



U2F & UAF Tutorial

How Secure is Authentication?

2014 1.2bn?

2013 397m

How an epic blunder in the hand of password crackers

Engineers flout universal taboo by encrypting passwords

by Dan Goodin - Nov 1, 2013 12:00 pm UTC



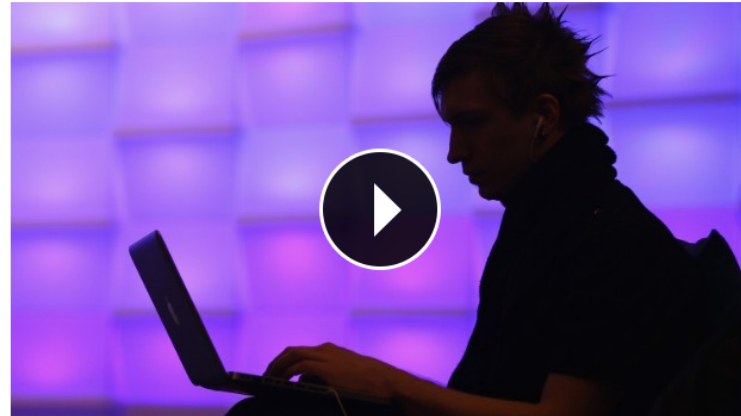
Wikipedia

Four weeks ago, Adobe disclosed a **sustained corporate network** that threatened to spawn meaner malware attacks by giving criminals a raw source code for the company's widely used ColdFusion applications. Now, researchers at the same breach could significantly strengthen the crackers' collective hand by revealing a staggering million passcodes used over the years by Adobe. Many of them from the FBI, large corporations and sensitive organizations.

That's because Adobe engineers used reversible encryption to scramble the passwords contained in a 9.3-gigabyte file that's now available online. They flouted almost universally recognized best practices that call for stored passwords to be protected by another one-way cryptographic hashing algorithm. A ground hamburger can't be converted back to its original cryptographic hashes and return them to the crackers by requiring crackers to pass individual password guesses through the same

Russian criminals steal 1.2 billion passwords

By James O'Toole and Jose Pagliery @CNNTech August 6, 2014: 6:56 AM ET



Russian hackers know your password

NEW YORK (CNNMoney)

Russian criminals have stolen 1.2 billion Internet user names and passwords, amassing what could be the largest collection of stolen digital credentials in history, a respected security firm said Tuesday.

There's **no need to panic at this point** -- Hold Security, the firm that discovered the theft, says the gang isn't in the business of stealing your bank account information. Instead, they make their money by sending out spam for bogus products like weight-loss pills.

The Milwaukee-based firm, didn't reveal the identities of the targeted websites, citing nondisclosure agreements and a desire to prevent existing vulnerabilities from being more widely exploited.

Hold Security founder Alex Holden told CNNMoney that the trove includes credentials gathered from over 420,000 websites -- both smaller sites as well as "household

Dec. 2013

145m

Oct. 2013

130m

May 2013

22m

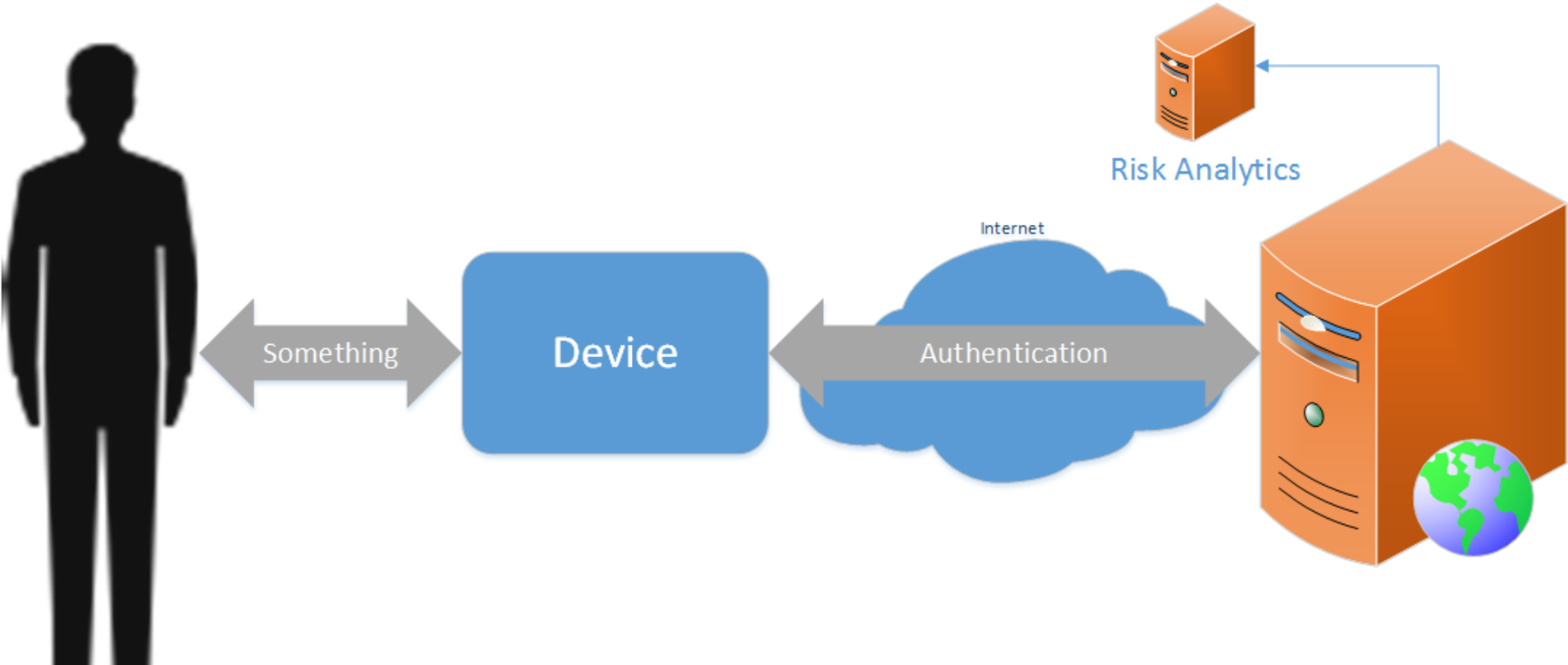
April 2013

50m

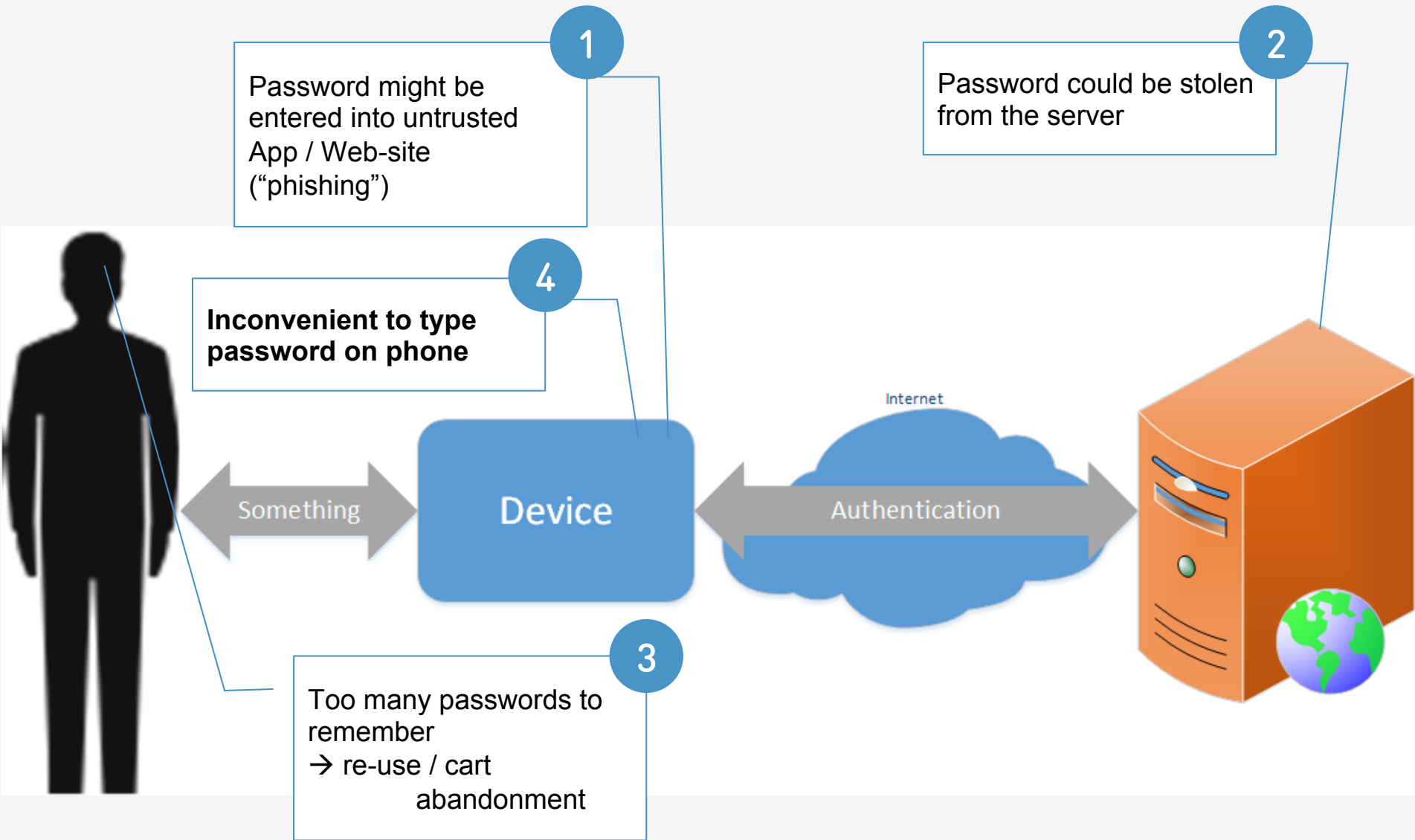
March 2013

50m

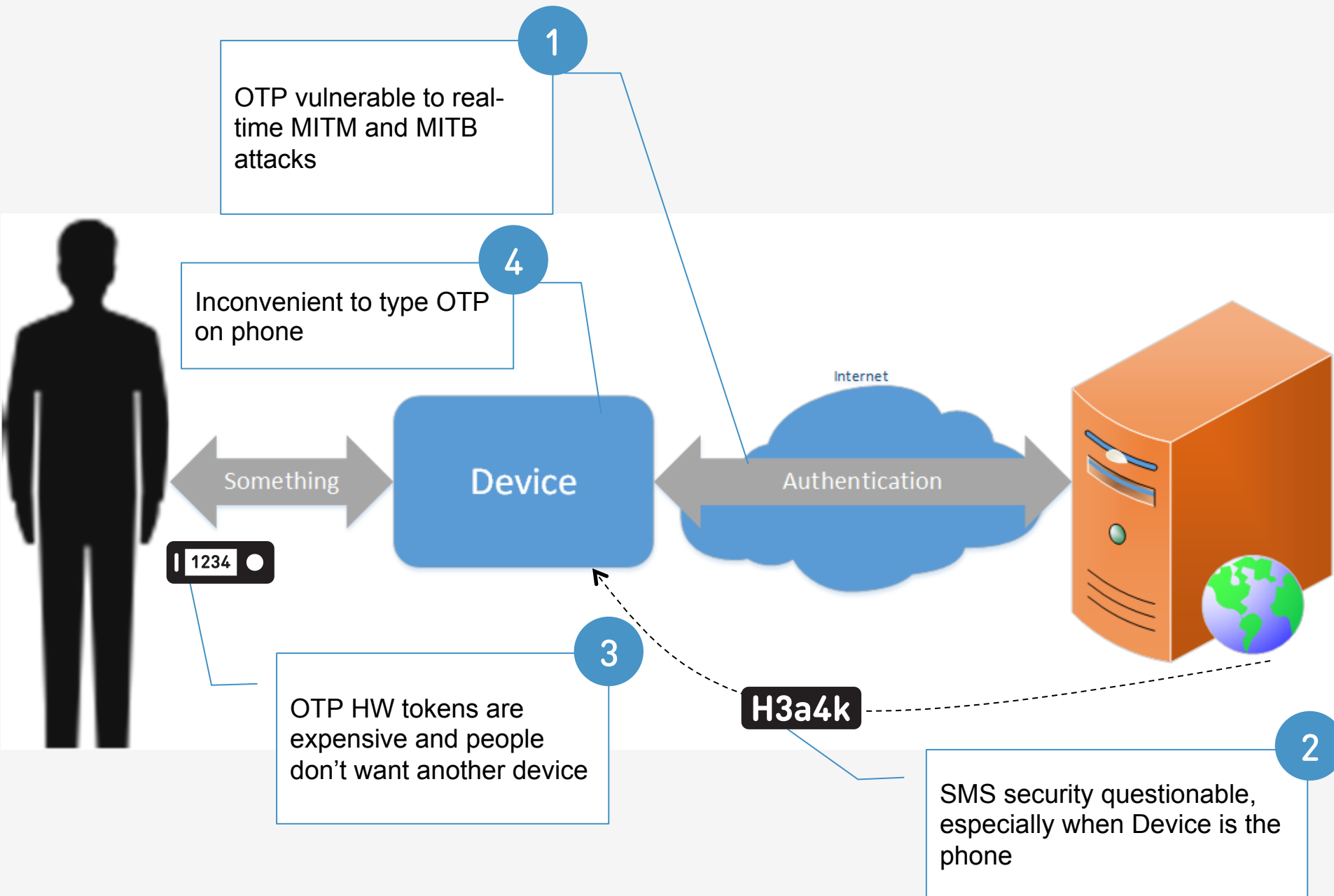
Cloud Authentication



Password Issues



OTP Issues



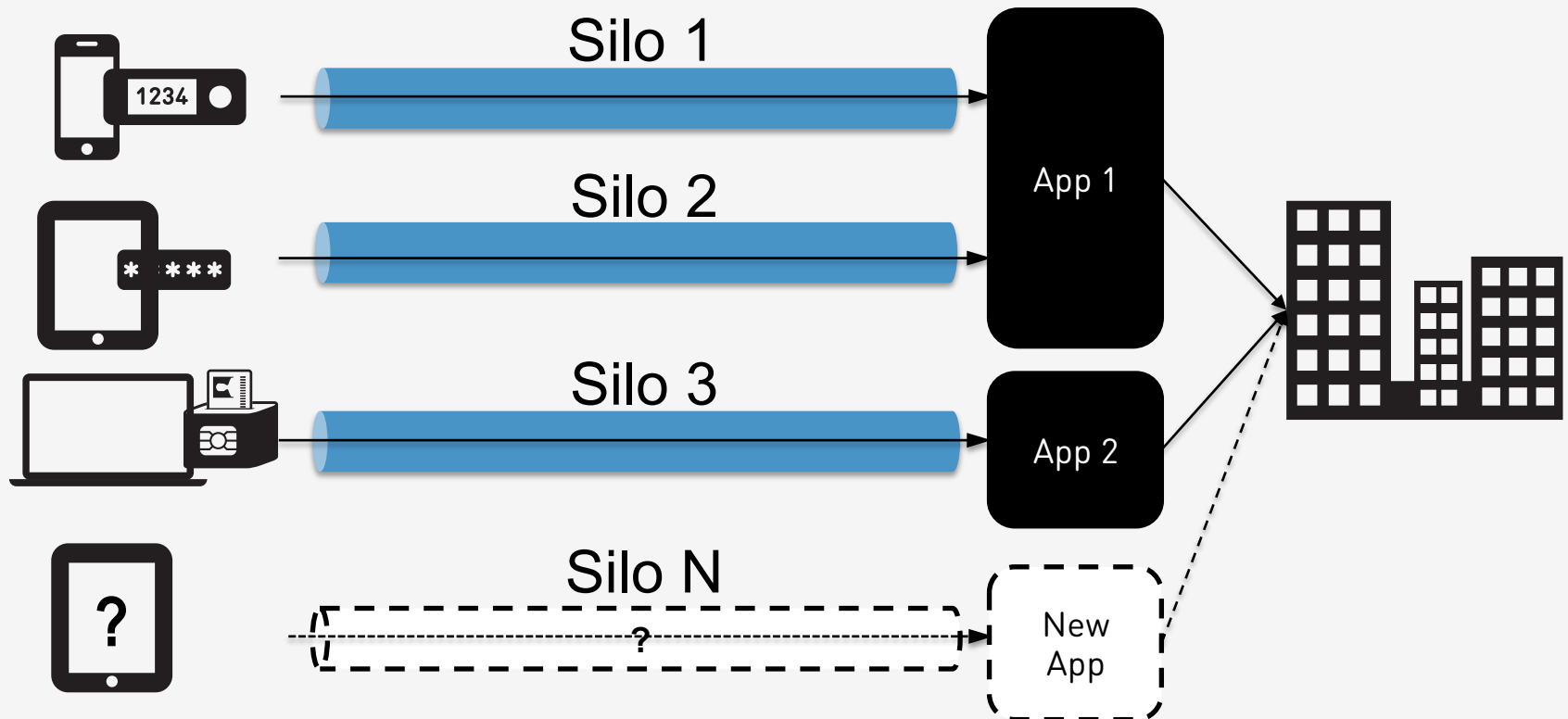
Implementation Challenge

A Plumbing Problem

User Verification Methods

Applications

Organizations



Authentication Needs

Do you want to login?

Do you want to transfer \$100 to Frank?

Do you want to ship to a new address?

Do you want to delete all of your emails?

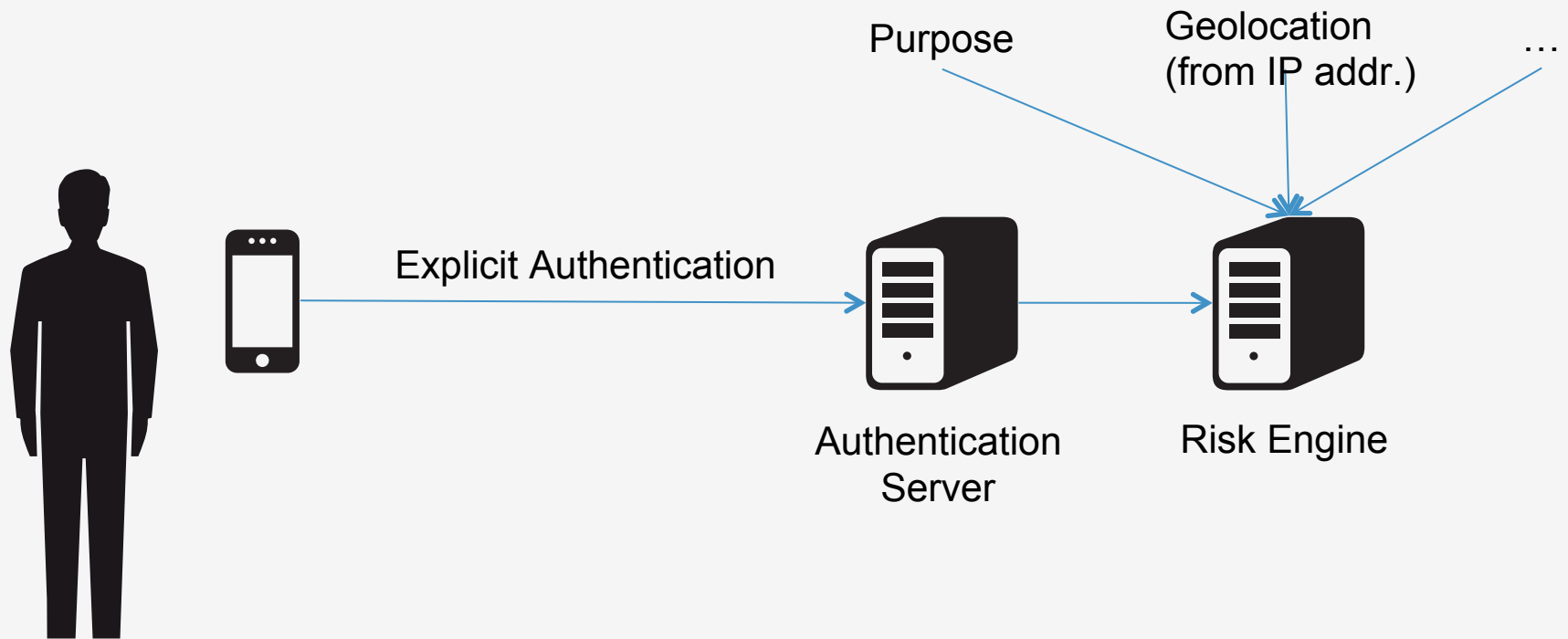
Do you want to share your dental record?

Authentication today:

Ask user for a password

(and perhaps a one time code)

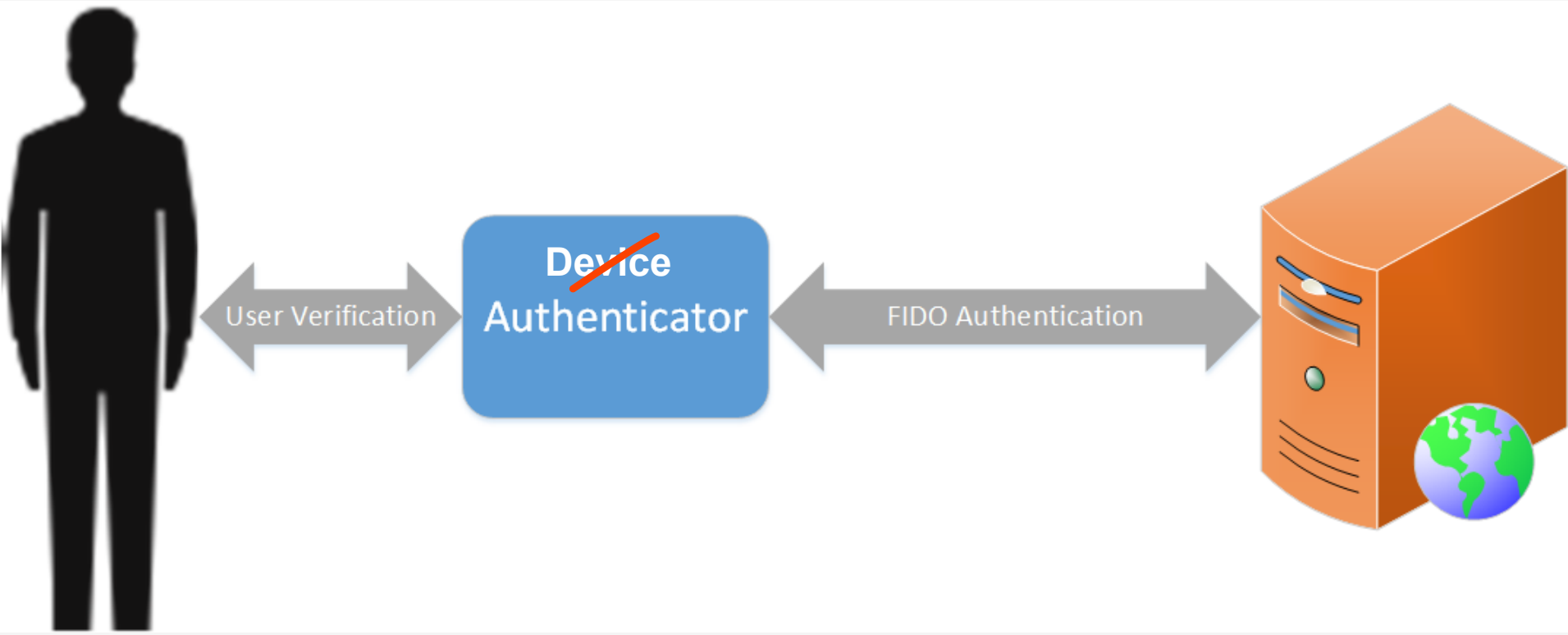
Authentication & Risk Engines



Summary

1. Passwords are insecure and inconvenient especially on mobile devices
2. Alternative authentication methods are silos and hence don't scale to large scale user populations
3. The required security level of the authentication depends on the use
4. Risk engines need information about the explicit authentication security for good decision

How does FIDO work?



FIDO Experiences

ONLINE AUTH REQUEST

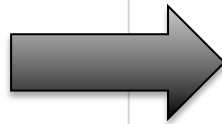
Local USER Verification

SUCCESS

PASSWORDLESS EXPERIENCE (UAF standards)



Transaction Detail



Show a biometric or PIN

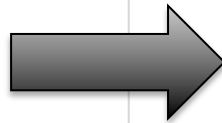


Done

SECOND FACTOR EXPERIENCE (U2F standards)



Login & Password



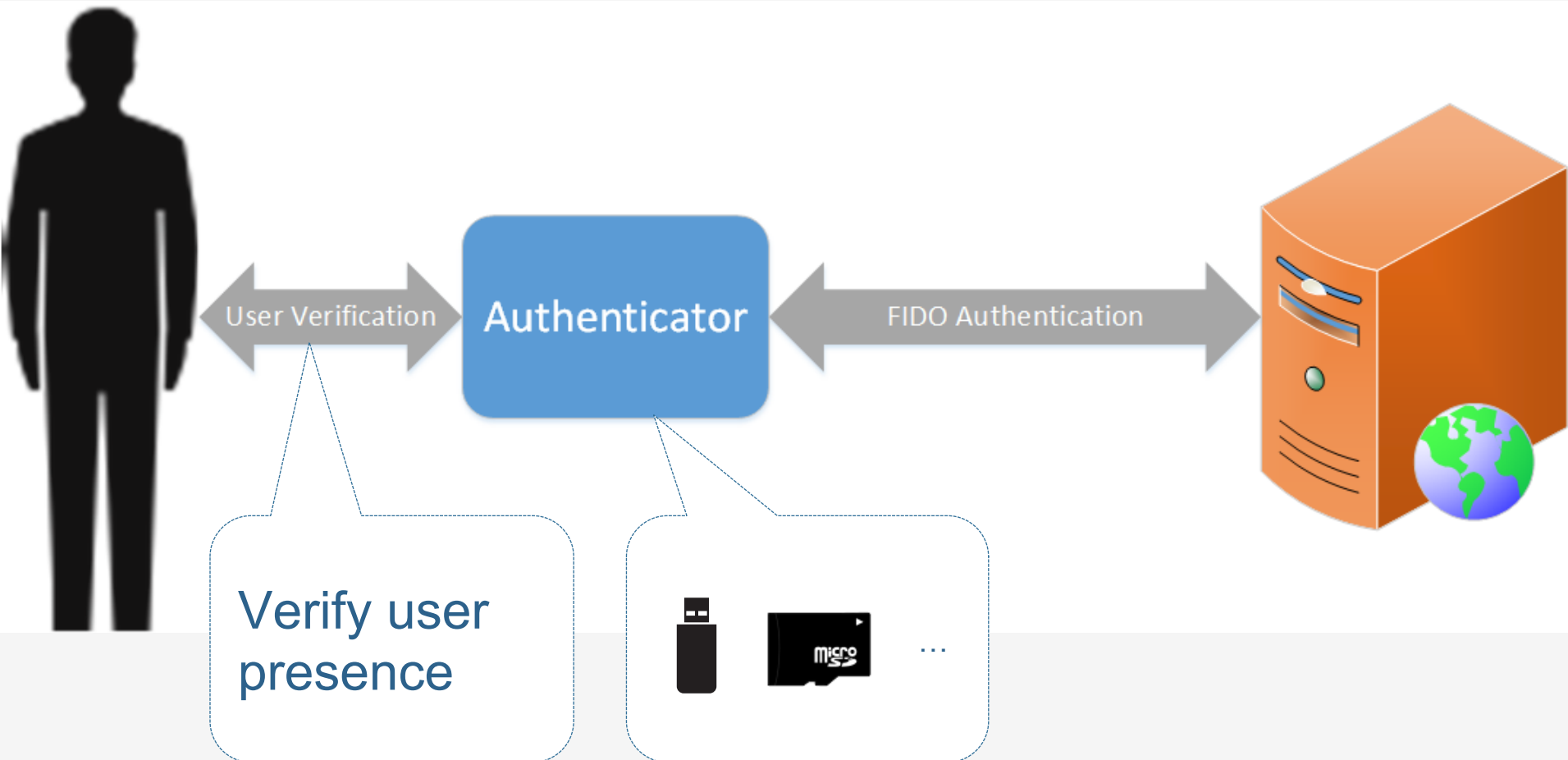
Insert Dongle, Press button



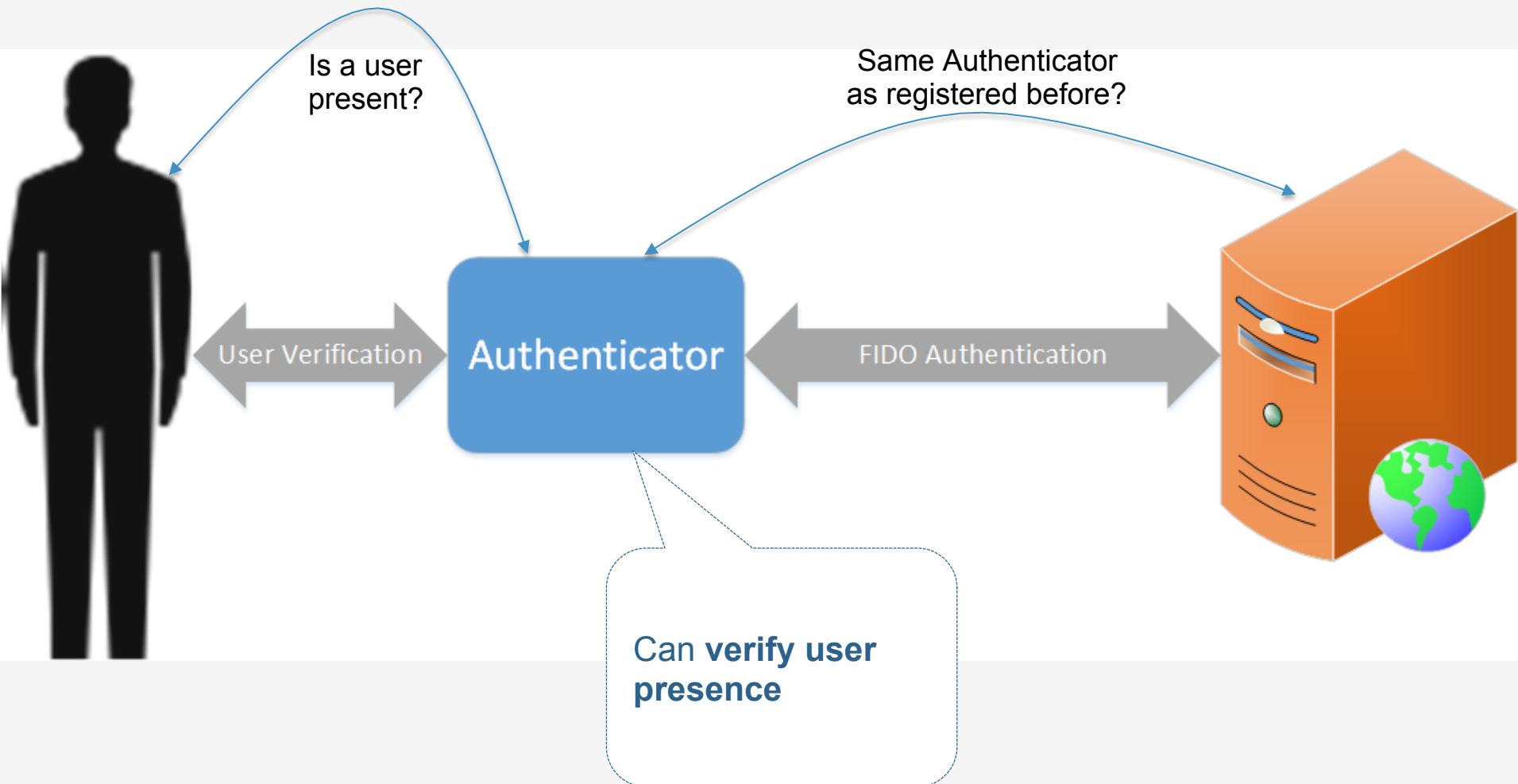
Done

FIDO Universal 2nd Factor (U2F)

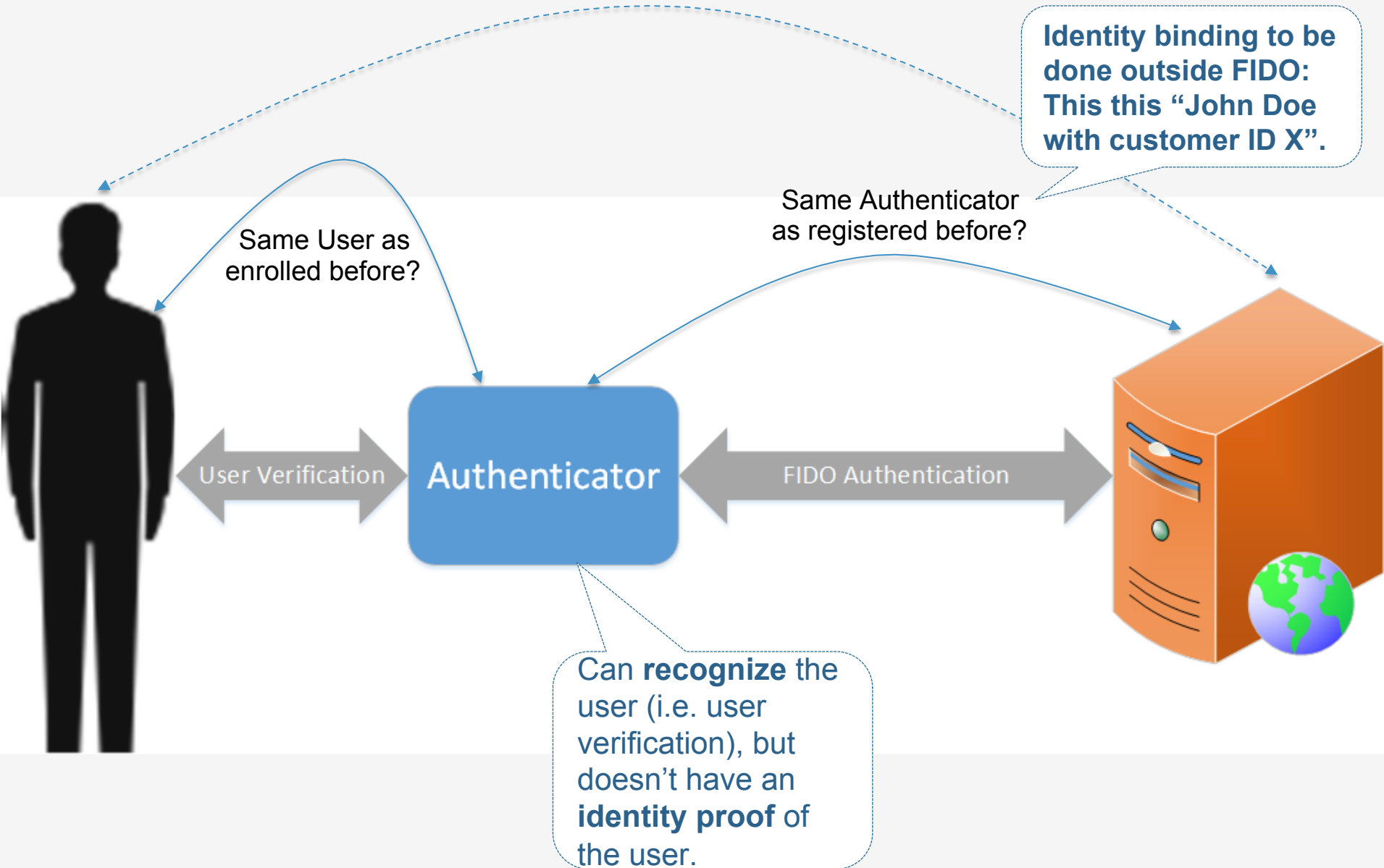
How does FIDO U2F work?



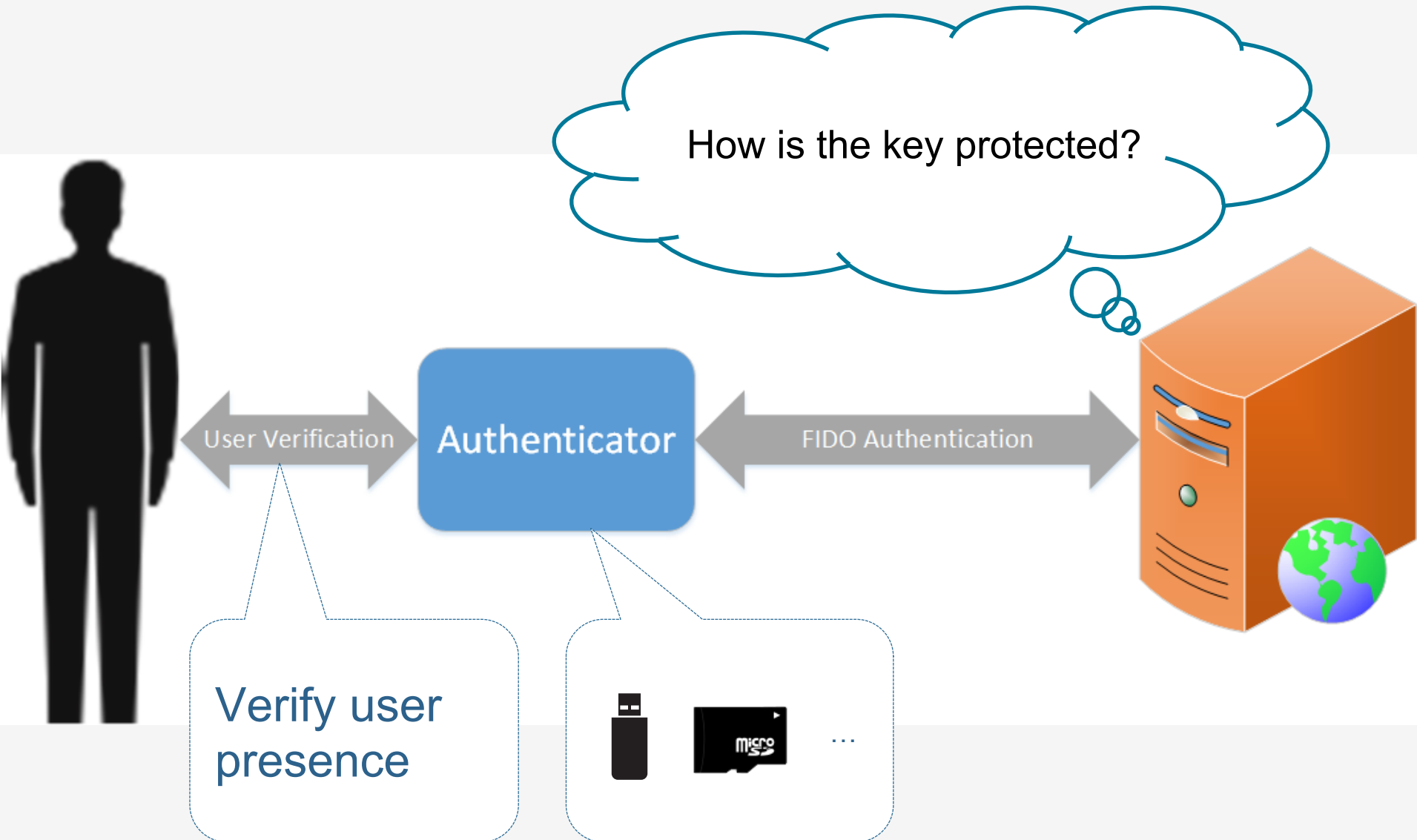
How does FIDO U2F work?



How does FIDO UAF work?



How does FIDO U2F work?

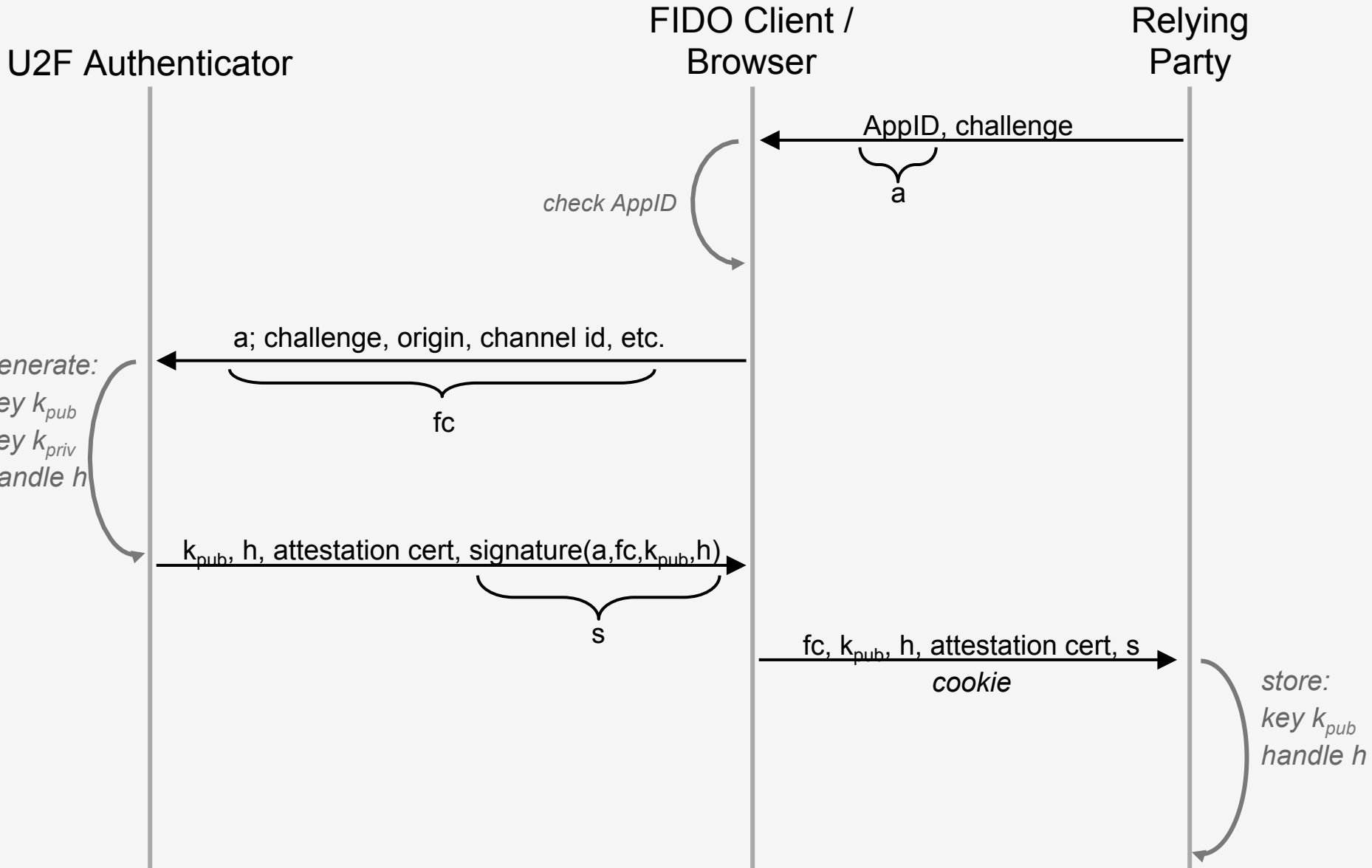


U2F Protocol

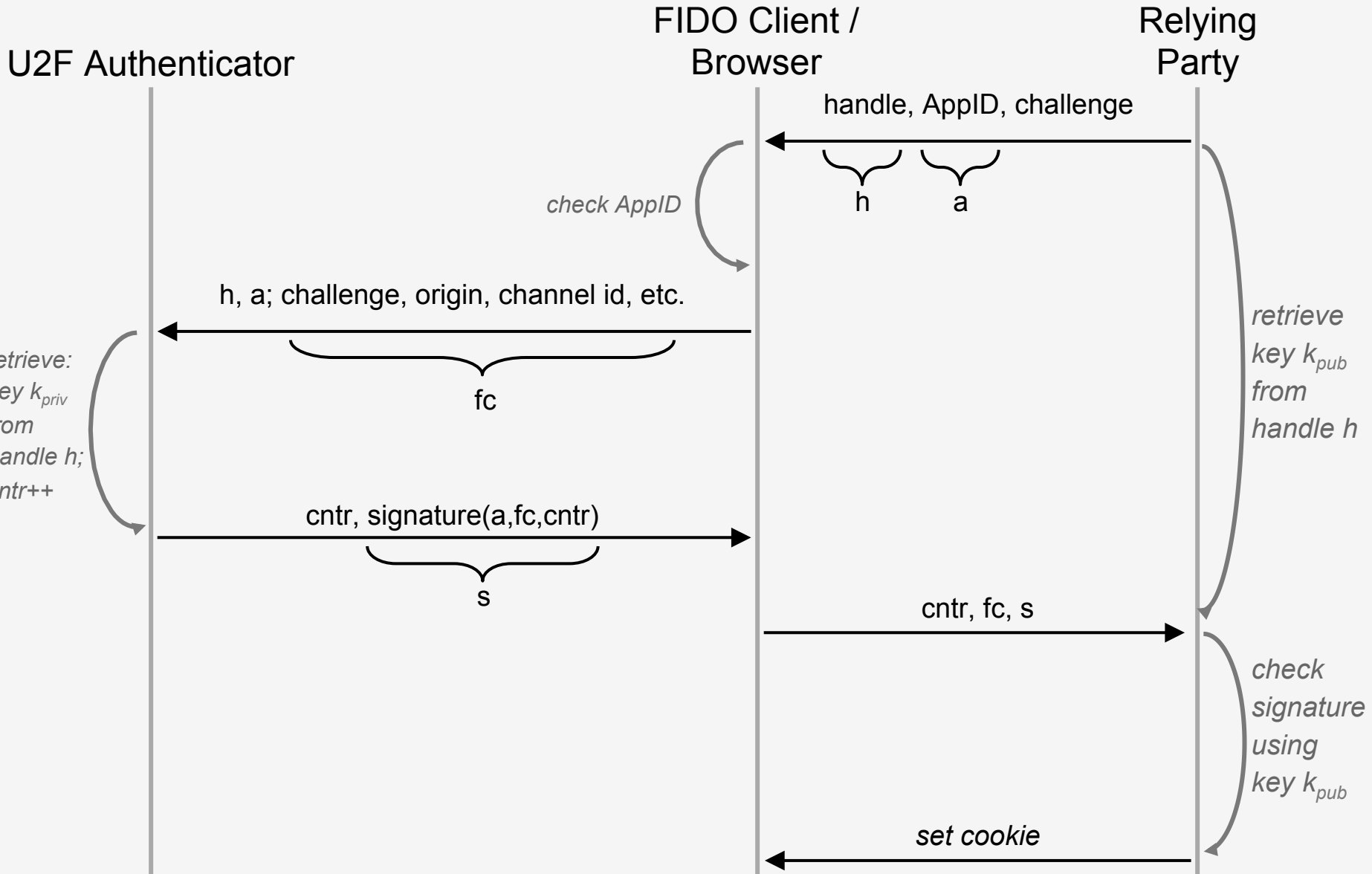
- **Core idea: Standard public key cryptography:**
 - User's device mints new key pair, gives public key to server
 - Server asks user's device to sign data to verify the user.
 - **One device, many services, "bring your own device" enabled**
- **Lots of refinement for this to be consumer facing:**
 - **Privacy:** Site specific keys, No unique ID per device
 - **Security:** No phishing, man-in-the-middles
 - **Trust:** Verify who made the device
 - **Pragmatics:** Affordable today, ride hardware cost curve down
 - **Speed for user:** Fast crypto in device (Elliptic Curve)

Think "Smartcard re-designed for modern consumer web"

U2F Registration



U2F Authentication



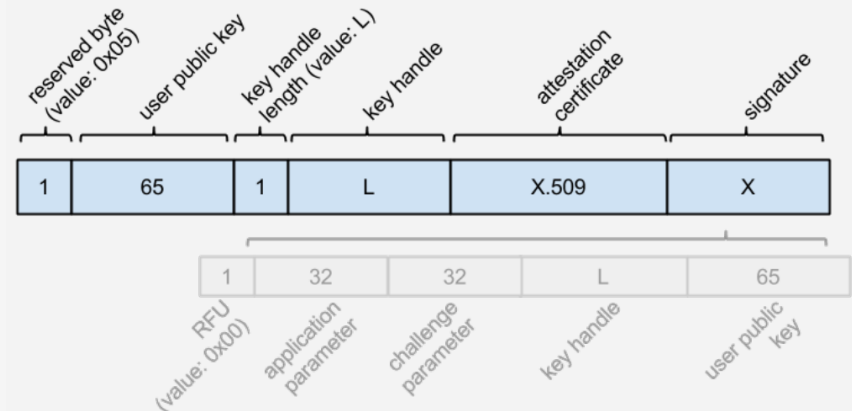
User Presence API: Registration

```
{
  "typ": "register",
  "challenge": "KSDJsdASAS-AIS_AsS",
  "cid_pubkey": {
    "kty": "EC",
    "crv": "P-256",
    "x": "HzQwlfXX7Q4S5MtCRMzPO9tOyWjBqRl4tJ8",
    "y": "XVguGFLIZx1fXg375hi4-7-BxhMljw42Ht4"
  },
  "origin": "https://accounts.google.com"
}
```

‘app_
callba

s://www.google.com/facets.json’},

```
callback = function(response) {
  sendToServer(
    response[‘clientId’],
    response[‘token’],
  };
```



User Presence API: Auth.

```
{
  "typ": "authenticate",
  "challenge": "KSDJsdASAS-AIS_Ass",
  "cid_pubkey": {
    "kty": "EC",
    "crv": "P-256",
    "x": "HzQwlfXX7Q4S5MtCRMzP09t0yWjBqRl4tJ8",
    "y": "XVguGFLIZx1fXg375hi4-7-BxhMljw42Ht4"
  },
  "origin": "https://accounts.google.com"
}
```

/facets.json',

'-sadsAJDKLSAD'},

call

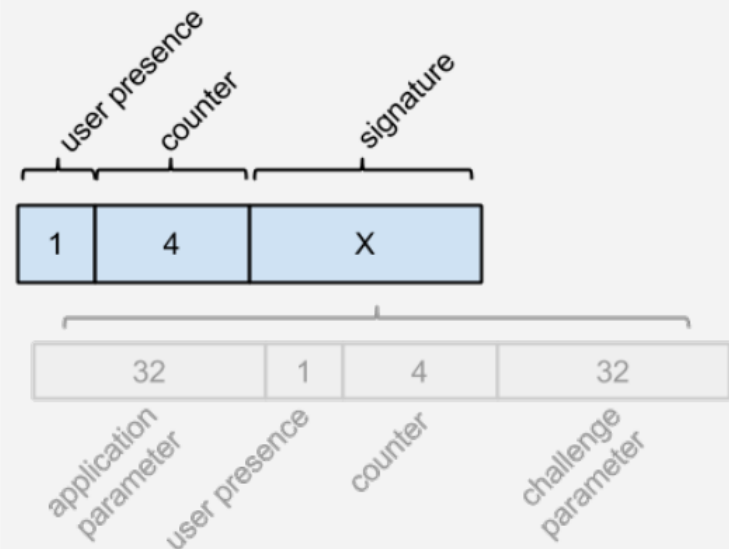
callback = function(response)

sendToServer(

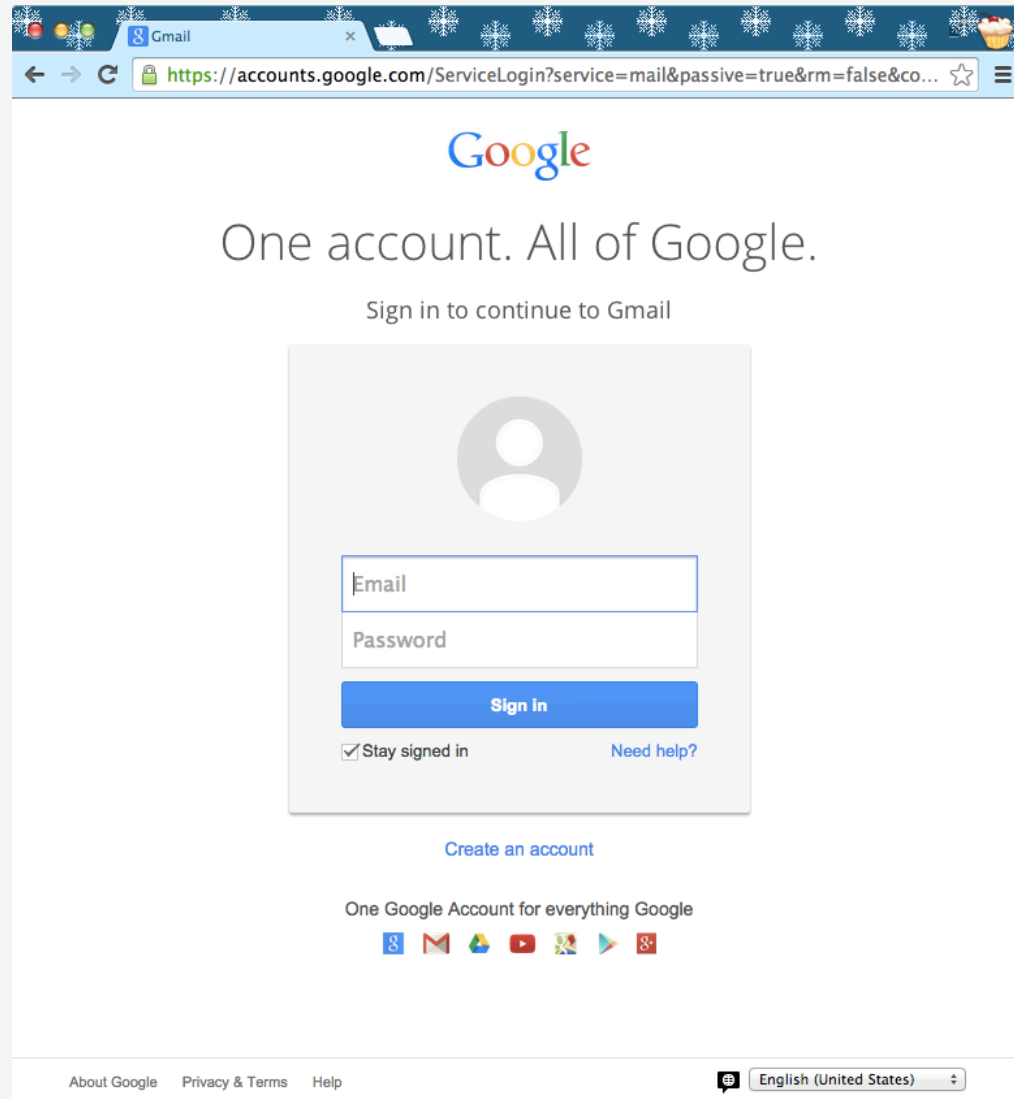
response['clientId'],

response['token'],

};



Authentication Example



A screenshot of the Gmail login page in a web browser. The browser's address bar shows the URL `https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=false&co...`. The page features the Google logo at the top, followed by the text "One account. All of Google." and "Sign in to continue to Gmail". Below this is a large gray box containing a circular profile icon placeholder, two input fields labeled "Email" and "Password", a blue "Sign in" button, a checked "Stay signed in" checkbox, and a "Need help?" link. At the bottom of the gray box is a "Create an account" link. Below the gray box, it says "One Google Account for everything Google" and shows icons for various Google services. The footer includes links for "About Google", "Privacy & Terms", and "Help", along with a language selector set to "English (United States)".

Gmail

<https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=false&co...>

Google

One account. All of Google.

Sign in to continue to Gmail

Email

Password

Sign in

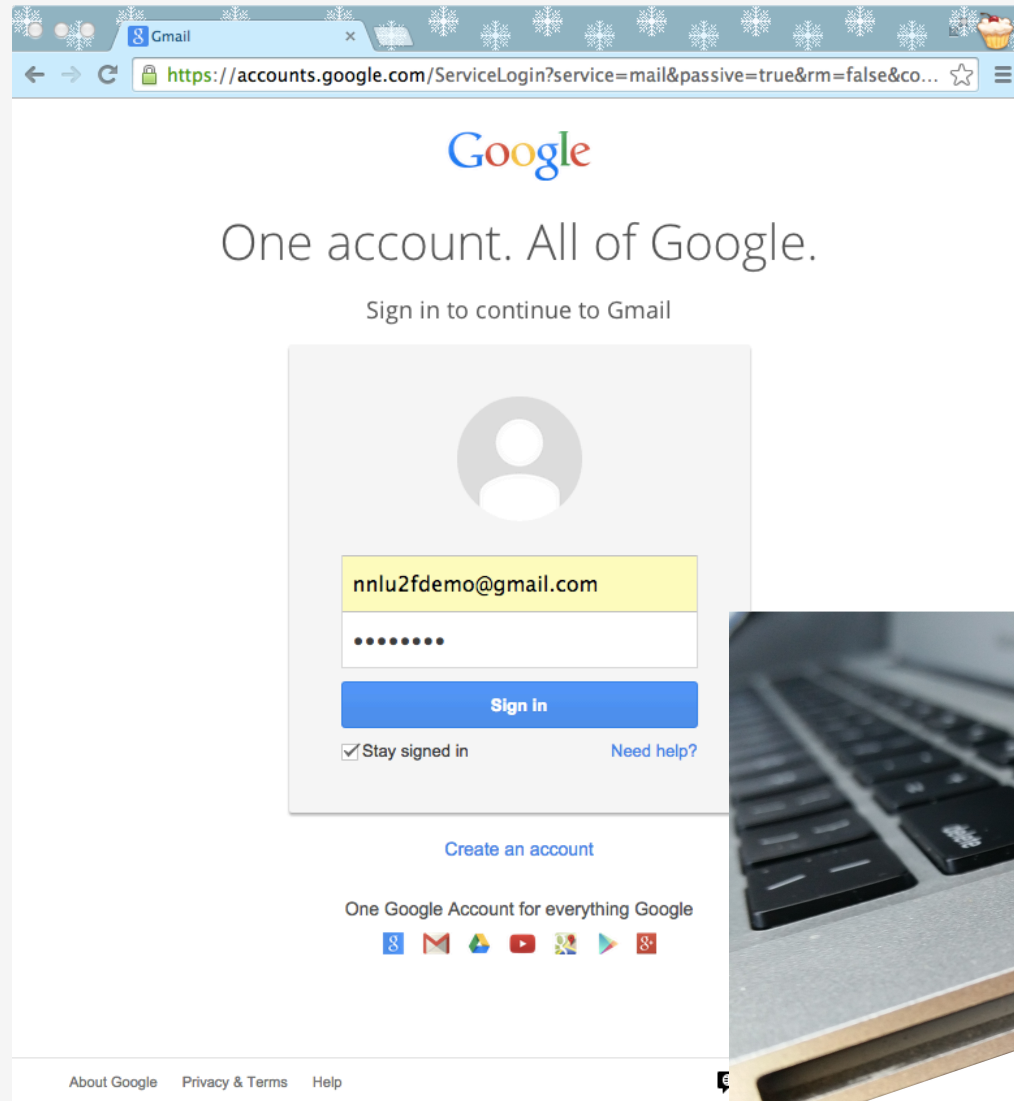
☒ Stay signed in [Need help?](#)

[Create an account](#)

One Google Account for everything Google

[About Google](#) [Privacy & Terms](#) [Help](#) [English \(United States\)](#)

Authentication Example




Gmail

← → ↻ <https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=false&co...> ☆ ☰

Google

One account. All of Google.

Sign in to continue to Gmail




[Sign in](#)

☒ Stay signed in [Need help?](#)

[Create an account](#)

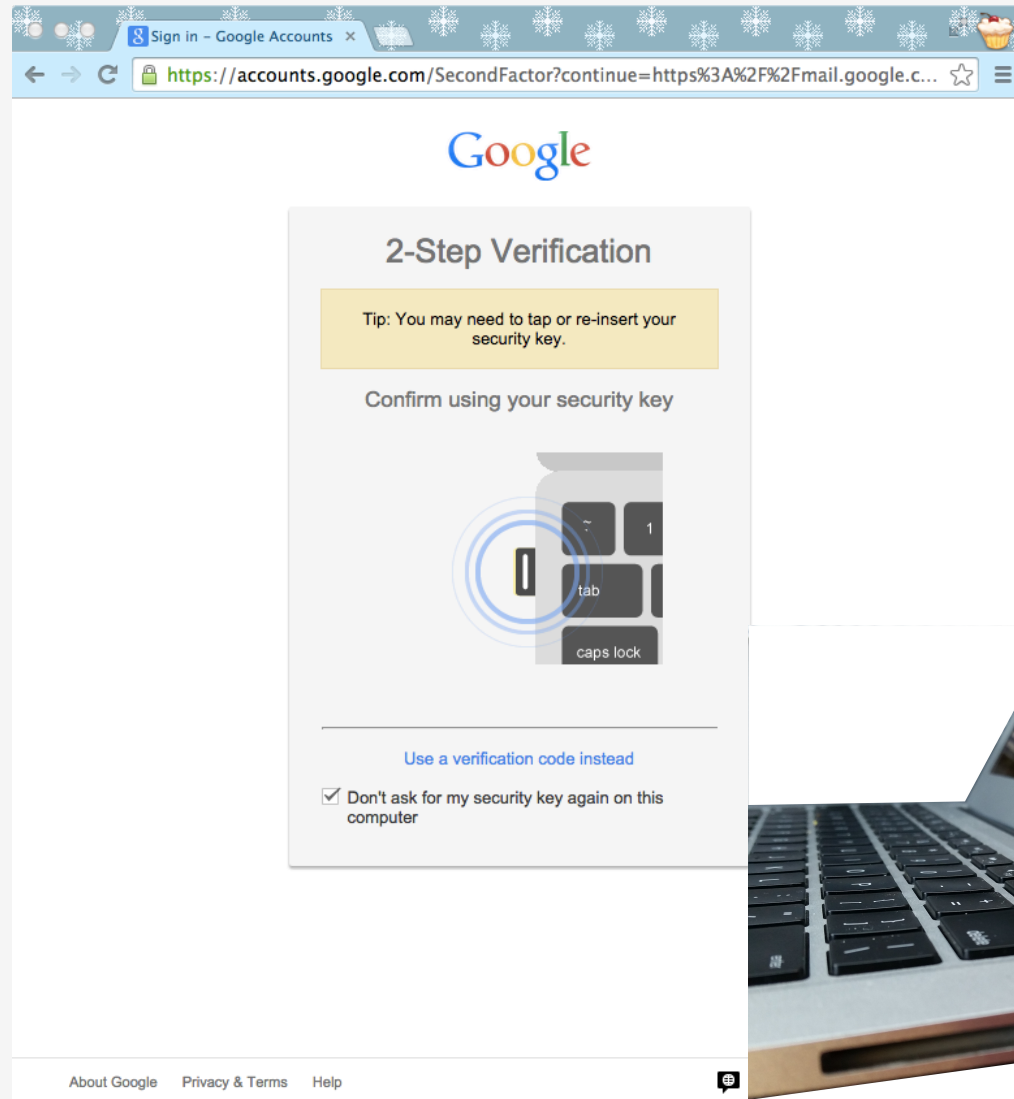
One Google Account for everything Google



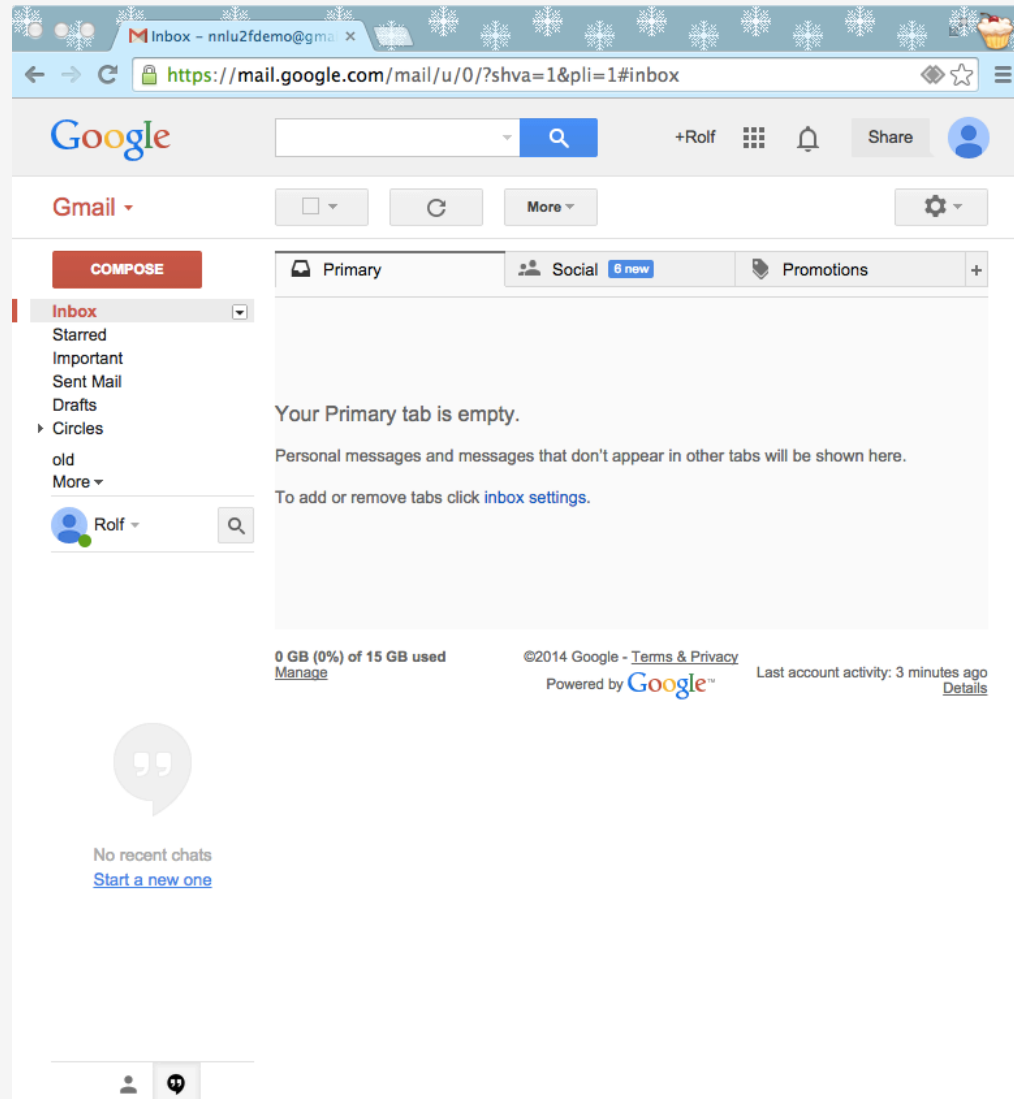
[About Google](#) [Privacy & Terms](#) [Help](#)



Authentication Example



Authentication Example



FIDO Universal Authentication Framework (UAF)

FIDO Experiences

ONLINE AUTH REQUEST

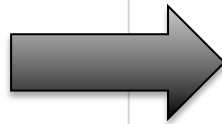
Local USER Verification

SUCCESS

PASSWORDLESS EXPERIENCE (UAF standards)



Transaction Detail



Show a biometric or PIN

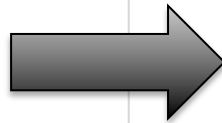


Done

SECOND FACTOR EXPERIENCE (U2F standards)



Login & Password

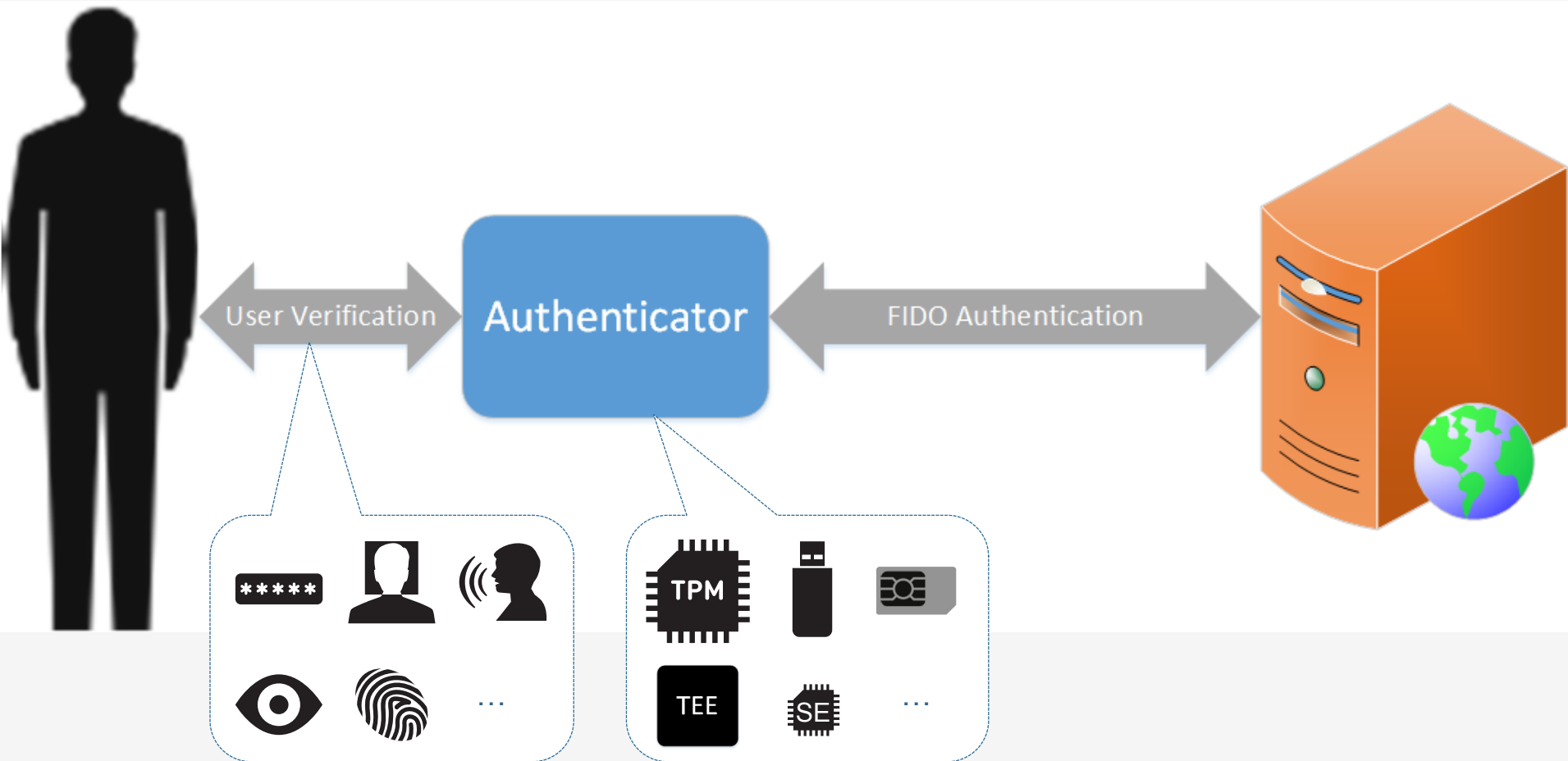


Insert Dongle, Press button

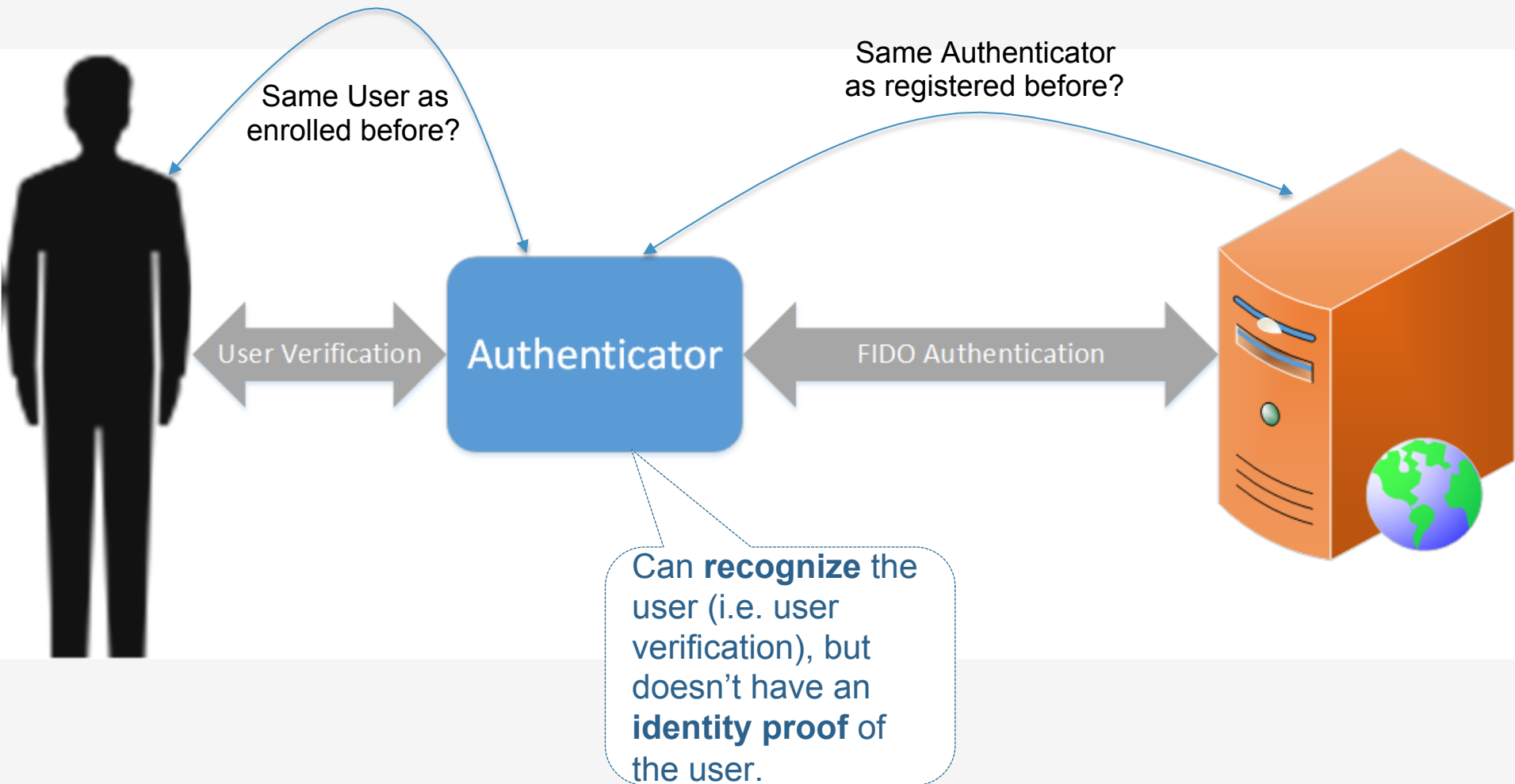


Done

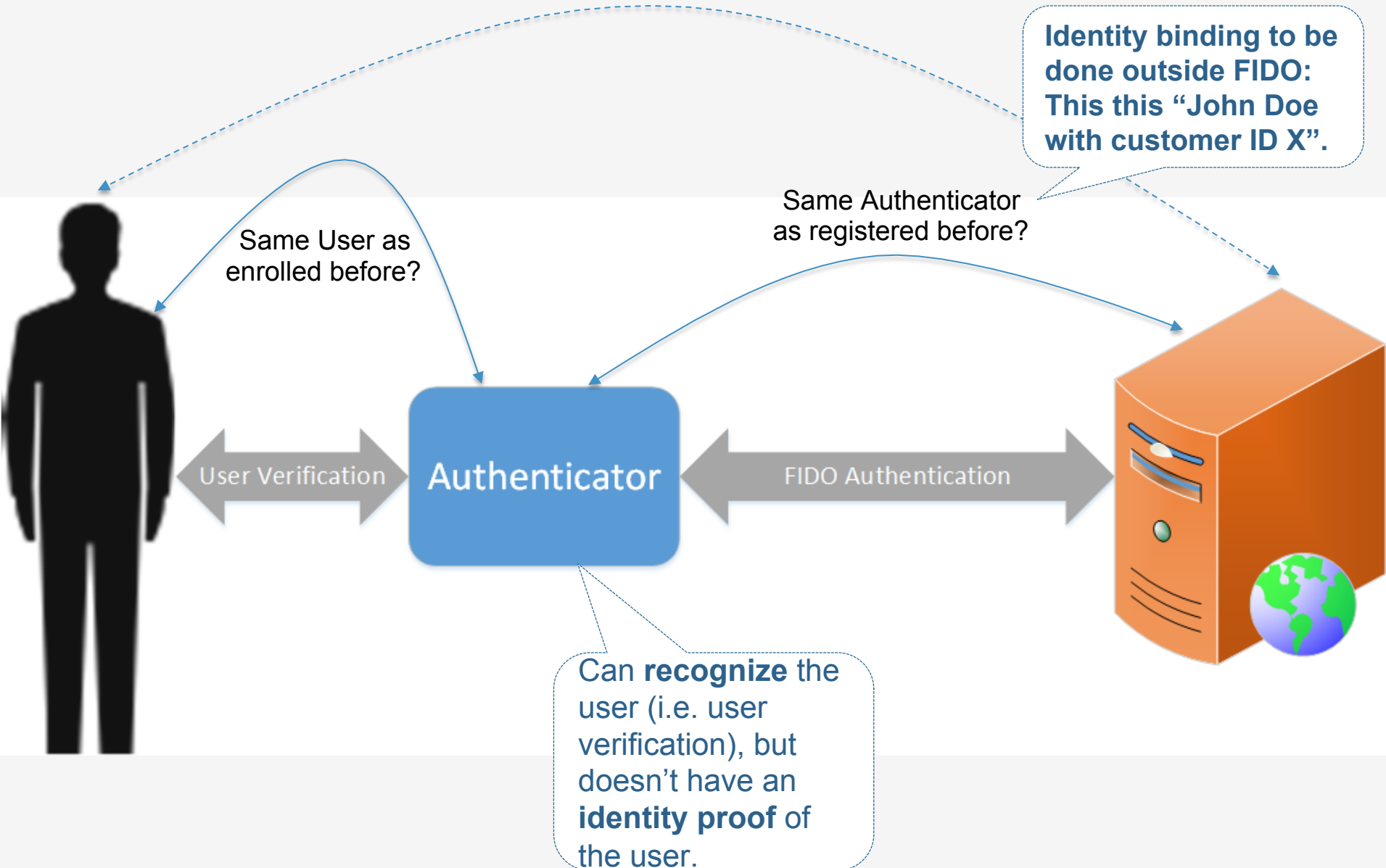
How does FIDO UAF work?



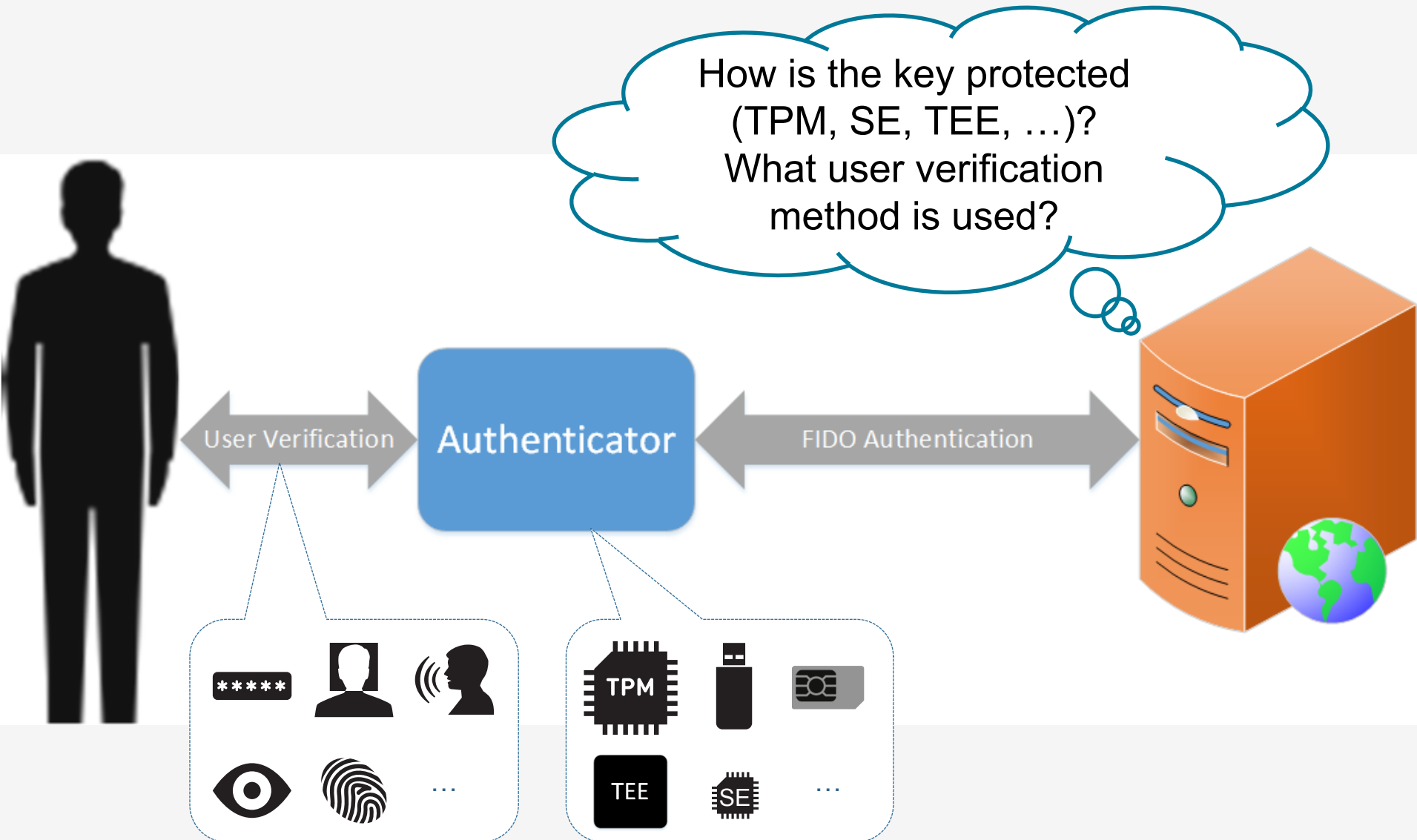
How does FIDO UAF work?



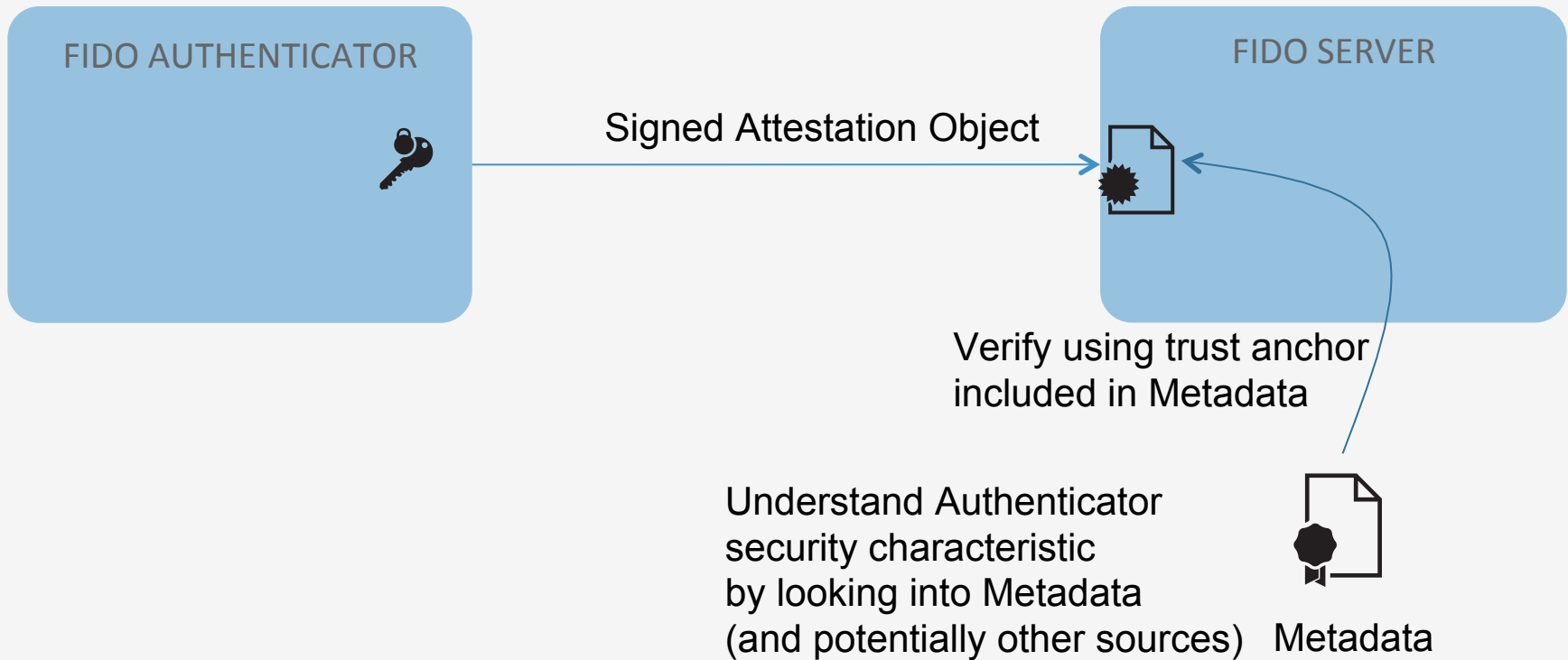
How does FIDO UAF work?



How does FIDO UAF work?

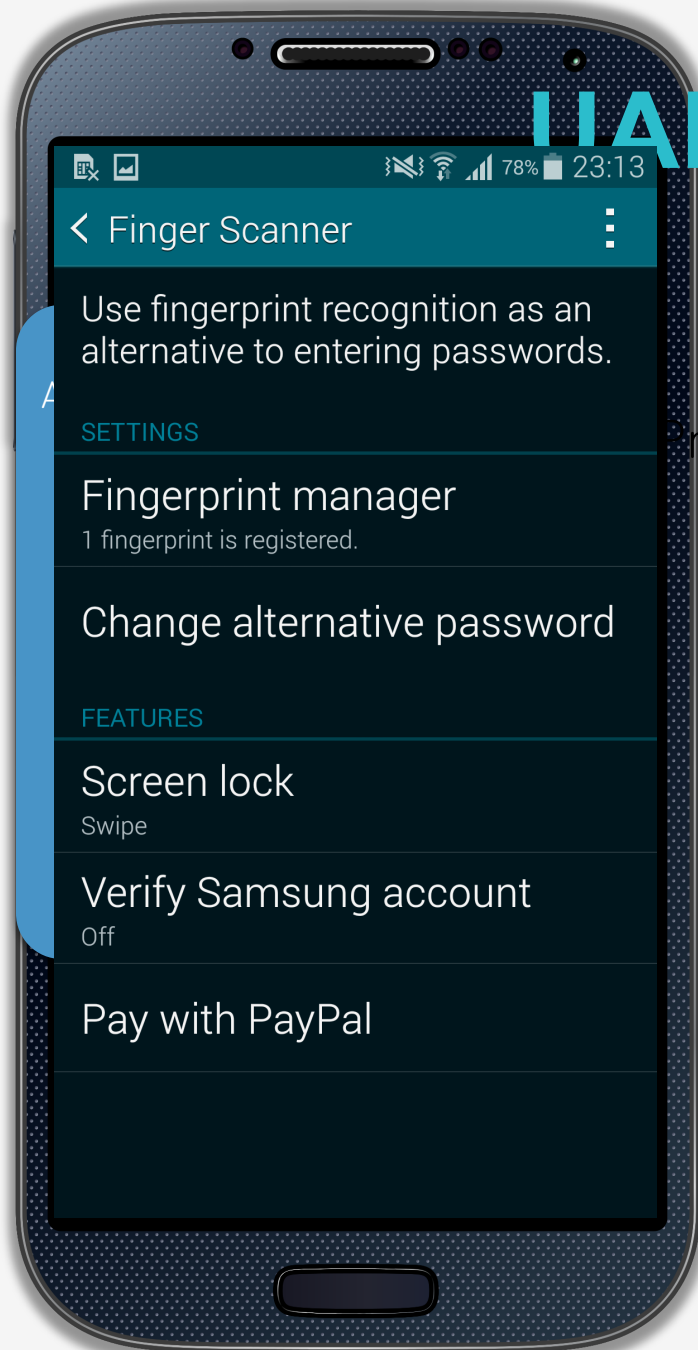


Attestation & Metadata



FIDO Registration

Relying Party

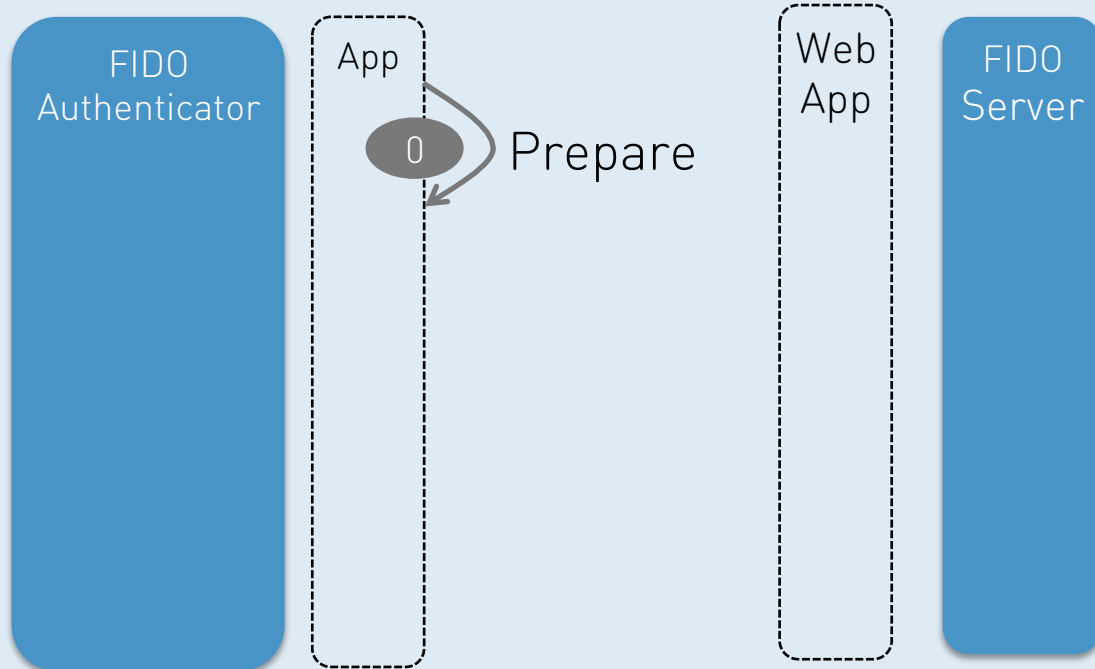
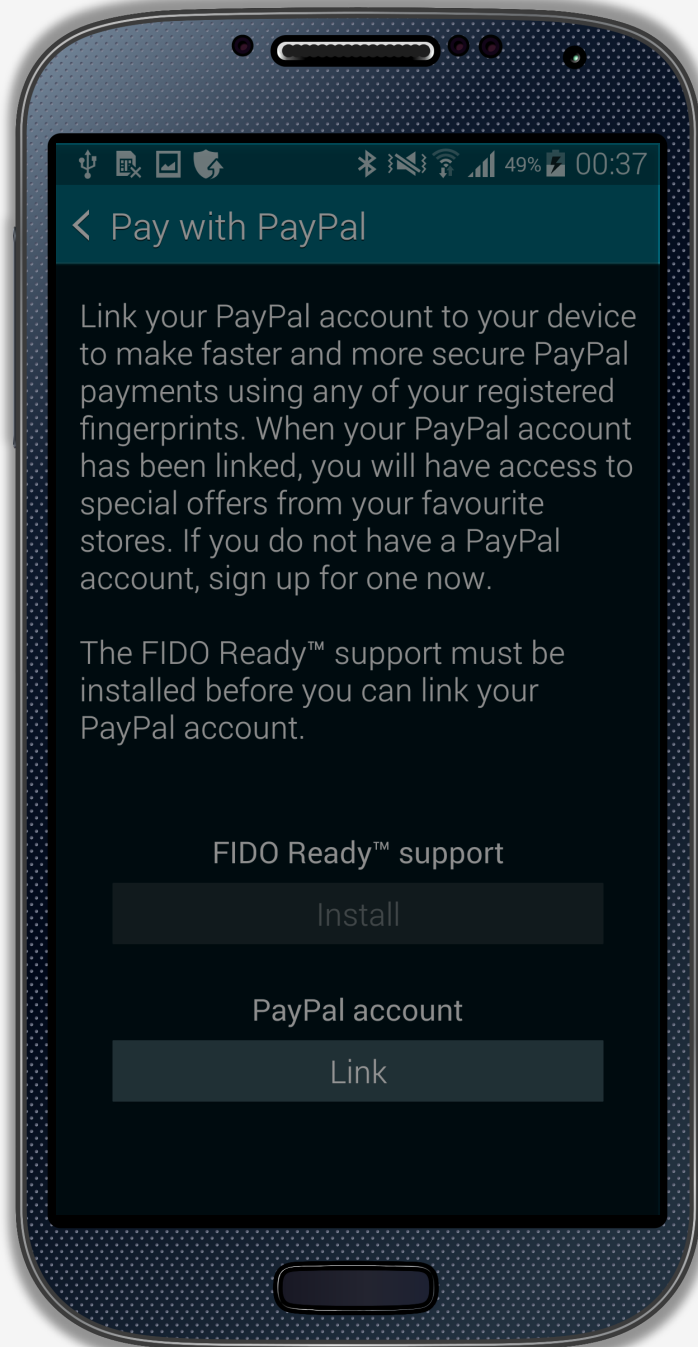


prepare

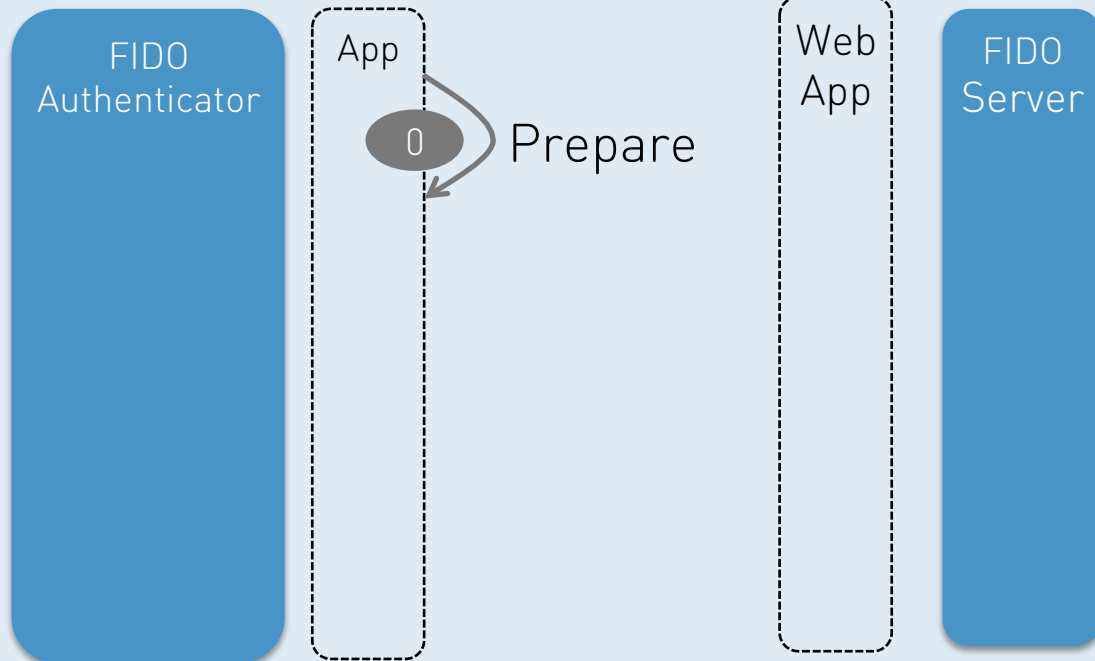
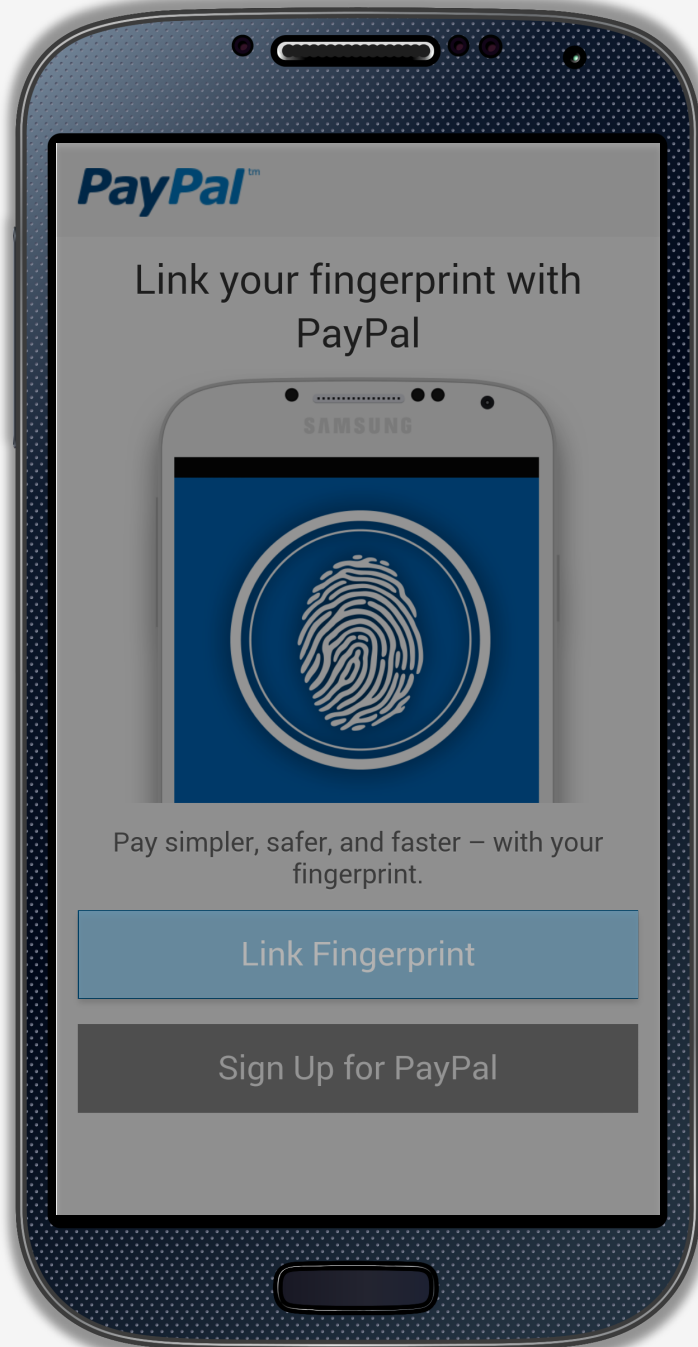
Web
App

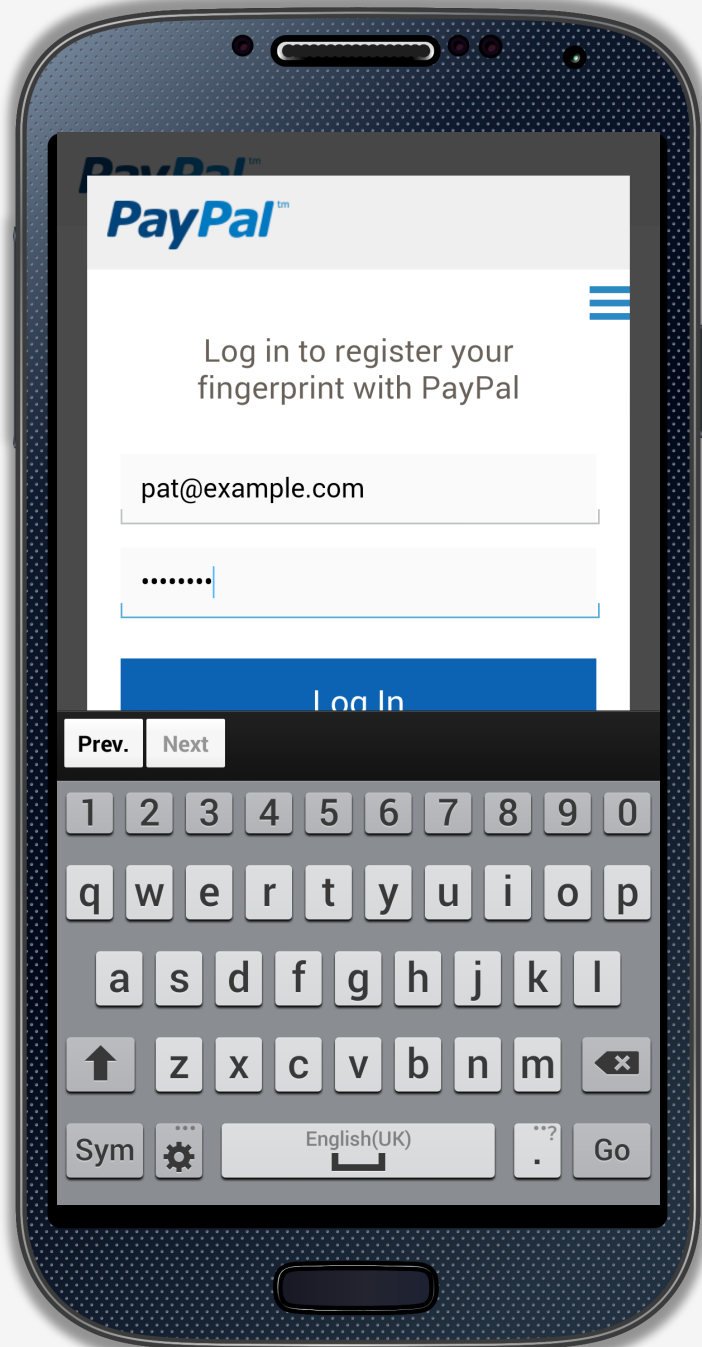
FIDO
Server

UAF Registration

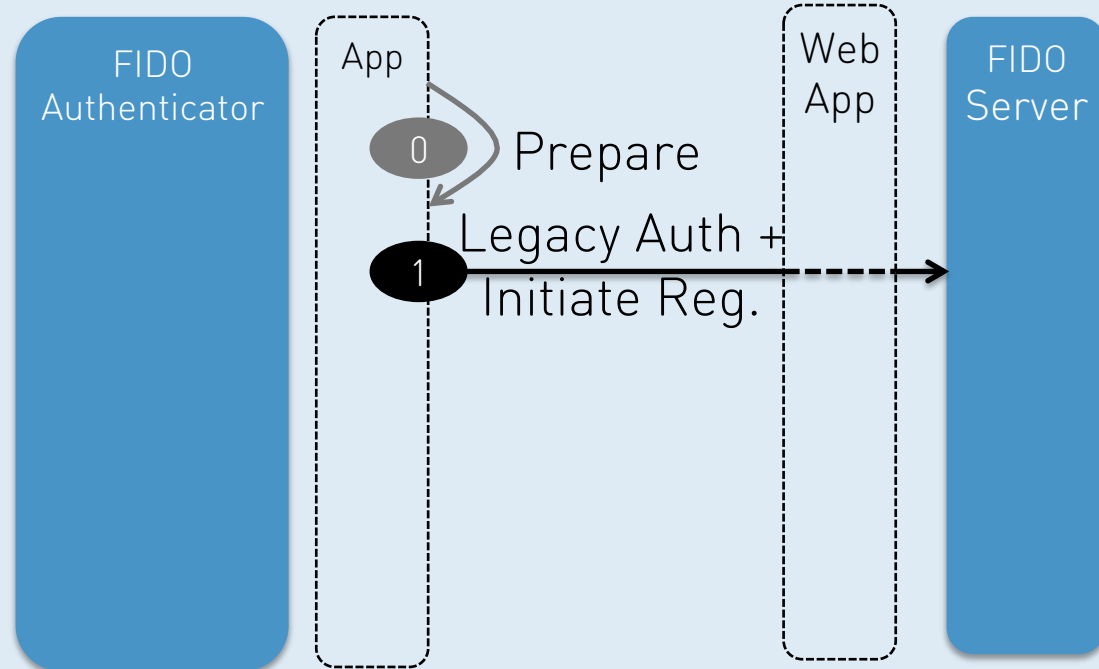


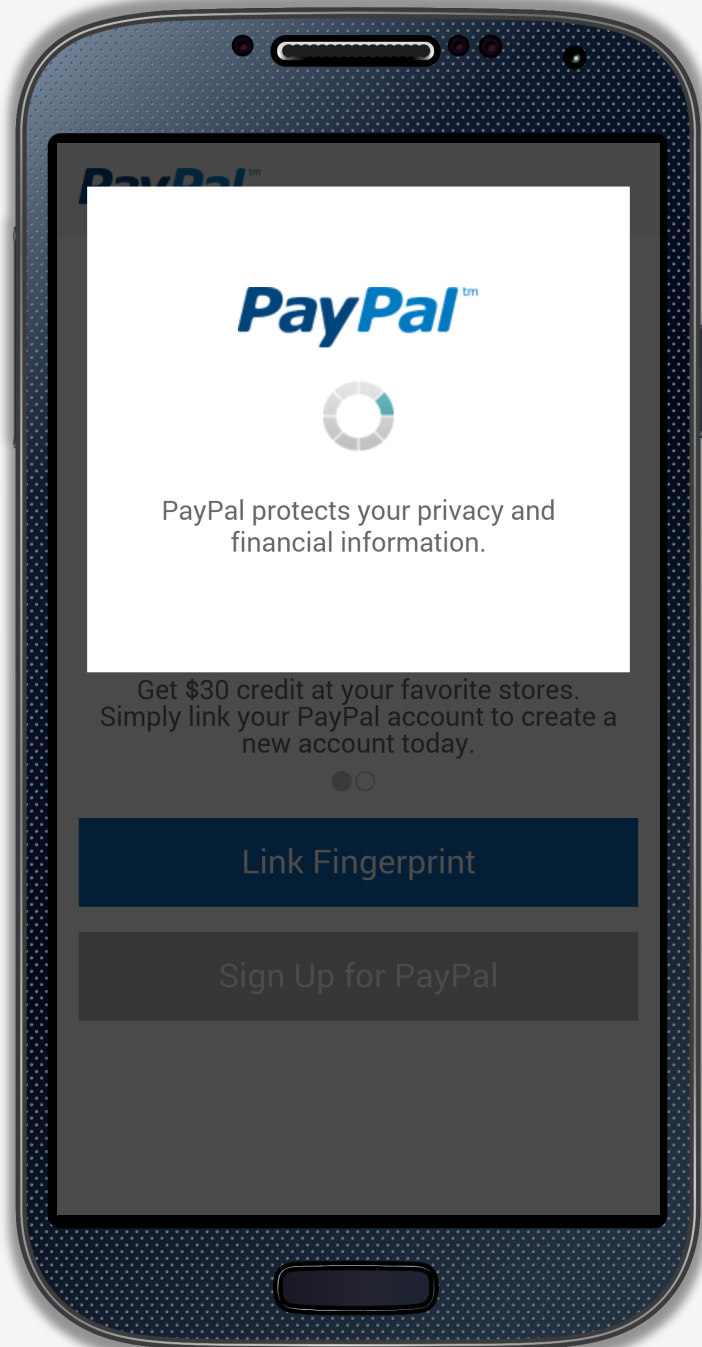
UAF Registration



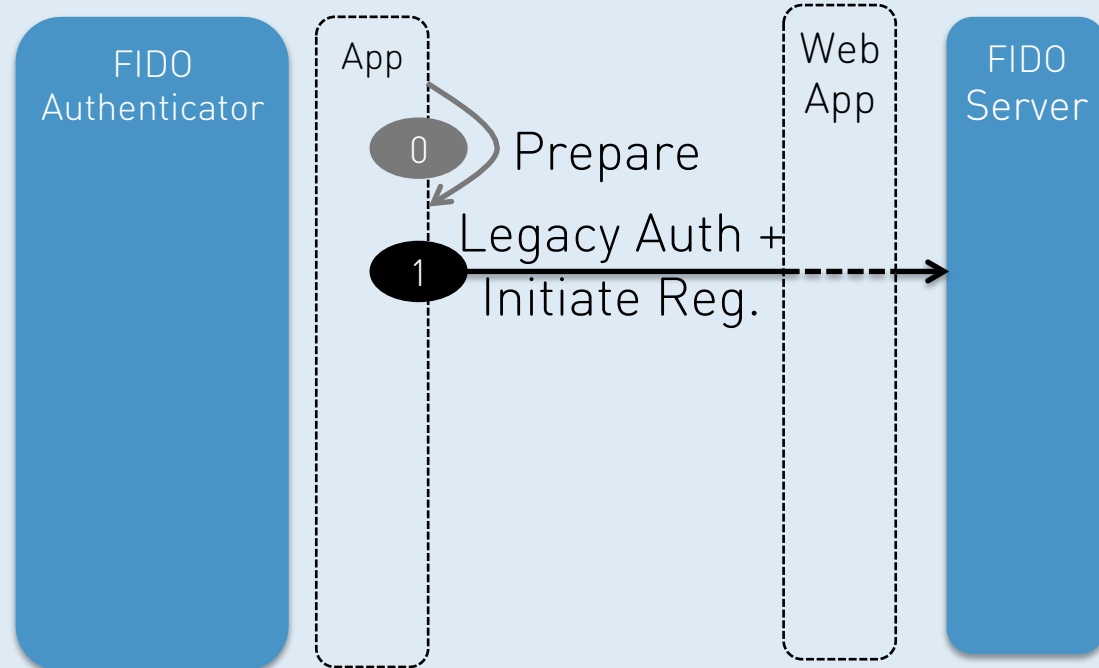


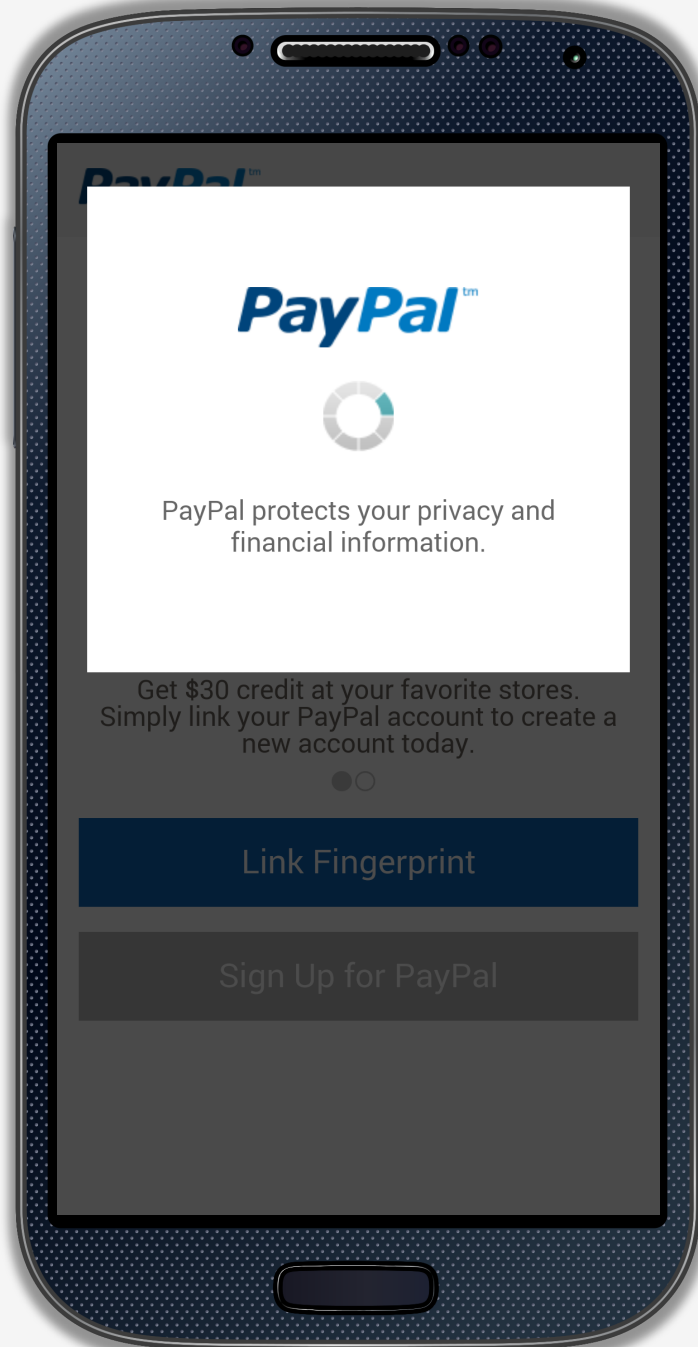
UAF Registration



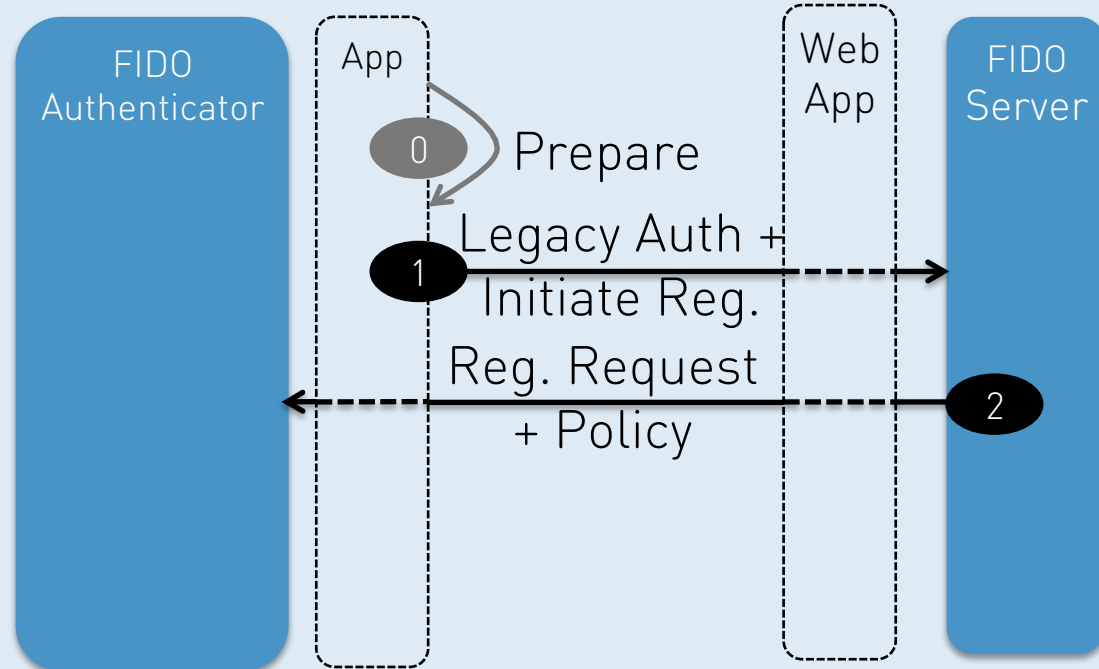


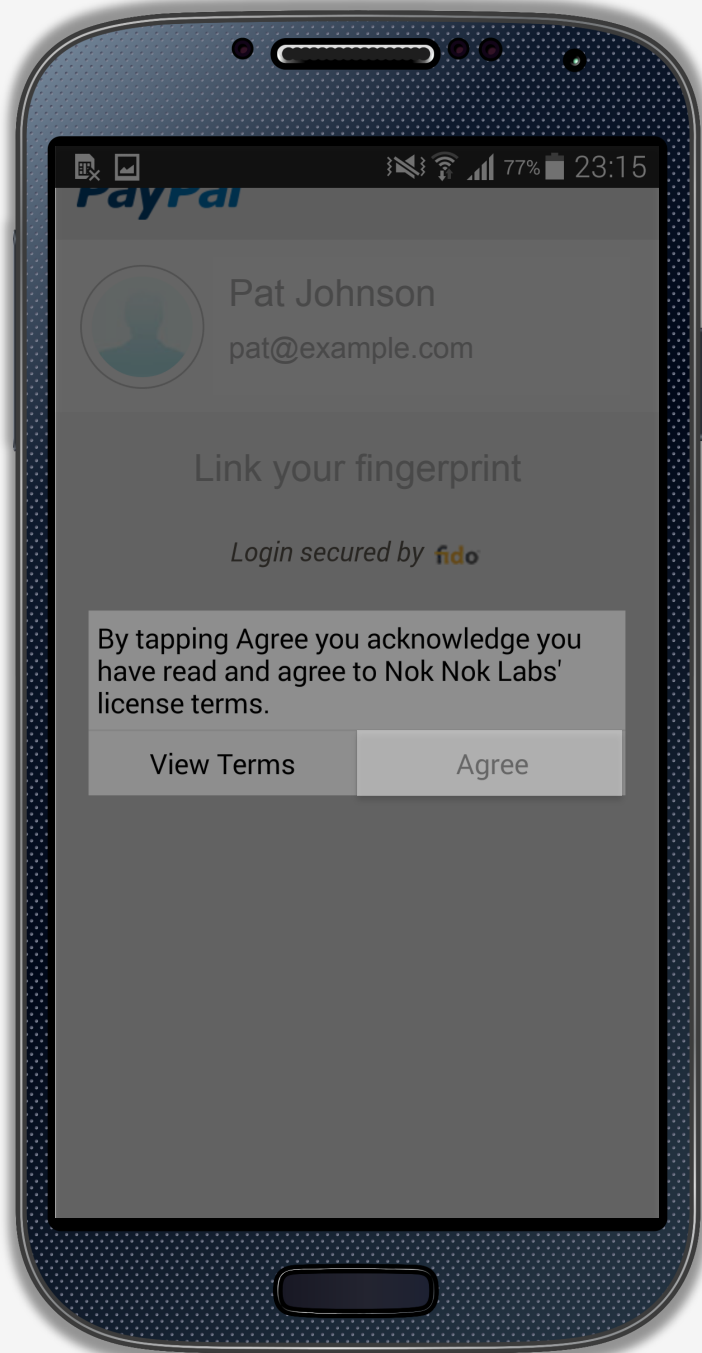
UAF Registration



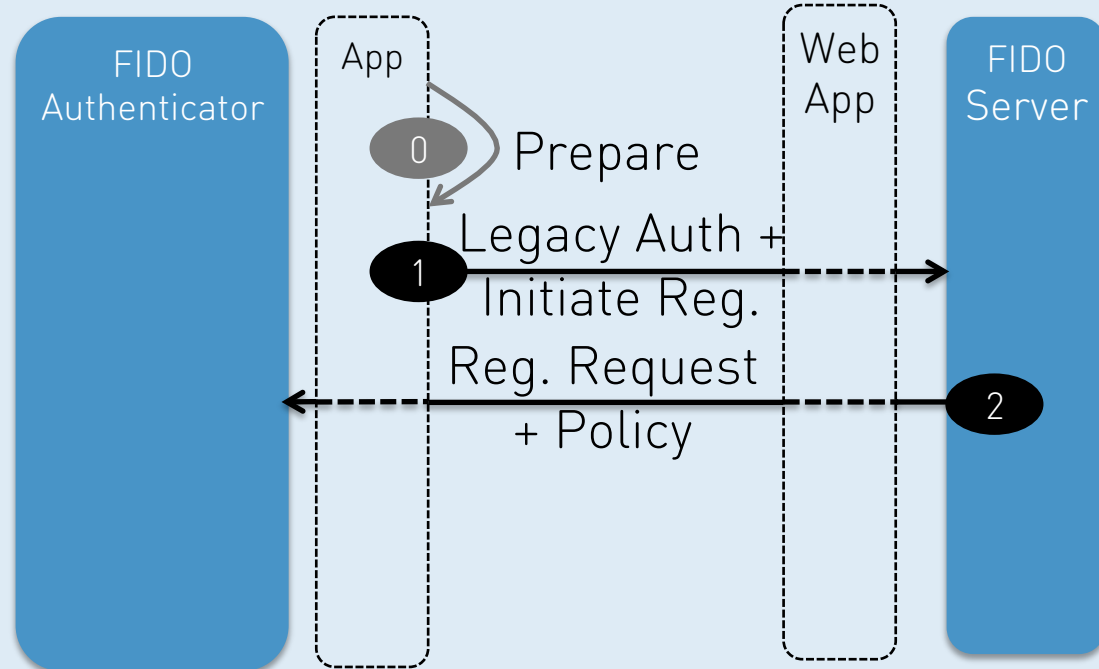


UAF Registration

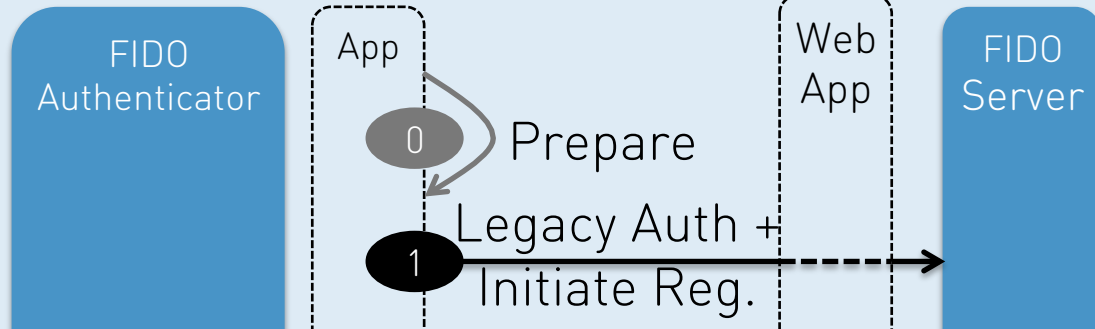
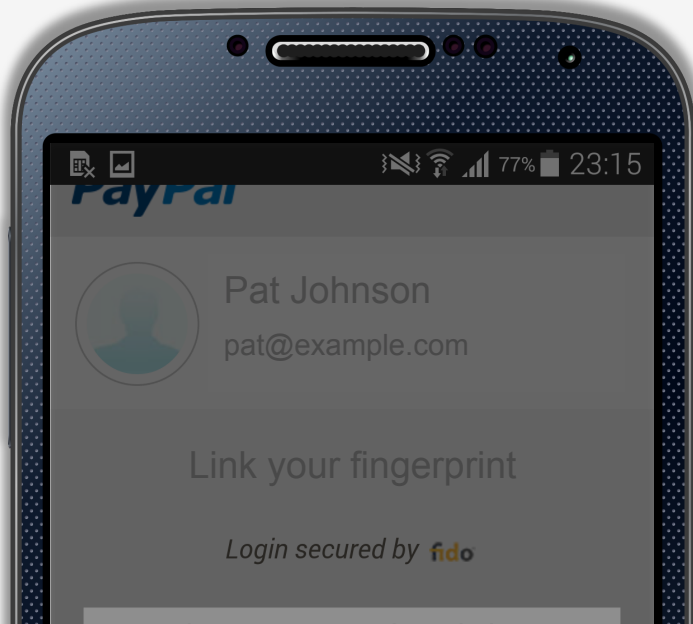




UAF Registration

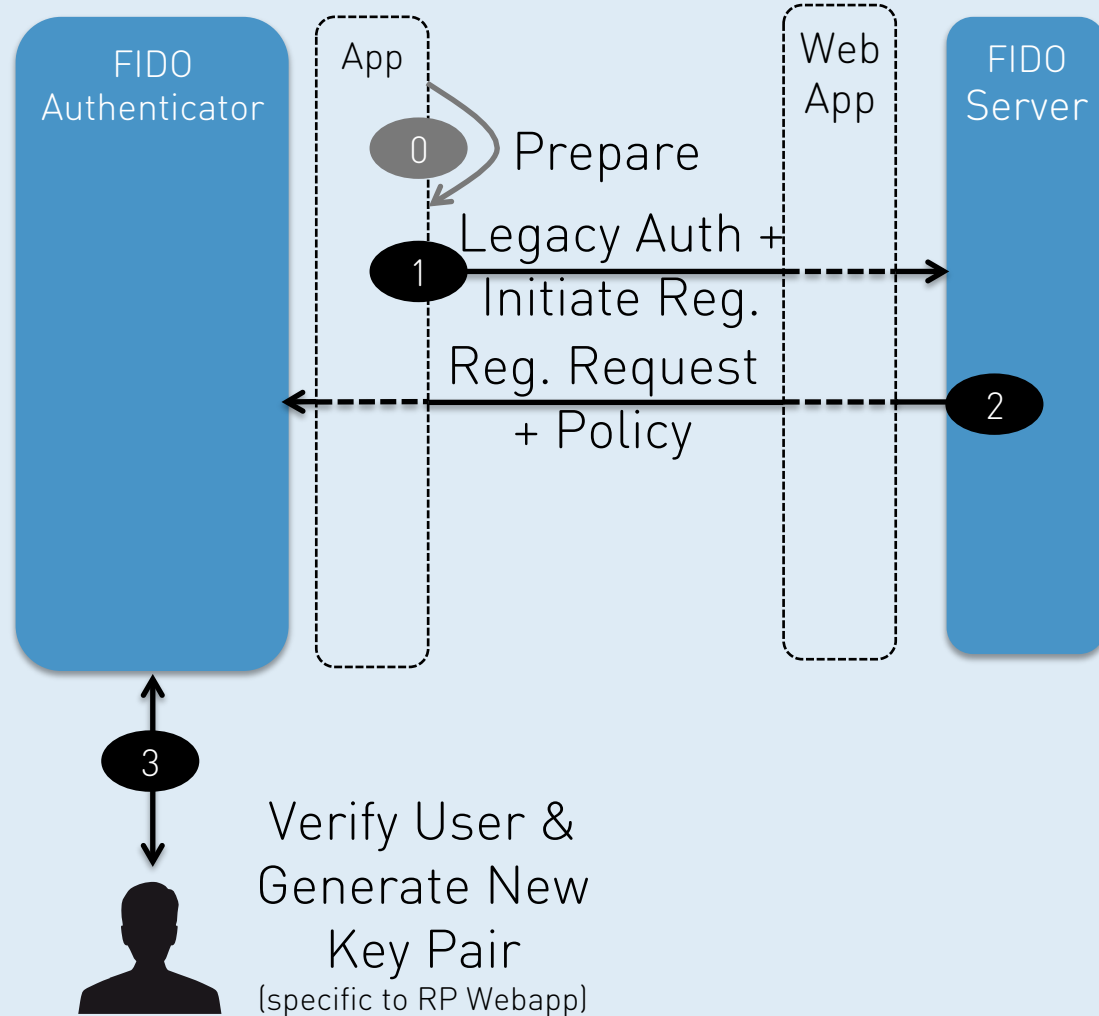
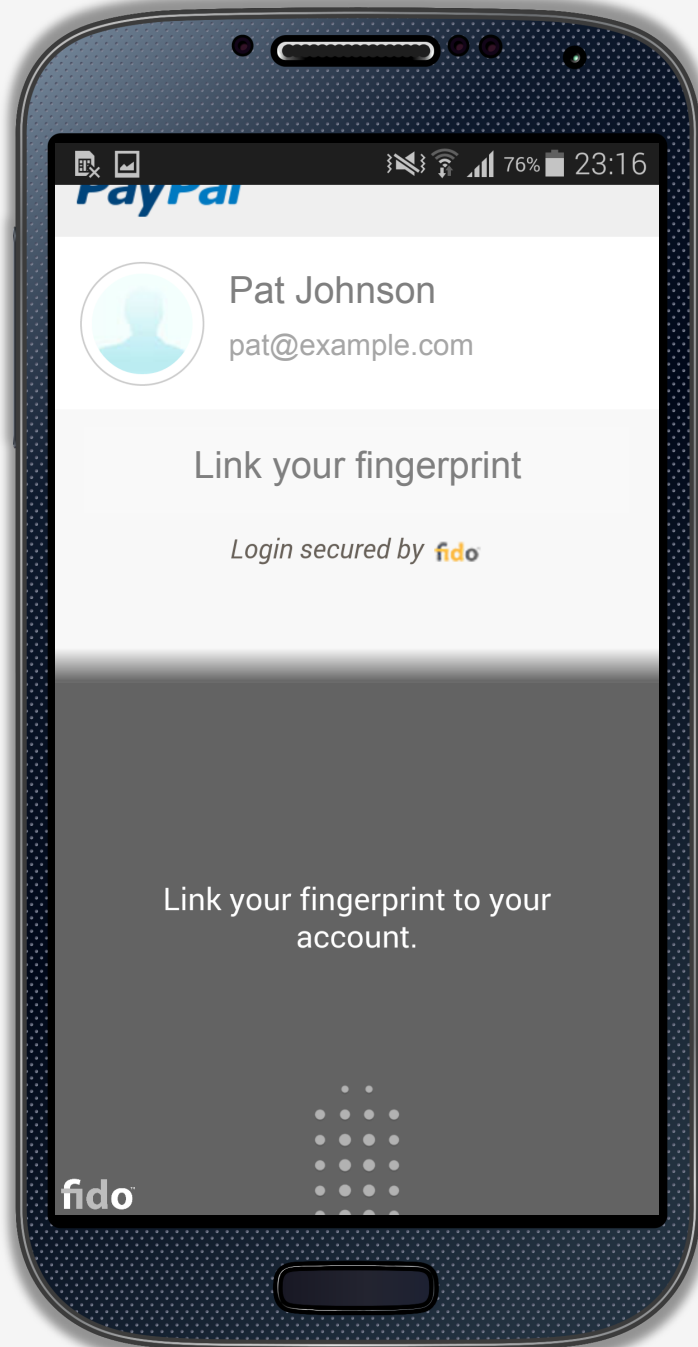


UAF Registration

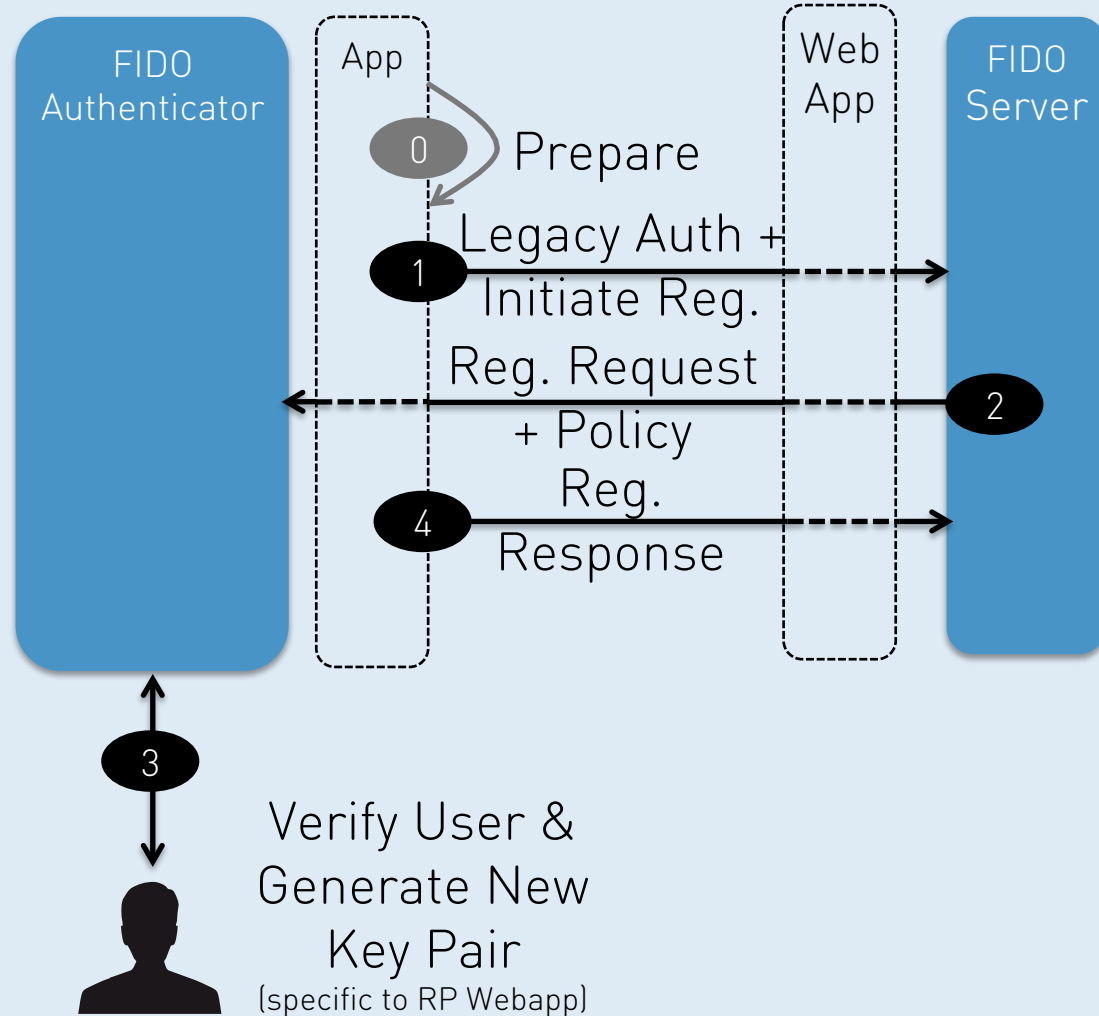
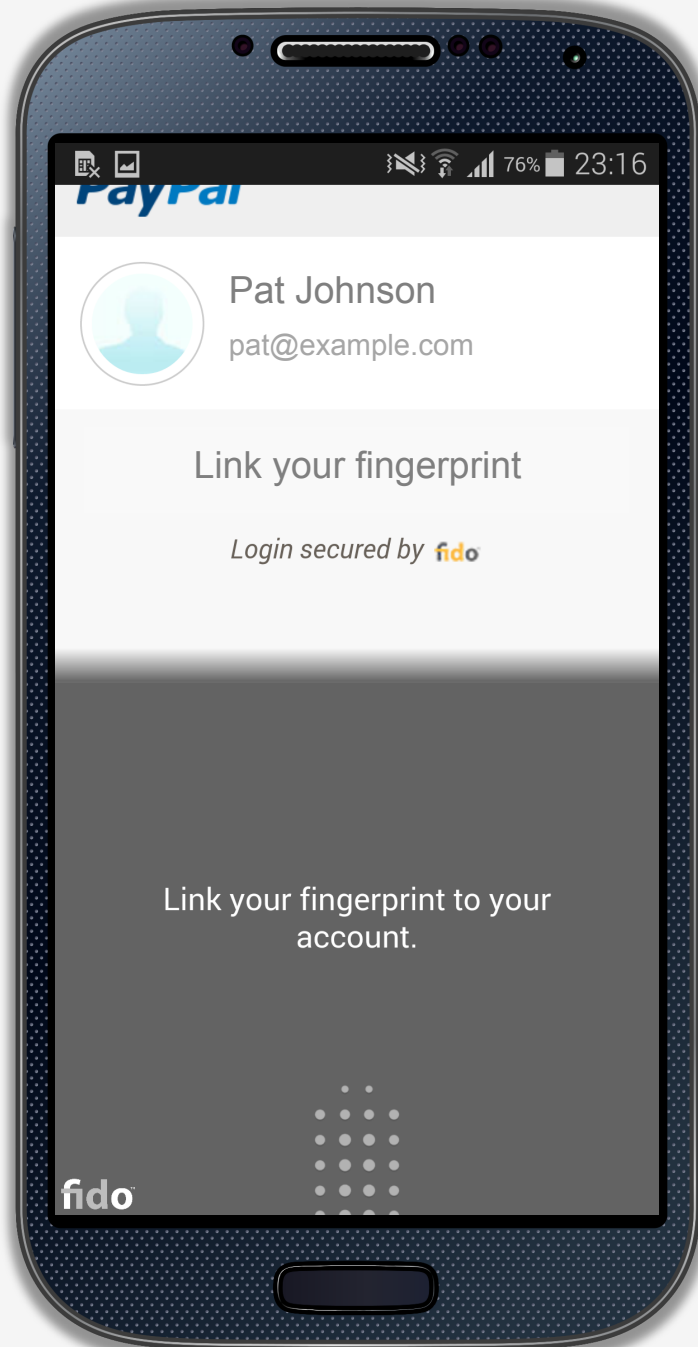


```
[{  
  "header": { "op": "Reg", "upv": "1.0", "appID": "https://mycorp.com/fido"},  
  "challenge": "qwudh827hddbawd8qbdqj3bduq3duq56t324zwasdq4wrt",  
  "username": "banking_personal",  
  "policy": {  
    "accepted": [[{  
      "authenticationFactor": 00000000000001ff,  
      "keyProtection": 000000000000000e,  
      "attachment": 00000000000000ff,  
      "secureDisplay": 000000000000001e,  
      "supportedSchemes": "UAFV1TLV"}]],  
    "disallowed": {"aaid": "1234#5678"}  
  }  
}]
```

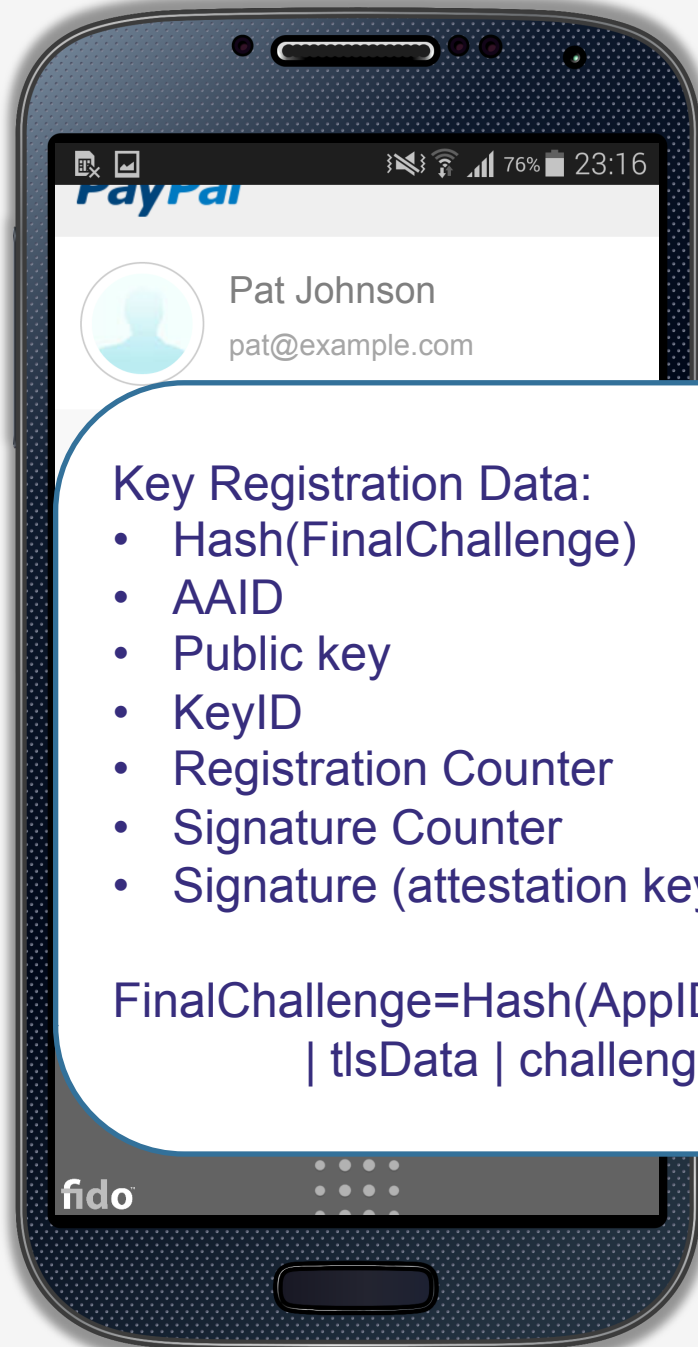
UAF Registration



UAF Registration



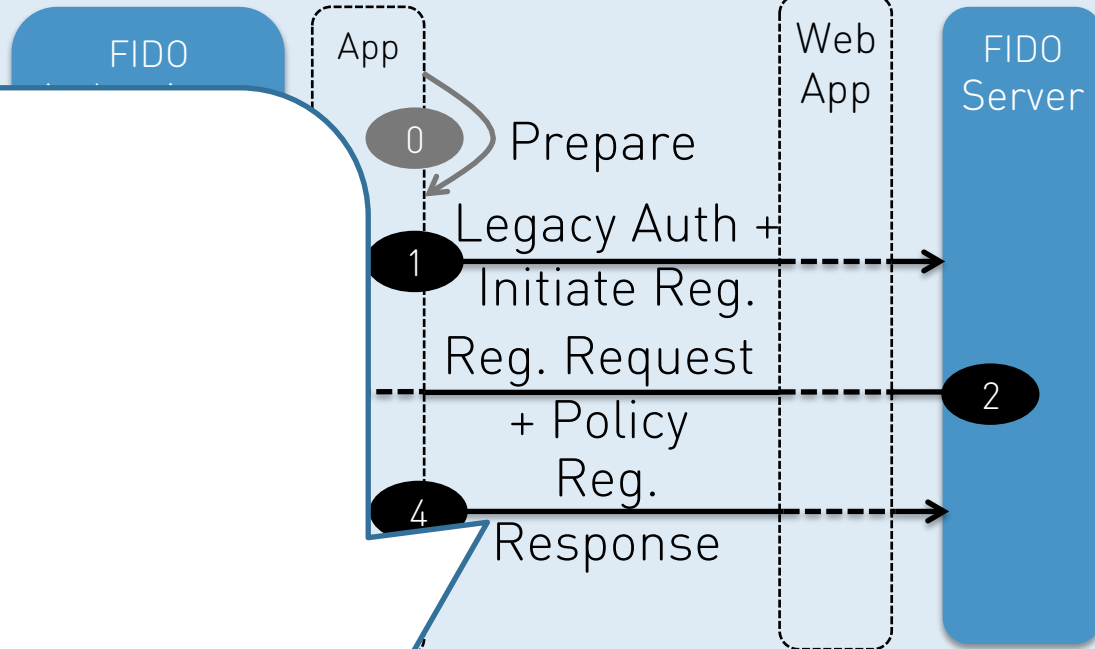
UAF Registration



Key Registration Data:

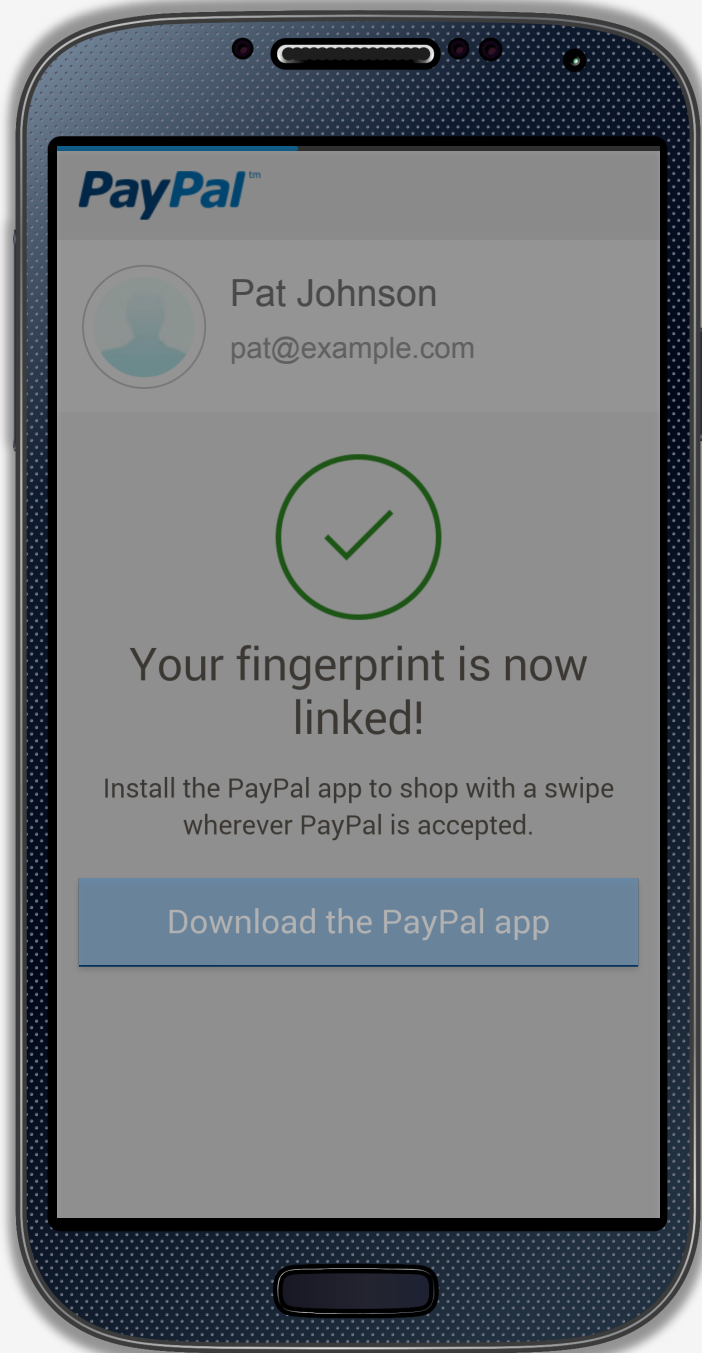
- Hash(FinalChallenge)
- AAID
- Public key
- KeyID
- Registration Counter
- Signature Counter
- Signature (attestation key)

$$\text{FinalChallenge} = \text{Hash}(\text{AppID} \mid \text{FacetID} \mid \text{tlsData} \mid \text{challenge})$$

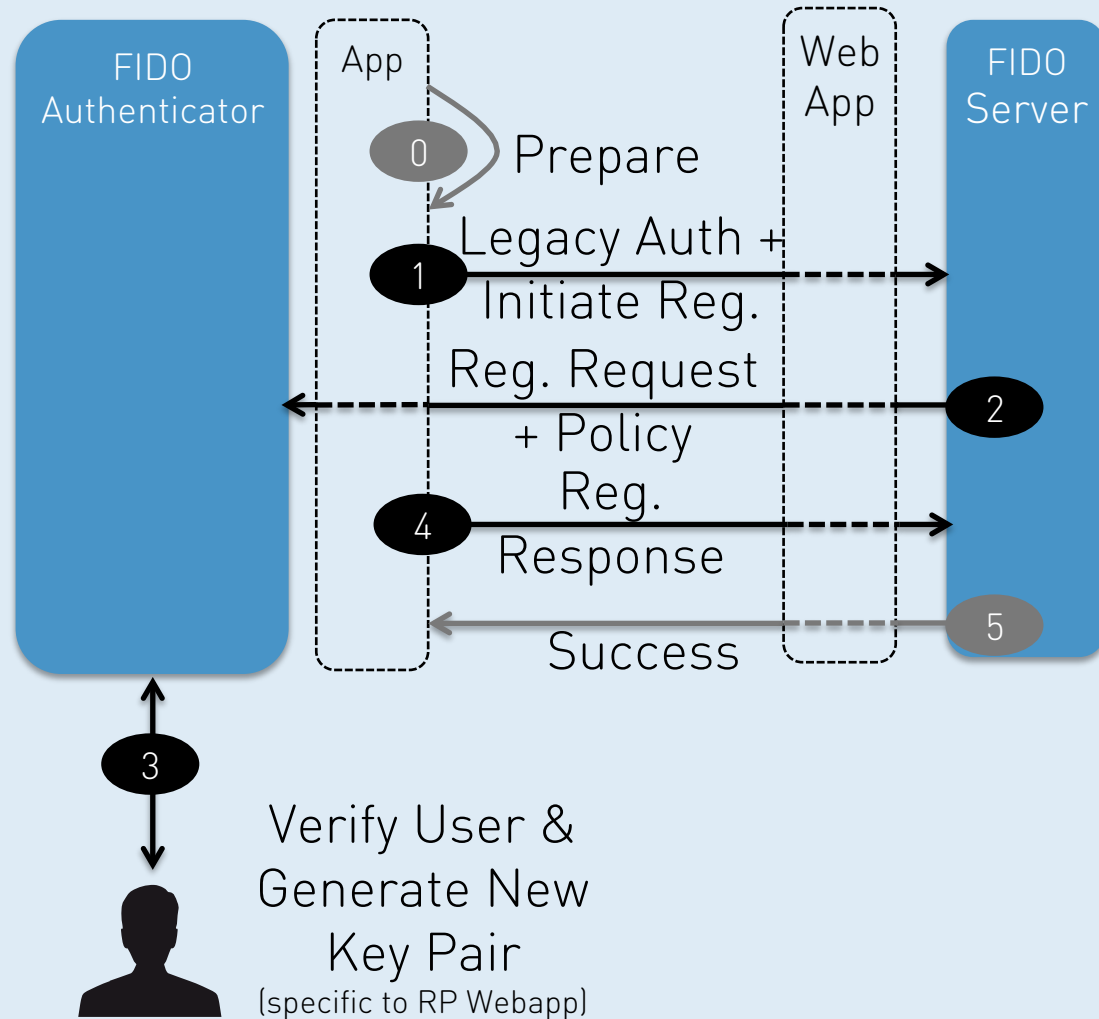


Verify User &
Generate New
Key Pair
(specific to RP Webapp)

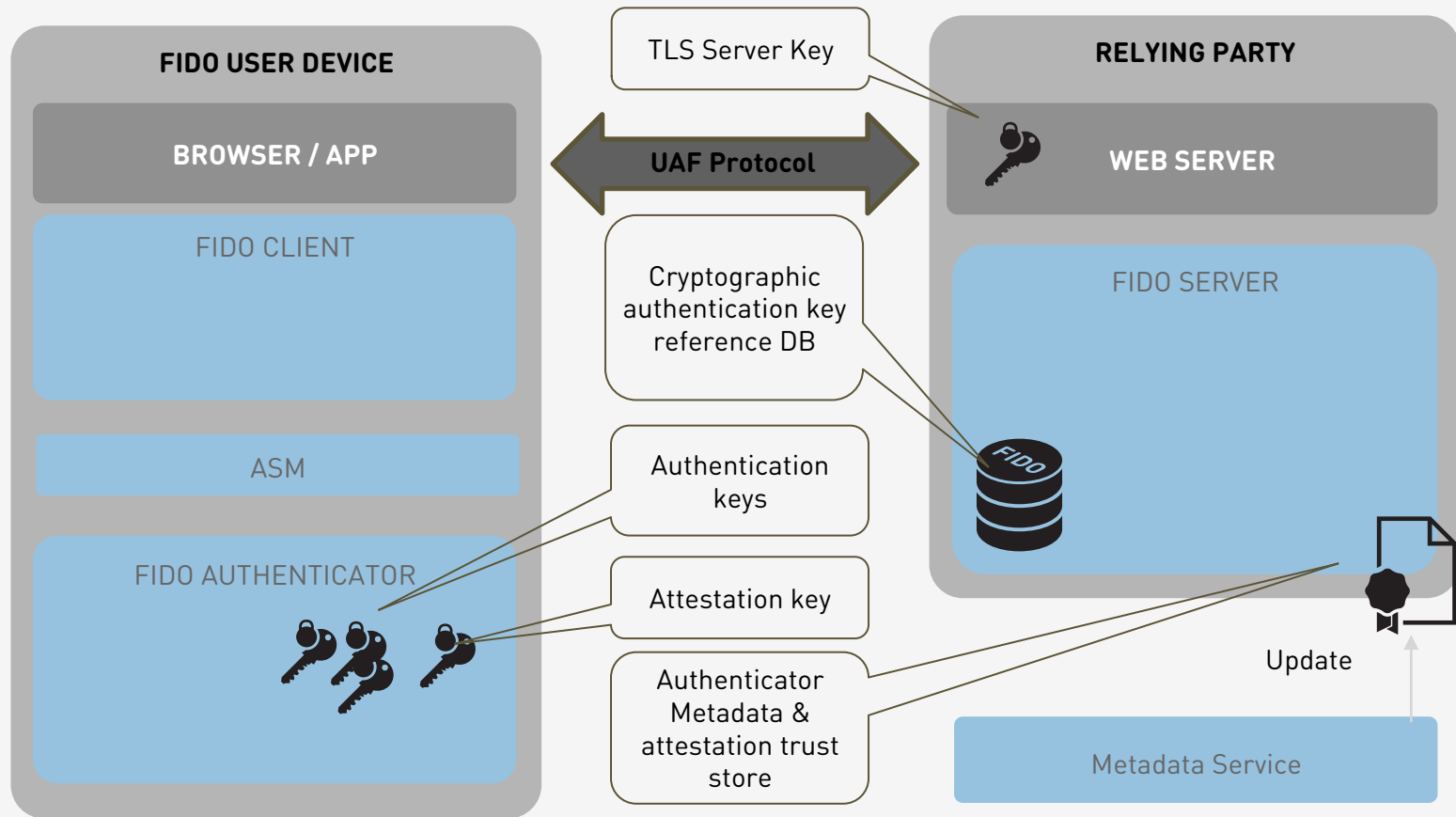




UAF Registration



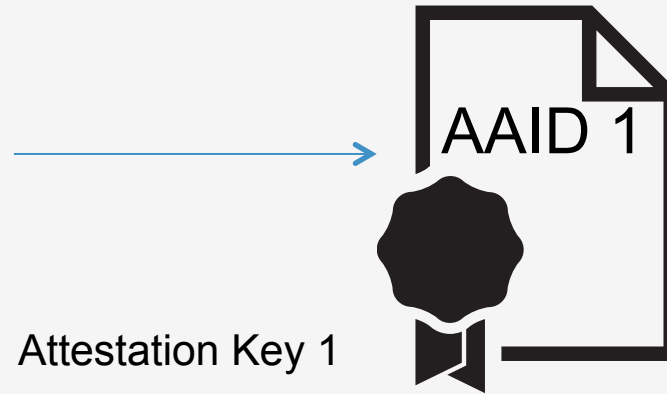
FIDO Building Blocks



AAID & Attestation

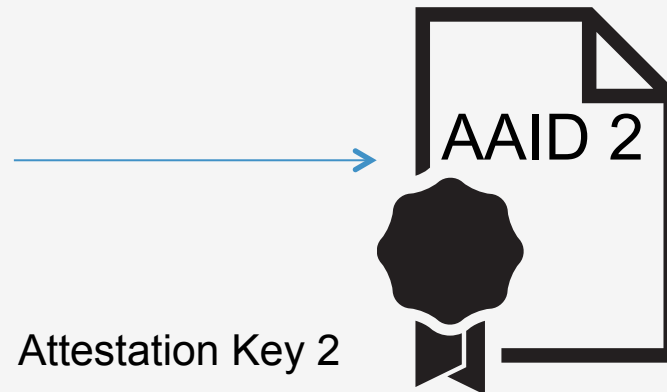
FIDO Authenticator

Using HW based crypto
Based on FP Sensor X



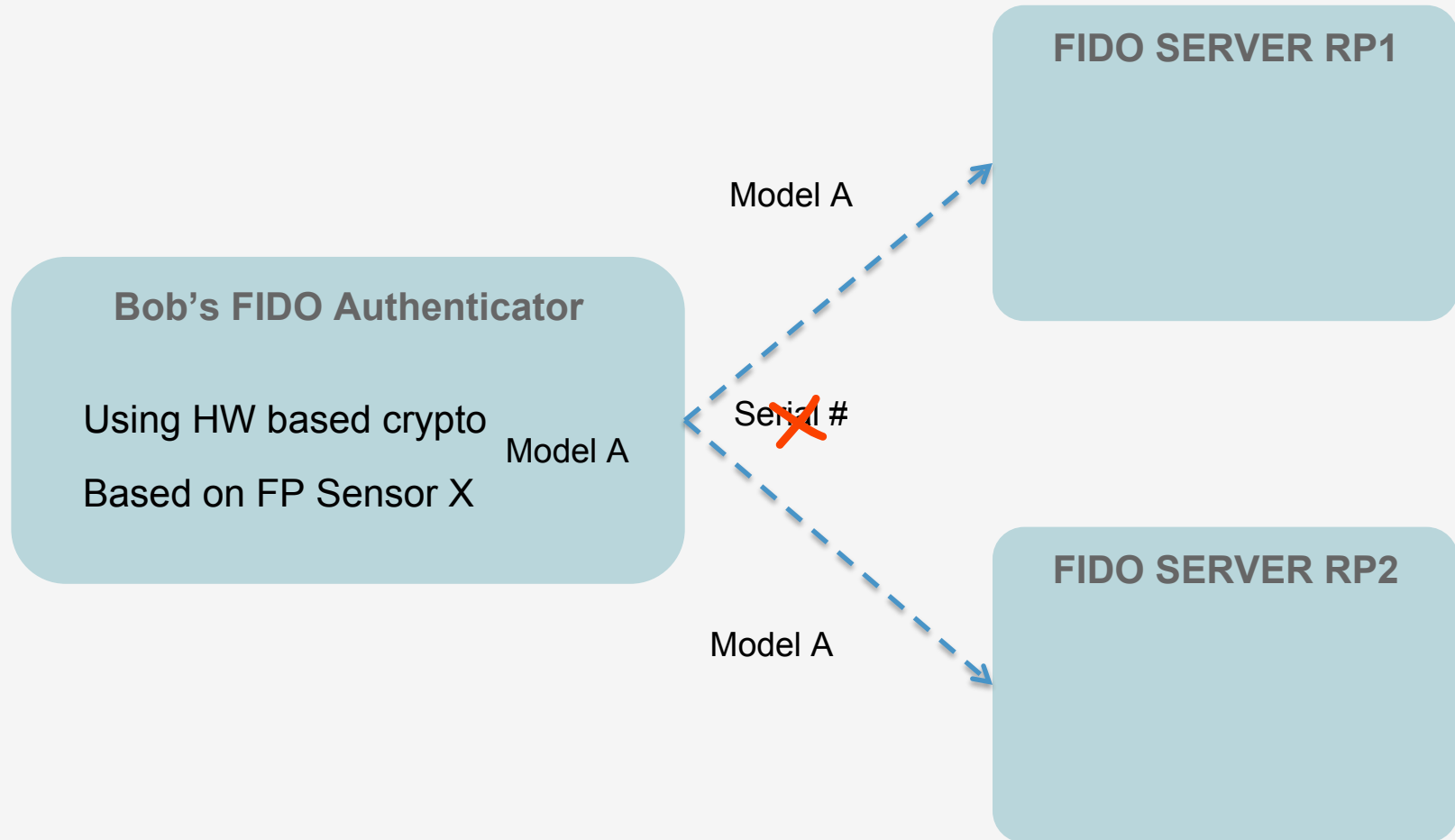
FIDO Authenticator

Pure SW based implementation
Based on Face Recognition alg. Y

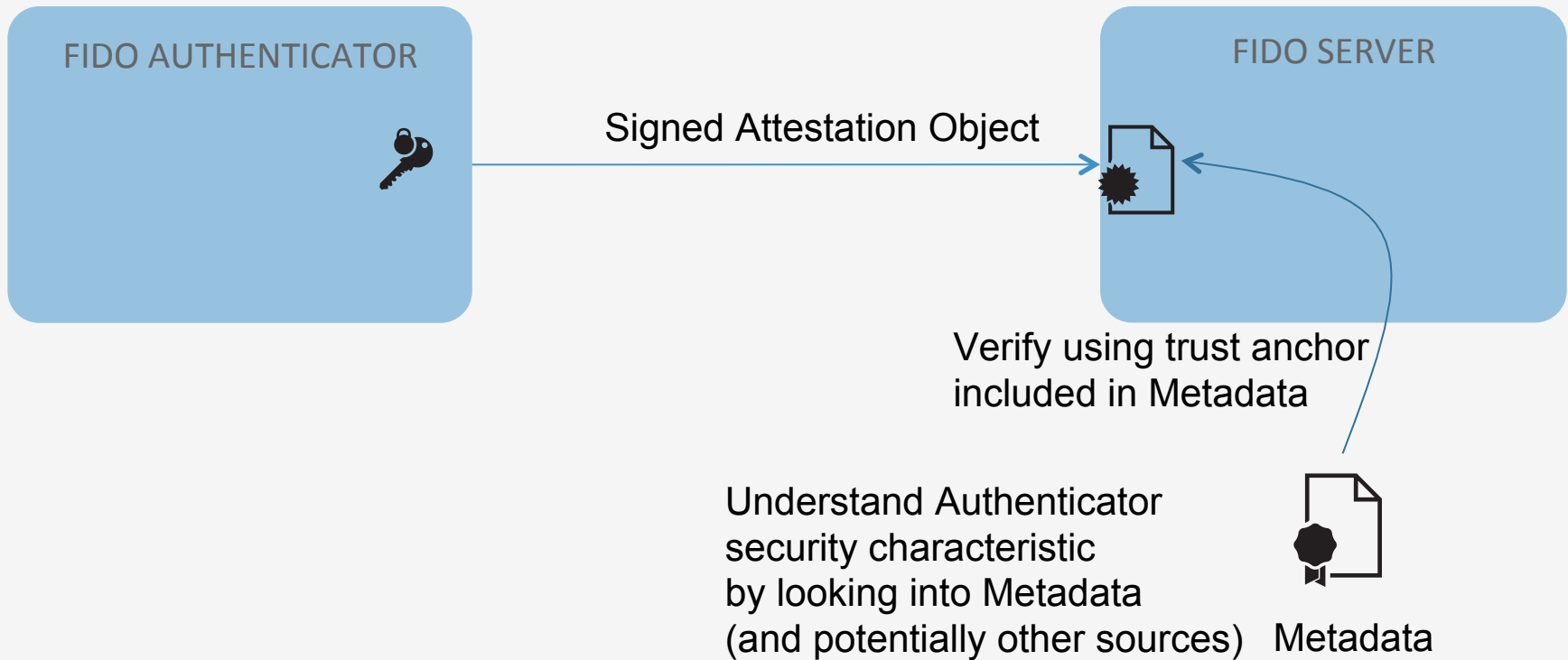


AAID: Authenticator Attestation ID (=model name)

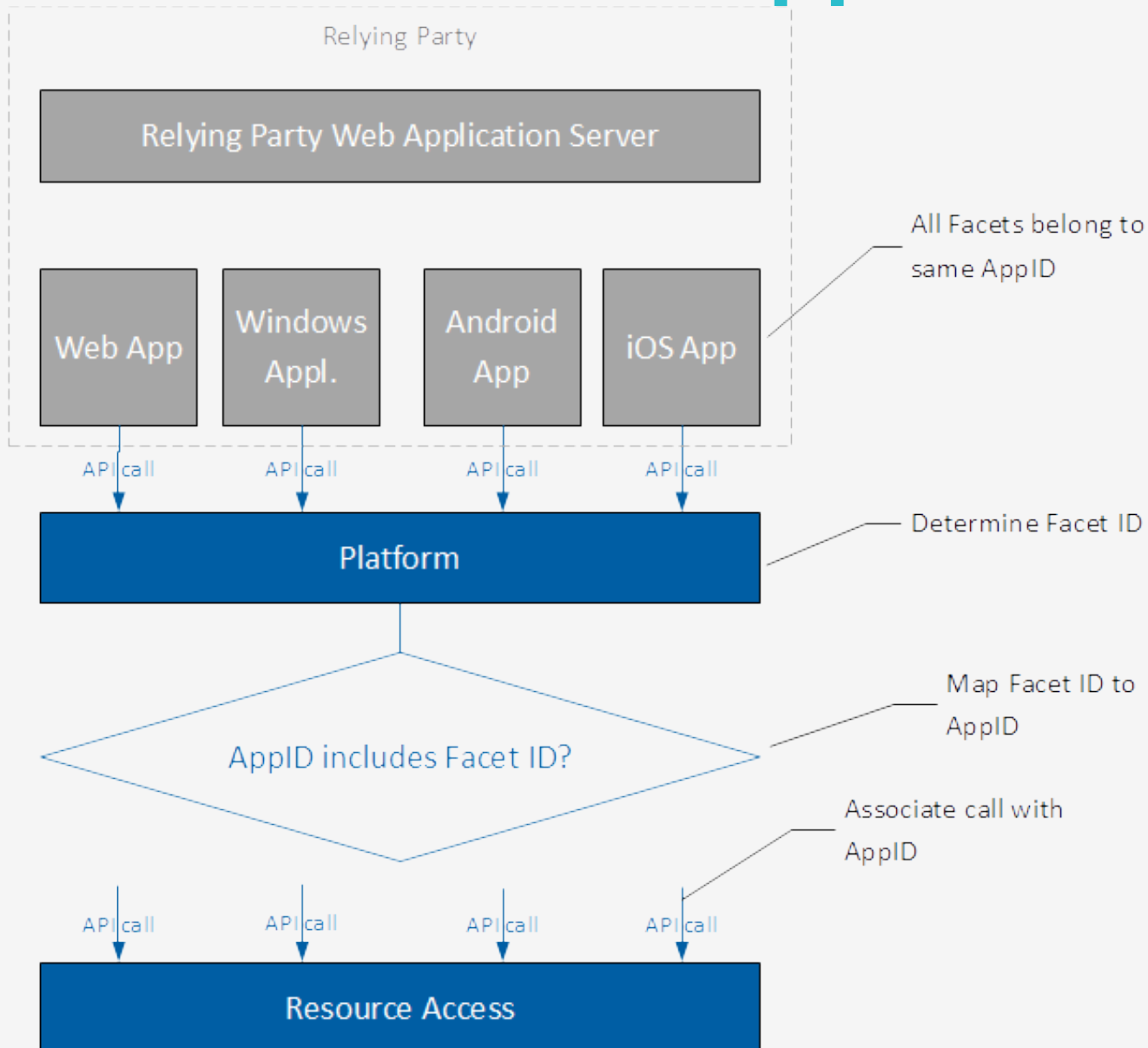
Privacy & Attestation

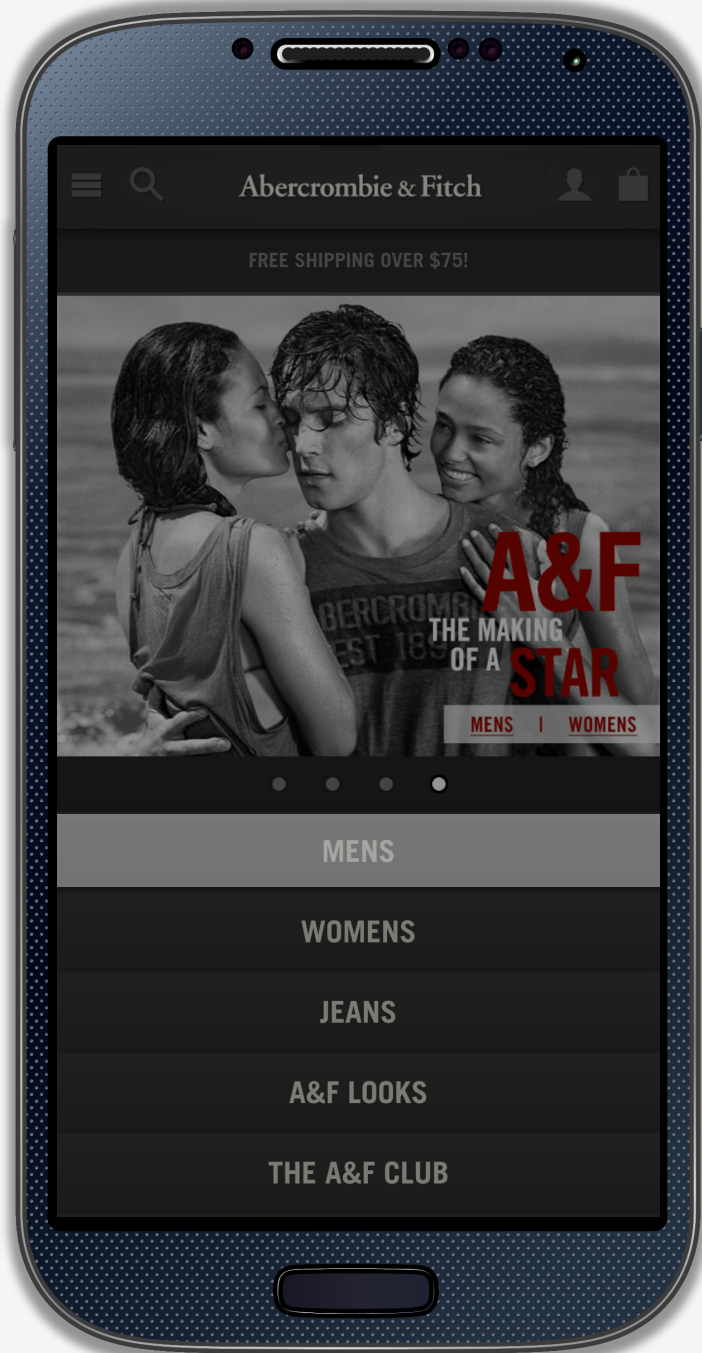


Attestation & Metadata

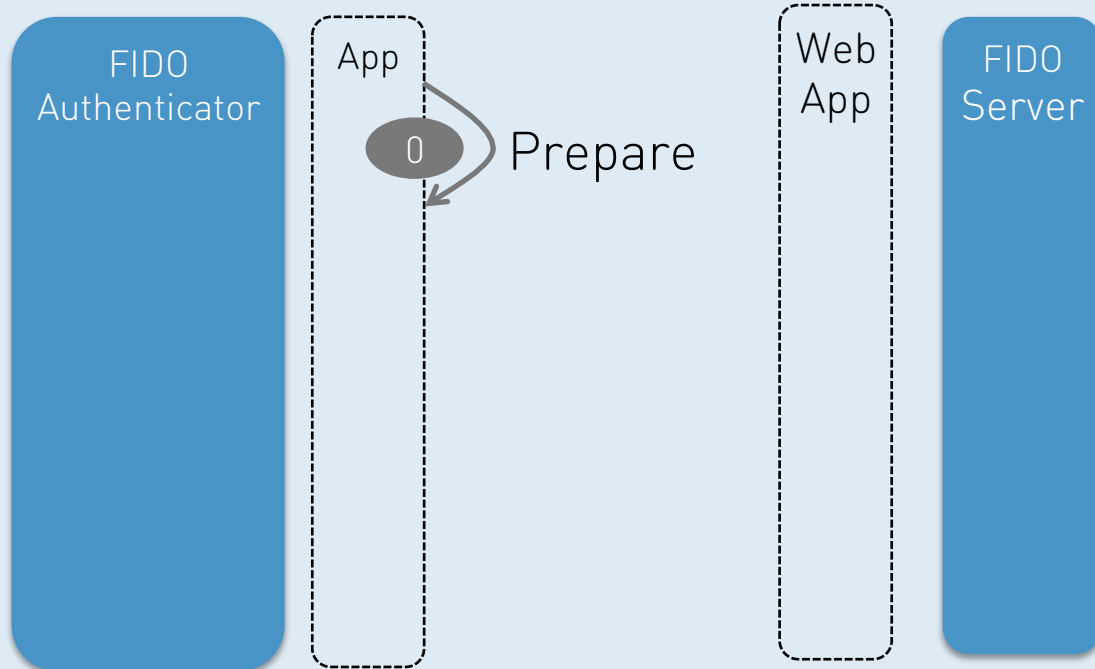


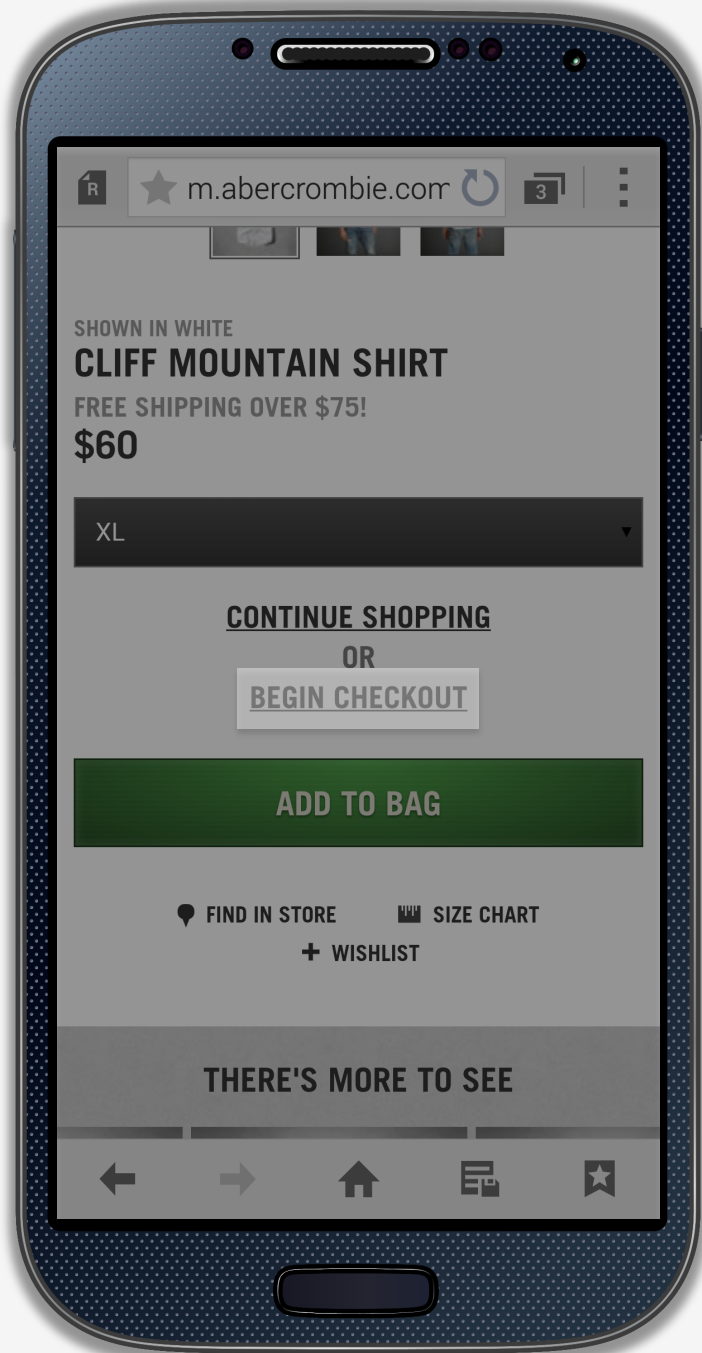
Facet ID / AppID



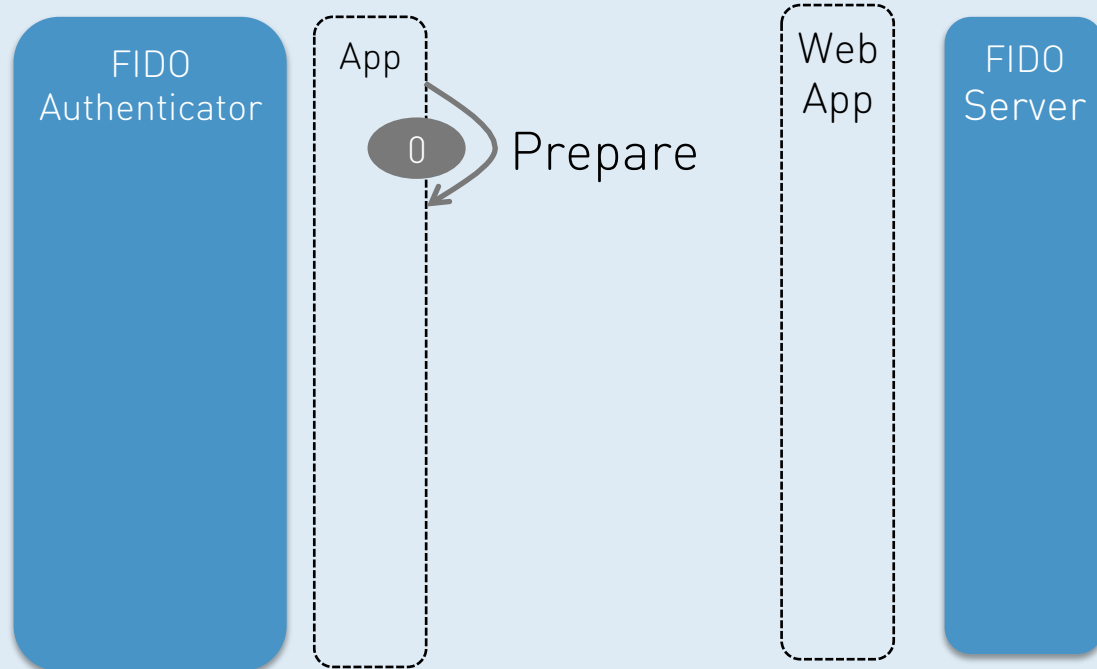


UAF Authentication

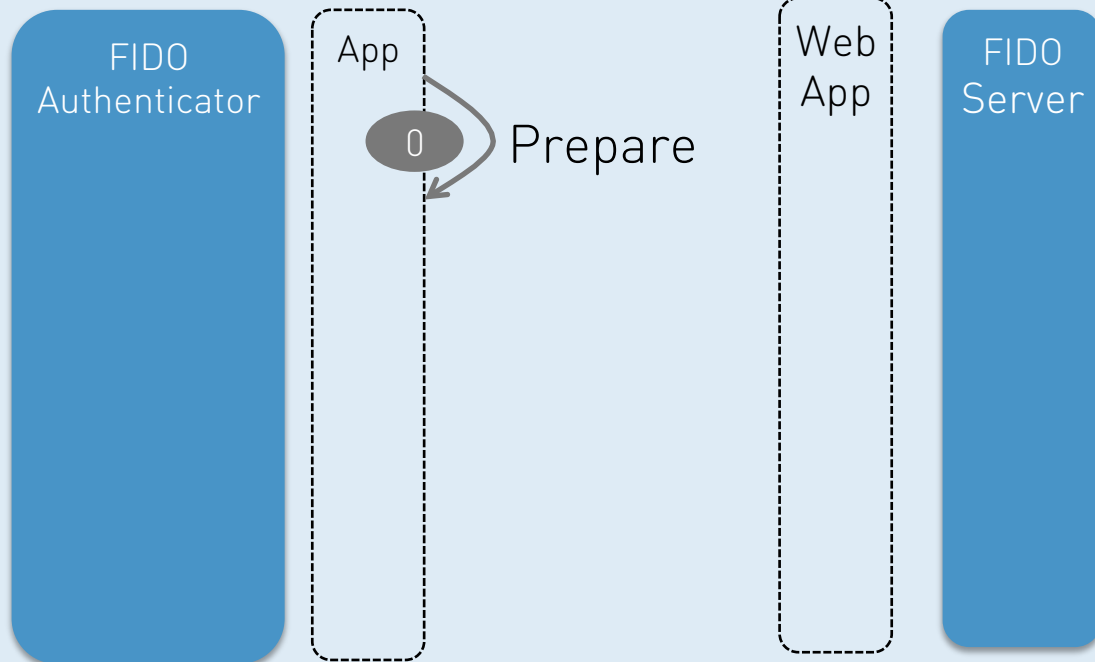
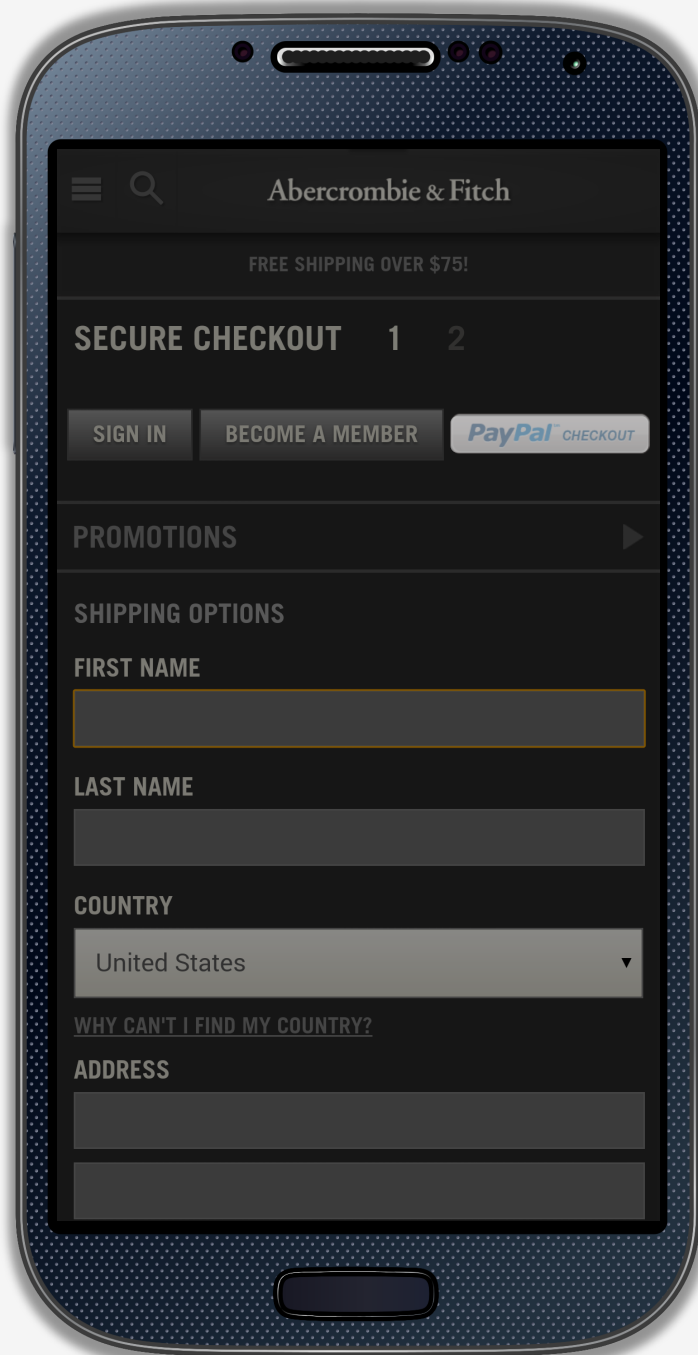




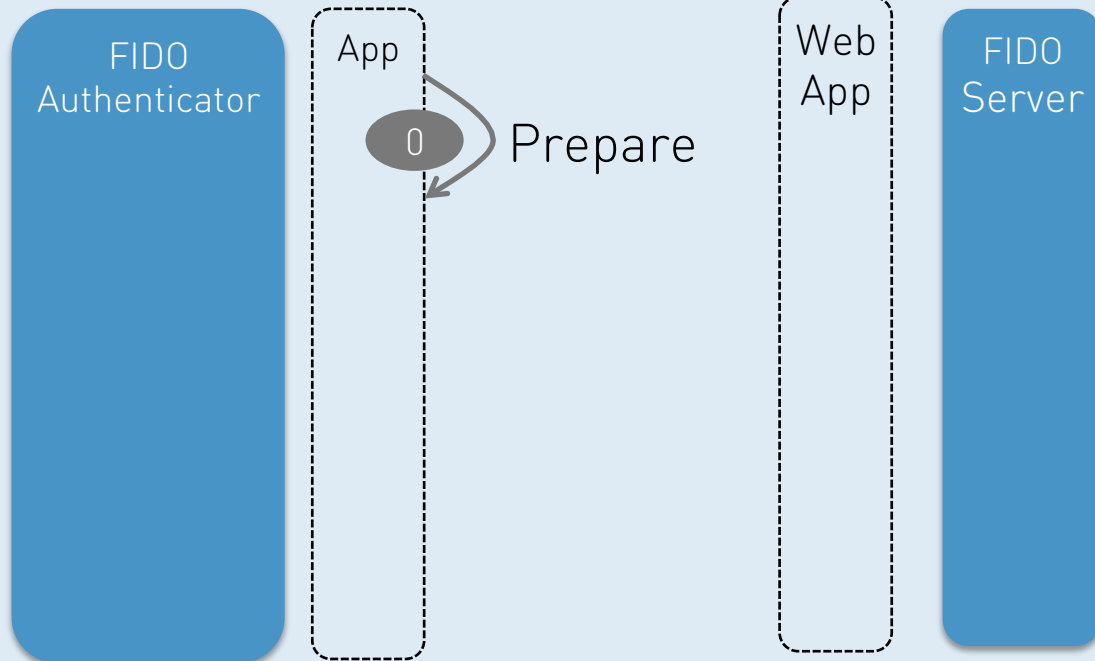
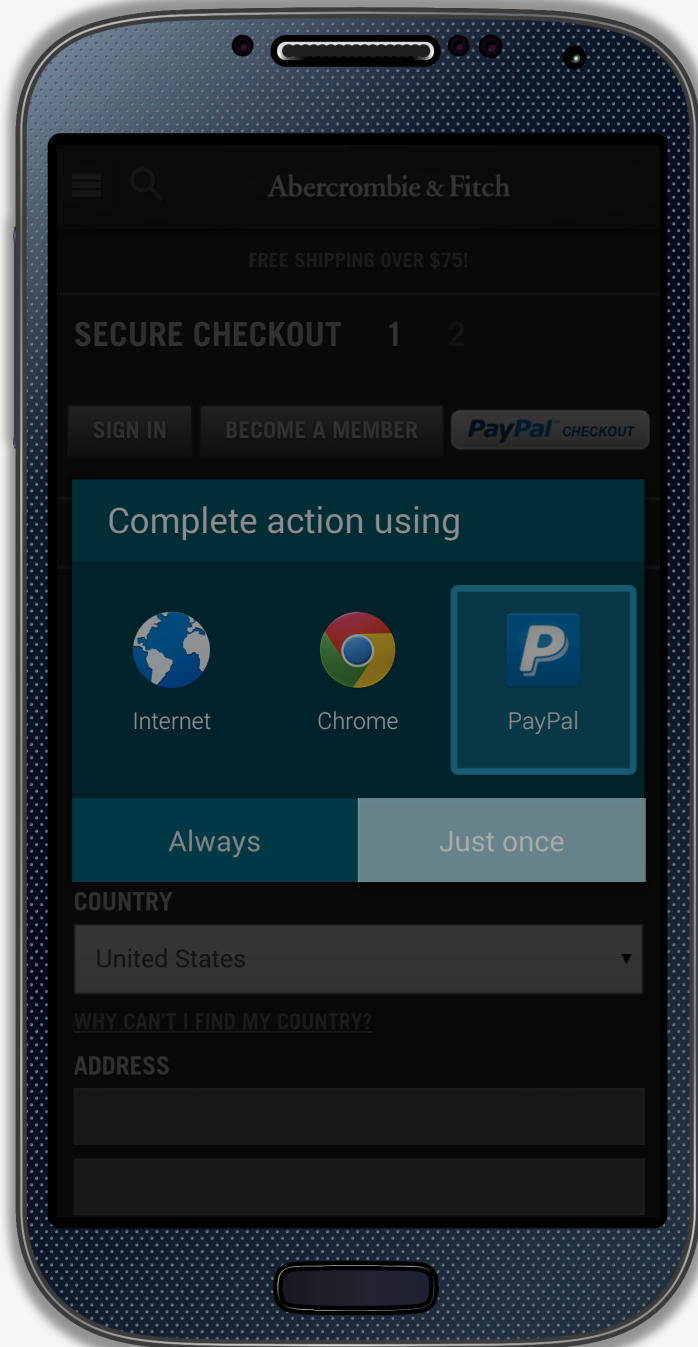
UAF Authentication



UAF Authentication

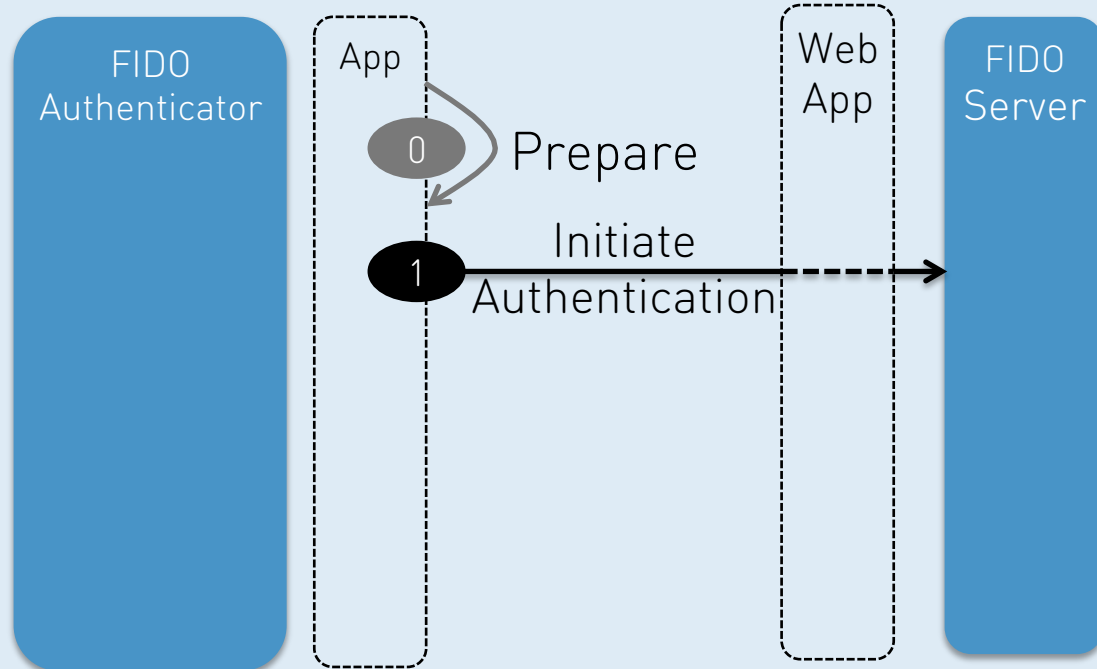


UAF Authentication



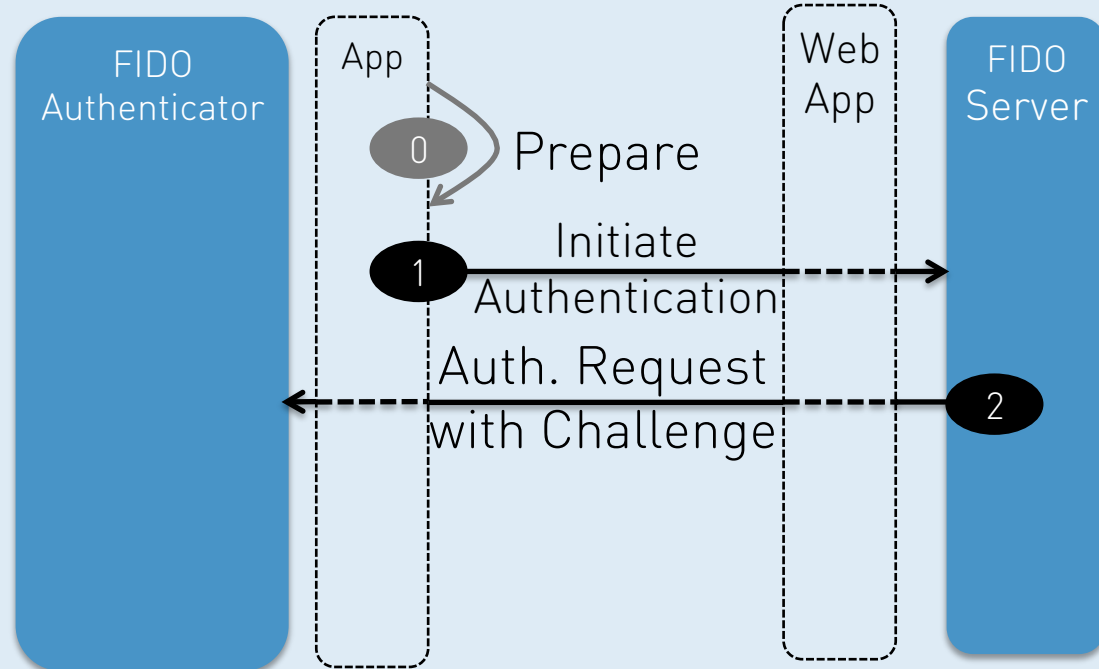


UAF Authentication





UAF Authentication



UAF Authentication



PayPal™

FIDO Authenticator

App

Web App

FIDO Server

0

Prepare

1

Initiate

Authentication
Auth. Request

2

with Challenge

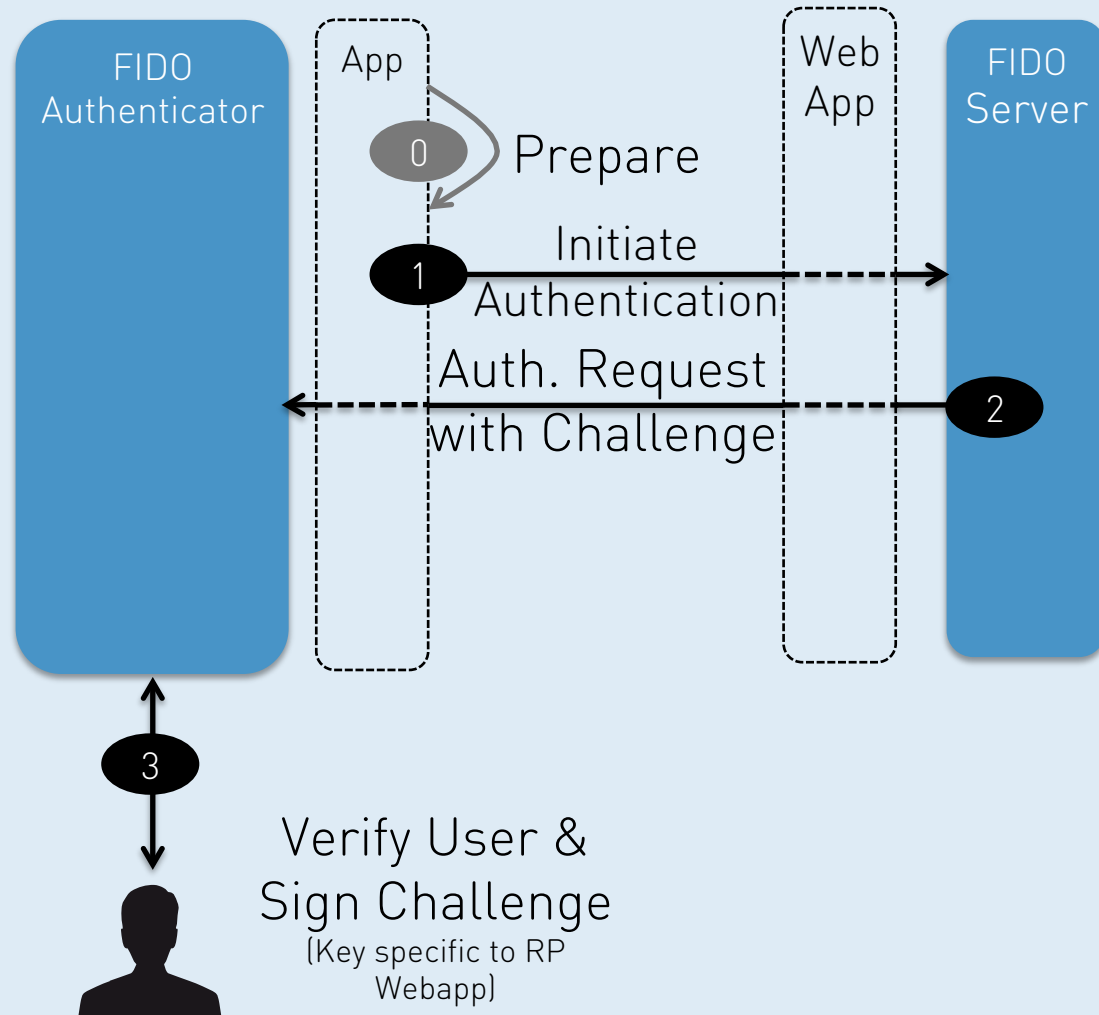
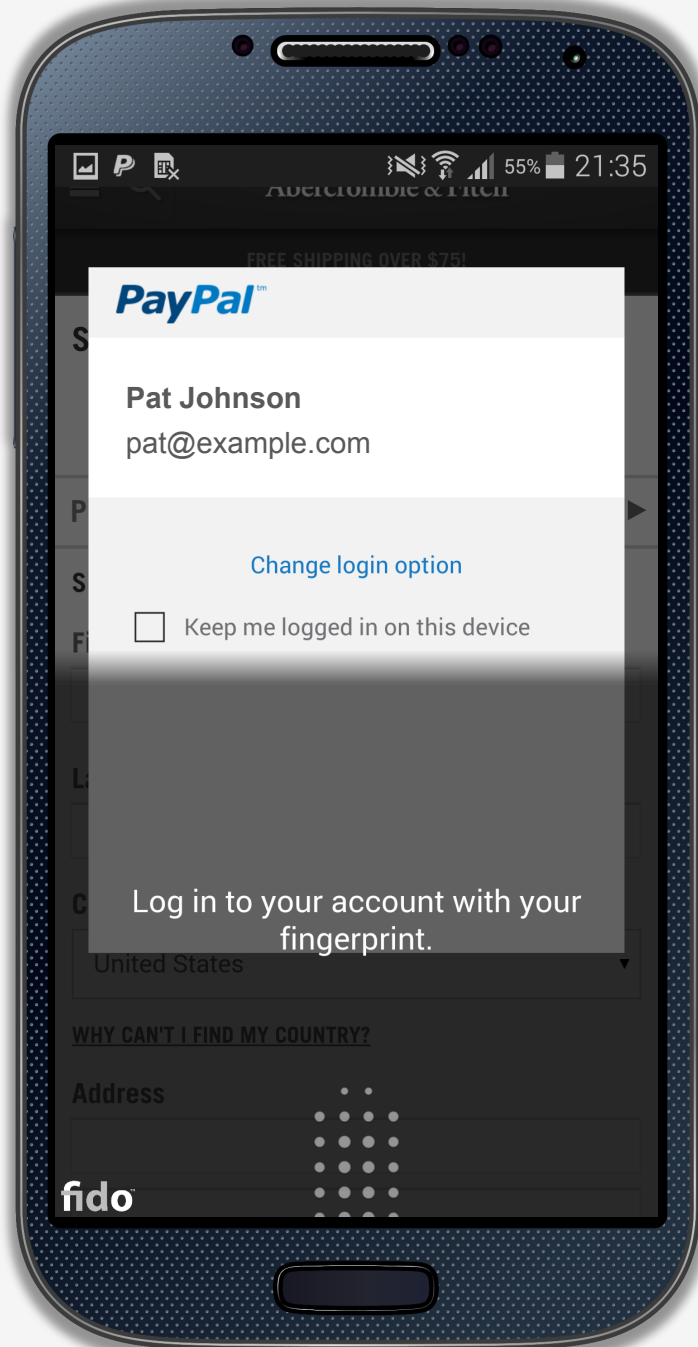
```
[{
```

```
"header": {"op": "Auth", "upv": "1.0", "appID": "https://mycorp.com/fido"},  
"challenge": "triz786ighwer8764g6574234515reg45z",  
"policy": {  
  "accepted": [[{  
    "authenticationFactor": 00000000000001ff,  
    "keyProtection": 000000000000000e,  
    "attachment": 00000000000000ff,  
    "secureDisplay": 000000000000001e,  
    "supportedSchemes": "UAFV1TLV"}]],  
  "disallowed": {"aaid": "1234#5678"}  
}
```

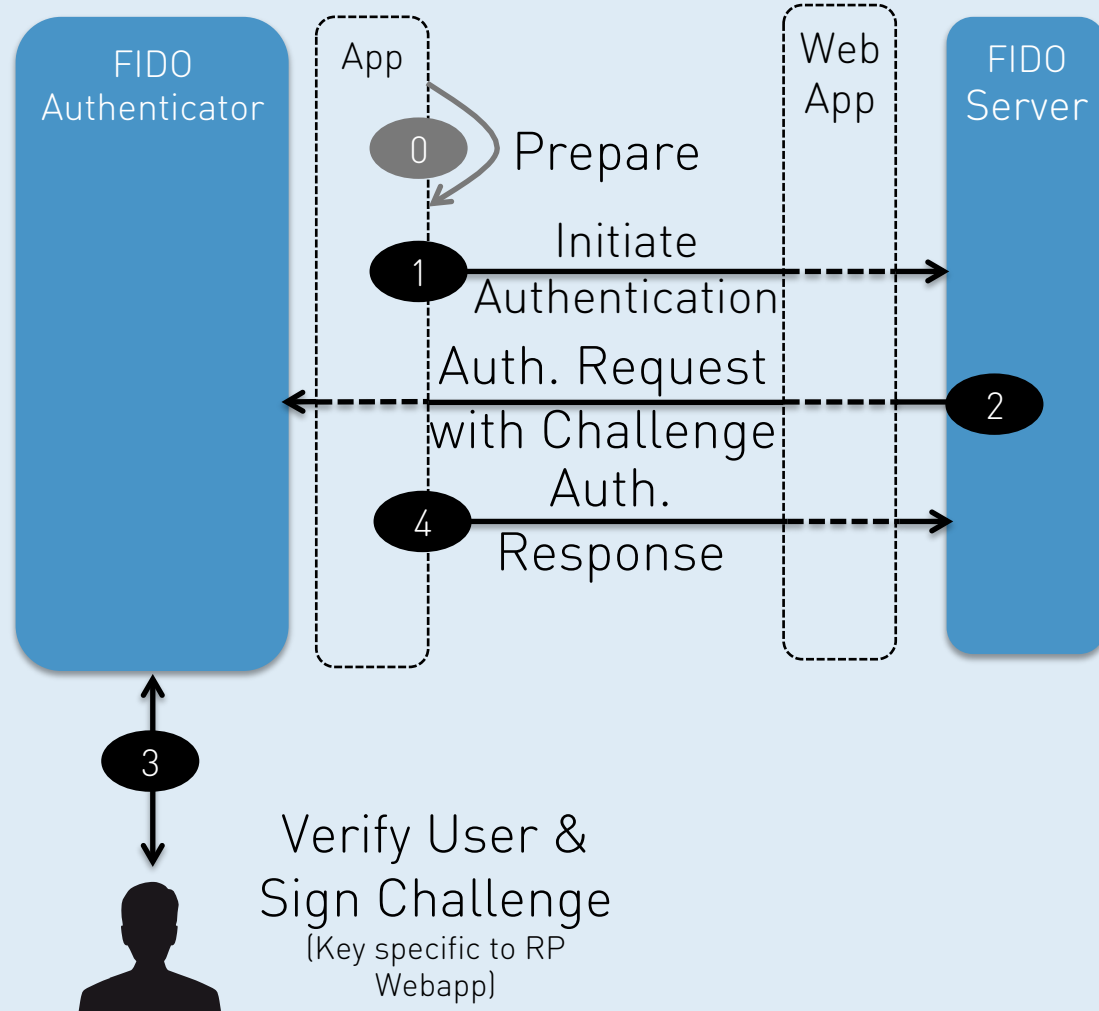
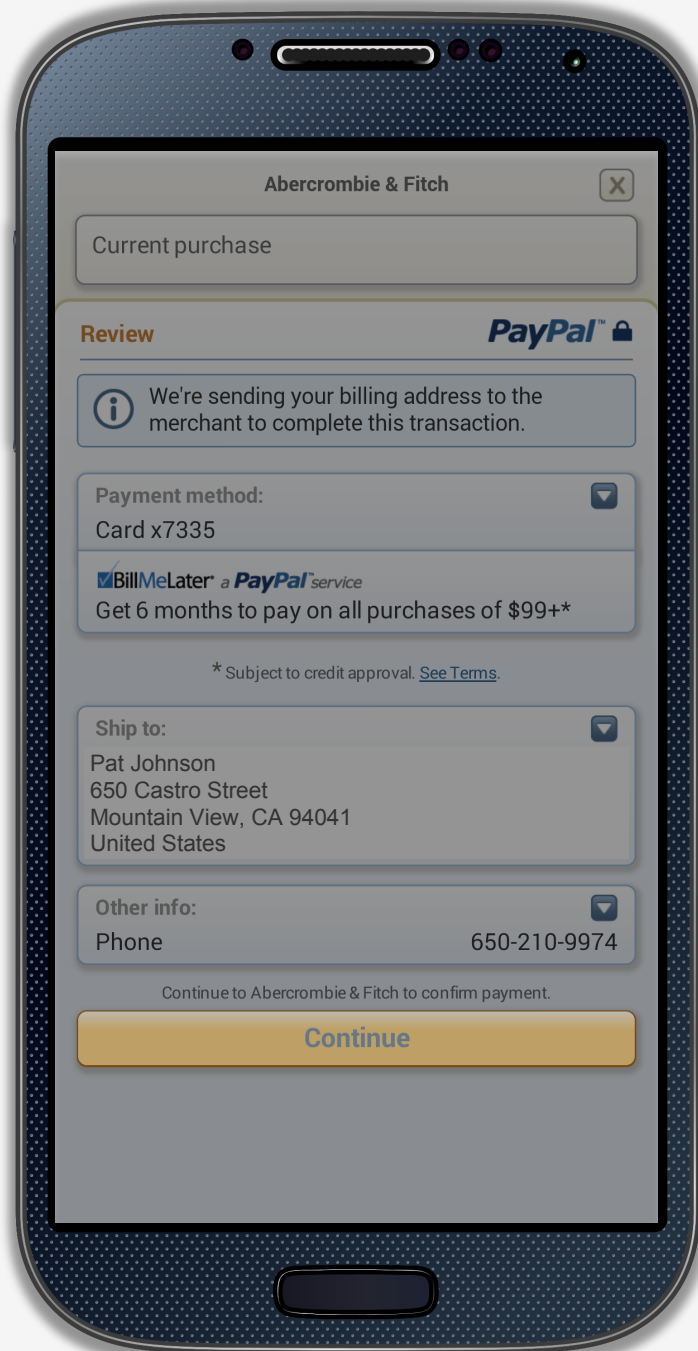
```
}
```

```
]}
```

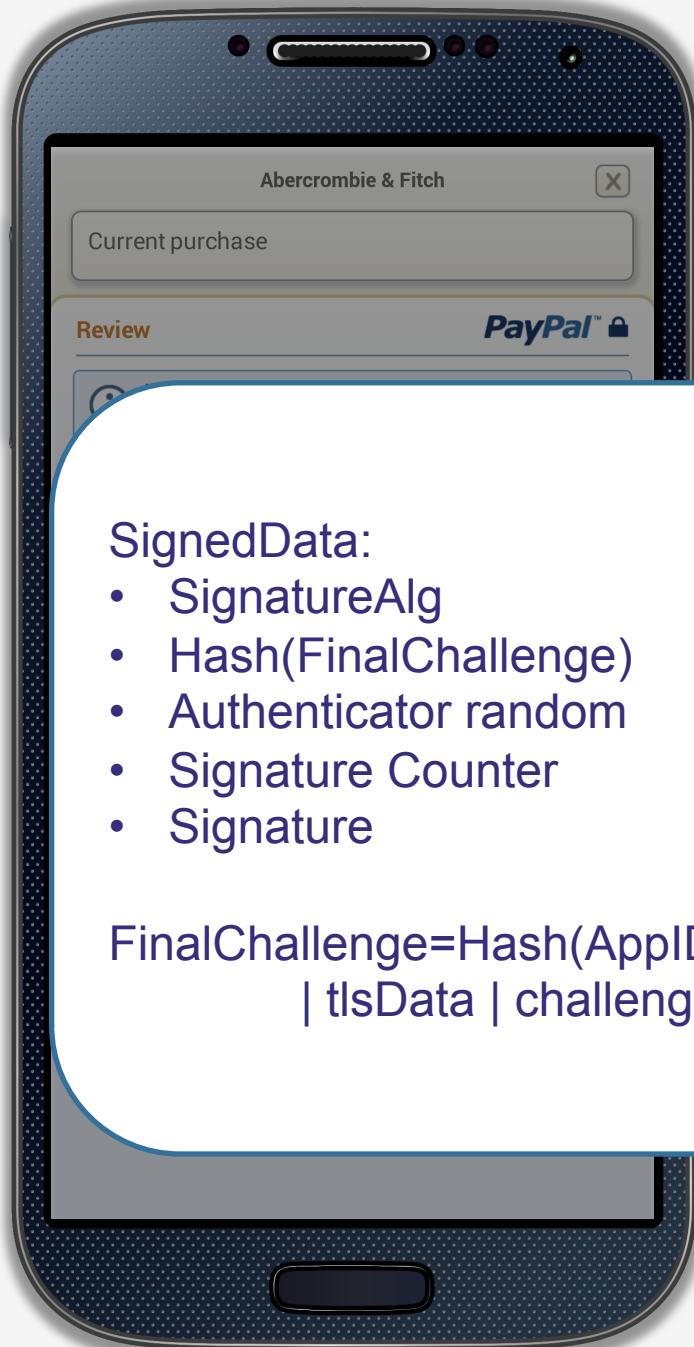
UAF Authentication



UAF Authentication



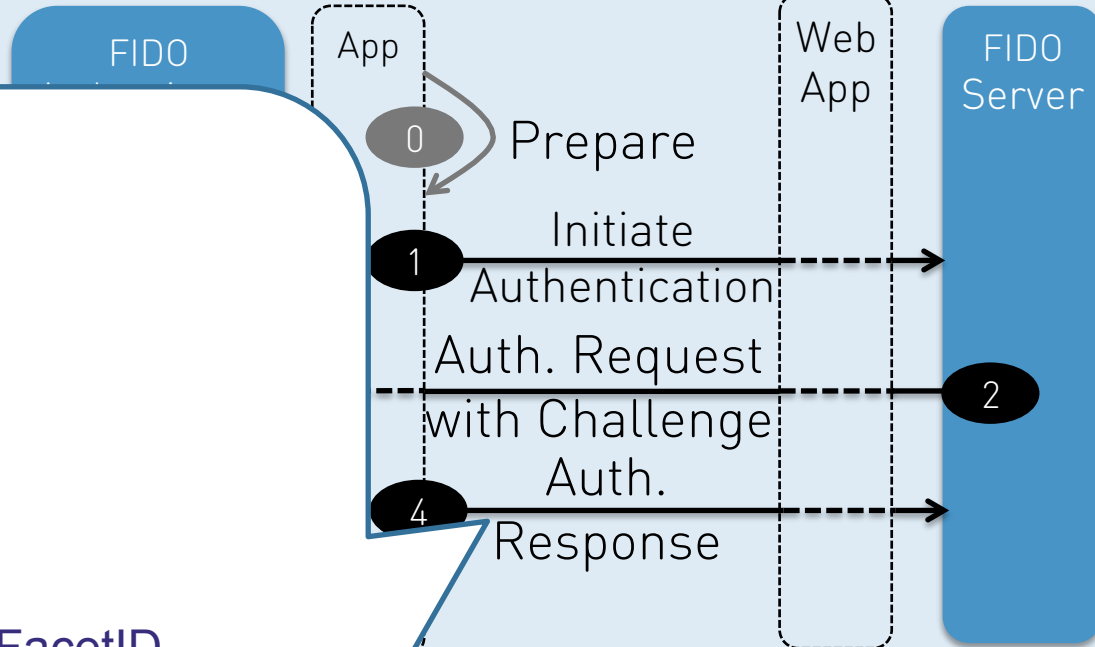
UAF Authentication



SignedData:

- SignatureAlg
- Hash(FinalChallenge)
- Authenticator random
- Signature Counter
- Signature

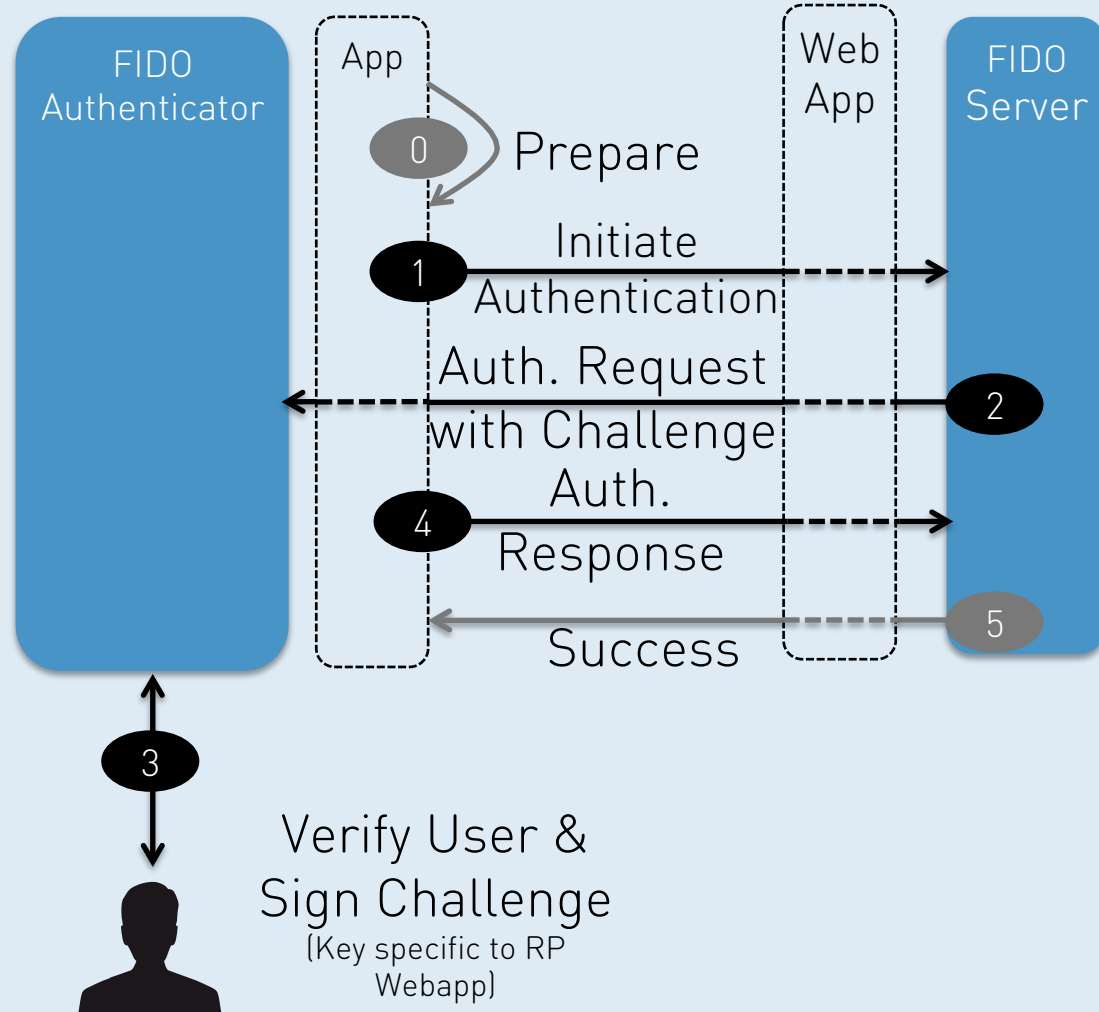
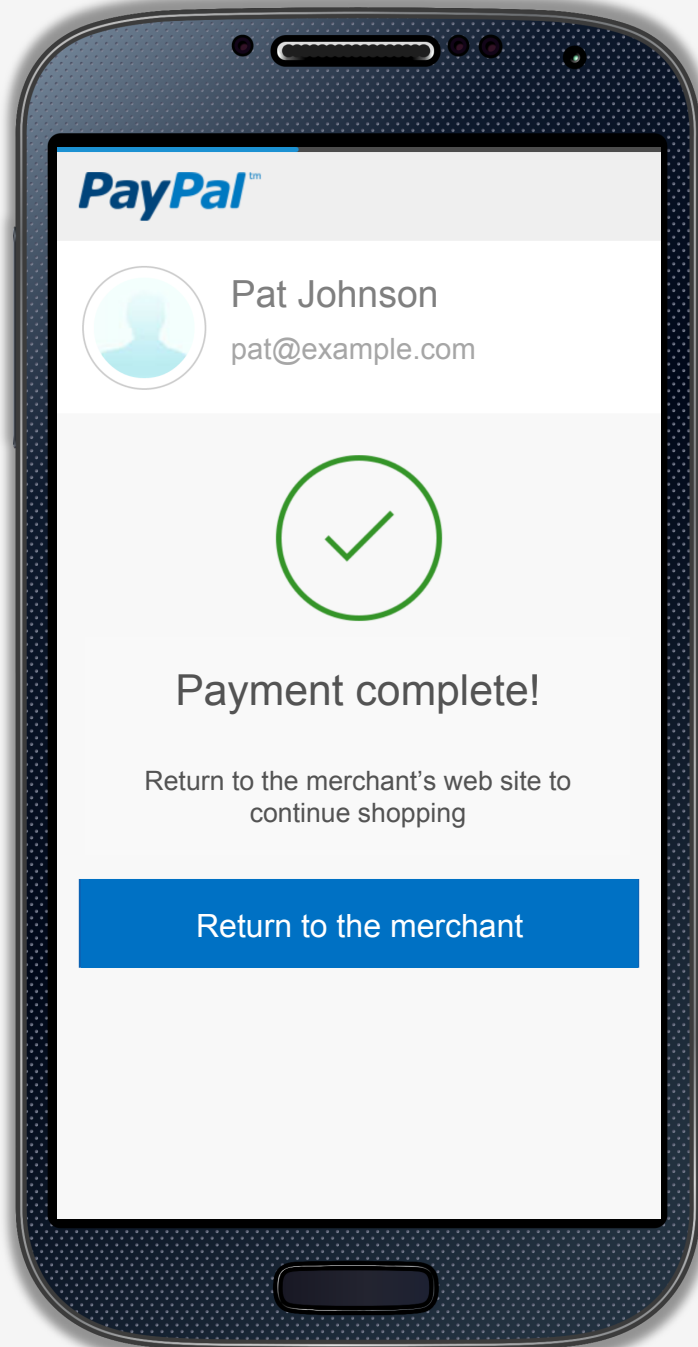
FinalChallenge=Hash(AppID | FacetID
| tlsData | challenge)



Verify User &
Sign Challenge
(Key specific to RP
Webapp)



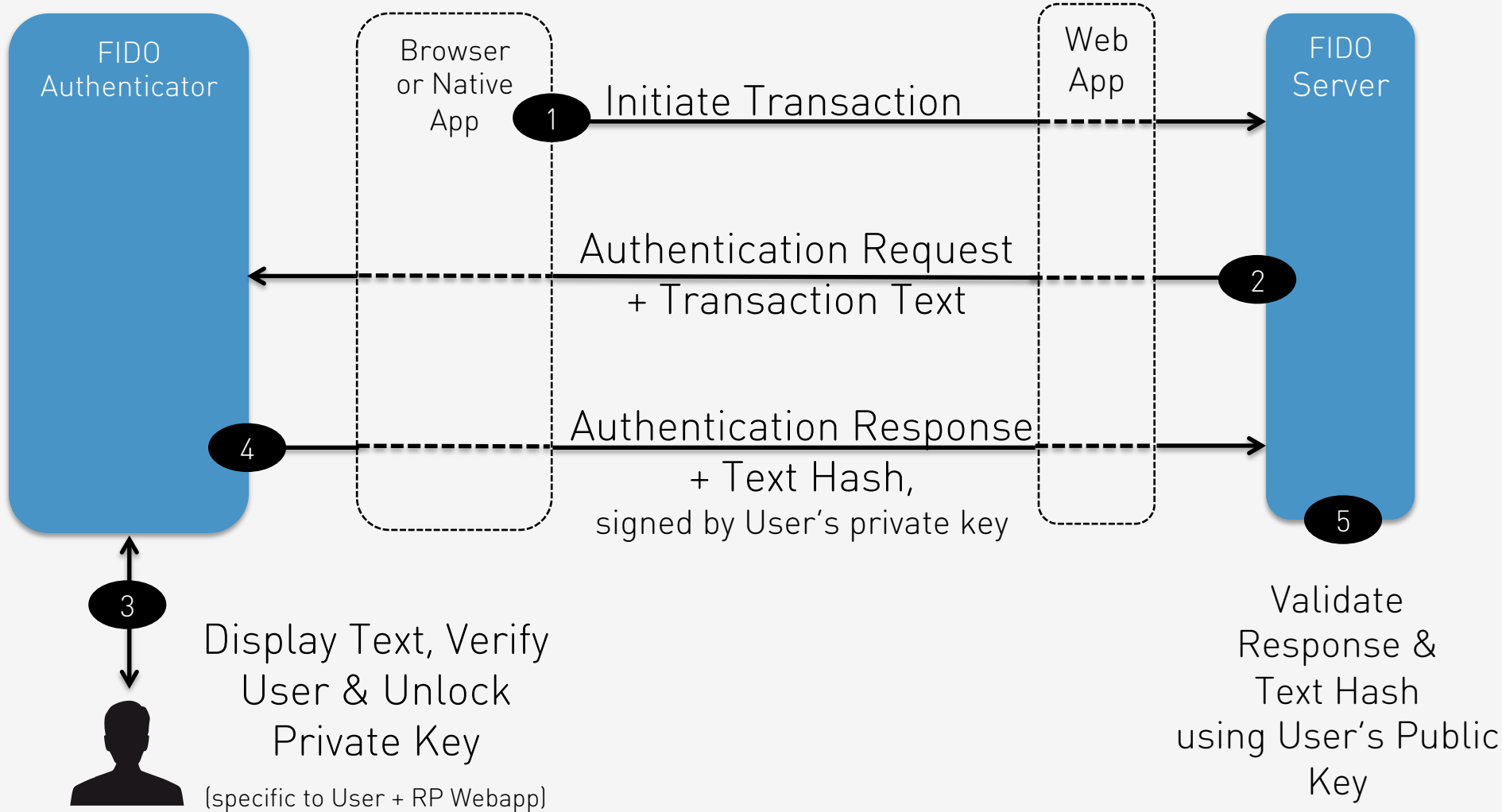
UAF Authentication



Transaction Confirmation

Device

Relying Party



Transaction Confirmation

Device

Relying Party

FIDO
Authenticator

Browser
or Native

Initiate Transaction

Web
App

FIDO
Server

SignedData:

- SignatureAlg
- Hash(FinalChallenge)
- Authenticator random
- Signature Counter
- Hash(Transaction Text)
- Signature

FinalChallenge=Hash(AppID | FacetID
| tlsData | challenge)

Private Key

(specific to User + RP Webapp)

request
Text

2

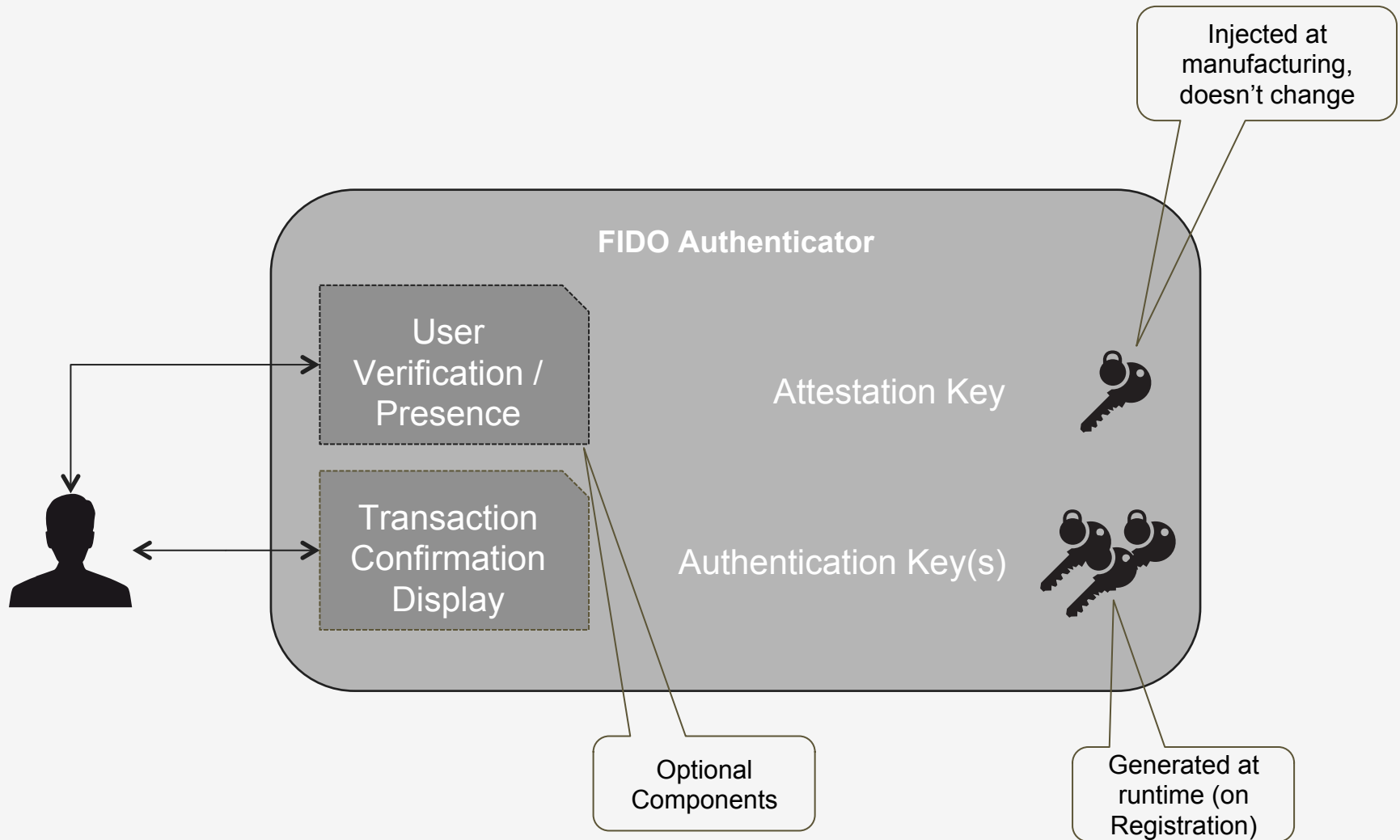
response

key

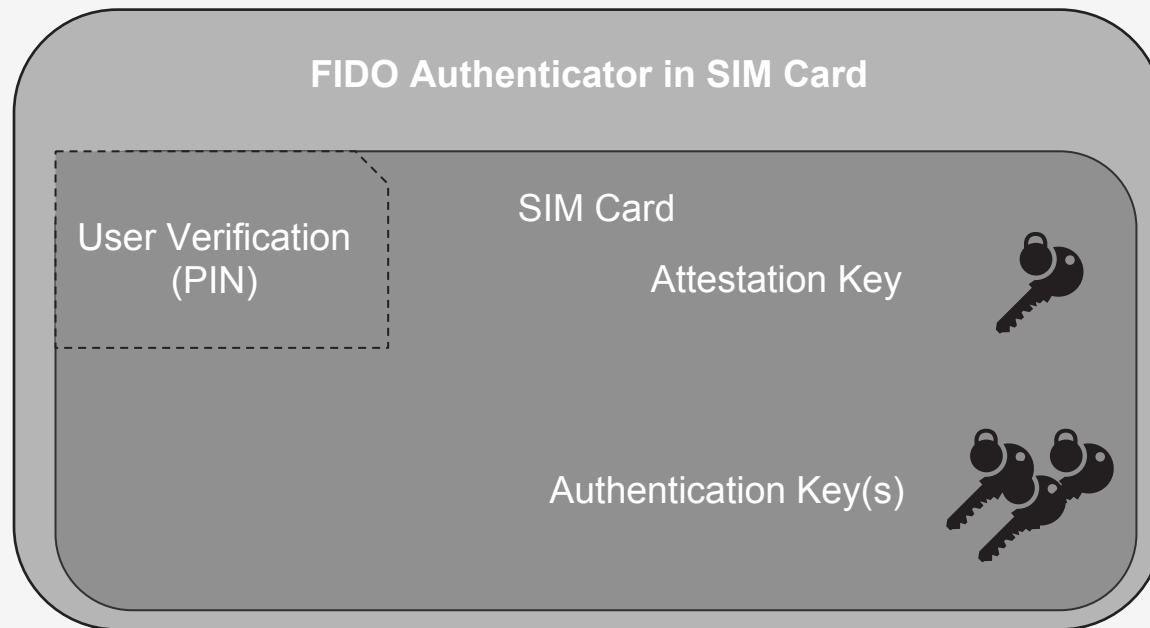
5

Validate
Response &
Text Hash
using User's Public
Key

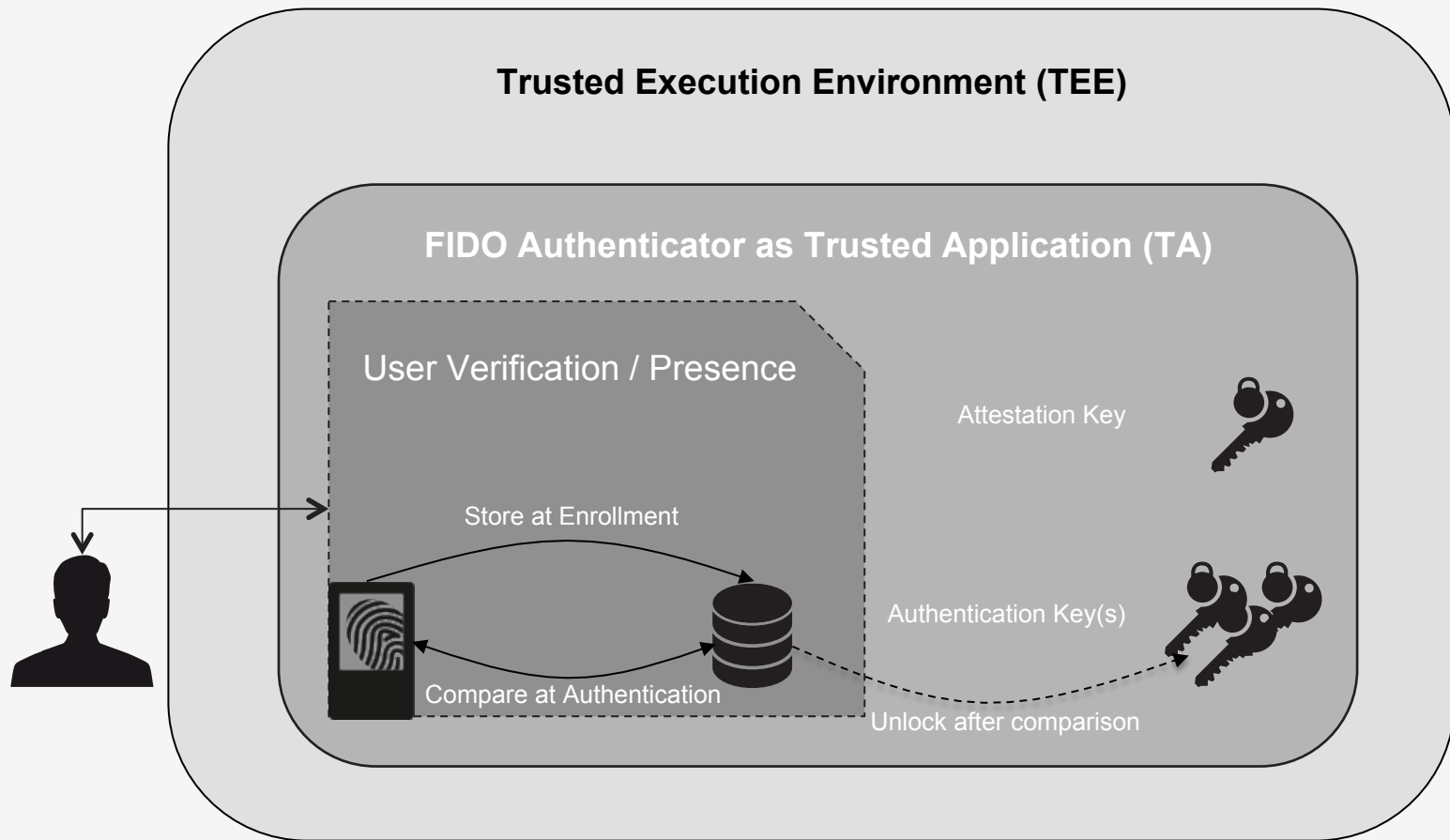
The FIDO Authenticator Concept



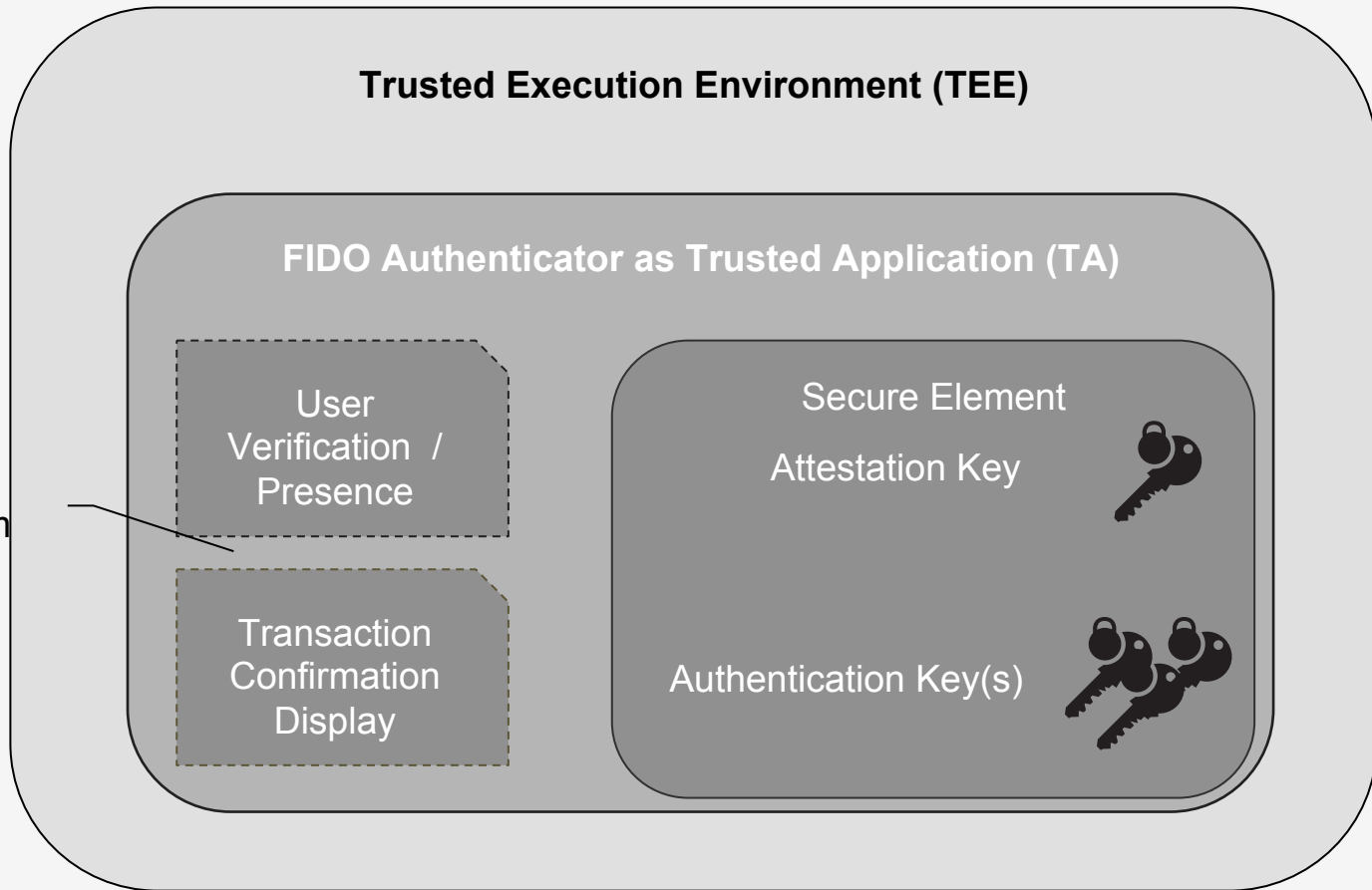
Using Secure Hardware



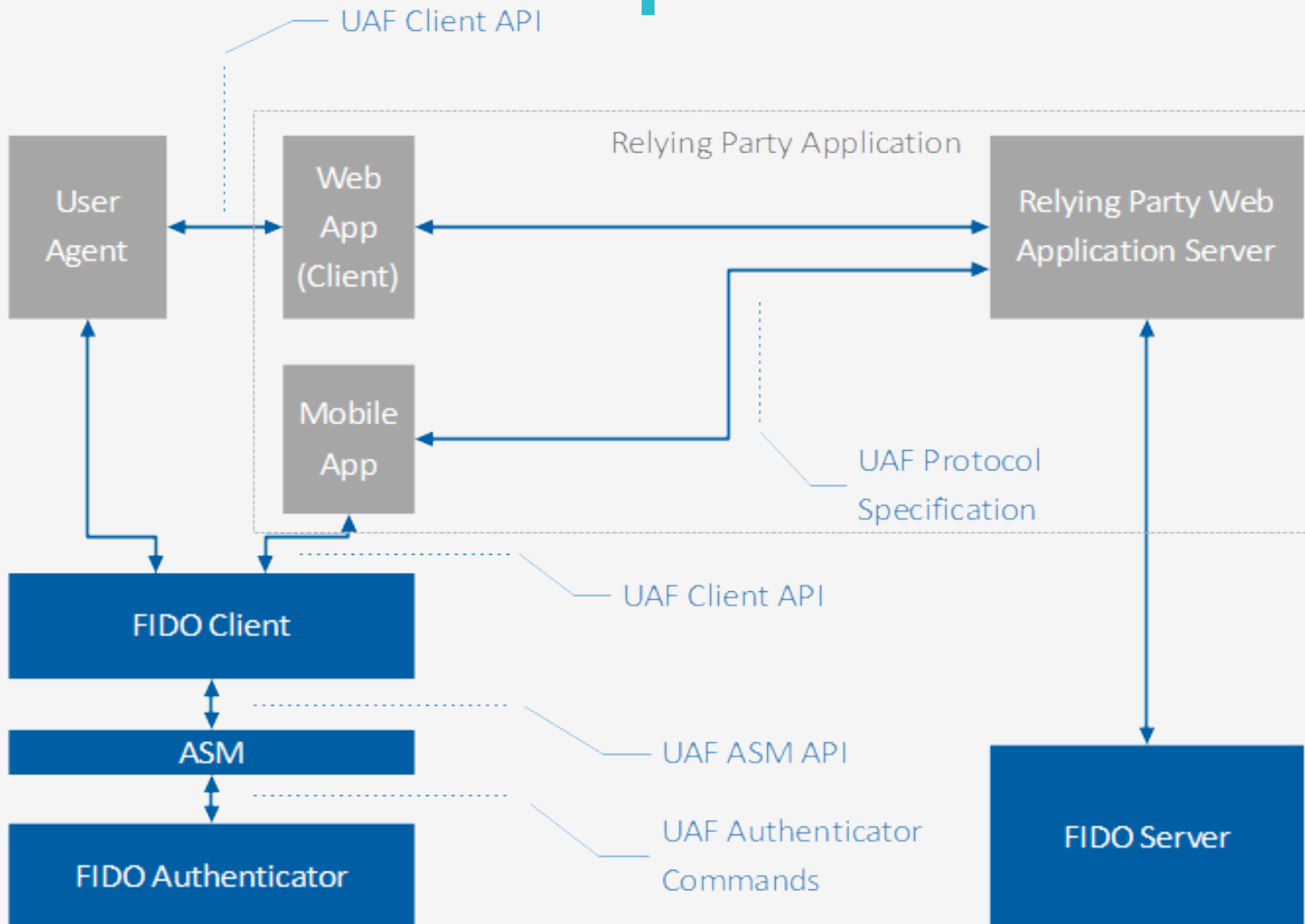
Client Side Biometrics



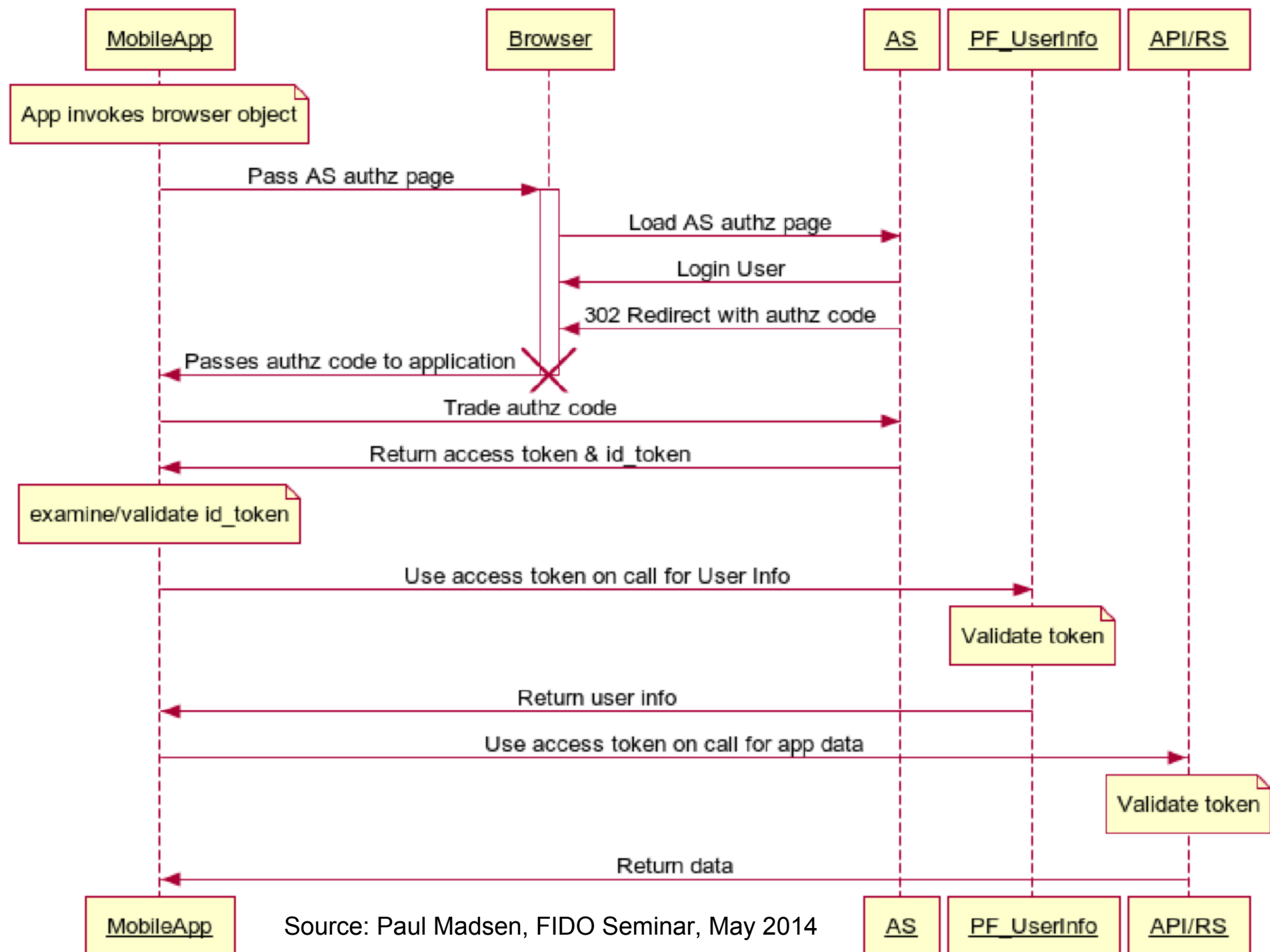
Combining TEE and SE

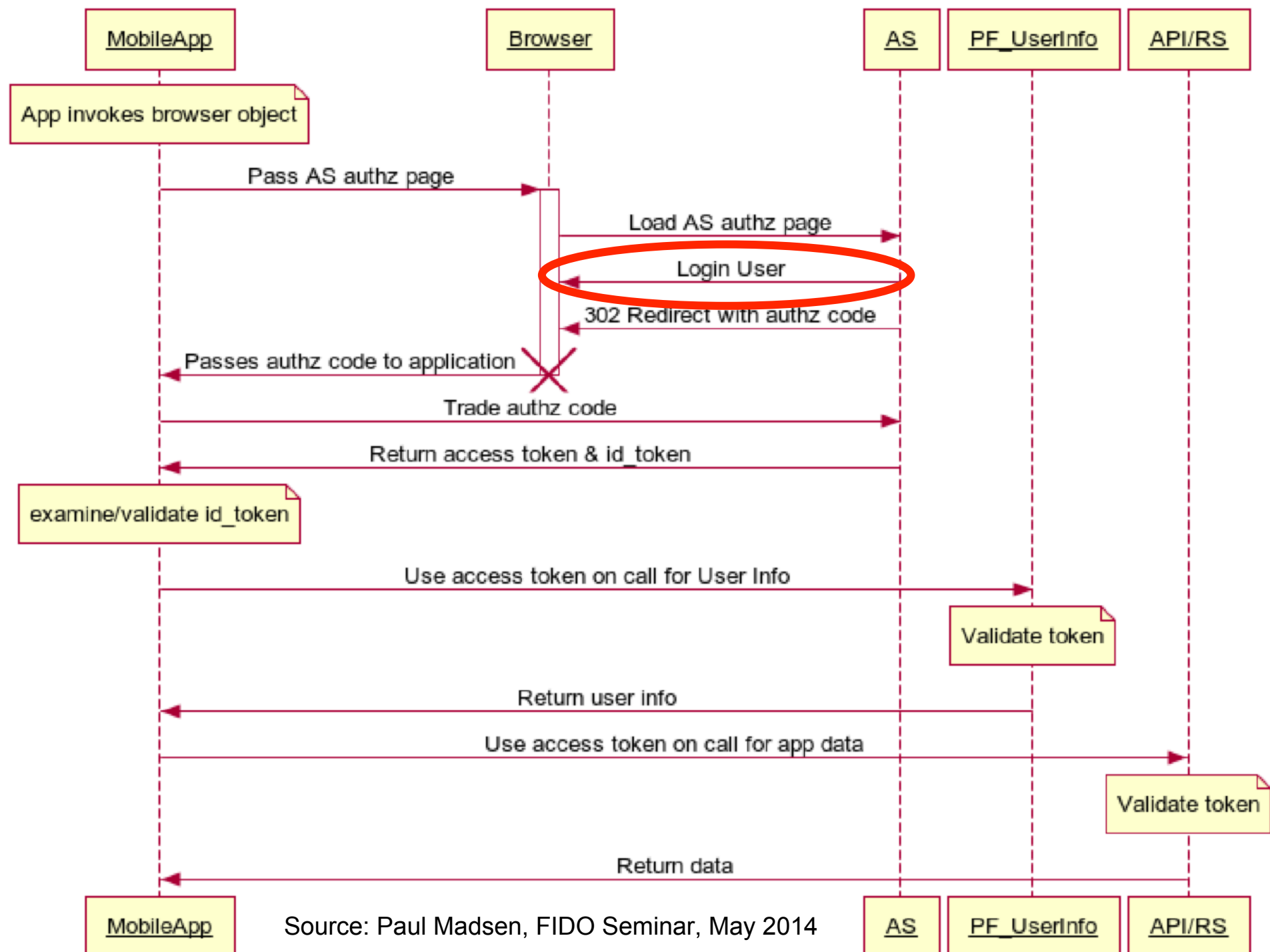


UAF Specifications



FIDO & Federation





Complementary

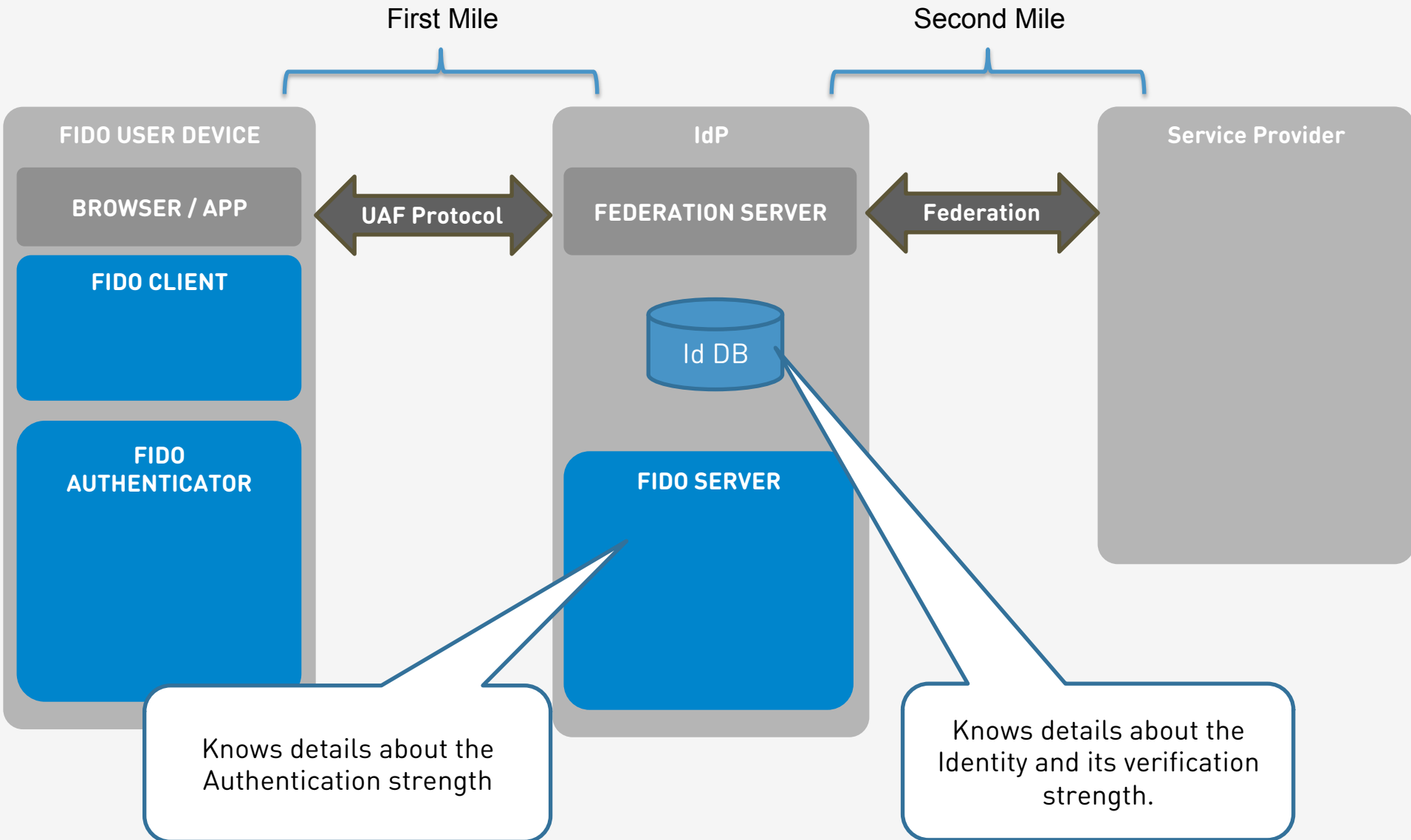
- FIDO

- Insulates authentication server from specific authenticators
- Focused solely on *primary* authentication
- Does not support attribute sharing
- Can communicate details of authentication to server

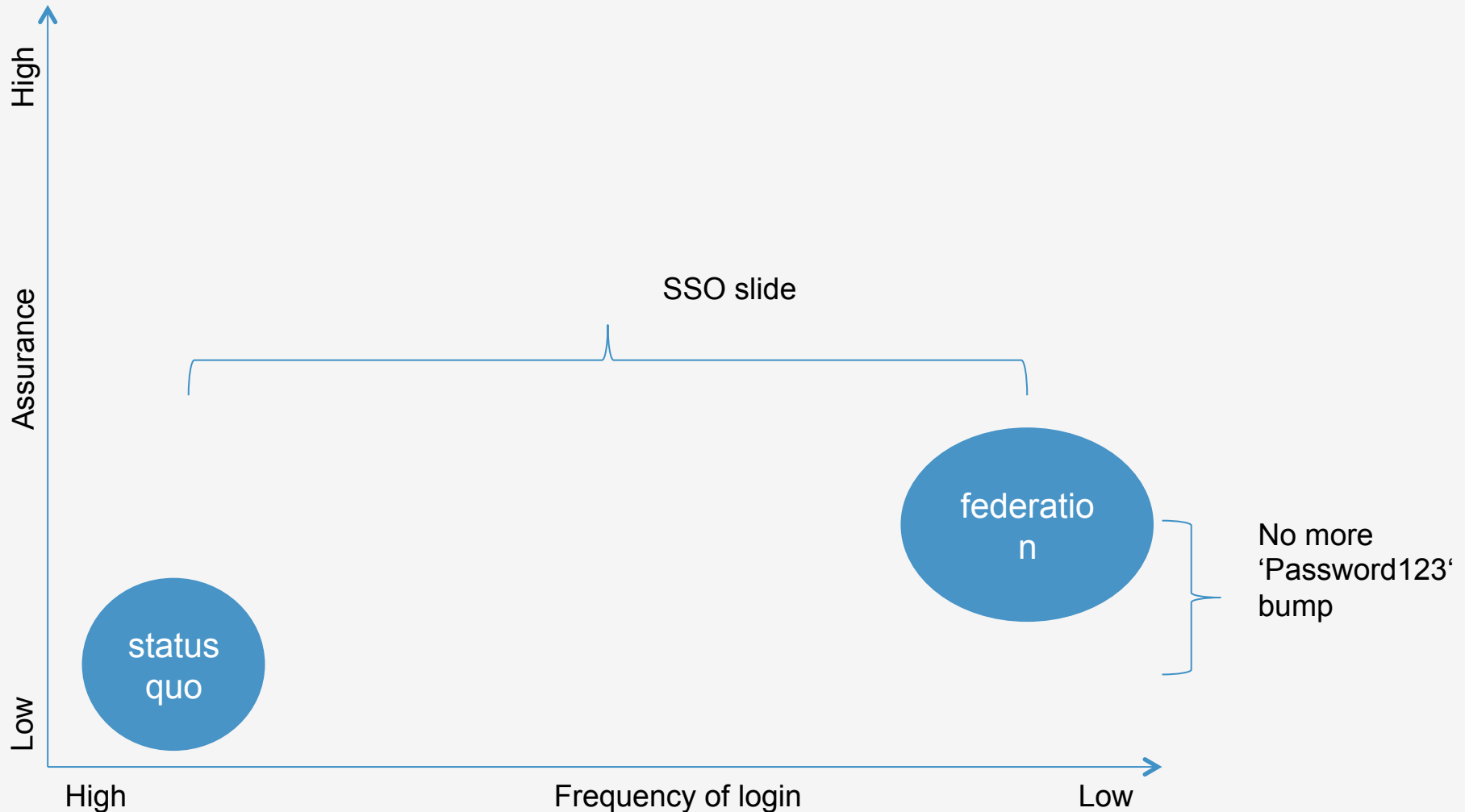
- Federation

- Insulates applications from identity providers
- Does not address primary authentication
- Does enable secondary authentication & attribute sharing
- Can communicate details of authentication from IdP to SP

FIDO & Federation

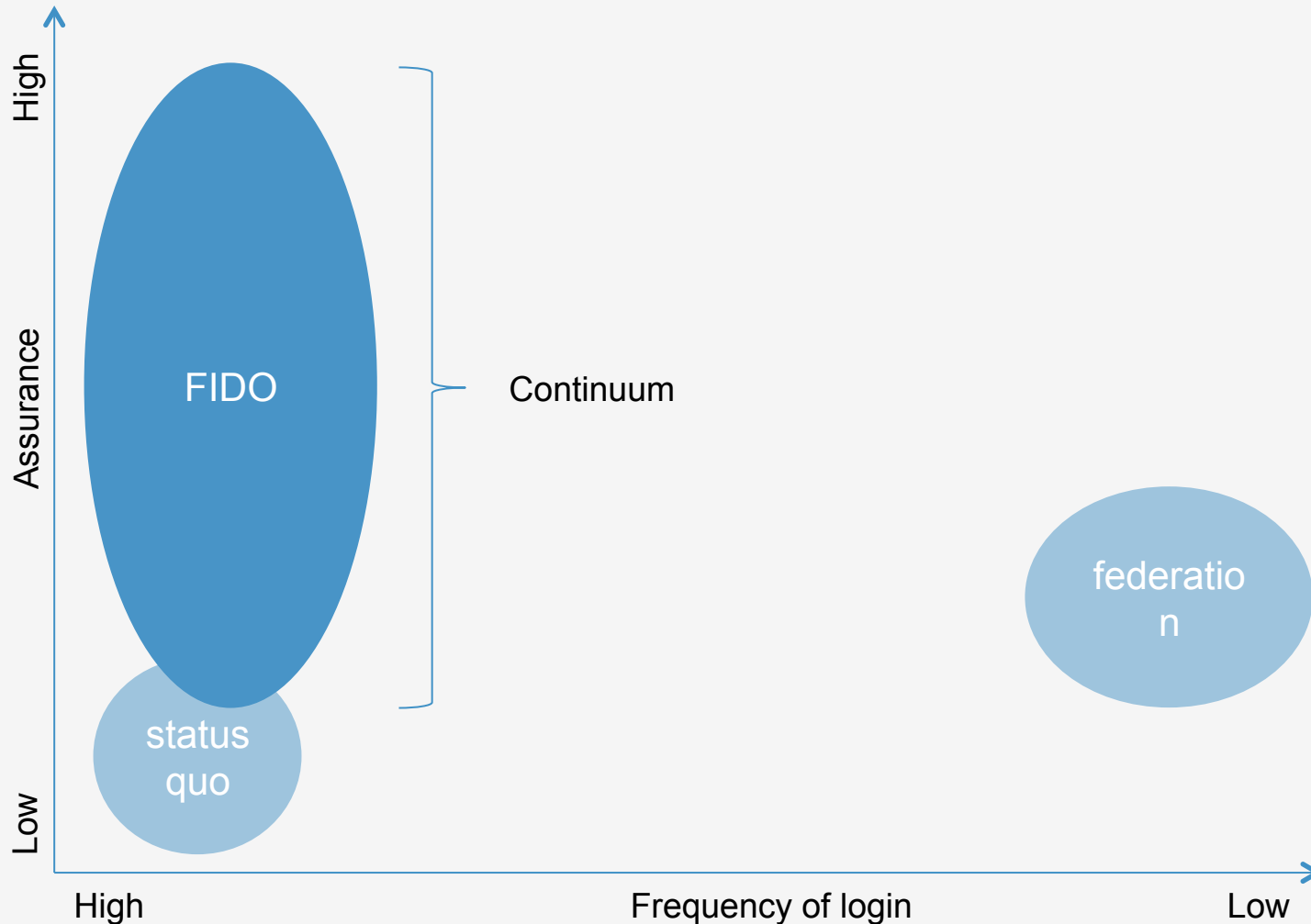


FIDO & Federation

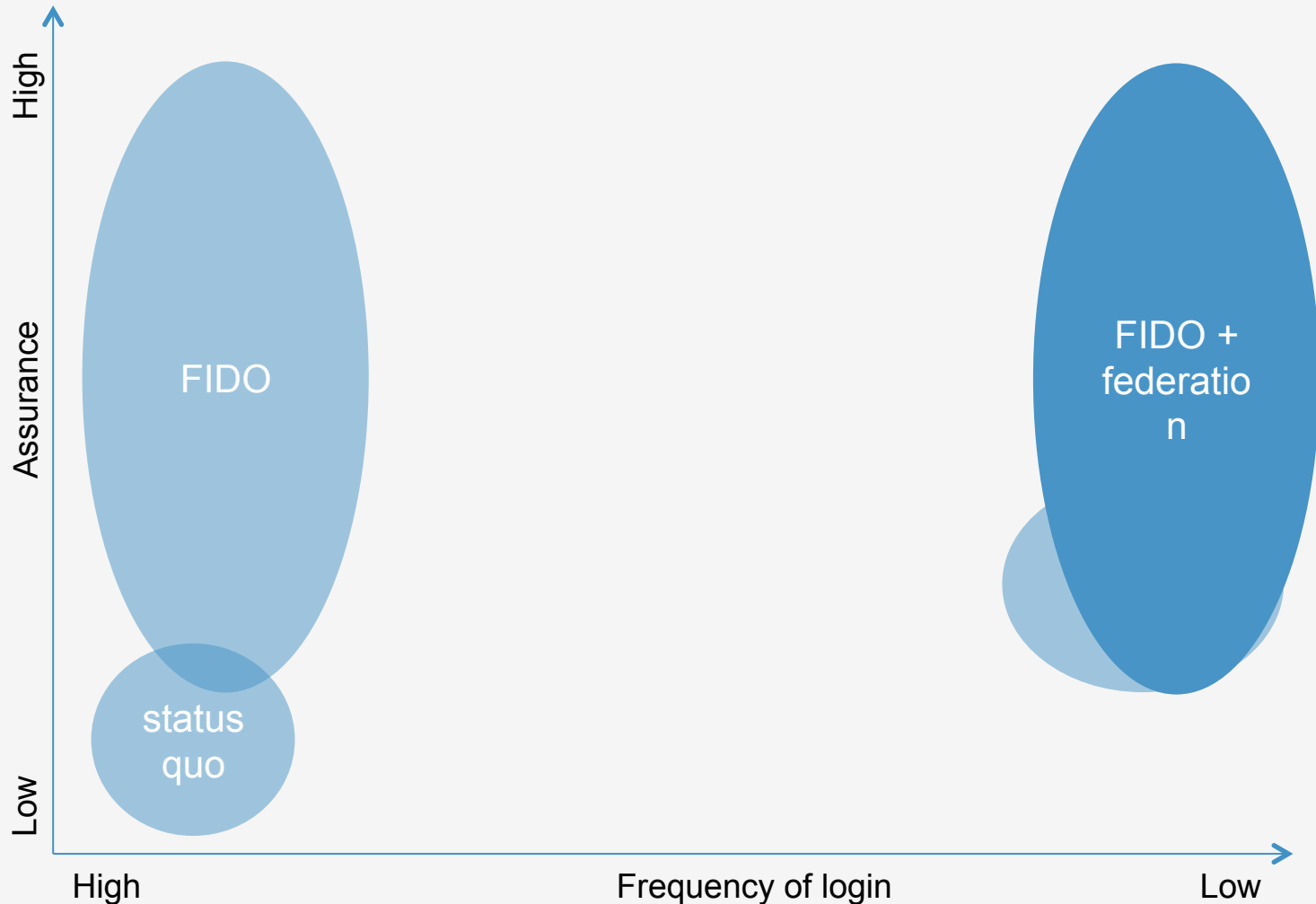


Source: Paul Madsen, FIDO Seminar, May 2014

FIDO & Federation



FIDO & Federation



Source: Paul Madsen, FIDO Seminar, May 2014

FIDO at Industry Event – Readiness



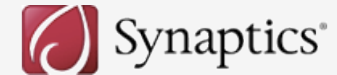
SIM as Secure Element



Fingerprint, TEE, Mobile



Speaker Recognition



Mobile via NFC



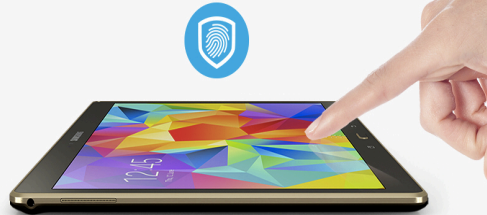
PIN + MicroSD



USB



FIDO Ready™ Products Shipping today



OEM Enabled: Samsung Galaxy S5 smartphone & Galaxy Tab S tablets

OEM Enabled: Lenovo ThinkPads with Fingerprint Sensors



Clients available for these operating systems:



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

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

FIDO is used Today

Alipay Offering Fingerprint Payment Partnering with Samsung

July 16, 2014 By CIW Team — [Leave a Comment](#)

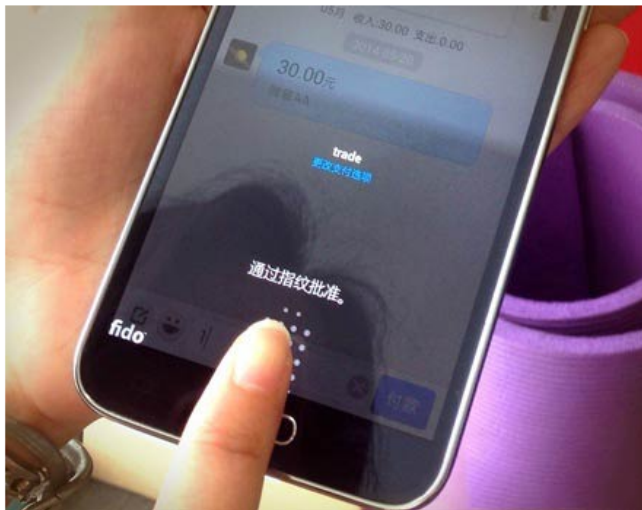


Hello there! If you are new here, you might want to [subscribe](#) to this topic.

+1

follow

Recommend



[Alipay](#) announced its cooperation with [Samsung](#) Galaxy S5 to offer Alipay users payment by fingerprint.

PayPal and Samsung launch FIDO authentication and fingerprint payments for Samsung Galaxy S5



By [Adam Vrankuli](#)

[Tweet](#)

February 25, 2014 - The [FIDO Alliance](#) has announced the first deployment of FIDO authentication and biometric fingerprint payment options, through a new collaboration between PayPal and Samsung, for the shiny new Galaxy S5 smartphone.

According to the alliance, users of the Galaxy S5 can now login and shop via fingerprint in online, mobile and in-store payments wherever PayPal is accepted. This is made possible through FIDO Ready software and a new embedded fingerprint sensor. Per FIDO specifications, the only information a user's device shares with PayPal is a unique encrypted key that allows PayPal to verify the identity of the customer without having to store any biometric information on PayPal servers.

Conclusion

- Different authentication use-cases lead to different authentication requirements
 - Today, we have authentication silos
 - FIDO separates user verification from authentication protocol and hence supports all user verification methods
 - FIDO supports scalable security and convenience
 - User verification data is known to Authenticator only
 - FIDO complements federation
- ➔ Consider developing or piloting FIDO-based authentication solutions