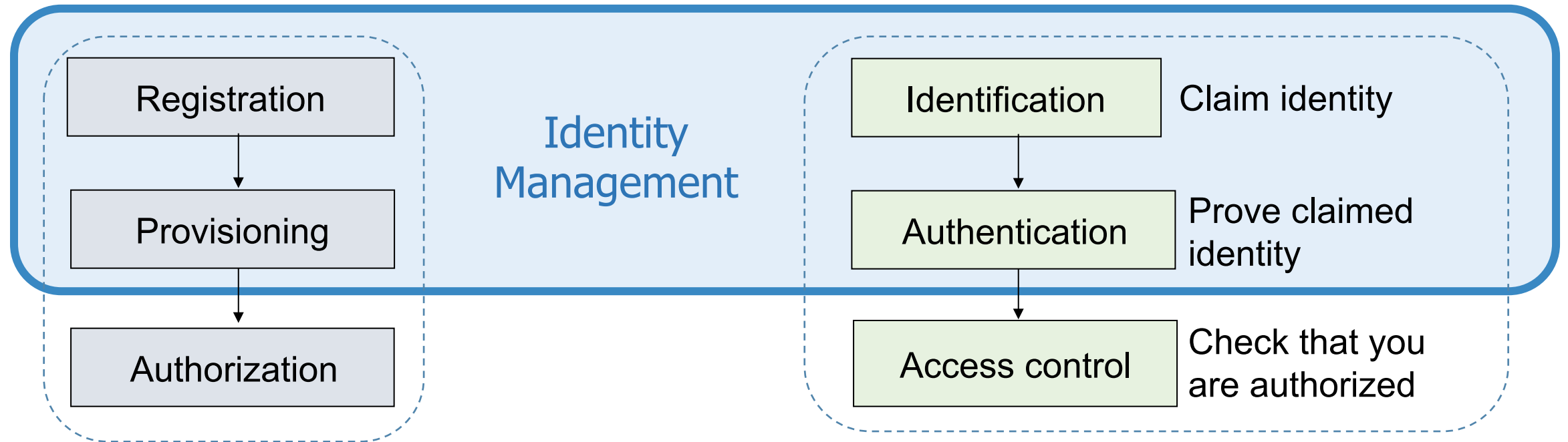
Operation phase



Termination phase



What is Identity Management ?



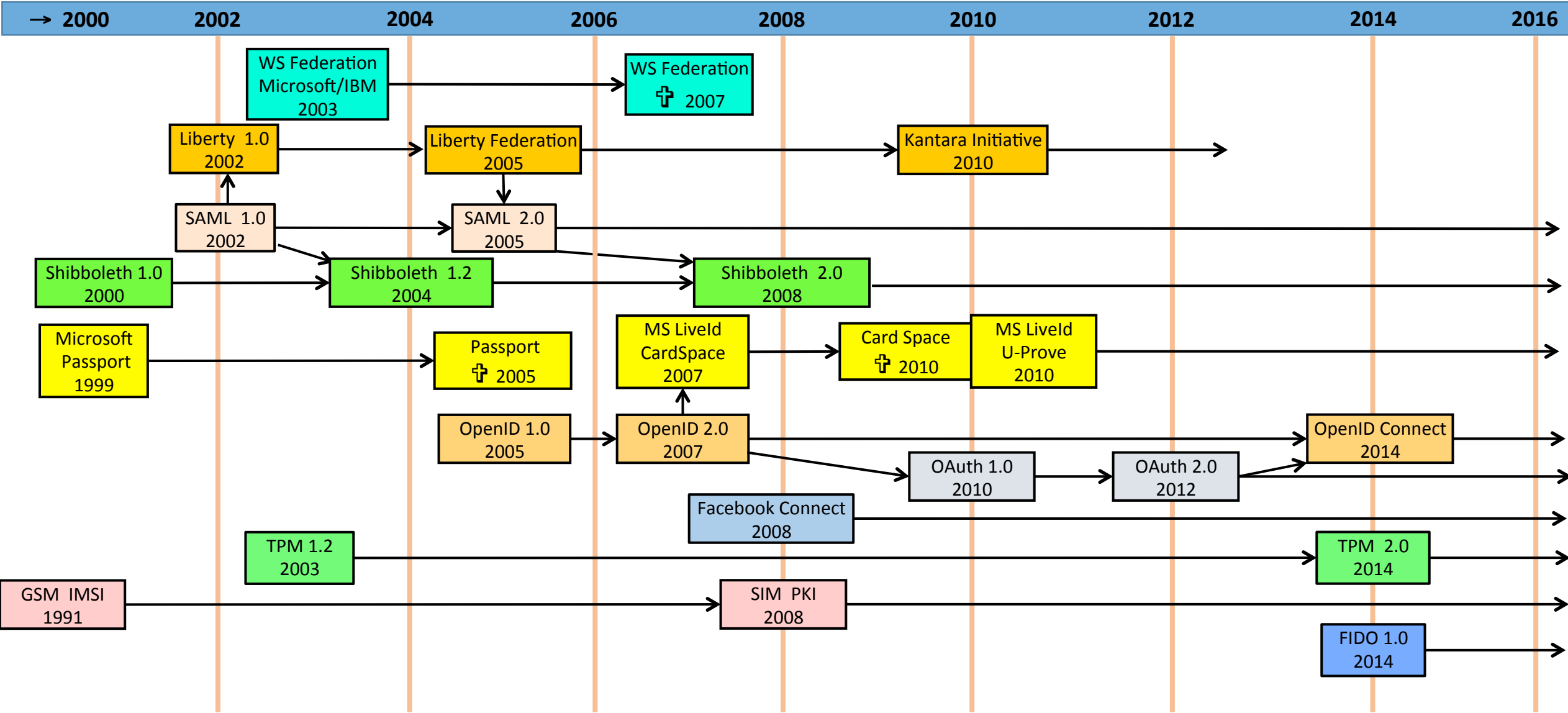
User Identification and Authentication

- Identification
 - Who you claim to be
 - Method: (user)name, biometrics
- User authentication
 - Prove that you are the one you claim to be
- Main threat: Unauthorized access
- Controls:
 - Passwords,
 - Personal cryptographic tokens,
 - OTP generators, etc.
 - Biometrics
 - Id cards
 - Cryptographic security/authentication protocols



Authentication token

Evolution of ID Management Initiatives



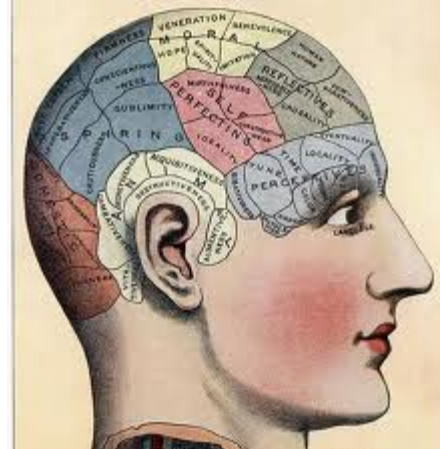
User authentication mechanisms

User authentication credentials

- A credential is the ‘thing’ used for authentication.
 - May also be referred to as a “token” or “authenticator”
 - e.g. reusable passwords, PIN, biometrics, smart cards, certificates, cryptographic keys, OTP hardware tokens.
- Credential categories:
 1. Knowledge-Based (Something you know): Passwords
 2. Ownership-Based (Something you have): Tokens
 3. Inherence-Based (Something you are/do): Biometrics
 - physiological biometric characteristics
 - behavioural biometric characteristics
- Combinations, called multi-factor authentication

Knowledge-Based Authentication

Something you know: Passwords



Authentication:

Reusable passwords

- Passwords are a simple and most-often-used authenticator.
 - Something the user knows
- Problems:
 - Easy to share (intentionally or not)
 - Easy to forget
 - Often easy to guess
 - Can be written down (both good and bad)
 - If written down, then “what you know” is “where to find it”

RockYou Hack

- 32 million cleartext passwords stolen from RockYou database in 2009
- SnapChat leaked pictures
- Posted on the Internet
- Contains accounts and passwords for websites
 - MySpace, Yahoo, Hotmail
- Analyzed by Imperva.com
 - 1% use 123456
 - 20% use password from set of 5000 different passwords

MOST POPULAR PASSWORDS

Nearly one million RockYou users chose these passwords to protect their accounts.

- | | |
|---------------------|----------------------|
| 1. 123456 | 17. michael |
| 2. 12345 | 18. ashley |
| 3. 123456789 | 19. 654321 |
| 4. password | 20. qwerty |
| 5. iloveyou | 21. iloveu |
| 6. princess | 22. michelle |
| 7. rockyou | 23. 111111 |
| 8. 1234567 | 24. 0 |
| 9. 12345678 | 25. tigger |
| 10. abc123 | 26. password1 |
| 11. nicole | 27. sunshine |
| 12. daniel | 28. chocolate |
| 13. babygirl | 29. anthony |
| 14. monkey | 30. angel |
| 15. jessica | 31. FRIENDS |
| 16. lovely | 32. soccer |

Secure password strategies

- Passwords length ≥ 13 characters
- Use ≥ 3 categories of characters
 - L-case, U-case, numbers, special characters
- Do not use ordinary words (names, dictionary wds.)
- Change typically every 3 – 13 months
- Reuse only between low-sensitivity accounts
- Store passwords securely
 - In brain memory
 - On paper
 - In cleartext on offline digital device
 - Encrypted on online digital device

Strategies for strong passwords

- User education and policies
 - Not necessarily with strict enforcement
- Proactive password checking
 - User selects a potential password which is tested
 - Weak passwords are not accepted
- Reactive password checking
 - SysAdmin periodically runs password cracking tool (also used by attackers) to detect weak passwords that must be replaced.
- Computer-generated passwords
 - Random passwords are strong but difficult to remember
 - FIPS PUB 181 <http://www.itl.nist.gov/fipspubs/fip181.htm> specifies automated pronounceable password generator

Password Caching

- Problem: the password is stored on medium
 - Buffers, caches, web pages
 - Outside user's control
- If you leave the browser open on a public machine, the next user can obtain information about you.

Password storage in OS

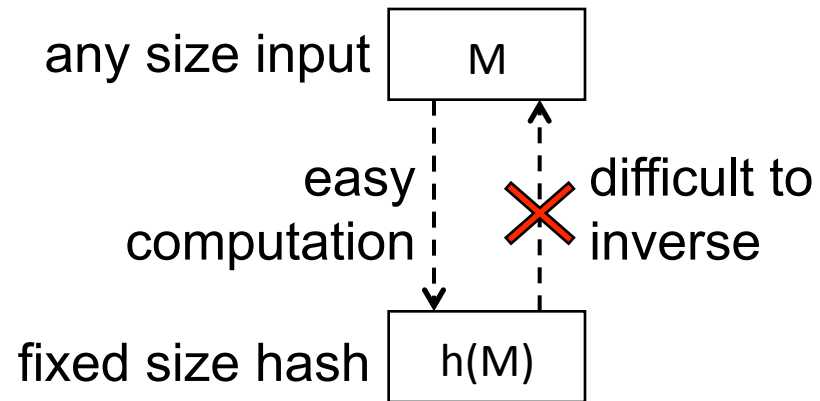
- /etc/shadow is the file where modern Linux/Unix stores its passwords
 - Earlier version stored it in /etc/passwd
 - Need root access to modify it
- \windows\system32\config\sam is the file Windows systems normally store its passwords
 - Undocumented binary format

Prevent exposure of password file

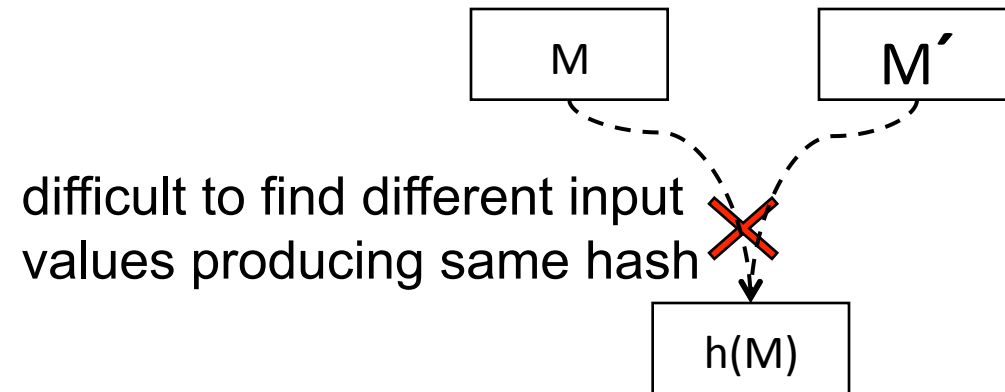
- The computer verifies user passwords against stored values in the password file
- Password file must be available to OS
 - This file need protection from users and applications
 - Avoid offline dictionary attacks
- Protection measures
 - Access control (only accessible by OS kernel)
 - Hashing or Encryption
- In case a password file gets stolen, then hashing/ encryption can provide protection.

Hash functions

One-way function



Collision free



- *A hash function is easy to compute but hard to invert.*
- Passwords can be stored as hash values.
- Authentication function first computes hash of received password, then compares against stored hash value

Cracking passwords

- Bruce Force
 - Trying all possible combinations
- Intelligent search
 - User name
 - Name of friends/relatives
 - Phone number
 - Birth dates
 - Dictionary attack
 - Try all words from an dictionary
 - Precomputed hashes: Rainbow tables

Hash table and rainbow table attacks

- Attackers can compute and store hash values for all possible passwords up to a certain size
- A list of password hashes is a **hash table**
- A compressed hash table is a **rainbow table**
- Comparing and finding matches between hashed passwords and hash/rainbow table is used to determine cleartext passwords.



Password salting: Defence against password cracking



- Prepend or append random data (salt) to a user's password before hashing
 - In Unix: a randomly chosen integer from 0 to 4095.
 - Different salt for each user
 - Produces different hashes for equal passwords
 - Prevents that users with identical passwords get the same password hash value
 - Increases the amount of work required for hash table attacks and rainbow table attacks

Methods of storing passwords on server

- Cleartext password (low security)
 - Password: 123456,
 - Stored on server: 123456
- Hashed password (moderately security)
 - Password: 123456
 - Stored on server: e.g. SHA1-hash of password:
7c4a8d09ca3762af61e59520943dc26494f8941b
- Salted password (good security)
 - Password: 123456
 - Stored on server: Salt + Salted hash
e.g. “salt”: f8b97abc30b72e54
eg. SHA1-hash of password + salt
1736f11fae29189749a8a54f45e25fb693c3959d



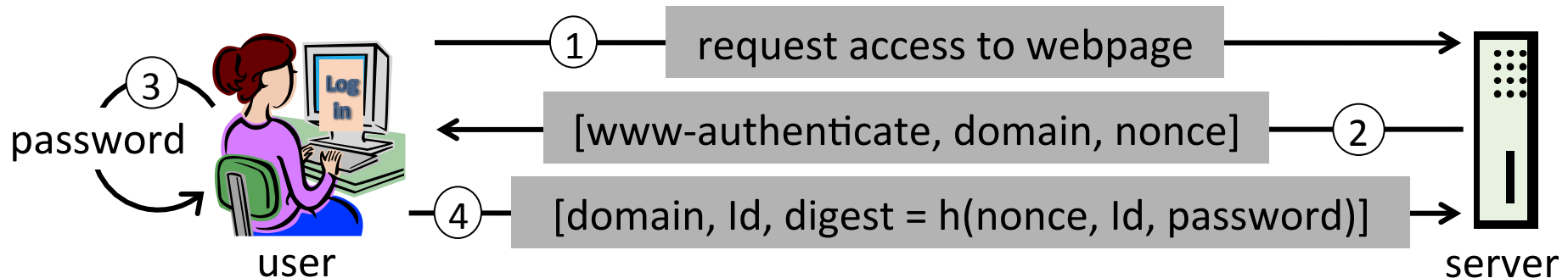
Problems with using passwords in the clear

- A password sent “in clear” can be captured during transmission, so an attacker may reuse it.
- An attacker setting up a fake server can get the password from the user
 - E.g. phishing attack.
- Solutions to these problems include:
 - Encrypted communication channel
 - One-time passwords (token-based authentication)
 - Challenge-response protocols

HTTP Digest Authentication

A simple challenge-response protocol

- A simple challenge response protocol specified in RFC 2069
- Server sends:
 - WWW-Authenticate = Digest
 - realm="service domain"
 - nonce="some random number"
- User types Id and password in browser window
- Browser produces a password digest from nonce, Id and password using a 1-way hash function (SHA-1....)
- Browser sends Id and digest to server that validates digest

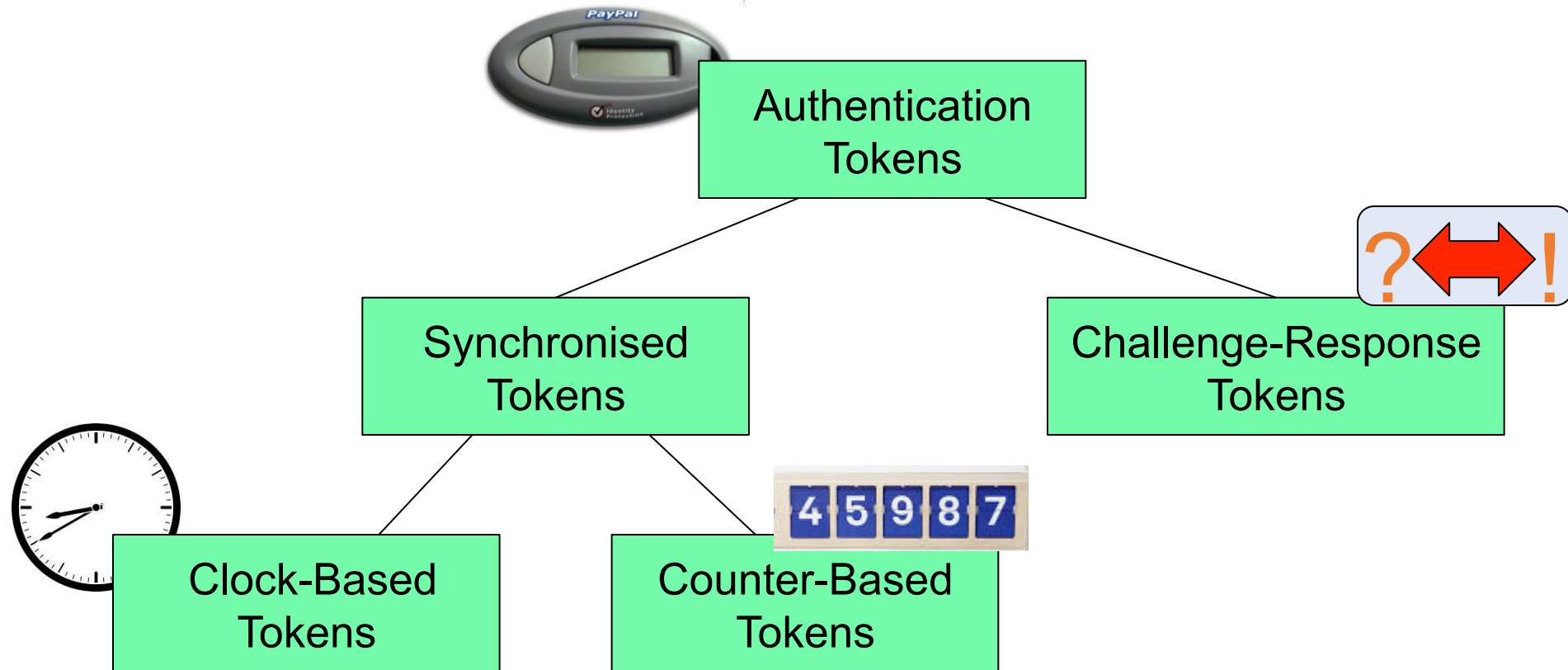


Ownership-Based Authentication

Something you have: Tokens



Taxonomy of Authentication Tokens



Synchronised OTP (One-Time-Password) Generator

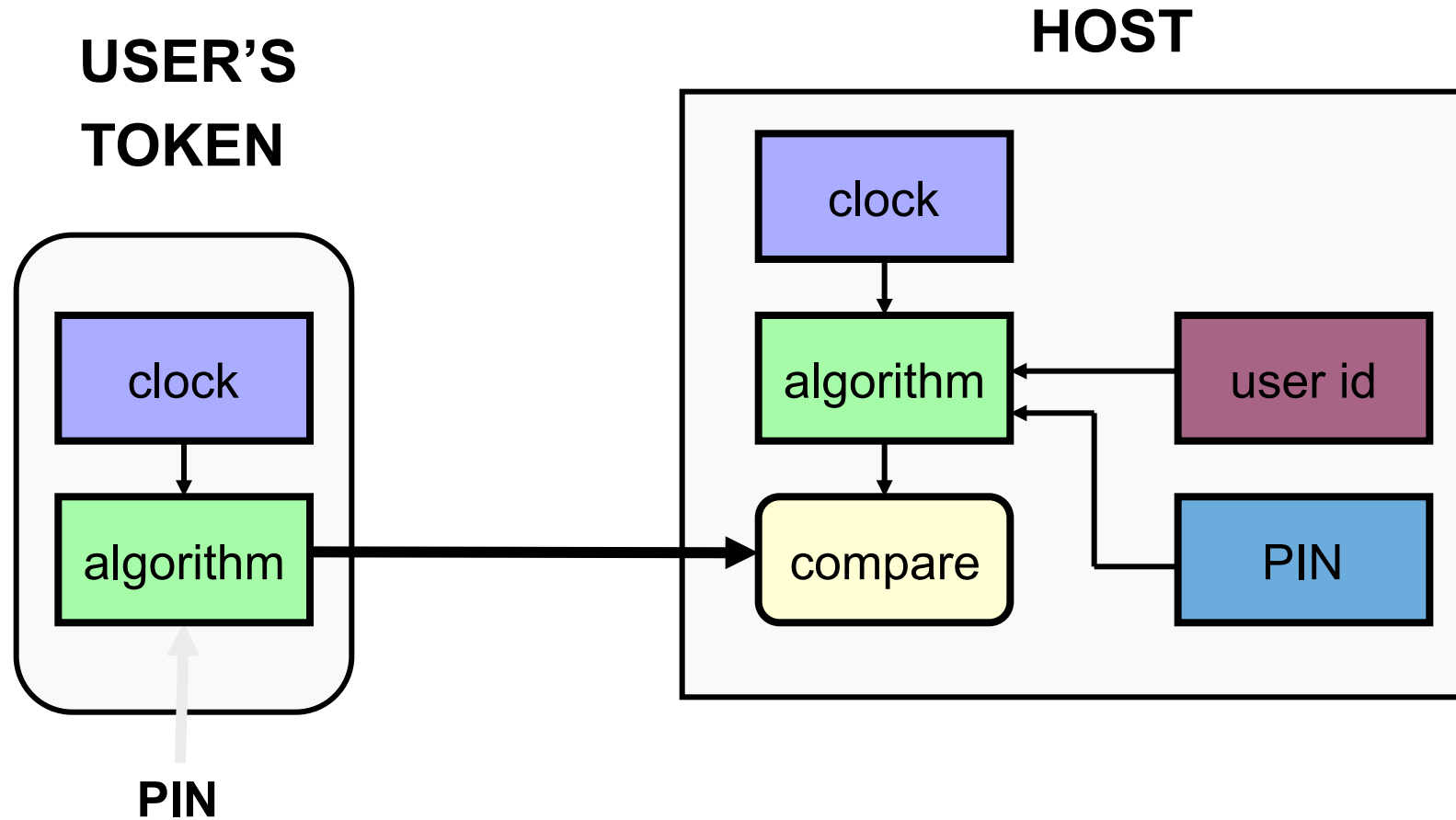
- Using a password only once significantly strengthens the strength of user authentication.
- Synchronized password generators produce the same sequence of random passwords both in the token and at the host system.
 - OTP is 'something you have' because generated by token
- There are two general methods:
 - Clock-based tokens
 - Counter-based tokens



Clock-based OTP Tokens: Operation

- Token displays time-dependent code on display
 - User copies code from token to terminal to log in
- Possession of the token is necessary to know the correct value for the current time
- Each code computed for specific time window
- Codes from adjacent time windows are accepted
- Clocks must be synchronised
- Example: BankID and SecurID

Clock-based OTP Tokens: Operation



Clock-based OTP Tokens: RSA SecurID tokens and BankID tokens



RSA SecurID SD600



RSA SecurID SID700



RSA SecurID SD200



BlackBerry with
RSA SecurID software token



BankID OTP
calculator with PIN



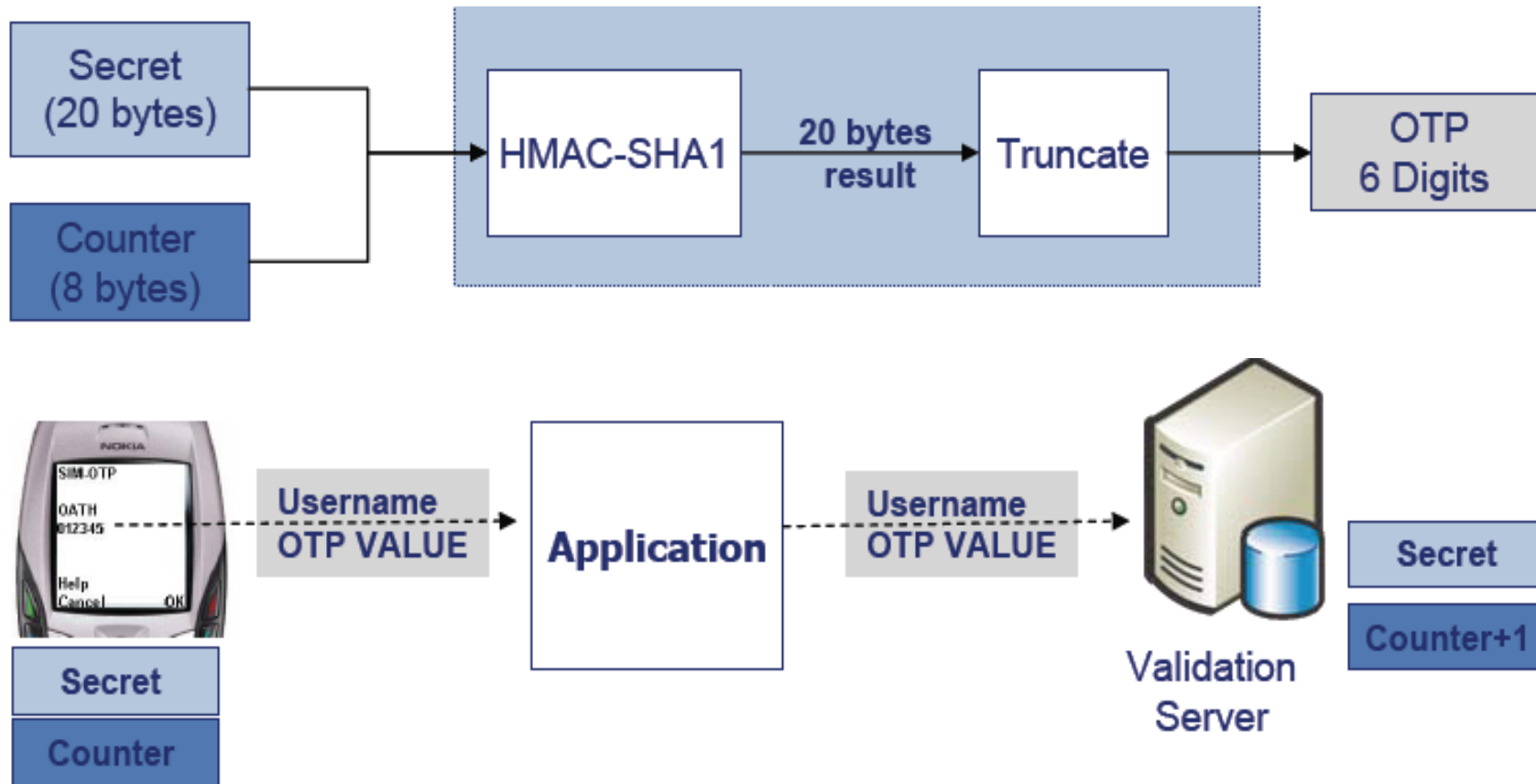
BankID OTP
calculator without PIN

Counter-based OTP Tokens: Overview

- Counter-based tokens generate a 'password' result value as a function of an internal counter and other internal data, without external inputs.
- HOTP is a HMAC-Based One-Time Password Algorithm described in RFC 4226 (Dec 2005) <http://www.rfc-archive.org/getrfc.php?rfc=4226>
 - Tokens that do not support any numeric input
 - The value displayed on the token is designed to be easily read and entered by the user.



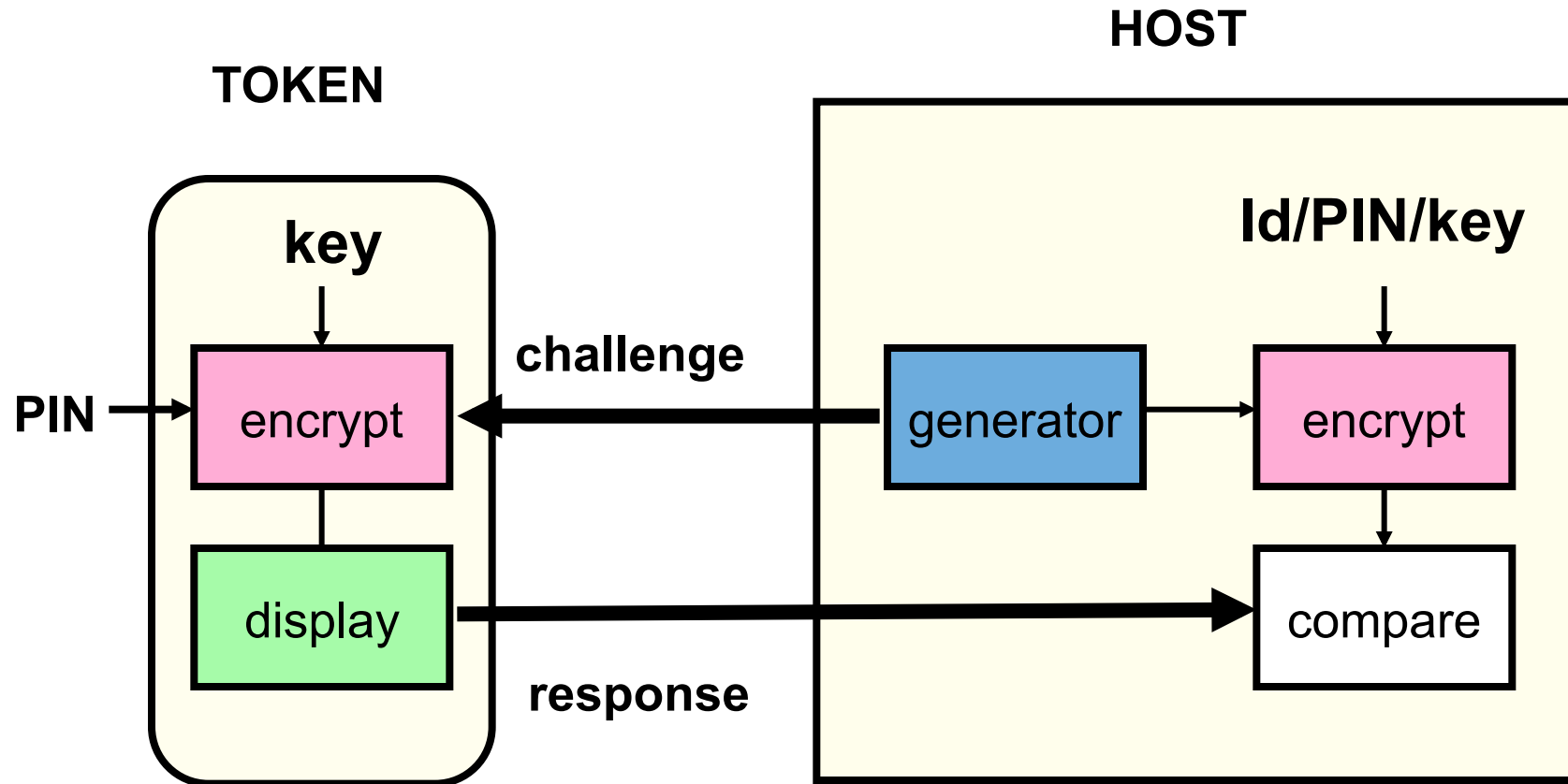
Counter-based OTP Tokens: HOTP



Token-based User Authentication: Challenge Response Systems

- A challenge is sent in response to access request
 - A legitimate user can respond to the challenge by performing a task which requires use of information only available to the user (and possibly the host)
- User sends the response to the host
 - Access is approved if response is as expected by host.
- Advantage: Since the challenge will be different each time, the response will be too – the dialogue can not be captured and used at a later time
- Could use symmetric or asymmetric crypto

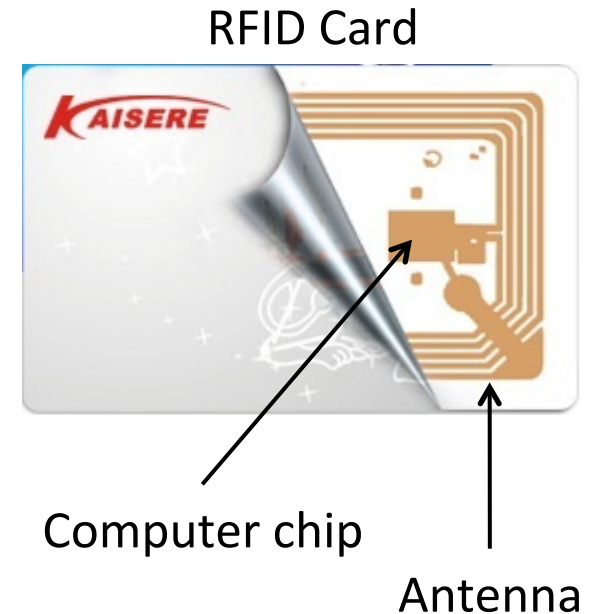
Token-based User authentication Challenge Response Systems



Symmetric algorithm case

Contactless Cards: Overview

- Contactless cards, also called RFID (Radio Frequency Id) cards, consists of a chip and an antenna.
 - No need to be in physical contact with the reader.
 - Uses radio signals to communicate
 - Powered by magnetic field from reader
 - When not within the range of a reader it is not powered and remains inactive.
 - Battery powered RFID tags also exist
- Suitable for use in hot, dirty, damp, cold, foggy environments



Inherence-Based Authentication

Biometrics



Something you are



Something you do

Biometrics: Overview

- What is it?
 - Automated methods of verifying or recognizing a person based upon a physiological characteristics.
- Biometric modalities, examples:
 - fingerprint
 - facial recognition
 - eye retina/iris scanning
 - hand geometry
 - written signature
 - voice print
 - keystroke dynamics

Biometrics: Requirements

- **Universality:**
Each person should have the characteristic;
- **Distinctiveness:**
Any two persons should be sufficiently different in terms of the characteristic;
- **Permanence:**
The characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time;
- **Collectability:**
The characteristic should be measurable quantitatively.

Biometrics: Practical considerations

- **Accuracy:**
 - The correctness of a biometric system, expressed as EER (Equal Error Rate), where a low ERR is desirable.
- **Performance:**
 - the achievable speed of analysis,
 - the resources required to achieve the desired speed,
- **Acceptability:**
 - the extent to which people are willing to accept the use of a particular biometric identifier (characteristic)
- **Circumvention resistance:**
 - The difficulty of fooling the biometric system
- **Safety:**
 - Whether the biometric system is safe to use

Biometrics Safety

- Biometric authentication can be safety risk
 - Attackers might want to “steal” body parts
 - Subjects can be put under duress to produce biometric authenticator
- Necessary to consider the physical environment where biometric authentication takes place.



Car thieves chopped off part of the driver's left index finger to start S-Class Mercedes Benz equipped with fingerprint key. Malaysia, March 2005
(NST picture by Mohd Said Samad)

Biometrics: Modes of operation

- **Enrolment:**
 - analog capture of the user's biometric attribute.
 - processing of this captured data to develop a template of the user's attribute which is stored for later use.
- **Identification (1:N, one-to-many)**
 - capture of a new biometric sample.
 - search the database of stored templates for a match based solely on the biometric.
- **Verification of claimed identity (1:1, one-to-one):**
 - capture of a new biometric sample.
 - comparison of the new sample with that of the user's stored template.

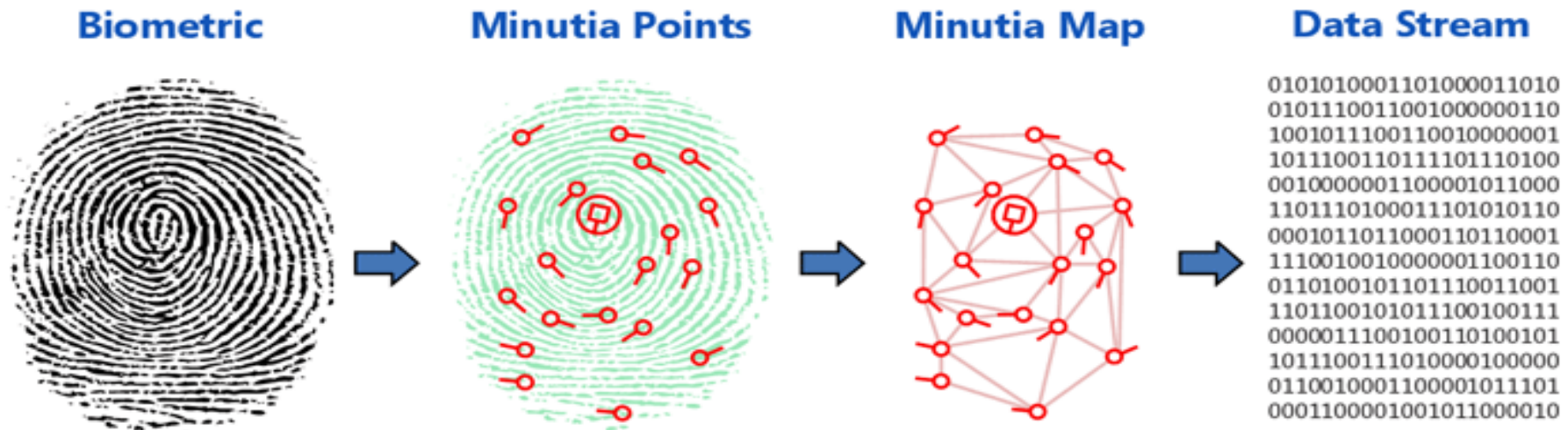
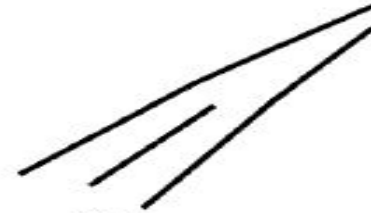
Extracting biometric features

Example fingerprints: Extracting minutia

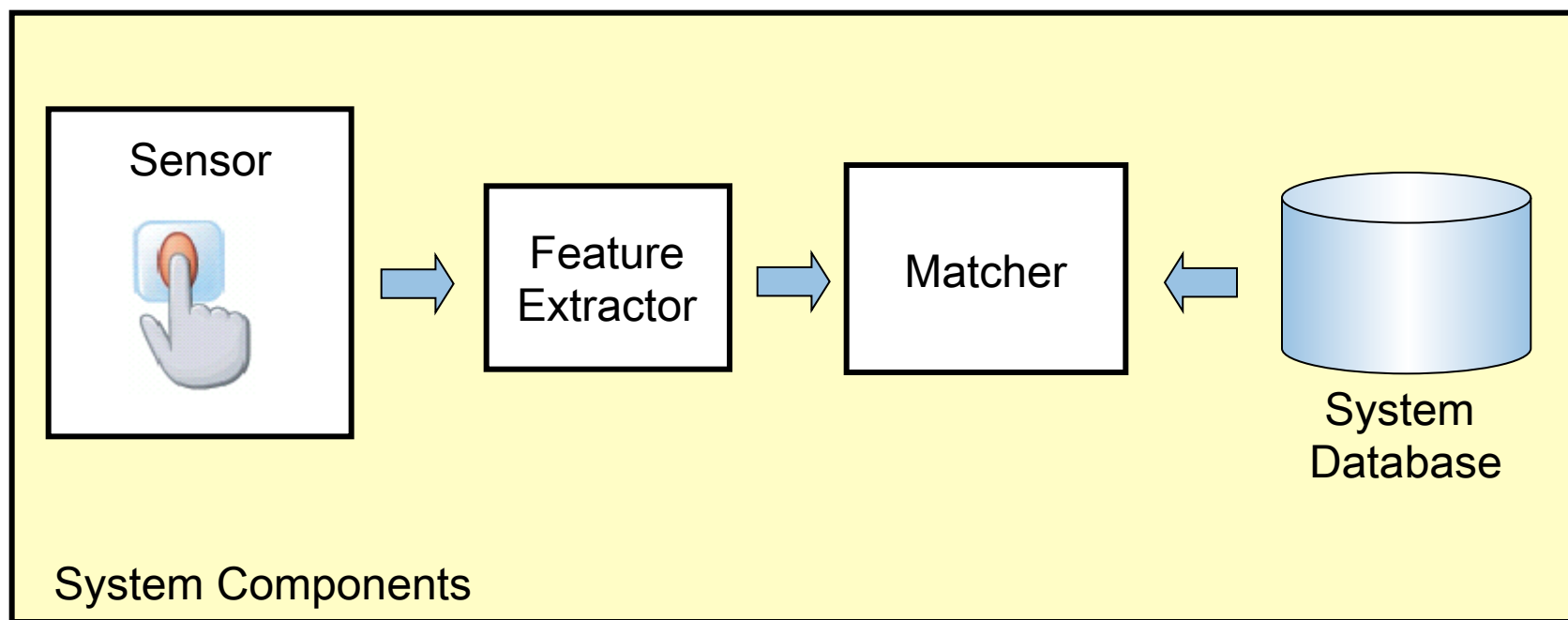
Bifurcation



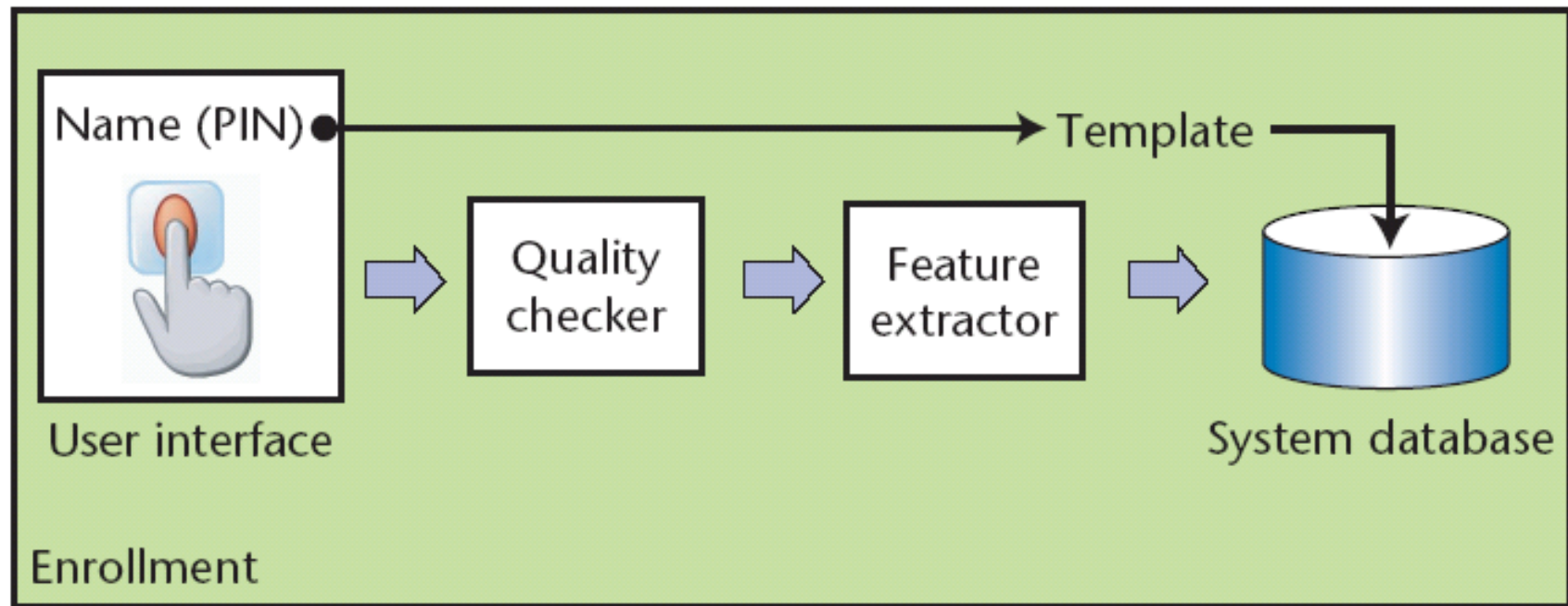
Ridge ending



Biometrics: System components

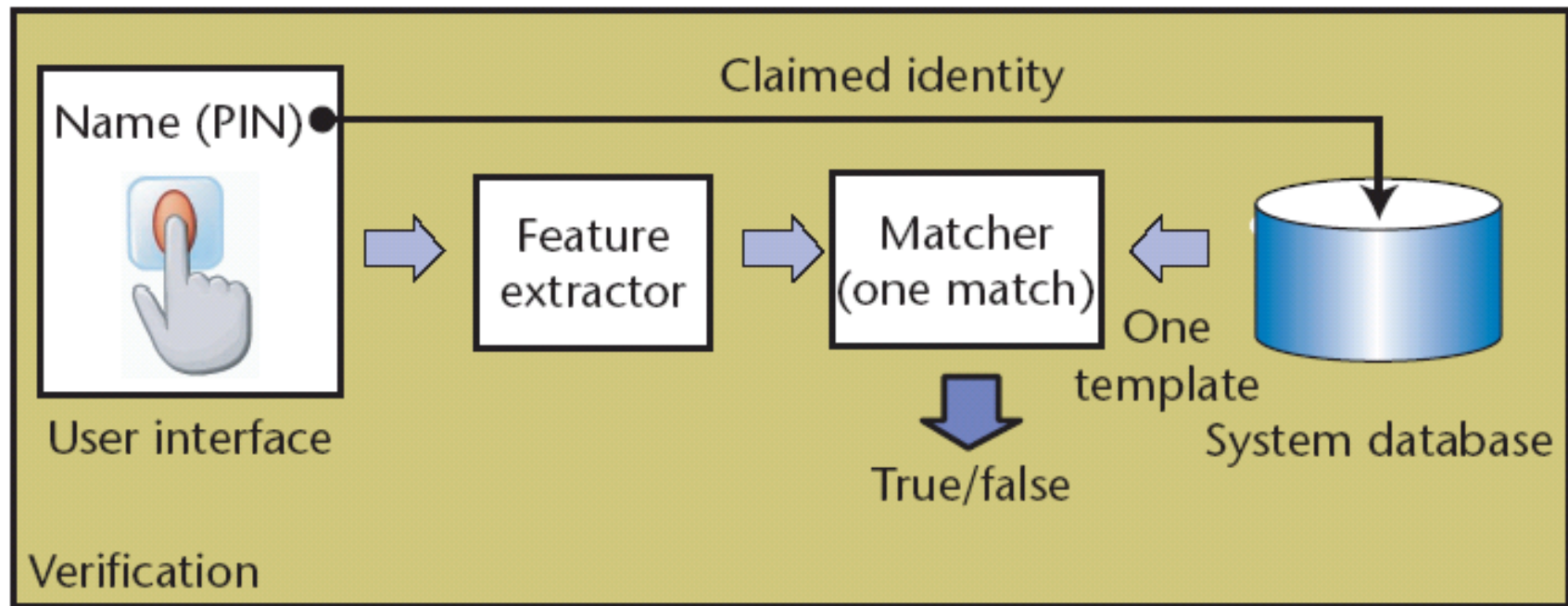


Biometrics: Enrolment



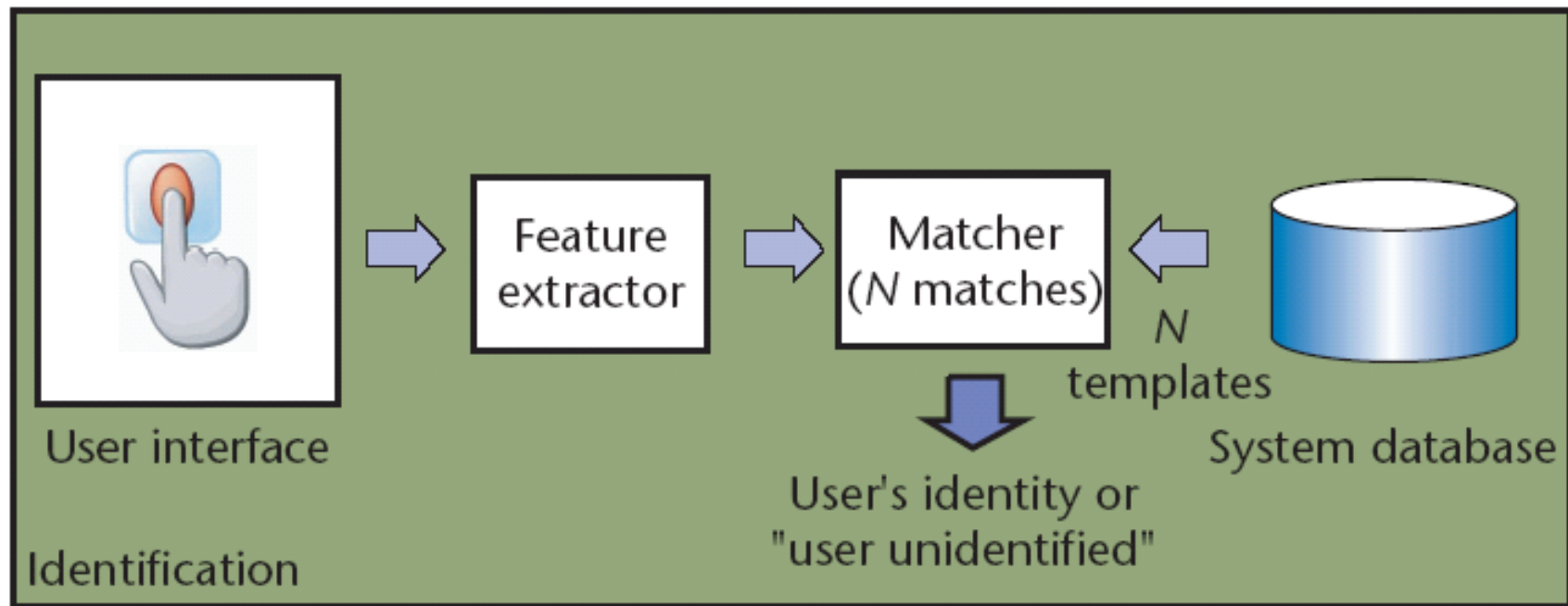
[Biometric Recognition: Security and Privacy Concerns](#)

Biometrics: Verification



[Biometric Recognition: Security and Privacy Concerns](#)

Biometrics: Identification



[Biometric Recognition: Security and Privacy Concerns](#)

Evaluating Biometrics:

- Features from captured sample are compared against those of the stored template sample
- Score s is derived from the comparison.
 - Better match leads to higher score.
- The system decision is tuned by threshold T :
 - System gives a **match** (same person) when the sample comparison generates a score s where $s \geq T$
 - System gives **non-match** (different person) when the sample comparison generates a score s where $s < T$

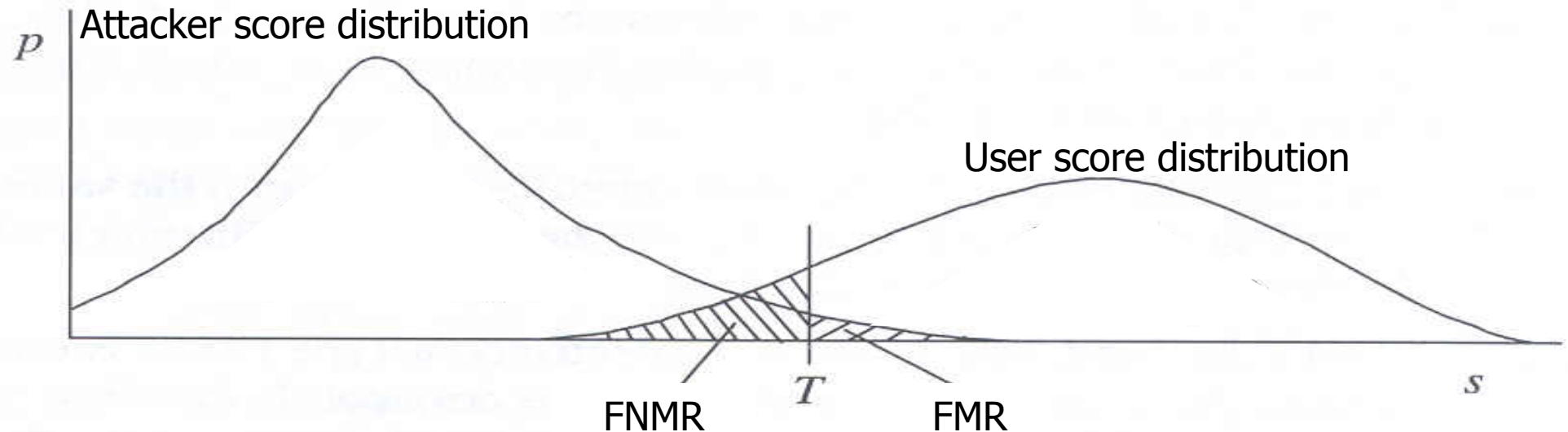
Matching algorithm characteristics

- True positive
 - User sample matches \rightarrow User is accepted
- True negative
 - Attacker sample does not match \rightarrow Attacker is rejected
- False positives
 - Attacker sample matches \rightarrow Attacker is accepted
- False negatives
 - User sample does not match \rightarrow User is rejected
- Computing FMR and FNMR

$$\text{FMR} = (\# \text{ matching attacker samples}) / (\text{total } \# \text{ attacker samples})$$
$$\text{FNMR} = (\# \text{ non-matching user samples}) / (\text{total } \# \text{ user samples})$$
- T determines tradeoff between FMR and FNMR

Evaluating Biometrics: System Errors

- Comparing biometric samples produces score s
- Acceptance threshold T determines FMR and FNMR
 - If T is set low to make the system more tolerant to input variations and noise, then FMR increases.
 - On the other hand, if T is set high to make the system more secure, then FNMR increases accordingly.
- EER (Equal Error Rate) is when $\text{FMR} = \text{FNMR}$.
- Low EER is good.



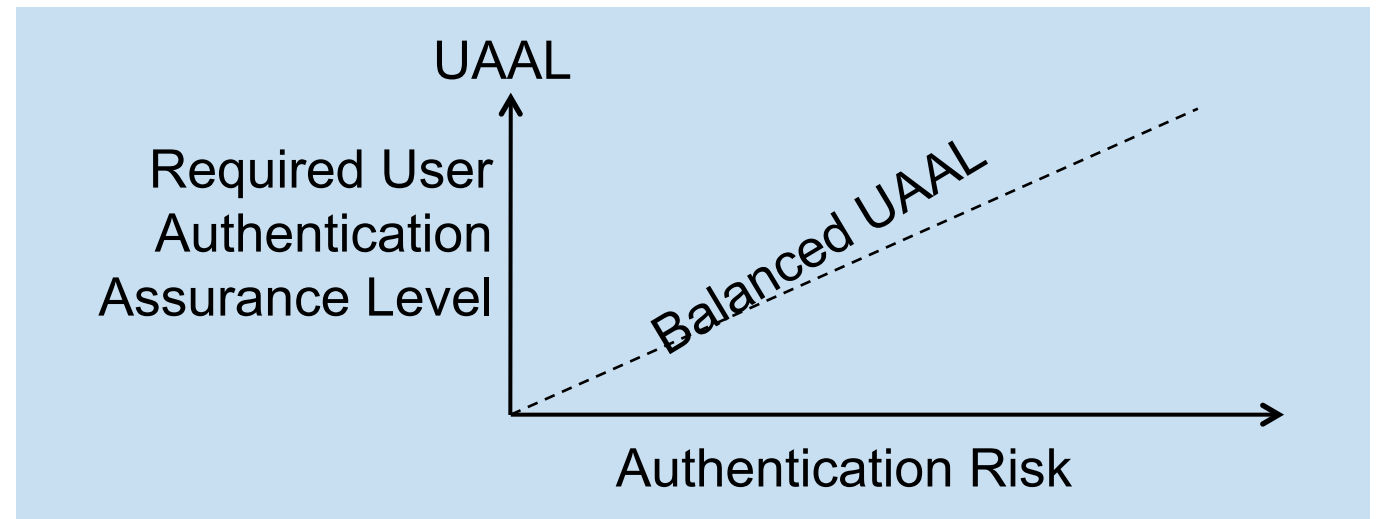
Authentication: Multi-factor



- Multi-factor authentication aims to combine two or more authentication techniques in order to provide stronger authentication assurance.
- Two-factor authentication is typically based on something a user knows (factor one) plus something the user has (factor two).
 - Usually this involves combining the use of a password and a token
 - Example: BankID OTP token with PIN + static password

Authentication Assurance

- Authentication assurance = robustness of authentication
- Resources have different sensitivity levels
 - High sensitivity gives high risk in case of authentication failure
- Authentication has a cost
 - Unnecessary authentication assurance is a waste of money
- Authentication assurance should balance resource sensitivity



e-Authentication Frameworks for e-Gov.

- Trust in identity is a requirement for e-Government
- Authentication assurance produces identity trust.
- Authentication depends on technology, policy, standards, practice, awareness and regulation.
- Consistent frameworks allow cross-national and cross-organisational schemes that enable convenience, efficiency and cost savings.

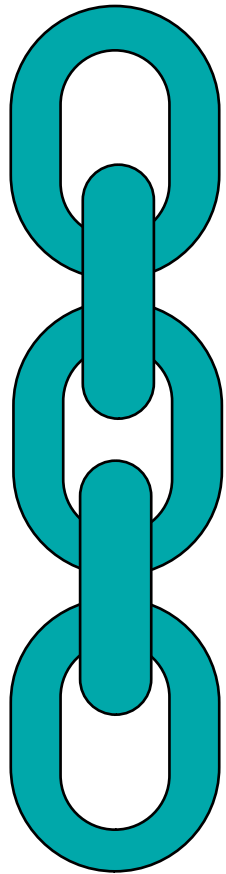


Alignment of e-Authentication Frameworks

<i>Authentication Framework</i>	<i>User Authentication Assurance Levels</i>				
OMB / NIST USA 2004 / 2011	Little or no assurance (1)	Some (2)	High (3)	Very High (4)	
RAU / FAD Norway 2008	Little or no assurance (1)	Low (2)	Moderate (3)	High (4)	
STORK QAA EU 2009	No or minimal (1)	Low (2)	Substantial (3)	High (4)	
NeAF Australia 2009	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
e-Pramaan India 2012	None (0)	Minimal (1)	Minor (2)	Significant (3)	Substantial (4)
ISO 29115 ISO/IEC 2013	Low (Little or no) (1)	Medium (2)	High (3)	Very High (4)	

UAAL: User Authentication Assurance Level

- UAAL is determined by the weakest of three links:



User Identity Registration
Assurance
(UIRA) requirements

Requirements for correct registration:

- Pre-authentication credentials, e.g.
 - birth certificate
 - biometrics

User Credential Management
Assurance
(UCMA) requirements

Requirements for secure handling
of credentials:

- Creation
- Distribution
- Storage

User Authentication Method
Strength
(UAMS) requirements

Requirements for mechanism strength:

- Password length and quality
- Cryptographic algorithm strength
- Tamper resistance of token
- Multiple-factor methods

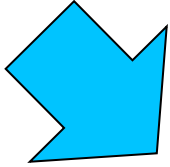
UAAL: User Authentication Assurance Levels

No Assurance	Minimal Assurance	Low Assurance	Moderate Assurance	High Assurance
Level 0	Level 1	Level 2	Level 3	Level 4
No registration of identity required	Minimal confidence in the identity assertion	Low confidence in the identity assertion	Moderate confidence in the identity assertion	High confidence in the identity assertion

Example taken from Australian NeAF 2009

Risk Analysis for Authentication

Determining the appropriate UAAL for an application

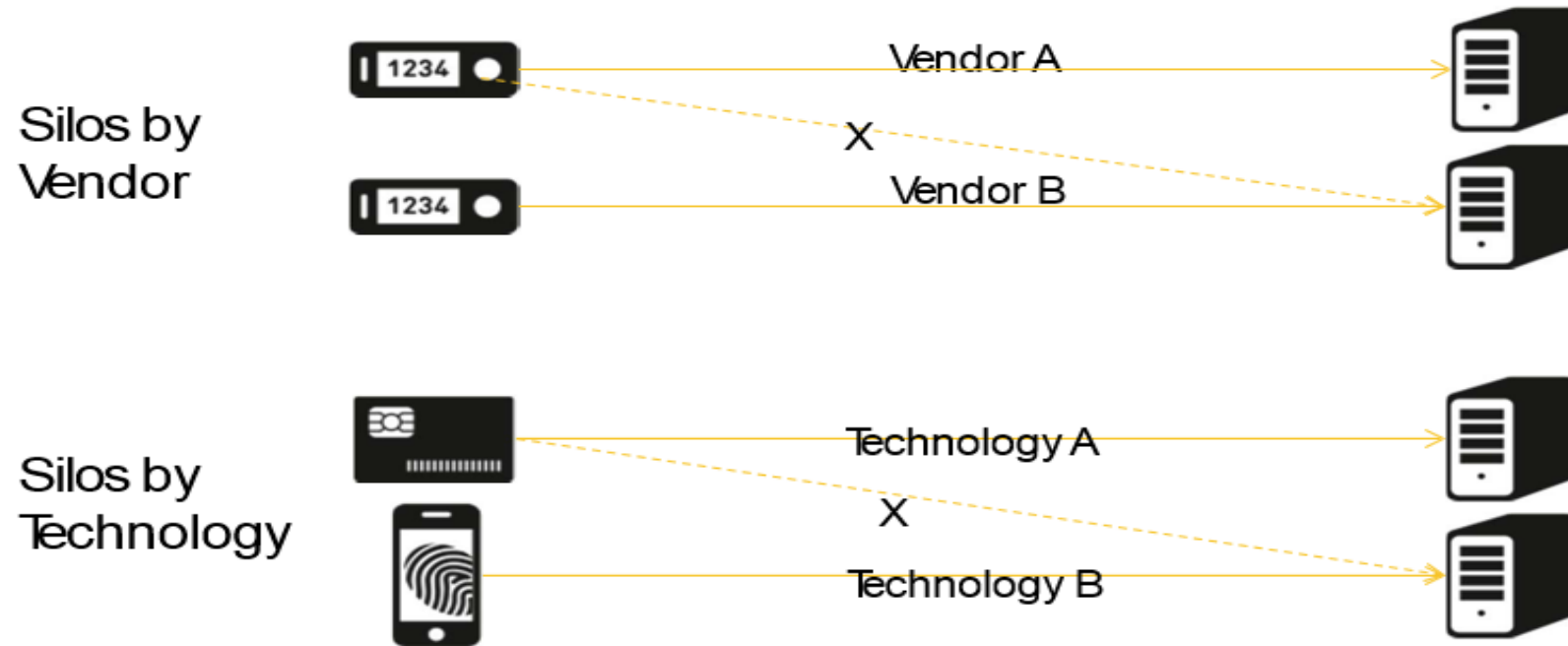


		Impact of e-Authentication failure				
		Insignificant	Minor	Moderate	Major	Severe
Likelihood	Almost Certain	None (0)	Low (2)	Moderate (3)	High (4)	High (4)
	Likely	None (0)	Low (2)	Moderate (3)	High (4)	High (4)
	Possible	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
	Unlikely	None (0)	Minimal (1)	Low (2)	Moderate (3)	Moderate (3)
	Rare	None (0)	Minimal (1)	Low (2)	Moderate (3)	Moderate (3)

Example: NeAF Australia

How do we implement
authentication mechanisms?

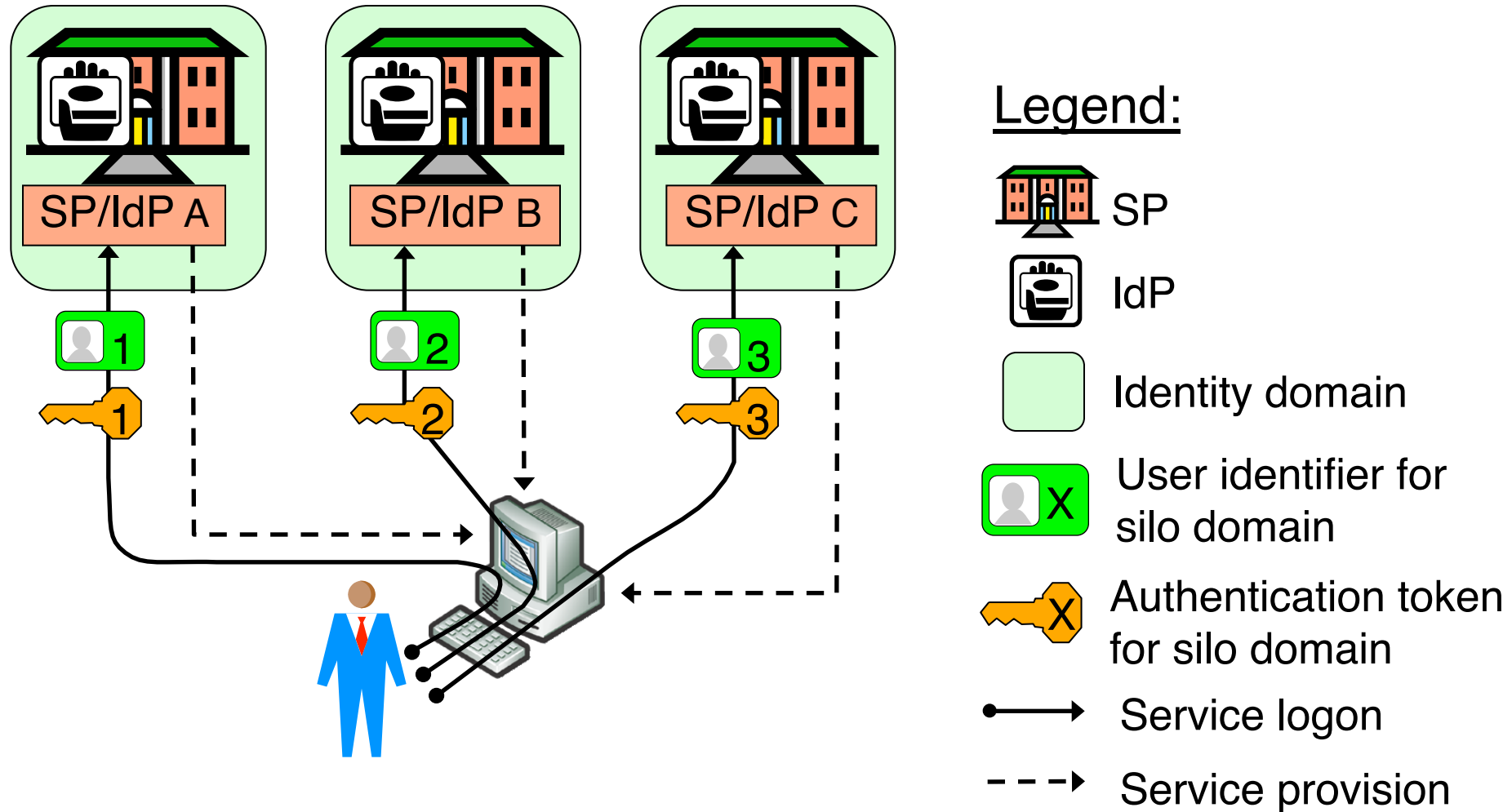
Implementation Is the challenge: Current State



Each new authentication solution requires new HW, SW, and Infrastructure.

➔ We're building 'Silos' of authentication

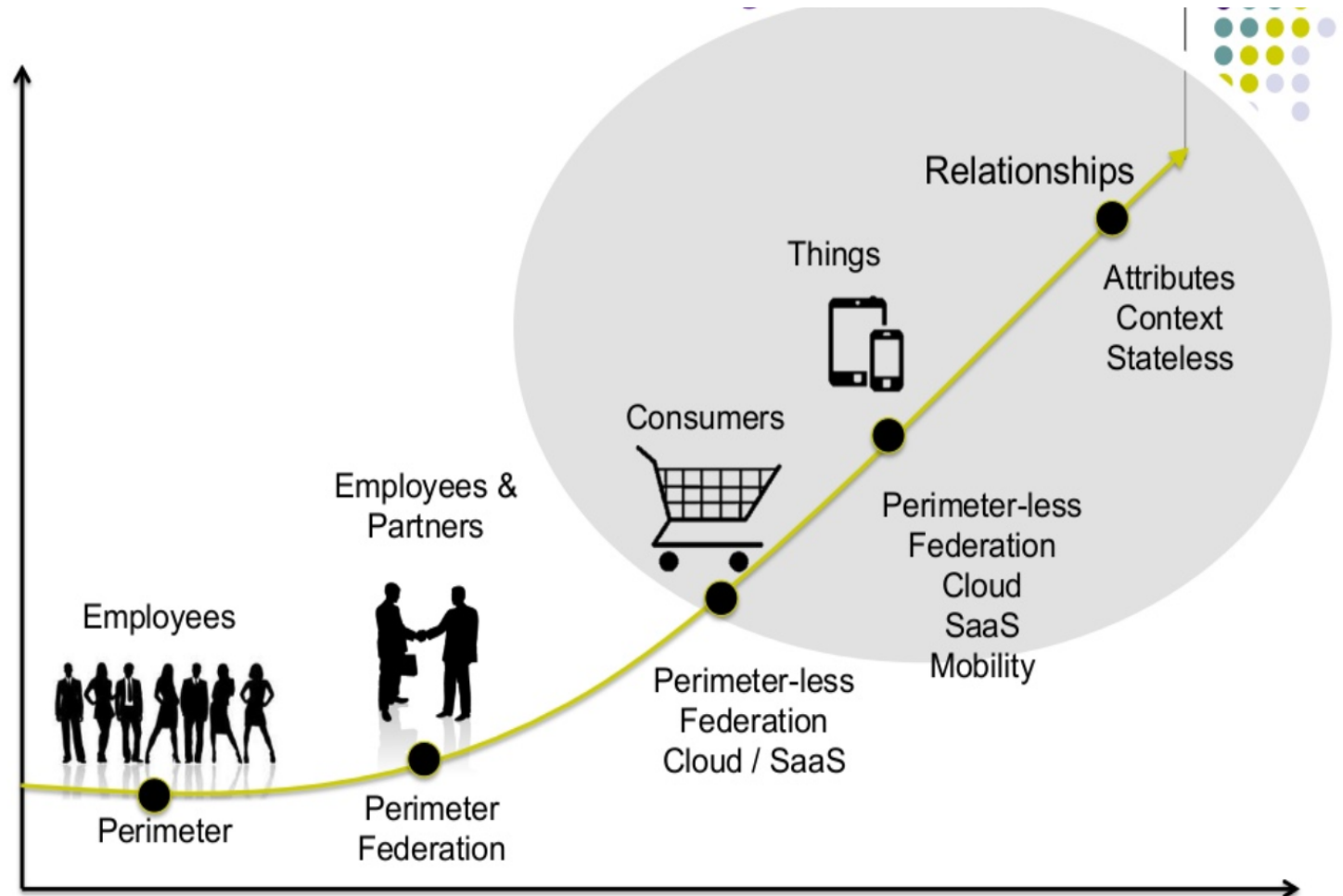
Current State: Silo identity management model



How are we doing ?

We are moving
at 2 (3?) speeds:

- Enterprise Scale.
- Internet Scale.
- BYOA/BYOD trend.



Traditional Enterprise Vs. Traditional Internet

- 100% managed infrastructure.
 - 100% inside perimeter infrastructure.
 - Known/Finite number of users.
 - Expected growth.
 - Non mobile.
 - Non agile.
- A small number of online services.
 - Non mobile.
 - Separate identity from the the enterprises.
 - Separate applications context that enterprises.

Modern Enterprise vs modern internet

- Blurred lines between work and non work identities.
- Mobility.
- Work applications are not exclusively deployed on the on premises infratsructure.
- Agility requirement.
- More social.
- BYOD/BYOA/BYOID.

Who is driving whom?

Traditional IdM enterprise solution

- On premise domains of trust.
- Centralized users directory.

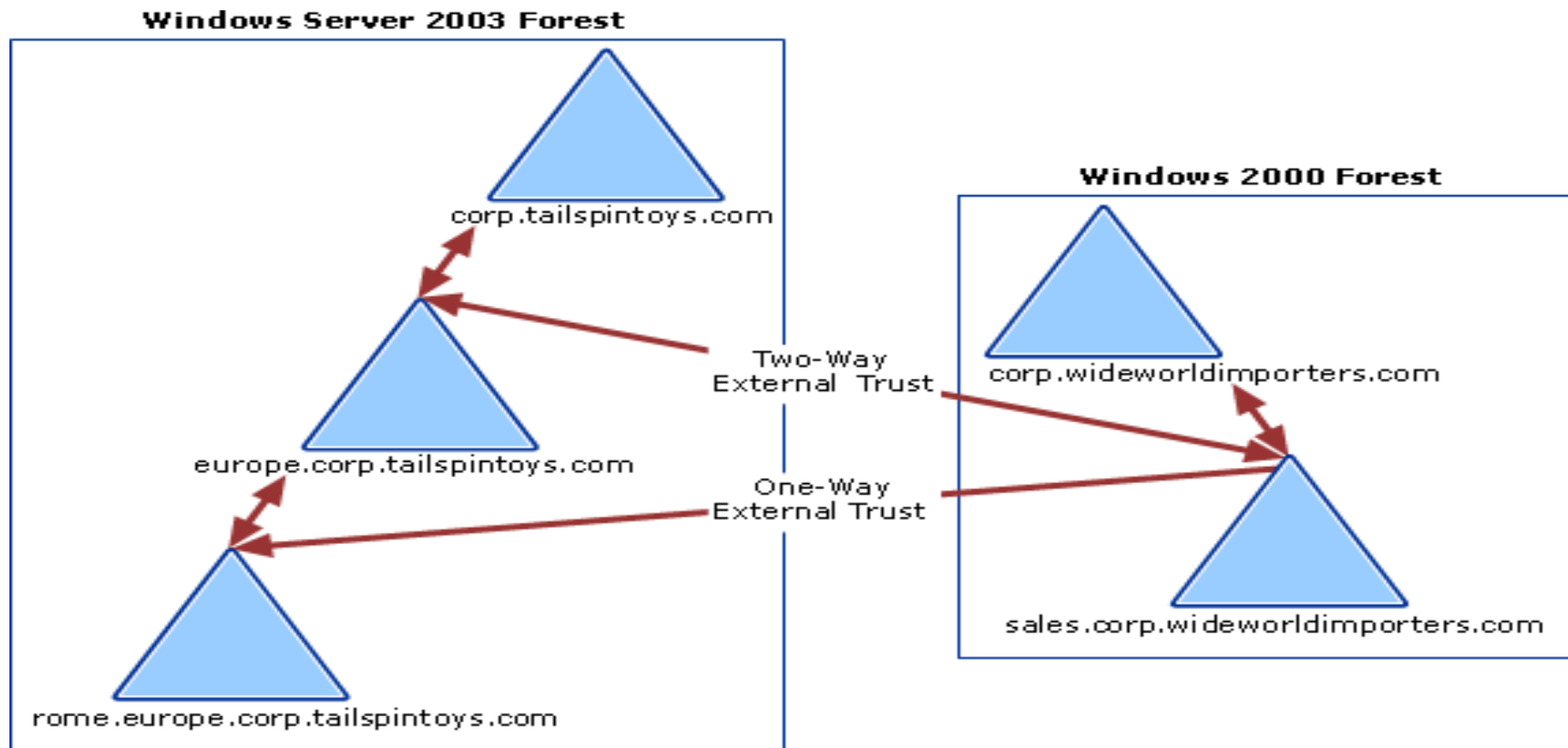
Example:

- Active directory/kerberos.

Collaboration needs

- 2 partner organizations.
- Established trust.
- Example:
 - Extending Trust between two active directory forests.
 - Using Active Directory Federation Services(ADFS).

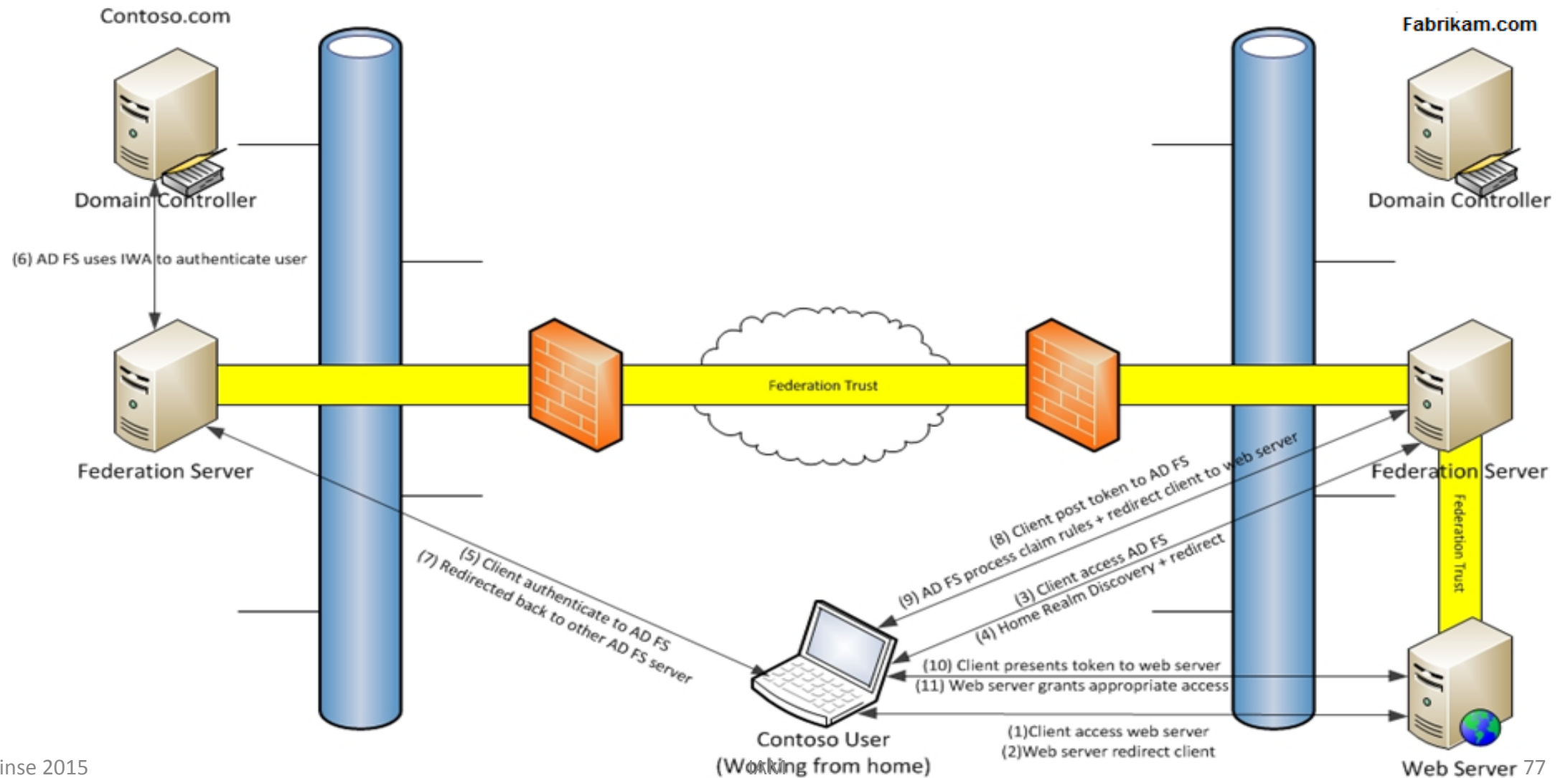
Trust between two active directory forests



However, there are scenarios in which forest trusts are not a viable option.

access across organizations may need to be limited to only a small subset of individuals, not every member of a forest.

ADFS



ADFS with partner organization with different technologies

- **AD FS shields the internal Active Directory Domain Services (AD DS) from external attack by accepting requests for authentication** from the Internet. AD FS translates the Active Directory-based identity into an Internet-friendly format.

In Parallel...

- Growth of Web distributed applications: SaaS
- SaaS Identity Providers Vs. SaaS Service Providers

- Salesforce
- GoogleDocs
- Sharepoint Online
- Facebook
- Microsoft.
- Google

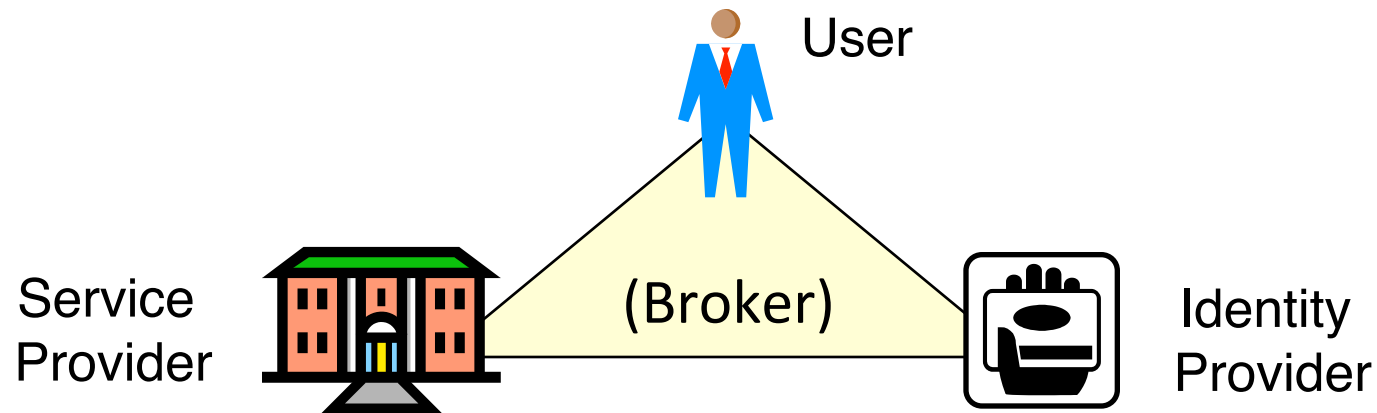
- HR/Collaboration/CRM tools
- Etc...

Web Distributed applications' Identity Management

- Challenge: Need to integrate with Enterprise IdM solutions
- Proposed Solutions:
 - Moving user directories to the cloud.
 - Web Federation.
- Adopted Solutions ?

Federation protocols

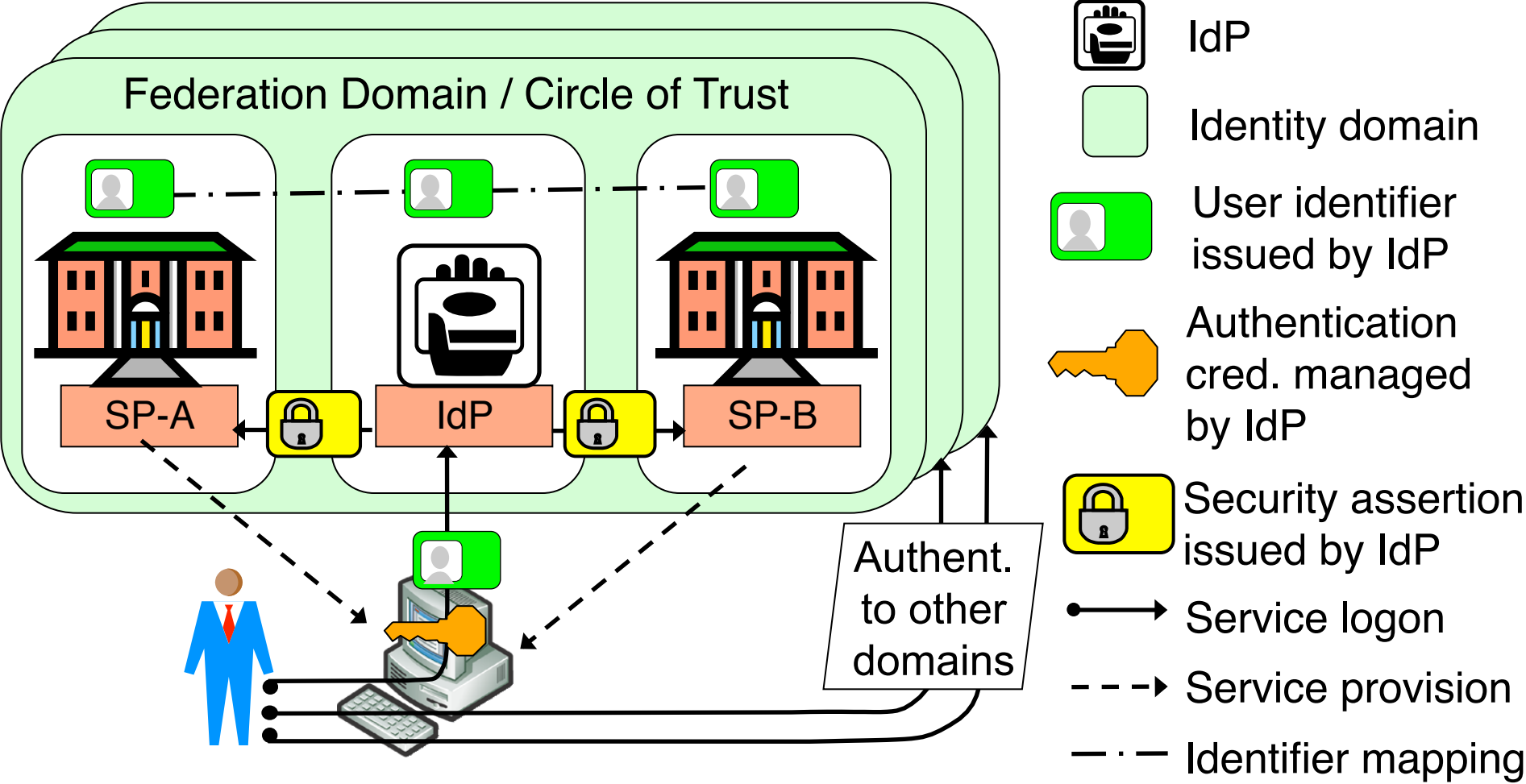
- Authentication by one IdP or SP is communicated as a security assertions (cryptographic token) to other SPs that trust and accept it
- Usually based on SAML protocol
 - Security Assertions Markup Language
- Involves multiple entities
 - User, IdP, SP, and sometimes broker entity



Federated Identity Management

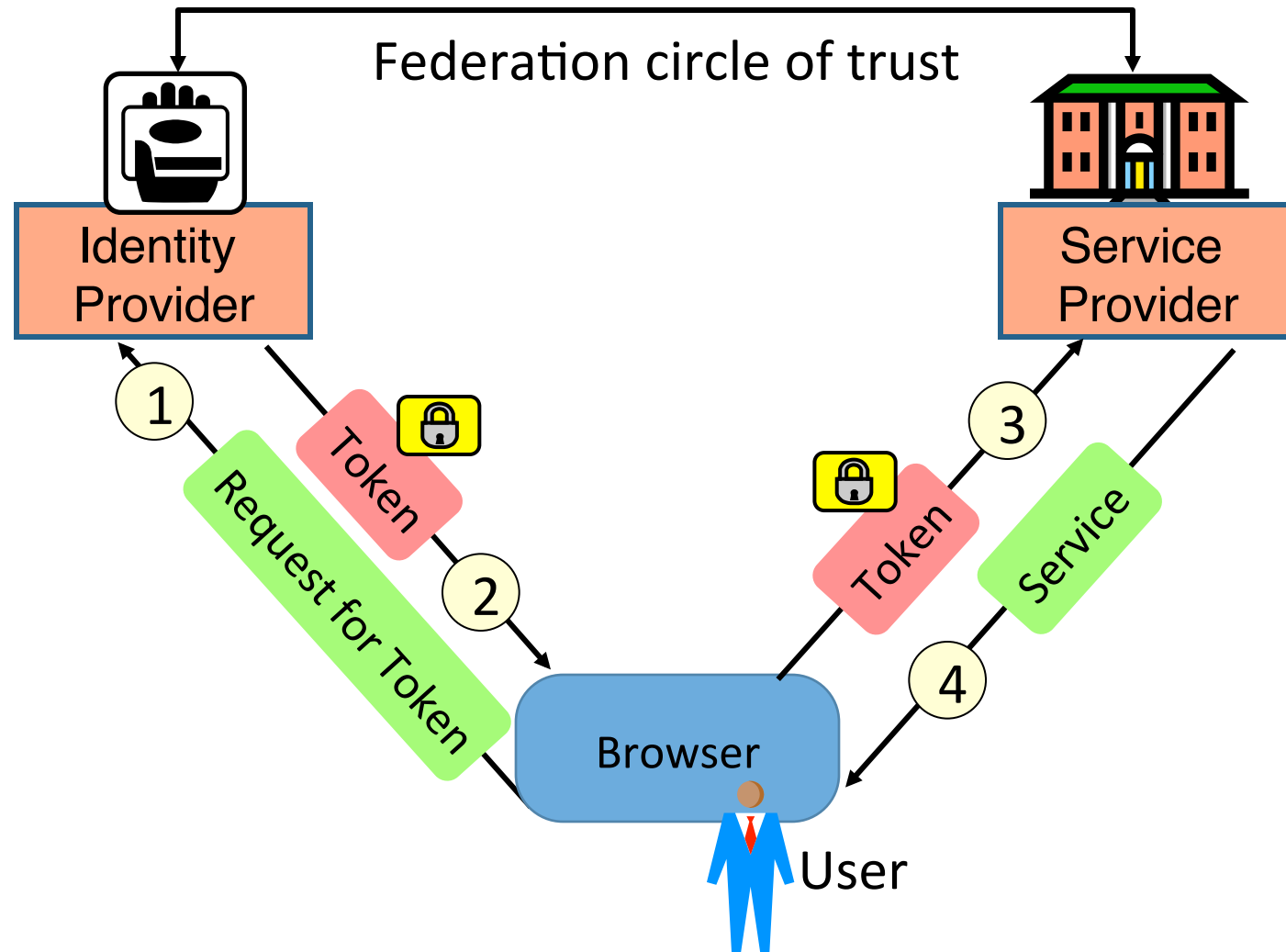
- Advantages
 - Improved usability
 - Compatible with silo user-identity domains
 - Allows SPs to bundle services and collect user info
- Disadvantages
 - High technical and legal complexity
 - High trust requirements
 - E.g. SP-A is technically able to access SP-B on user's behalf
 - Privacy issues,
 - IdP collects info about user habits wrt. which SPs are used
 - Limited scalability,
 - Can only federate SPs with similar interests
 - An Identity federation becomes a new silo

Federated model (centralised)

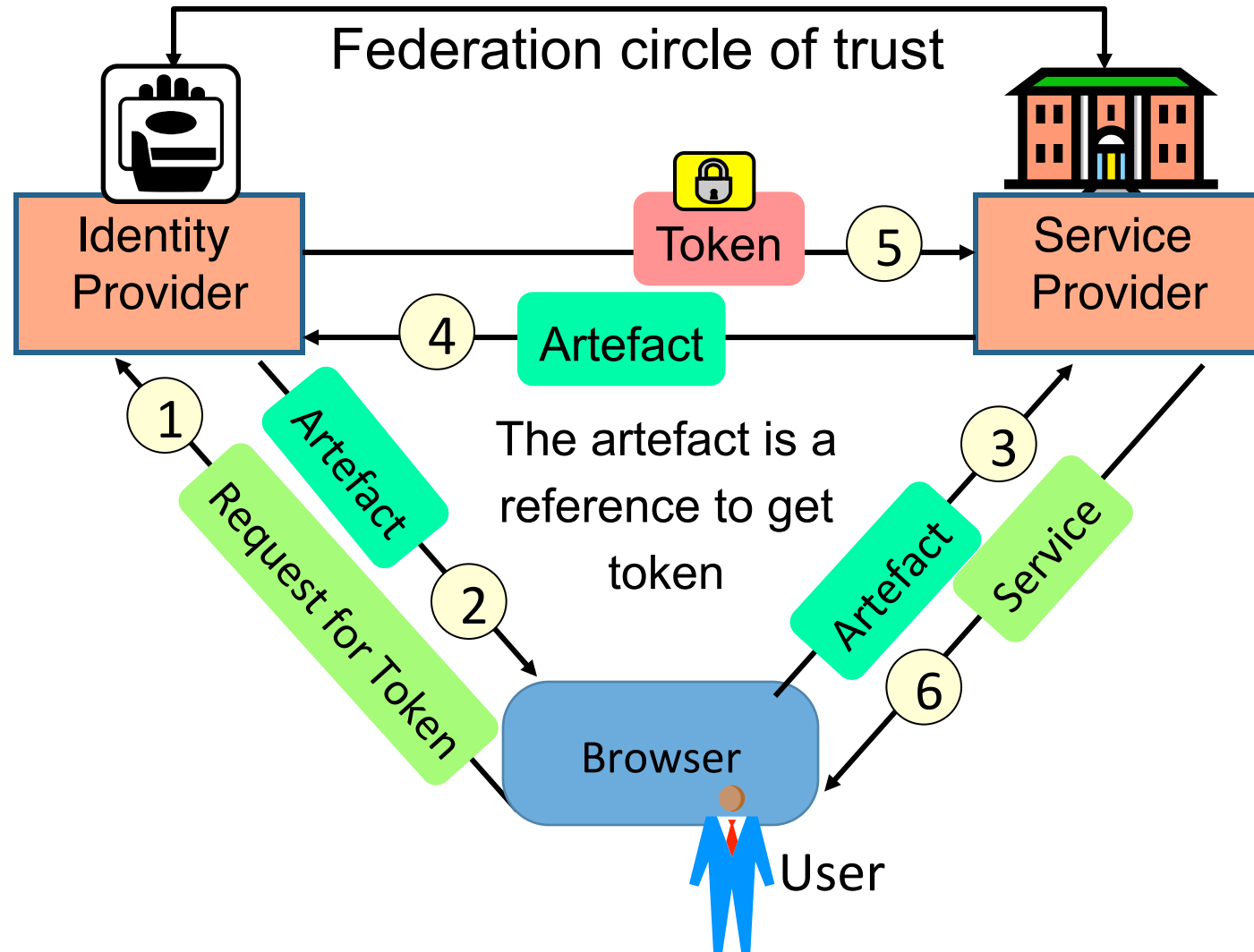


Examples: Liberty Alliance, SAML2.0, WS-Federation, Shibboleth

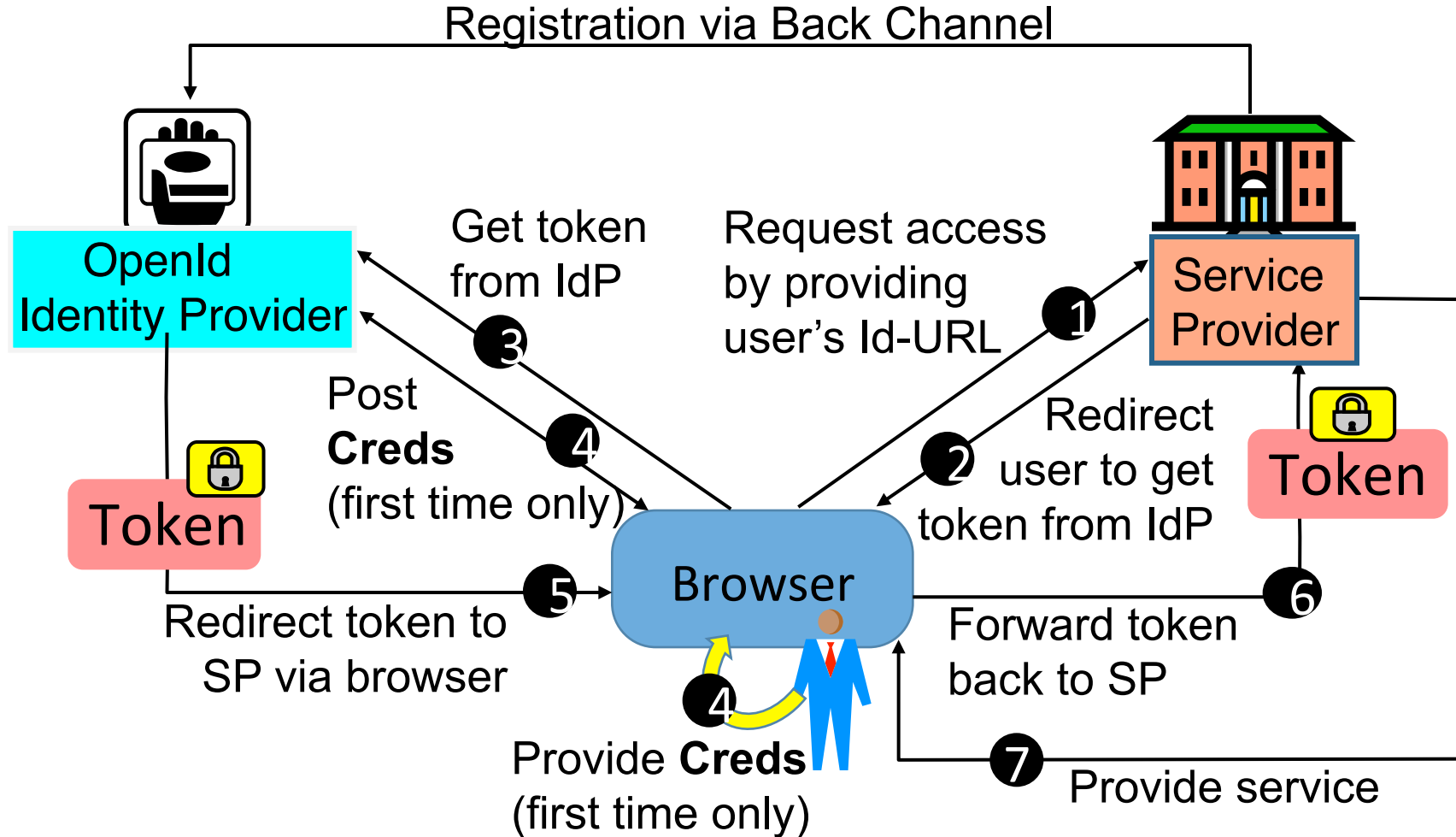
SAML protocol profile: Browser Post Security token via front-end



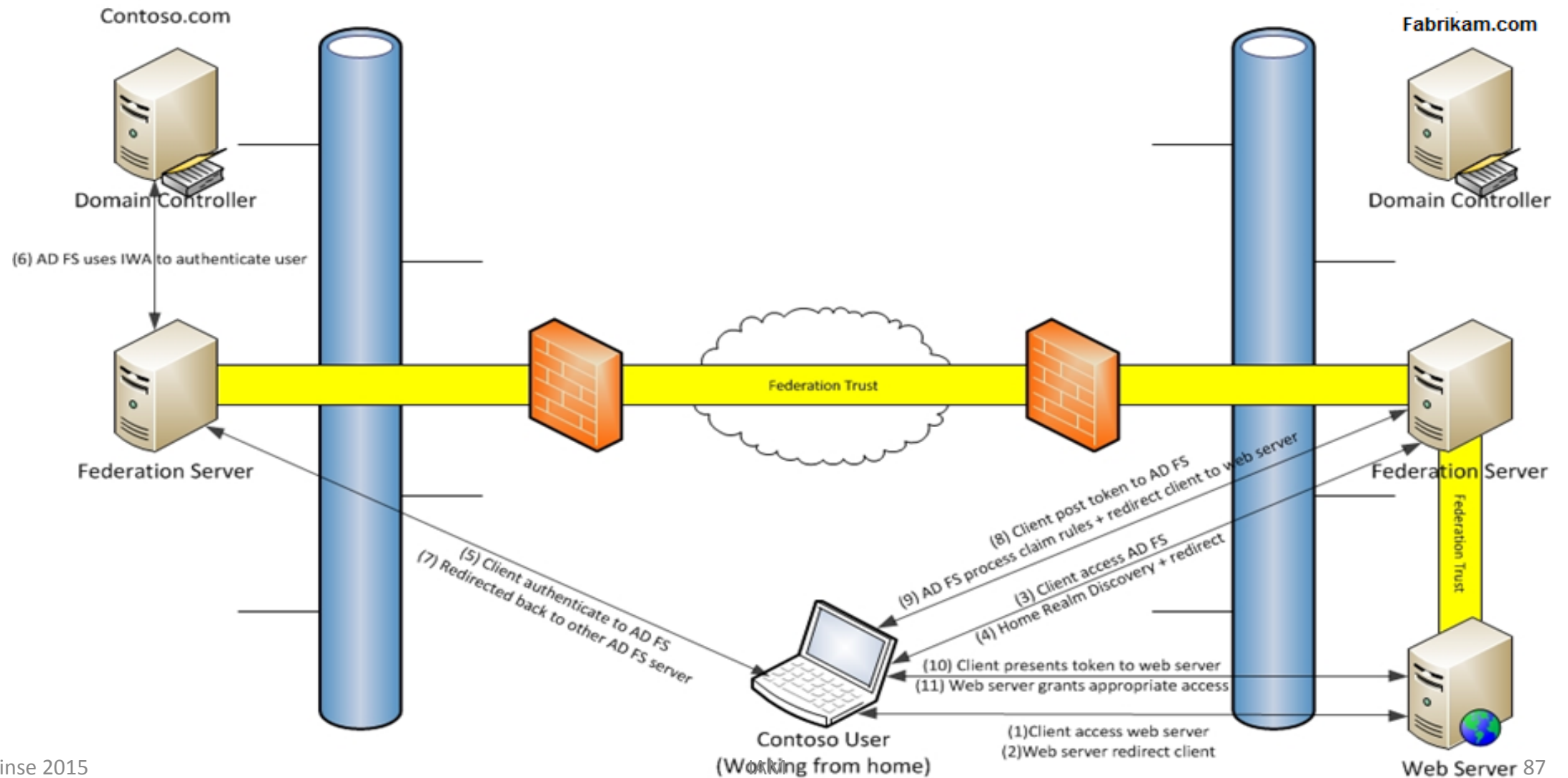
SAML protocol profile: Browser Artefact Security token via back-end



OpenID Distributed Federation

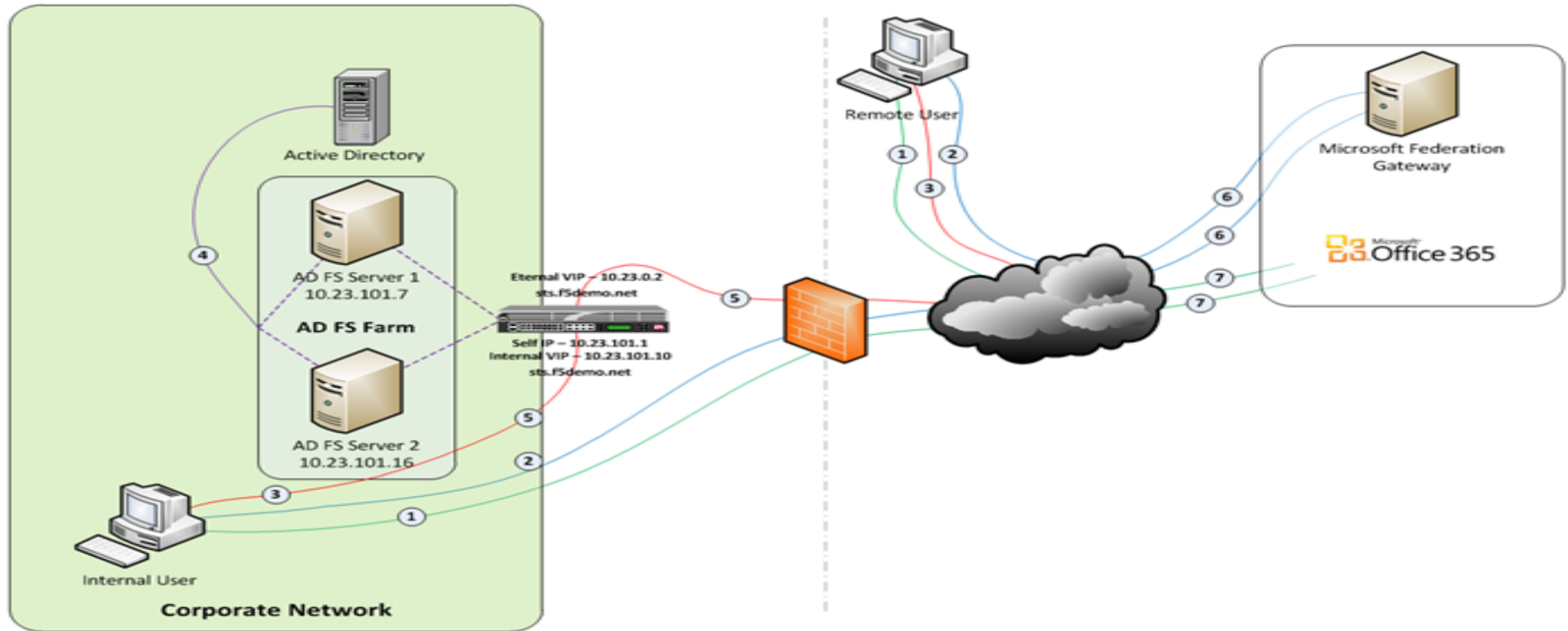


ADFS



ADFS with SaaS

Figure 2 – BIG-IP with APM replacing the ADFS Proxy layer



OpenID self registration

Sign Up - Windows Internet Explorer

https://www.myopenid.com/signup

File Edit View Favorites Tools Help

Sign Up

1. CHOOSE YOUR USERNAME

Your OpenID URL is how [sites that accept OpenID](#) know you. You can use your name or anything that you want to be known by.

Username
John Doe, jdoe123

OpenID URL http://josang.myopenid.com/

2. CHOOSE A PASSWORD

You'll use this password to sign in to myOpenID, but you won't have to give it to any other site.

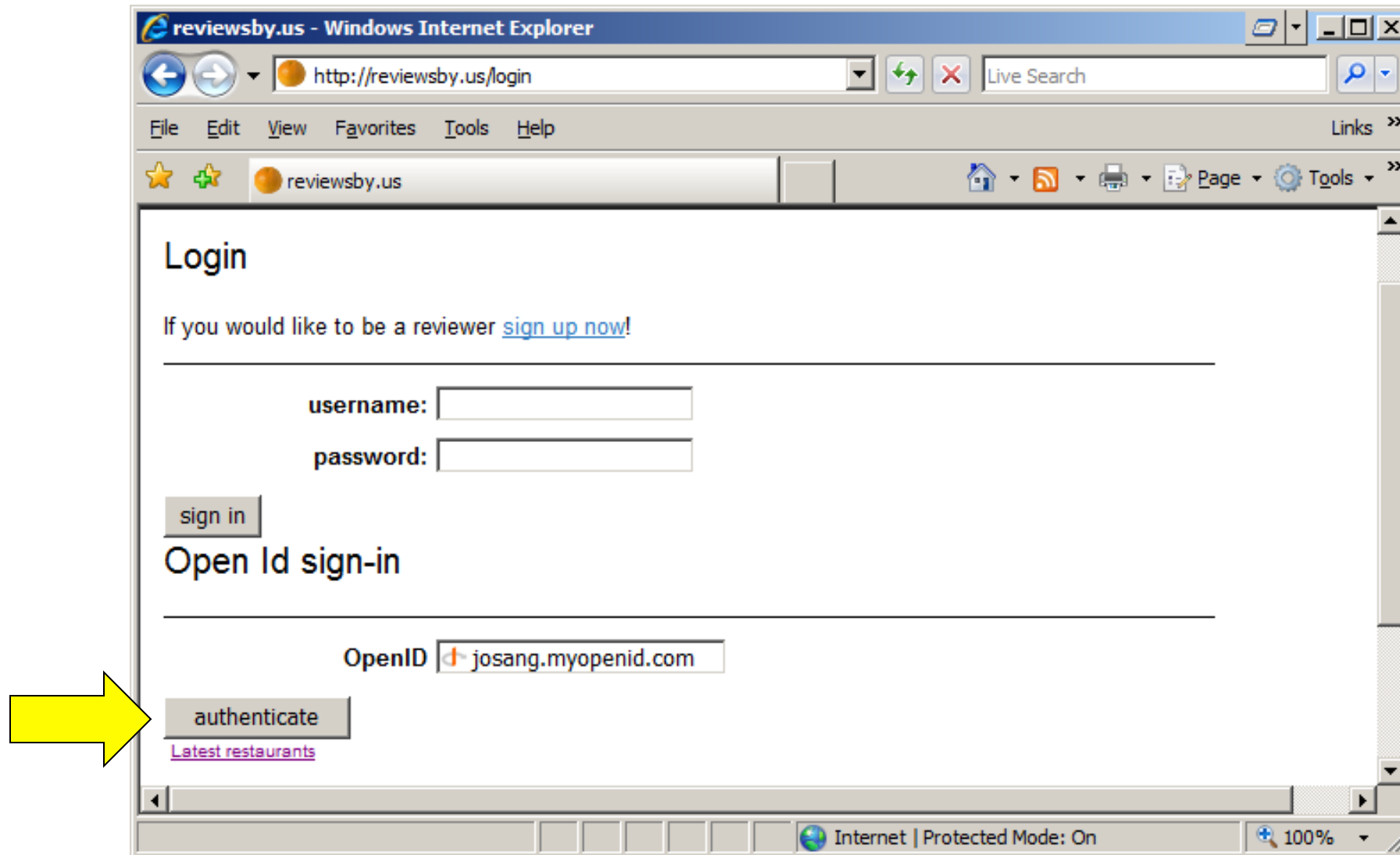
Password fred

Password (confirm)

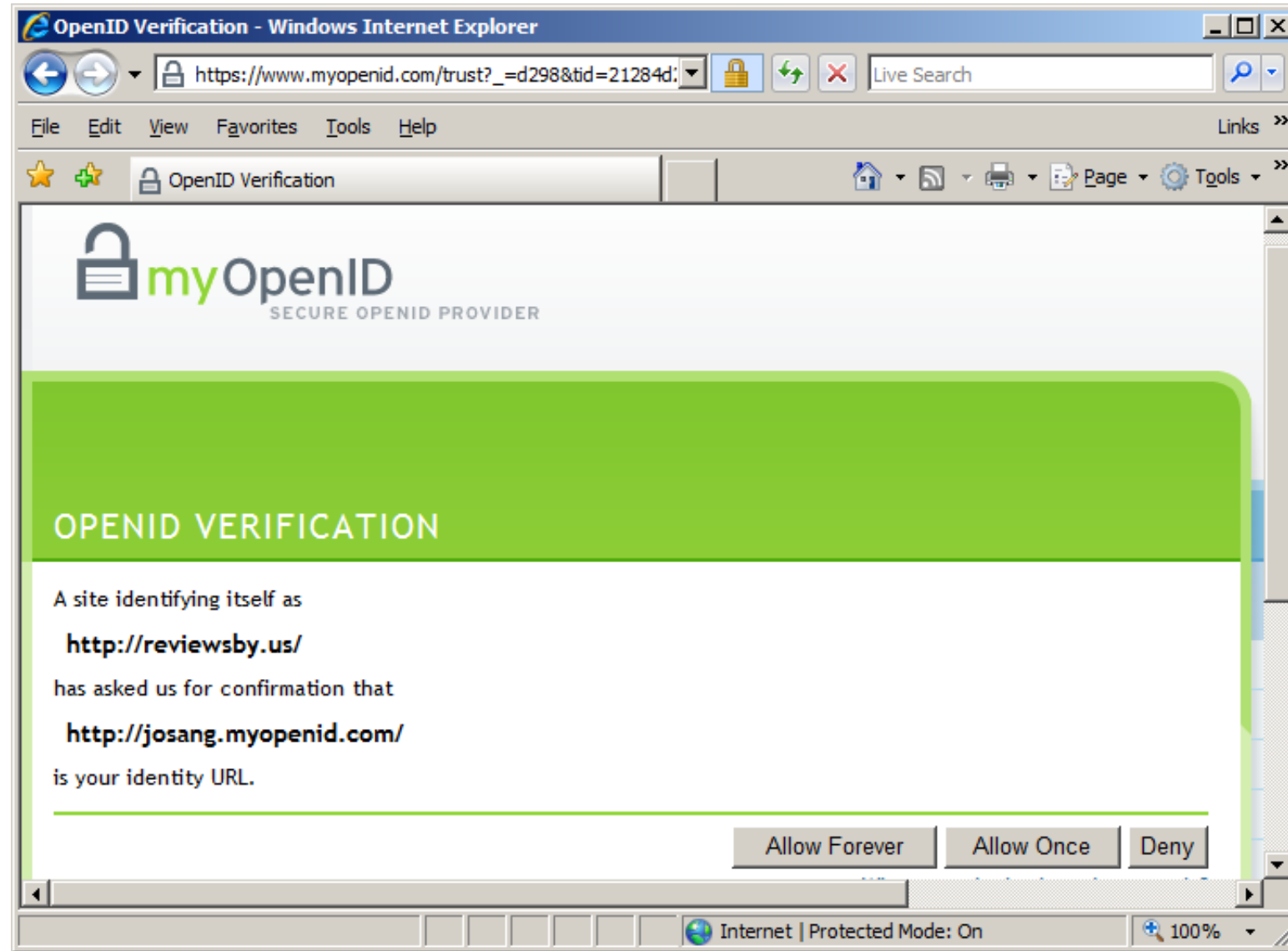
Strength bad password

Internet | Protected Mode: On 100%

Service Access Without Password



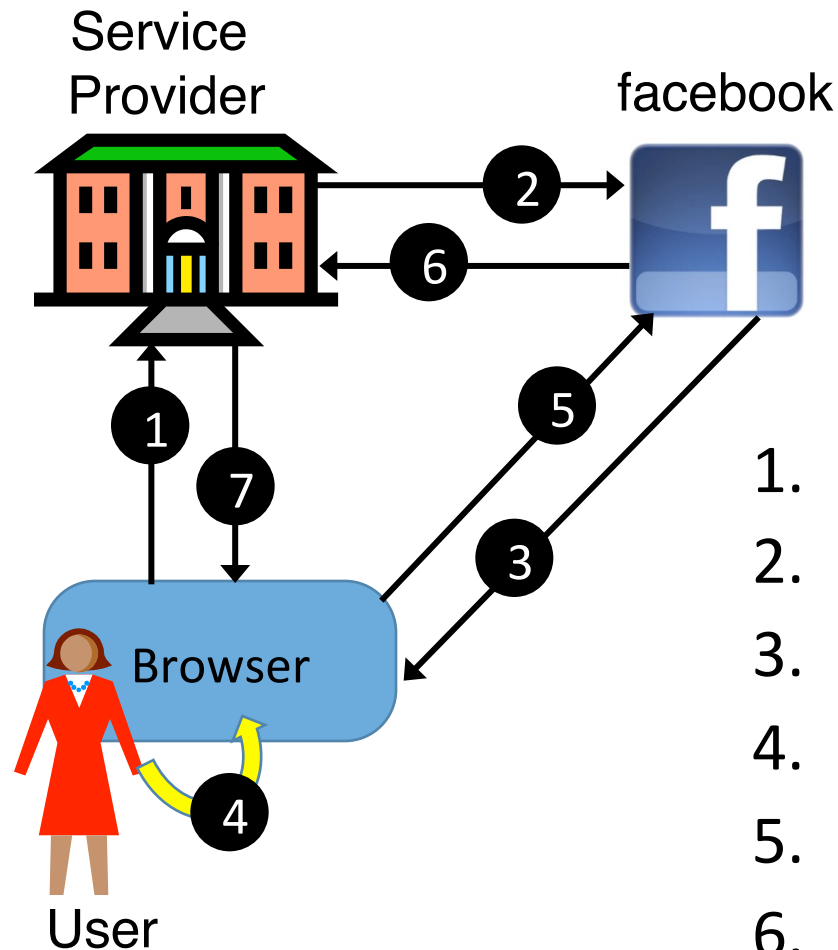
First Time Service Access



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 for services that only require low authentication assurance
- Vulnerable to multiple forms of abuse

Facebook Centralised Federation



Authentication with Facebook Connect

1. User requests service
2. Redirect to facebook authentication
3. Present facebook login form
4. User provides Id + credential
5. Credentials forwarded to facebook
6. Confirm authenticated user
7. Provide service

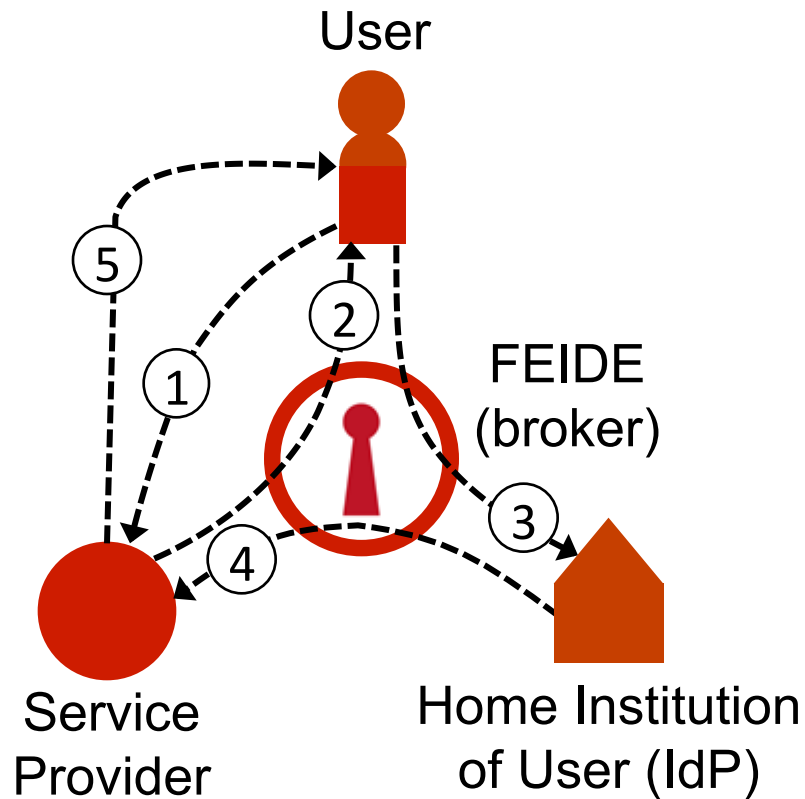
(Felles Elektronisk Identitet)

- FEIDE is a distributed federation with centralised broker for the Norwegian national education sector.
- Users register username and password with own home organisation
- Users authenticate to web-services via FEIDE's centralized login service
- The Service Provider receives user attributes from the user's Home Institution
- The Service Providers never sees the user's password/credential, it only receives user attributes that it need to know in order to provide the service.

FEIDE (continued)

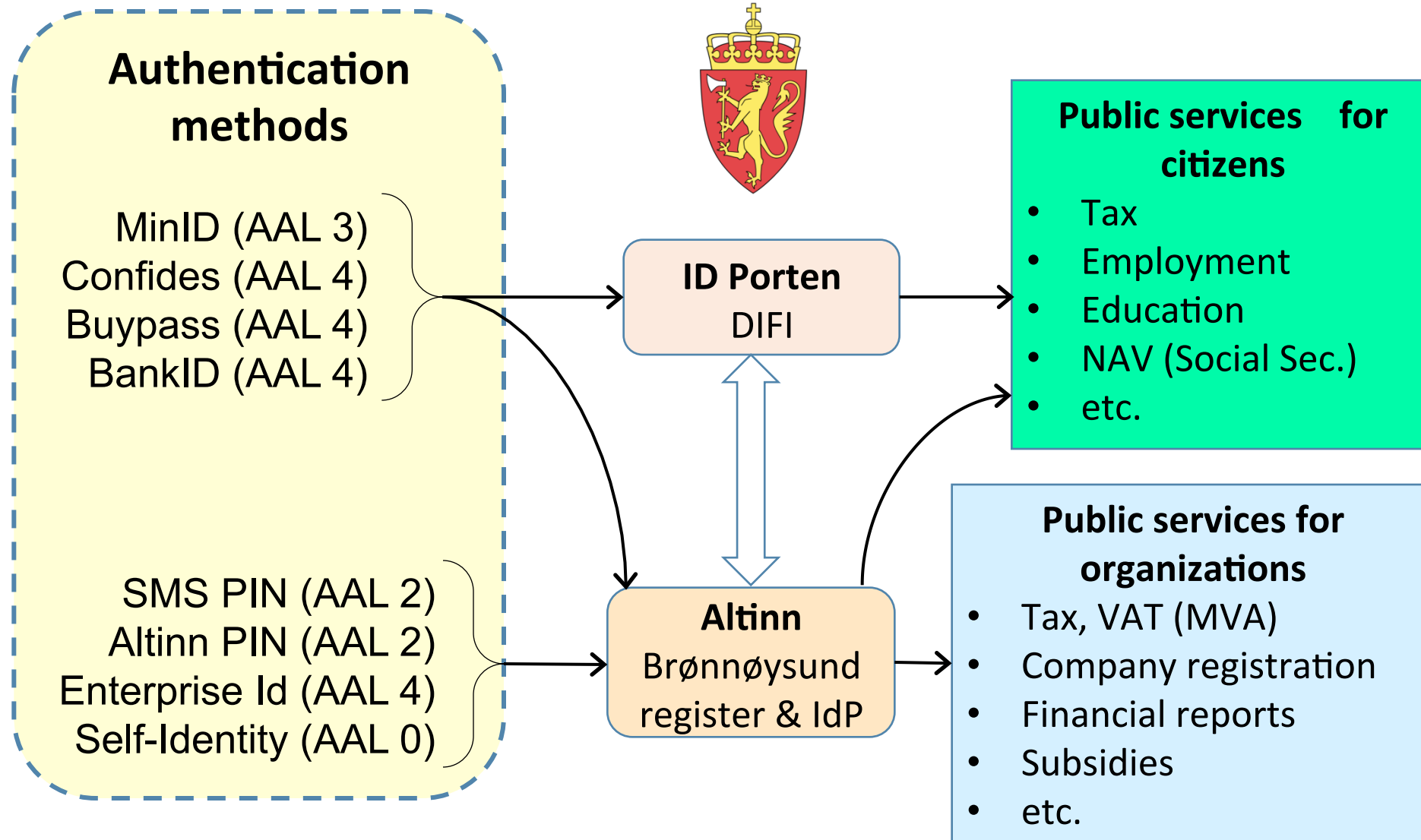
- FEIDE has formal agreements with the universities and schools before they are connected
- Home Institutions (universities and schools) are responsible for keeping user data correct and up-to-date
- Service Providers decide themselves what services their own users and other users should be able to access via FEIDE's central log-in service.

FEIDE Scenario



1. User requests access to service
2. Service Provider sends authentication request to FEIDE, and displays FEIDE login form to user.
3. User enters name and password in FEIDE login form, which are sent for validation to Home Institution of user.
4. Home Institution confirms authentic user and provides user attributes to FEIDE which forwards these to SP
5. Service Provider analyses user attributes and provides service according to policy

Norw. e-Gov. Distributed Fed. with Broker

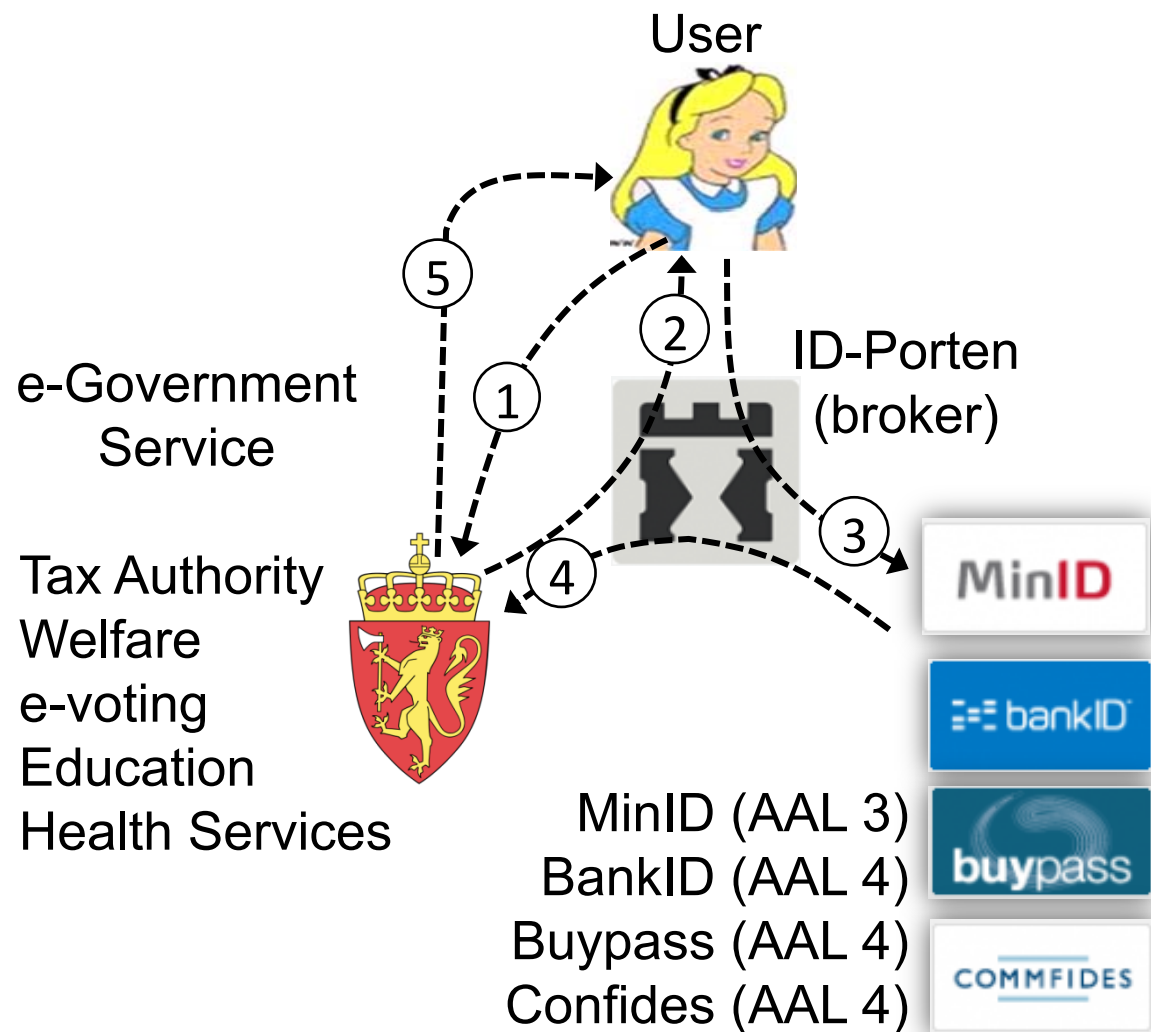




ID-porten/MinID
Innlogging til offentlige tjenester

Scenario

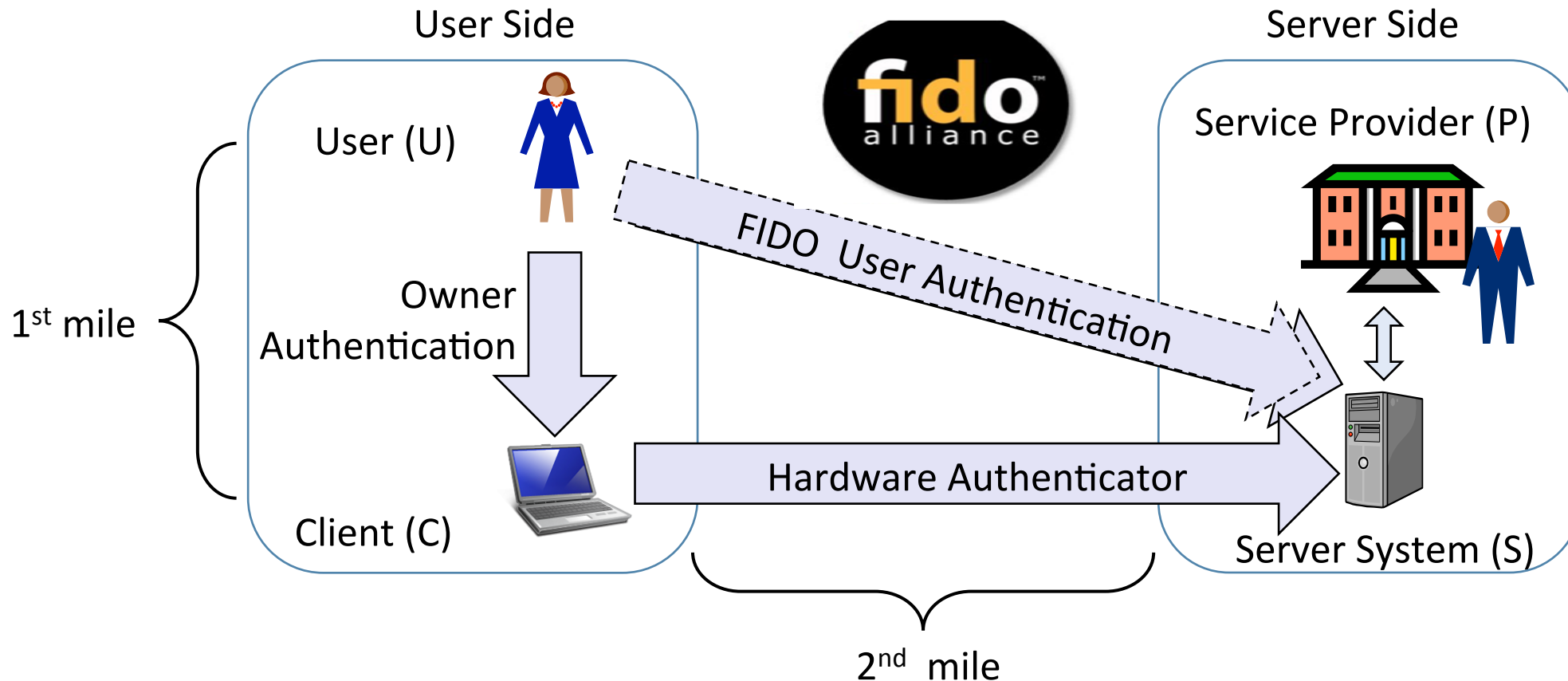
1. User requests access to e-Gov. SP
2. e-Gov. SP forwards authentication request to ID-porten, and displays ID-porten web-page with list of authentication providers.
3. User selects authentication provider.
4. User enters name and credentials in login form, which are sent for validation to authentication provider.
5. Authentication provider confirms authentic user and provides user attributes to ID-porten which forwards these to e-Gov. SP
6. Service Provider analyses user attributes and provides service according to policy



Classification of Identity Federations

	Centralized Identity	Distributed Identity
Centralized Authentication		
Distributed Authentication	 ID-porten/MinID Innlogging til offentlige tjenester	 

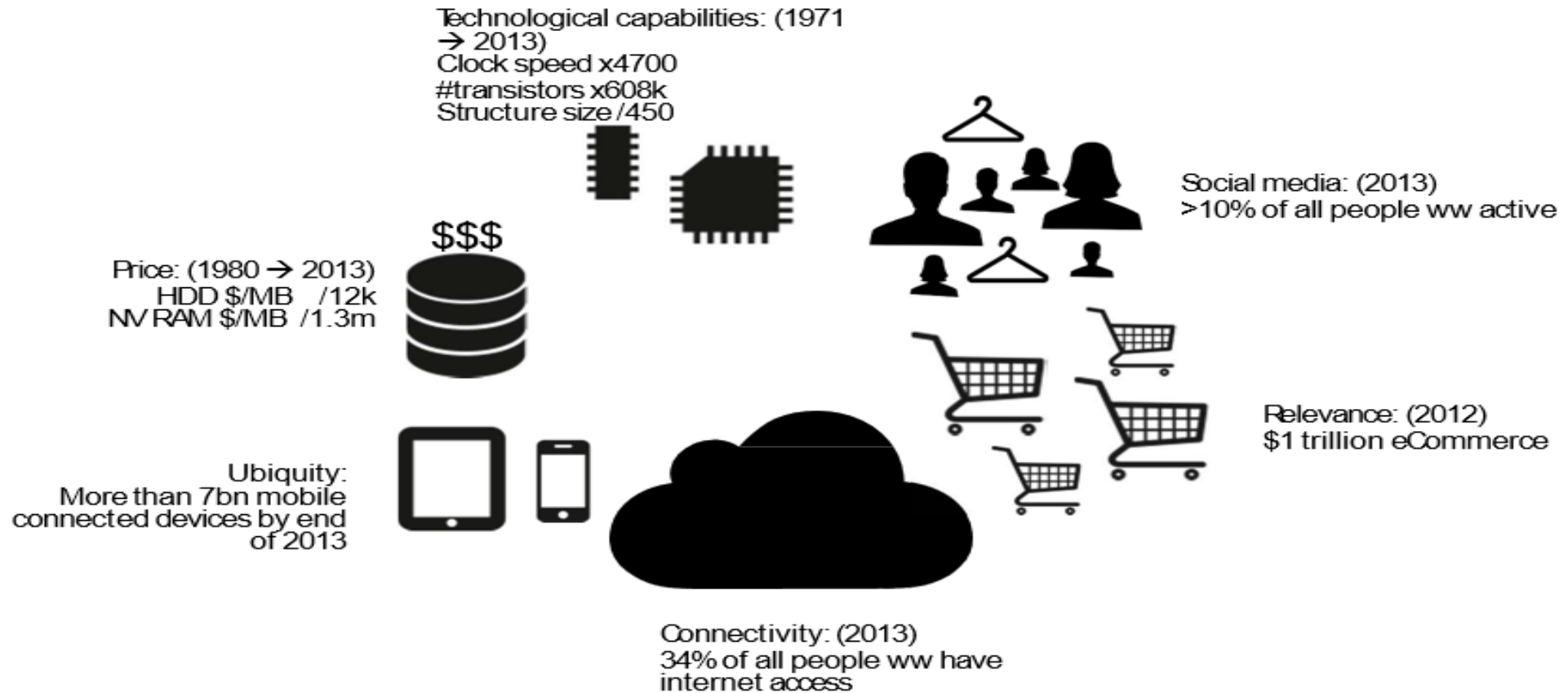
FIDO: Reusing owner authentication



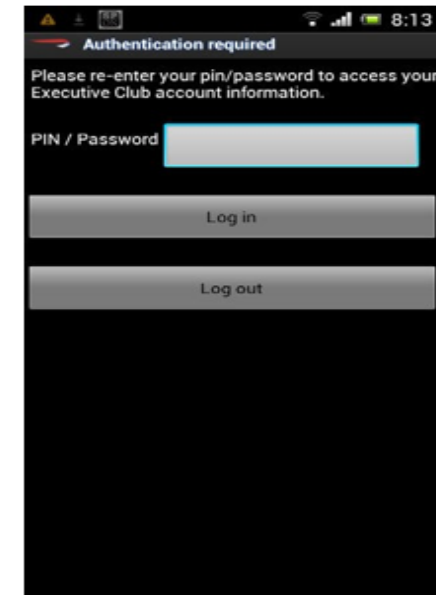
Fido Alliance

Simpler Stronger Authentication

IT has scaled!

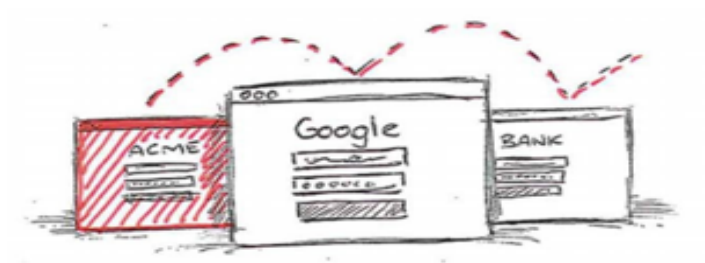


Authentication hasn't



Passwords?

Too many to remember, difficult to type,
and not secure



REUSED



PHISHED



KEYLOGGED

One Time Passwords

Improve security, but not easy to use



SMS USABILITY

Coverage | Delay | Cost



DEVICE USABILITY

One per site | Fragile



USER EXPERIENCE

User confusion



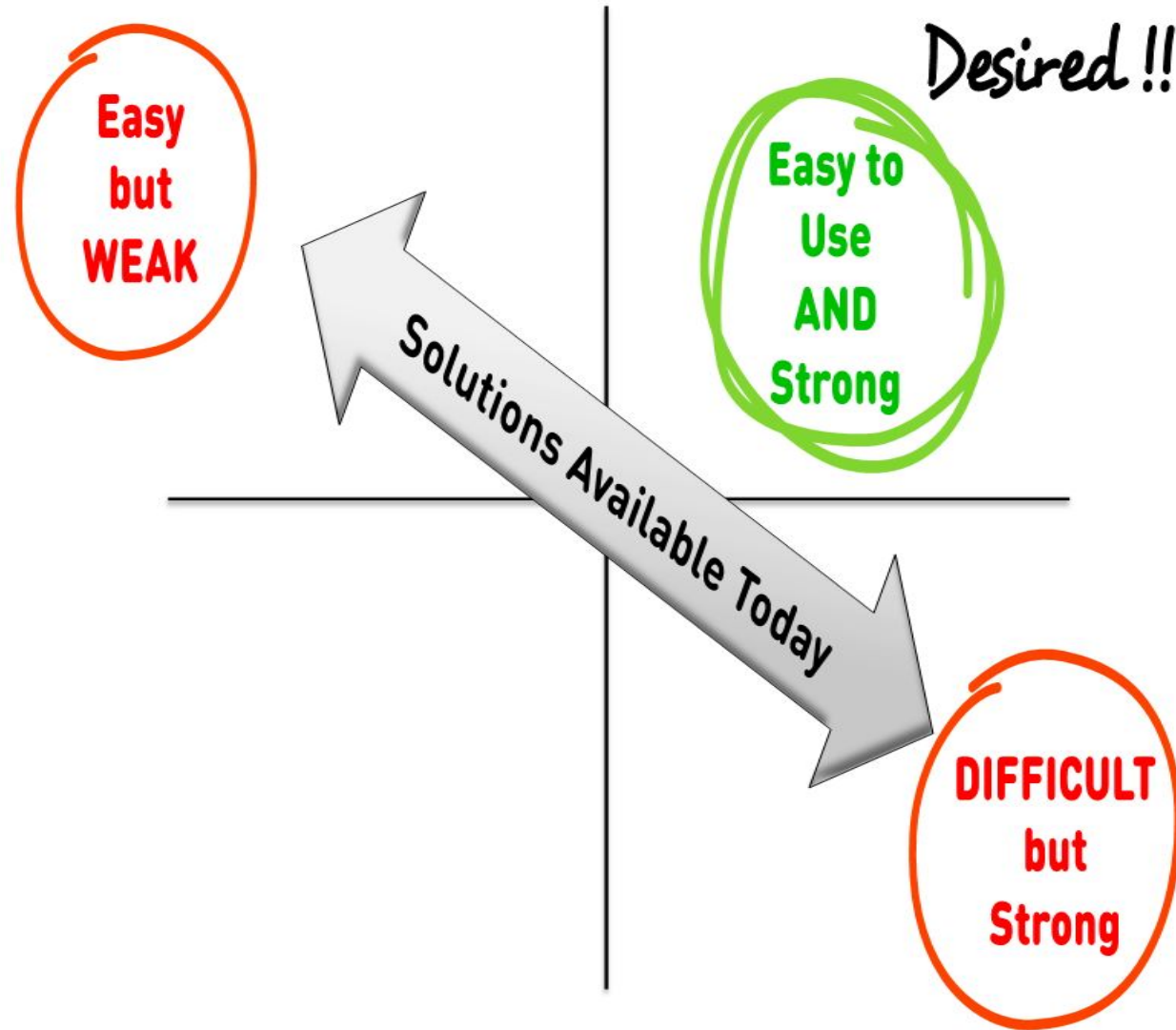
STILL PHISHABLE

Social engineering

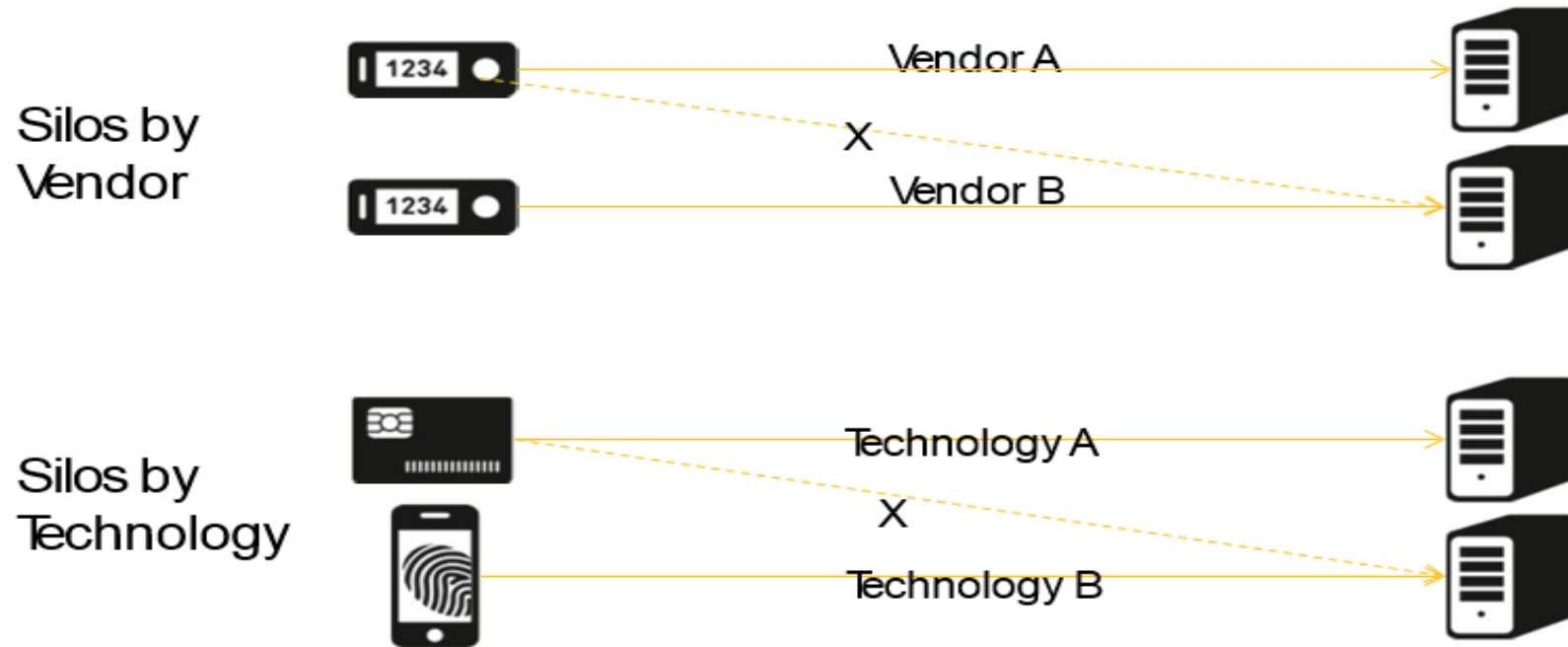
There are alternatives



TODAY'S AUTHENTICATION SOLUTIONS FALL SHORT



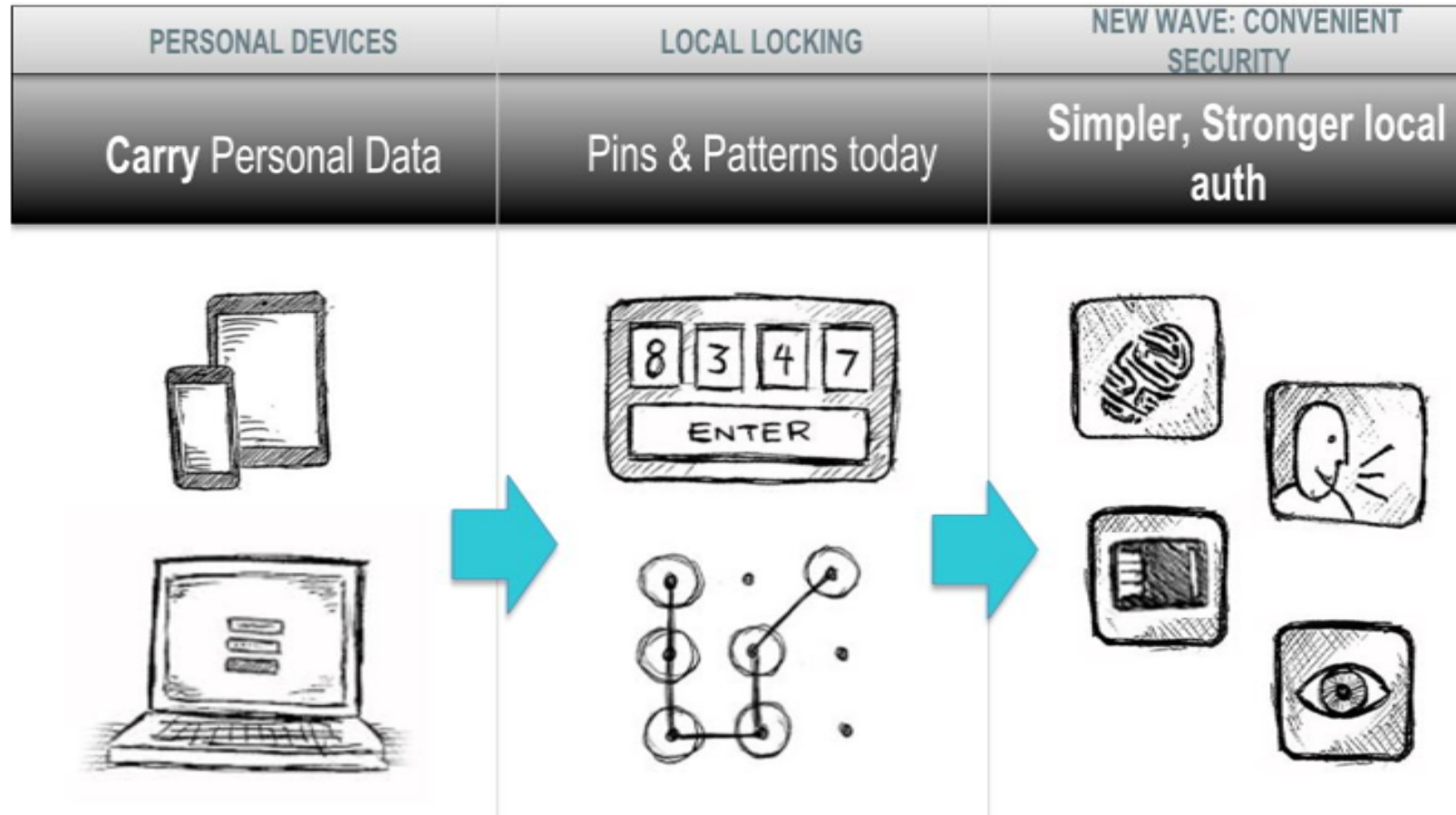
Implementation Is the challenge



Each new authentication solution requires new HW, SW, and Infrastructure.

➔ We're building 'Silos' of authentication

Megatrend



Putting It Together

The problem:

Simpler, Stronger online

The trend:

Simpler, Stronger local device auth

Why not:

Use local device auth for online auth?

This is the core idea behind FIDO standards!

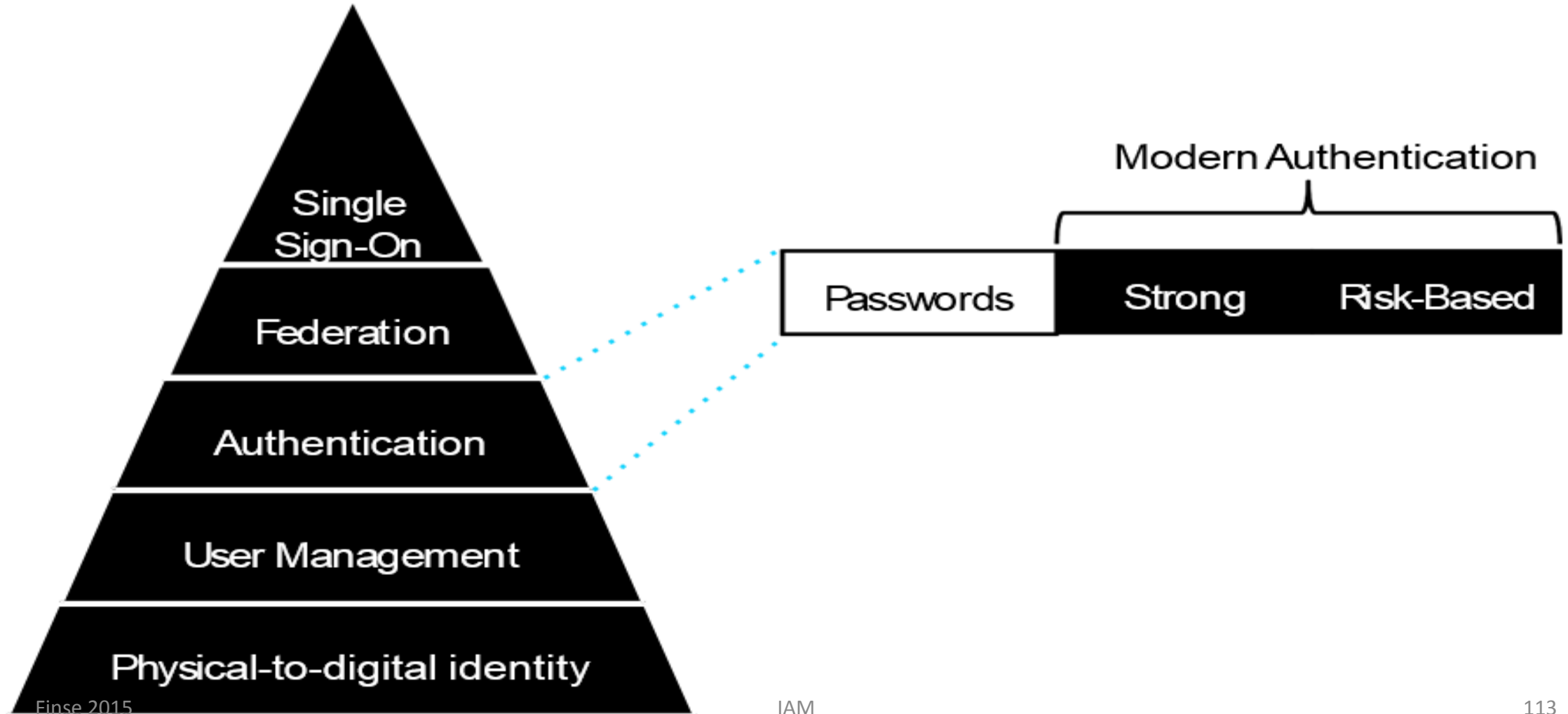
FIDO Goals

Enable online services and websites, whether on the open Internet or within enterprises, to leverage **native security features of end-user computing devices** for **strong** user authentication and to reduce the problems associated with creating and remembering many online credentials.

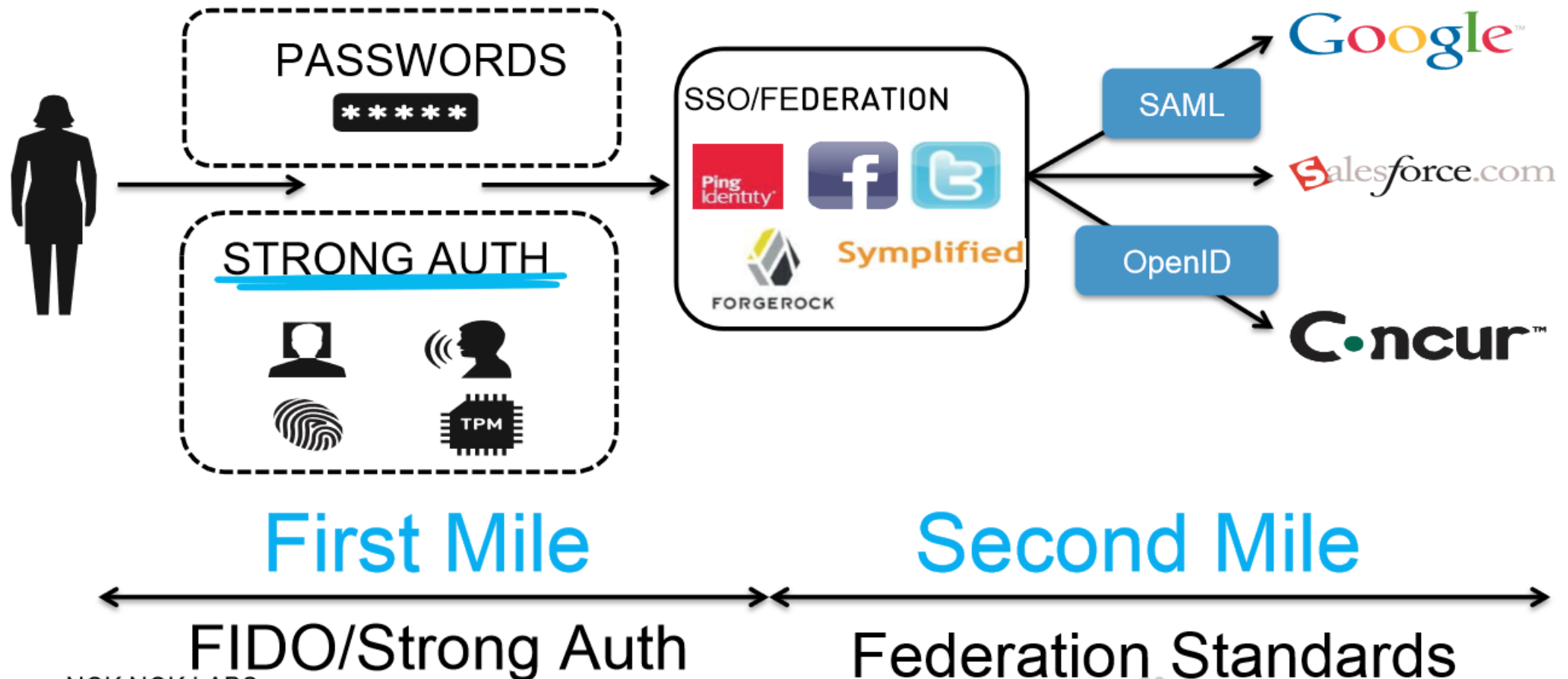
Design Considerations

- No 3rd Party in the Protocol
- No secrets on Server side
- Focus on User Privacy:
 - Biometric data never leaves user's device
 - No linkability between RPs
 - No linkability between RP accounts
- Embrace all kinds of Authenticators software, proprietary hardware, certified hardware, ...

What FIDO is not



Fido and federation



FIDO Experiences

ONLINE AUTH REQUEST

LOCAL DEVICE AUTH

SUCCESS

PASSWORDLESS EXPERIENCE (UAF standards)



Transaction Detail



Show a biometric



Done

SECOND FACTOR EXPERIENCE (U2F standards)



Finse 2015 Login & Password

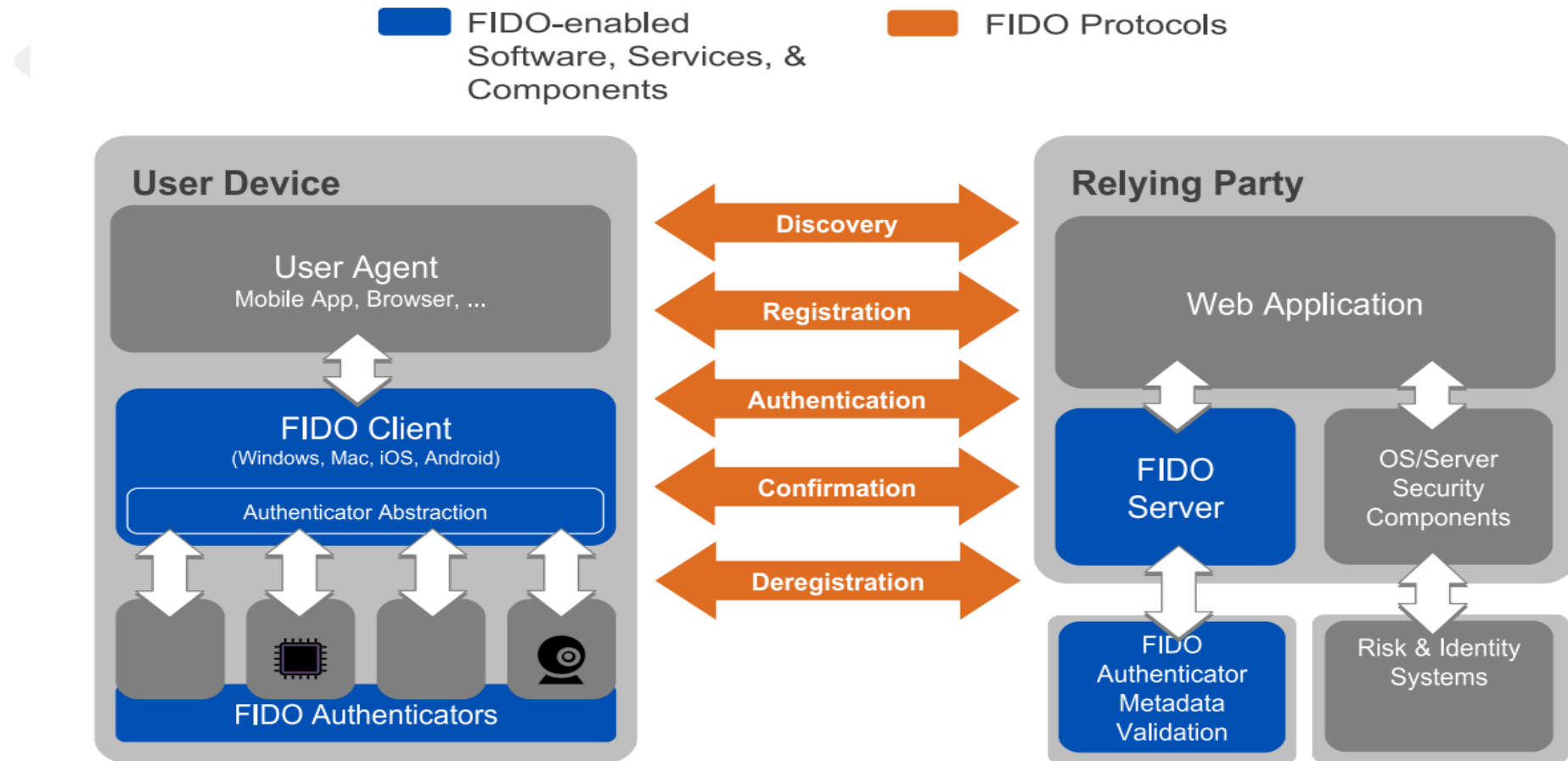


Insert Dongle, Press button

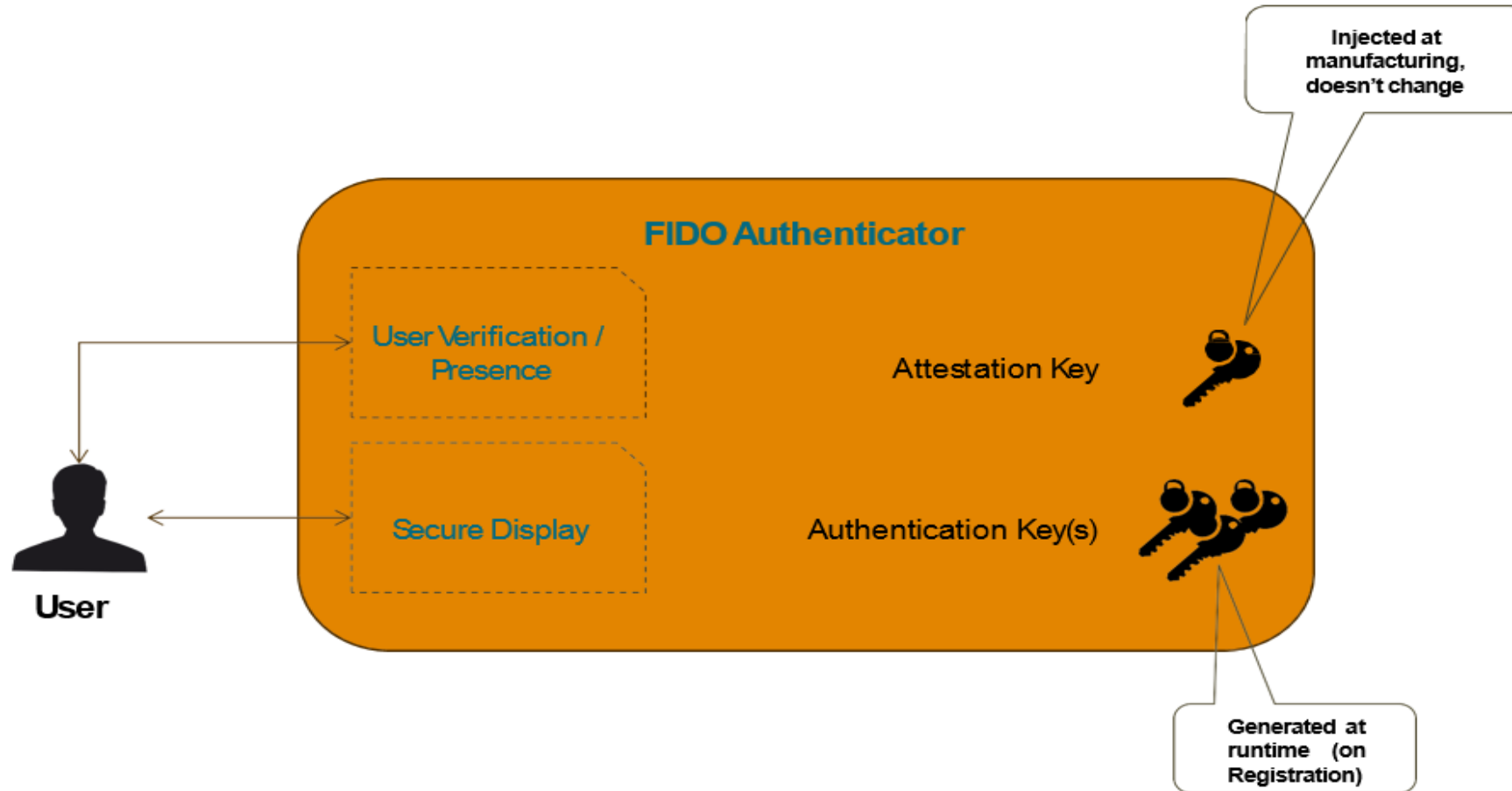


Done

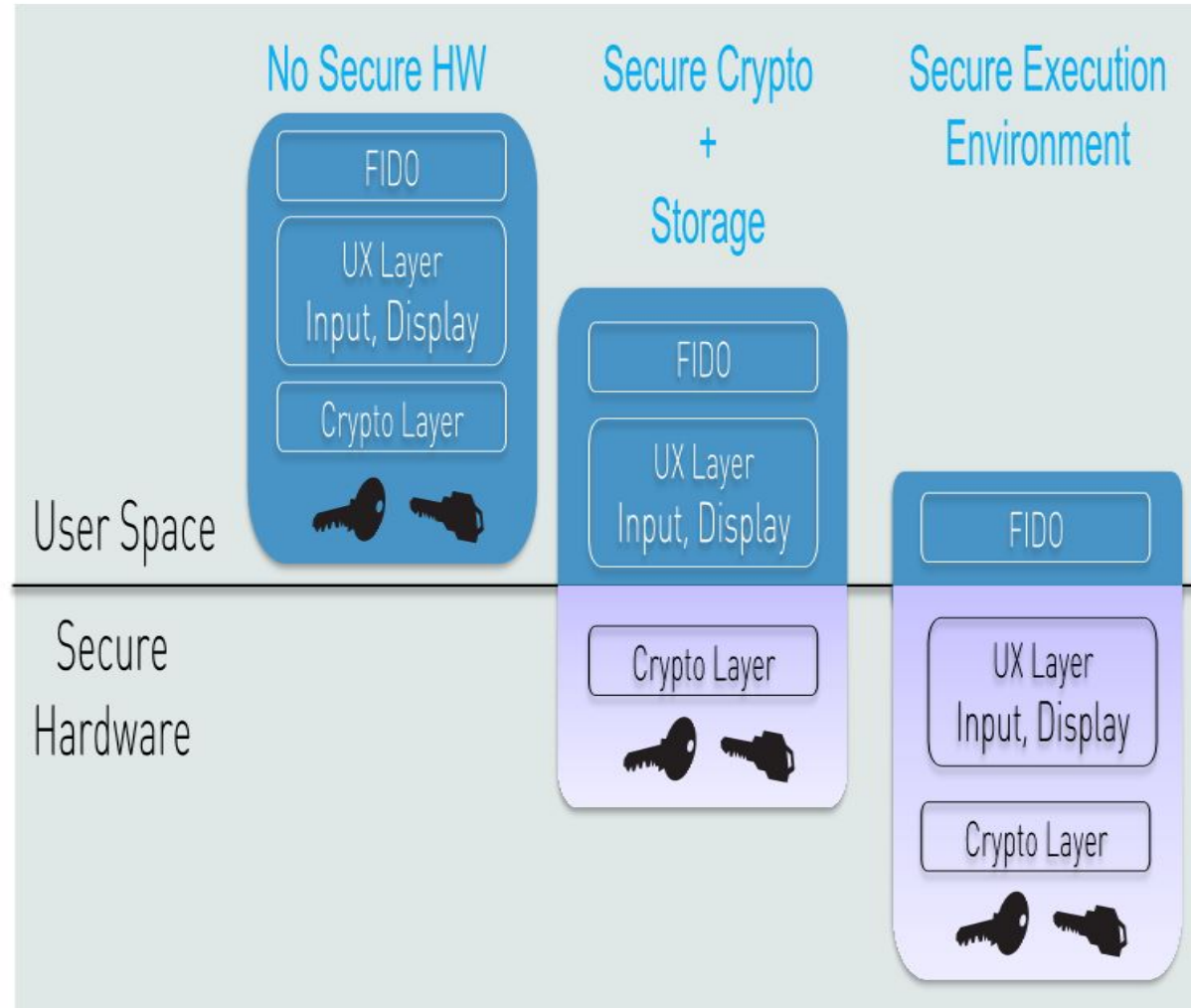
FIDO high level architecture



Authenticator concept



Choice of Security Profiles

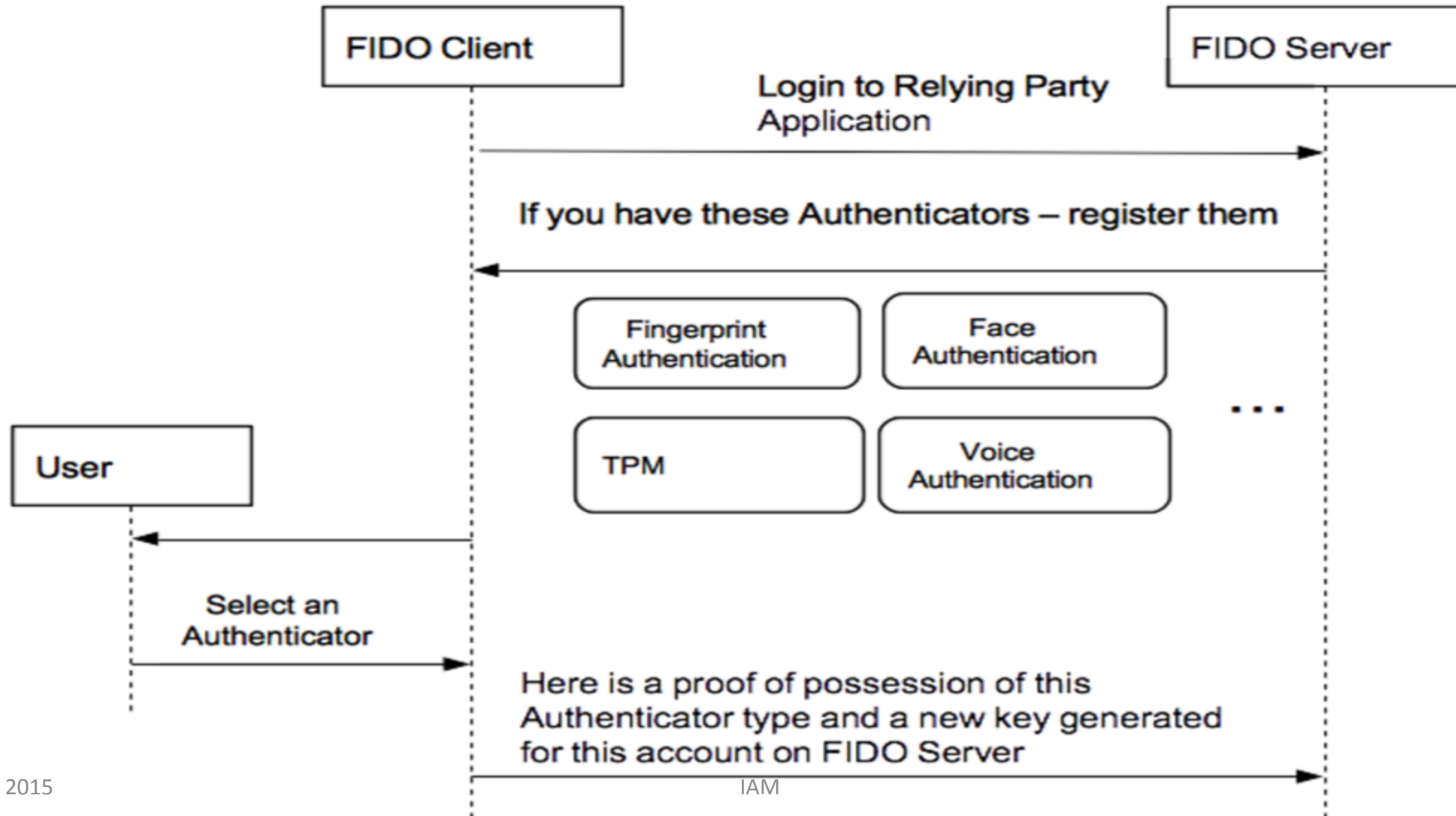


Universal Authentication Framework

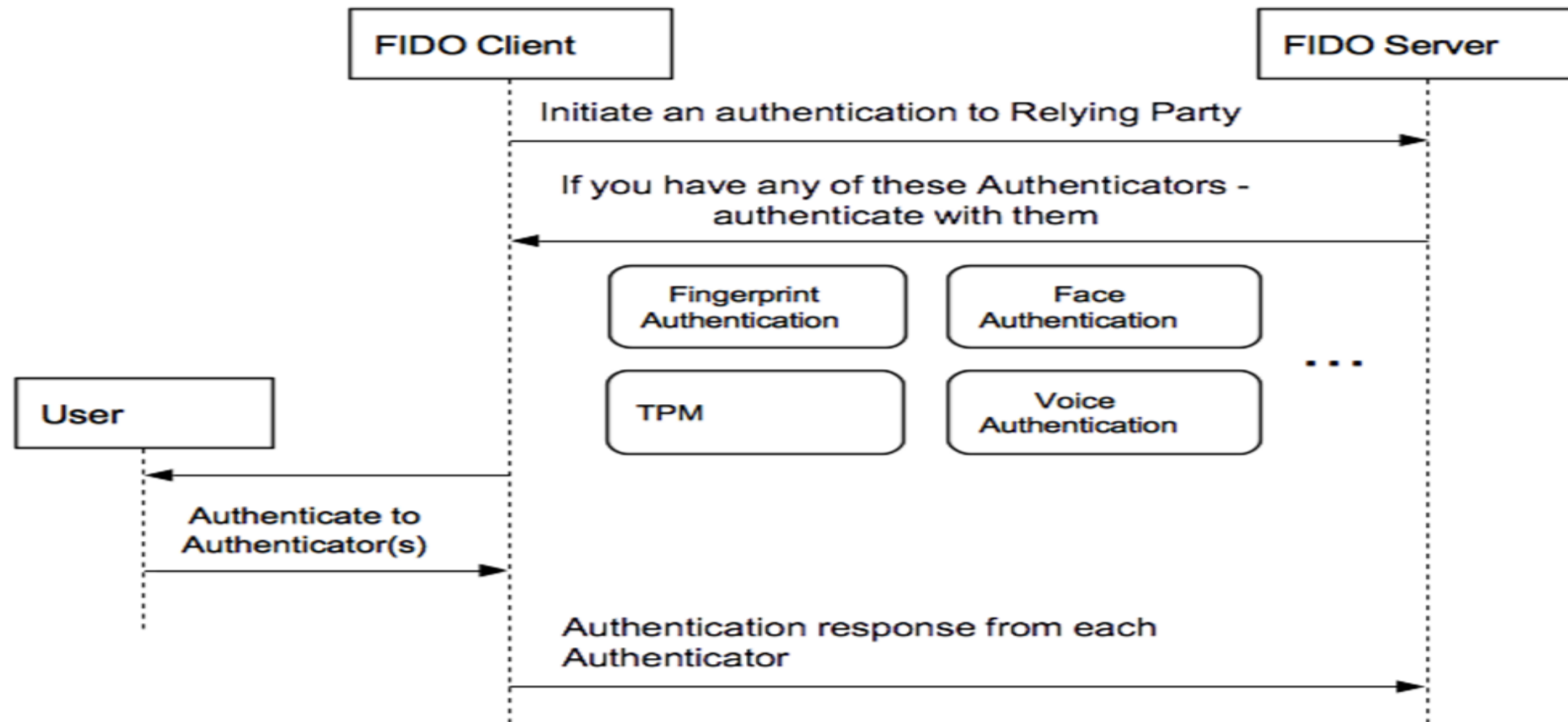
UAF

FIDO UAF authenticator acquisition and user enrollment

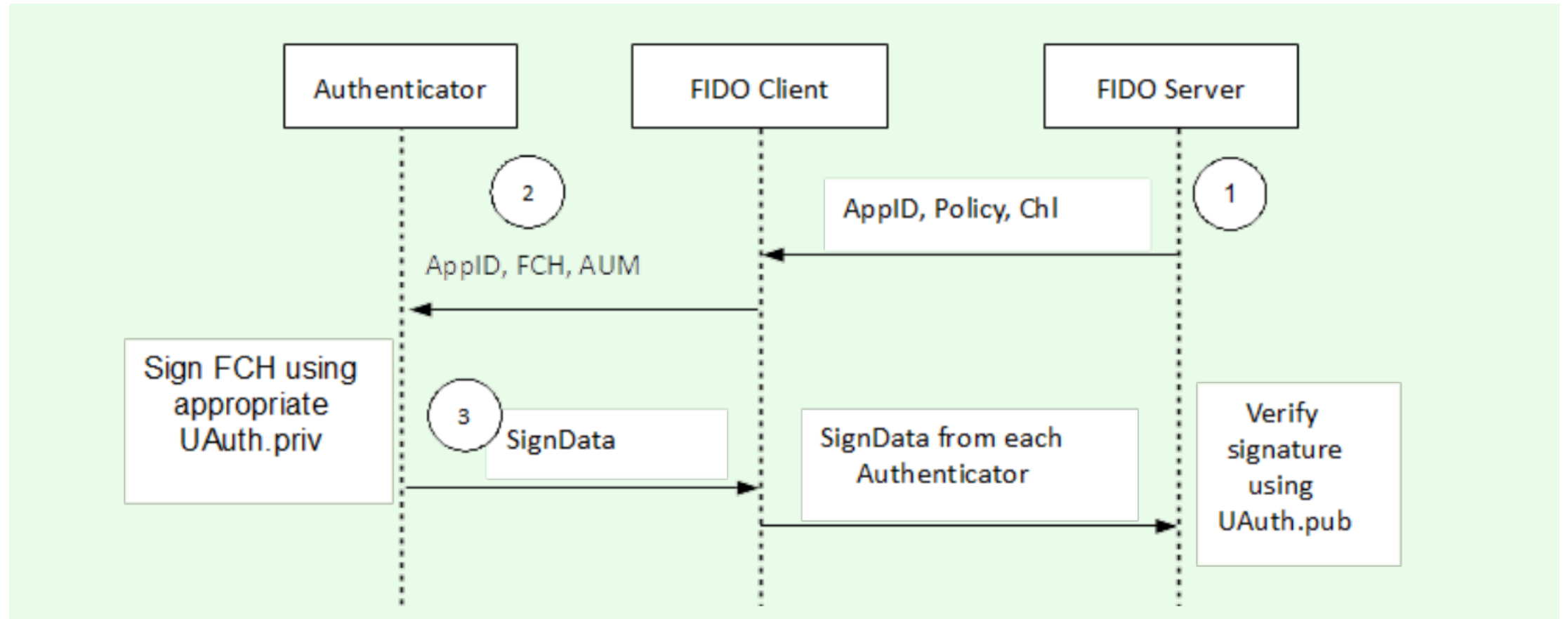
Authenticator Registration



UAF authentication: high level



UAF Authentication



Adoption of New Types of FIDO

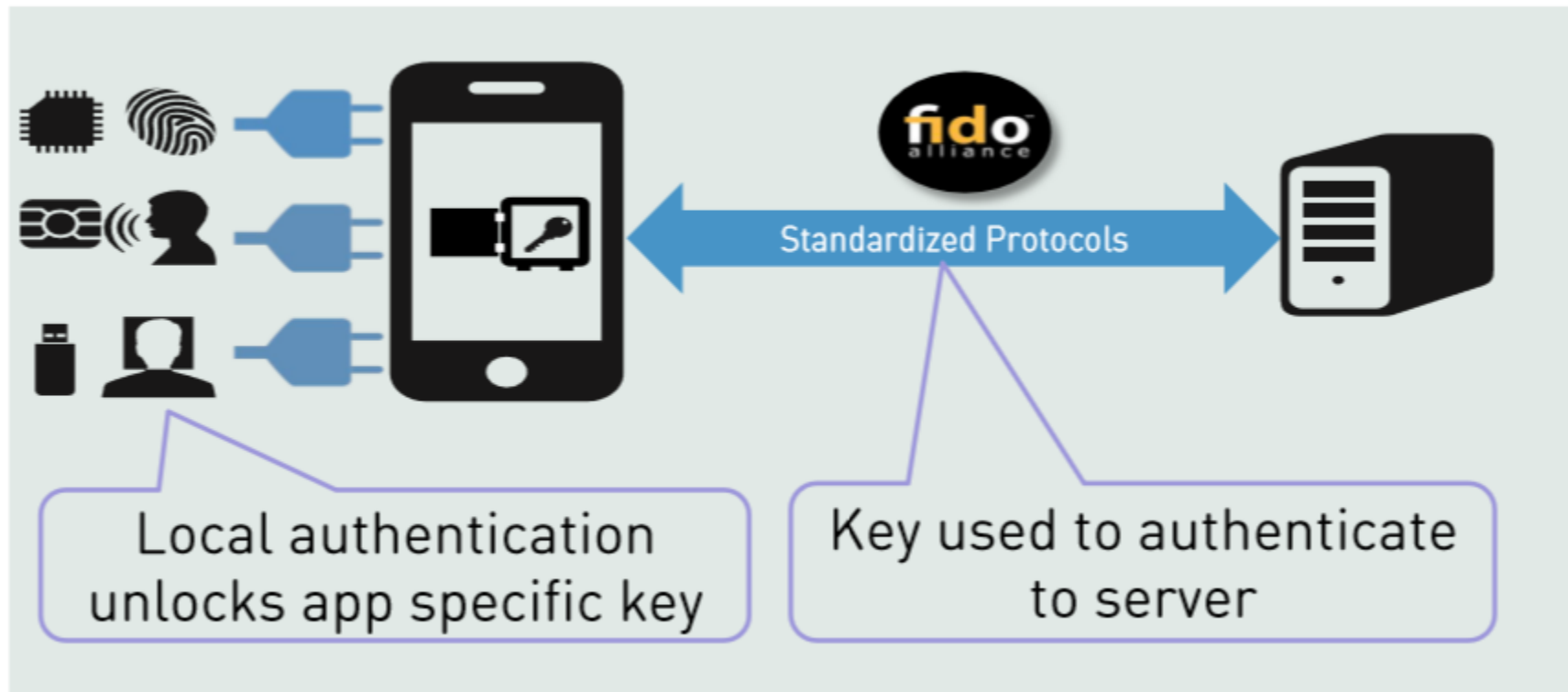
In order to support a new FIDO UAF Authenticator type:

- Relying Parties add a new entry to their configuration describing the new authenticator, along with its FIDO Attestation Certificate.

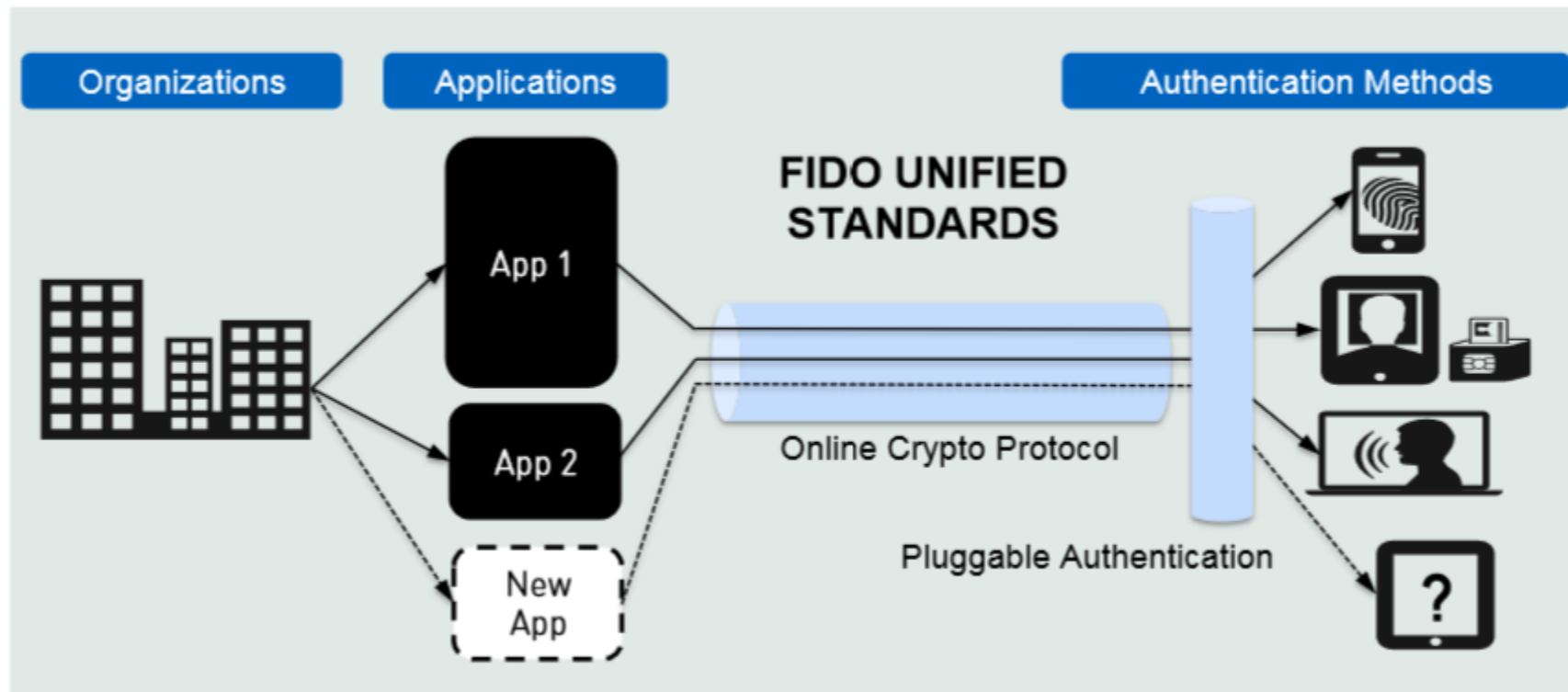
Privacy Considerations

- A UAF device does not have a global identifier visible across relying parties.
- A UAF device does not have a global identifier within a particular relying party.
- The UAF protocol generates unique asymmetric cryptographic key pairs **on a per-device, per-user account, and per-relying party basis.**
- The UAF protocol operations require minimal personal data collection: user verification is performed locally.

Fido and federation

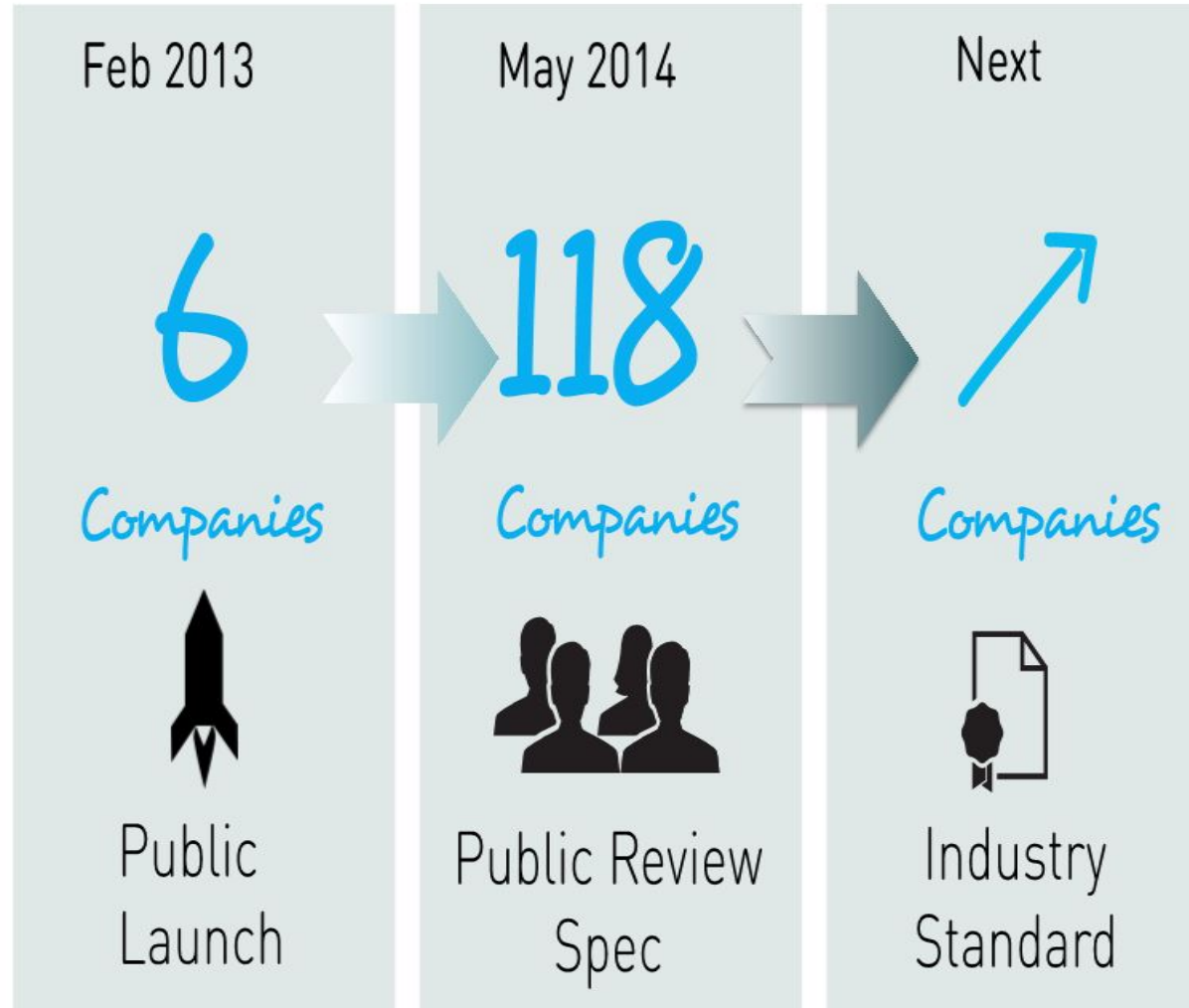


New authentication landscape



20

FIDO'S EXPLOSIVE GROWTH





FIDO: Advantages and disadvantages

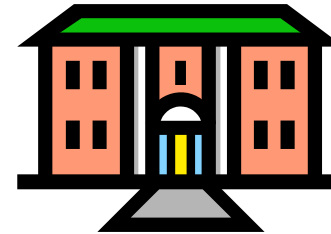
- End users:

- Better user experience?
- Client device compatibility.
- Will we trust it?



- Service providers:

- Flexible identification mechanisms.
- FIDO server configuration cost.
- Reduced IdMgt overhead?



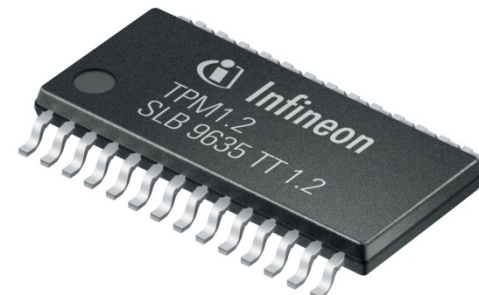
- Trust providers:

- Increased trust requirements.
- CAs will benefit (selling more certificates)

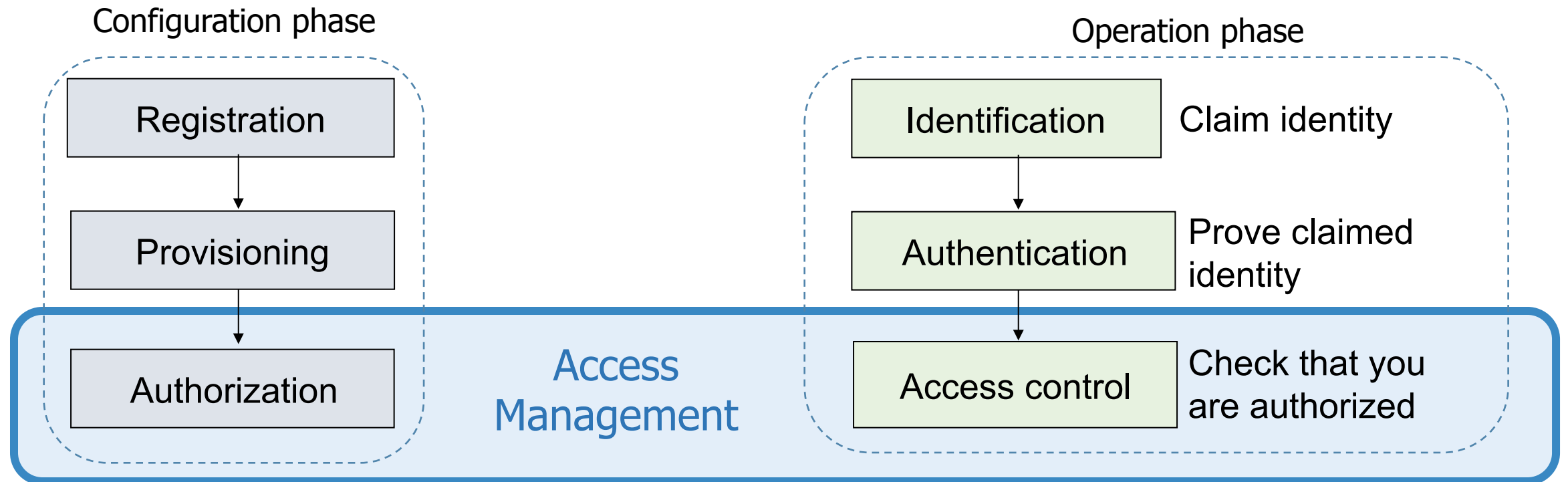


- Hardware manufacturers:

- Certificate management complexity.
- Increased liability.

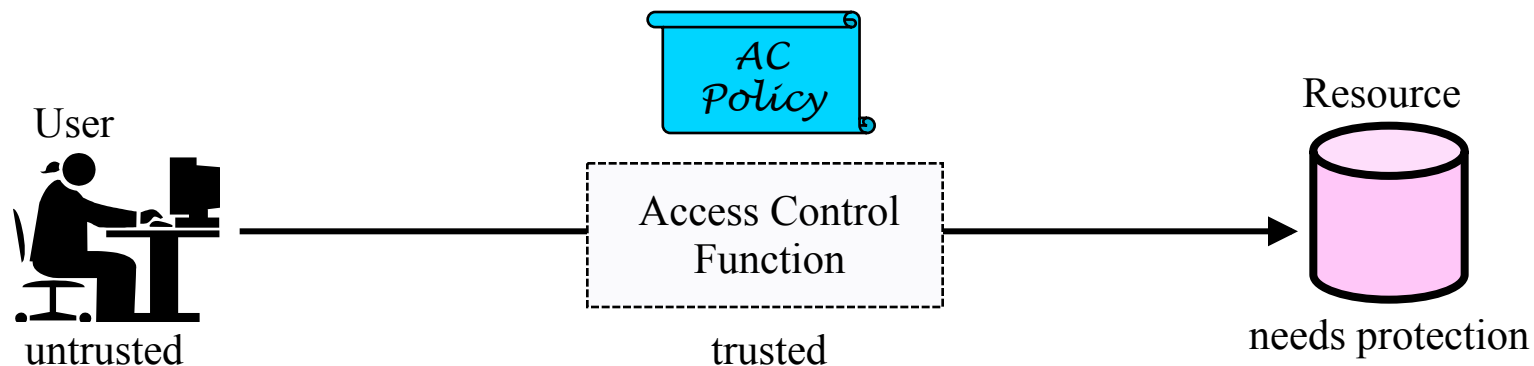


What is Access Management ?

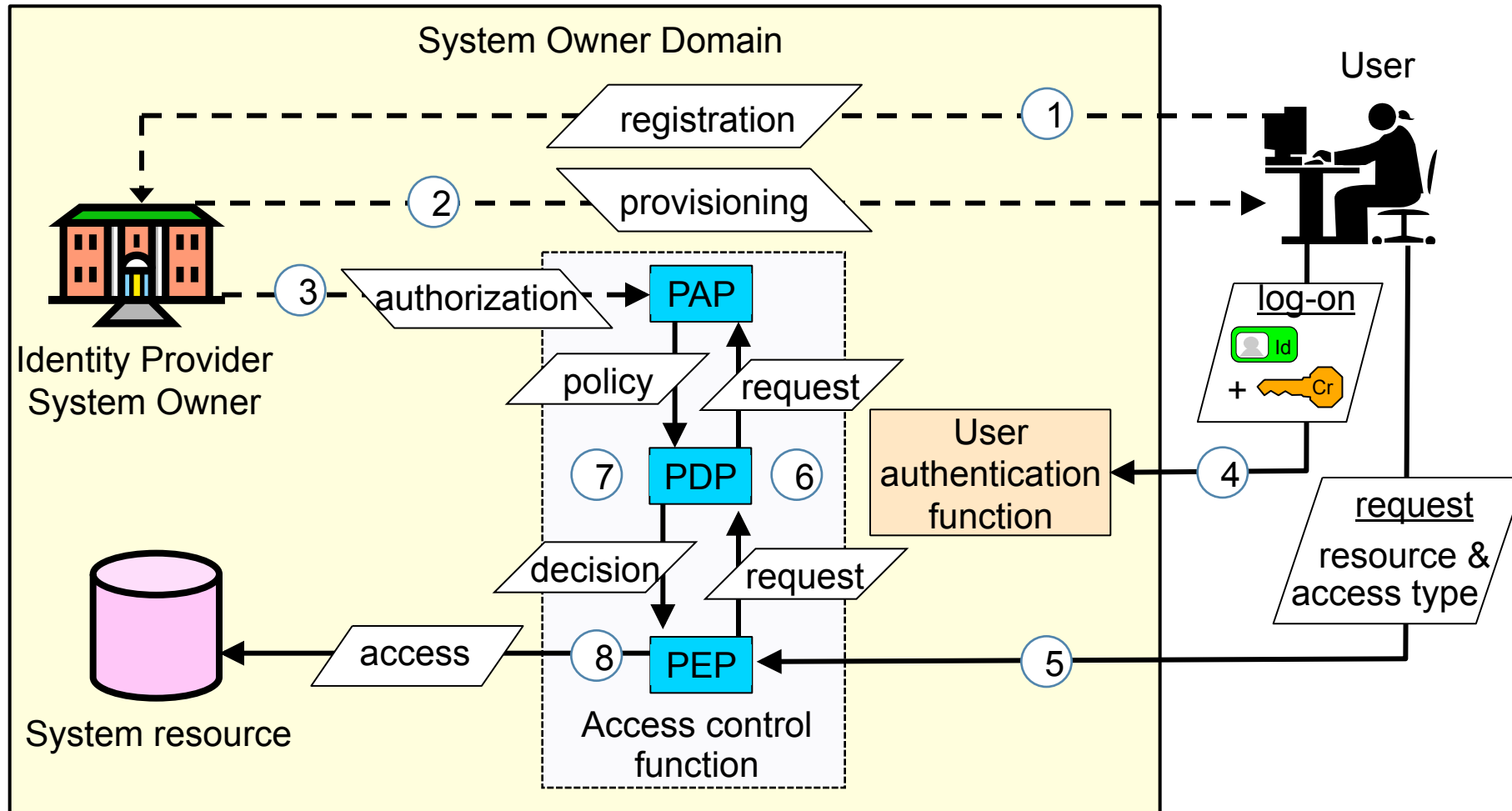


Access Control

- Assumes that users can not be trusted to follow security policy wrt. resource access
- AC functions enforce AC policies
- AC gives the ability to permit or deny the use of a particular resource by a particular entity



Access control concepts (abstract model)



Terminology

- **Subject/Principal**: active entity – user or process.
 - Subject often used for users
 - Principal often used for processes
- **Object**: passive entity – file or resource.
- **Access operations**: vary from basic memory access (read, write) to method calls in an object-oriented system.
- **Security Model** typically means AC model
 - i.e. a model/method for enforcing AC policies

Basic concepts

- Access control security models:
 - *How to define which subjects can access which objects with which access modes?*
- Three classical approaches
 - Discretionary Access Control (DAC)
 - Mandatory access control (MAC)
 - Role-Based Access Control (RBAC)
- Advanced approach for distributed environments:
 - Attribute-Based Access Control (ABAC)
 - Generalisation of DAC, MAC and RBAC

DAC / MAC According to the Orange Book (TCSEC)

TCSEC (1985) specifies two AC security models

- Discretionary AC (DAC)

- AC policy based on user identities
- e.g. *John has (r, w) - access to HR-files*

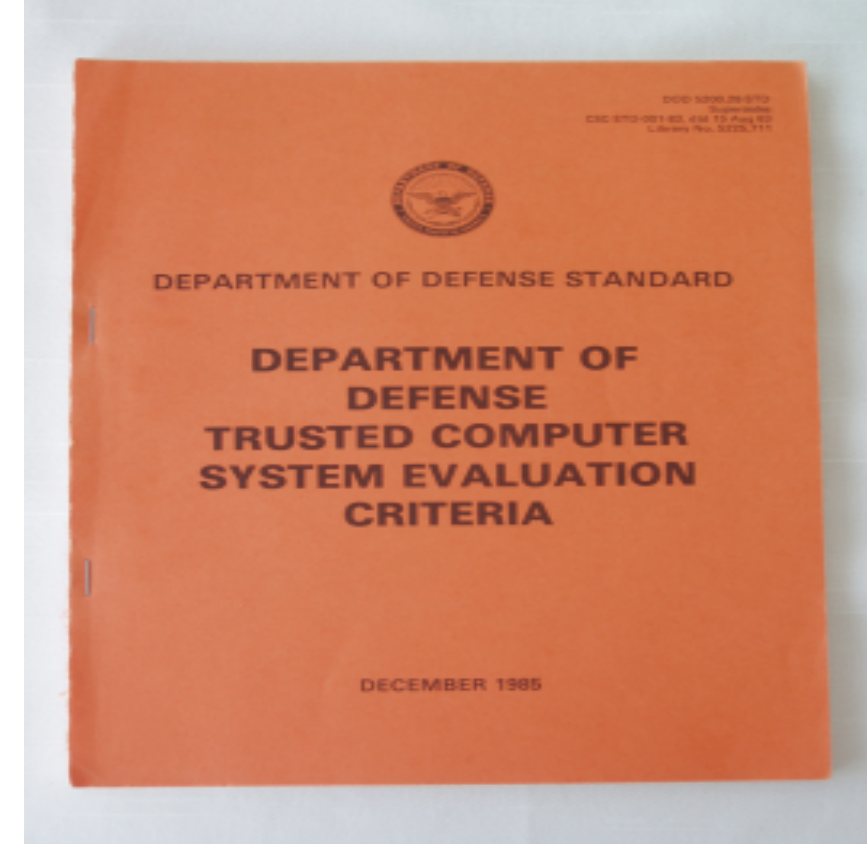
	HR	Sales
John	r, w	
Mary		r, w

- Mandatory AC (MAC)

- AC policy based on security labels
- e.g. *secret clearance needed for access*



Secret



Orange Book, 1985

DAC – Discretionary Access Control

- Access authorization is specified and enforced based on the identity of the user.
- DAC is typically implemented with ACL (Access Control Lists)
- DAC is discretionary in the sense that the owner of the resource can decide at his/her discretion who is authorized
- Operating systems using DAC:
 - Windows and Linux

DAC principles

- AC Matrix
 - General list of authorizations
 - Impractical, too many empty cells
- Access Control Lists (ACL)
 - Associated with an object
 - Represent columns from AC Matrix
 - Tells who can access the object

Columns→ ↓Rows		Objects			
		O1	O2	O3	O4
Subject names	S1	r,w	-	x	r
	S2	r	-	r	r,w
	S3	-	x	-	-
	S4	r,w	x	x	x

AC Matrix

- AC lists →

	O1
S1	r,w
S2	r
S3	-
S4	r,w

	O2
S1	-
S2	-
S3	x
S4	x

	O3
S1	x
S2	r
S3	-
S4	x

	O4
S1	r
S2	r,w
S3	-
S4	x

ACL in Unix

Each file and directory has an associated ACL

◆ Three access operations:

- read: from a file
- write: to a file
- execute: a file

◆ Access applied to a directory:

- read: list contents of dir
- write: create or rename files in dir
- execute: search directory

• Permission bits are grouped in three triples that define read, write, and execute access for **owner**, **group**, and **others**. 

• A '-' indicates that the specific access right is not granted.

• **rw-r--r--** means: read and write access for the owner, read access for group, and for others (world).

• **rw-x-----** means: read, write, and execute access for the owner, no rights for group and no rights for others

Capabilities

- Focus on the subjects:
 - access rights stored with subjects
 - Represents rows of AC Matrix
- Must be impossible for users to create fake capabilities
- Subjects may **grant** own capabilities to other subjects. Subjects may grant the right to grant rights.
- Challenges:
 - How to check who may access a specific object?
 - How to revoke a capability?
- Similar to SAML security token

AC
Capabilities
↓

	O1	O2	O3	O4
S1	r,w	-	x	r

	O1	O2	O3	O4
S2	r	-	r	r,w

	O1	O2	O3	O4
S3	-	x	-	-

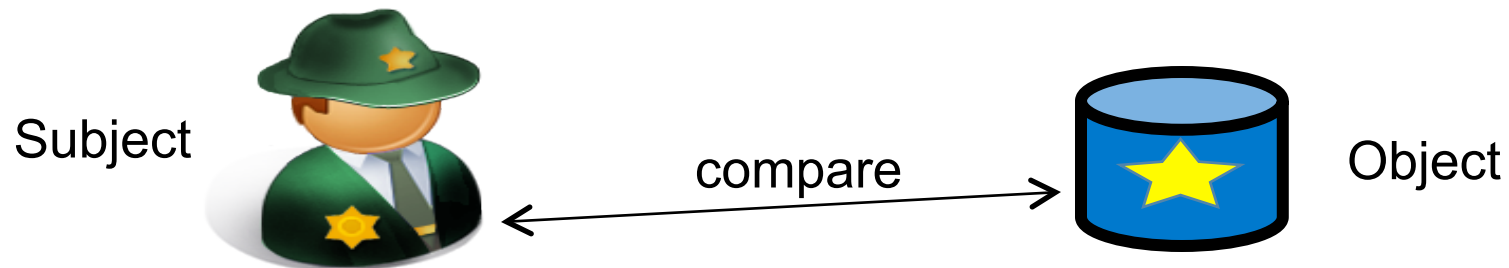
	O1	O2	O3	O4
S4	r,w	x	x	x

MAC – Mandatory Access Control

- Access authorization is specified and enforced with security labels
 - Security clearance for subjects
 - Classification levels for objects
- MAC compares subject and object labels
- MAC is mandatory in the sense that users do not control access to the resources they create.
- A system-wide set of **AC policy rules** for subjects and objects determine modes of access
- OS with MAC:
 - SE Linux supports MAC

MAC principles: Labels

- Security Labels can be assigned to subjects and objects
 - Can be strictly ordered security levels, e.g. “Confidential” or “Secret”
 - Can also be partially ordered categories, e.g. {Sales-dep, HR-dep}
- Dominance relationship between labels
 - $(L_A \geq L_B)$ means that label L_A dominates label L_B
- Object labels are assigned according to sensitivity
- Subject labels are determined by security clearance
- Access control decisions are made by comparing the subject label with the object label according to specific model
- MAC is typically based on Bell-LaPadula model (see later)



Bell-LaPadula: The classical MAC model

SS-property (Simple Security): No Read Up

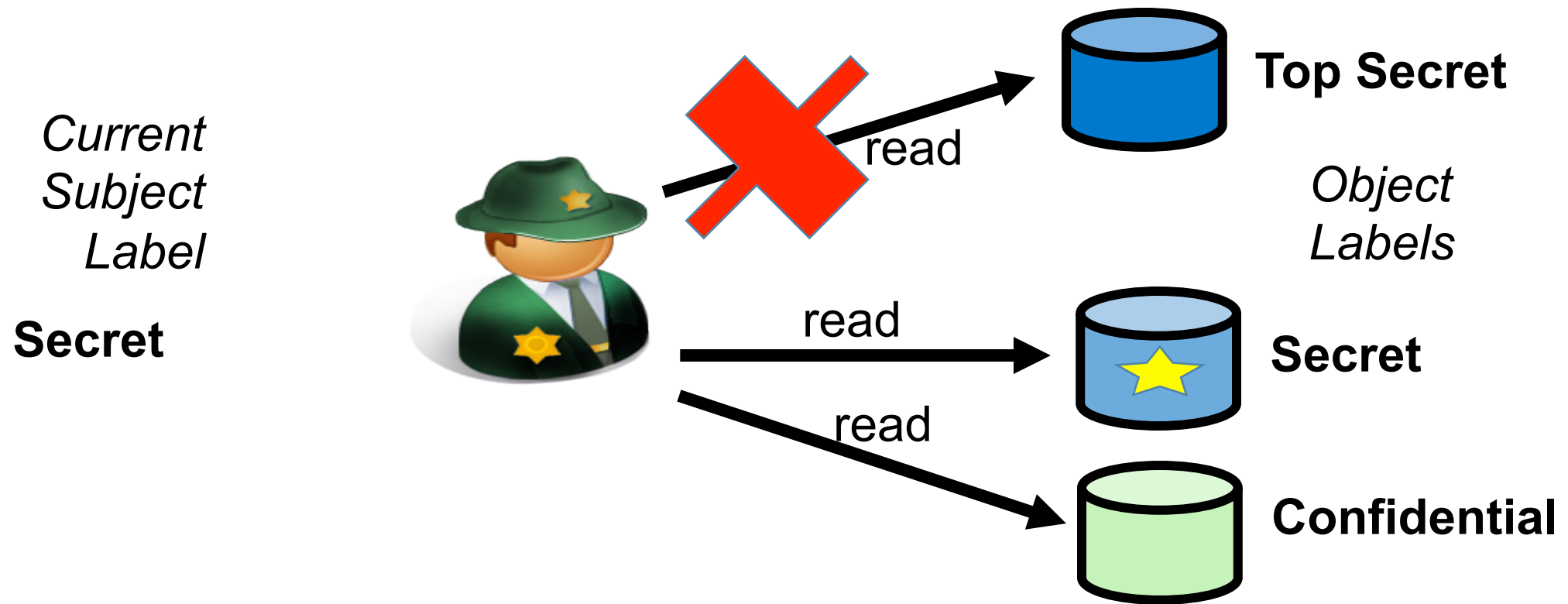
- A subject should not be able to read files with a higher label than its own label, because otherwise it could cause unauthorized disclosure of sensitive information.
- So you should only be able to read documents with an equal or lower label as your security clearance level.

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

- Subjects working on information/tasks at a given level should not be allowed to write to a lower level, because otherwise it could create unauthorized information flow.
- So you should only be able write to files with an equal or higher label as your security clearance level.

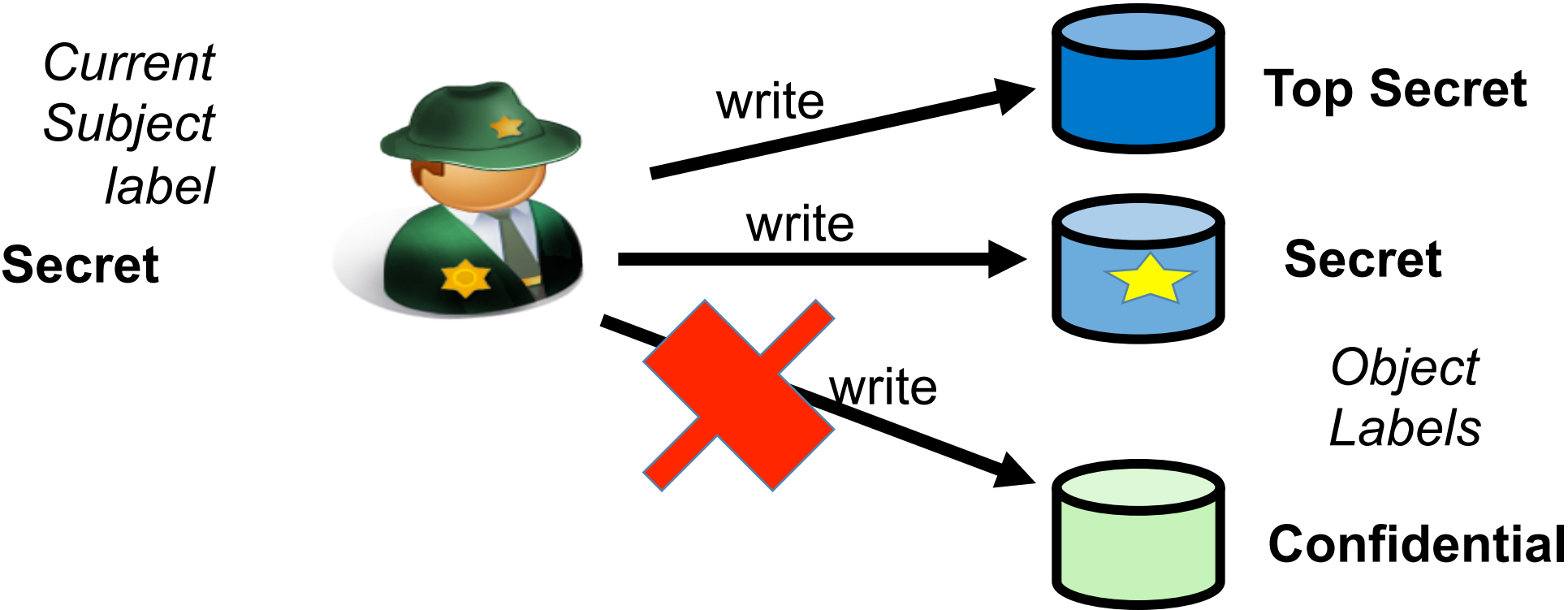
Bell-LaPadula (MAC model)

SS-Property: No Read Up



Bell-LaPadula (MAC model)

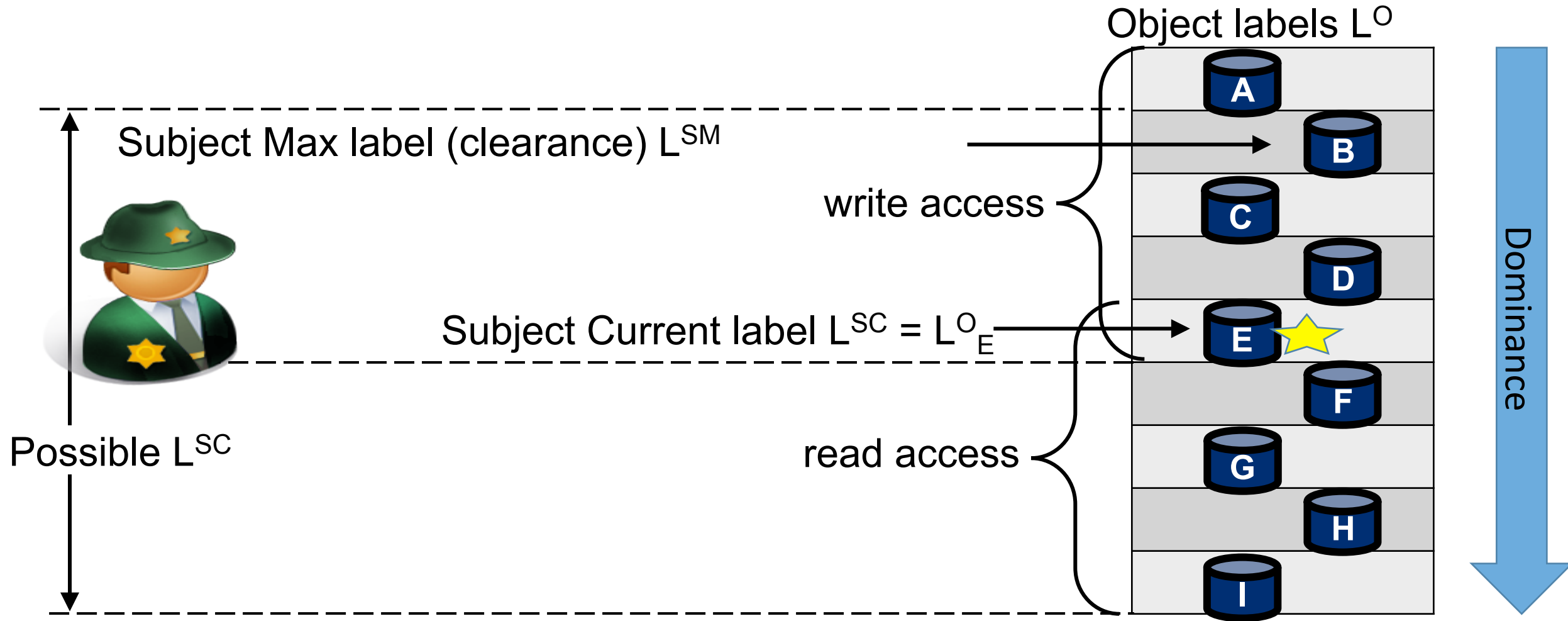
*-Property: No Write Down



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



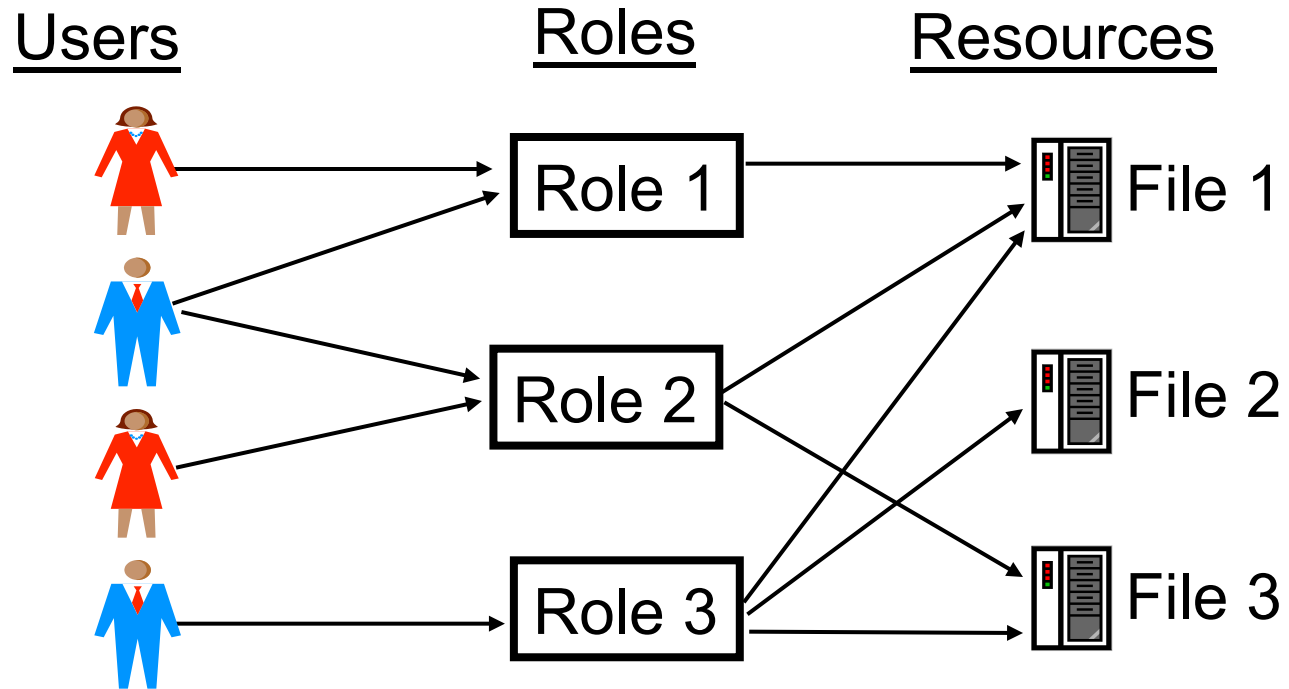
Combined MAC & DAC

- Combining access control approaches:
 - A combination of mandatory and discretionary access control approaches is often used
 - MAC is applied first,
 - DAC applied second after positive MAC
 - Access granted only if both MAC and DAC positive
 - Combined MAC/DAC 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

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 invoked 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.

ABAC and XACML

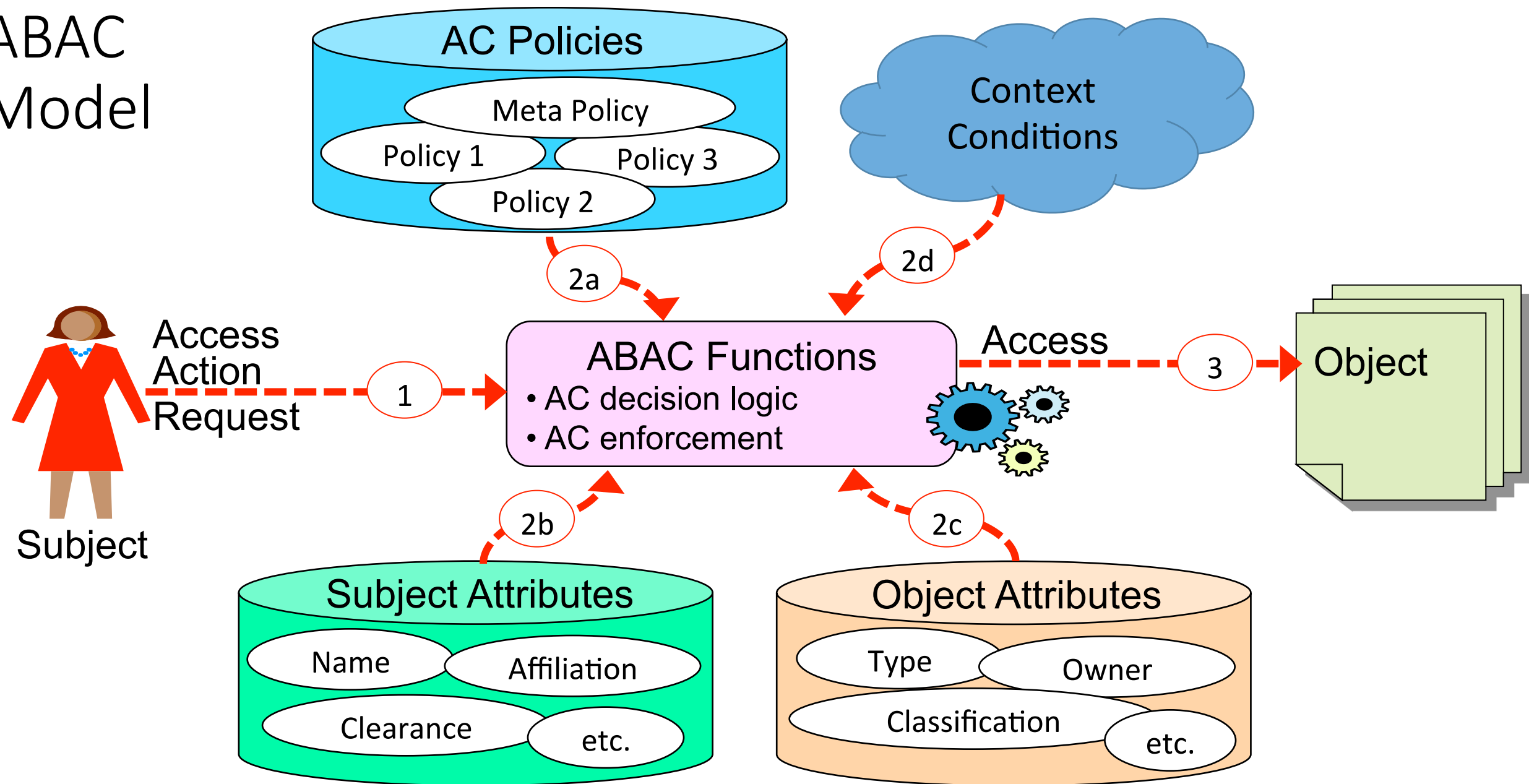
ABAC = Attribute Based Access Control

- ABAC specifies access authorizations and approves access through policies combined with attributes. The policy rules can apply to any type of attributes (user attributes, resource attribute, context attributed etc.).
- XACML used to express ABAC attributes and policies.

XACML = eXtensible Access Control Markup Language

- The XACML standard defines a language for expressing access control attributes and policies implemented in XML, and a processing model describing how to evaluate access requests according to the rules defined in policies.
- XACML attributes are typically structured in ontologies

ABAC Model



Attribute Based Access Control

- ABAC makes AC decisions based on Boolean conditions on attribute values.
- **Subject, Object, Context, and Action** consist of attributes
 - Subject attributes could be: Name, Sex, DOB, Role, etc.
 - Each attributes has a value, e.g.:
 - (Name (subject) = Alice), (Sex(subject) = F), (Role(subject) = HR-staff), (AccessType(action) = {read, write}), (Owner(object) = HR), (Type(object) = salary)
- The AC logic analyses all (attribute = value) tuples that are required by the relevant policy.
 - E.g. permit if:
[Role(subject) = HR-staff) and (AccessType(action) = read) and (Owner(object) = HR)] and (Time(query) = office-hours)]

Global Consistence

- ABAC systems require an XML terminology to express all possible attributes and their values,
- Must be consistent across the entire domain,
 - e.g. the attribute Role and all its possible values, e.g. (Role(subject) = HR-staff), must be known and interpreted by all systems in the AC security domain.
- Requires standardization:
 - e.g. for access to medical journals, medical terms must be interpreted in a consistent way by all systems
 - current international work on XML of medical terms
- Consistent interpretation of attributes and values is a major challenge for implementing ABAC.

ABAC: + and -

On the positive side:

- ABAC is much more flexible than DAC, MAC or RBAC
 - DAC, MAC and RBAC can be implemented with ABAC
- Can use any type of access policies combined with an unlimited number of attributes
- Suitable for access control in distributed environments
 - e.g. national e-health networks

On the negative side:

- Requires defining business concepts in terms of XML and ontologies which is much more complex than what is required in traditional DAC, MAC or RBAC systems.
- Political alignment and legal agreements required for ABAC in distributed environments

Meta-policies i.c.o. inconsistent policies

- Sub-domain authorities defined their own policies
- Potential for conflicting policies
 - E.g. two policies dictate different access decisions
- Meta-policy rules needed in case the ABAC logic detects policy rules that lead to opposite decisions
- Meta-policy takes priority over all other policies, e.g.
 - Meta-Policy Deny Overrides: If one policy denies access, but another policy approves access, then access is denied.
This is a conservative meta-policy.
 - Meta-Policy Approve Overrides: If one policy denies access, but another policy approves access, then access is approved.
 - This is a lenient meta-policy.

Research Challenges for IAM - (1)

- High trust requirements in IAM architectures
 - How to create sufficient trust ?
 - Is PKI enough ?
- Trust negotiations
 - Some identity players have gained high trust.
 - How can existing trust be leveraged, e.g. for service access, in business, etc. ?
- Advanced IAM architectures use complex protocols.
 - Are there vulnerabilities to be discovered ?
 - IAM protocol verification is a challenge.
- IAM can be both a strength and a threat to privacy.
 - How can IAM architectures be used for intelligence and information warfare ?
- The role of SIM/IMSI and of the TPM

Research Challenges for IAM - (2)

- Robust biometrics.
 - High circumvention resistance is crucial for high assurance
 - Tamper proof hardware integration of biometric sensors
- Liveness detection.
 - E-Government services need assurance about persons being alive
 - How can robust biometrics be used for liveness proof ?
- Private ID vs. Professional ID .
 - Often problematic to use private ID in professional work roles, e.g. In e-health.
 - How to create Professional IDs with high trust, that protect Private IDs
- Quo vadis FIDO ?
 - FIDO is a completely new and radical IAM architecture.
 - Urgent need to analyse strengths and vulnerabilities.