



U2F & UAF Tutorial

How Secure is Authentication?

How an epic blunder handed a hand of password crackers
Engineers flout universal taboo by encr...

50,000 lost after a hacker
by Paul Du
FILED UNDER

LivingSocial largish pa userbase up to a da

That's rig password

At least L salted an

Naked Se copy of th

[Wikipedia](#)

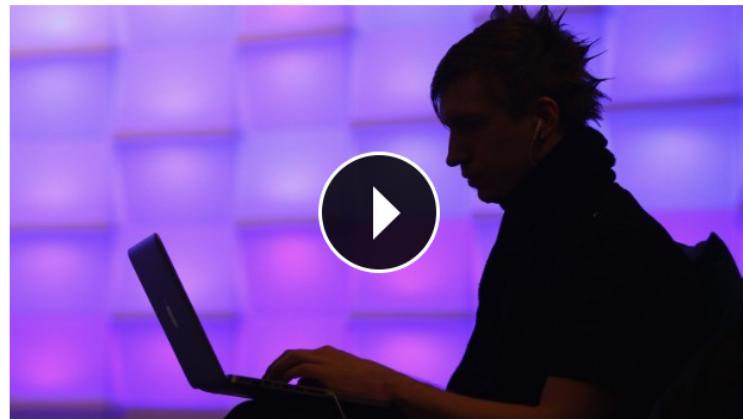
Four weeks ago, Adobe disclosed a [sustained corporate network](#) that threatened to spawn meaner malware attacks by giving criminals a raw source code for the company's widely used ColdFusion applications. Now, researchers a same breach could significantly strengthen crackers' collective hand by revealing a staggering 1.2 billion passcodes used over the years by 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 by stored passwords to be protected with another one-way cryptographic hashing algorithm that can't be converted back into the original password.

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 names" that thwart cracking 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

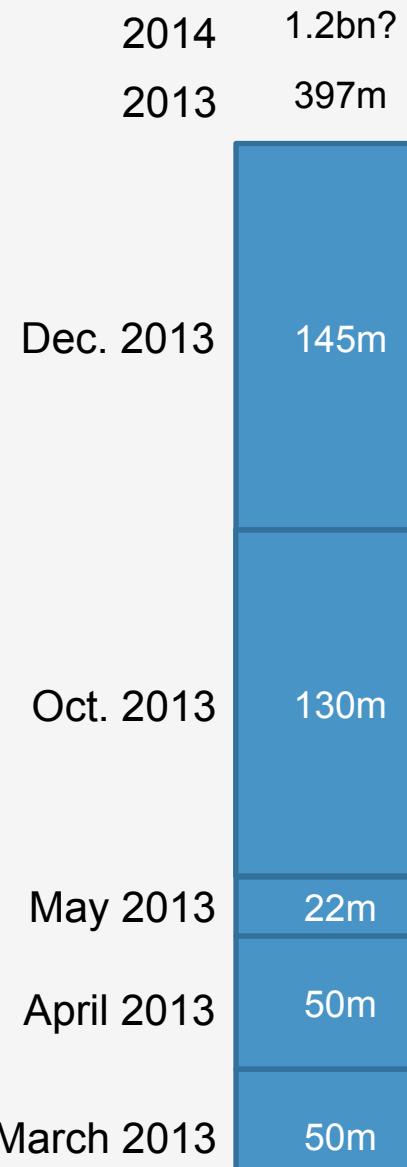
NEW YORK (CNNMoney)

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.

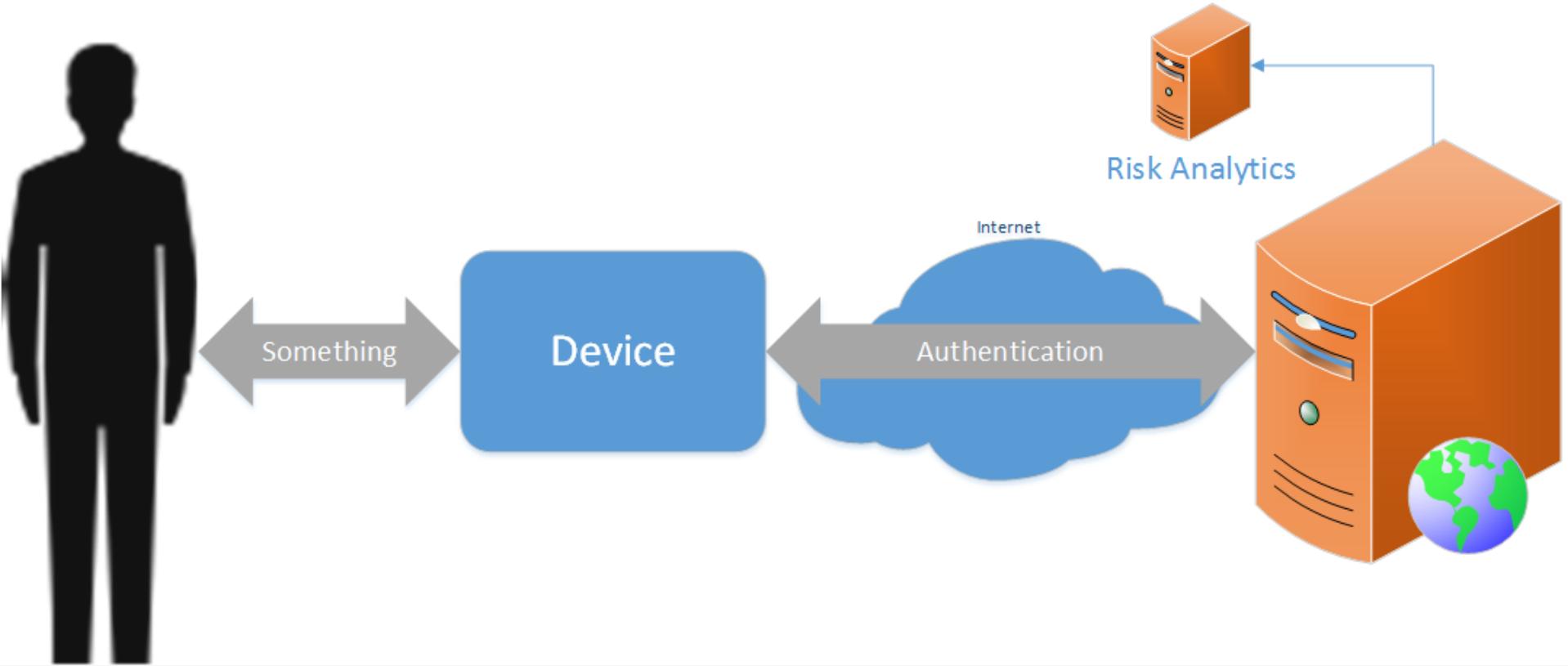
There's [no need to panic at this point](#) -- Hold Security, the firm that discovered the theft, says the gang isn't in the business of stealing your bank account information. Instead, they make their money by sending out spam for bogus products like weight-loss pills.

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.

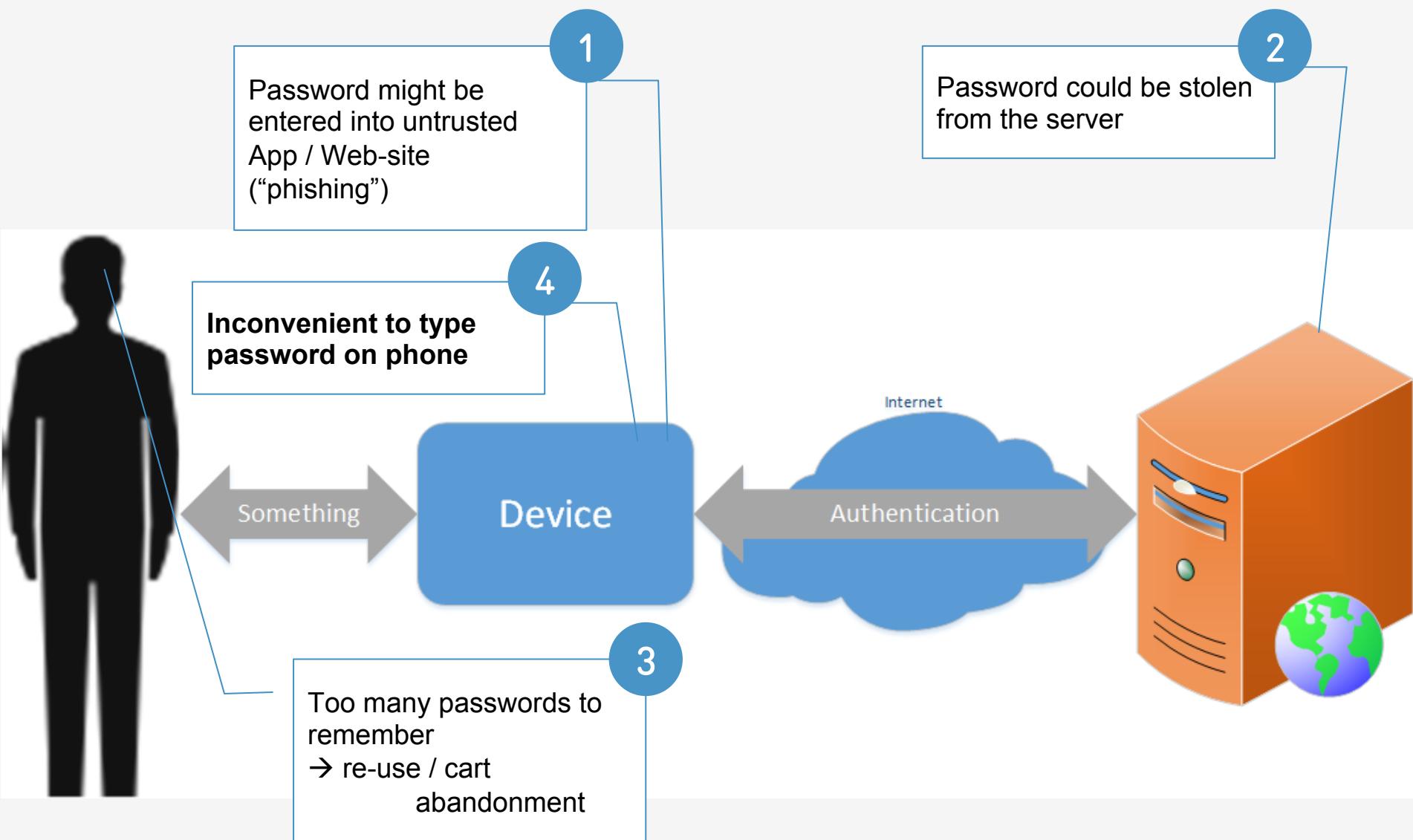
Hold Security founder Alex Holden told CNNMoney that the trove includes credentials gathered from over 420,000 websites -- both smaller sites as well as "household names" that thwart cracking by requiring crackers to pass individual password guesses through the same



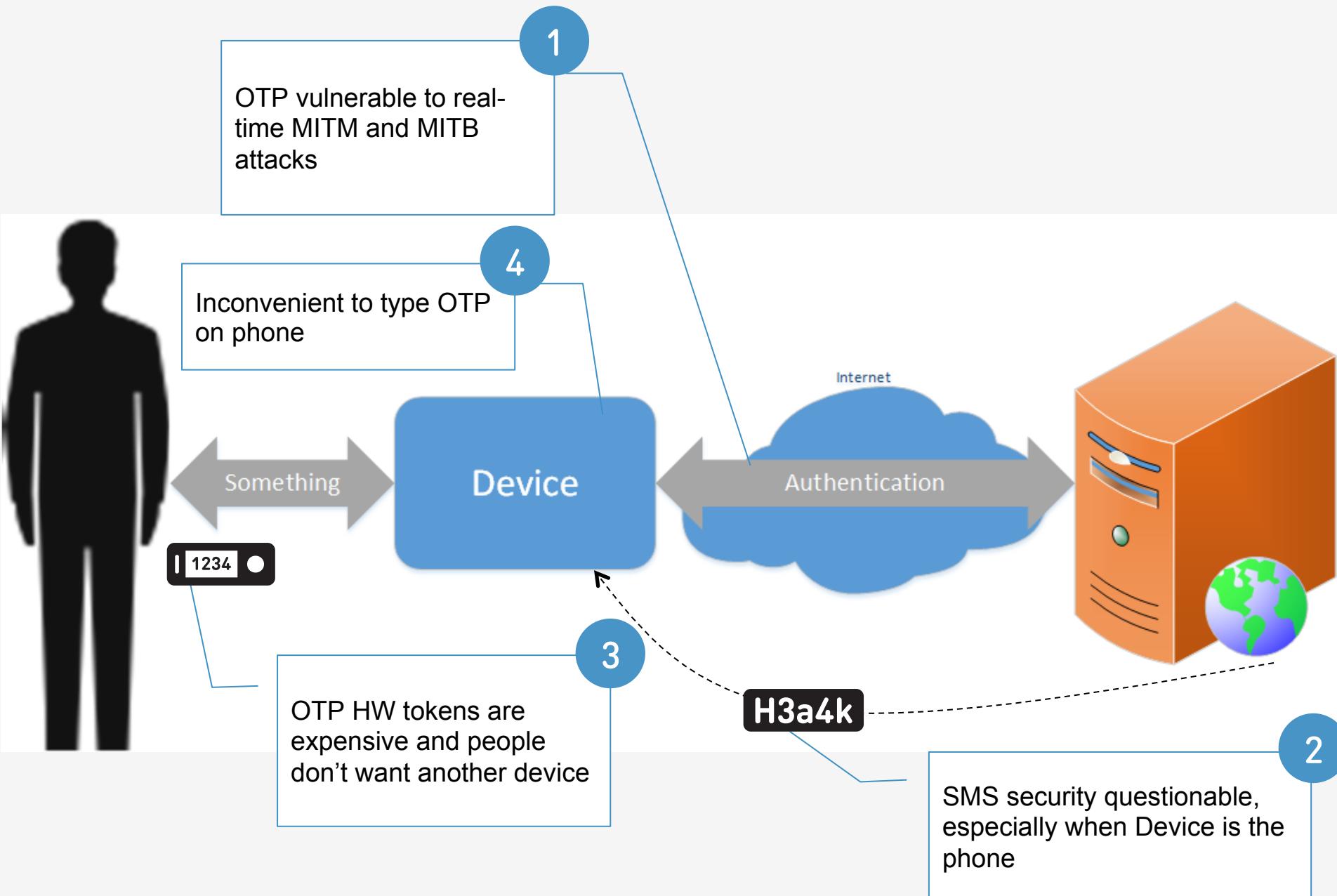
Cloud Authentication



Password Issues



OTP Issues



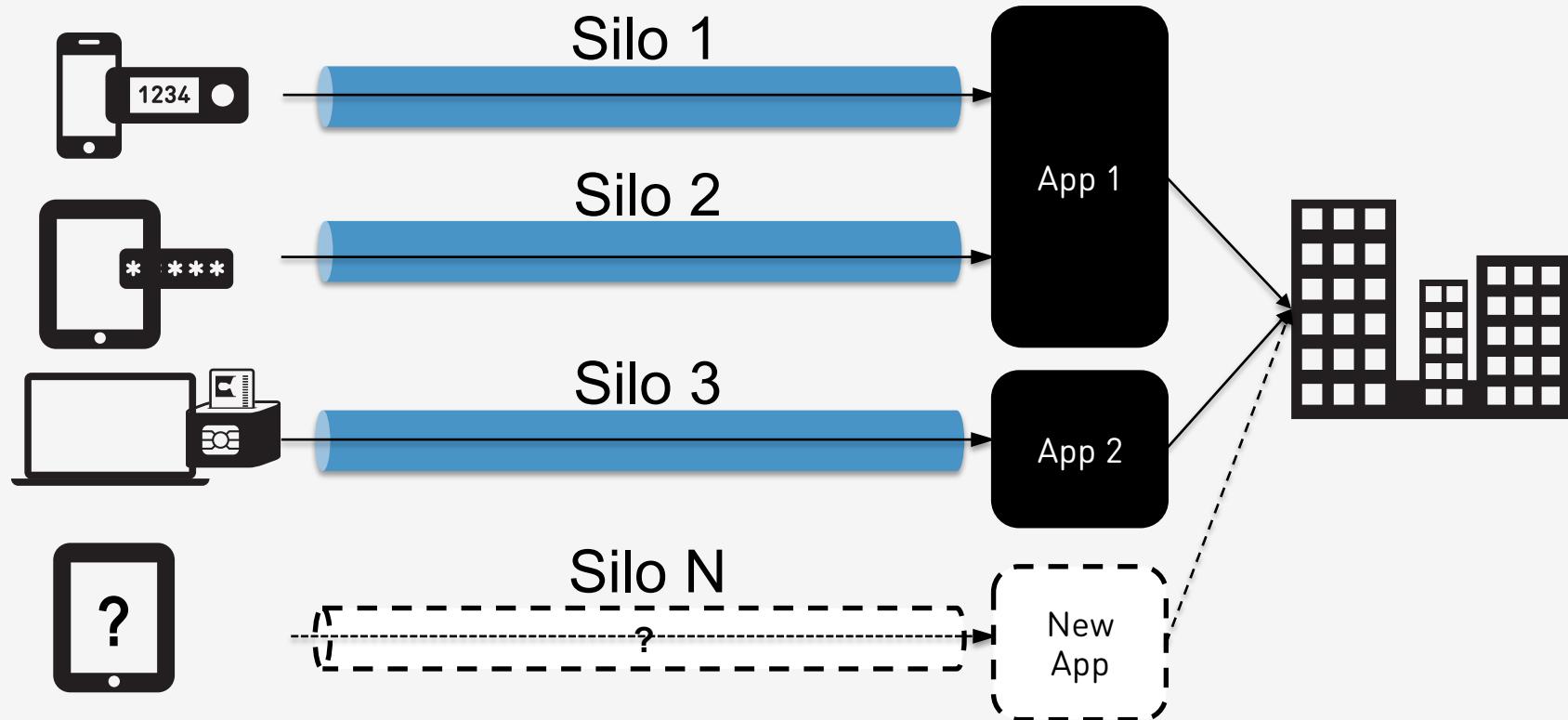
Implementation Challenge

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?

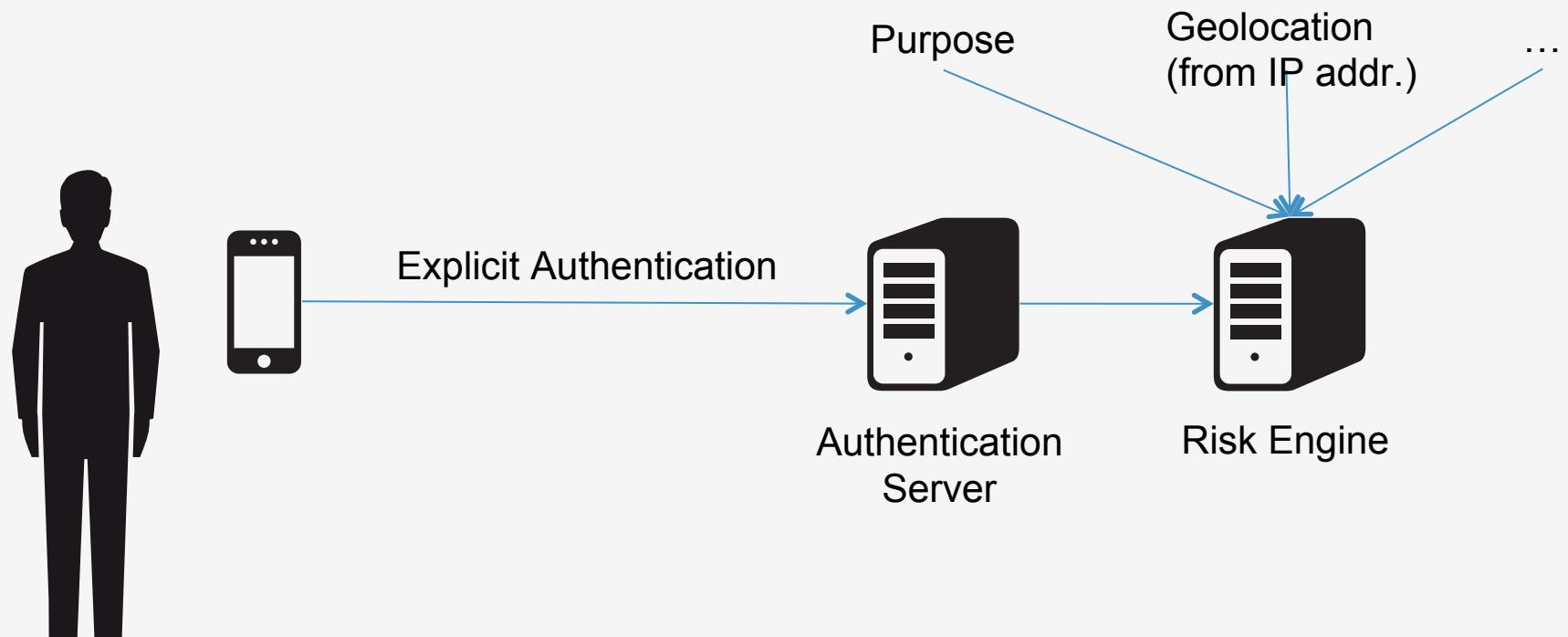
Do you want to share your dental record?

Authentication today:

Ask user for a password

(and perhaps a one time code)

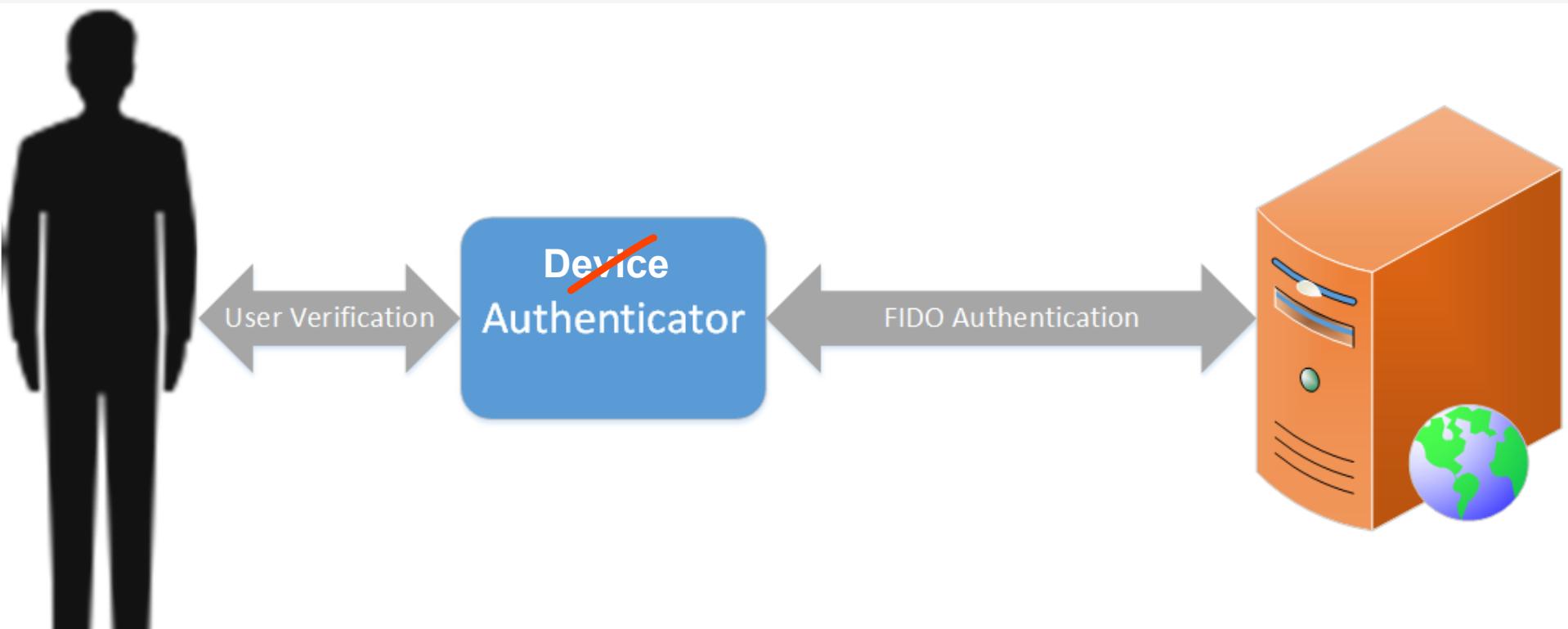
Authentication & Risk Engines



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?



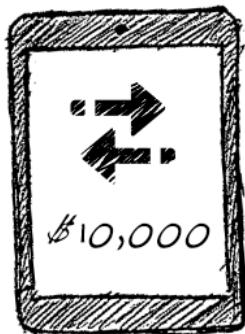
FIDO Experiences

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



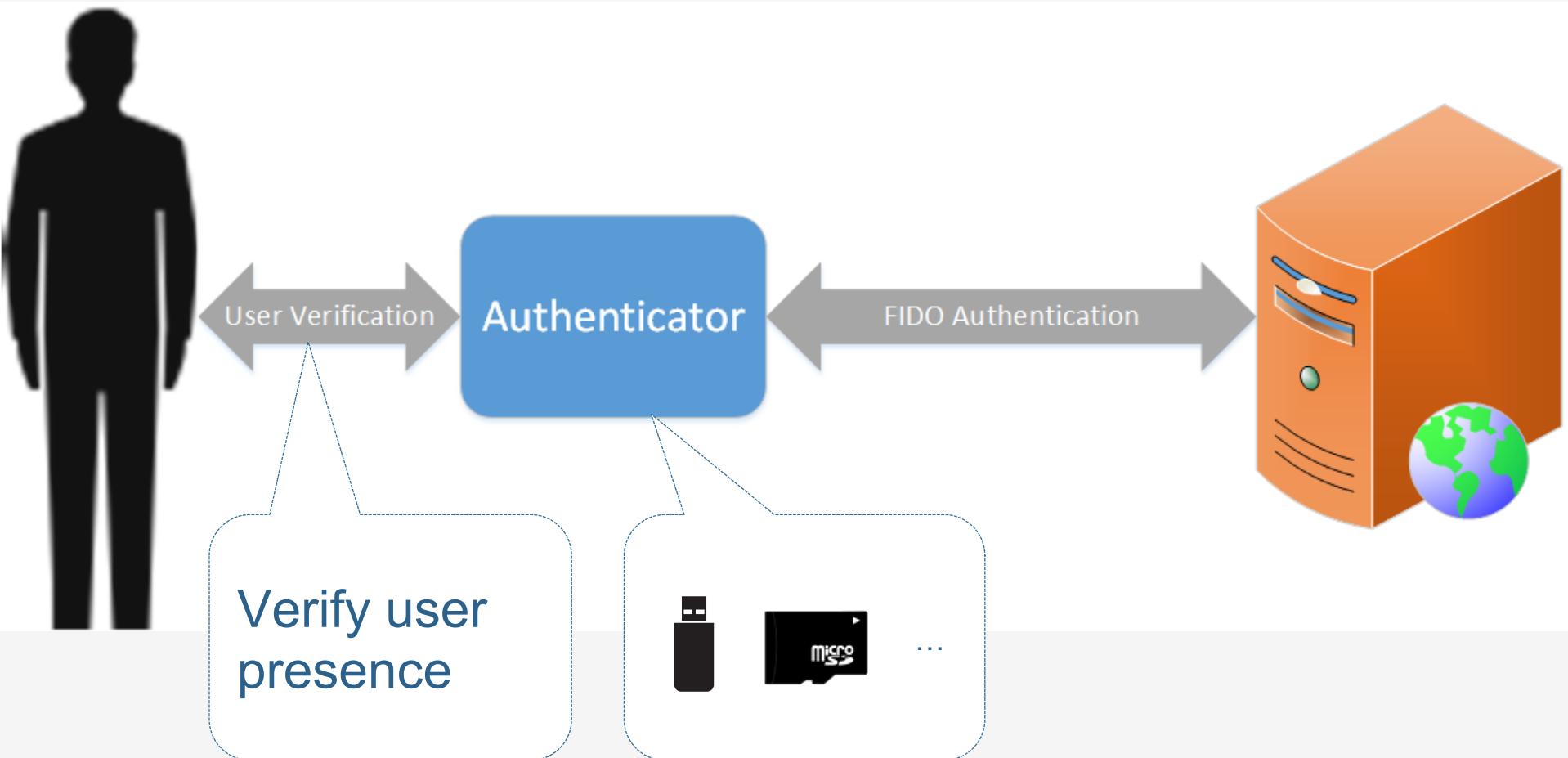
Insert Dongle, Press button



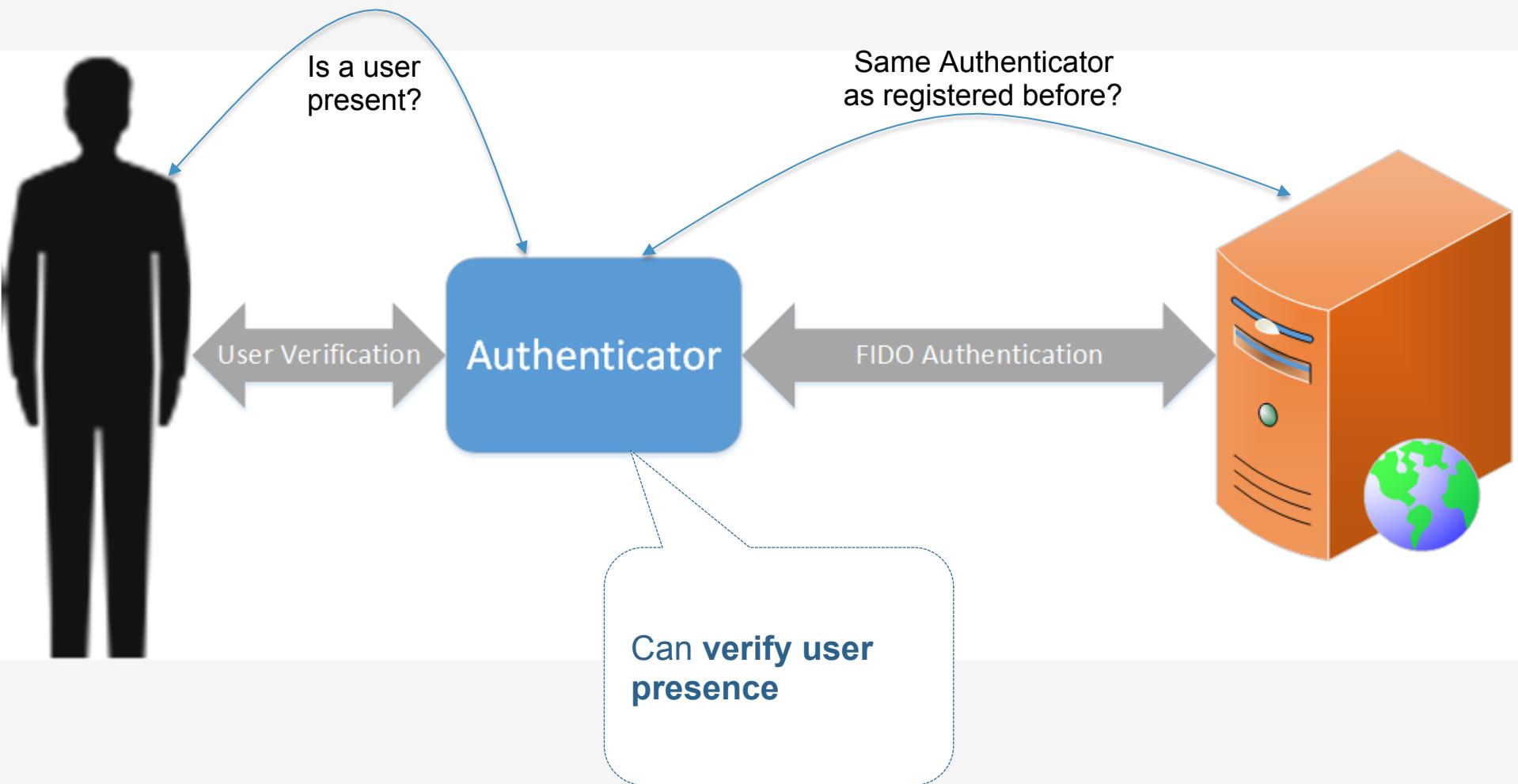
Done

FIDO Universal 2nd Factor (U2F)

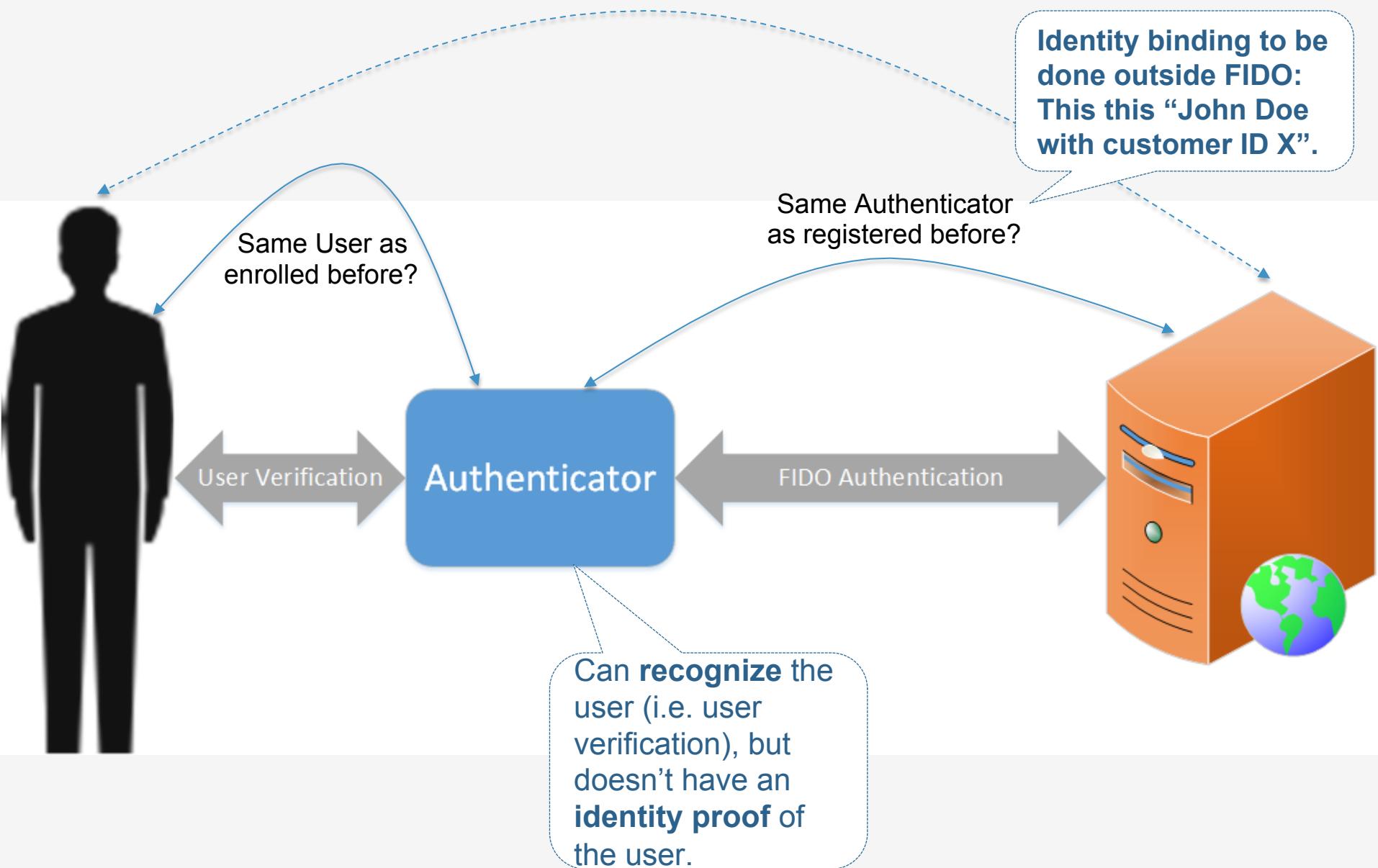
How does FIDO U2F work?



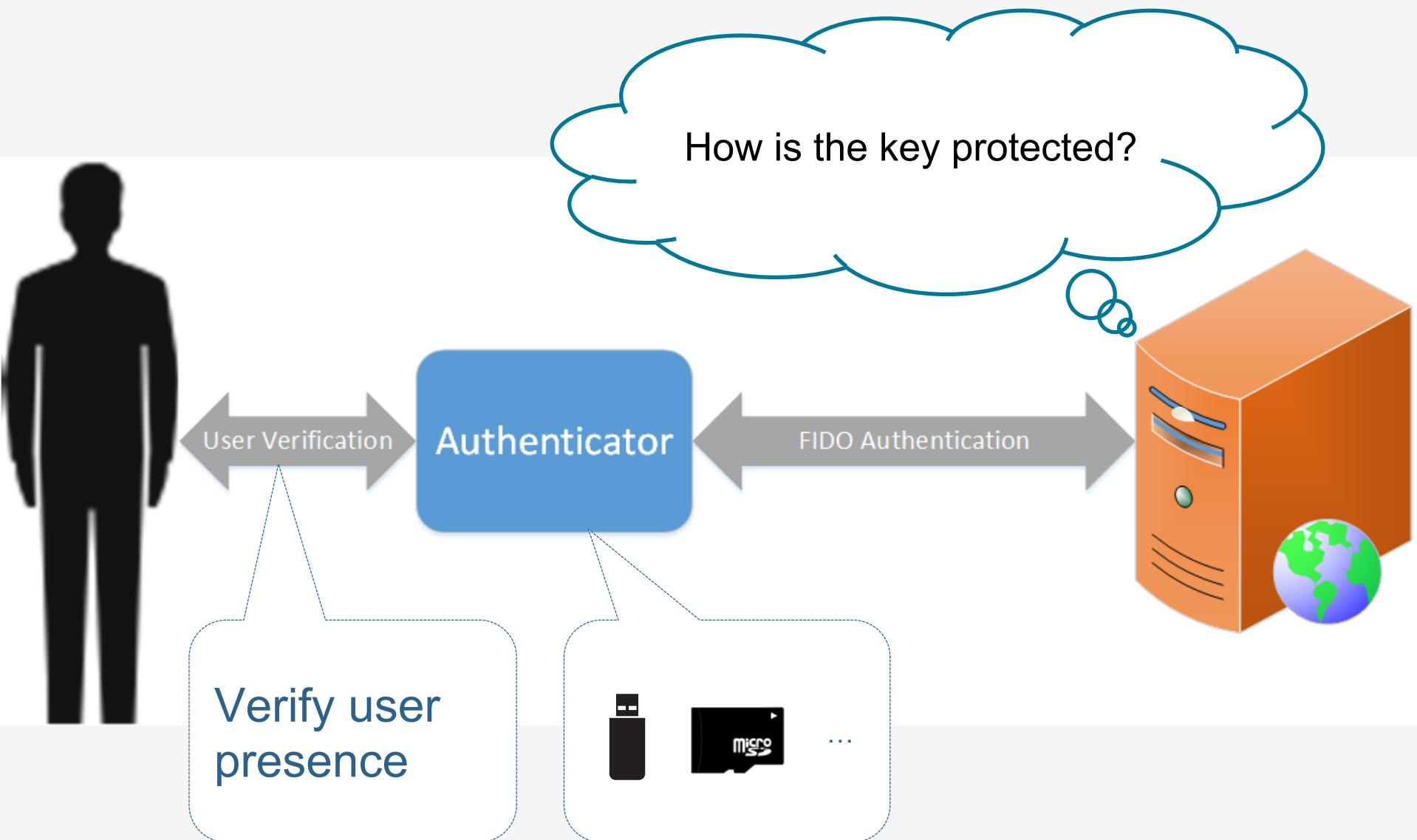
How does FIDO U2F work?



How does FIDO UAF work?



How does FIDO U2F work?

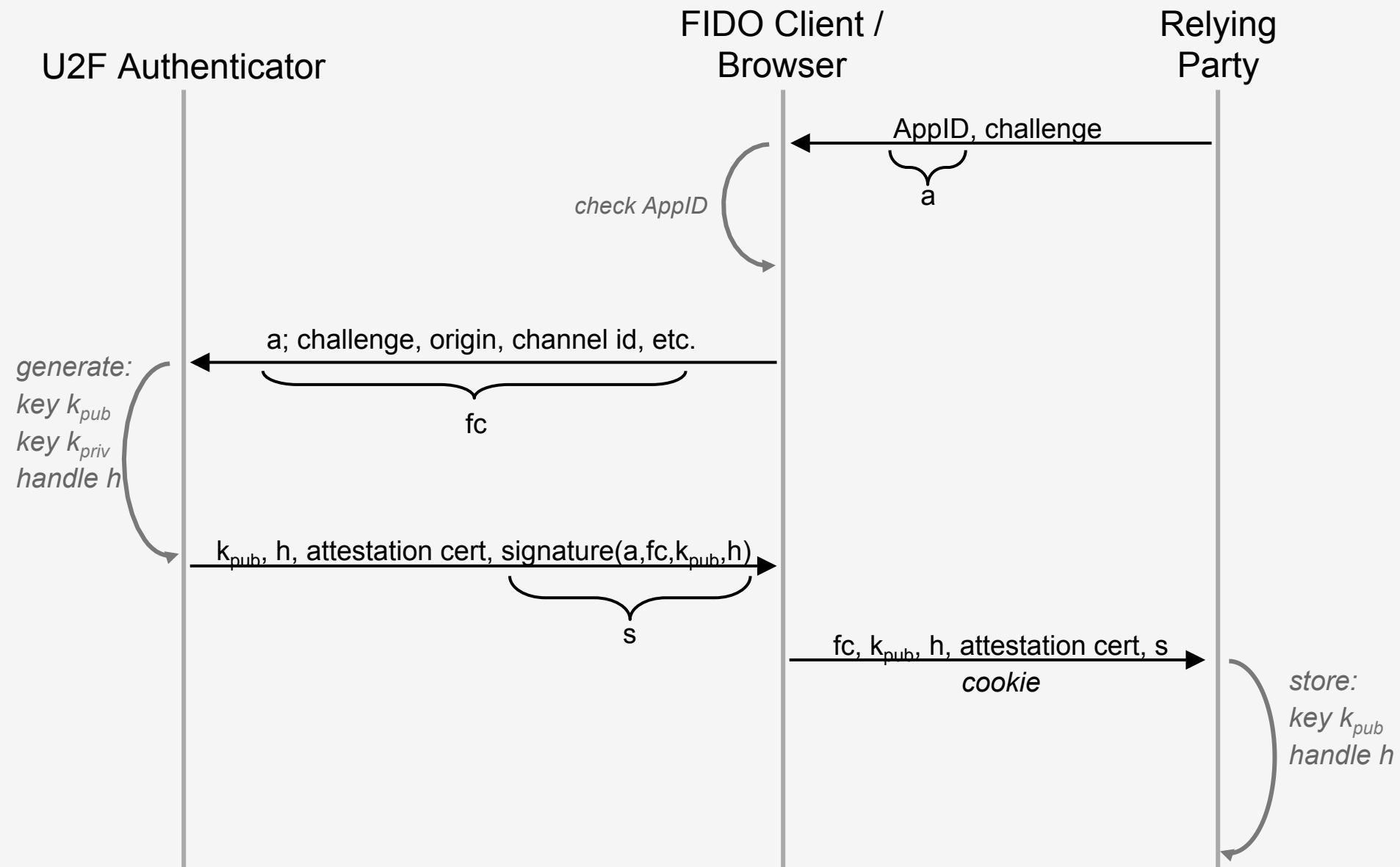


U2F Protocol

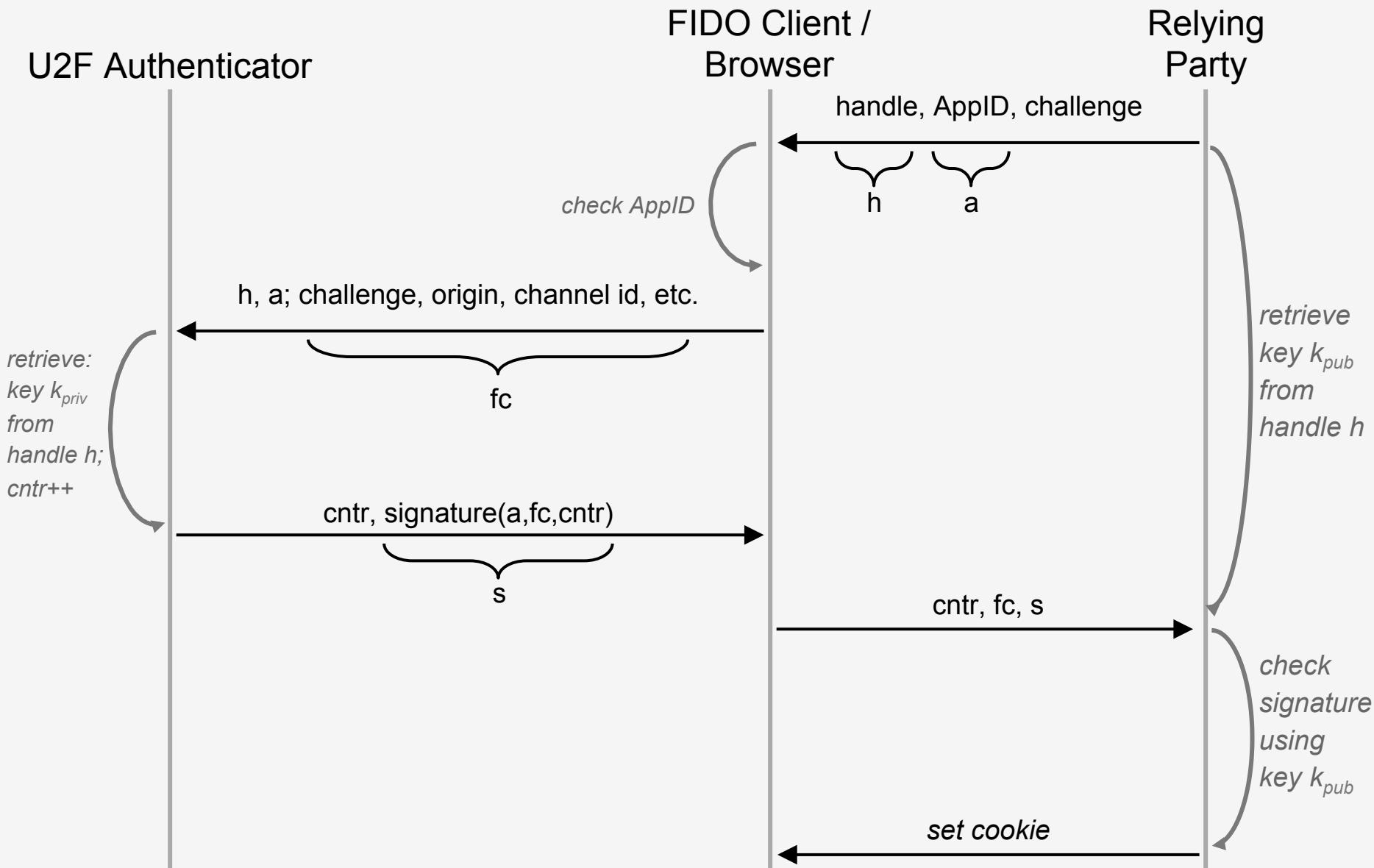
- **Core idea: Standard public key cryptography:**
 - User's device mints new key pair, gives public key to server
 - Server asks user's device to sign data to verify the user.
 - **One device, many services**, "bring your own device" enabled
- **Lots of refinement for this to be consumer facing:**
 - **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



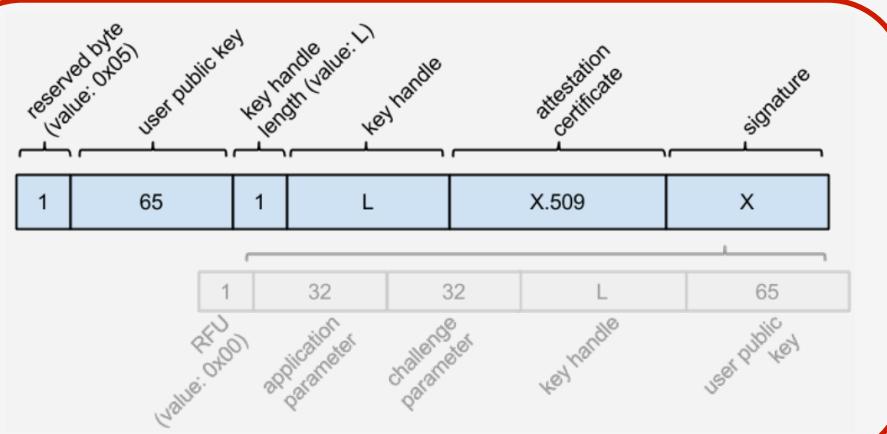
User Presence API: Registration

```
{ "typ": "register",
  "challenge": "KSDJsdASAS-AIS_AsS",
  "cid_pubkey": {
    "kty": "EC",
    "crv": "P-256",
    "x": "HzQwlfXX7Q4S5MtCRMzP09t0yWjBqR14tJ8",
    "y": "XVguGFLIZx1fXg375hi4-7-BxhMljw42Ht4"
  },
  "origin": "https://accounts.google.com"
}
```

```
'app_callback'
```

```
callback = function(response) {
  sendToServer(
    response['clientData'],
    response['token']
  );
};
```

```
s://www.google.com/facets.json' },
```



User Presence API: Auth.

```
nav = [
    {
        "typ": "authenticate",
        "challenge": "KSDJsdASAS-AIS_AsS",
        "cid_pubkey": {
            "kty": "EC",
            "crv": "P-256",
            "x": "HzQwlfXX7Q4S5MtCRMzP09t0yWjBqR14tJ8",
            "y": "XVguGFLIZx1fXg375hi4-7-BxhM1jw42Ht4"
        },
        "origin": "https://accounts.google.com"
    }
], "/facets.json",
call = [
    {
        "method": "POST",
        "url": "https://accounts.google.com/.well-known/_sadsAJDKLSAD' "
    }
],
callback = function(response) {
    sendToServer([
        response['clientId'],
        response['token']
    ]);
};
```

The diagram illustrates the structure of the challenge-response handshake. It shows two main components: the challenge message and the response message.

Challenge Message Structure:

- application parameter (32 bytes)
- user presence counter (1 byte)
- counter (4 bytes)
- challenge parameter (32 bytes)

Response Message Structure:

- user presence counter (32 bytes)
- signature (X bytes)

The challenge message is composed of the application parameter, user presence counter, counter, and challenge parameter. The response message is composed of the user presence counter and the signature.



Authentication Example

A screenshot of a web browser window displaying the Google Sign-in page. The browser has a blue header with snowflake icons and a logo. The address bar shows the URL <https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=false&co...>. The main content area features the Google logo and the tagline "One account. All of Google." Below this, a "Sign in to continue to Gmail" link is visible. The central part of the page contains a large user icon, followed by two input fields labeled "Email" and "Password". A blue "Sign in" button is positioned below the password field. To the left of the "Sign in" button is a checked checkbox labeled "Stay signed in". To the right is a "Need help?" link. At the bottom of the sign-in form, there is a "Create an account" link. Below the sign-in form, a banner reads "One Google Account for everything Google" and lists various Google services: GMail, Google Drive, YouTube, Google Photos, Google Play, and Google Sheets. The footer of the page includes links for "About Google", "Privacy & Terms", and "Help", along with a language selection dropdown set to "English (United States)".

Authentication Example

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

Google

One account. All of Google.

Sign in to continue to Gmail

nnlu2fdemo@gmail.com

•••••••

Sign In

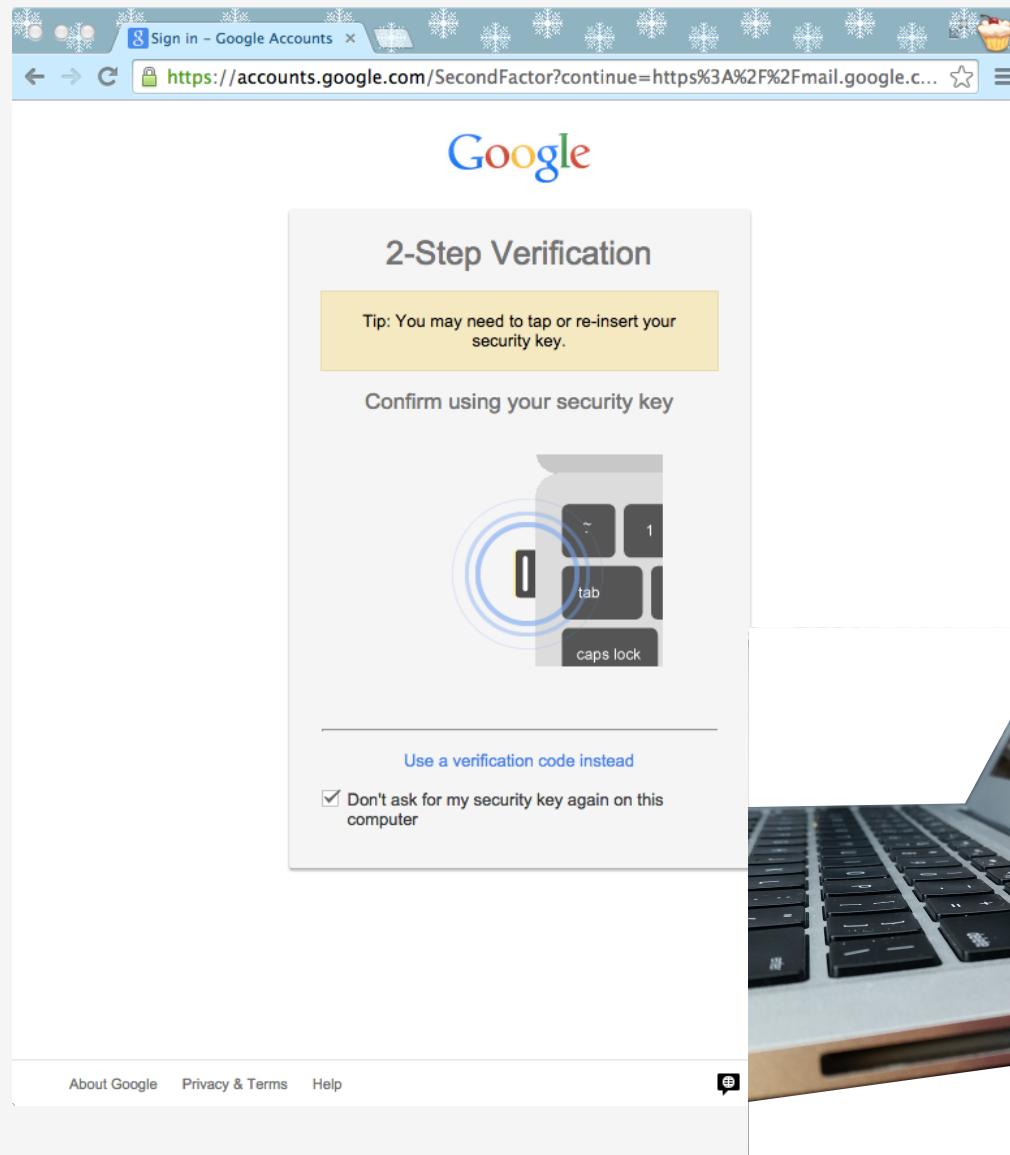
Stay signed in Need help?

Create an account

One Google Account for everything Google

About Google Privacy & Terms Help

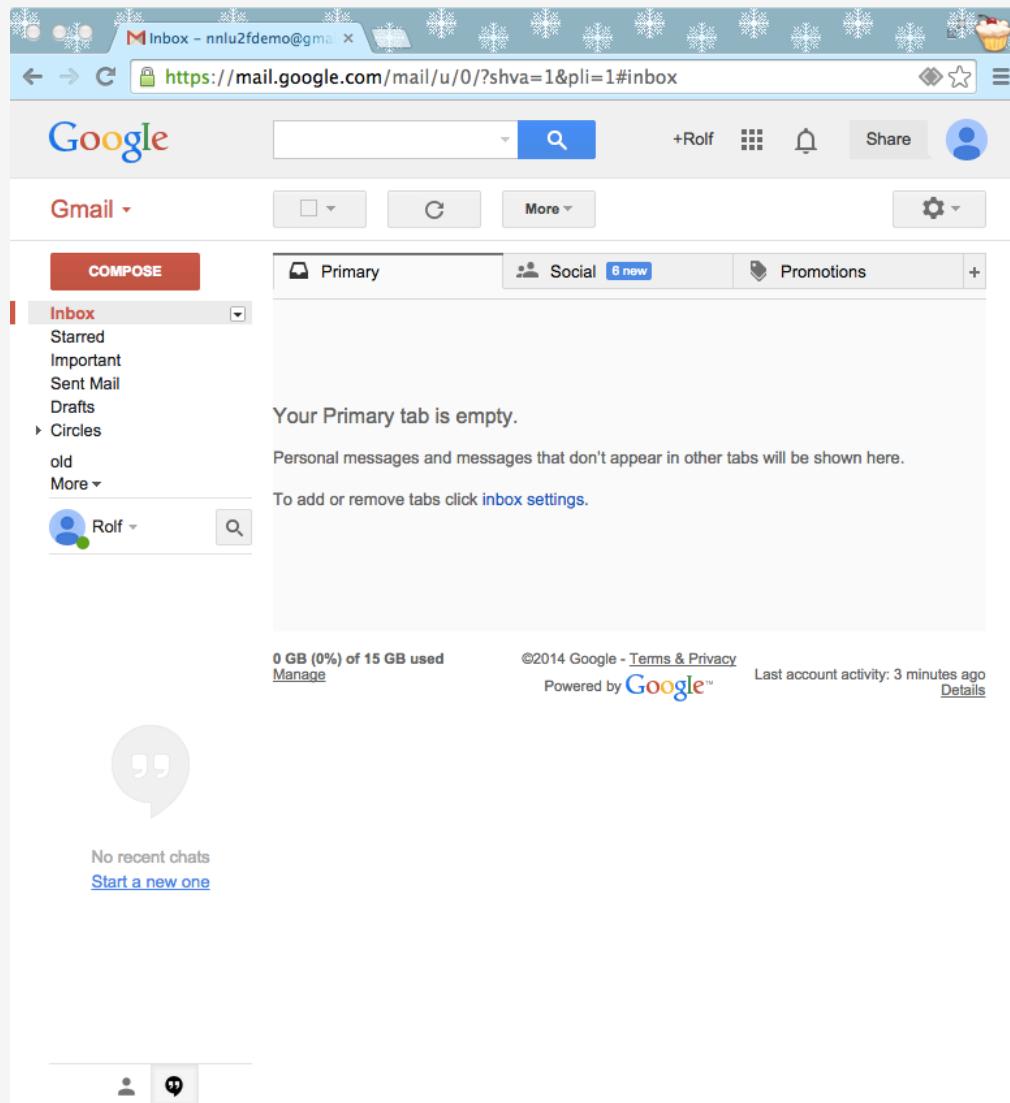
Authentication Example



The image shows a laptop screen displaying a Google 2-Step Verification page. The browser title bar reads "Sign in - Google Accounts" and the URL is "https://accounts.google.com/SecondFactor?continue=https%3A%2F%2Fmail.google.c...". The main content is a "Google" logo and a "2-Step Verification" box. Inside the box, there is a tip: "Tip: You may need to tap or re-insert your security key." Below this, it says "Confirm using your security key" and shows an illustration of a smartphone with a circular tap gesture on its screen. At the bottom of the verification box, there are two options: "Use a verification code instead" and a checked checkbox for "Don't ask for my security key again on this computer". The footer of the page includes links for "About Google", "Privacy & Terms", and "Help".



Authentication Example



FIDO Universal Authentication Framework (UAF)

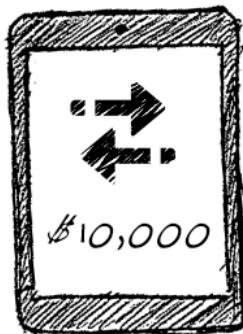
FIDO Experiences

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

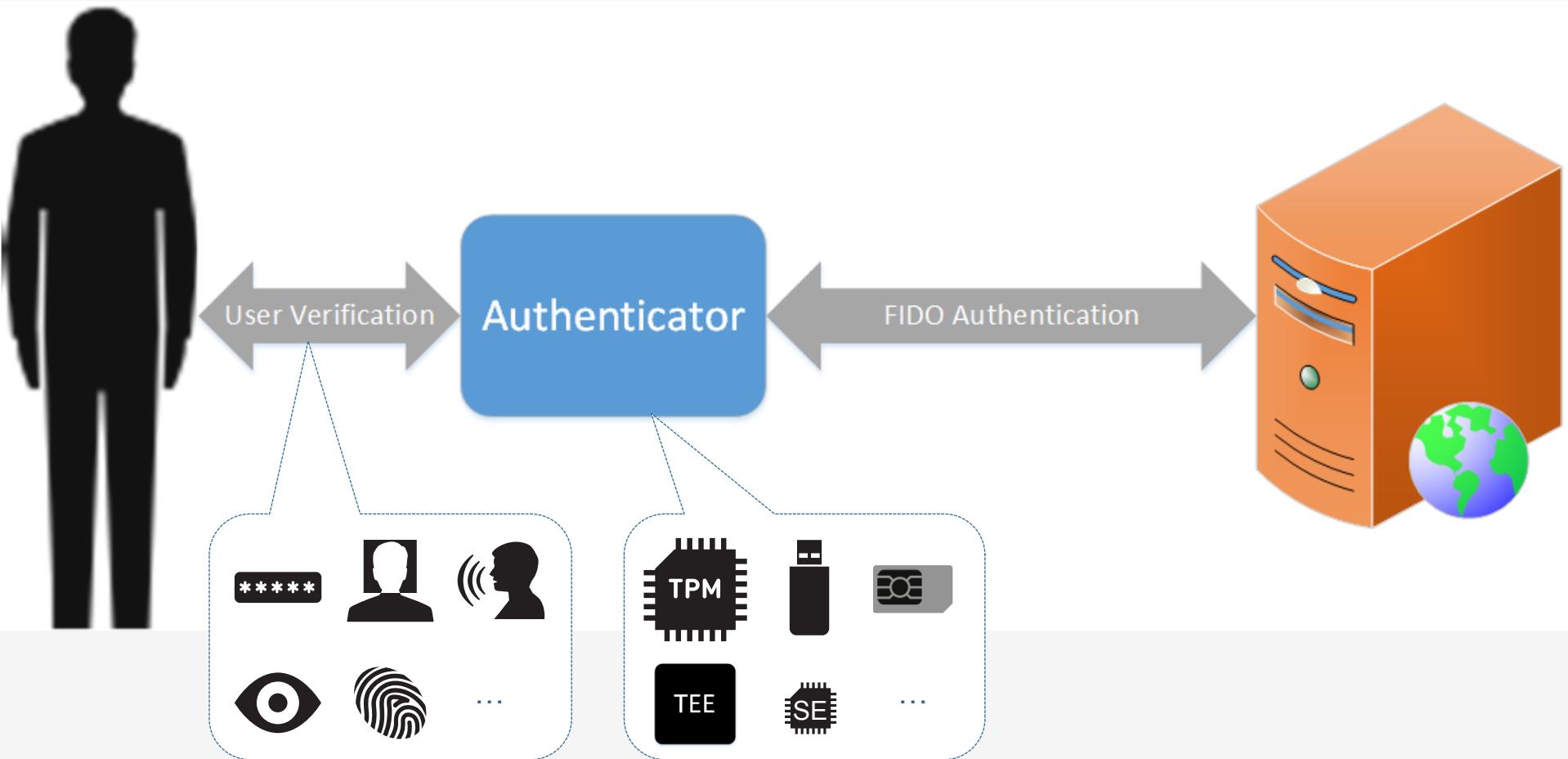


Insert Dongle, Press button

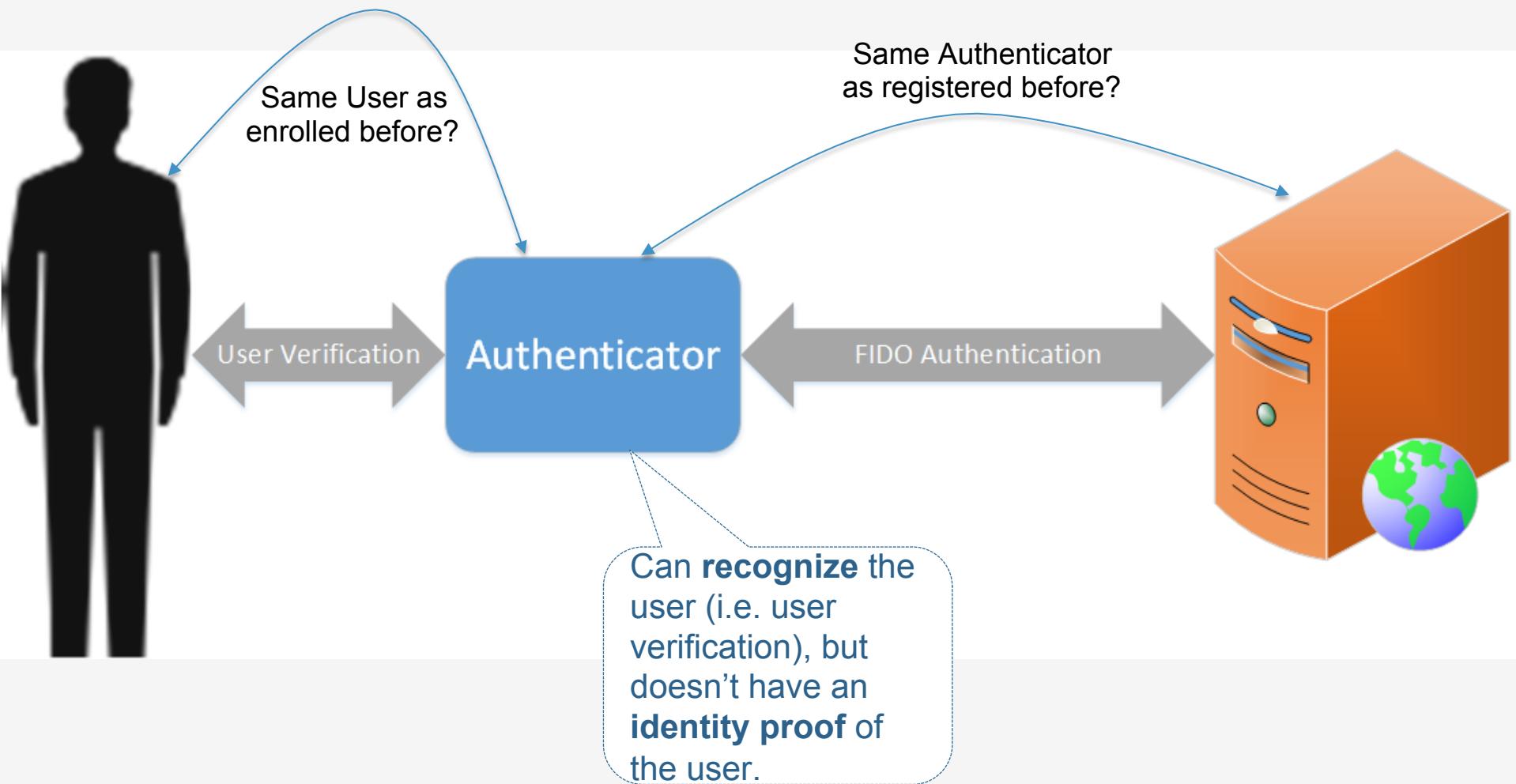


Done

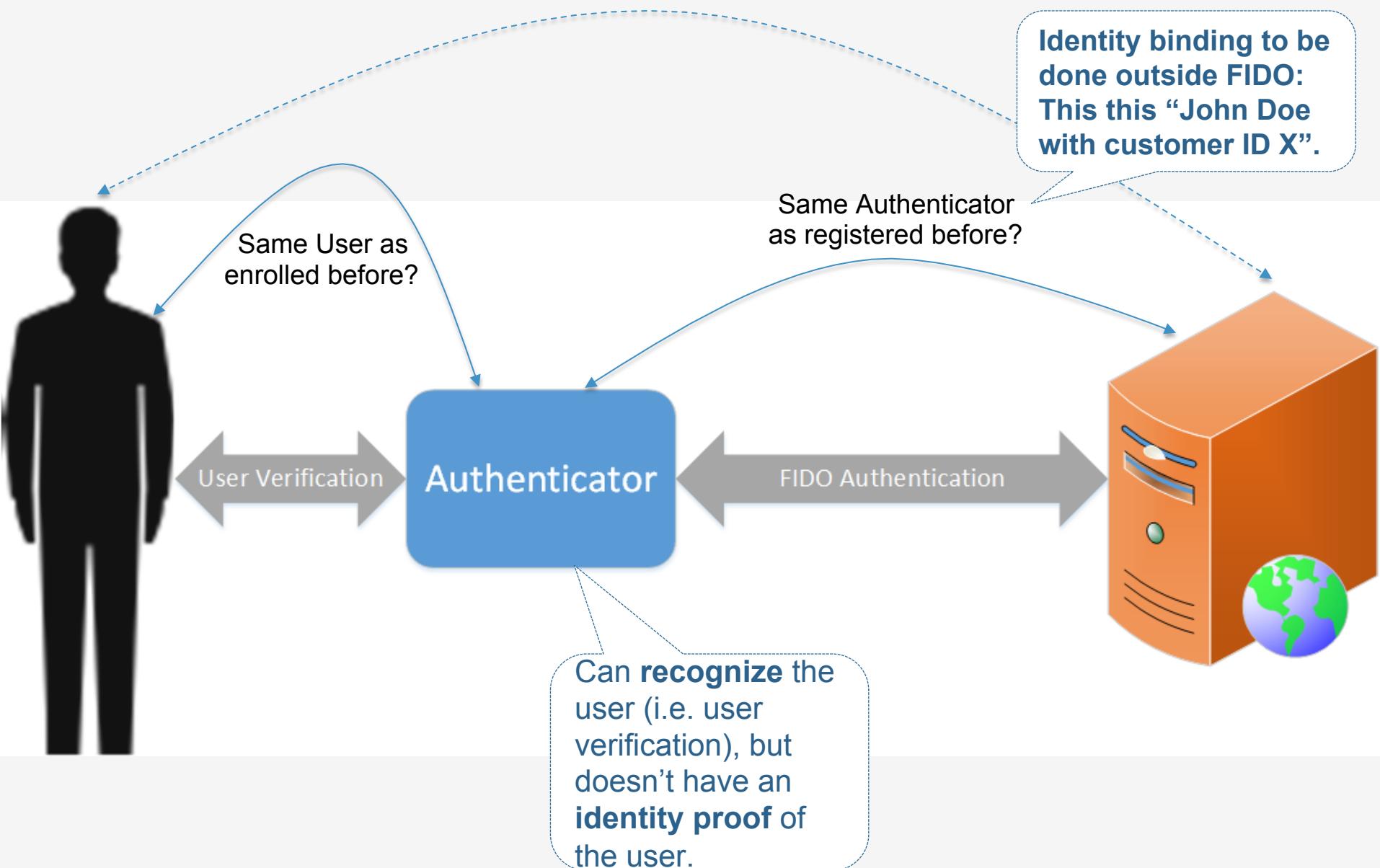
How does FIDO UAF work?



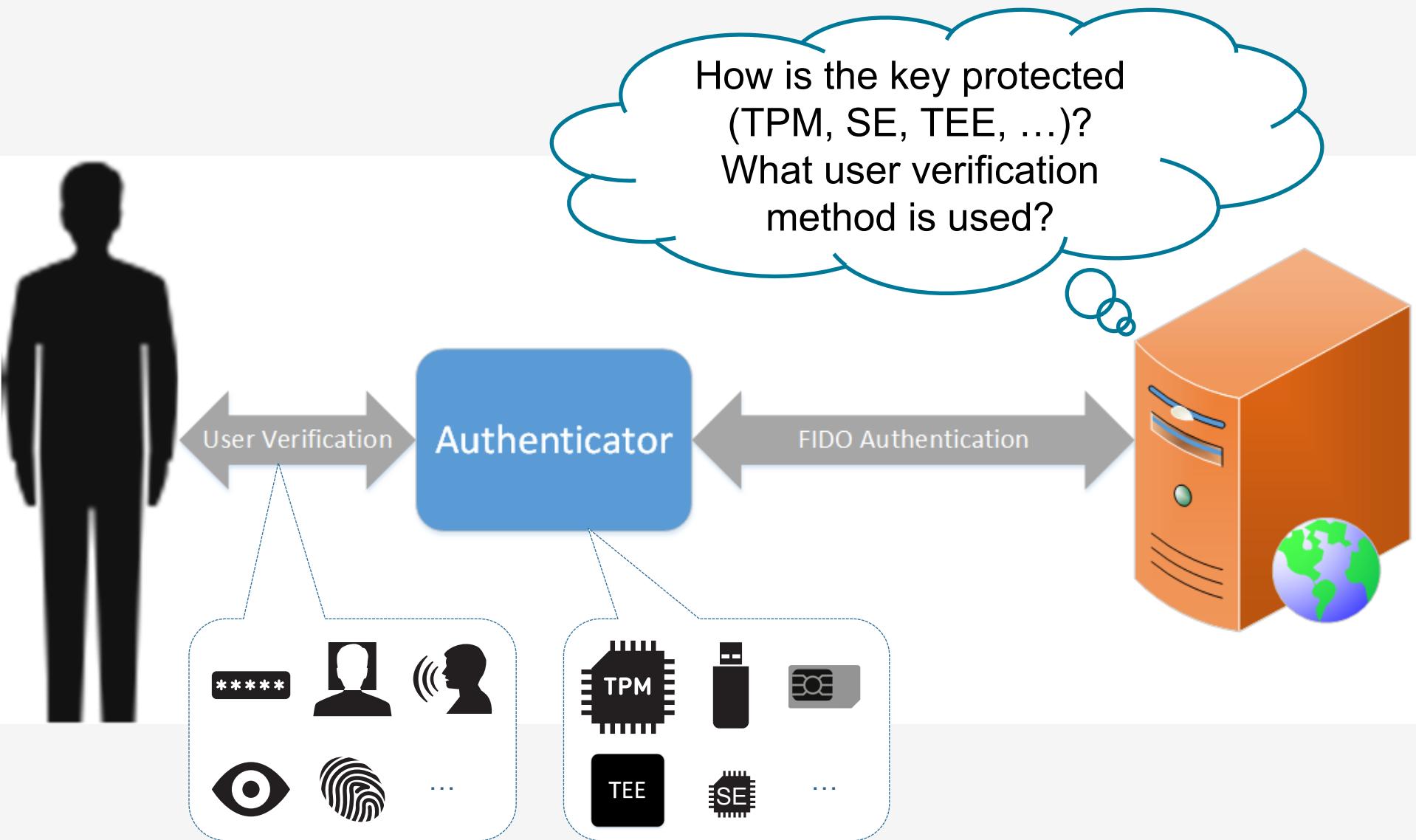
How does FIDO UAF work?



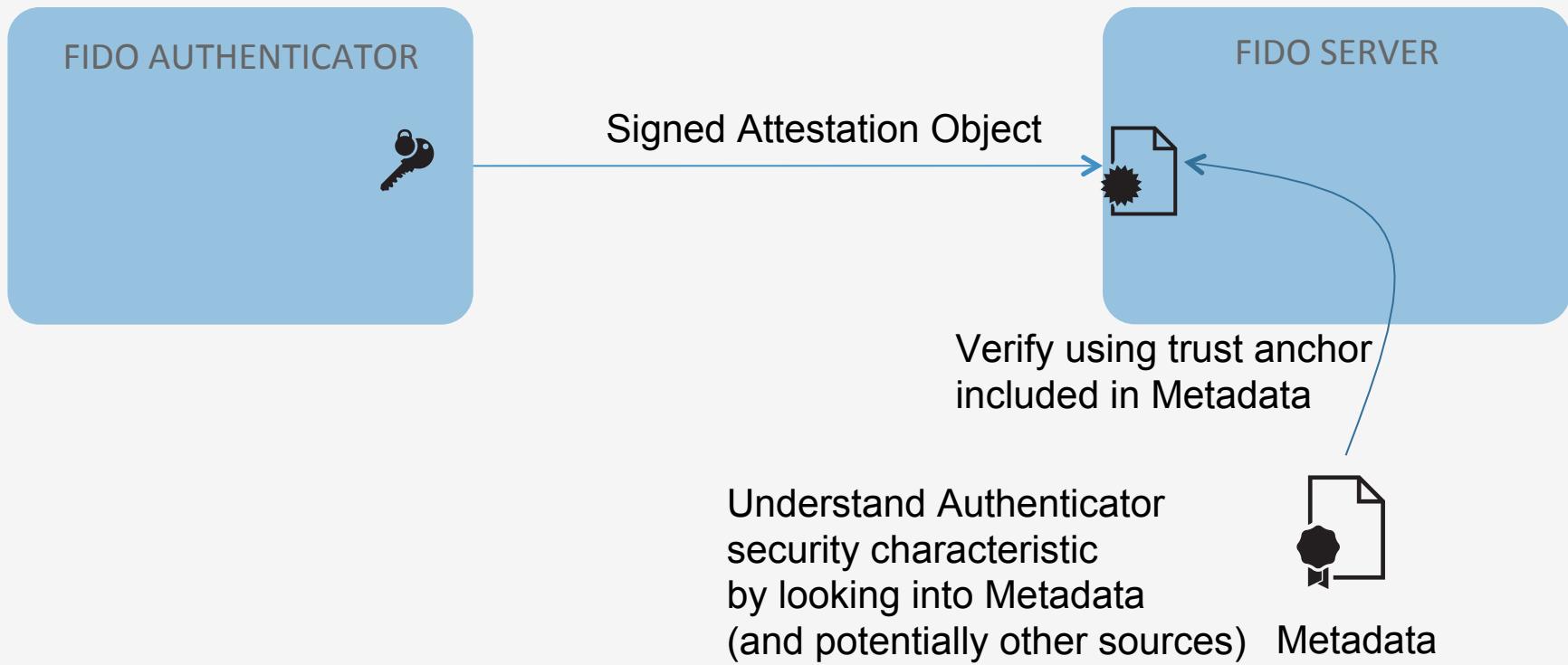
How does FIDO UAF work?



How does FIDO UAF work?

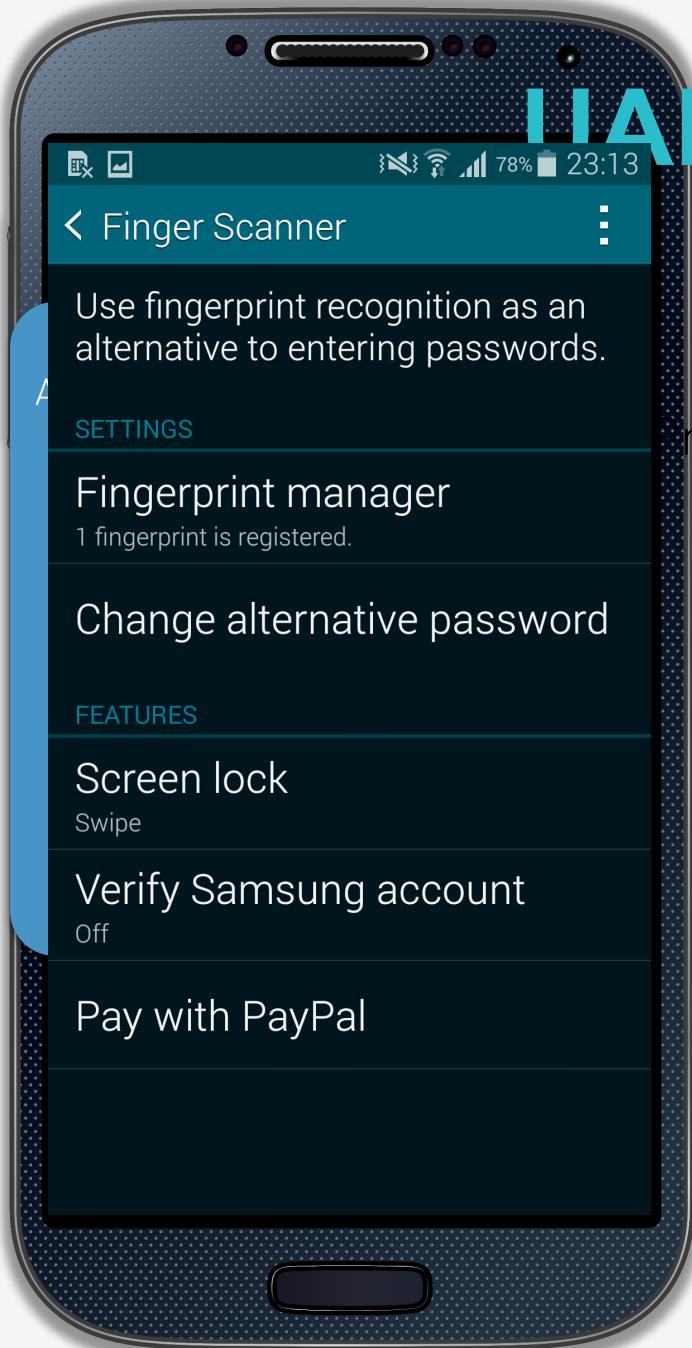


Attestation & Metadata



UAF Registration

Relying Party

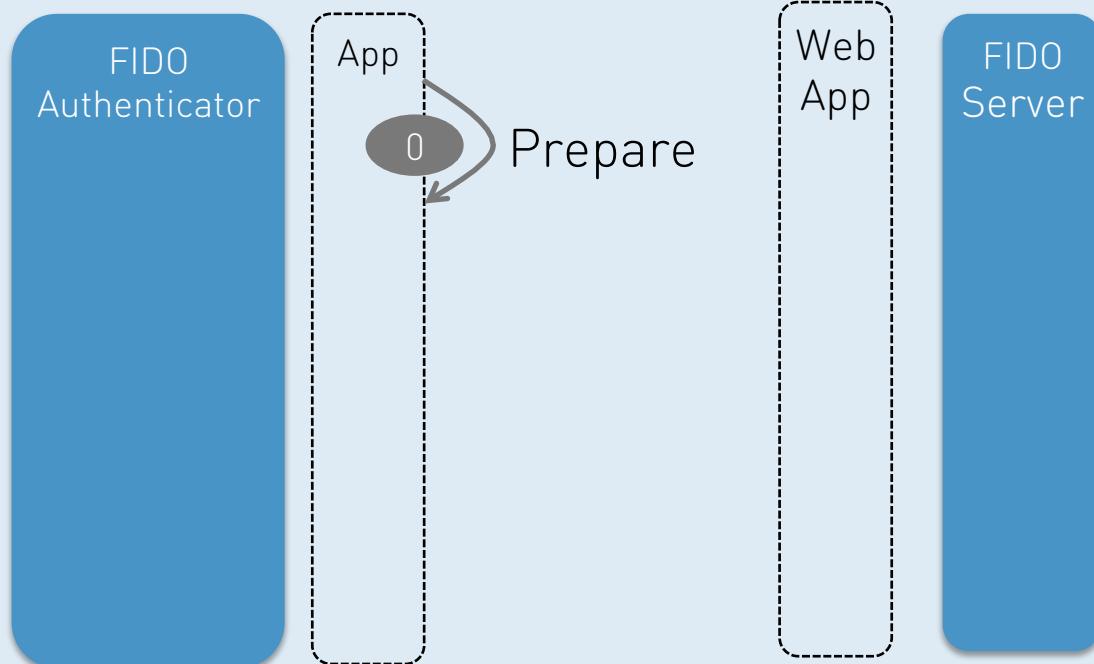
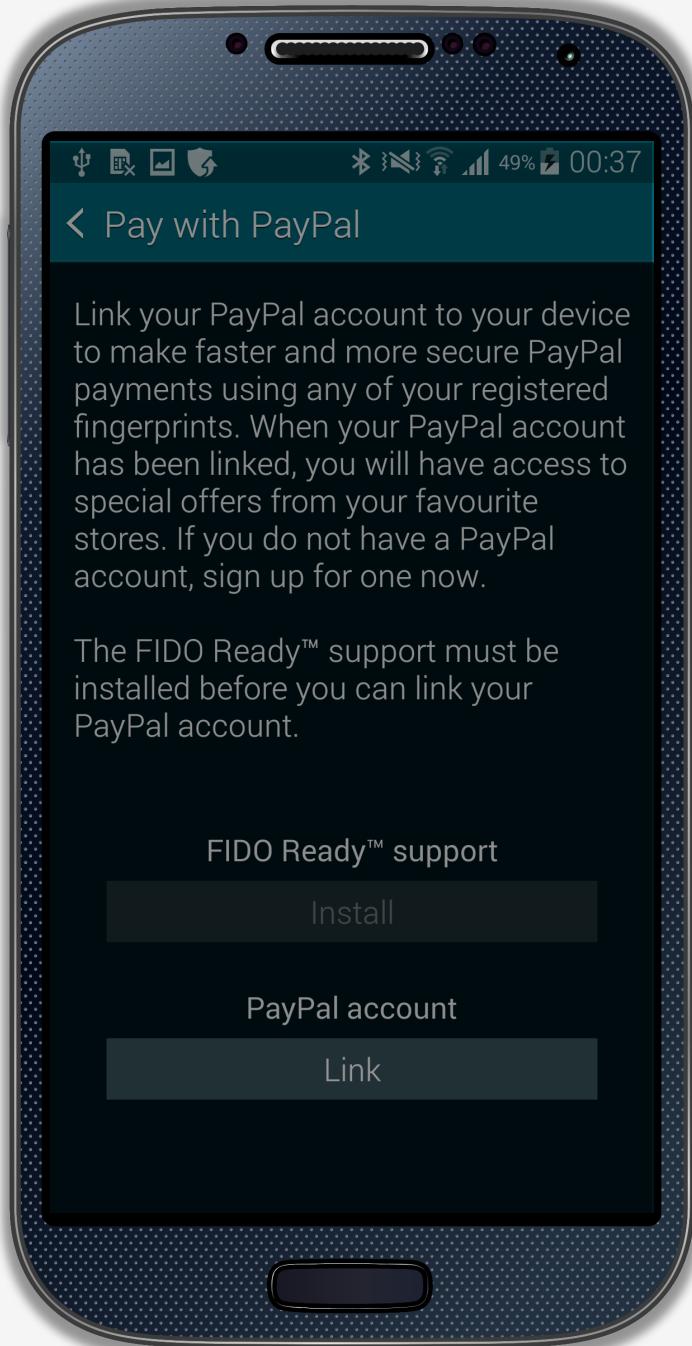


repare

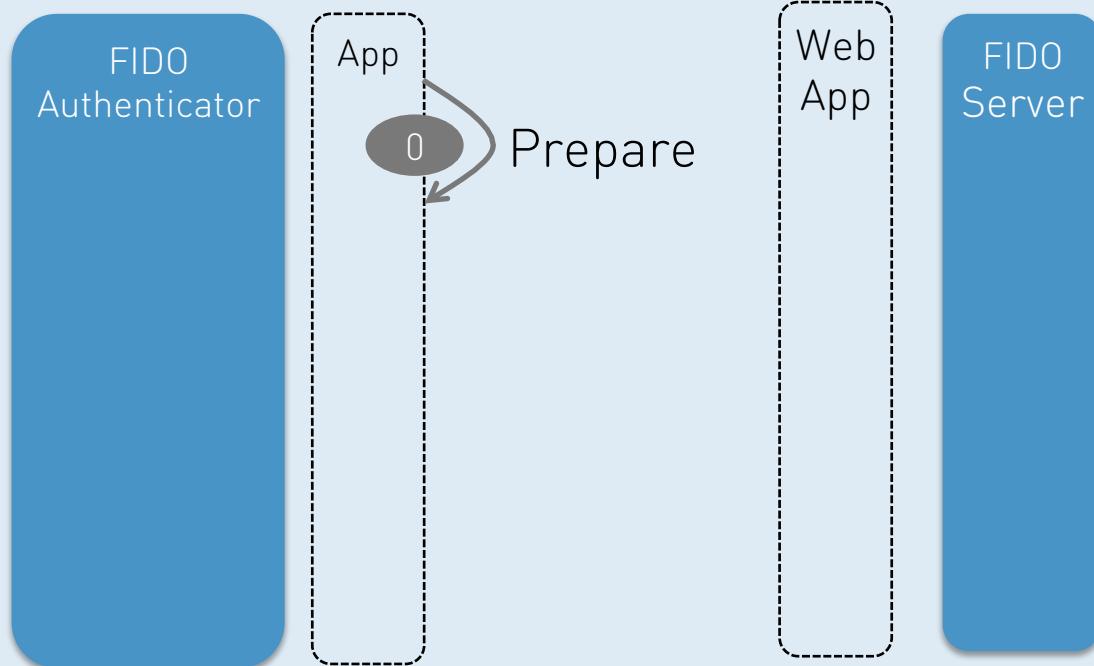
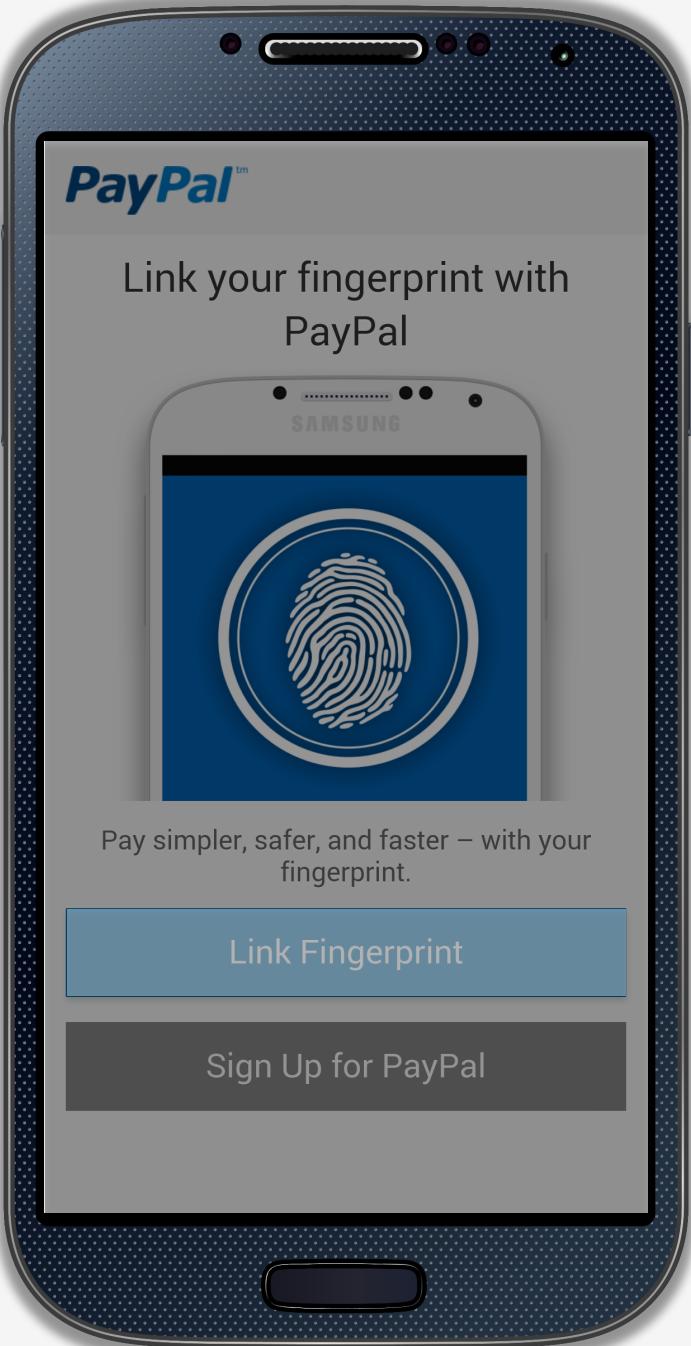
Web
App

FIDO
Server

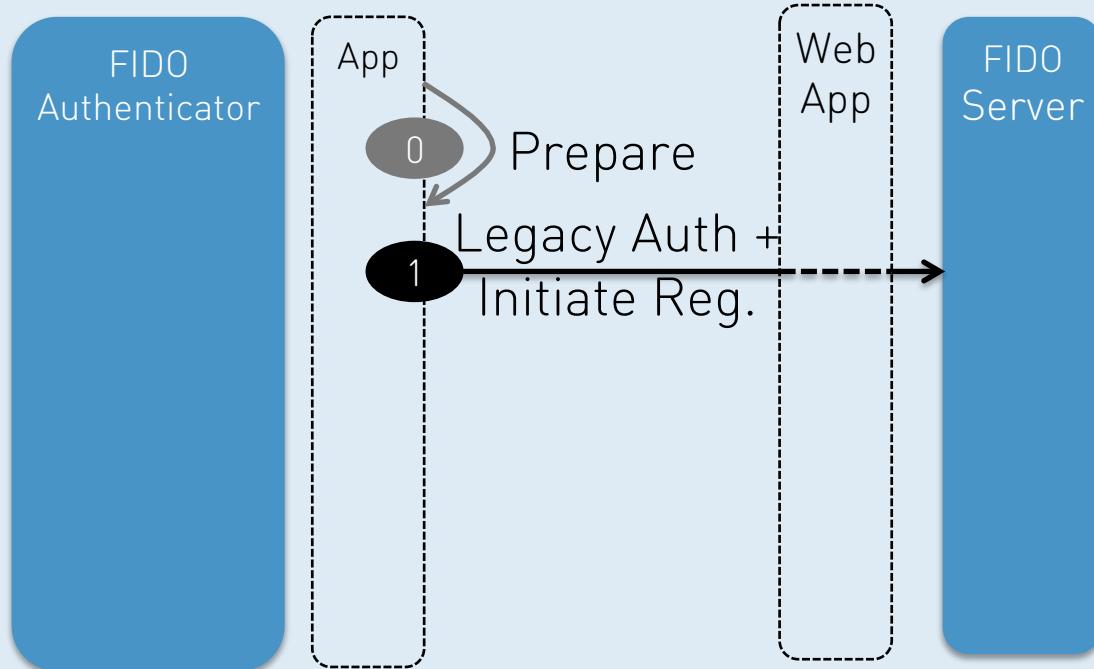
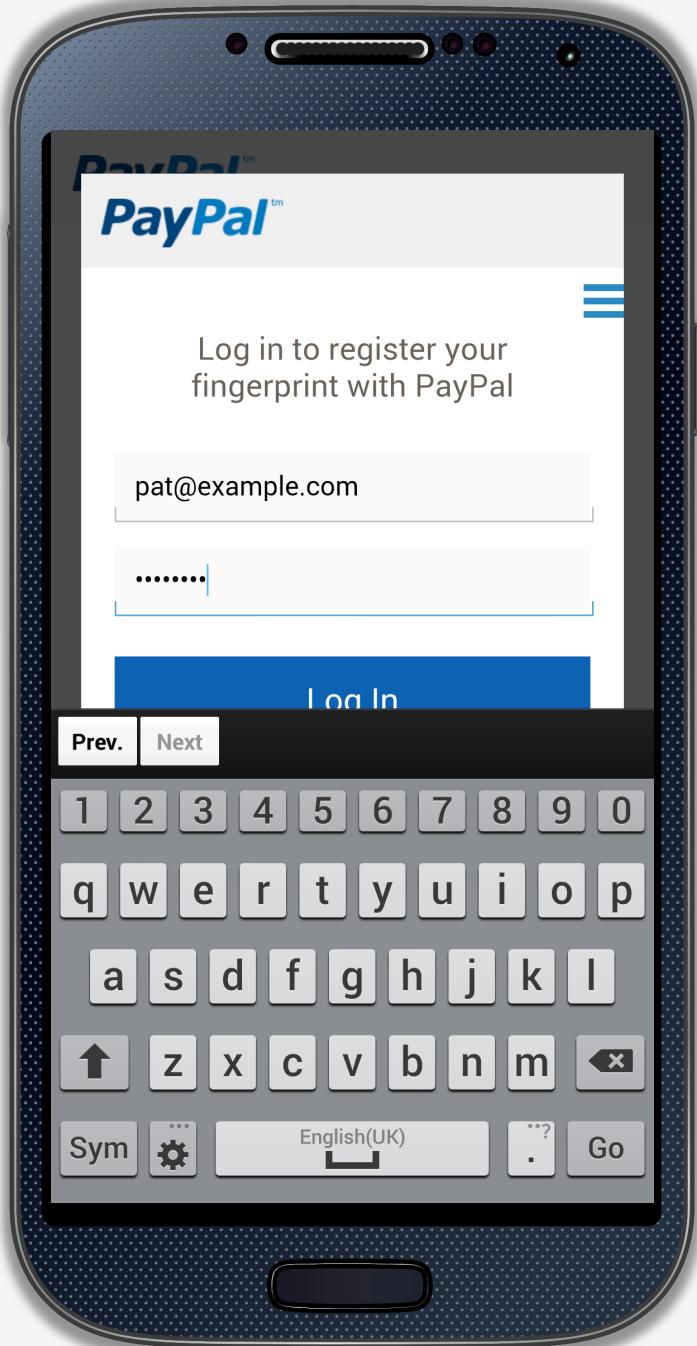
UAF Registration



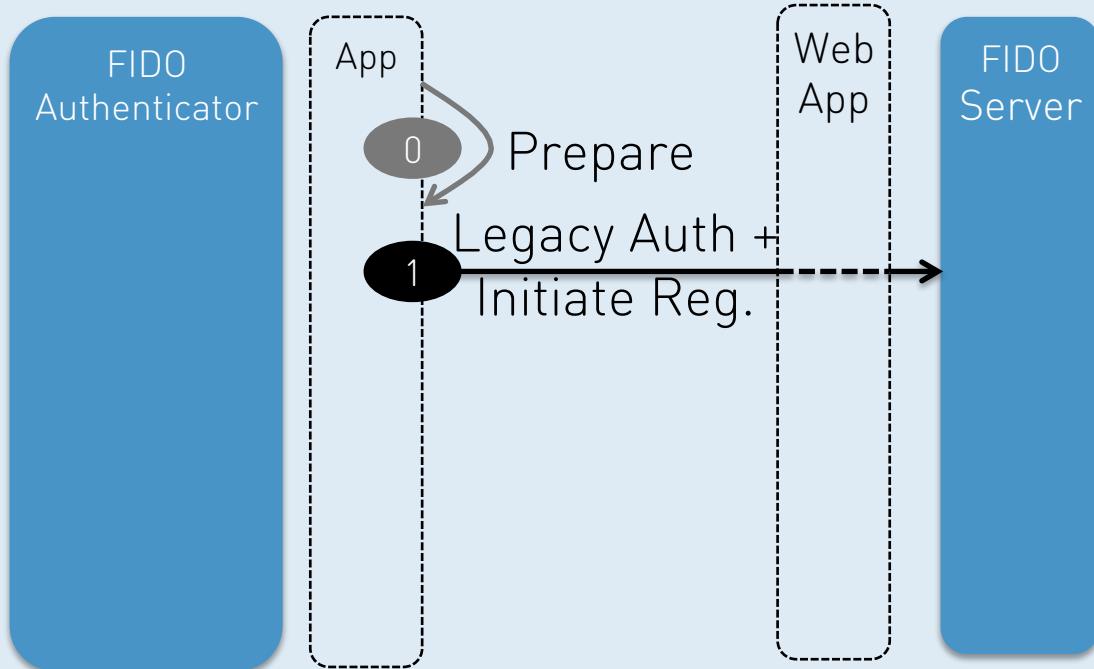
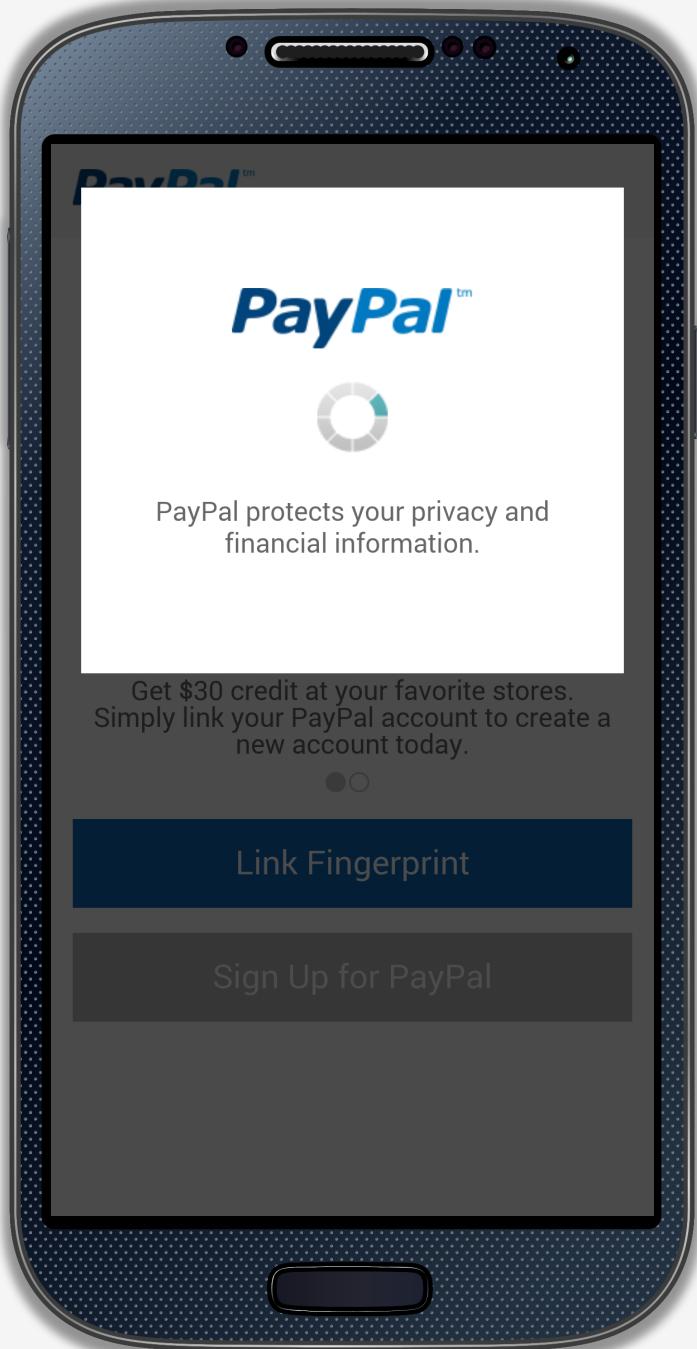
UAF Registration



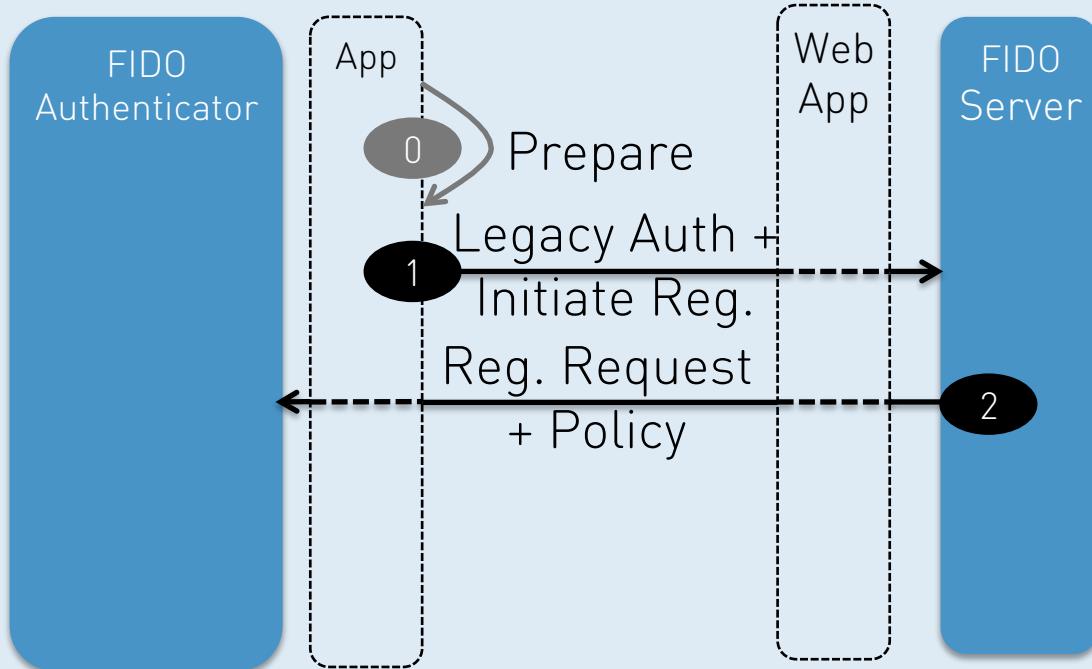
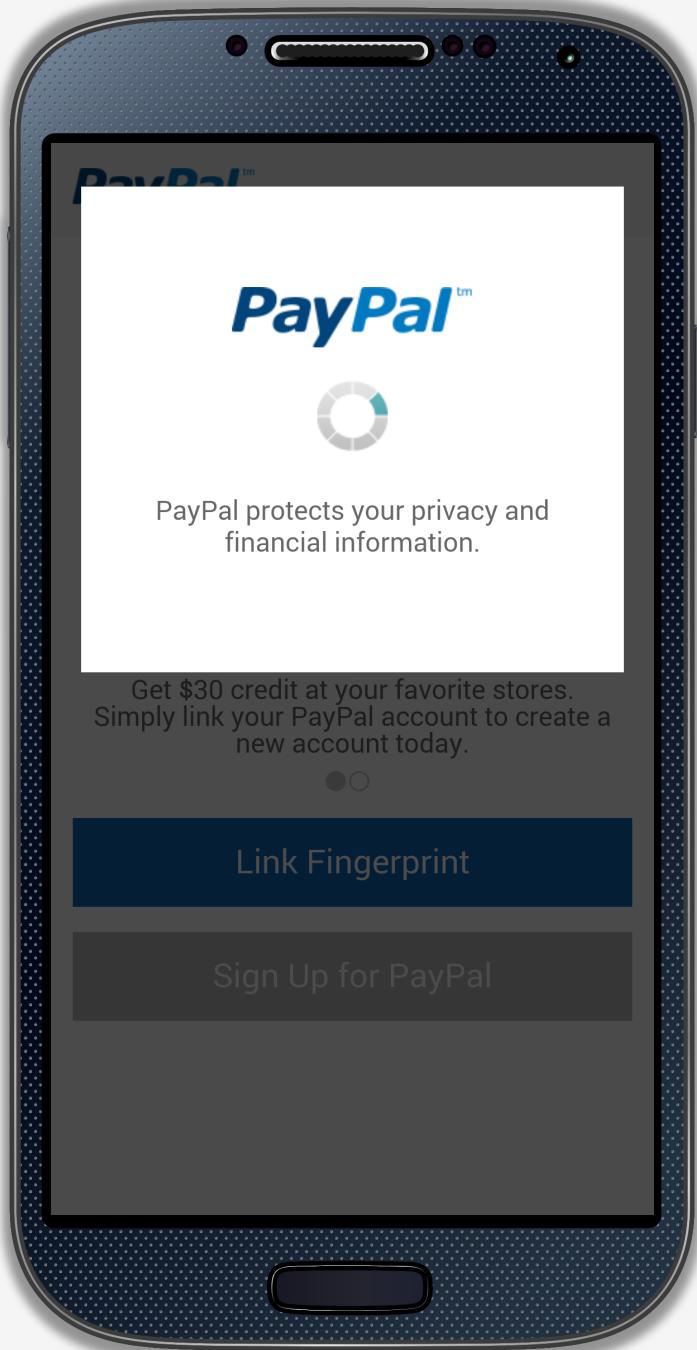
UAF Registration



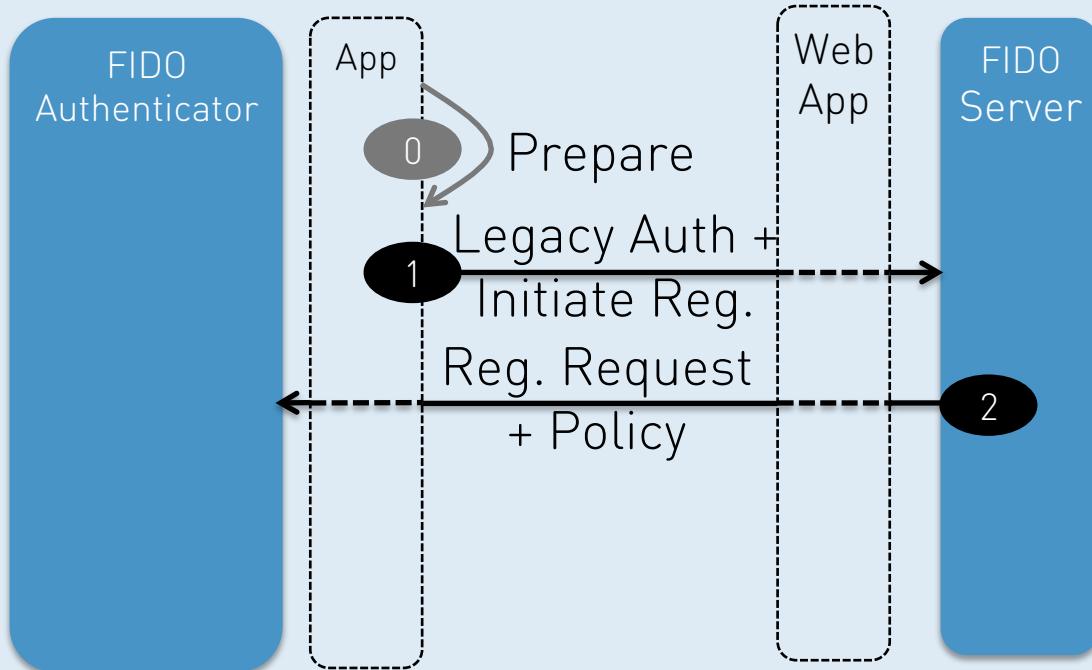
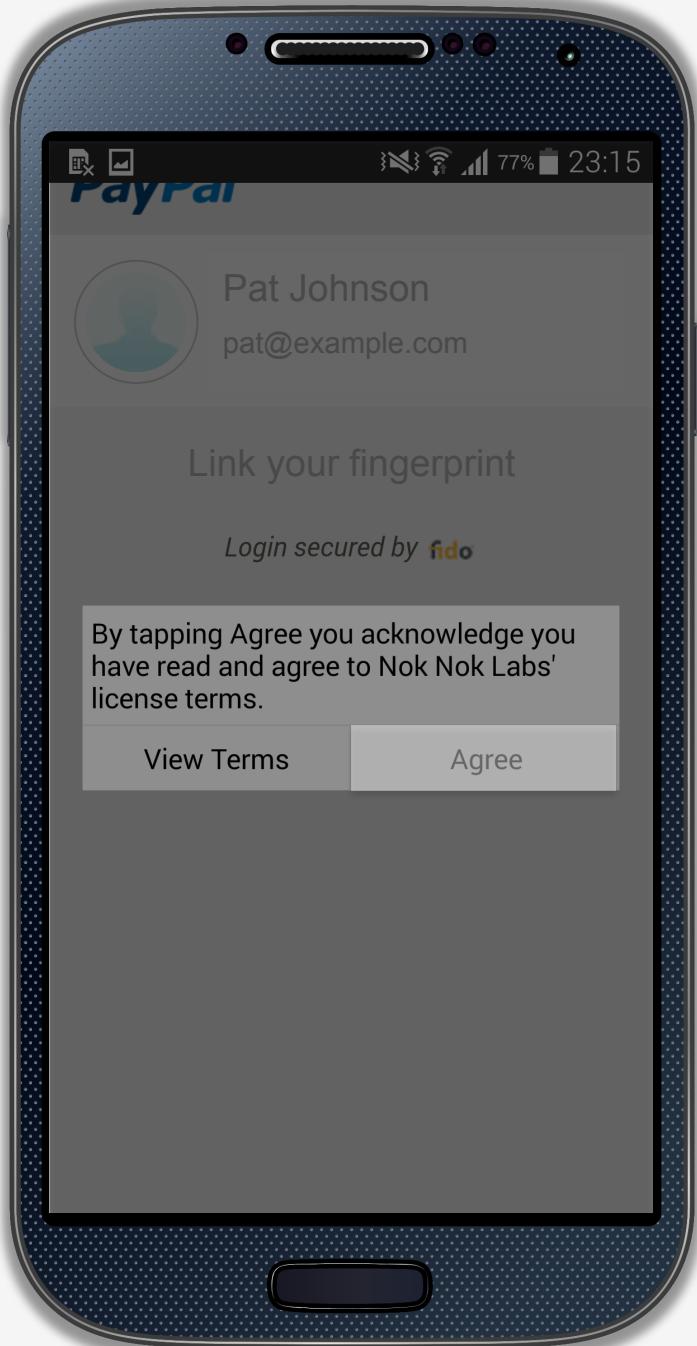
UAF Registration



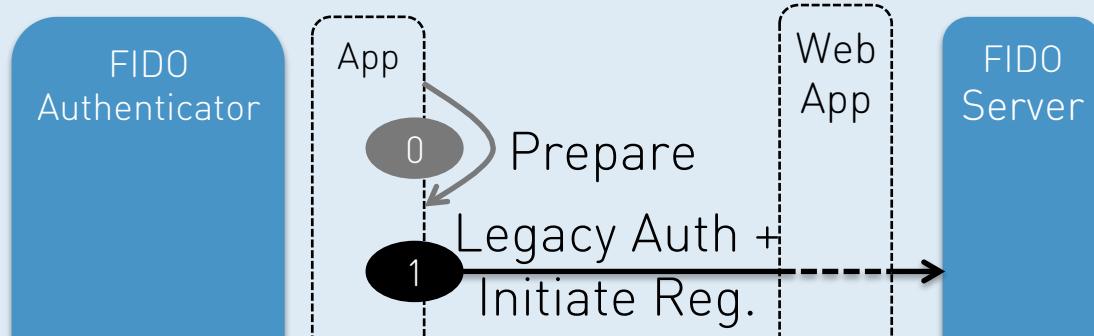
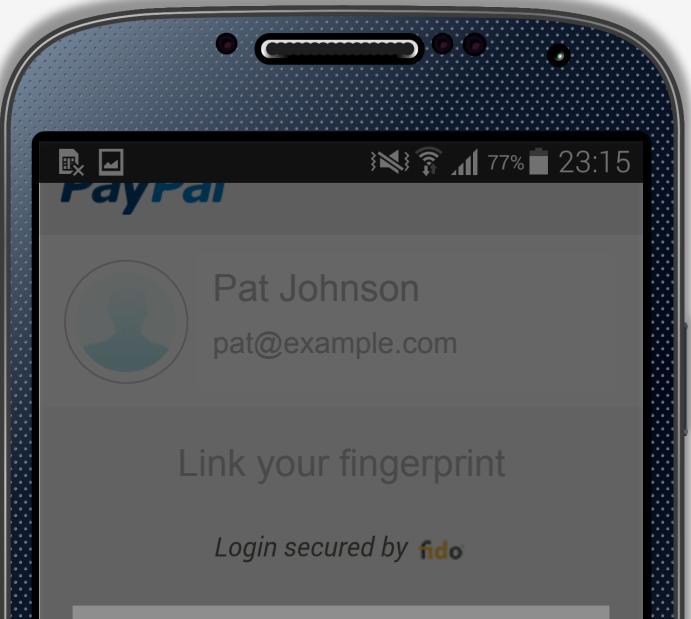
UAF Registration



UAF Registration

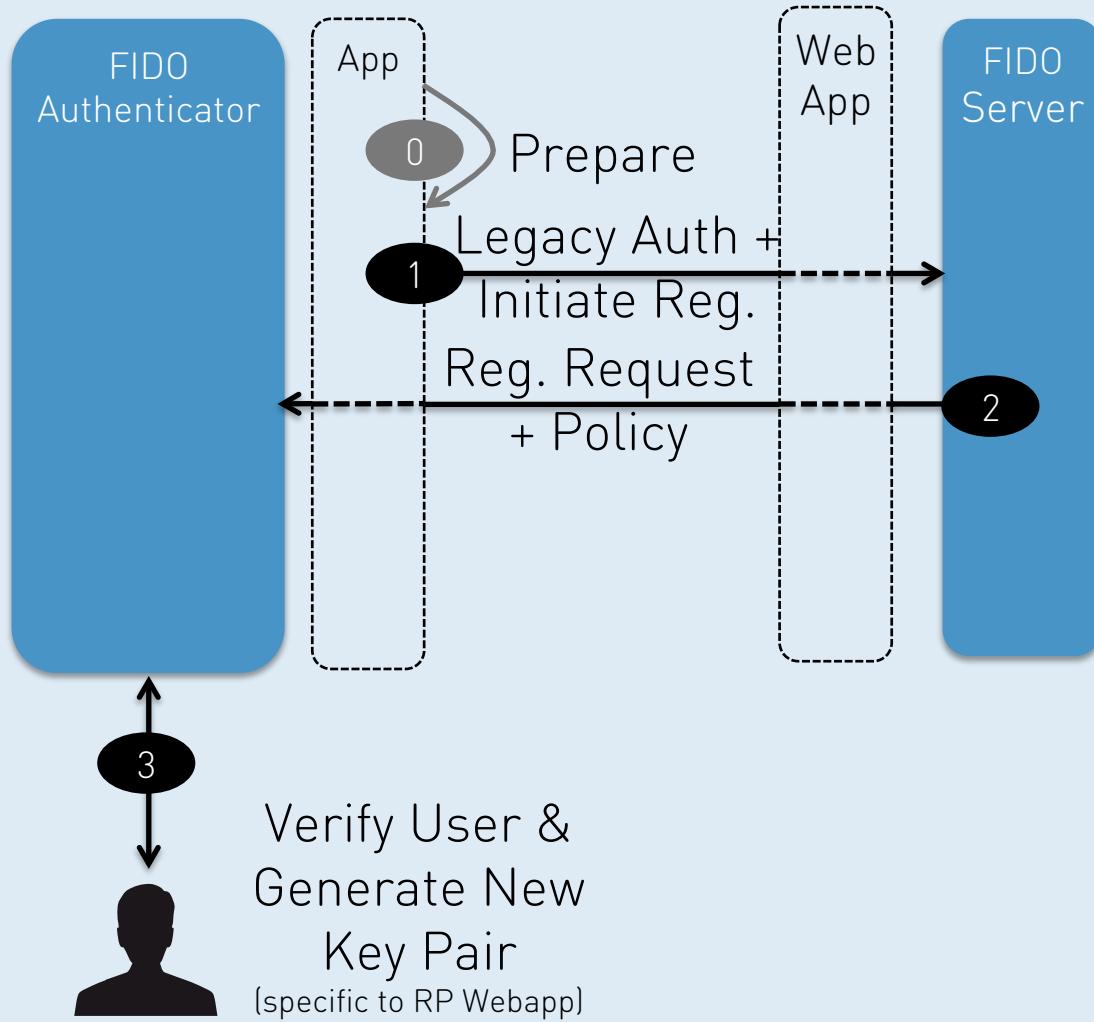
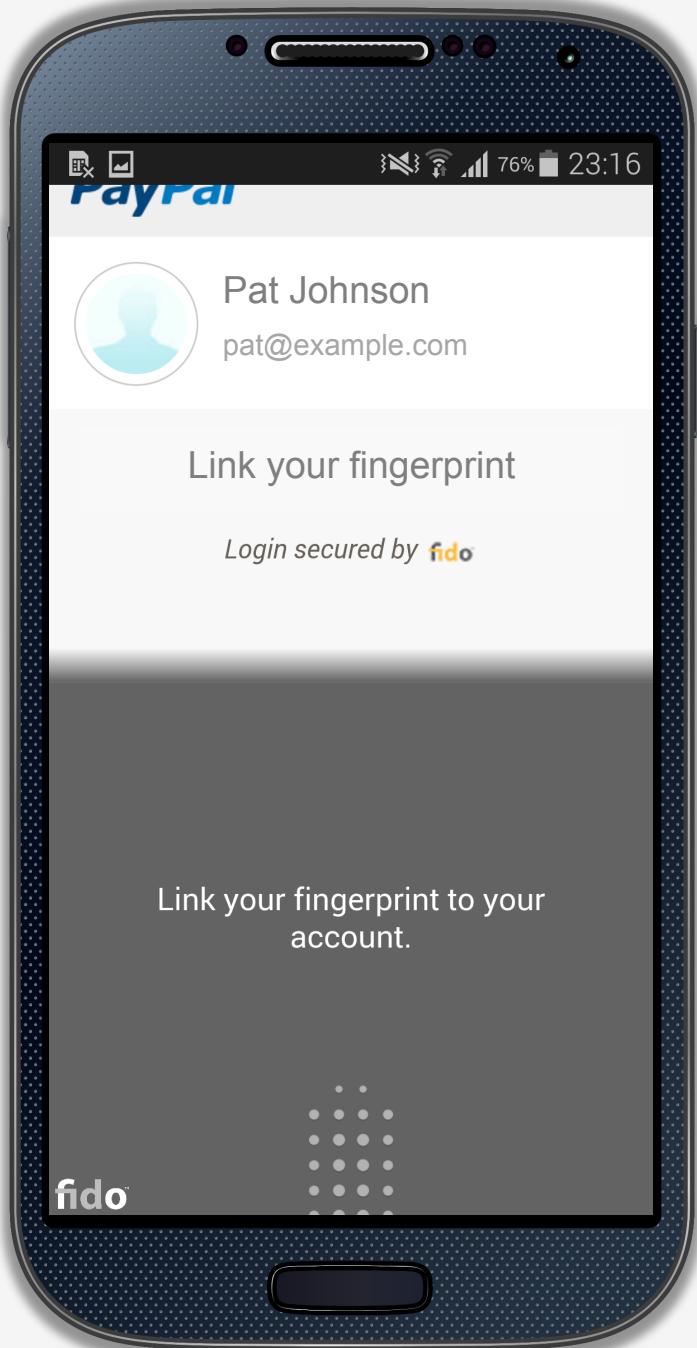


UAF Registration

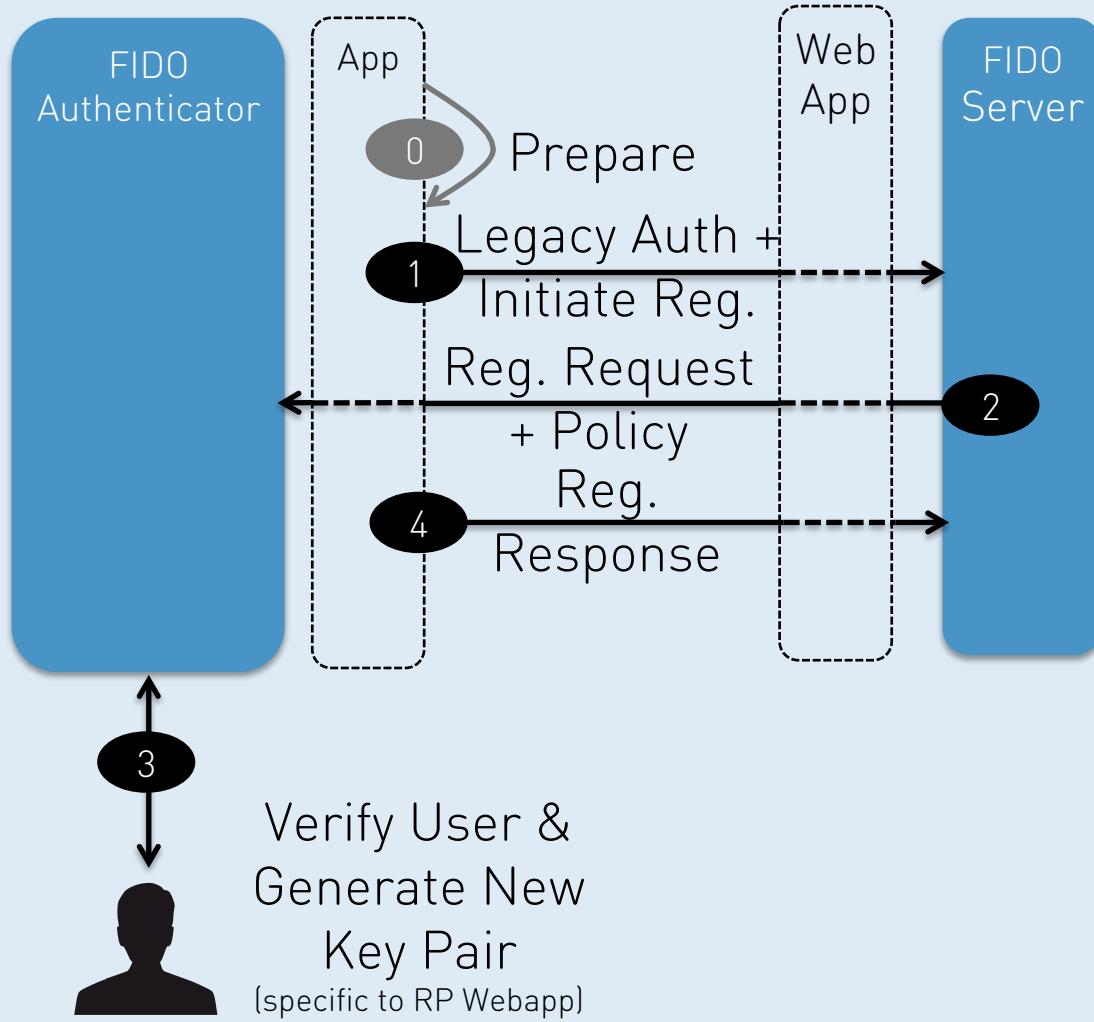
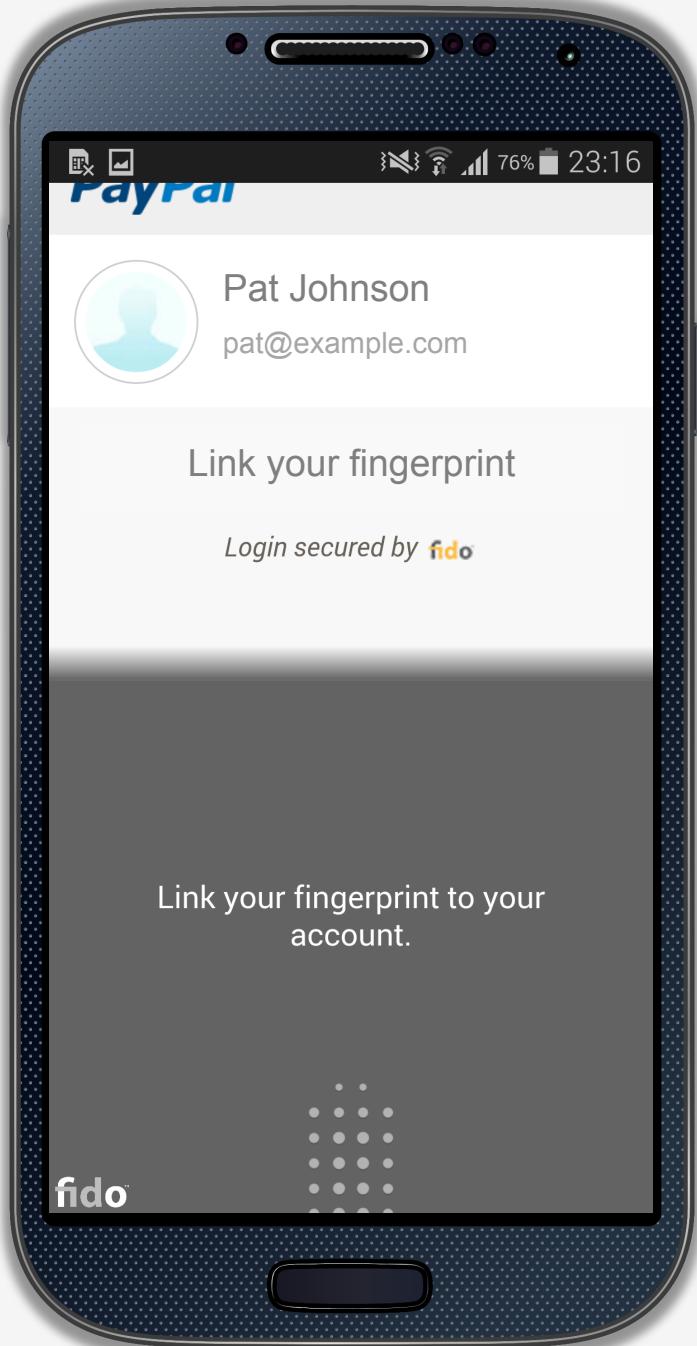


```
[{
    "header": { "op": "Reg", "upv": "1.0", "appID": "https://mycorp.com/fido" },
    "challenge": "qwudh827hddbawd8qbdqj3bduq3duq56t324zwasdq4wrt",
    "username": "banking_personal",
    "policy": {
        "accepted": [ [ {
            "authenticationFactor": 000000000000001ff,
            "keyProtection":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

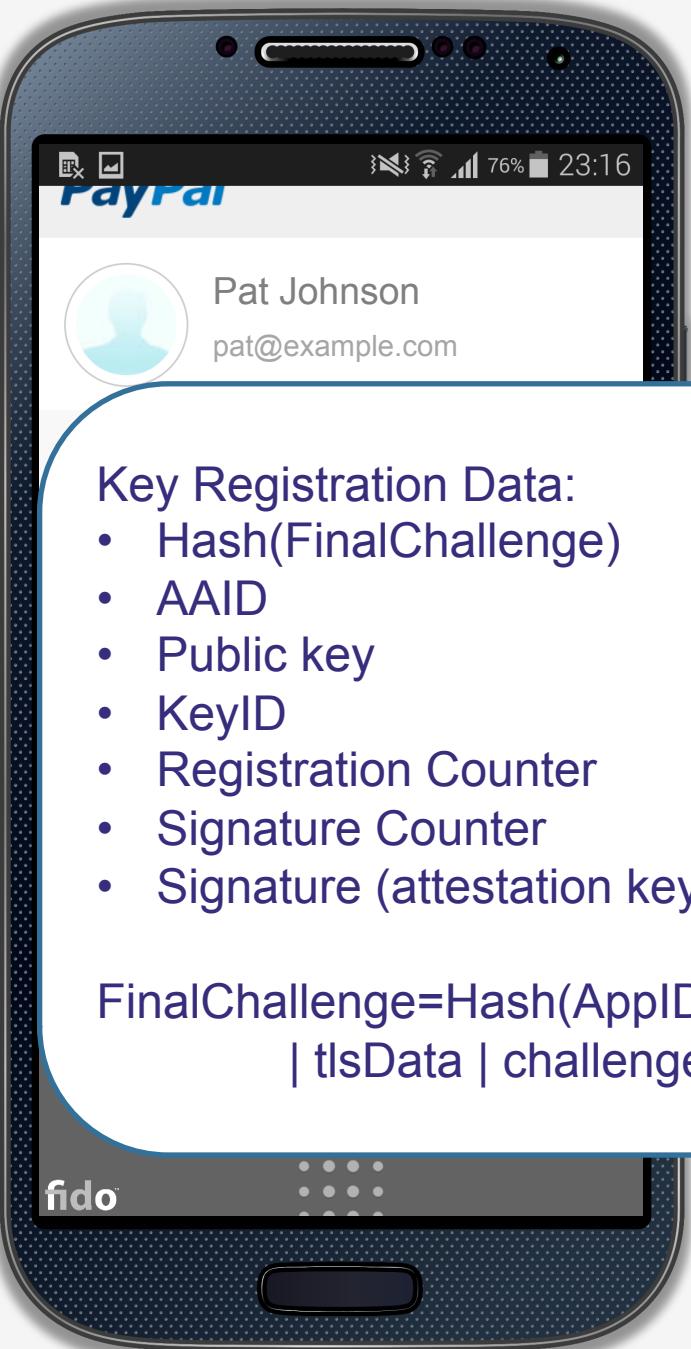
UAF Registration



UAF Registration



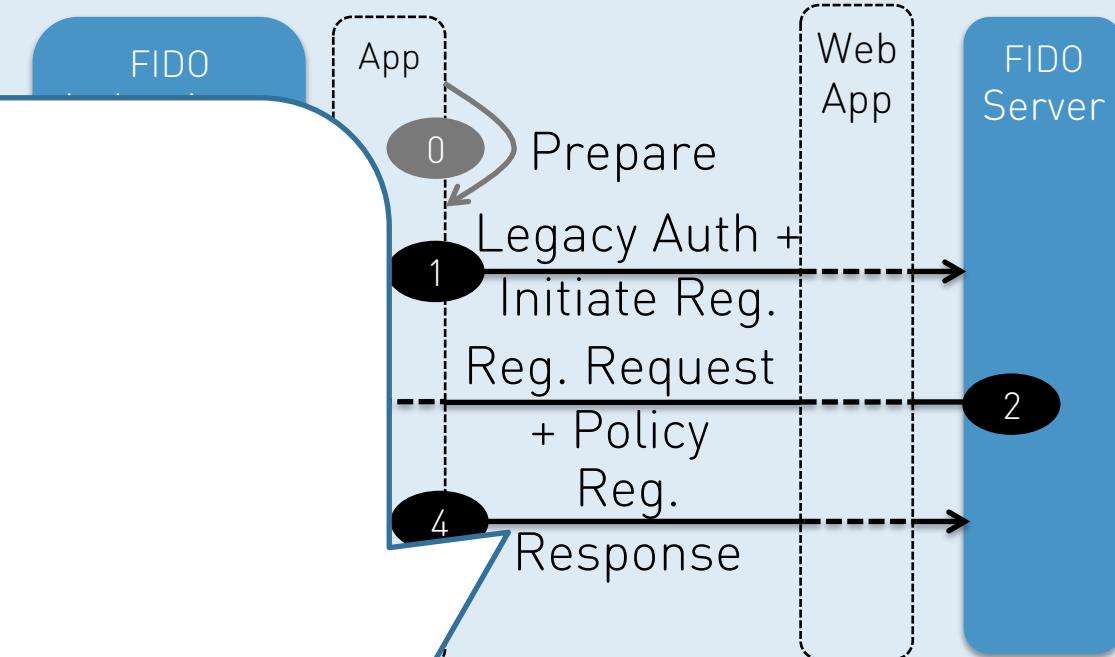
UAF Registration



Key Registration Data:

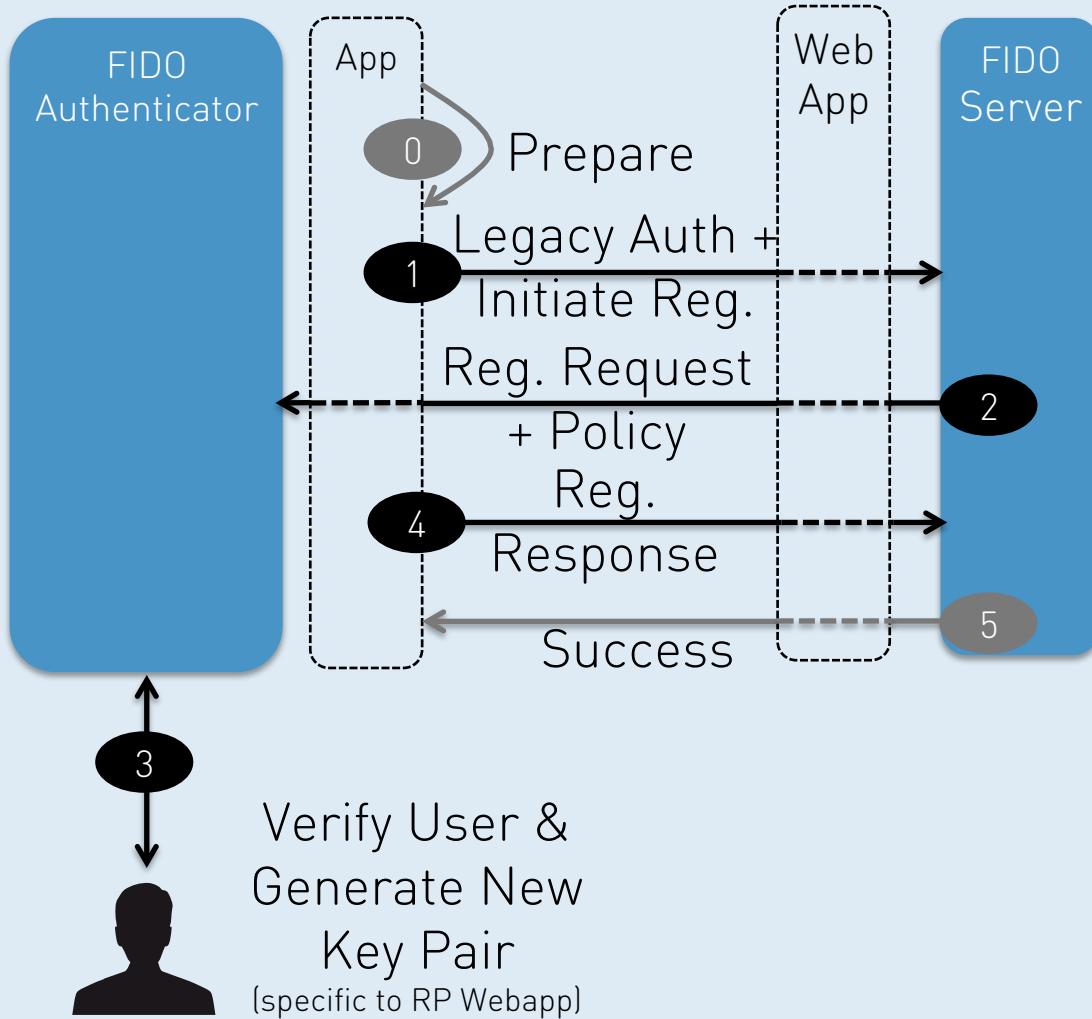
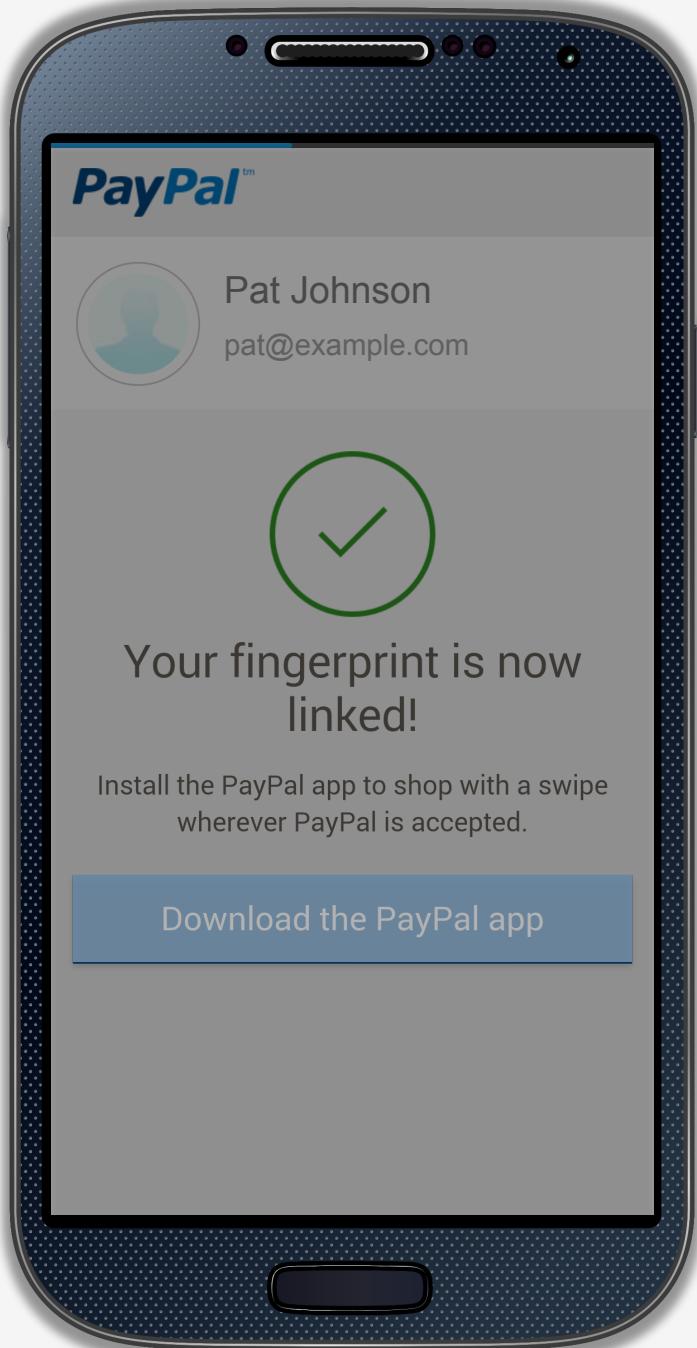
- Hash(FinalChallenge)
- AAID
- Public key
- KeyID
- Registration Counter
- Signature Counter
- Signature (attestation key)

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

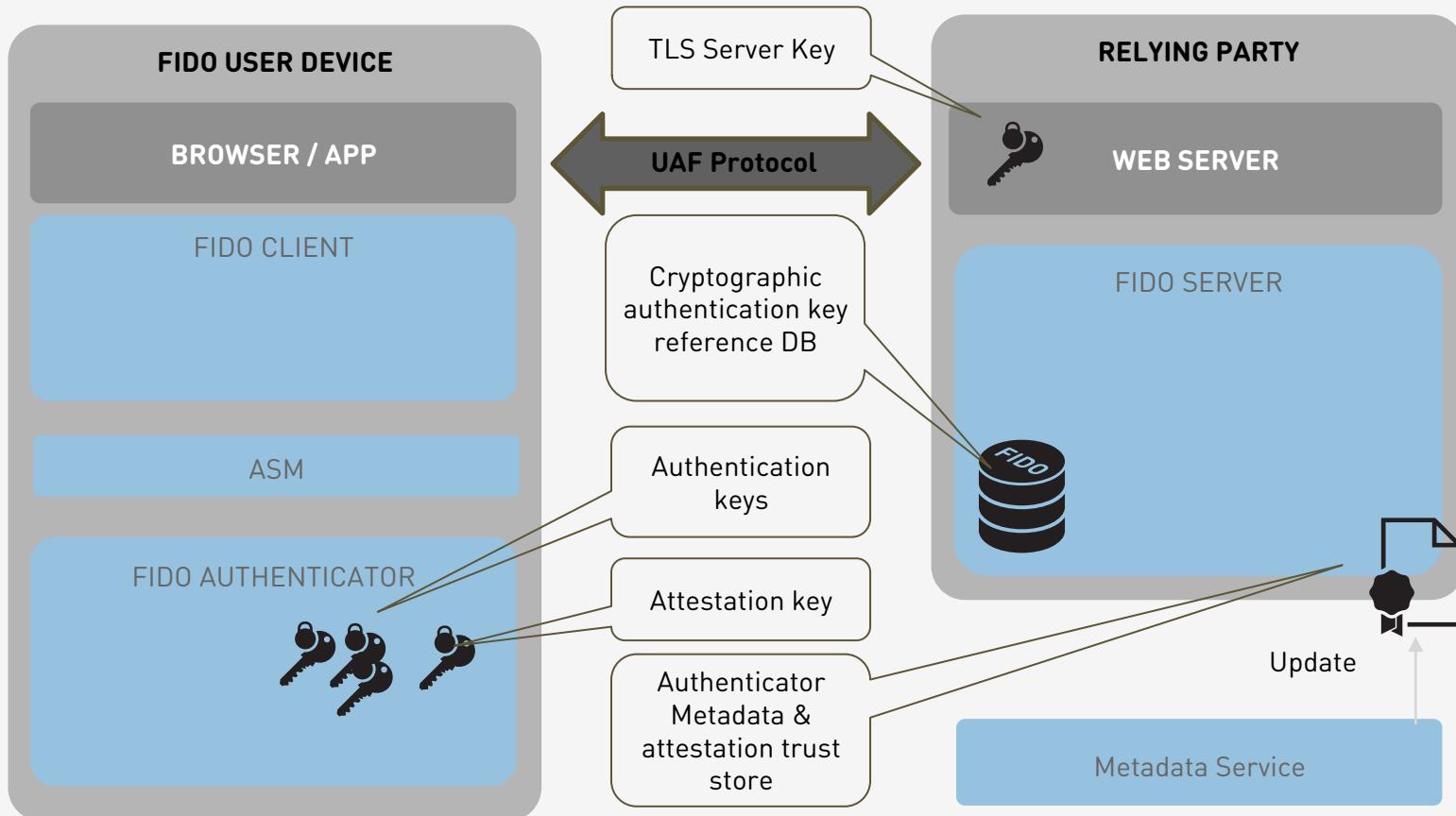


Verify User &
Generate New
Key Pair
(specific to RP Webapp)

UAF Registration



FIDO Building Blocks



AAID & Attestation

FIDO Authenticator

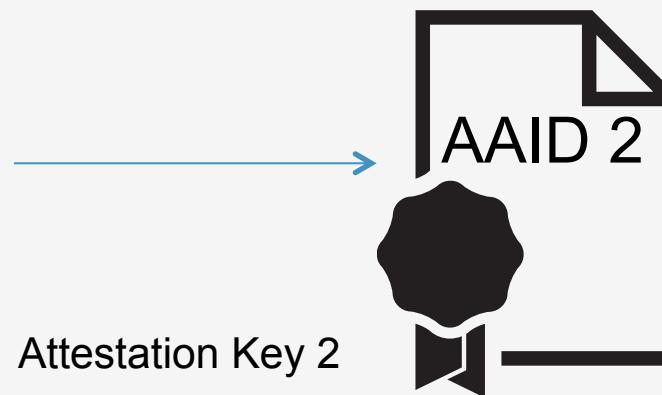
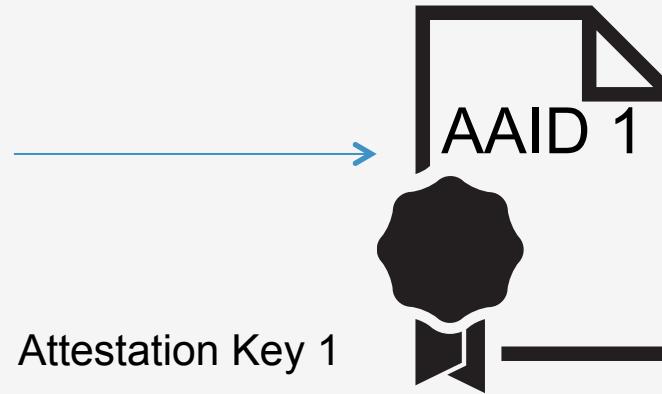
Using HW based crypto

Based on FP Sensor X

FIDO Authenticator

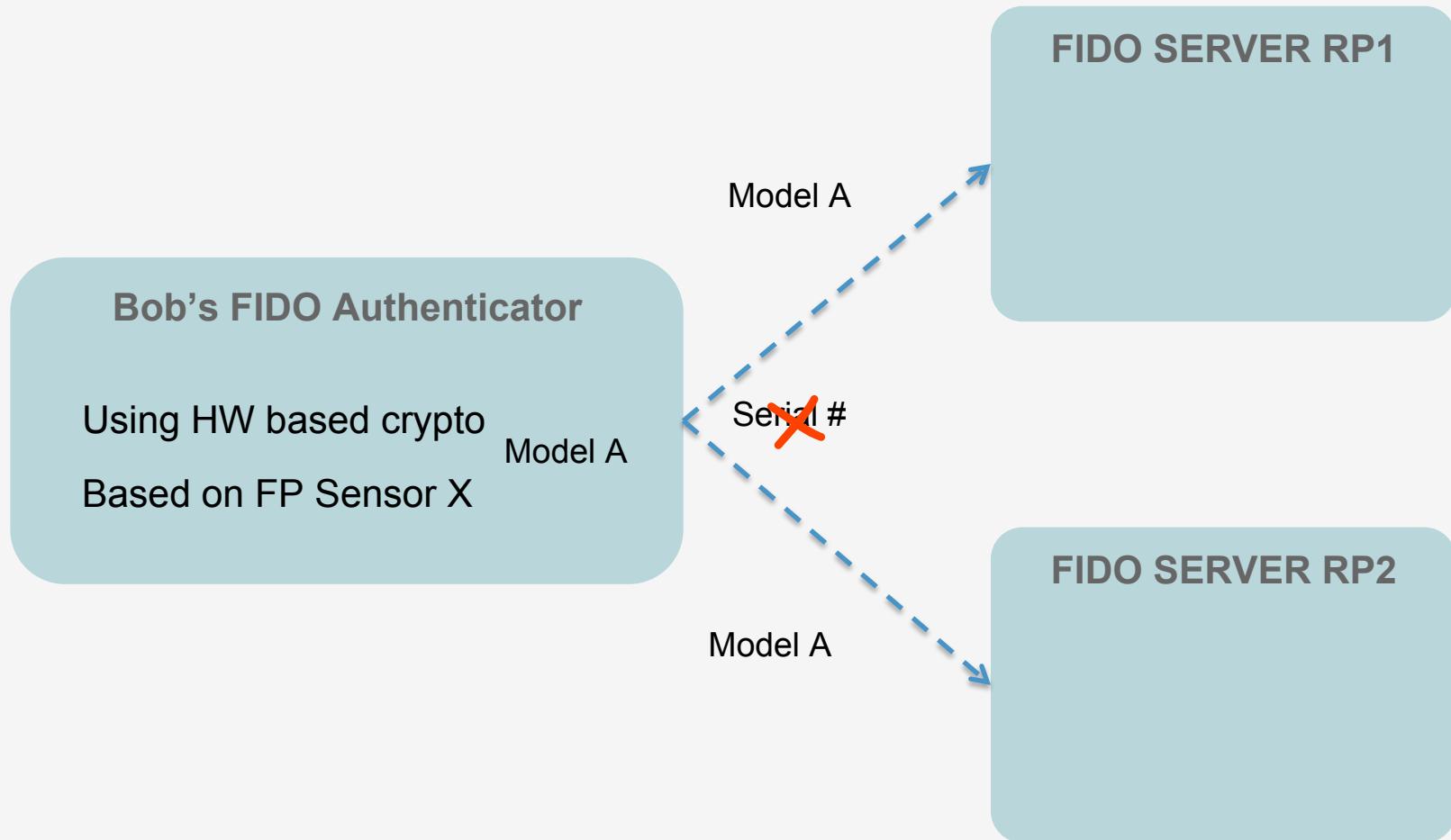
Pure SW based implementation

Based on Face Recognition alg. Y

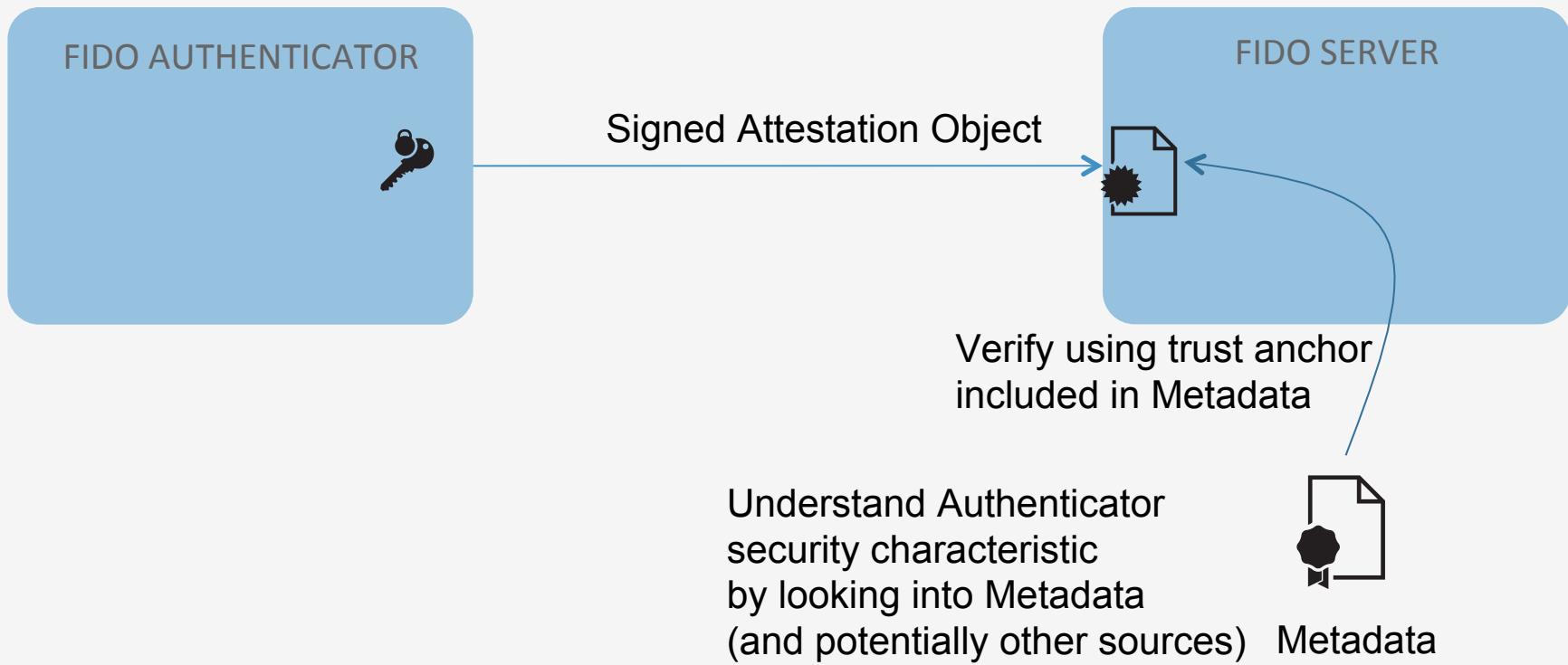


AAID: Authenticator Attestation ID (=model name)

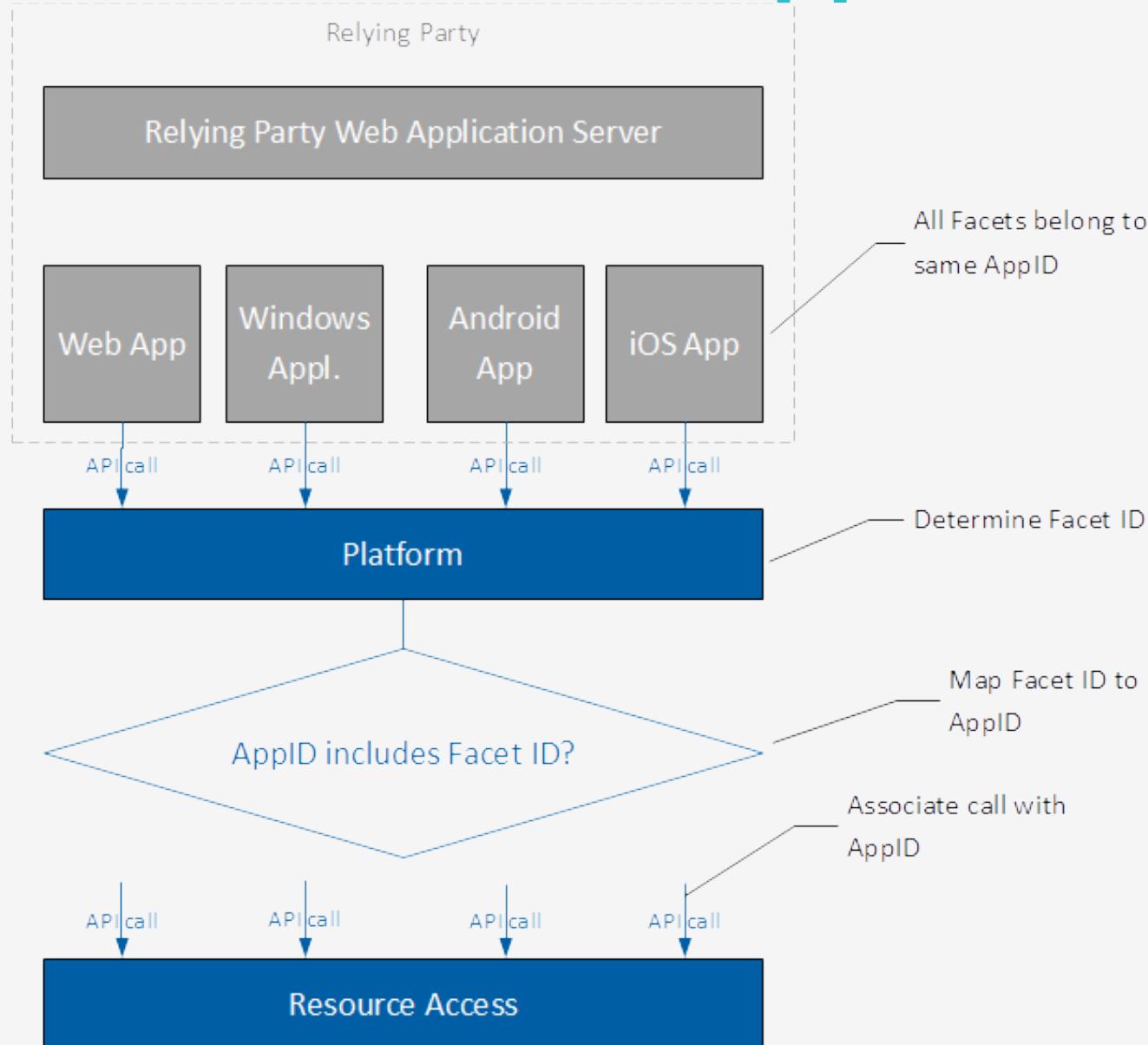
Privacy & Attestation

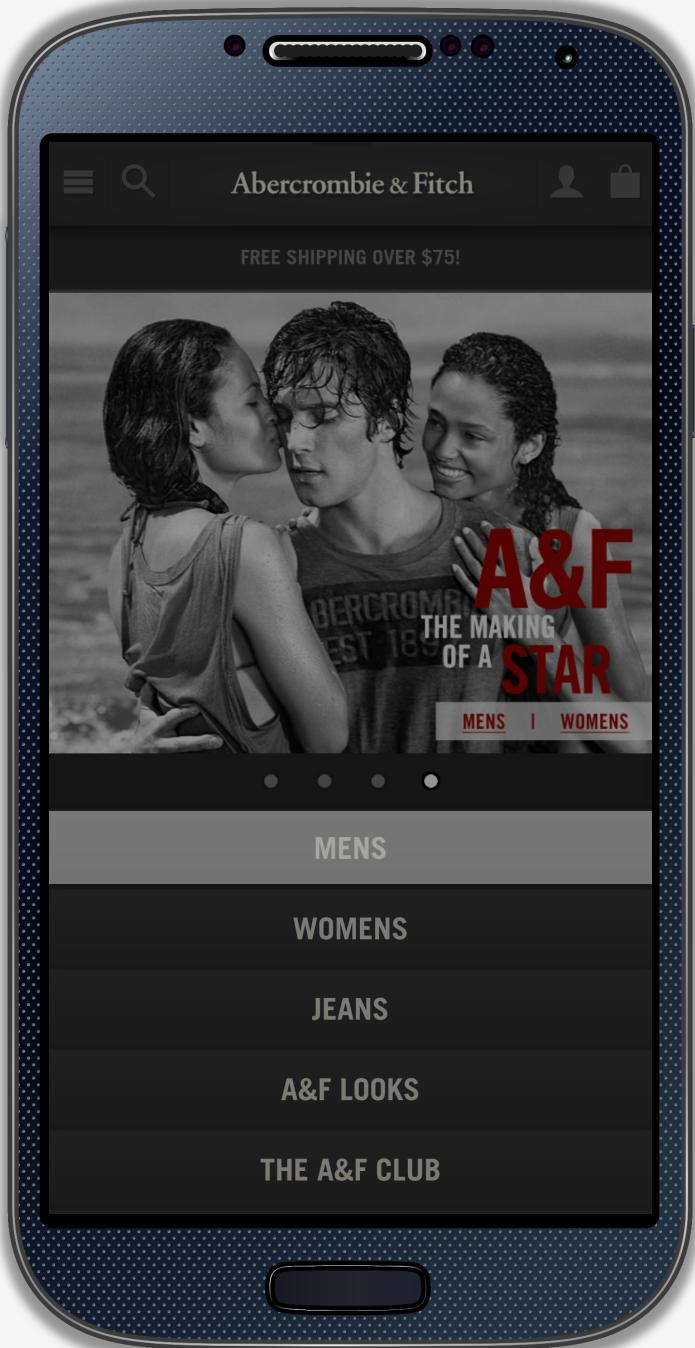


Attestation & Metadata

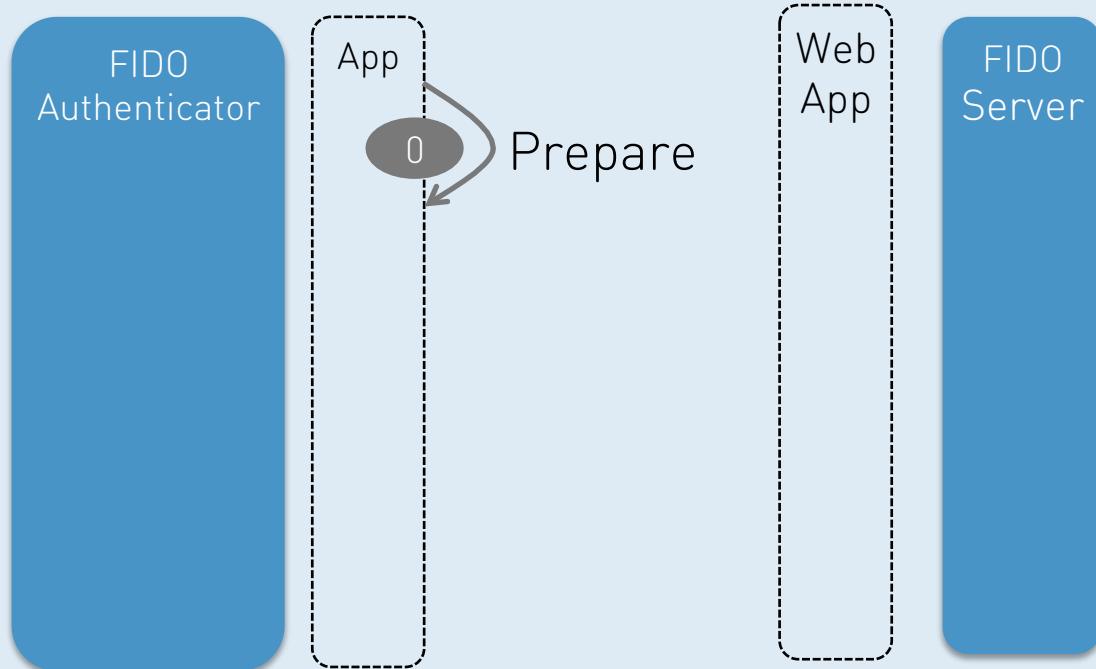


Facet ID / AppID

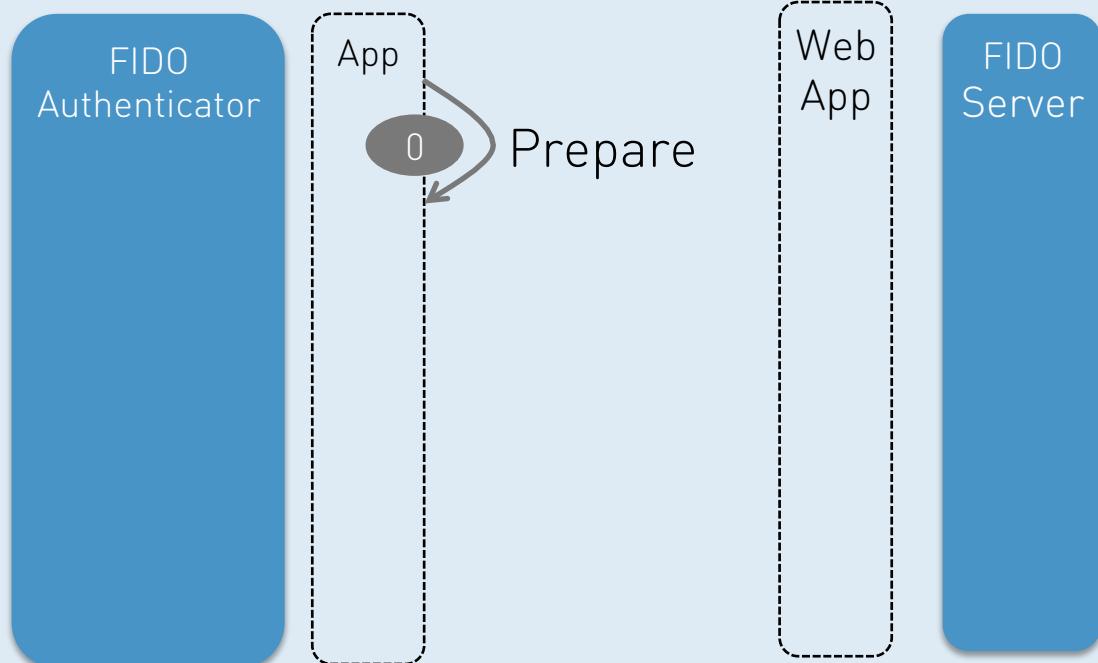
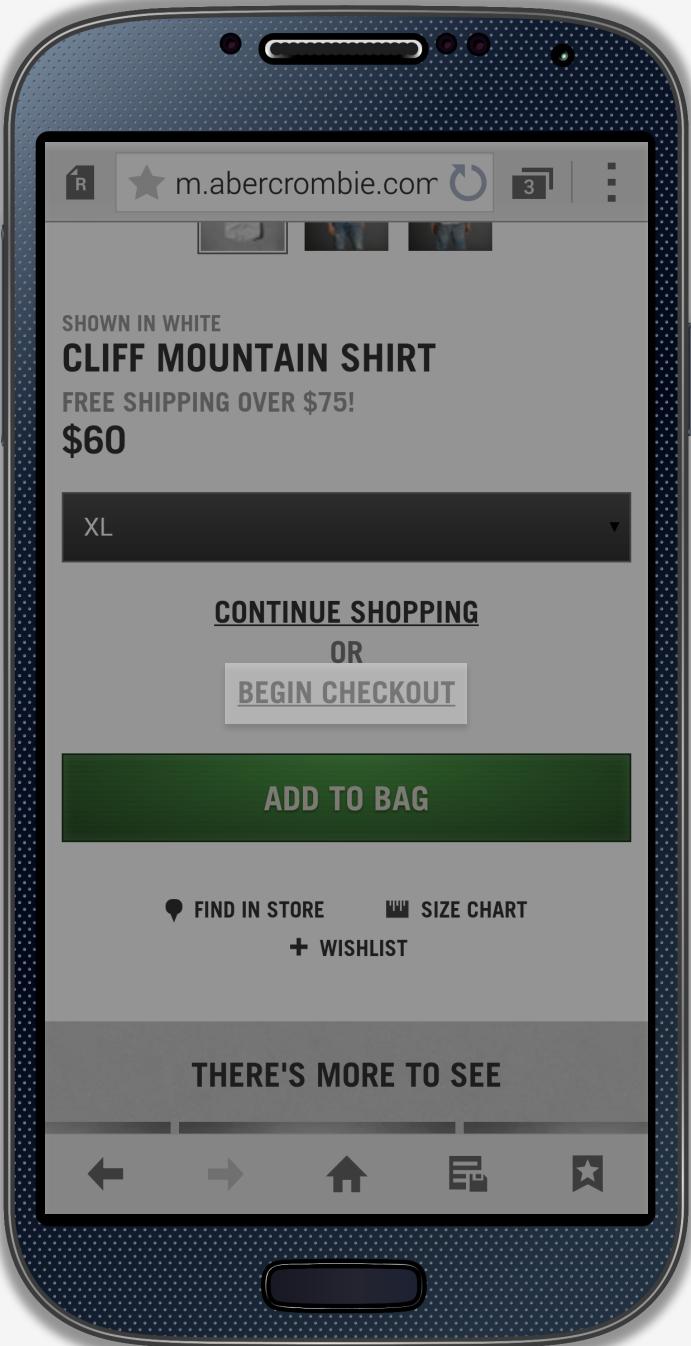




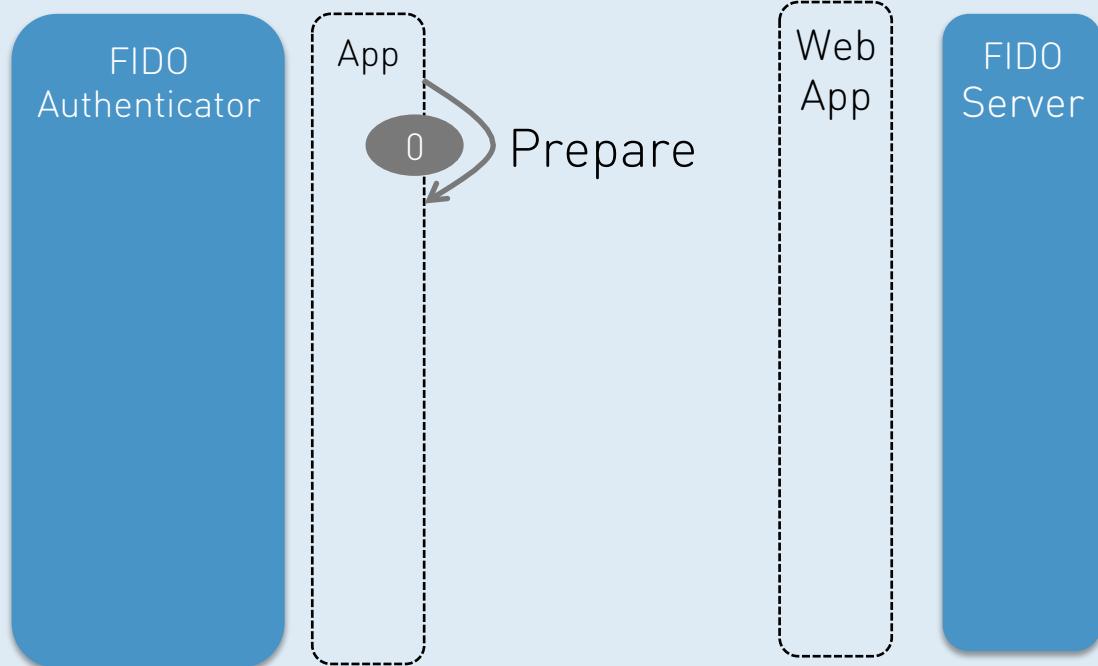
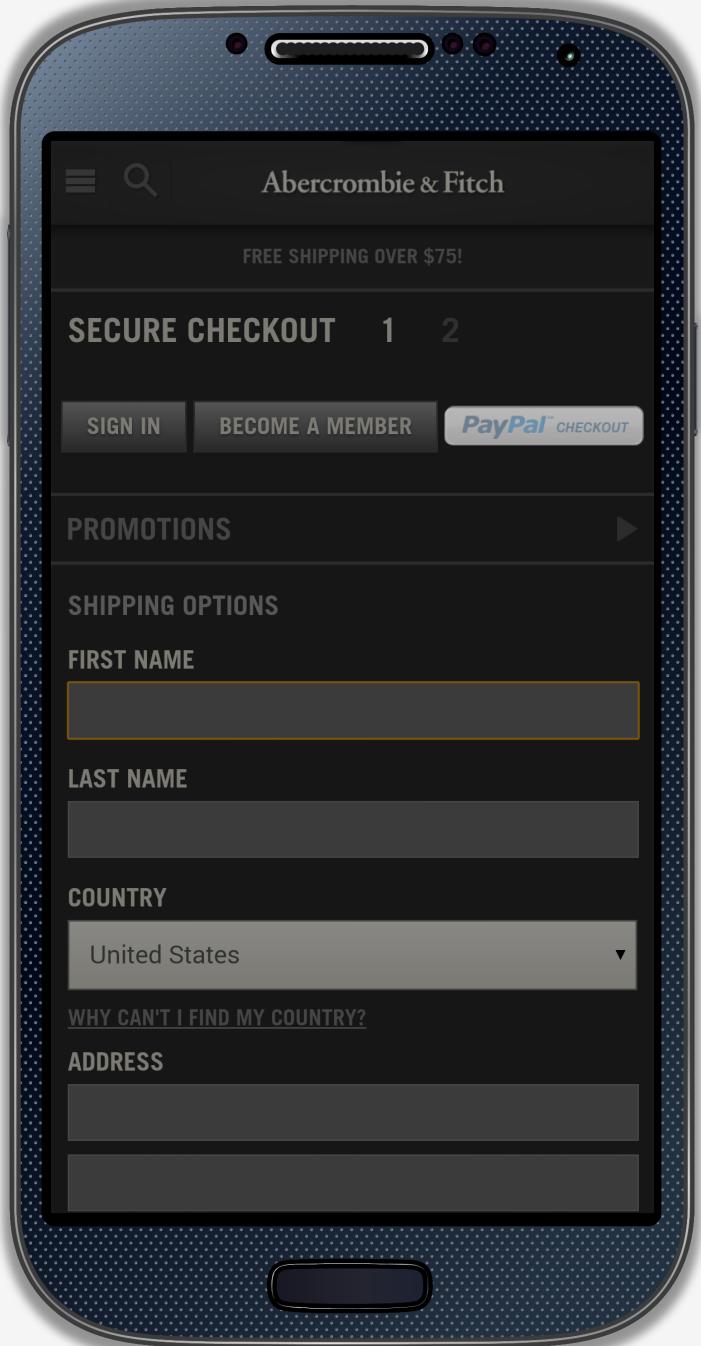
UAF Authentication



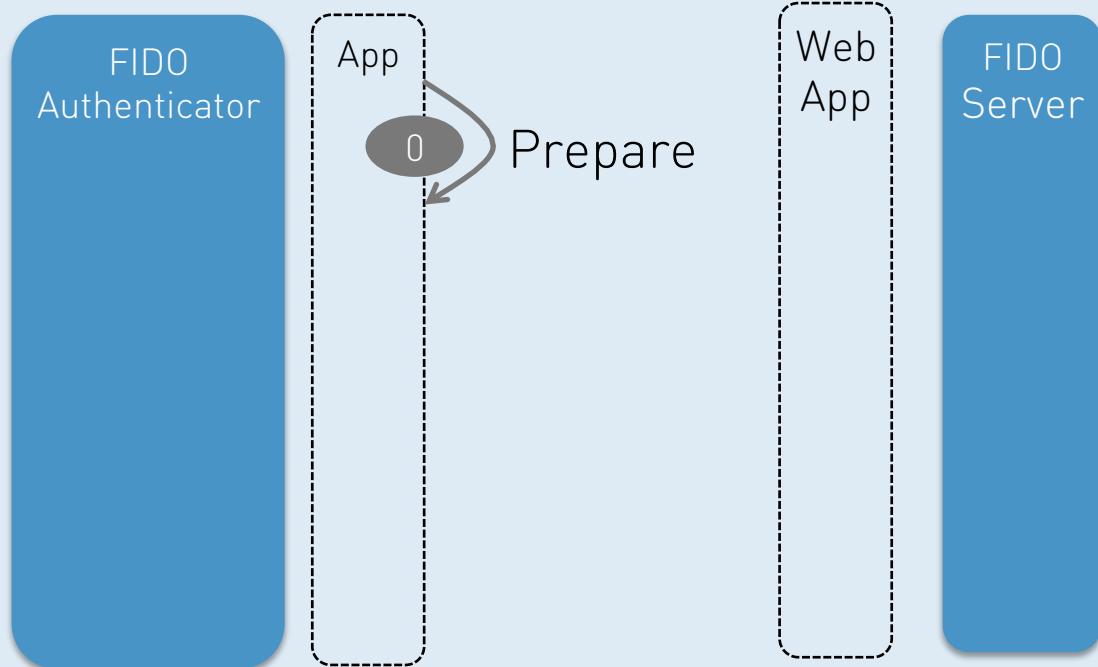
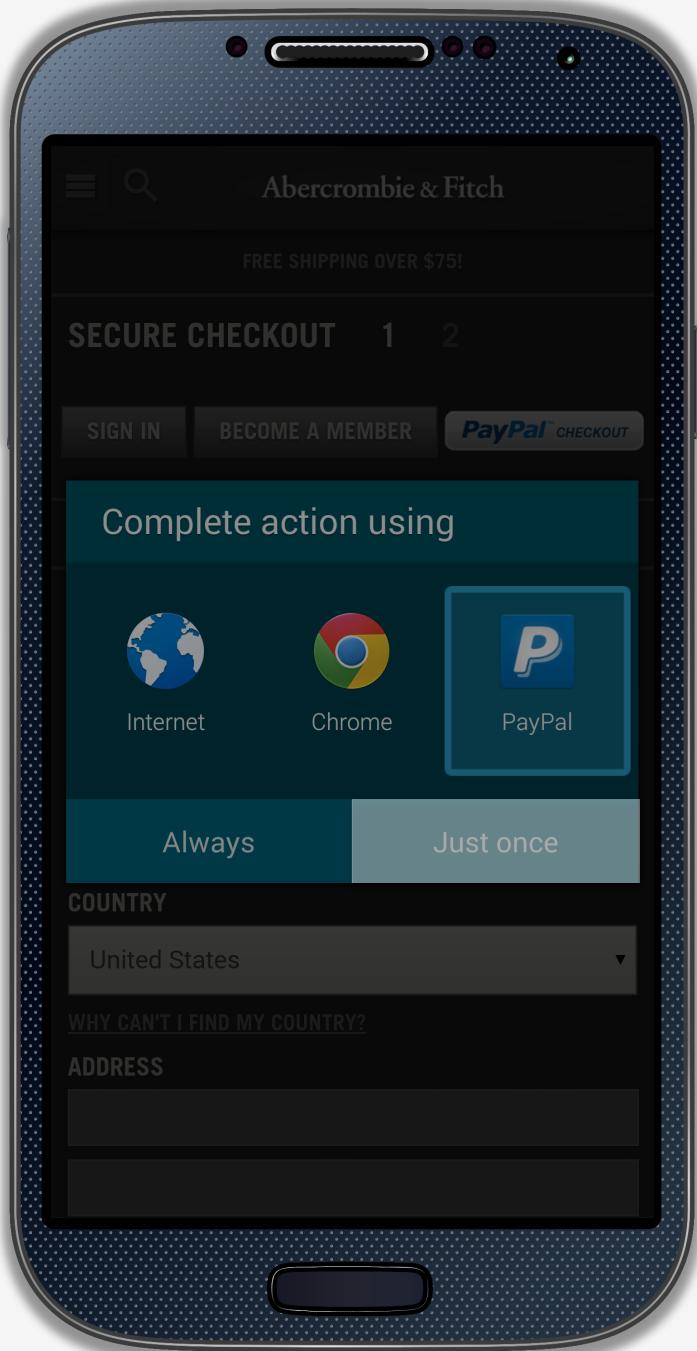
UAF Authentication



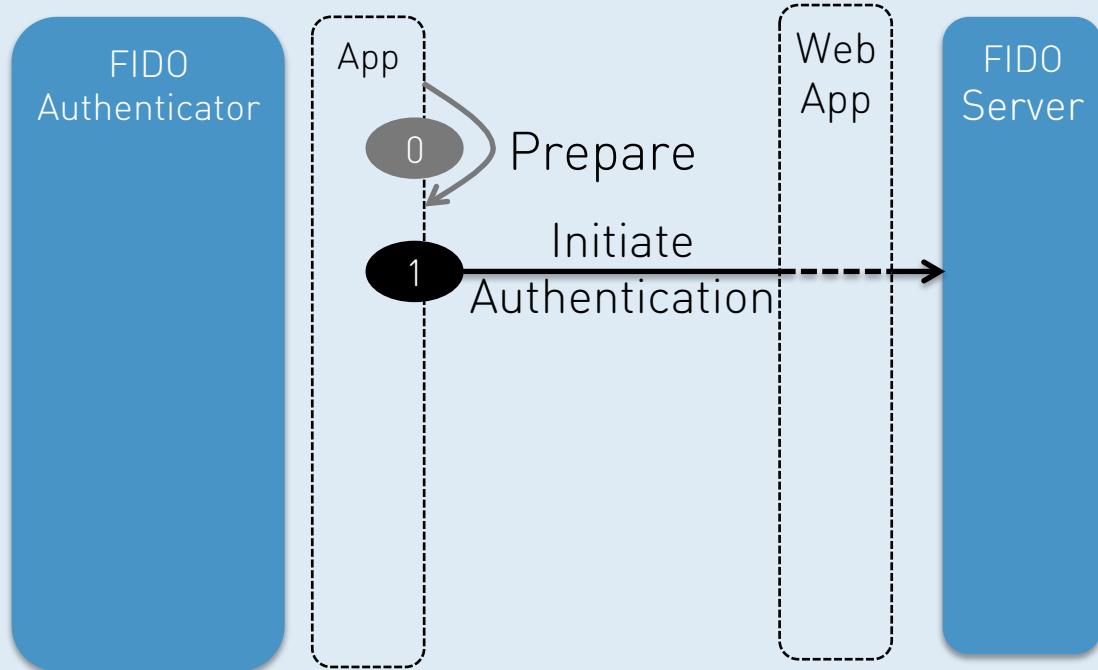
UAF Authentication



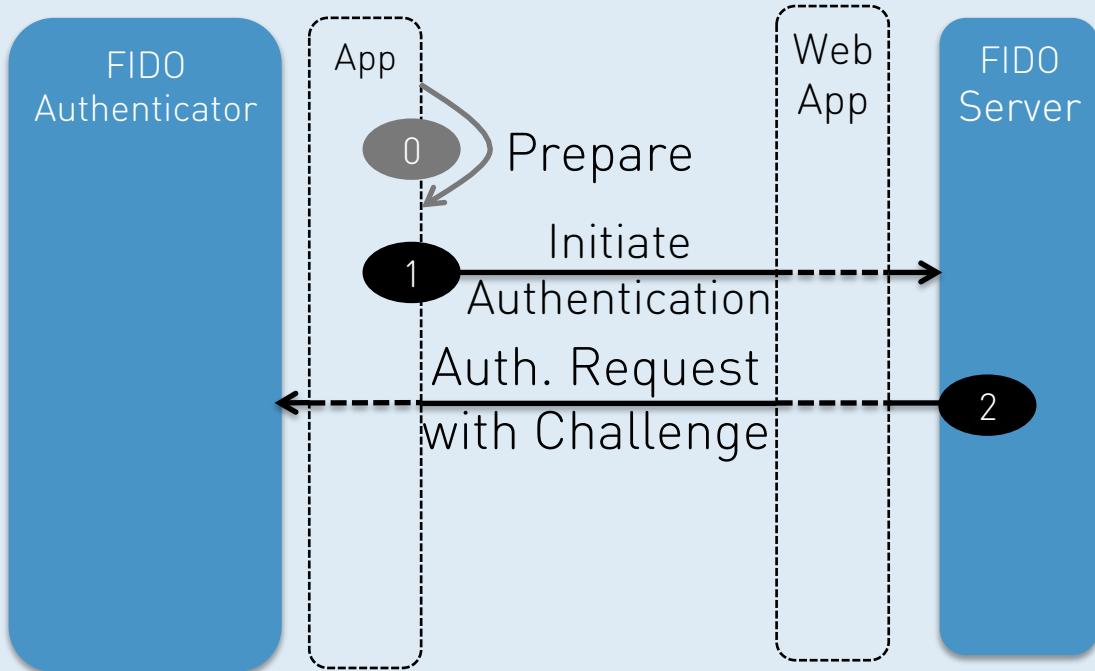
UAF Authentication



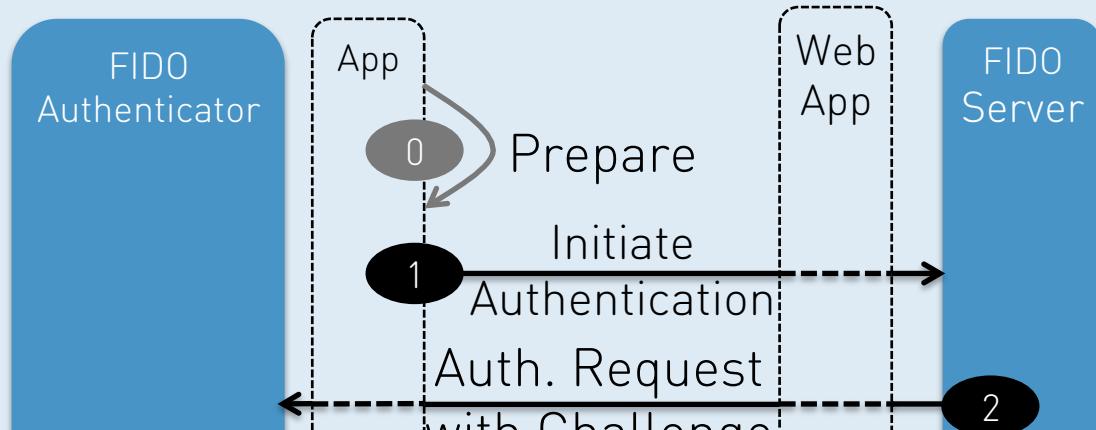
UAF Authentication



UAF Authentication

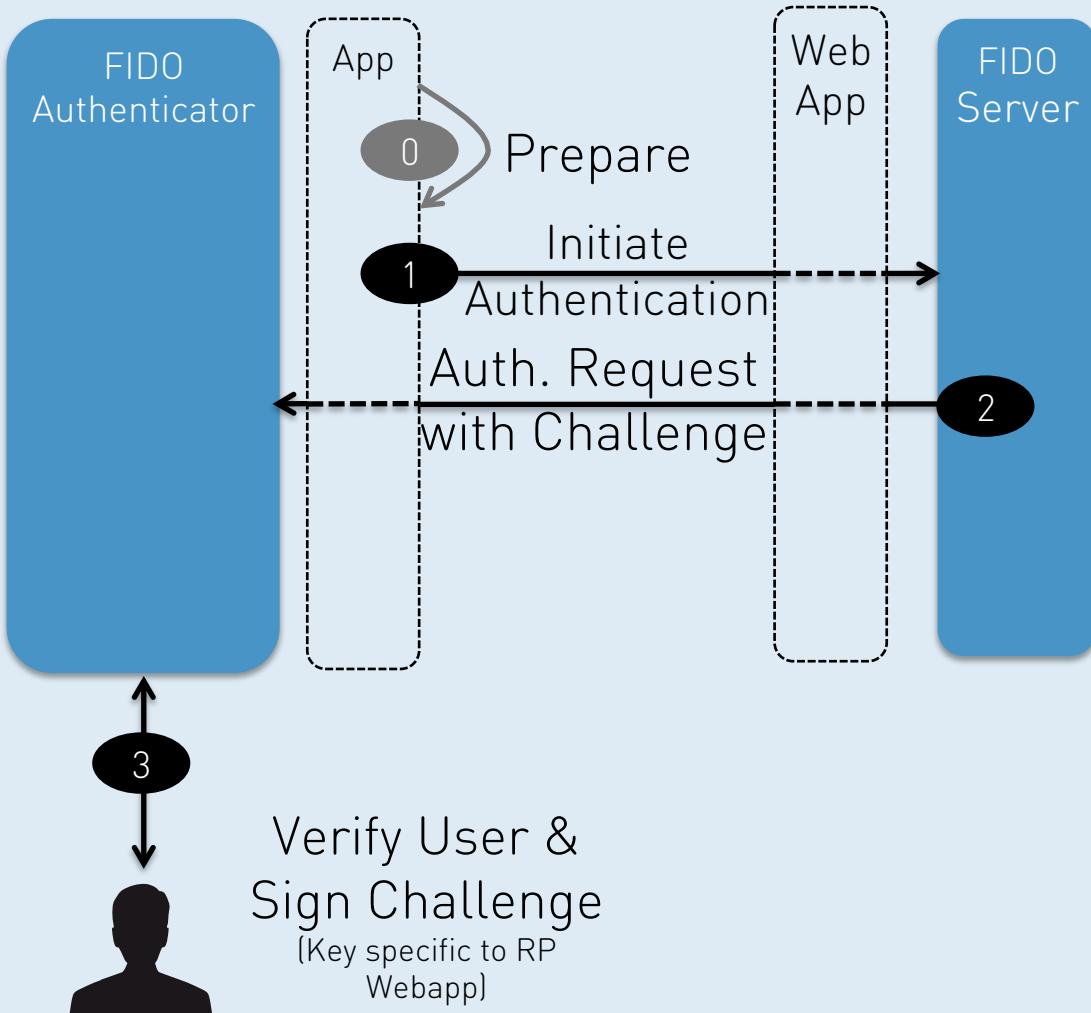
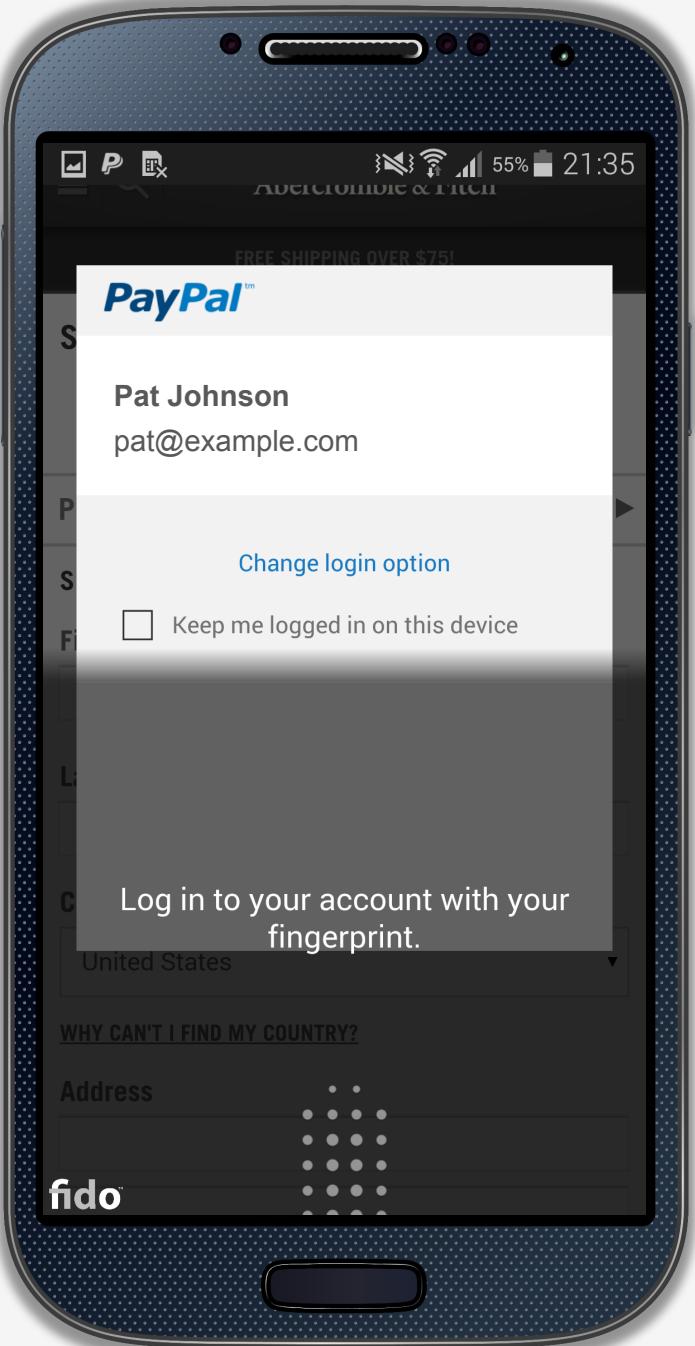


UAF Authentication

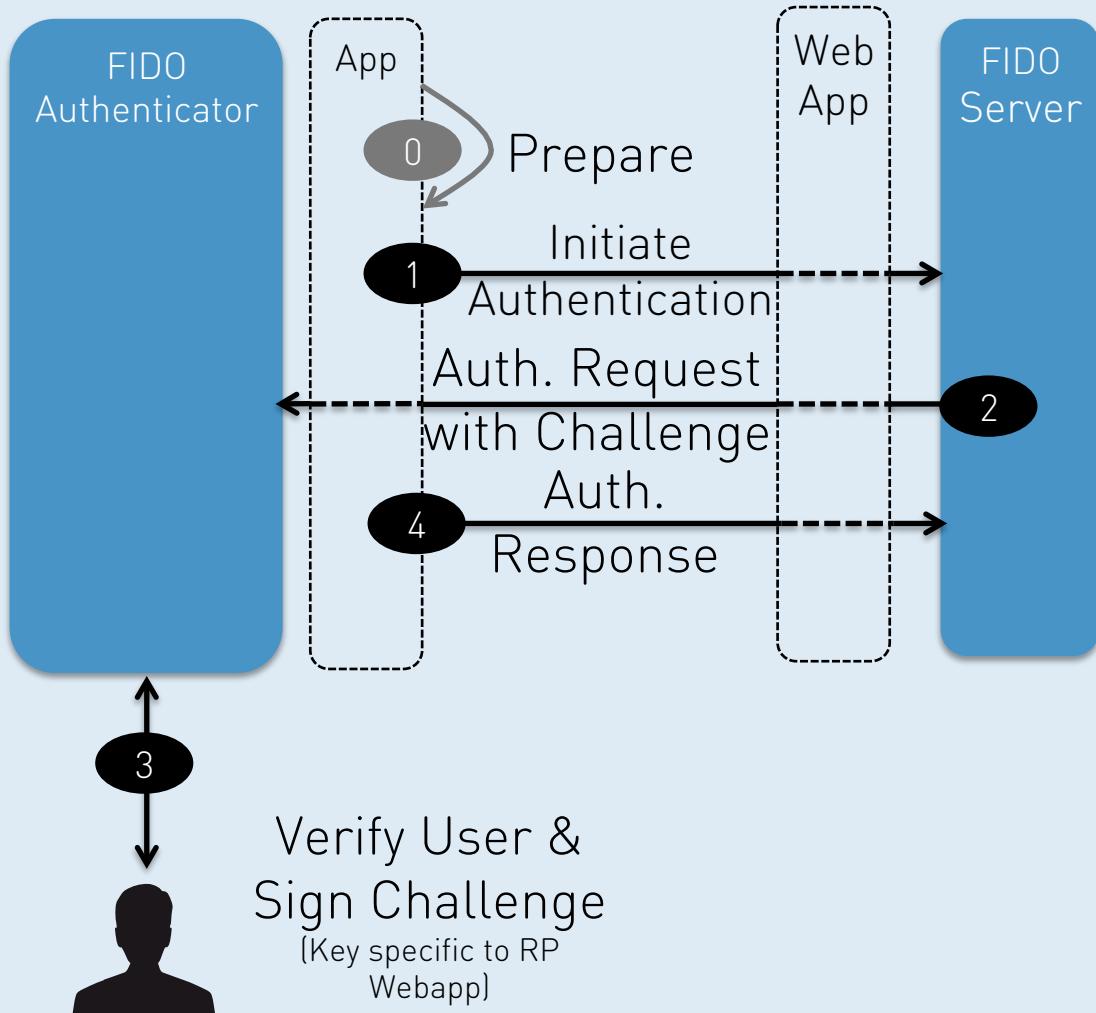
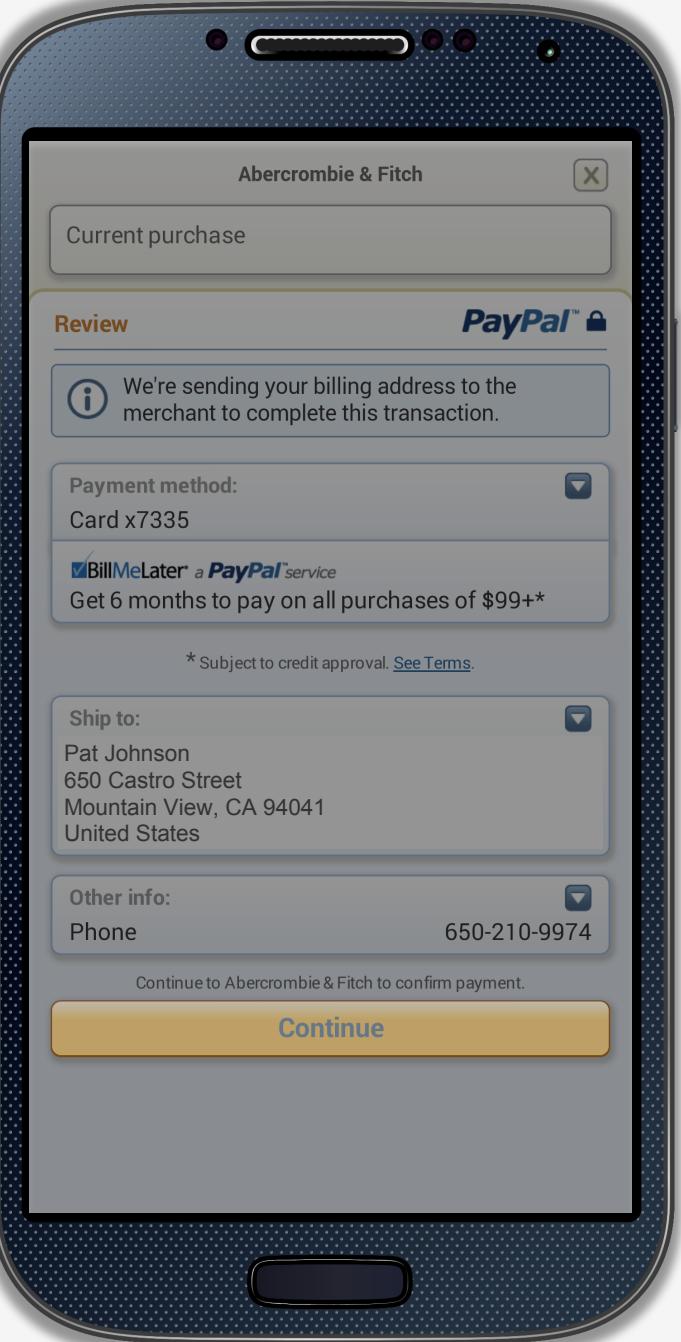


```
[{"header": {"op": "Auth", "upv": "1.0", "appID": "https://mycorp.com/fido"},  
 "challenge": "triz786ighwer8764g6574234515reg45z",  
 "policy": {  
     "accepted": [[{  
         "authenticationFactor": 0000000000001ff,  
         "keyProtection": 00000000000000e,  
         "attachment": 000000000000ff,  
         "secureDisplay": 000000000000001e,  
         "supportedSchemes": "UAFV1TLV"}]],  
     "disallowed": {"aaid": "1234#5678"}  
 }]
```

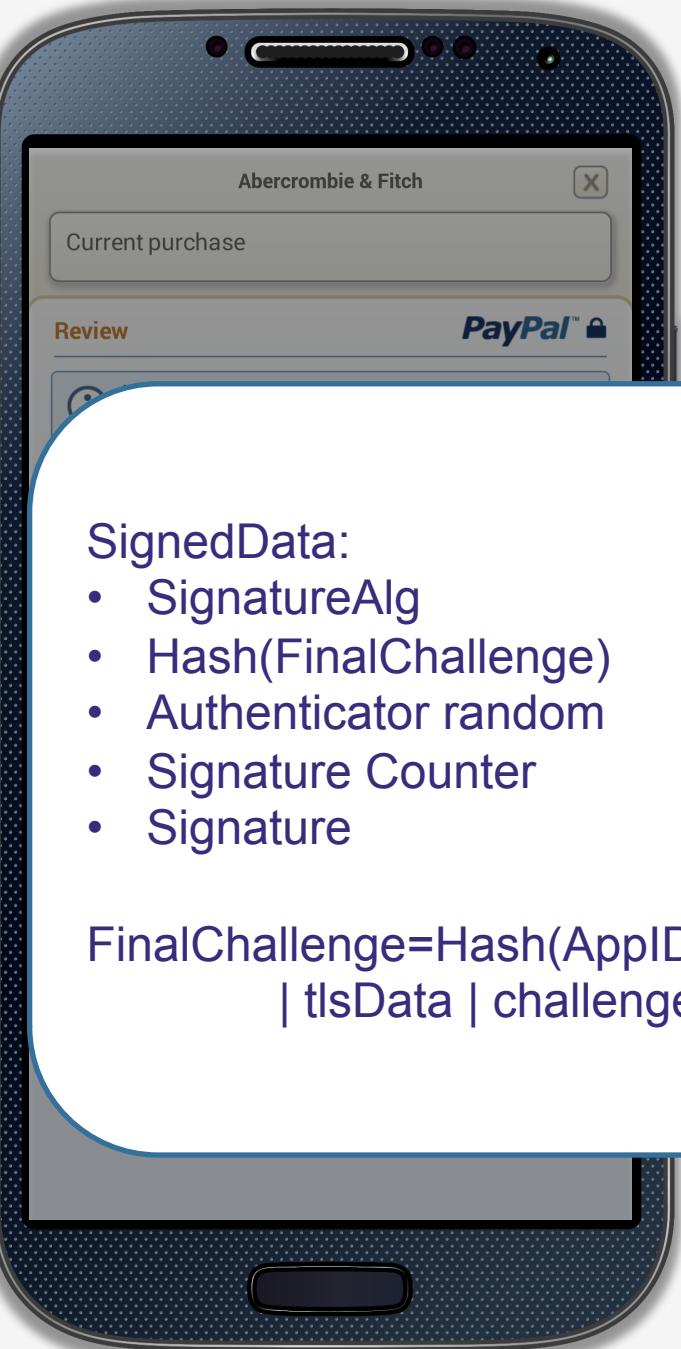
UAF Authentication



UAF Authentication



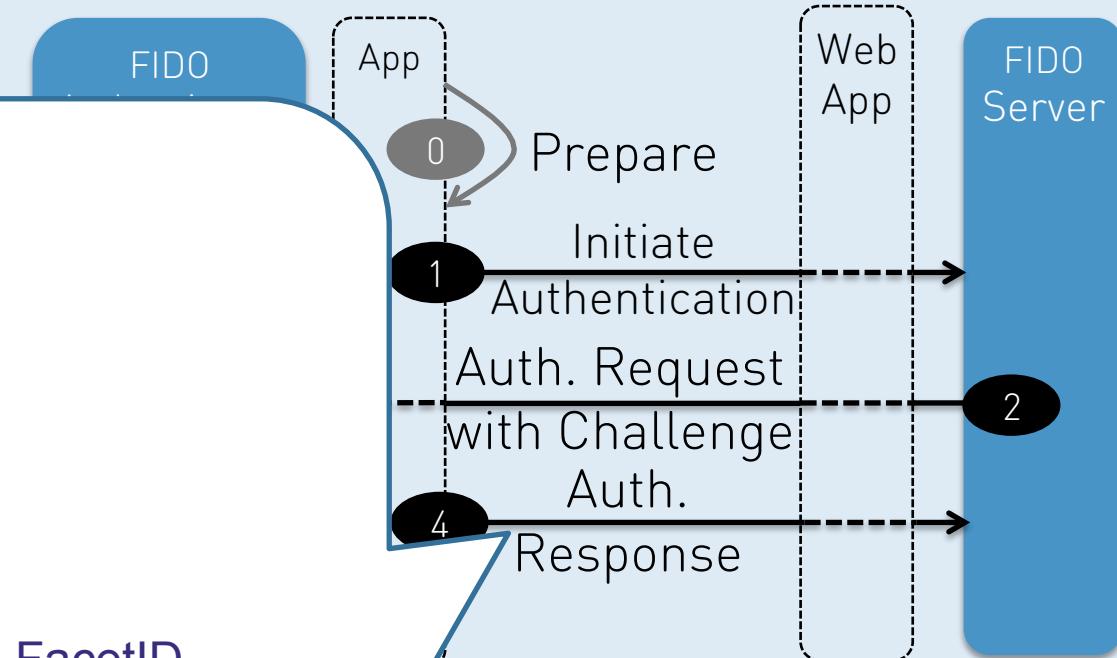
UAF Authentication



SignedData:

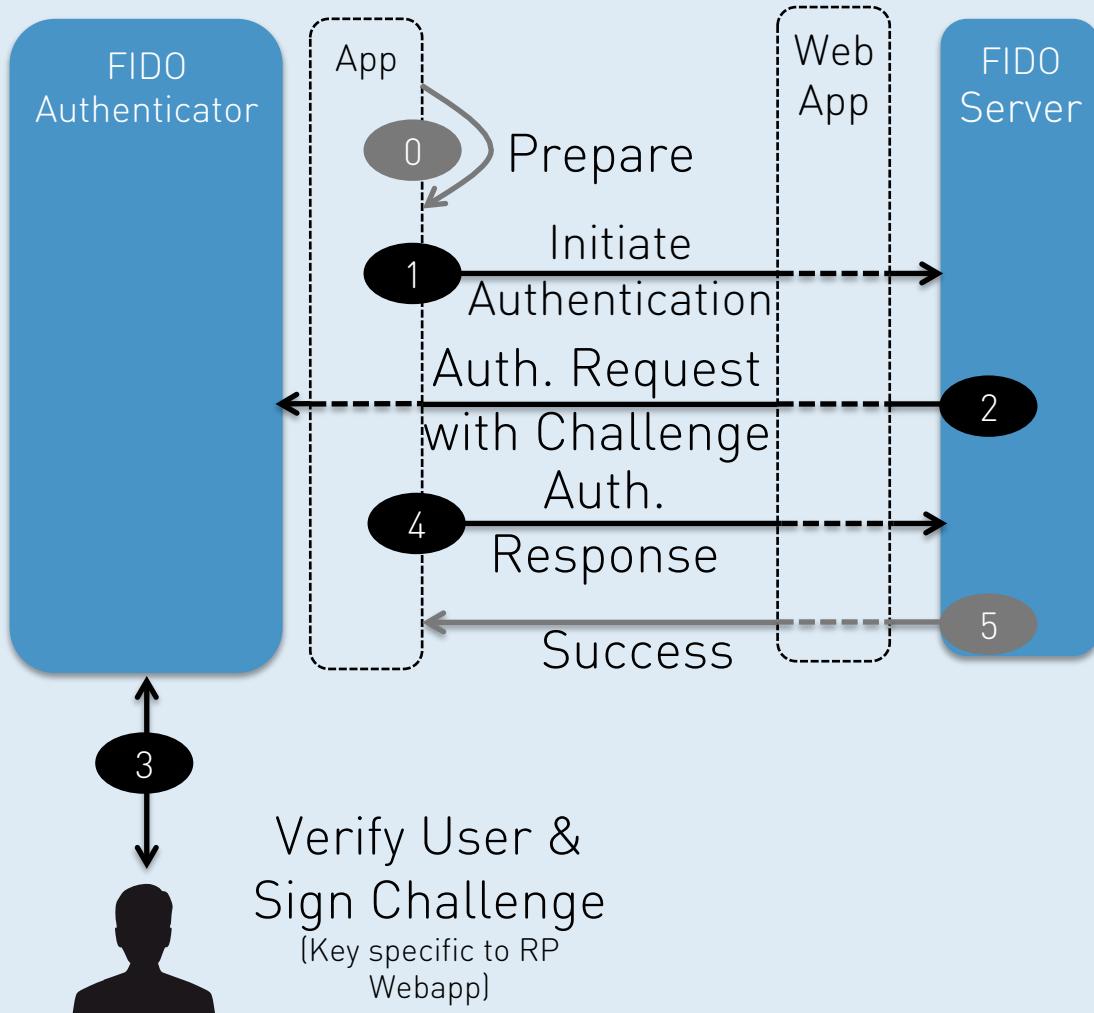
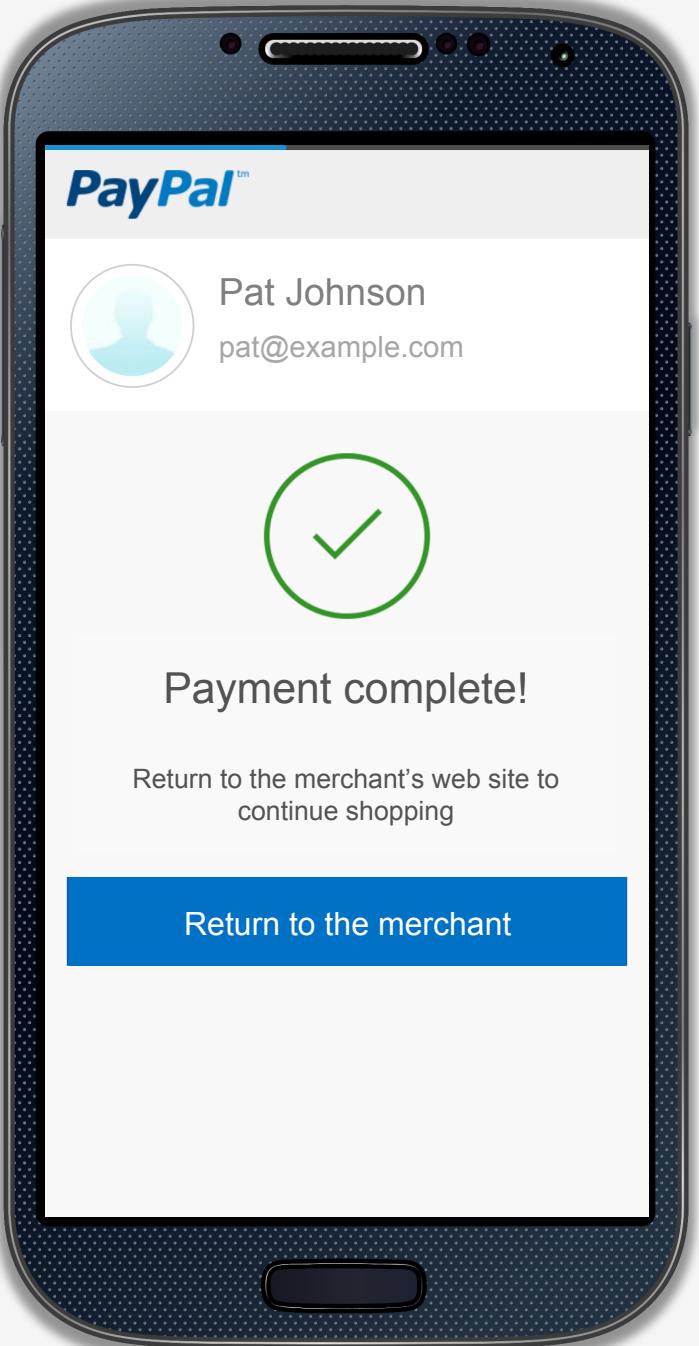
- SignatureAlg
- Hash(FinalChallenge)
- Authenticator random
- Signature Counter
- Signature

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



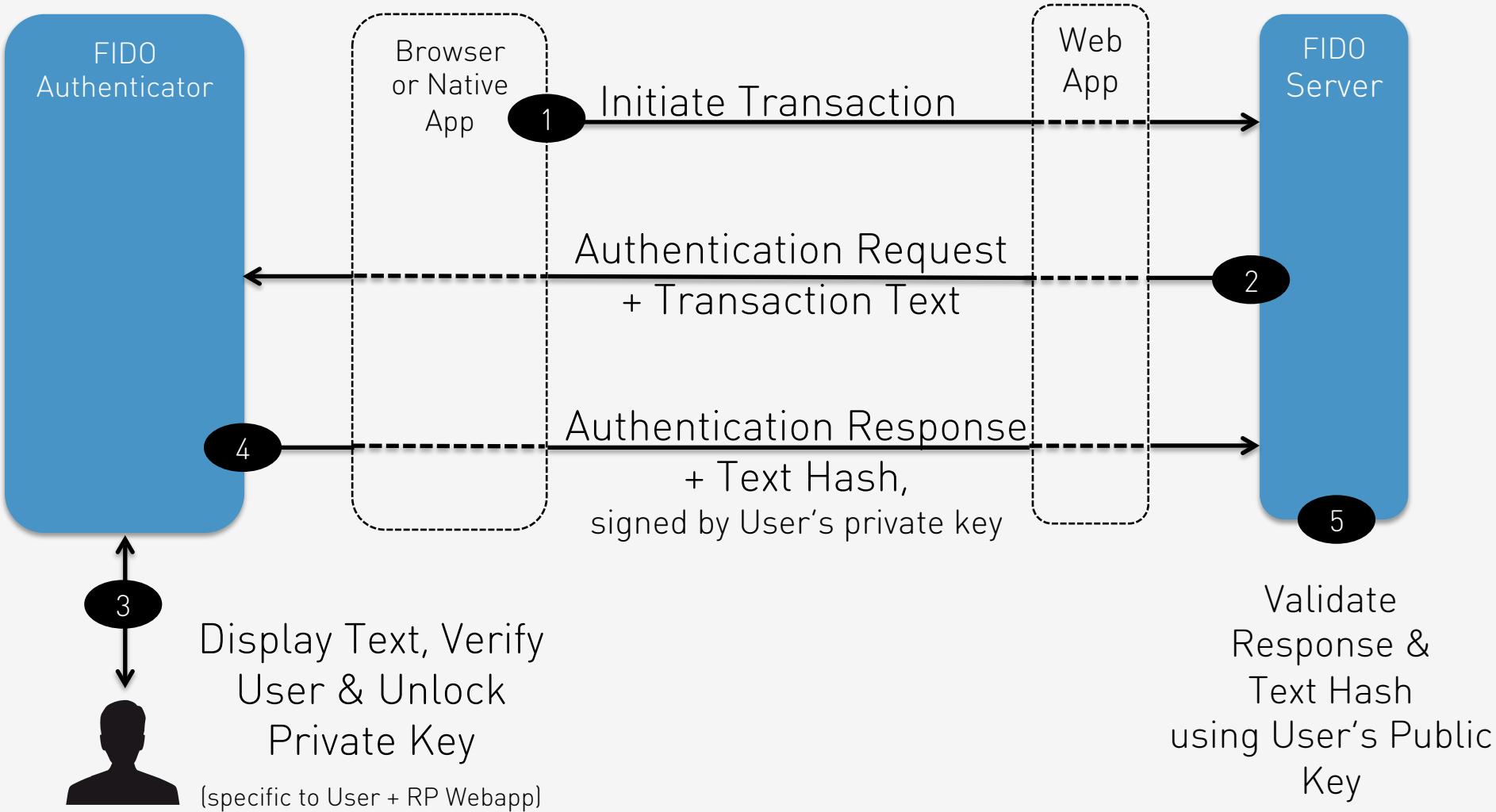
Verify User &
Sign Challenge
(Key specific to RP
Webapp)

UAF Authentication



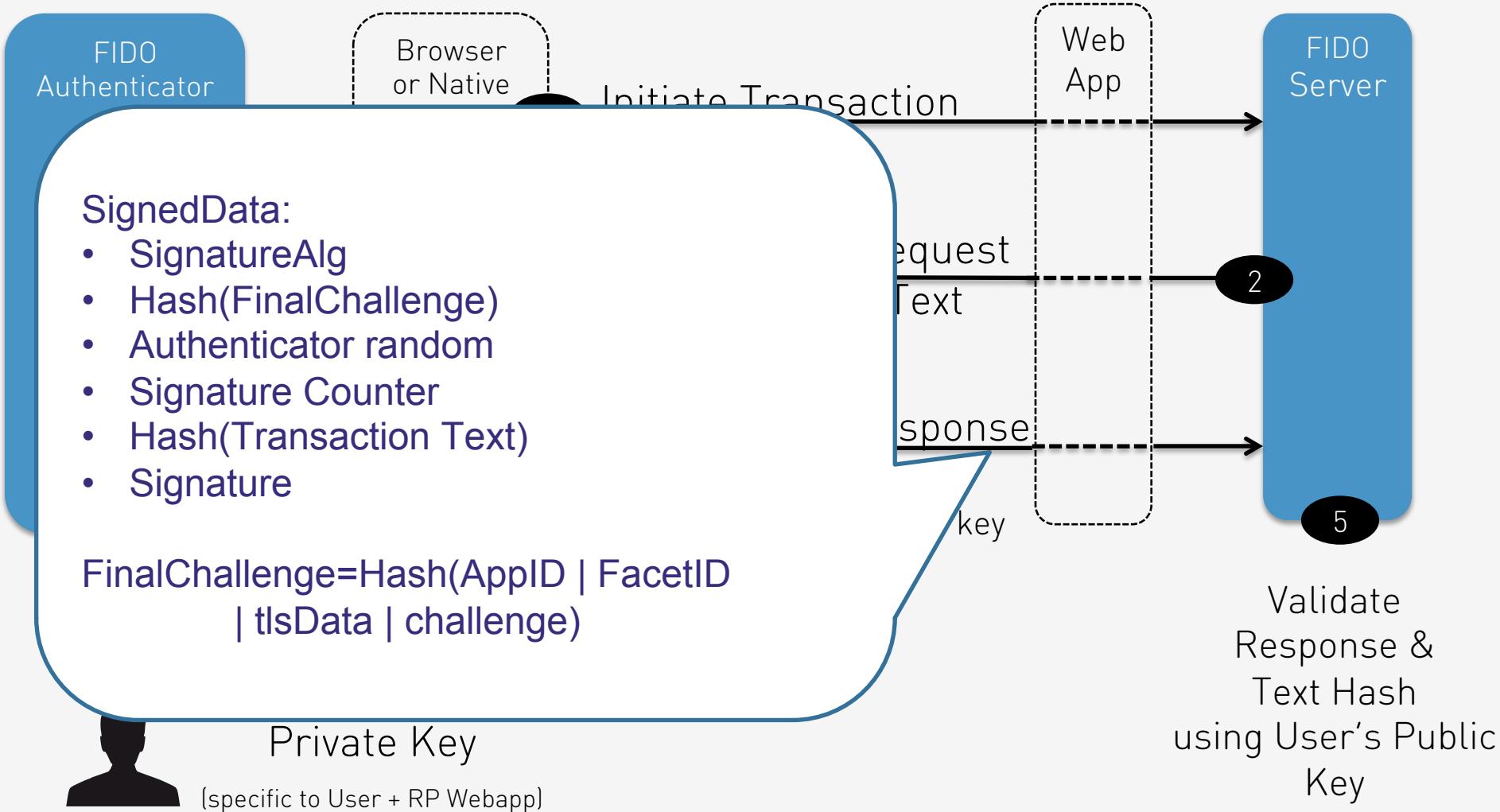
Transaction Confirmation

Device	Relying Party
--------	---------------

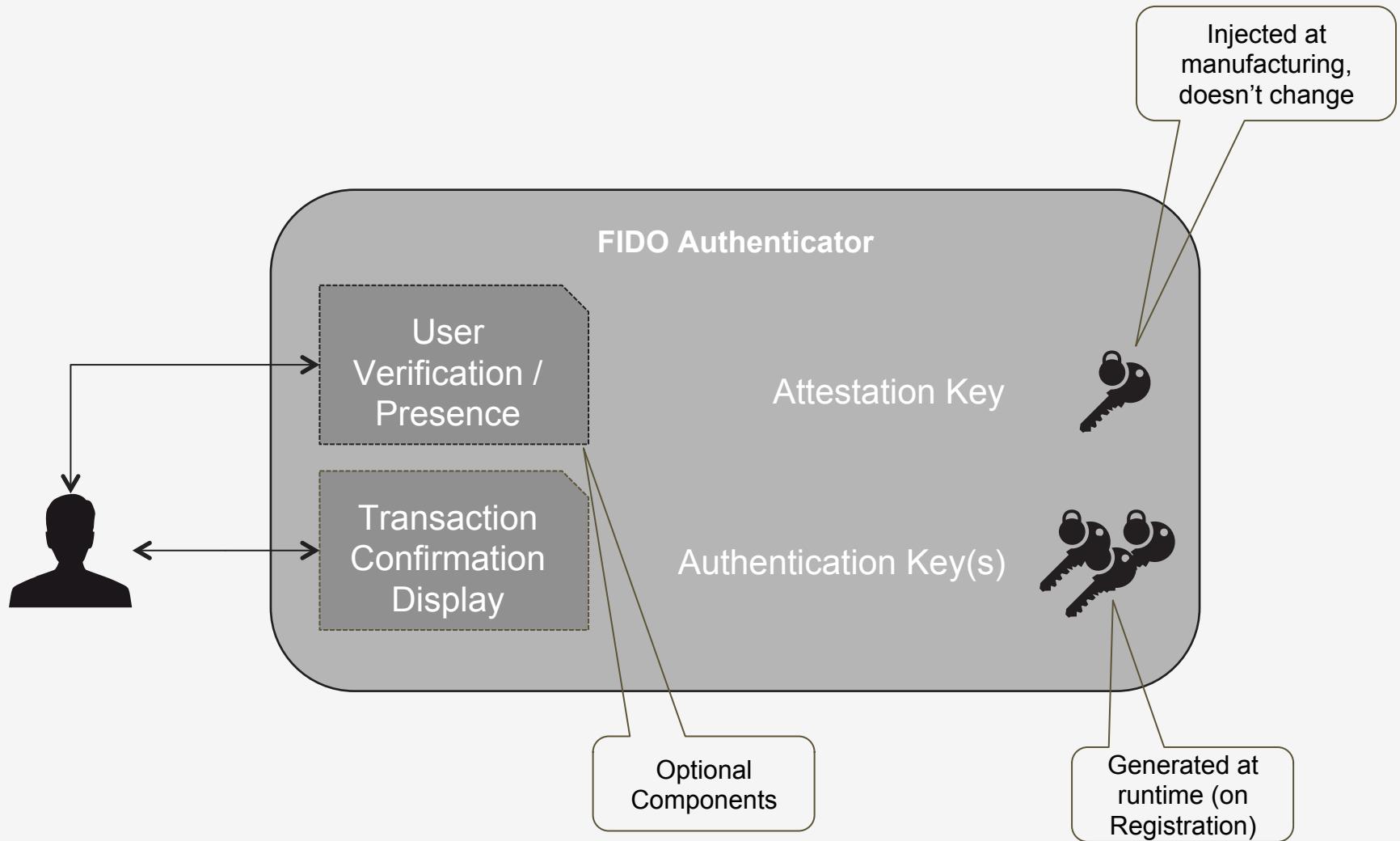


Transaction Confirmation

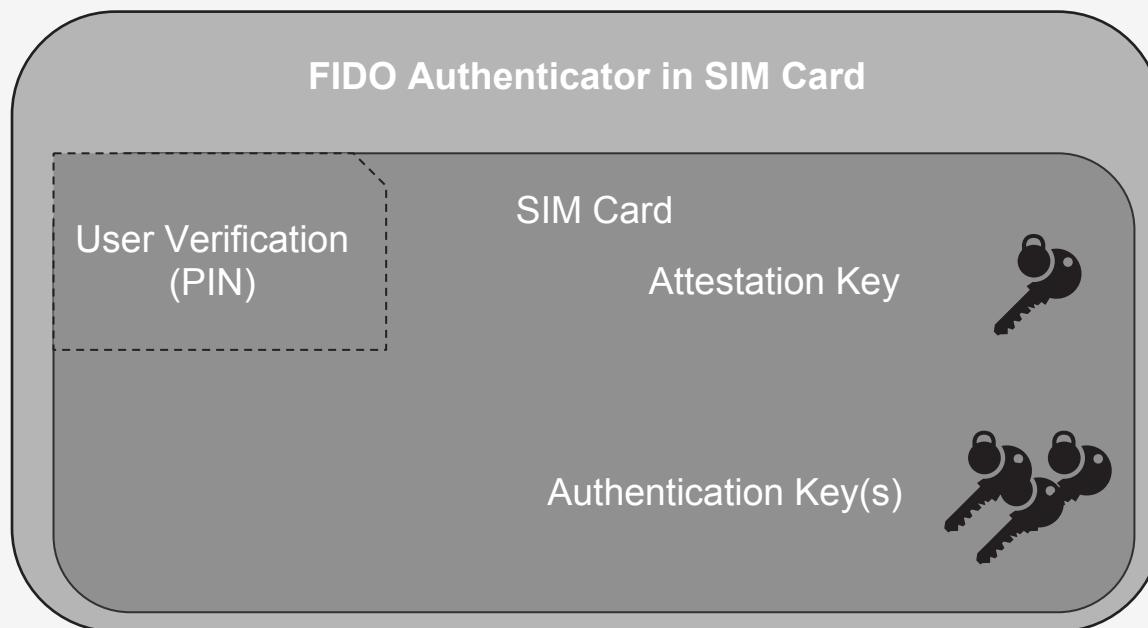
Device	Relying Party
--------	---------------



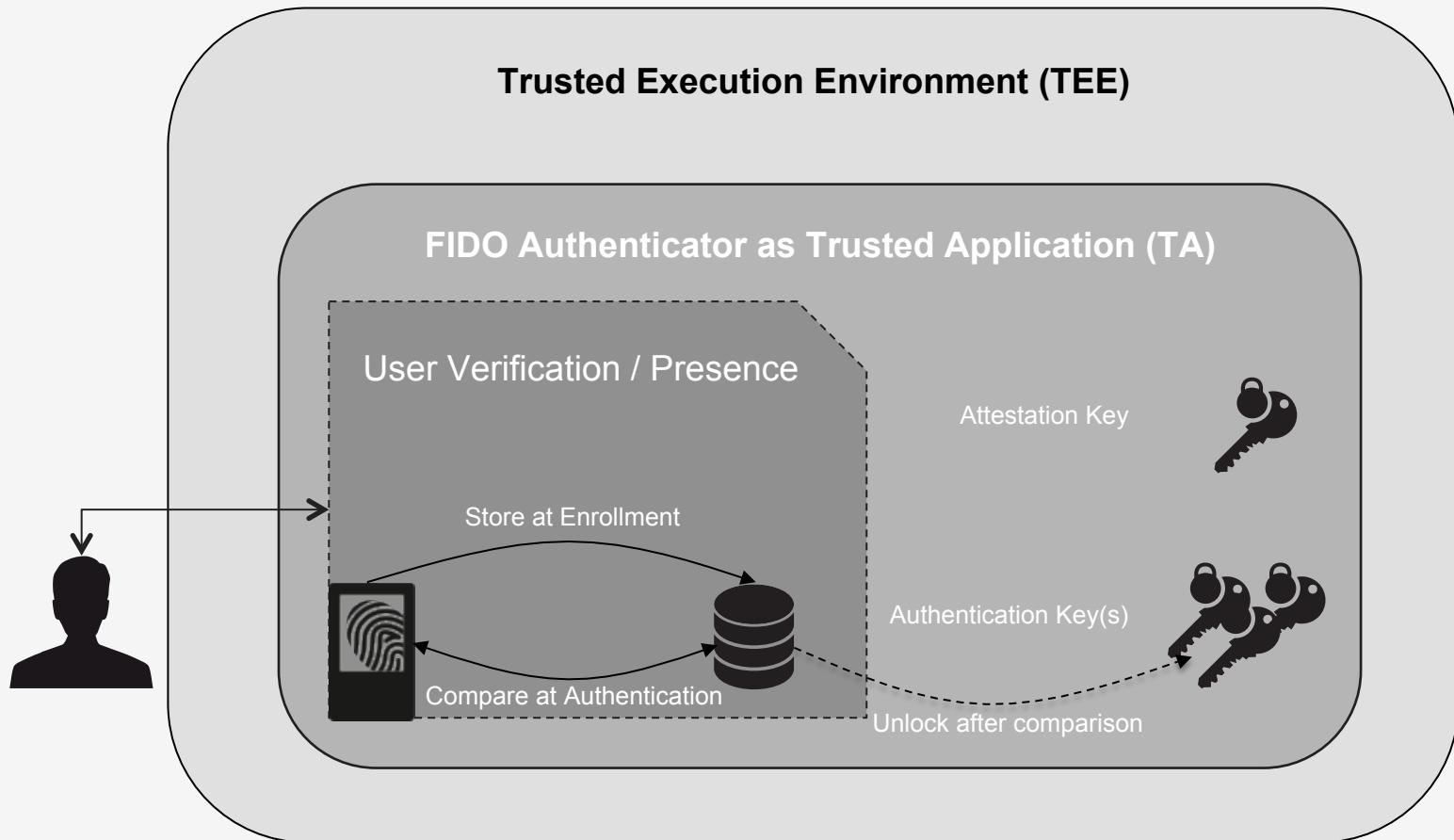
The FIDO Authenticator Concept



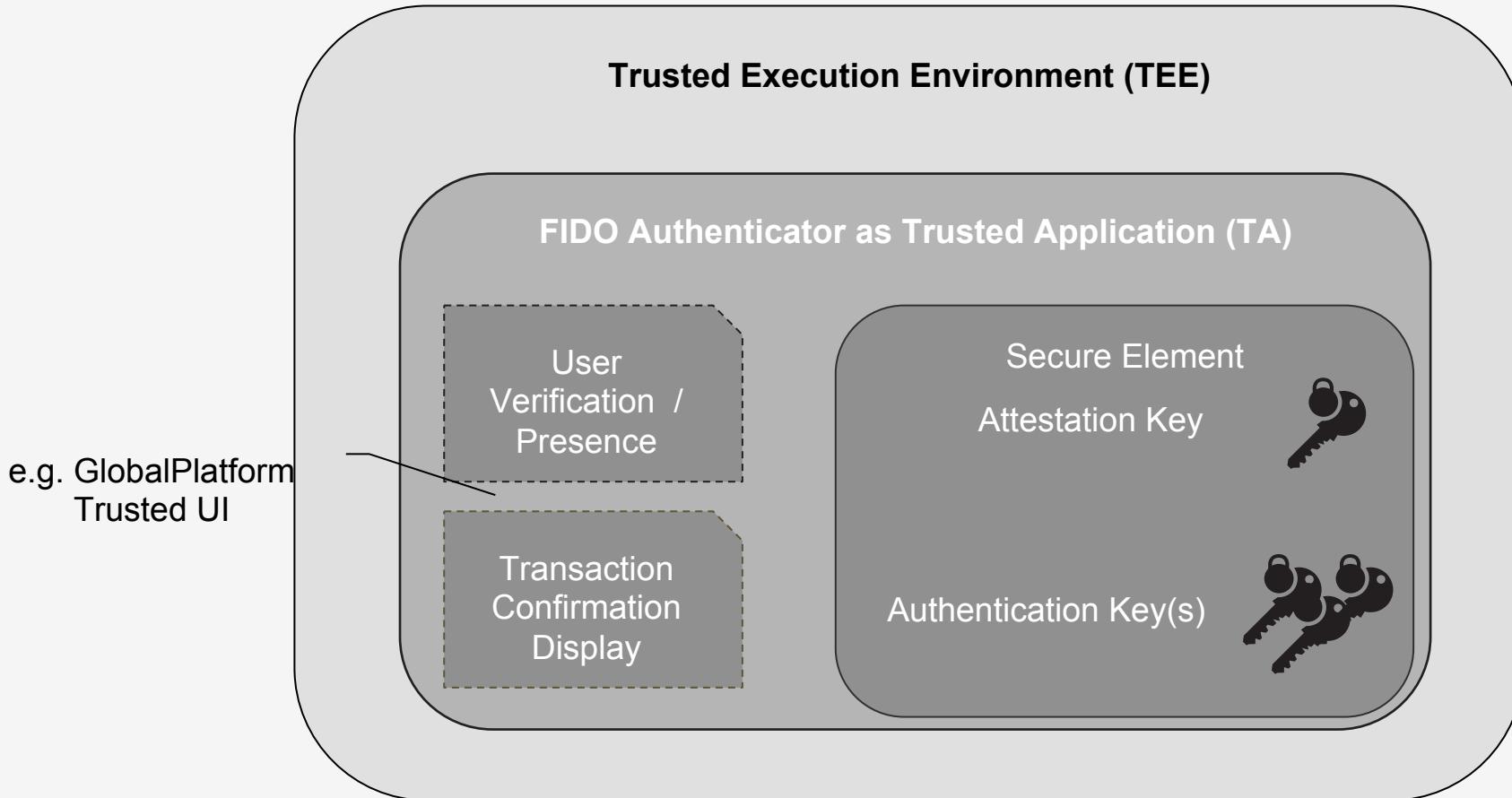
Using Secure Hardware



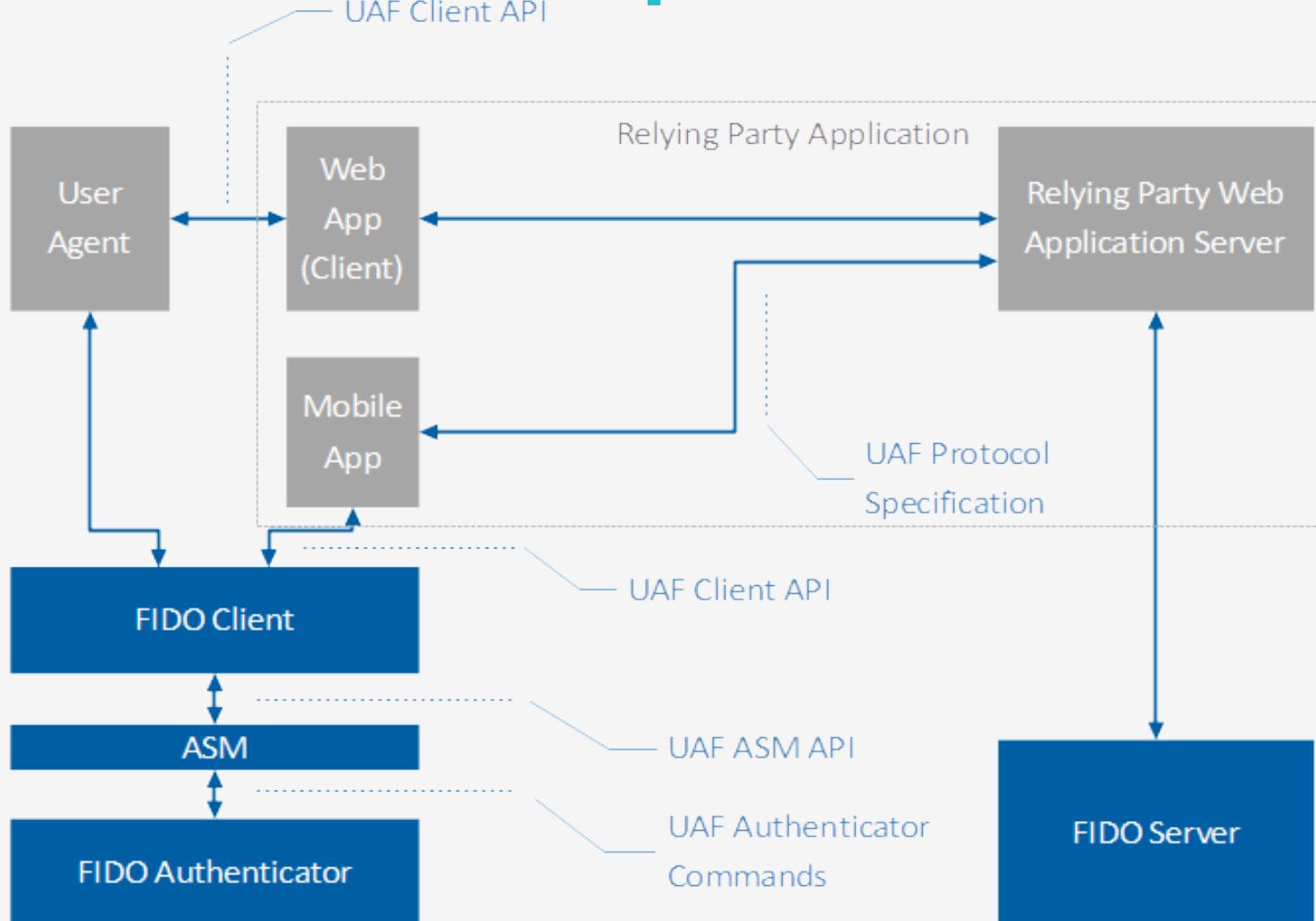
Client Side Biometrics



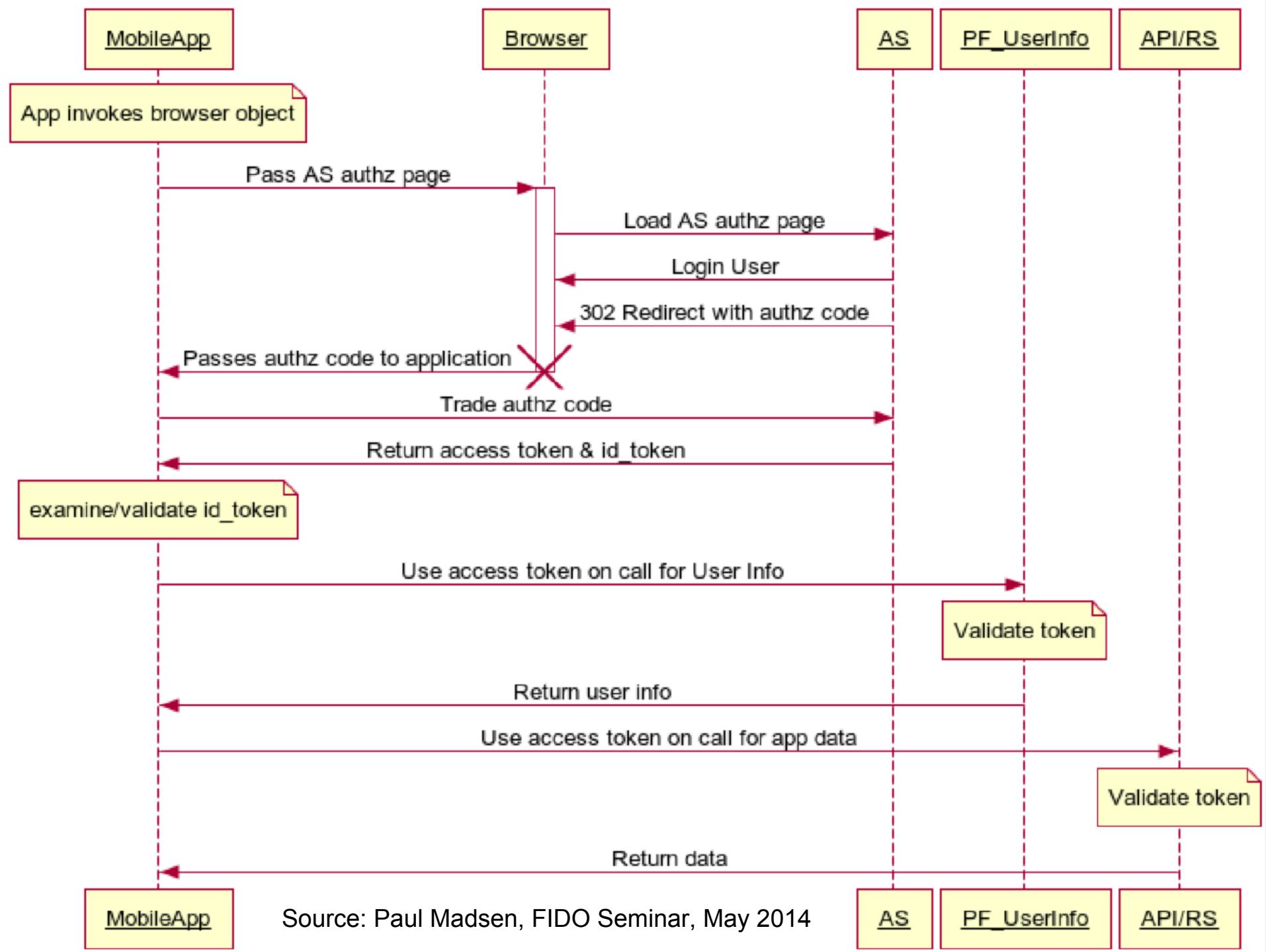
Combining TEE and SE

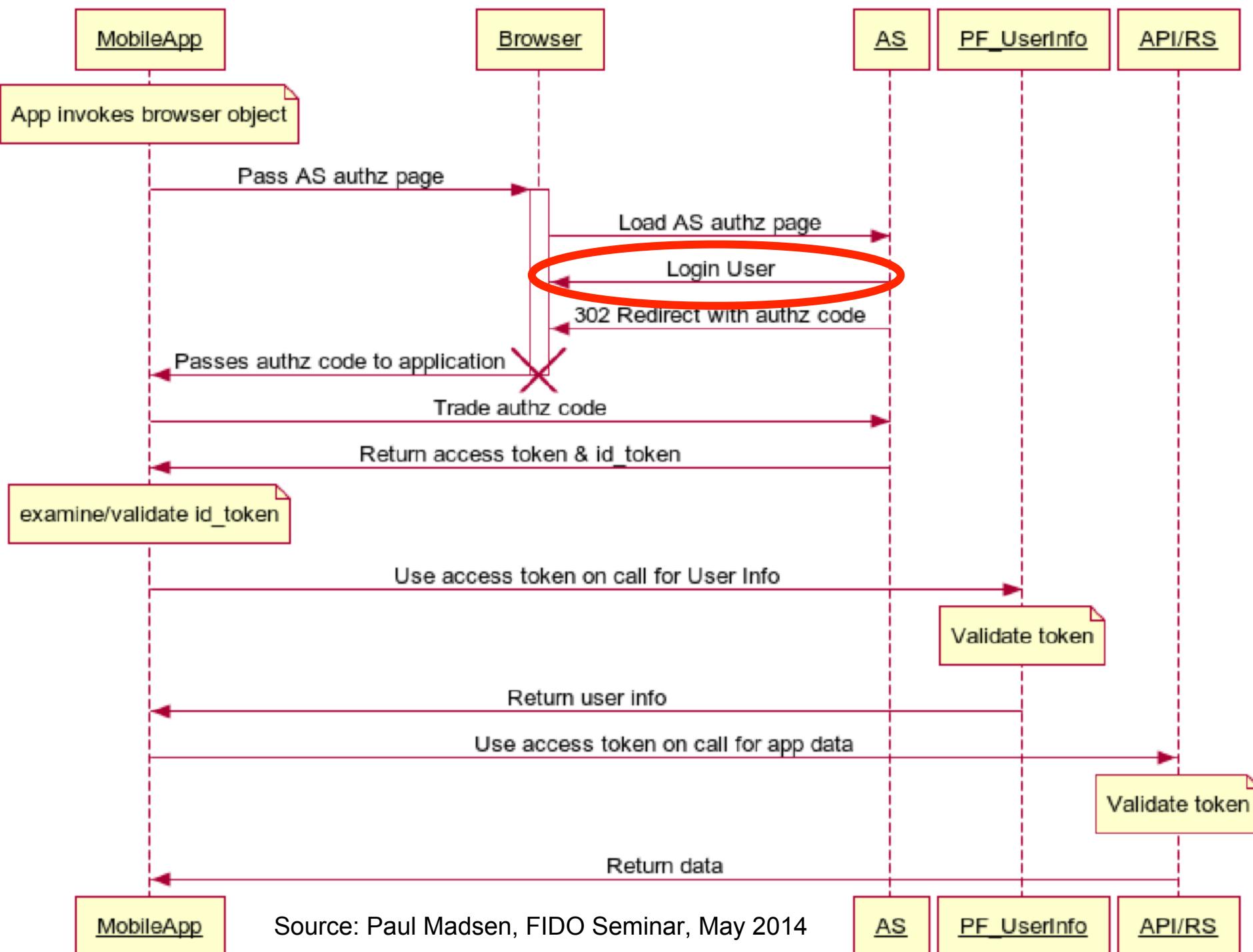


UAF Specifications



FIDO & Federation

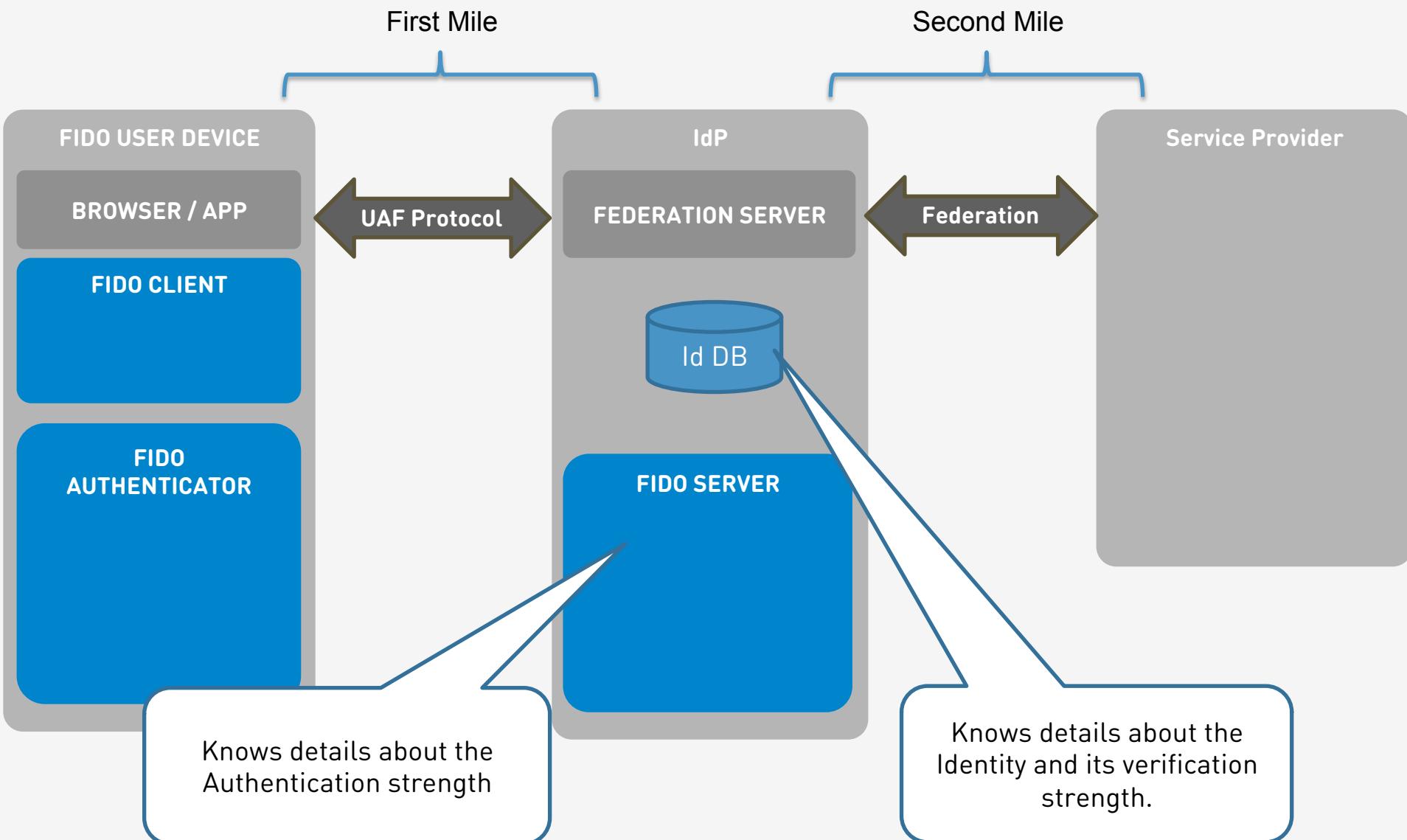




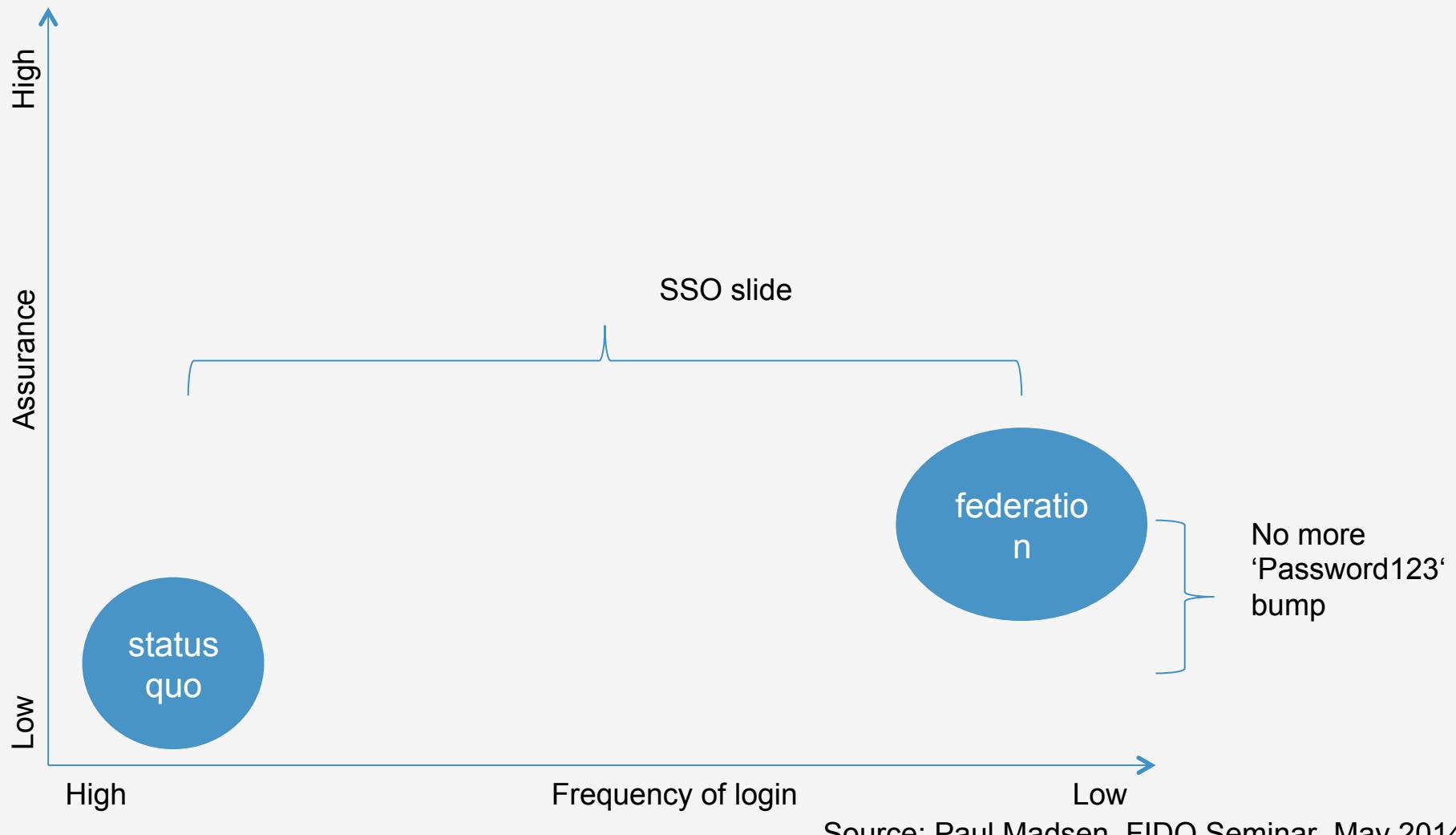
Complementary

- 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

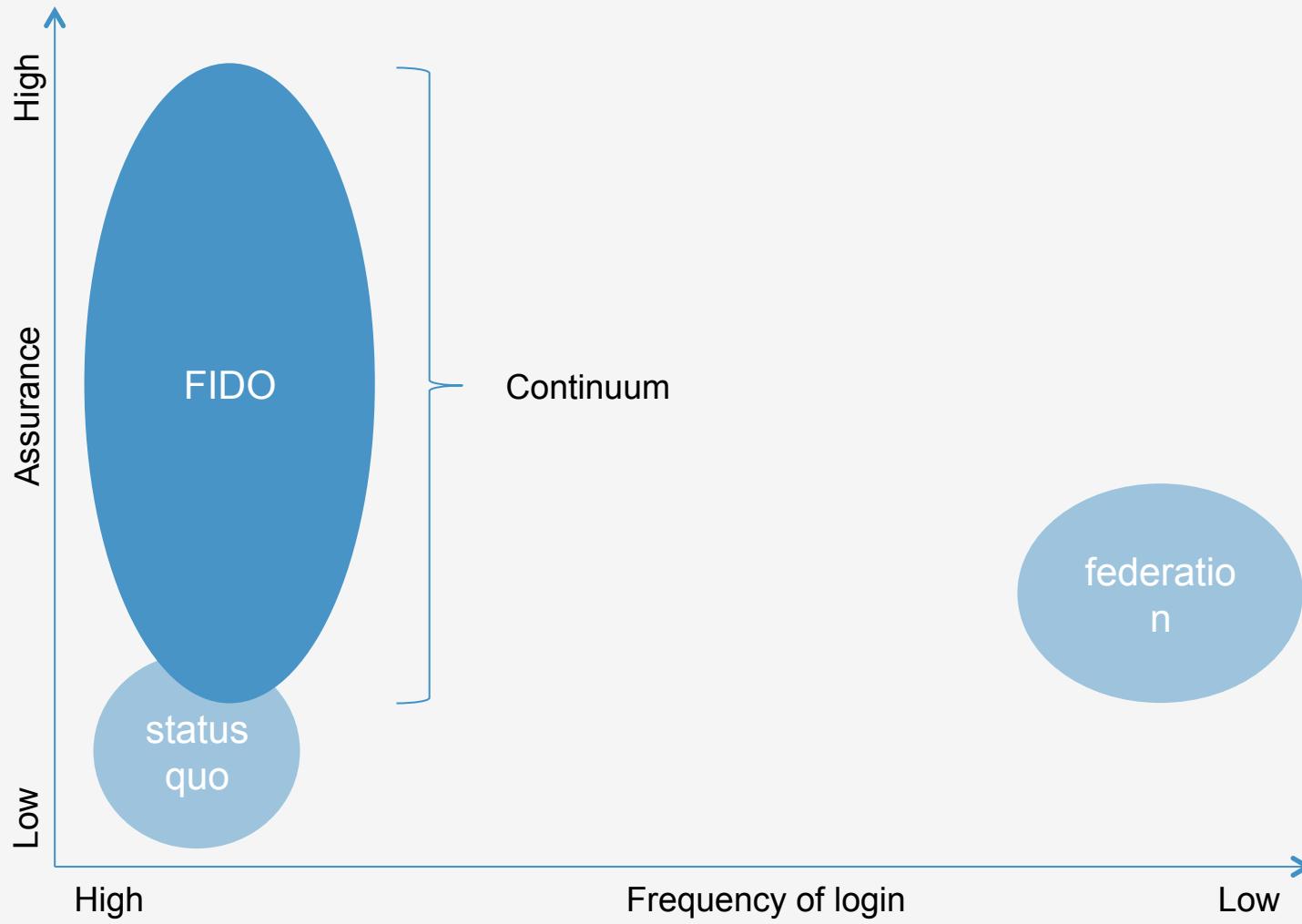
FIDO & Federation



FIDO & Federation

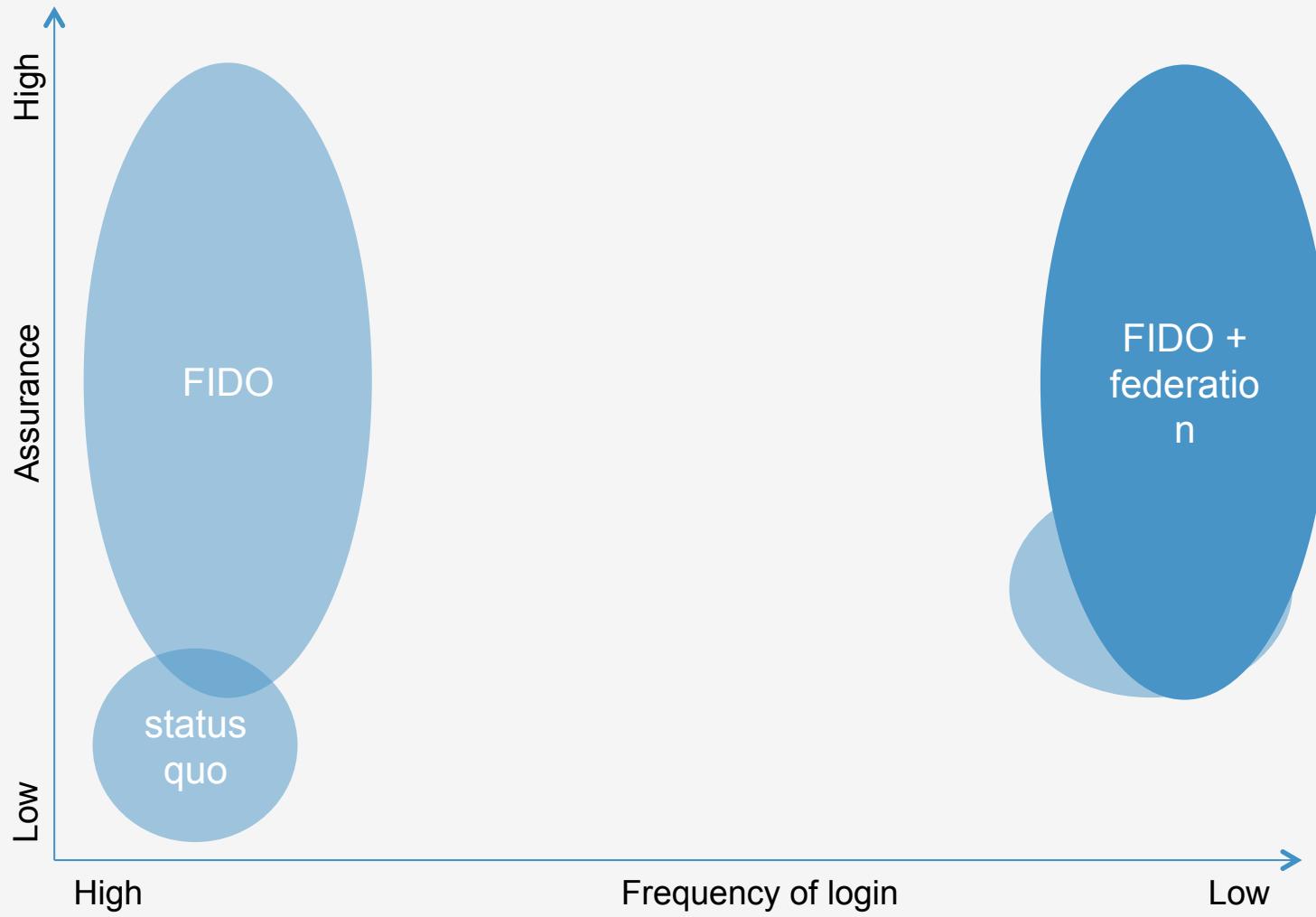


FIDO & Federation



Source: Paul Madsen, FIDO Seminar, May 2014

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

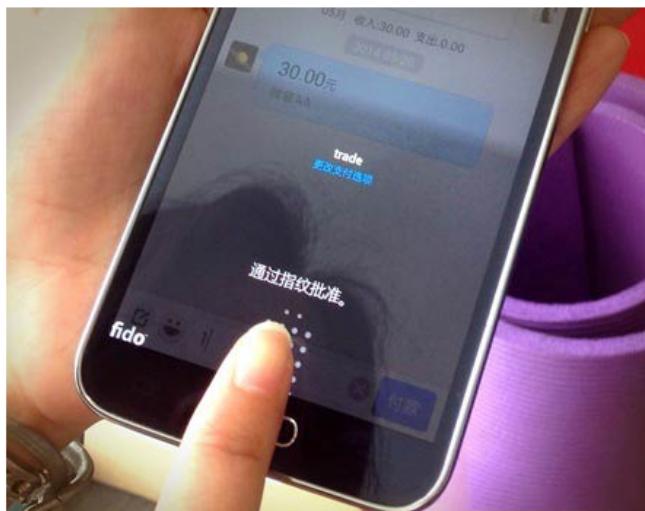


Hello there! If you are new here, you might want to [subscribe](#) to this topic.

+1

follow

Recommend



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



By [Adam Vrankulj](#)

[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.

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

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