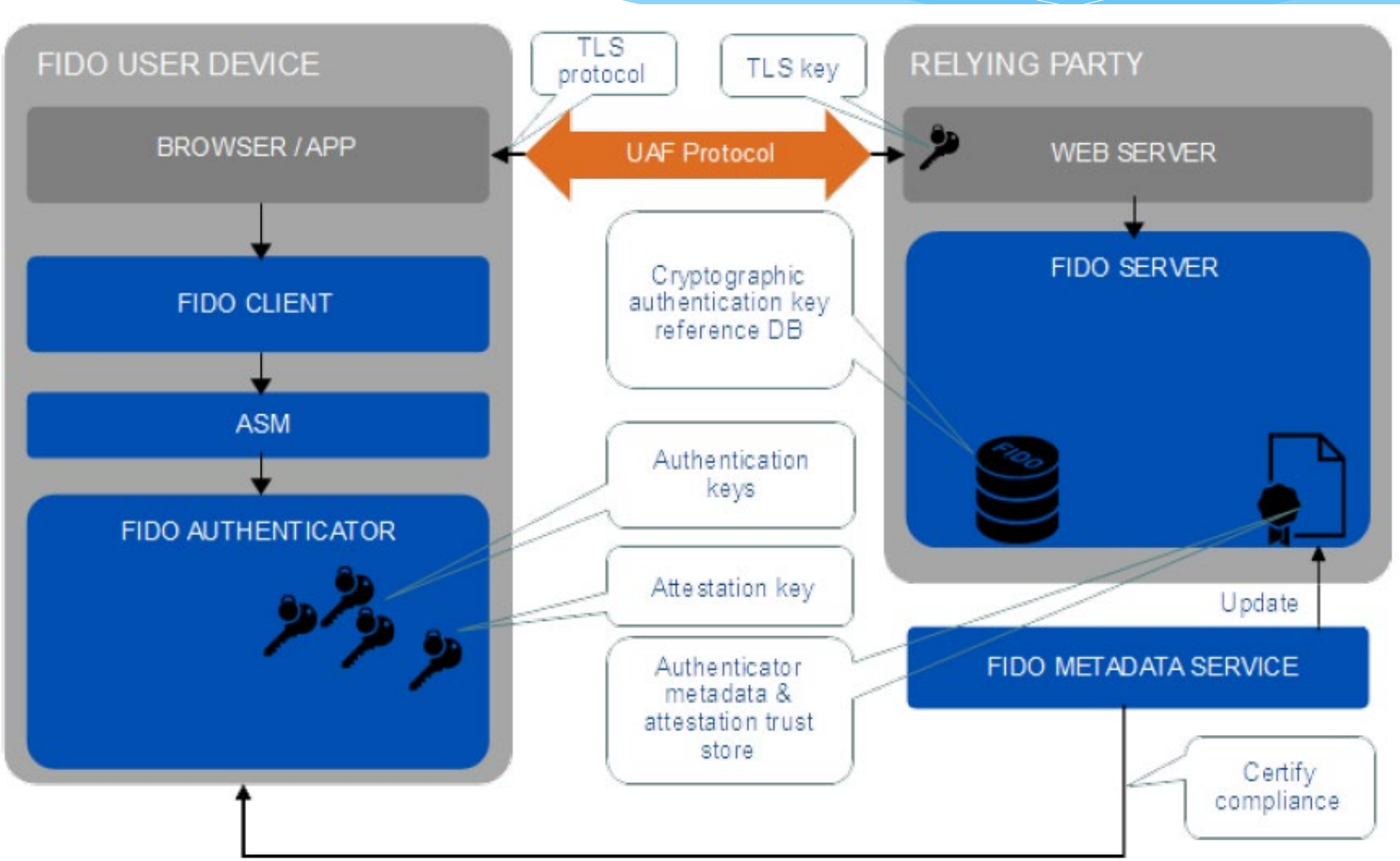
Segurança de Sistemas e dados (MSI 2020/2021)

Aula 4

Rolando Martins
DCC – FCUP
Slides Adaptados do Prof. Manuel Eduardo Correia

FIDO UAF (Universal Authentication Framework) High-Level Architecture



FIDO UAF Client

* A FIDO UAF Client implements the client side of the FIDO UAF protocols, and is responsible for:
  * Interacting with specific FIDO UAF Authenticators using the FIDO UAF Authenticator Abstraction layer (**ASM – Authenticator Specific Module**) via the FIDO UAF Authenticator API.
  * Interacting with a user agent on the device (e.g. a mobile app, browser) using user agent-specific interfaces to communicate with the FIDO UAF Server.
    * For example, a FIDO-specific browser plugin would use existing browser plugin interfaces or a mobile app may use a FIDO specific SDK. The user agent is then responsible for communicating FIDO UAF messages to a FIDO UAF Server at a Relying Party.

FIDO UAF Server

* A FIDO UAF server implements the server side of the FIDO UAF protocols and is responsible for:
  * Interacting with the Relying Party web server to communicate FIDO UAF protocol messages to a FIDO UAF Client via a device user agent.
  * Validating FIDO UAF authenticator attestations against the configured authenticator metadata to ensure only trusted authenticators are registered for use.
  * Manage the association of registered FIDO UAF Authenticators to user accounts at the Relying Party.
  * Evaluating user authentication and transaction confirmation responses to determine their validity.

The FIDO UAF server is conceived as being deployable as an on-premise server by Relying Parties or as being outsourced to a FIDO-enabled third-party service provider.

https://fidoalliance.org/developers/

FIDO UAF Protocols - **Authenticator Registration**

* The FIDO UAF protocols carry FIDO UAF messages between user devices and Relying Parties.
* There are protocol messages addressing:
  * Authenticator Registration:
    * ***Discover the FIDO UAF Authenticators available*** on a user's system or device
    * ***Verify attestation assertions made by the FIDO UAF Authenticators***
      * to ensure the authenticator is authentic and trusted. Verification occurs using the attestation public key certificates distributed via authenticator metadata
    * ***Register the authenticator and associate it with the user's account*** at the Relying Party.
      * Once an authenticator attestation has been validated, the Relying Party can provide a unique secure identifier that is specific to the Relying Party and the FIDO UAF Authenticator.

FIDO UAF Protocols – **Authentication, Confirmation, Deregistration**

- ***User Authentication*:**
  - Authentication is typically based on cryptographic challenge-response authentication protocols and will facilitate user choice regarding which FIDO UAF Authenticators are employed in an authentication event.
- **Secure Transaction Confirmation:**
  - A Relying Party can present the user with a secure message for confirmation.
    - could be used in a variety of contexts such as confirming a financial transaction, a user agreement ,or releasing patient records.
- **Authenticator Deregistration:**
  - Deregistration is typically required when the user account is removed at the Relying Party. The Relying Party can trigger the deregistration by requesting the Authenticator to delete the associated UAF credential with the user account.

FIDO UAF Authenticator Abstraction Layer

* The FIDO UAF Authenticator Abstraction Layer provides a uniform API to FIDO Clients enabling the use of authenticator-based cryptographic services for FIDO-supported operations.
* It provides a uniform lower-layer "authenticator plugin" API facilitating the **deployment of multi-vendor** FIDO UAF Authenticators and their requisite drivers.
* Web application interact with this abstraction layer by the means of **a well defined Javascript API/Library.**
  * FIDO U2F Javascript API
    * https://fidoalliance.org/specs/fido-u2f-javascript-api-v1.0-rd-20140209.pdf
  * WebAuthn for FIDO2 (W3C Candidate Recommendation)
    * https://www.w3.org/TR/webauthn/

FIDO UAF Authenticator

* A FIDO UAF Authenticator **is a secure entity, connected to or housed within FIDO user devices, that can create key material associated to a Relying Party**.
* The key can then be used to participate in FIDO UAF strong authentication protocols.
  * For example, the FIDO UAF Authenticator can provide a response to a cryptographic challenge using the key material thus authenticating itself to the Relying Party.
* In order to meet the goal of simplifying integration of trusted authentication capabilities, a FIDO UAF Authenticator **will be able to attest to its particular type**
  * (e.g., **biometric**) and capabilities (e.g., supported crypto algorithms), as well as to its provenance.
  * This provides a Relying Party with a high degree of confidence that the user being authenticated is indeed the user that originally registered with the site.

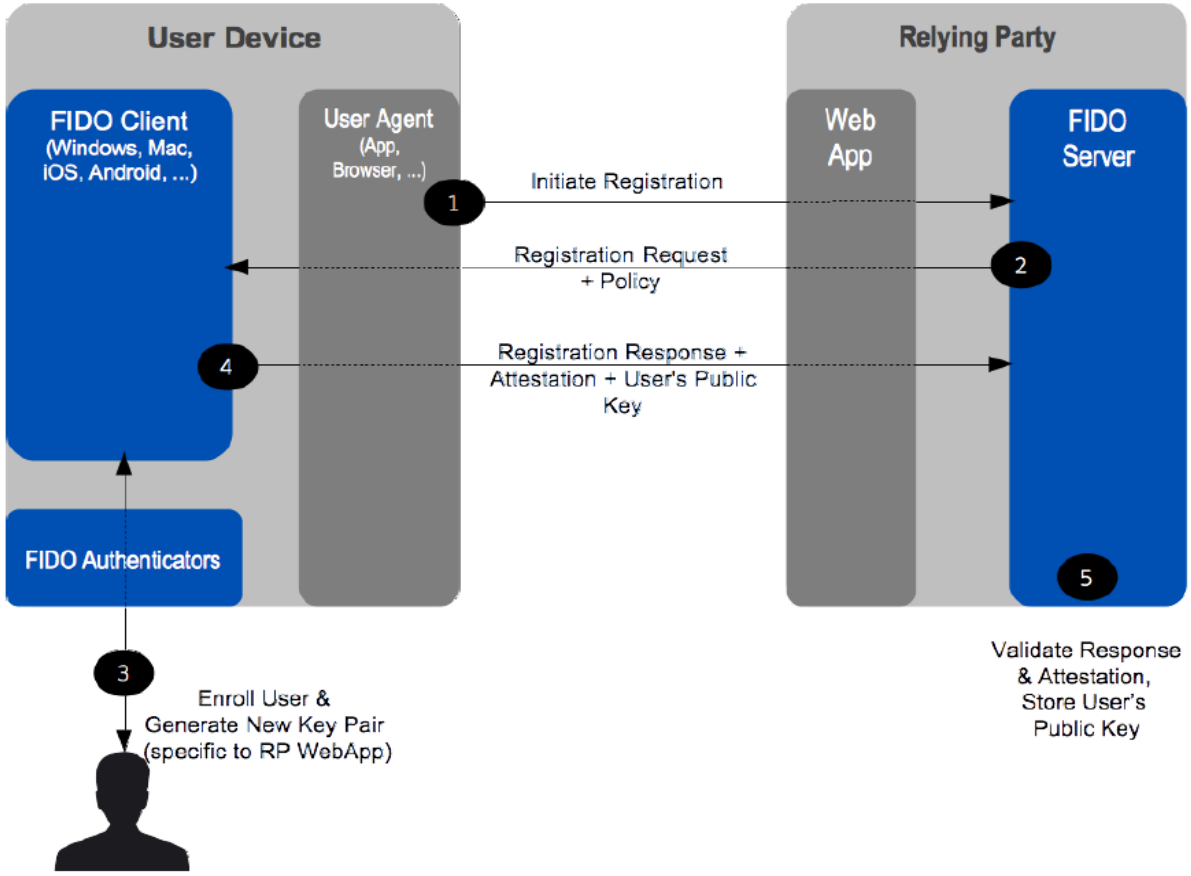
FIDO UAF Authenticator Metadata Validation

* In the FIDO UAF context, **attestation** is how Authenticators **make claims to a Relying Party** during registration that the *keys they generate, and/or certain measurements they report*, originate from ***genuine devices with certified characteristics.***
  * An **attestation signature**, carried in a FIDO UAF registration protocol message is validated by the FIDO UAF Server.
  * FIDO UAF Authenticators are created with attestation private keys used to create the signatures
    * The FIDO UAF Server validates the signature using that authenticator's attestation public key certificate located in the authenticator metadata.
    * **The metadata holding attestation** certificates is shared with FIDO UAF Servers **out of band**.

FIDO UAF Authenticator Acquisition and User Enrollment

* It is expected that users will **acquire** FIDO UAF Authenticators in various ways:
  * they purchase a new system that comes with embedded FIDO UAF Authenticator capability;
  * they purchase a device with an embedded FIDO UAF Authenticator,
  * or they are given a FIDO Authenticator by their employer or some other institution such as their bank.
* After receiving a FIDO UAF Authenticator, the user must go through an authenticator-specific **enrollment** process,
  * For example, in the case of a fingerprint sensing authenticator, the user must register their fingerprint(s) with the authenticator.
* Once enrollment is complete, the FIDO UAF Authenticator is ready for **registration** with FIDO UAF enabled online services and websites.

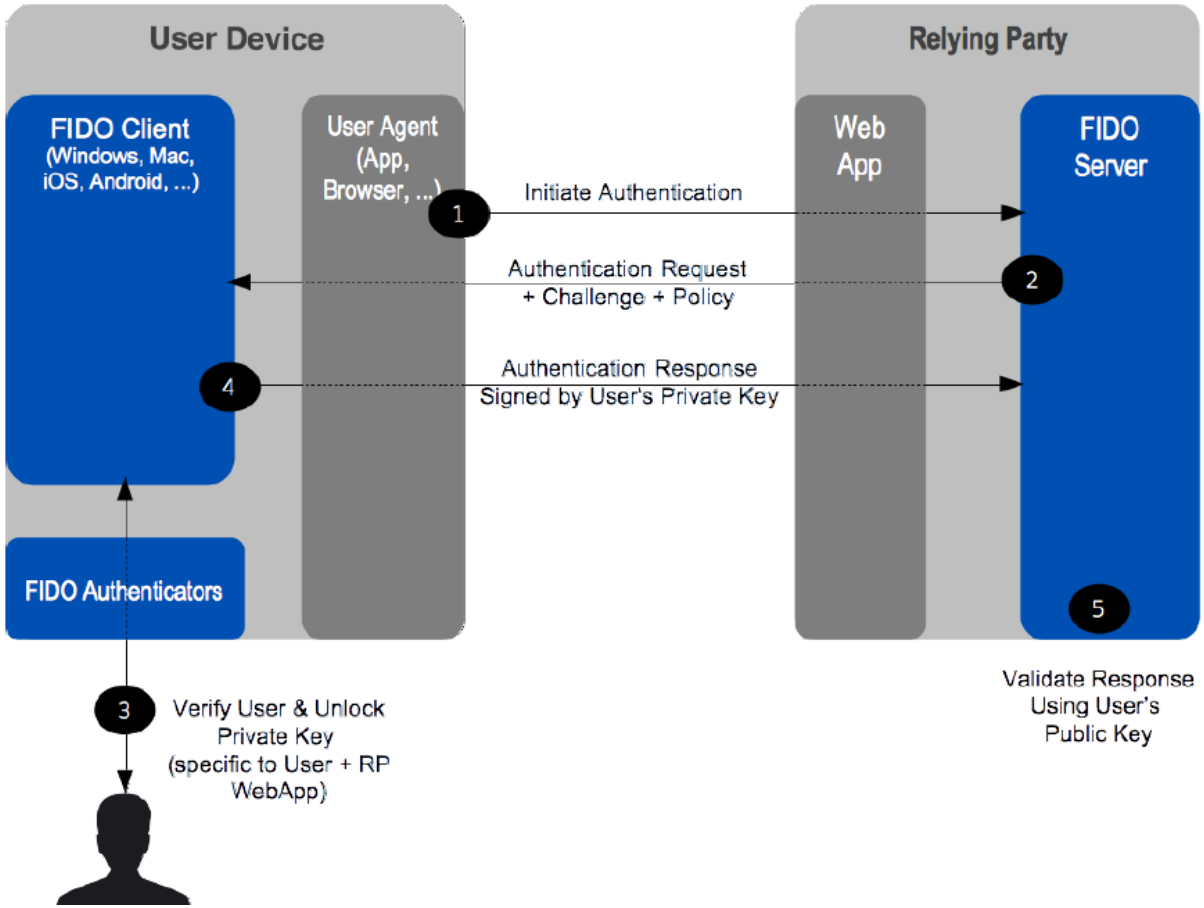
FIDO Authenticator Registration



A Relying Party is able to transparently detect when a user begins interacting with them while possessing an initialized FIDO UAF Authenticator.

In this initial introduction phase, the website will prompt the user regarding any detected FIDO UAF Authenticator(s), giving the user options regarding registering it with the website or not.

FIDO - Authentication



* Following registration, the FIDO UAF Authenticator will be subsequently employed whenever the user authenticates with the website (and the authenticator is present).
* The website can implement various fallback strategies for those occasions when the FIDO Authenticator is not present.
  * These might range from allowing conventional login with diminished privileges to disallowing login.

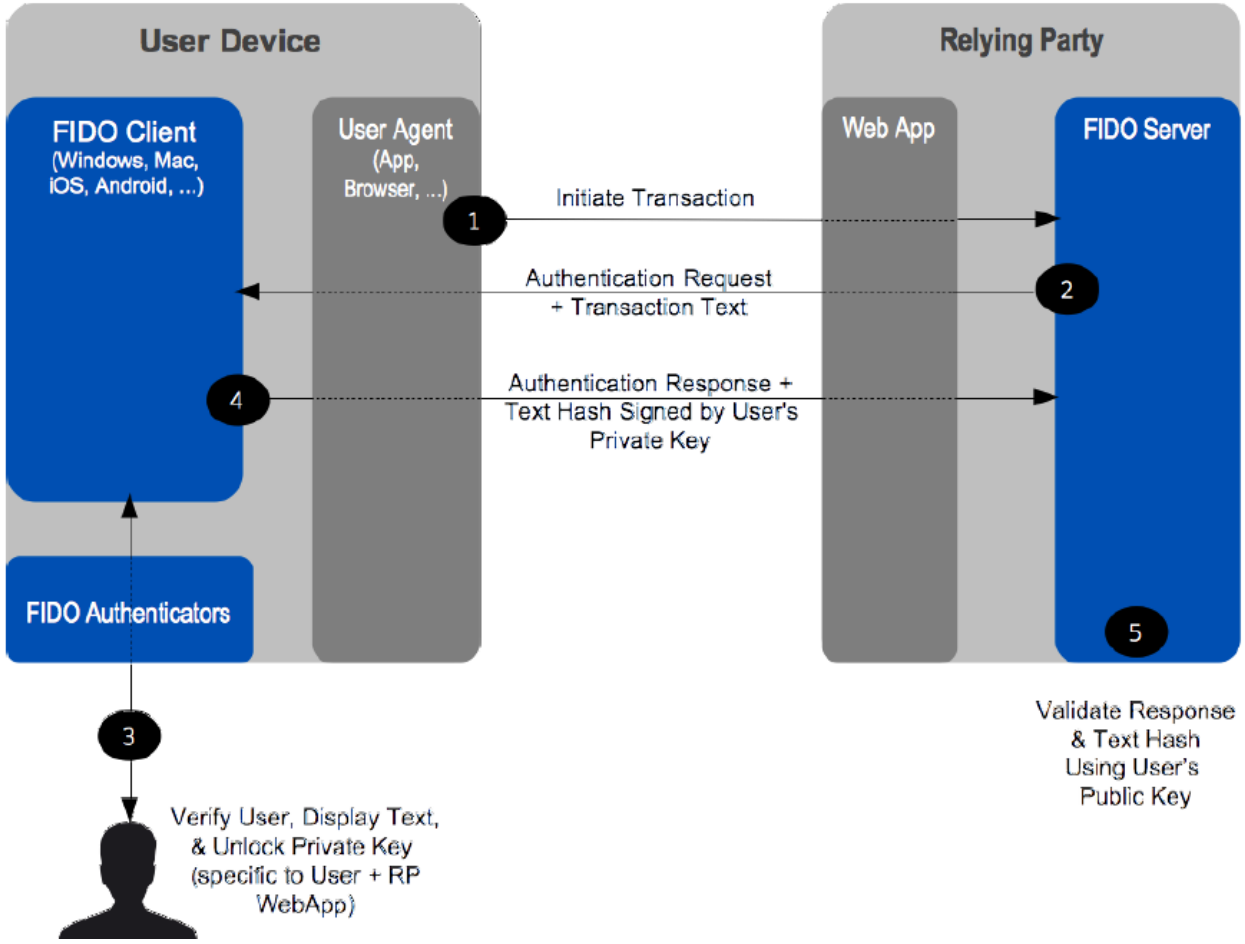
FIDO - Authentication

* This overall scenario will vary slightly depending upon the type of FIDO UAF Authenticator being employed.
* Some authenticators may
  * sample biometric data such as a face image, fingerprint, or voice print.
  * Others will require a PIN or local authenticator-specific passphrase entry.
  * Still others may simply be a hardware bearer authenticator.
* Note that it is permissible for a FIDO Client to interact with external services as part of the authentication of the user to the authenticator as long as the FIDO Privacy Principles are adhered to.

FIDO - Step-up Authentication

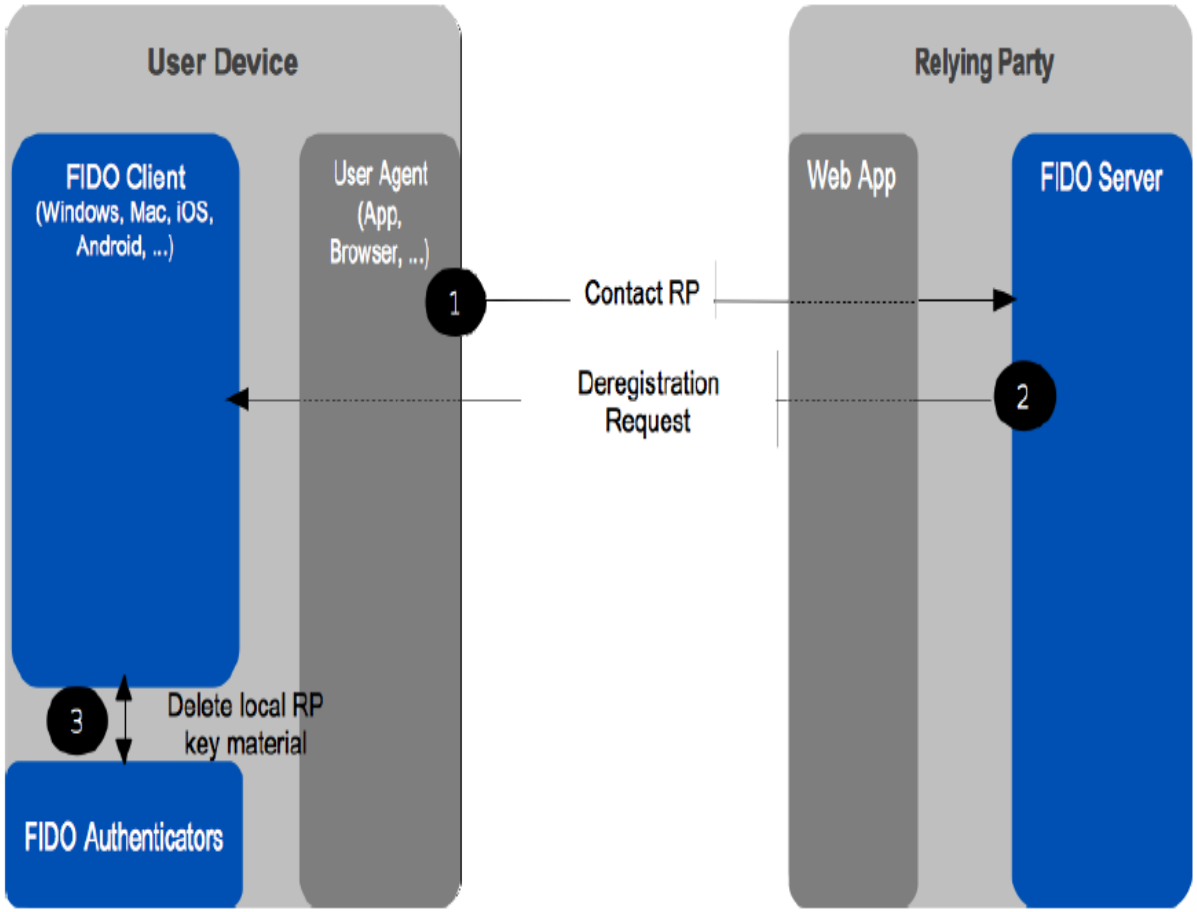
* Often, online services and websites allow unauthenticated, and/or only nominally authenticated use -- for informational browsing, for example.
* However, once users request **more valuable interactions**, such as when entering a members-only area, the website may request further **higher-assurance authentication**.
  * This could proceed in several steps if the user then wishes to purchase something, with **higher-assurance steps with increasing transaction value.**
  * FIDO UAF will smoothly facilitate this interaction style since **the website will be able to discover which FIDO UAF Authenticators are available** on FIDO-wielding users' systems, and **select incorporation of the appropriate one(s) in any particular authentication interaction**

FIDO - Transaction Confirmation



* A Relying Party wants the end-user to confirm a transaction (e.g. financial operation, privileged operation, etc) so that any tampering of a transaction message during its route to the end device display and back can be detected.
* FIDO architecture has a concept of **"secure transaction"** which provides this capability.
* Basically if a FIDO UAF Authenticator has **a transaction confirmation display capability**, FIDO UAF architecture makes sure that **the system supports What You See is What You Sign mode (WYSIWYS).**

FIDO - Authenticator Deregistration



* There are some situations where a Relying Party may need to **remove the UAF credentials** associated with a specific user account in FIDO Authenticator.
  * For example, the user's account is **cancelled** or **deleted**, the user's FIDO Authenticator is **lost** or **stolen**, etc.
  * In these situations, the RP may request the FIDO Authenticator *to delete authentication keys that are bound to the user's account.*

Adoption of New Types of FIDO UAF Authenticators

* Authenticators will evolve and new types are expected to appear in the future.
* Their adoption on the part of both users and Relying Parties is facilitated by the FIDO architecture.
* In order to support a new FIDO UAF Authenticator type, Relying Parties need only to add a new entry to their configuration describing the new authenticator, along with its FIDO Attestation Certificate.
* Afterwards, end users will be able to use the new FIDO UAF Authenticator type with those Relying Parties.

Privacy Considerations

* User privacy is fundamental to FIDO and is supported in UAF by design:
  * A UAF device **does not have a global identifier visible across relying parties** and does **not have a global identifier within a particular relying party**.
    * If for example, a person looses their UAF device, someone finding it cannot "point it at a relying party" and discover if the original user had any accounts with that relying party.
    * Similarly, if two users share a UAF device and each has registered their account with the same relying party with this device, the relying party will not be able to discern that the two accounts share a device, based on the UAF protocol alone.

Privacy Considerations

* The UAF protocol generates unique asymmetric cryptographic key pairs on a per-device, per-user account, and per-relying party basis.
  * Cryptographic keys used with different relying parties will not allow any one party to link all the actions to the same user, hence the **unlinkability** property of UAF.
* The UAF protocol operations require **minimal personal data collection:**
  * at most they incorporate a user's relying party username. This personal data is only used for FIDO purposes, for example to perform user registration, user verification, or authorization.
  * This personal data **does not leave the user's computing environment** and is only persisted locally when necessary

Privacy Considerations

* In UAF, user **verification is performed locally**.
  * The UAF protocol does not convey biometric data to relying parties, nor does it require the storage of such data at relying parties.
* Users **explicitly approve the use of a UAF device with a specific relying party**.
  * Unique cryptographic keys are generated and bound to a relying party during registration only after the user's explicit consent.
* UAF authenticators can only be identified by their attestation certificates on a production batch-level or on manufacturer- and device mode level.
  * They **cannot be identified individually**. The UAF specifications require implementers to ship UAF authenticators with the same attestation certificate and private key in batches of 100,000 or more, in order to provide **unlinkability**.

Secure Transactions

* **Secure transactions:** https://www.intel.com/content/www/us/en/architecture-and-technology/identity-protection/identity-protection-technology-general.html?eu-cookie-notice

* **Metadata Service:** https://fidoalliance.org/metadata/

U2F - Tokens

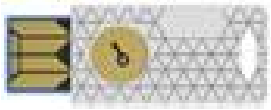

FEITIAN
FEITIAN
WE BUILD SECURITY


fido


fido
CERTIFIED
U2F


FIDO U2F


fido
CERTIFIED
U2F



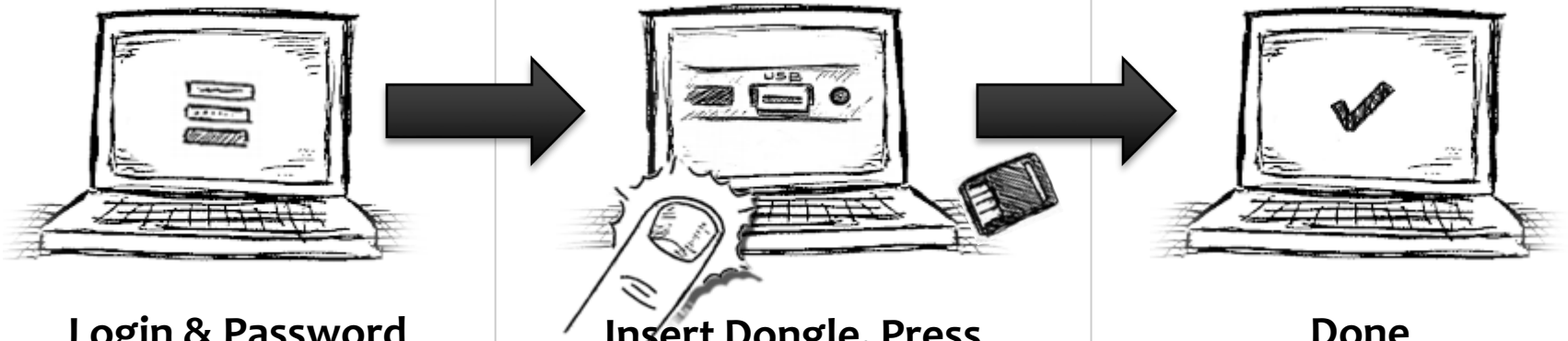
FIDO Experiences

ONLINE AUTH REQUEST	LOCAL DEVICE AUTH	SUCCESS

PASSWORDLESS EXPERIENCE (UAF standards)

Transaction Detail → Show a biometric → Done

SECOND FACTOR EXPERIENCE (U2F standards)

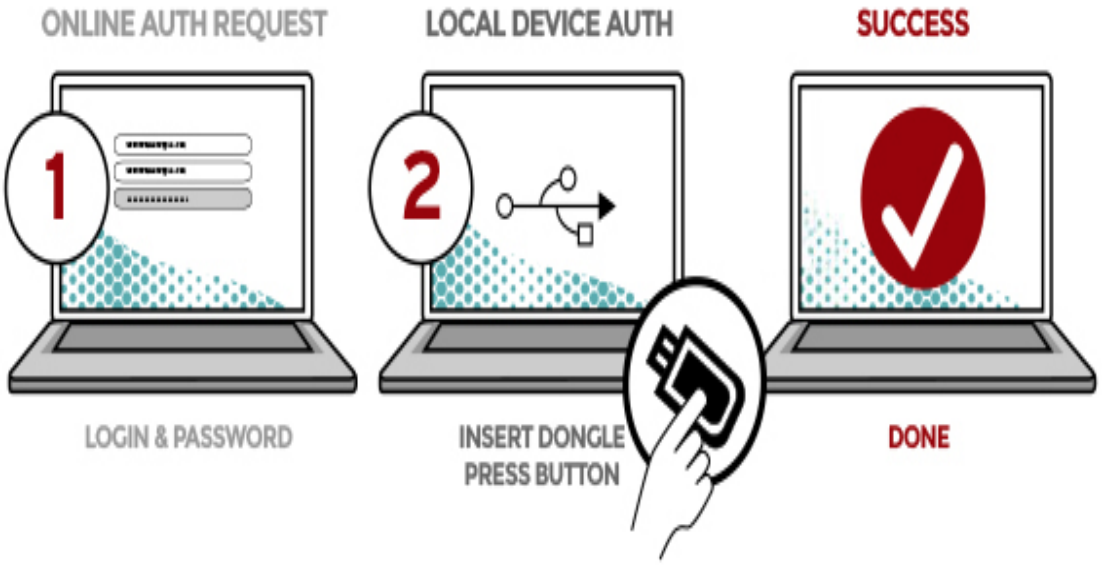


Login & Password → Insert Dongle, Press button → Done

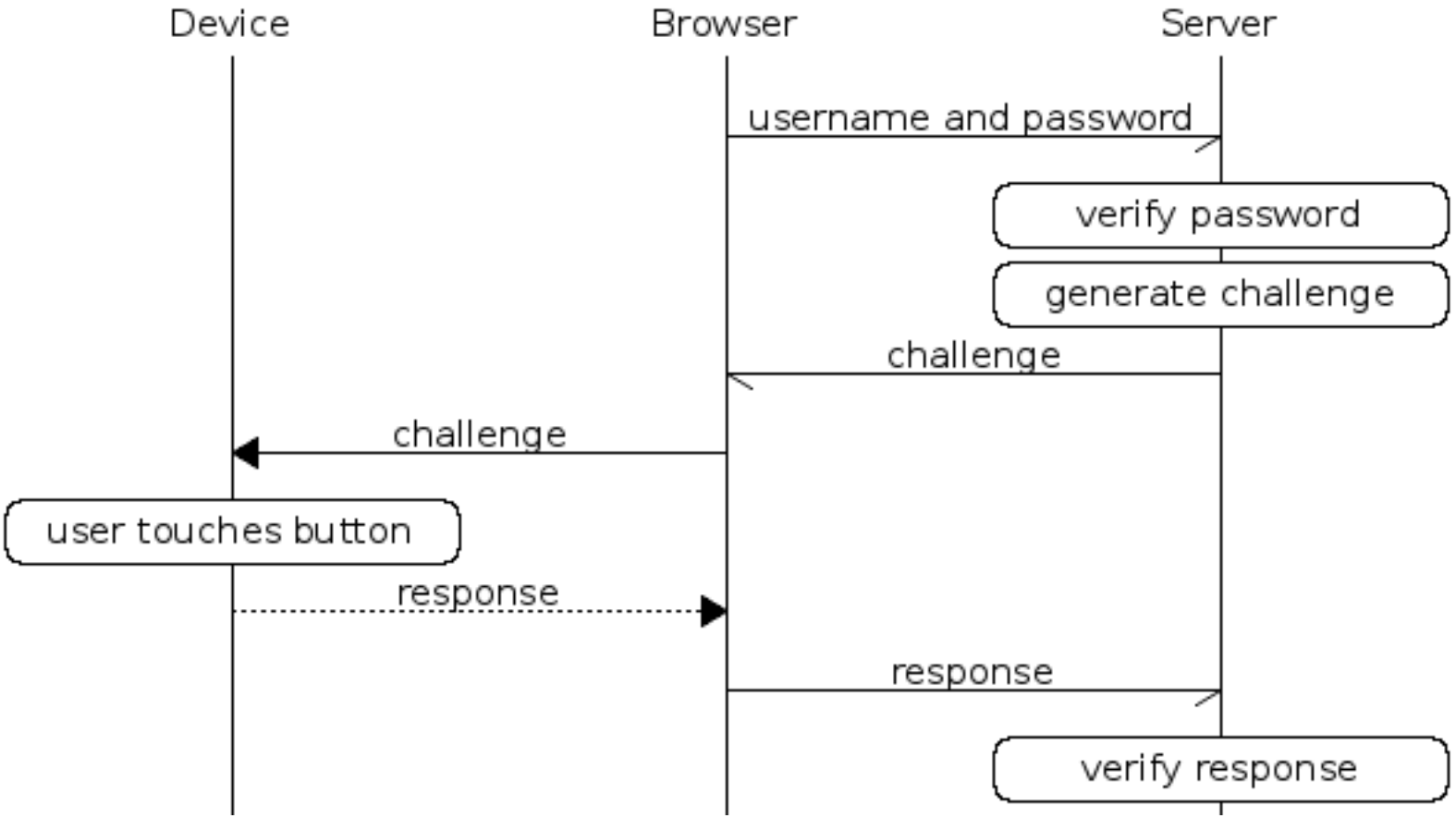
Universal Second Factor - U2F

U2F is an open 2-factor authentication standard that enables keychain devices, mobile phones and other devices to securely access any number of web-based services — instantly and with no drivers or client software needed.

The U2F specifications are today hosted by the FIDO Alliance (http://fidoalliance.org/specifications/download)



Simplified schema



Universal Second Factor - U2F

Apart from the general advantages of 2-factor authentication, U2F has the following characteristics:

- **Phishing protection**, using application isolation (:appid) and TLS channel IDs (https://tools.ietf.org/html/draft-balfanz-tls-channelid-00).
- **MITM protection**, using appid / keyhandle mismatch
- **No shared secrets**. Unlike for example OATH, U2F uses public key cryptography and uses no shared secrets.
- **Anonymous**. New public keys are minted for each site.
- **User experience**. There are no codes to enter and no drivers to install.
- **Universal**. Devices can be of different types (hardware token, fingerprint reader, software token, etc.), communicate using different transports (USB, NFC, BLE) and can be registered to any number of sites.
- **Open standard**.
- **Backed by industry leaders** such as Google, Paypal, Microsoft, Bank of America, MasterCard and VISA.

Getting Started

U2F Client

Go to https://developers.yubico.com/u2f

Download your preferred library!
C, Java, Python, PHP!

U2F Server

Go to https://developers.yubico.com/u2f

Have a look at the python u2f server

Download your preferred library!
C, Java, Python, PHP!

https://fidoalliance.org/resources/videos/

W3C: FIDO 2.0 Platform Specifications 1.0

* W3C Members *Microsoft*, *Google*, *PayPal* and *NokNok Labs* submitted on November 2015 the following specifications to the W3C:
  * *FIDO 2.0 Web APIs* (*http://www.w3.org/Submission/2015/SUBM-fido-web-api-20151120/*)
    * An API that enables web pages to access FIDO 2.0 compliant strong cryptographic credentials through browser script.
  * *FIDO 2.0 Attestations* (*http://www.w3.org/Submission/2015/SUBM-fido-key-attestation-20151120/*
    * An attestation is used to provide a cryptographic proof of the authenticator model to the relying party.
  * *FIDO 2.0 Signature* (*http://www.w3.org/Submission/2015/SUBM-fido-signature-format-20151120/*)
    * FIDO 2.0 signature proves possession of a private key of a FIDO 2.0 credential and asserts contextual information about the client and authenticator that generated it.

State of Market Adoption in the real world

OEMs SHIPPING FIDO-READY ™ PRODUCTS

New and existing devices are supported

OEM Enabled: Lenovo ThinkPads with Fingerprint Sensors

OEM Enabled: Samsung Galaxy S5

Clients available for these operating systems :

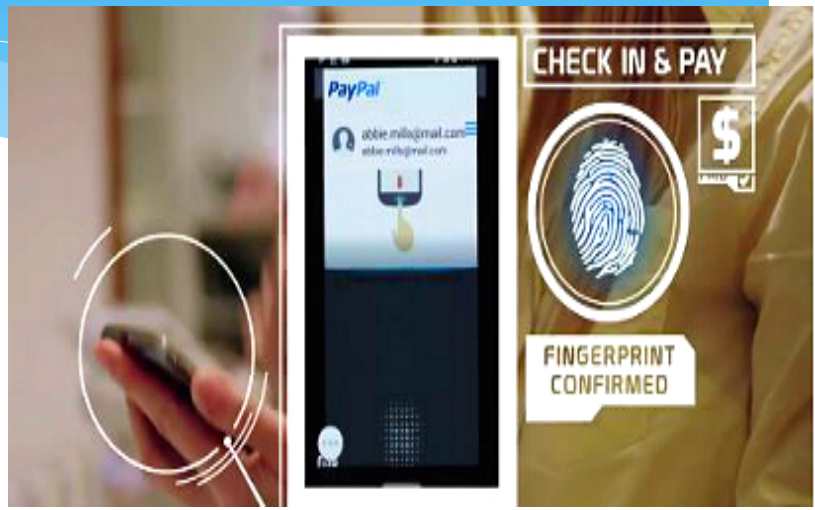
Windows 7 · Windows 8 · Android · iOS 7

Software Authenticator Examples:
Voice/Face recognition, PIN, QR Code, etc.

Aftermarket Hardware Authenticator Examples:
USB fingerprint scanner, MicroSD Secure Element

FIDO Deployment already live…

- Customers can use their finger to pay with PayPal from their new Samsung Galaxy S5 because the FIDO Ready™ software on the device securely communicates between the fingerprint sensor on their device and PayPal's service in the cloud. **The only information the device shares with PayPal is a unique cryptographic "public key"** that allows PayPal to verify the identity of the customer **without having to store any biometric information on PayPal's servers.**



PayPal™

SAMSUNG

Synaptics®

Nok Nok LABS

FIDO Deployment already live…

* Alipay – formerly a part of Alibaba Group in China
* Processed $519 Billion in transactions in 2013
* Launched FIDO-based payments using Galaxy S5

Nok Nok LABS

SAMSUNG

lenovo FOR THOSE WHO DO.

Synaptics

Secure authentication for Government services

* **UK BECOMES THE FIRST GOVERNMENT TO OFFER SECURE ONLINE IDENTITIES BASED ON FIDO U2F STANDARDS**
* **STOCKHOLM & AMSTERDAM, March 23, 2016 – Yubico, the leading provider of simple, open and strong authentication, and Digidentity, a leading identity service provider, today announced a partnership to enable FIDO Universal 2nd Factor (U2F) authentication and YubiKeys for UK government services.**
* **The joint solution allows all UK citizens to easily and securely access GOV.UK Verify digital public services.**

FIDO2

* FIDO2 is the passwordless evolution of the FIDO Universal 2nd Factor (U2F) standard.
* FIDO2 supports more use cases, including passwordless authentication for various operating systems.
* Organizations will soon have the option to enable employees and customers to sign in to an Azure AD joined device with no password, by simply using a Security Key to get single sign-on to all Azure AD based applications and services.
* Google and Mozilla also announced Chrome and Firefox support for the Web Authentication API (WebAuthn) developed by Yubico and members of the World Wide Web Consortium (W3C) and included in the FIDO2 specification.
  * https://www.w3.org/TR/webauthn/

FIDO2

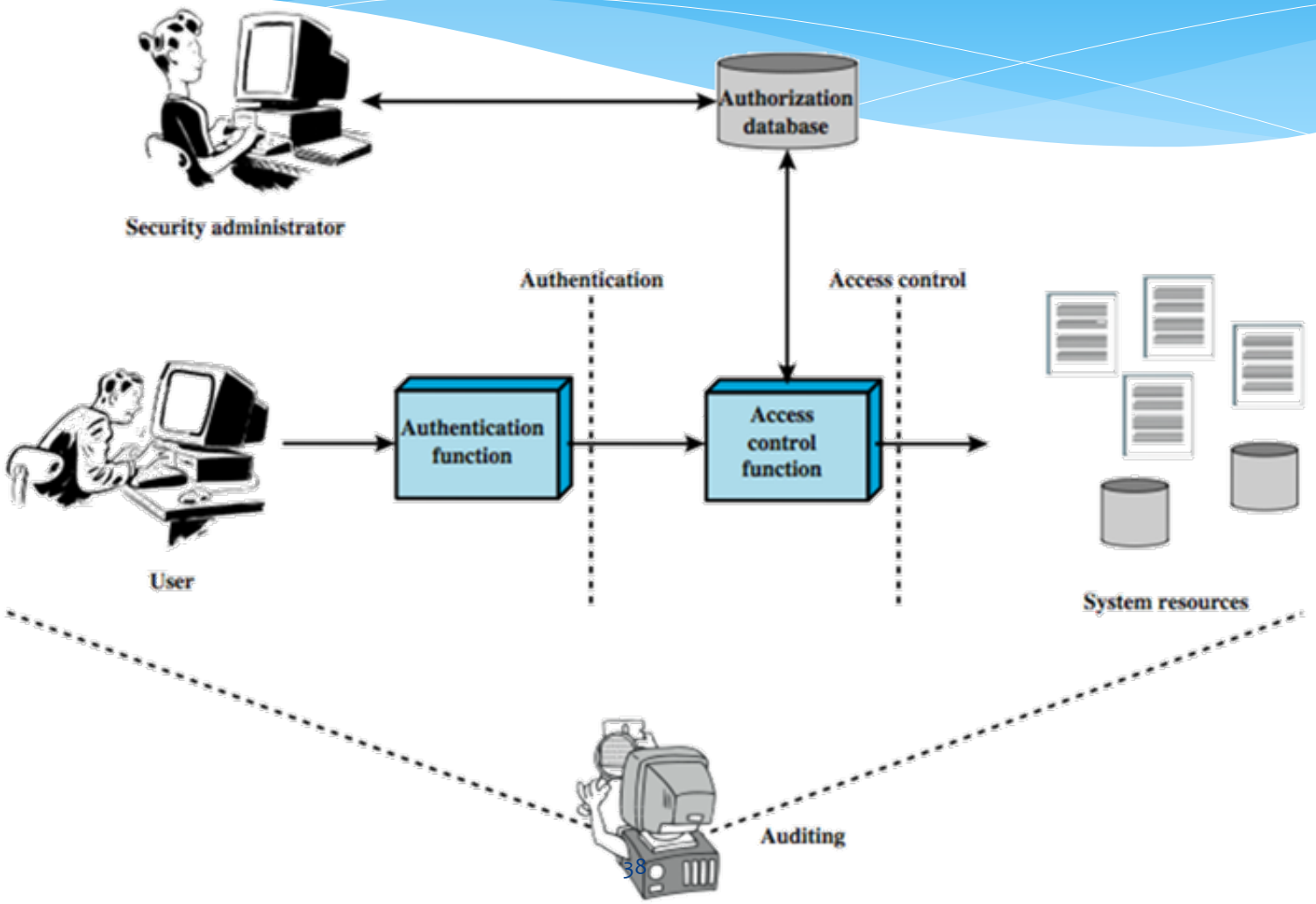
* FIDO2 is built on the same security and privacy features of FIDO U2F: strong public key cryptography, no drivers or client software and one key for unlimited account access with no shared secrets.
  * **Single Factor:** This only requires possession of the Security Key to log in, allowing for a passwordless tap-and-go experience.
  * **Second-Factor:** In a two-factor authentication scenario, such as the current Google and Facebook FIDO U2F implementations, the Security Key is used as a strong second factor along with a username and password.
  * **Multi-Factor:** This allows the use of the Security Key with an additional factor such as a PIN (instead of a password), to meet the high-assurance requirements of operations like financial transactions, or submitting a prescription.

Access Control

Access Control

* "The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner"
* central element of computer security
* assume have users and groups
  * authenticate to system
  * assigned access rights to certain resources on system

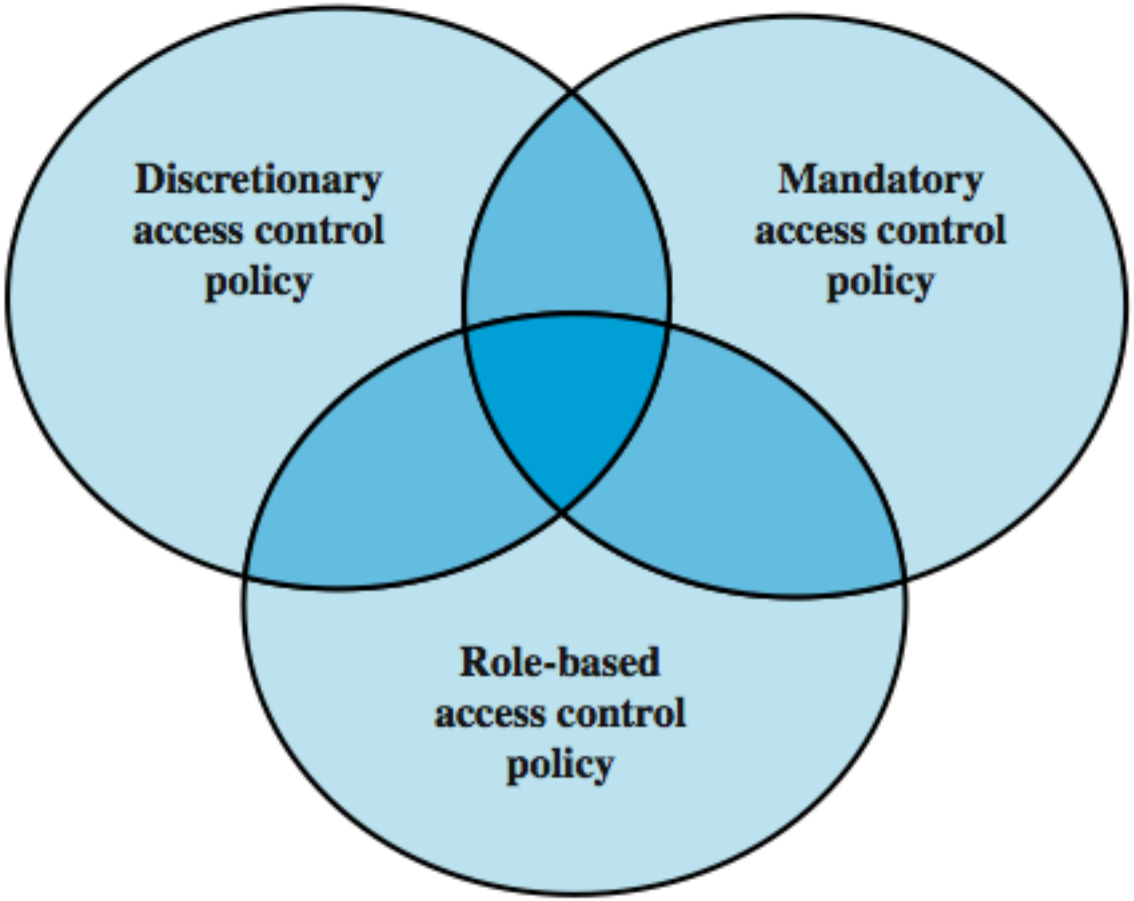
Access Control Principles



Authentication vs Authorization

* Authentication — Are you who you say you are?
  * Restrictions on who (or what) can access system
* **Authorization** — Are you allowed to do that?
  * Restrictions on actions of authenticated users
* Authorization is a form of **access control**
* Classic authorization enforced by
  * Access Control Lists (ACLs)
  * Capabilities (C-lists)

Access Control Policies



Access Control Requirements

* reliable input
* fine and coarse specifications
* least privilege
* separation of duty
* open and closed policies
* policy combinations, conflict resolution
* administrative policies

Access Control Elements

* subject - entity that can access objects
  * a process representing user/application
  * often have 3 classes: owner, group, world
* object - access controlled resource
  * e.g. files, directories, records, programs etc
  * number/type depend on environment
* access right - way in which subject accesses an object
  * e.g. read, write, execute, delete, create, search

Discretionary Access Control

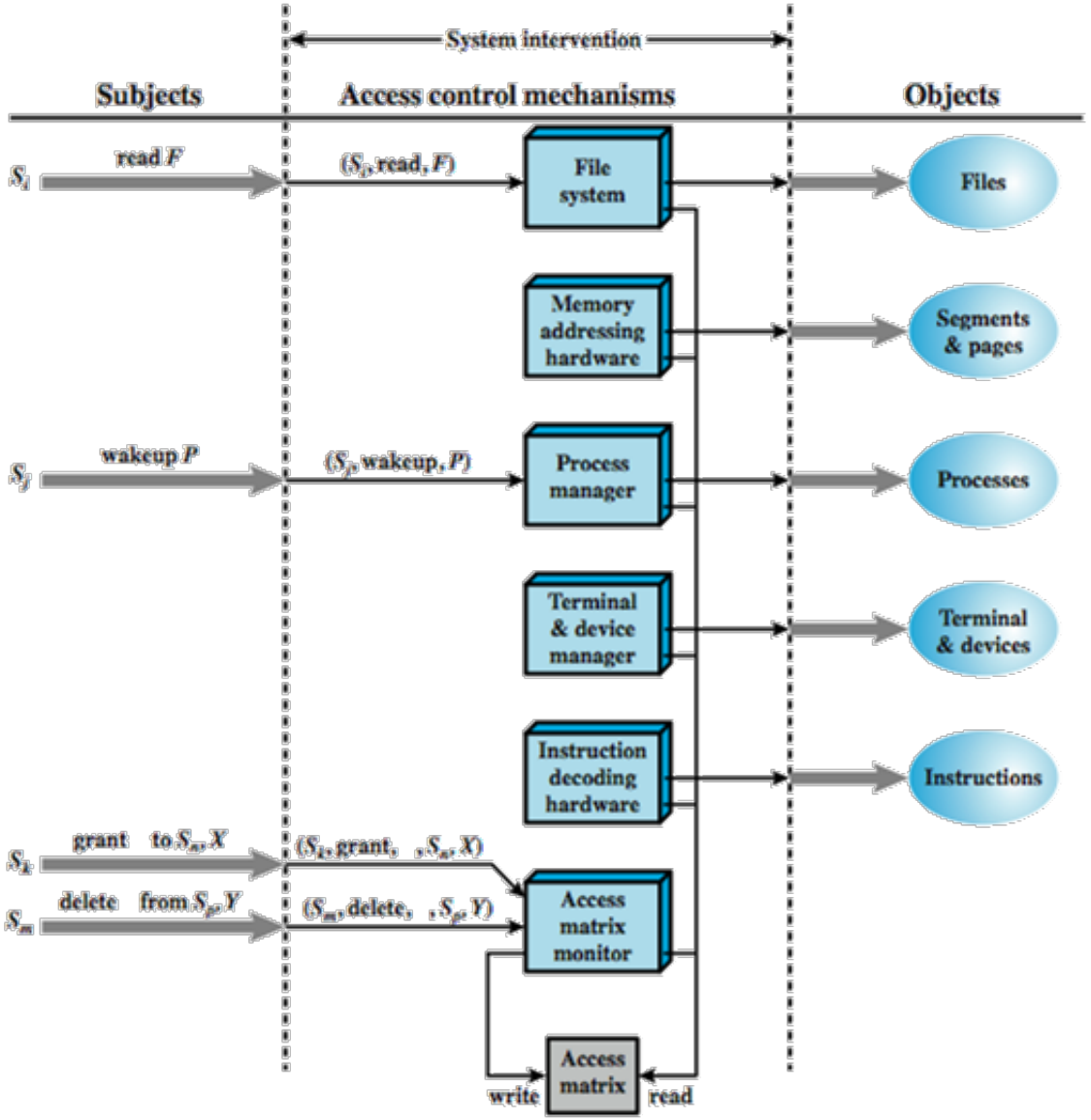
* often provided using an access matrix
  * lists subjects in one dimension (rows)
  * lists objects in the other dimension (columns)
  * each entry specifies access rights of the specified subject to that object
* access matrix is often sparse
* can decompose by either row or column

Lampson's Access Control Matrix

- **Subjects** (users) index the rows
- **Objects** (resources) index the columns

	OS	Accounting program	Accounting data	Insurance data	Payroll data
Bob	rx	rx	r	---	---
Alice	rx	rx	r	rw	rw
Sam	rwx	rwx	r	rw	rw
Accounting program	rx	rx	rw	rw	rw

Access Control Function

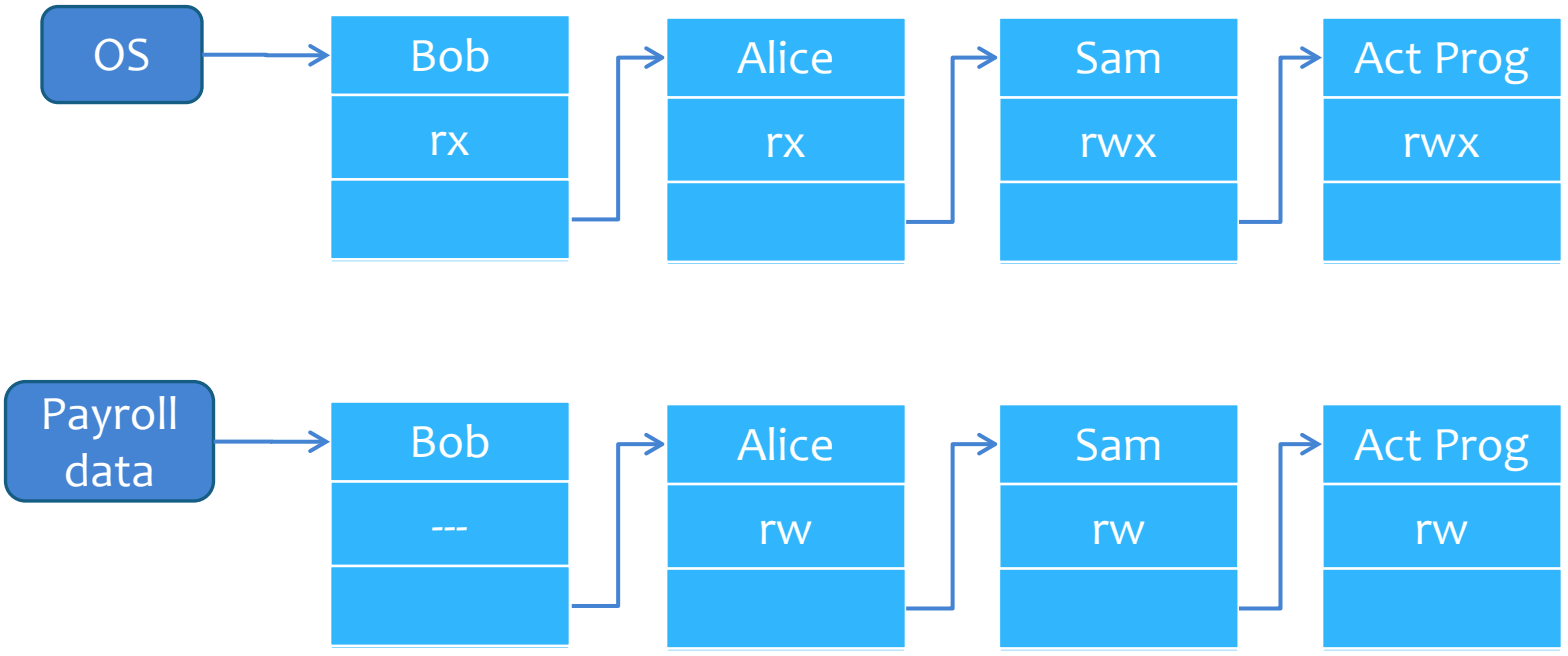


Are You Allowed to Do That?

* **Access control matrix** has **all** relevant info
* Could be 1000's of users, 1000's of resources
* Then matrix with 1,000,000's of entries
* How to manage such a large matrix?
* Need to check this matrix before access to any resource is allowed
* How to make this efficient?

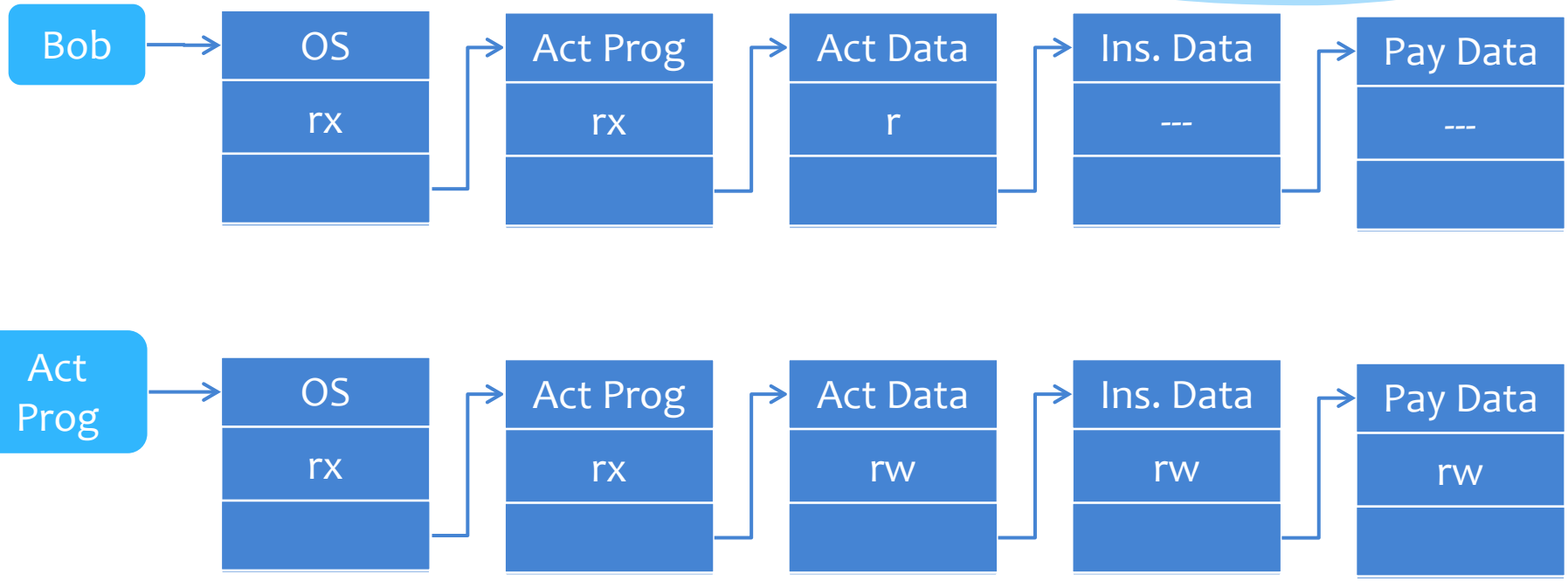
Access Control Lists

* ACLs: store access control matrix by **column**

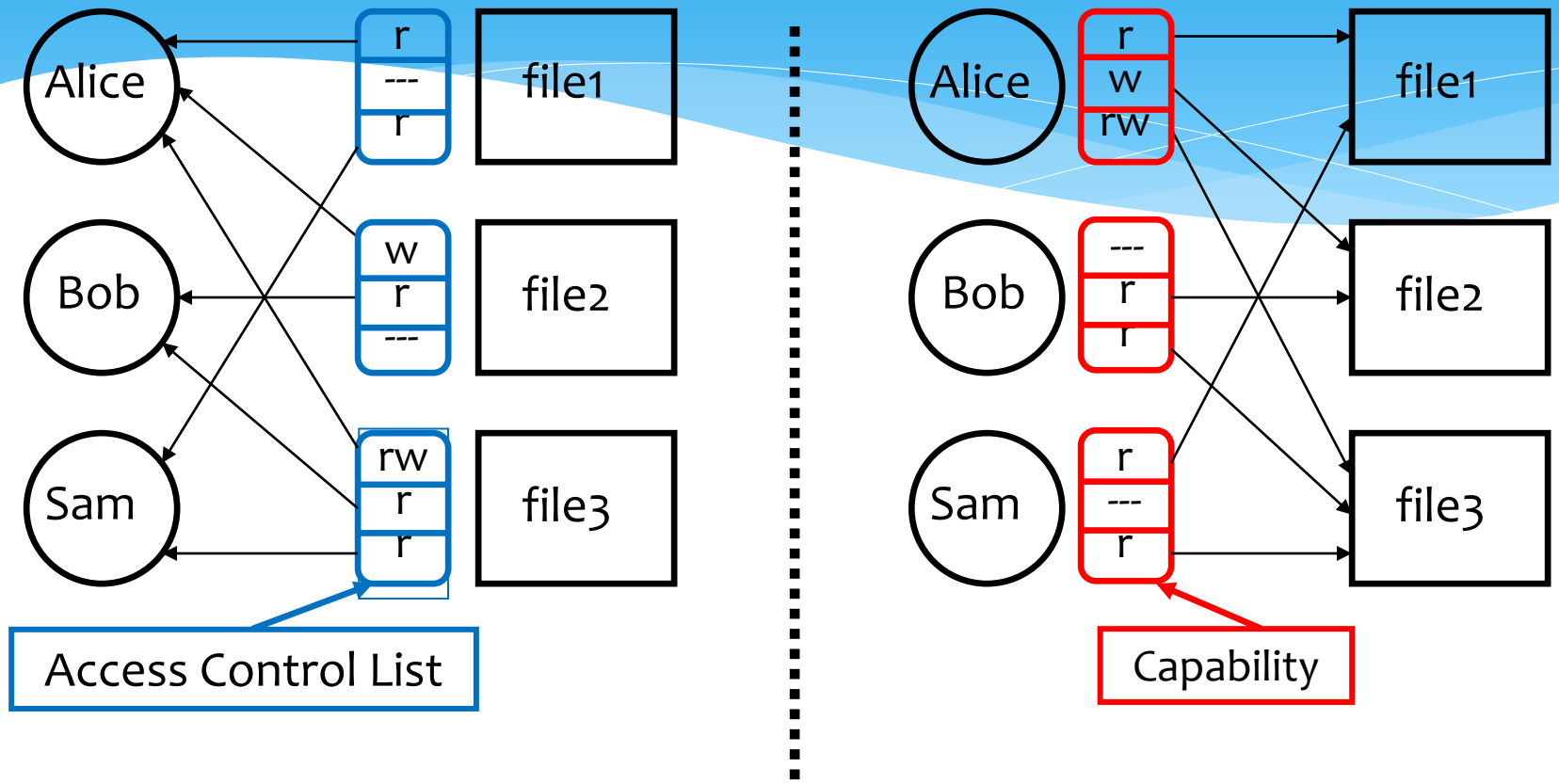


Capabilities (C-lists)

* Store access control matrix by row



ACLs vs Capabilities



* Note that arrows point in opposite directions…
* With ACLs, need to associate users to files

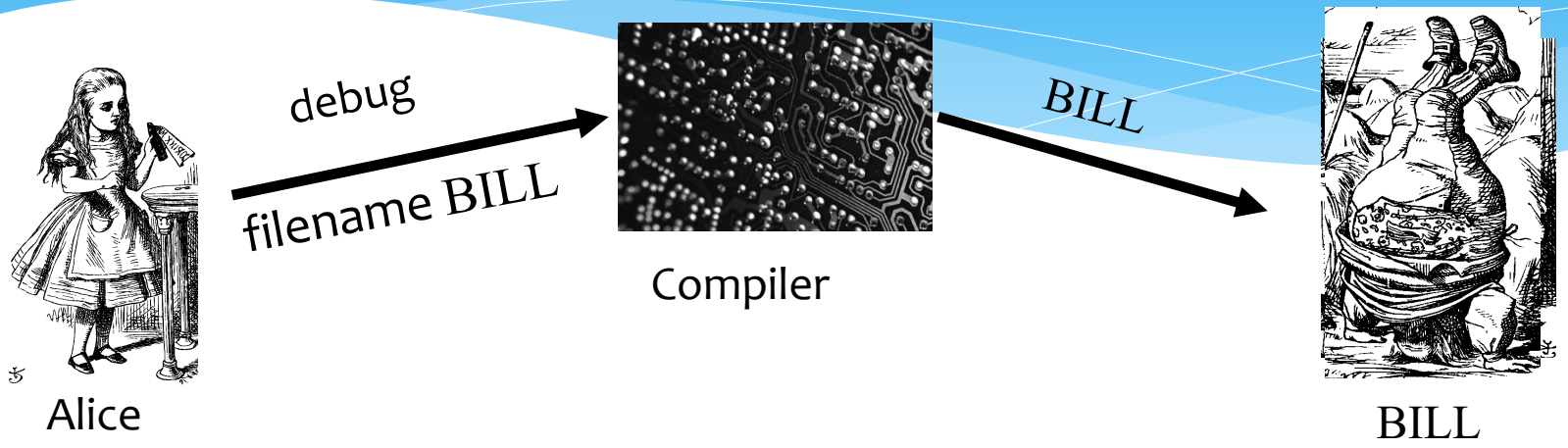
Confused Deputy

* Two resources
  * Compiler and BILL file (billing info)
* Compiler can write file BILL
* Alice can invoke compiler with a debug filename
* Alice not allowed to write to BILL

* Access control matrix

	Compiler	BILL
Alice	x	---
Compiler	rx	rw

ACL's and Confused Deputy



* Compiler is **deputy** acting on behalf of Alice
* Compiler is **confused**
  * Alice is not allowed to write BILL
* Compiler has confused its rights with Alice's

Confused Deputy

* Compiler acting for Alice is confused
* There has been a separation of **authority** from the **purpose** for which it is used
* With ACLs, difficult to avoid this problem
* With Capabilities, easier to prevent problem
  * Must maintain association between authority and intended purpose
  * Capabilities also easy to **delegate** authority

ACLs vs Capabilities

* ACLs
  * Good when users manage their own files
  * Protection is data-oriented
  * Easy to change rights to a resource
* Capabilities
  * Easy to delegate – avoid the confused deputy
  * Easy to add/delete users
  * More difficult to implement
  * The "Zen of information security"
* Capabilities loved by academics
  * Capability Myths Demolished