FIDO2 – No more Phishing

by Yves Bieri

Yves Bieri – IT Security Analyst – yves.bieri@compass-security.com

POST SCHLÄGT ALARM

«Es grassiert eine Phishing-Welle»

Publiziert 18. September 2020, 20:39

Zurzeit sind im Namen der Post besonders viele Fake-E-Mails im Umlauf. Die Betrüger werden immer cleverer, warnt die Post. Auch Leserreporterin R. A. wäre fast auf die Masche reingefallen.

Achtung Phishing-Welle!

Betrüger geben sich als Swisscom aus

Die basellandschaftliche Polizei rät zur Vorsicht: Betrüger versuchen mittels gefälschter Rechnungen an die Logins und Kreditkartendaten ihrer potentiellen Opfer zu kommen.

Datendiebstahl per Mail

SBB und Post kämpfen mit Phishingwelle

In den letzten Tagen sind als Quittungen getarnte Mails im Umlauf. Diese sehen täuschend echt aus – doch es sind Fallen für unachtsame Kundinnen und Kunden.

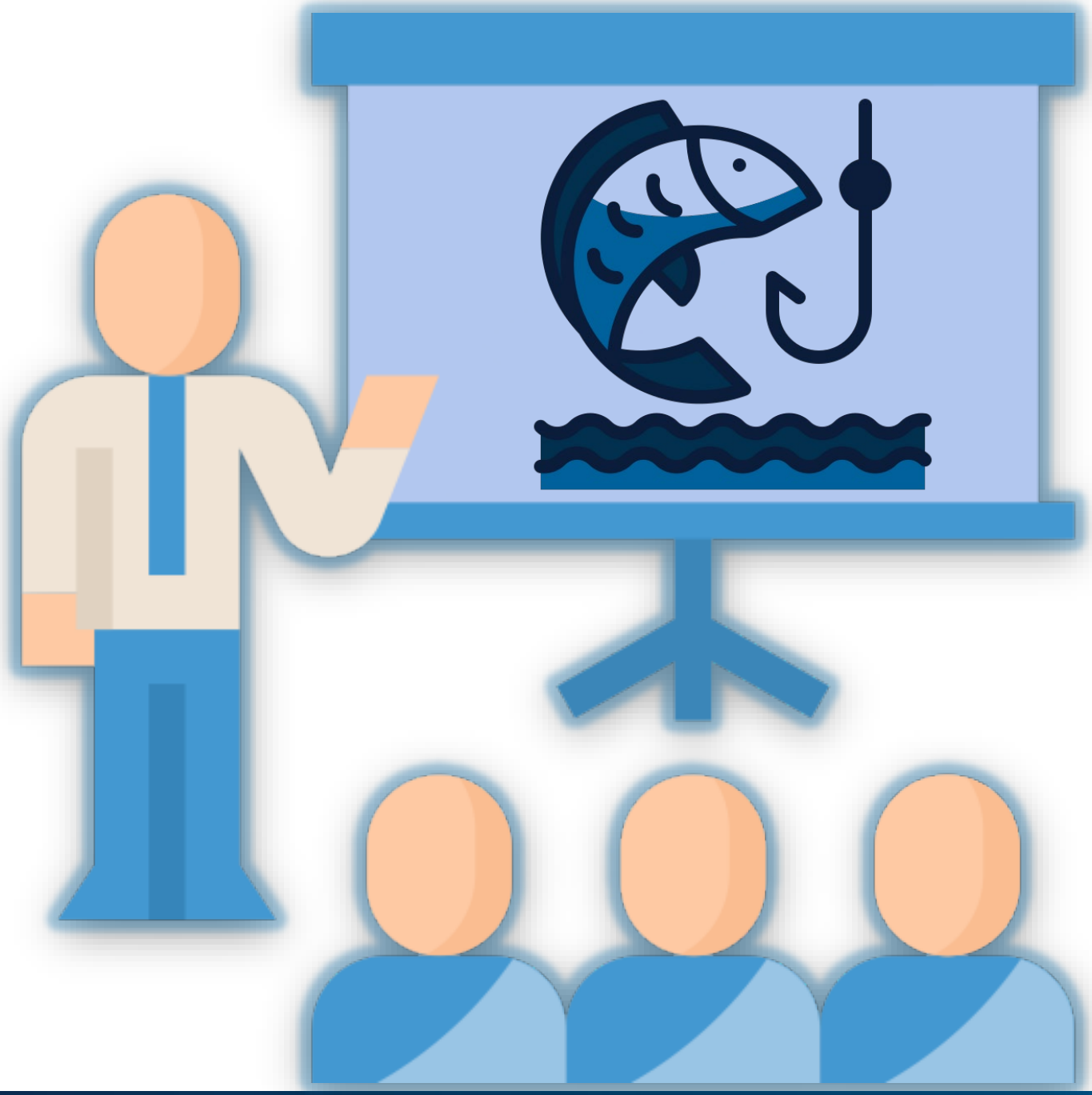
Achtung vor Phishing-Mails!

So dreist zocken Betrüger Postkunden ab

Plötzlich eine hohe Rechnung im Briefkasten, obwohl Sie nichts bestellt haben? Kriminelle nutzen Phishing-Mails um an persönliche Daten zu kommen. So ergaunern sie sich hunderte Pakete. Die Post hat nun reagiert.

45 Milliarden Dollar Schaden: Betrüger weiterhin mit Phishing und Sextortion erfolgreich

Phishing-Fall schädigte Dutzende Schweizer Bankkunden




FIDO2

“Traditional” 2FA

Authentication Factors

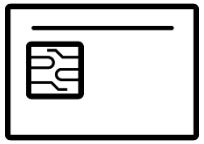
Authentication may involve different factors:

To ***KNOW*** something

Password, PIN

To ***OWN*** something

Smartcard, SecurId, Safeword, Vasco, OTP, Yubikey



To ***BE*** something

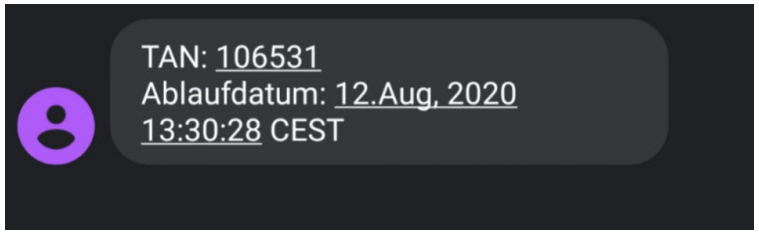
Fingerprint, Iris, Voice, Face

Multi-Factor Authentication

Combination of at least 2 ***DIFFERENT*** factors

Possession Factor Examples

RSA SecurID

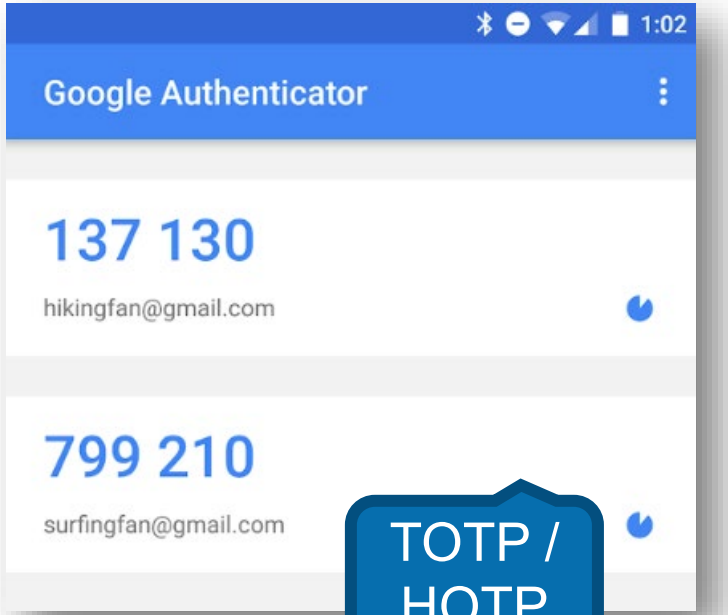


mTAN / SMS

SmartCard

PhotoTAN / Cronto

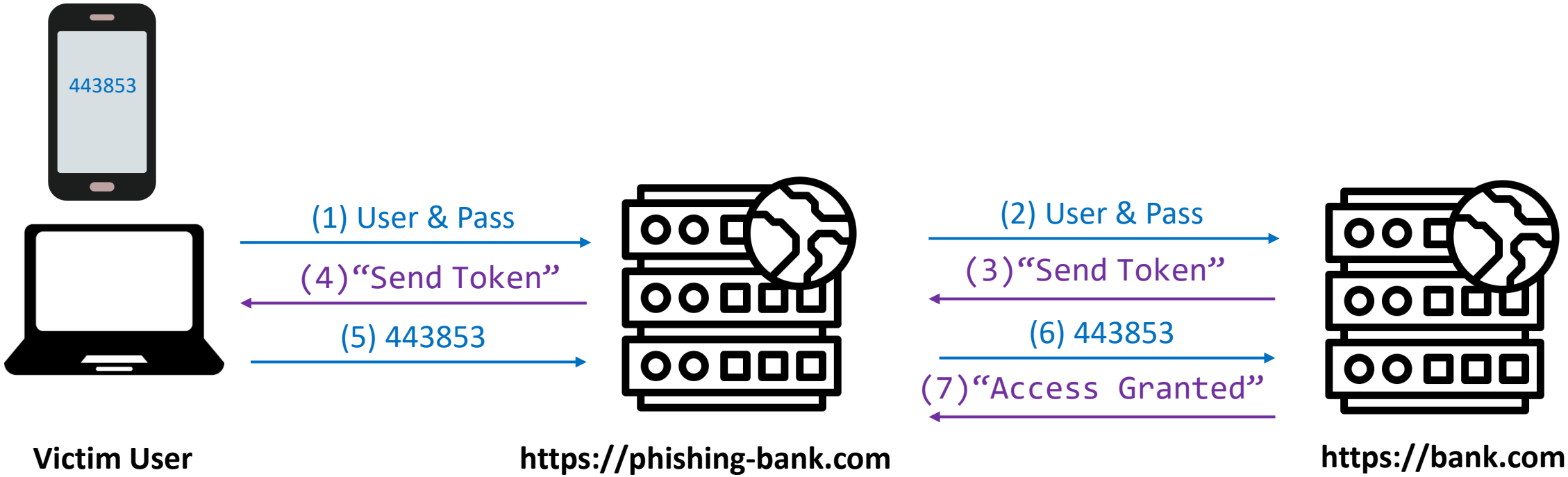
MobileID



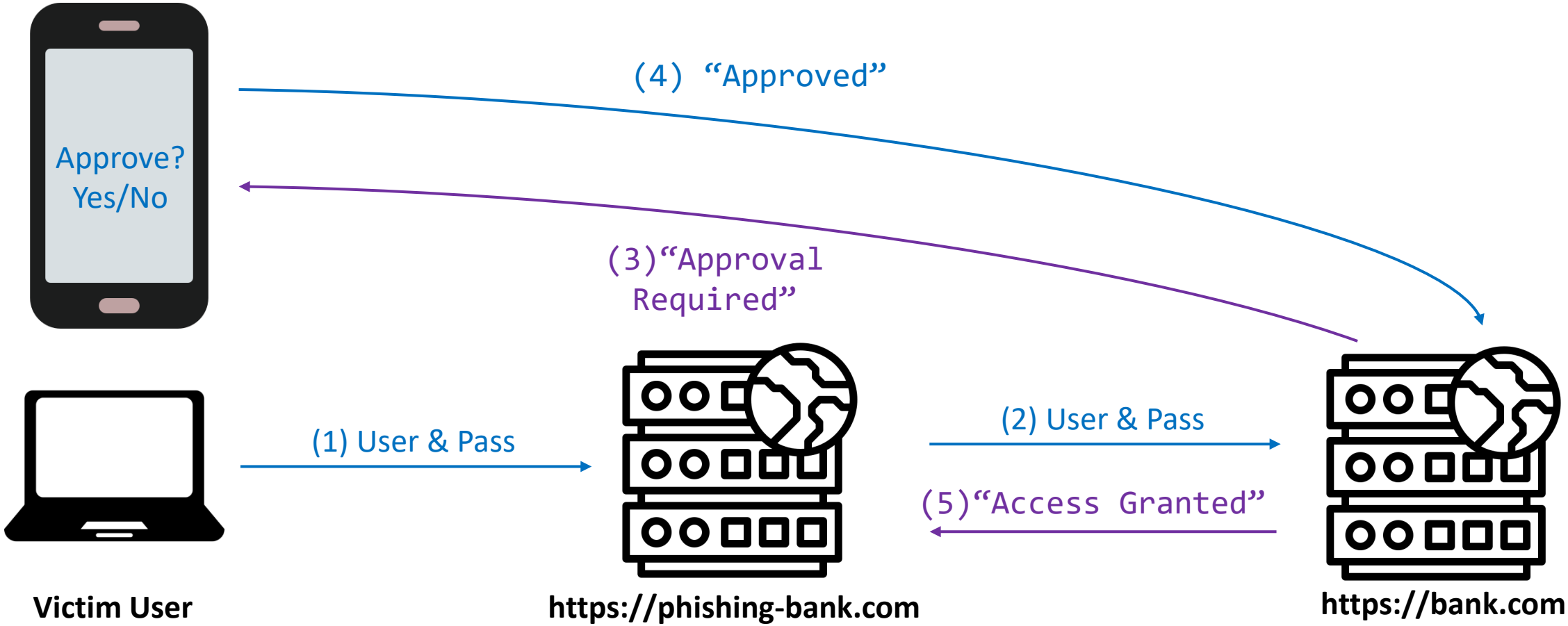
TOTP / HOTP

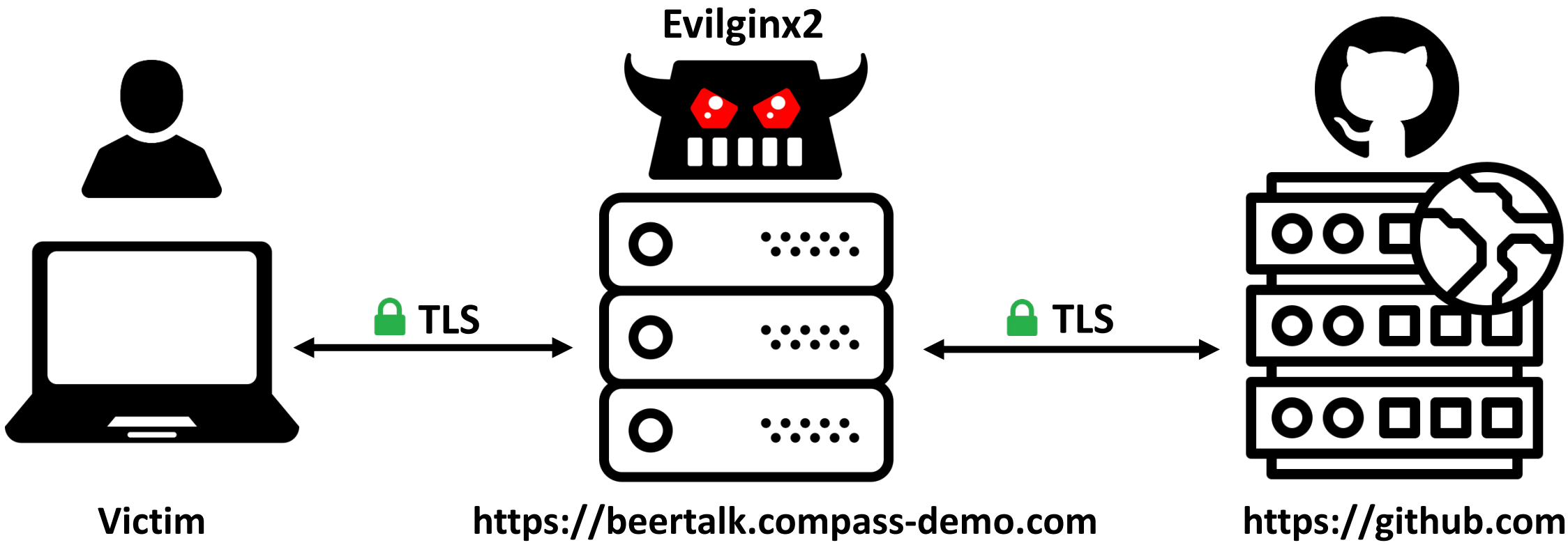
Most second factor mechanisms do not protect against credential phishing

2FA Phishing



HOTP / TOTP Phishing with Push Notification





- gitzhub.com
- githujb.com
- githuʙ.com
- githụb.com

DEMO

Phishing Solutions

User Awareness Campaigns

- Costly, time intensive, and ineffective after a while

Traditional 2FA

- Useful but attackers can bypass it

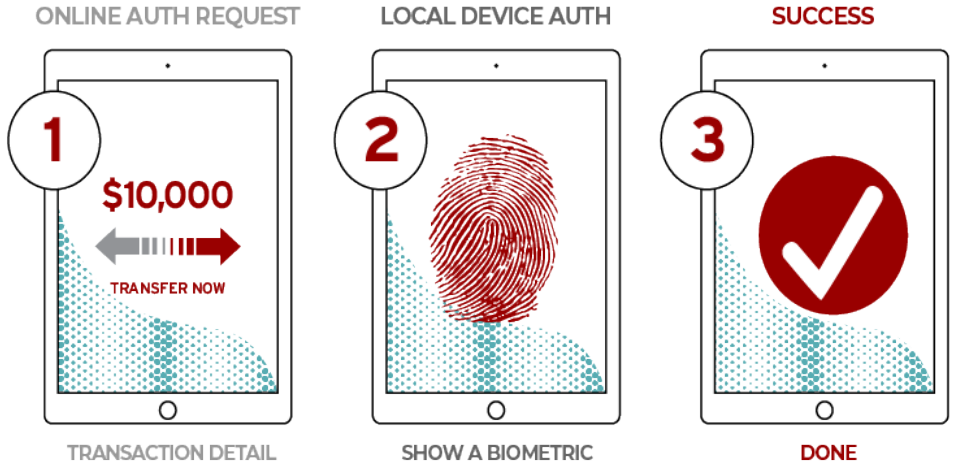
FIDO / FIDO2 Specifications

- A new set of specifications that define ***phishing-resistant*** authentication mechanisms

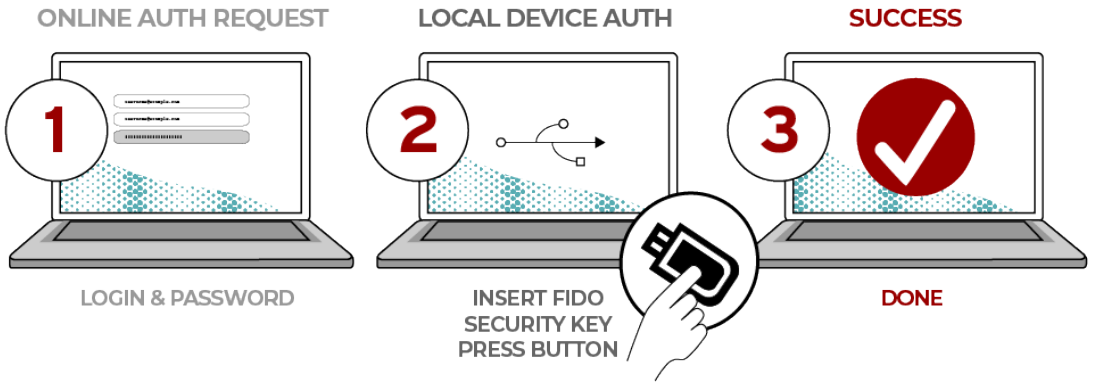
U2F, UAF, CTAP, WebAuthn, FIDO, FIDO2 oh my…

FIDO (Fast Identity Online)

PASSWORDLESS EXPERIENCE (UAF standards)



SECOND FACTOR EXPERIENCE (U2F standards)



The FIDO2 specification replaces FIDO U2F and FIDO UAF

Source: https://fidoalliance.org/specifications/

FIDO2 Building Blocks

FIDO2 Authenticators

https://www.yubico.com/

https://cloud.google.com/titan-security-key/

https://onlykey.io/

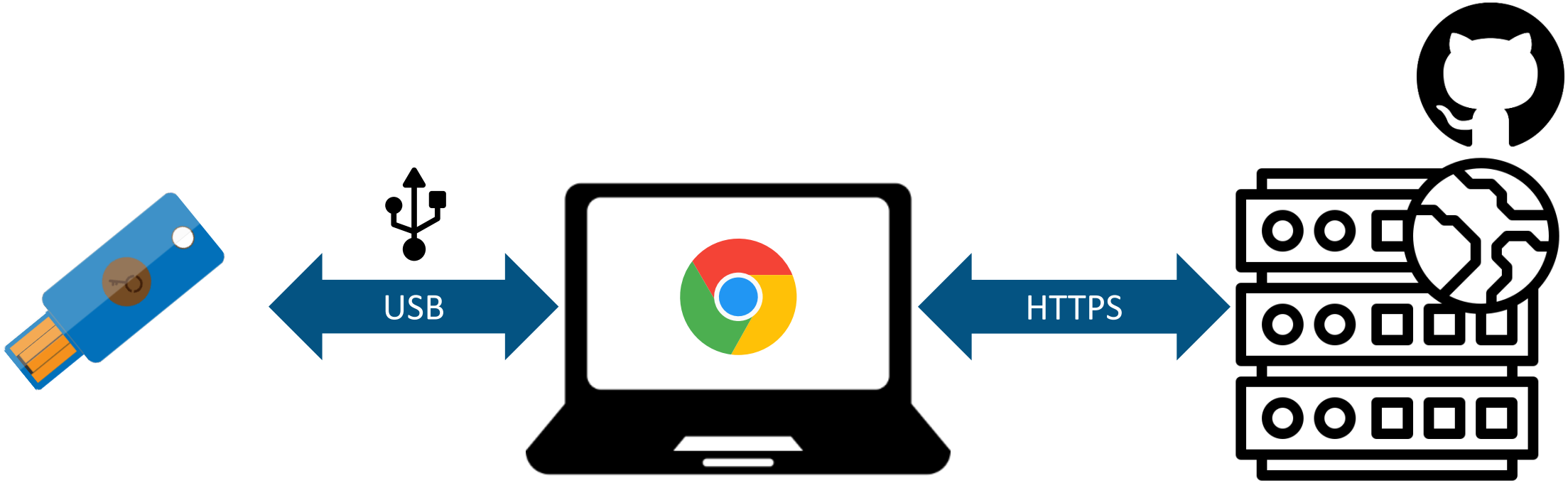
https://solokeys.com/

FIDO2 Clients

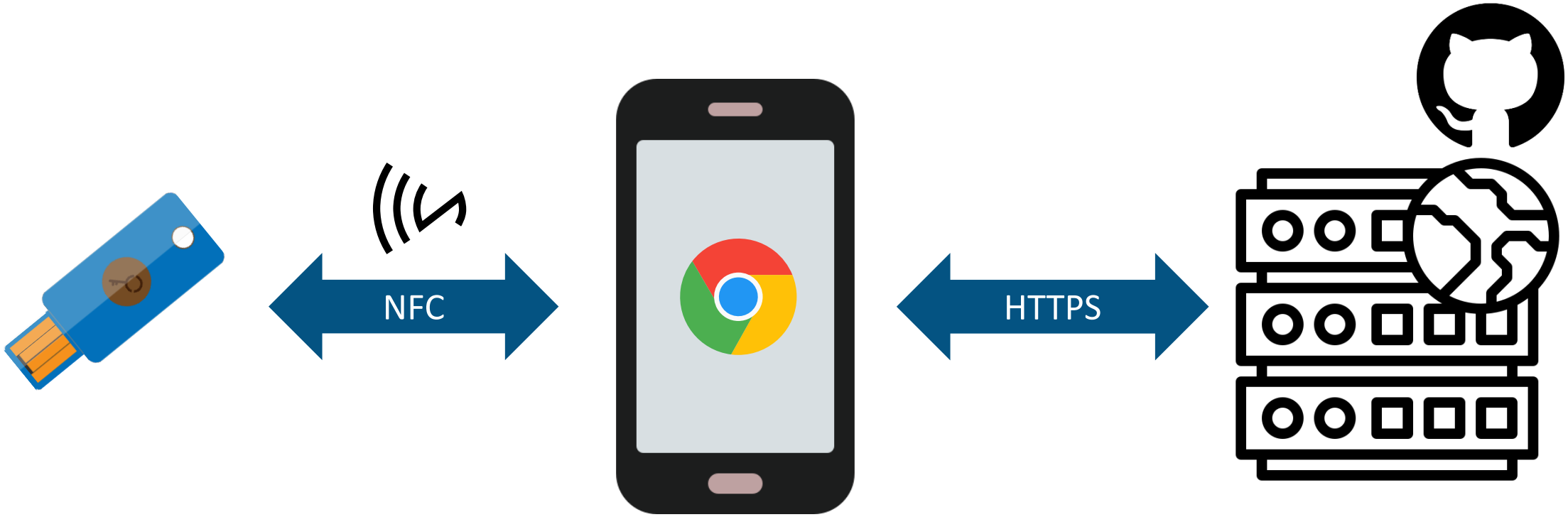
Source: https://fidoalliance.org/fido2/)

FIDO2 Relying Parties

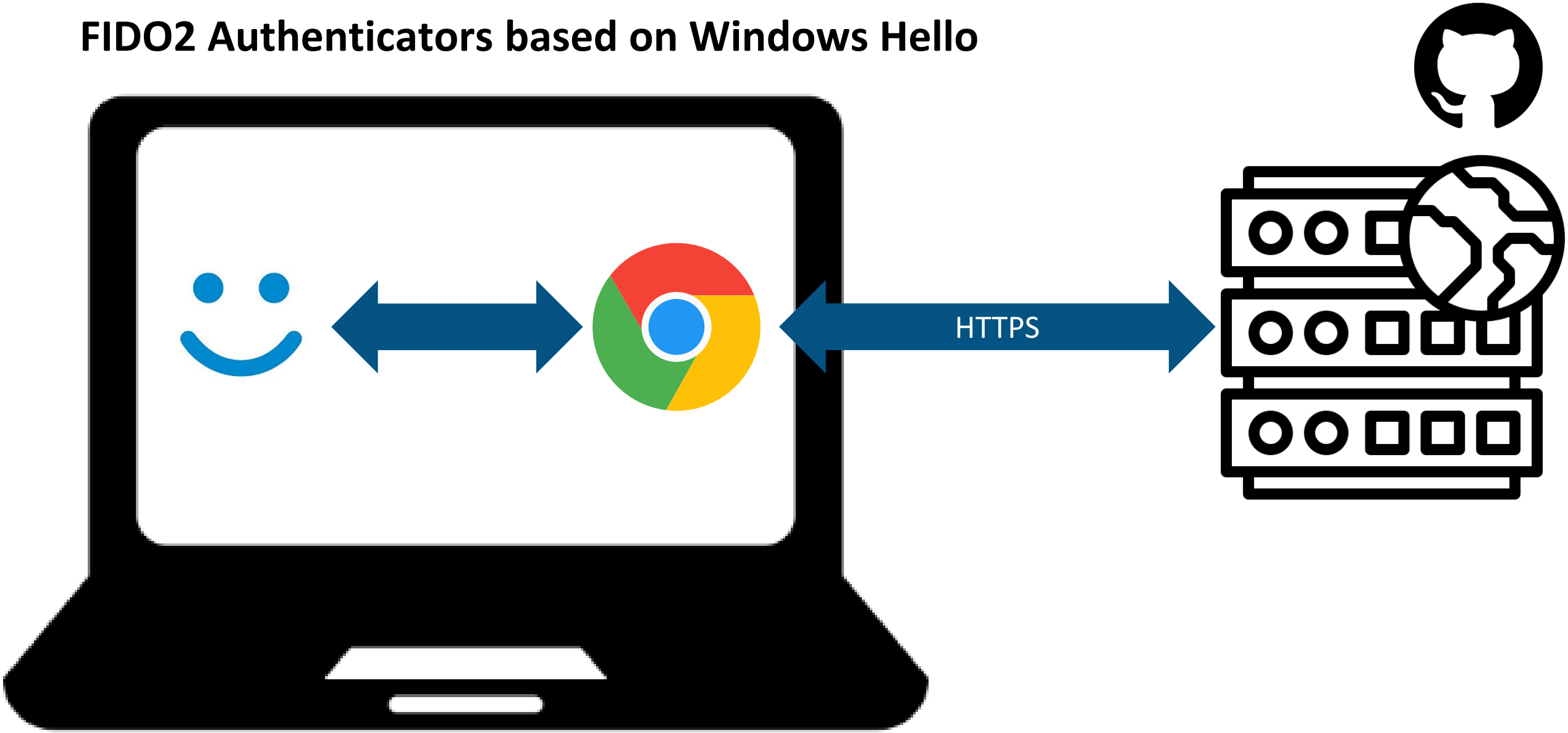
FIDO2 Authenticators based on USB Dongles



FIDO2 Authenticators based on NFC Devices



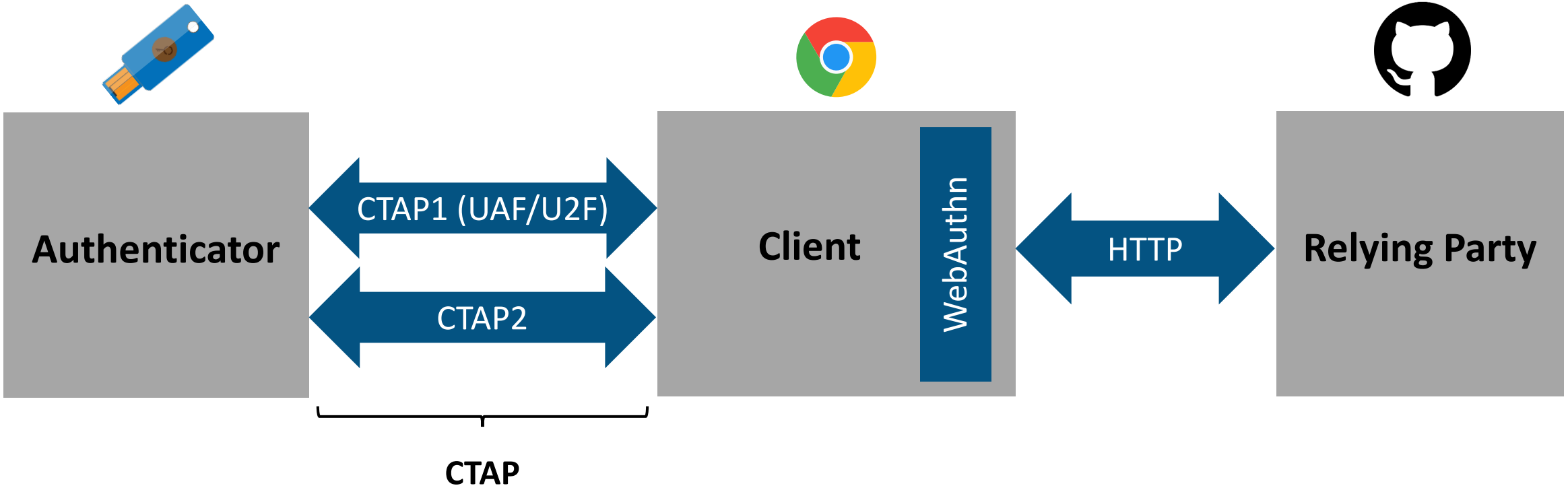
FIDO2 Authenticators based on Windows Hello



FIDO2

Souce: https://fidoalliance.org/fido2/

FIDO2 Building Blocks



Client to Authenticator Protocol (CTAP)

Client To Authenticator Protocol - CTAP

The communication from client to a *roaming* authenticator can use any of the following transport bindings:

- USB Human Interface Device (USB HID)

- Near Field Communication (NFC)

- Bluetooth Smart / Bluetooth Low Energy Technology

Application developers usually need not be concerned with CTAP

WebAuthn

WebAuthentication (WebAuthn)

- Standardized JavaScript Web API for FIDO2 Authentication
- Official web standard since March 2019
- Implemented by browsers and related web platform infrastructure

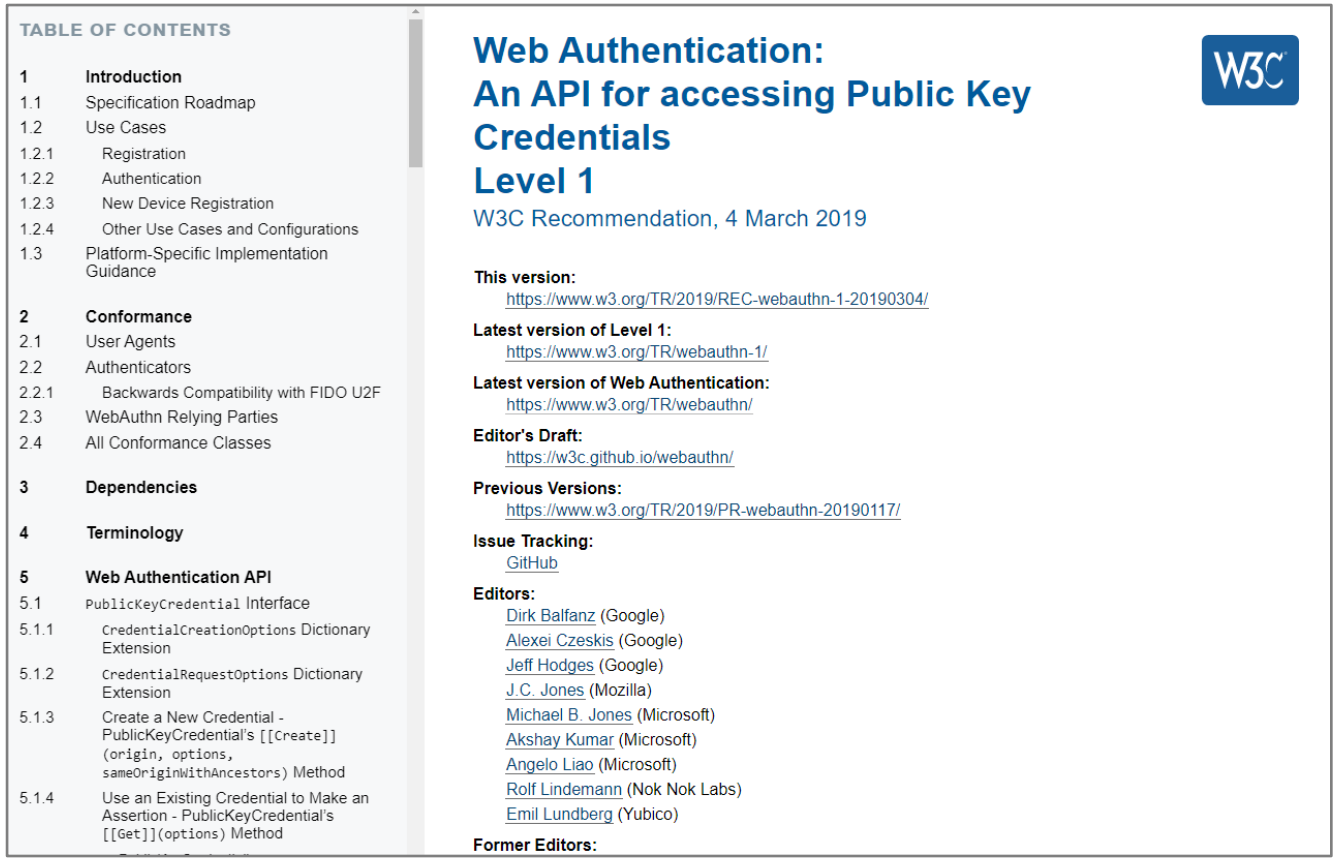


TABLE OF CONTENTS

Web Authentication:
An API for accessing Public Key Credentials
Level 1

W3C Recommendation, 4 March 2019

This version:
https://www.w3.org/TR/2019/REC-webauthn-1-20190304/

Latest version of Level 1:
https://www.w3.org/TR/webauthn-1/

Latest version of Web Authentication:
https://www.w3.org/TR/webauthn/

Editor's Draft:
https://w3c.github.io/webauthn/

Previous Versions:
https://www.w3.org/TR/2019/PR-webauthn-20190117/

Issue Tracking:
GitHub

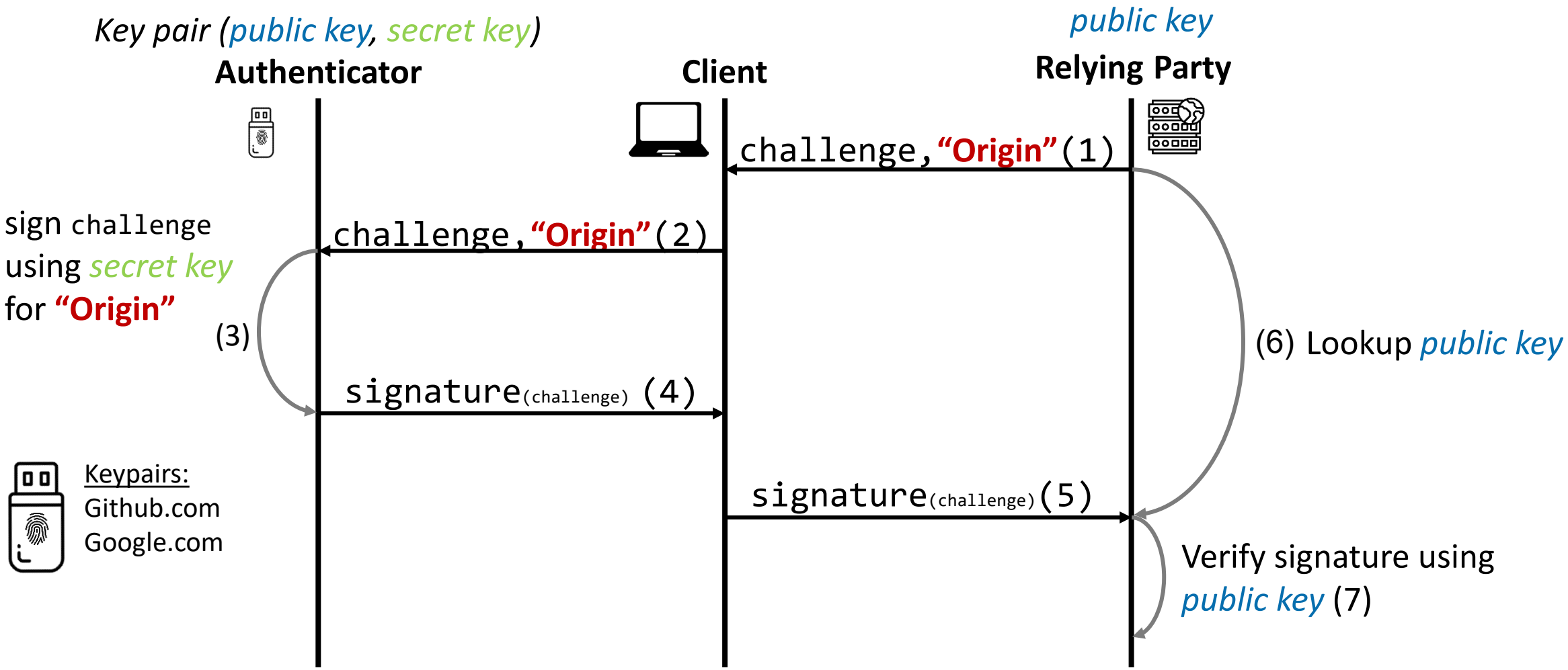
Editors:
Dirk Balfanz (Google)
Alexei Czeskis (Google)
Jeff Hodges (Google)
J.C. Jones (Mozilla)
Michael B. Jones (Microsoft)
Akshay Kumar (Microsoft)
Angelo Liao (Microsoft)
Rolf Lindemann (Nok Nok Labs)
Emil Lundberg (Yubico)

Former Editors:

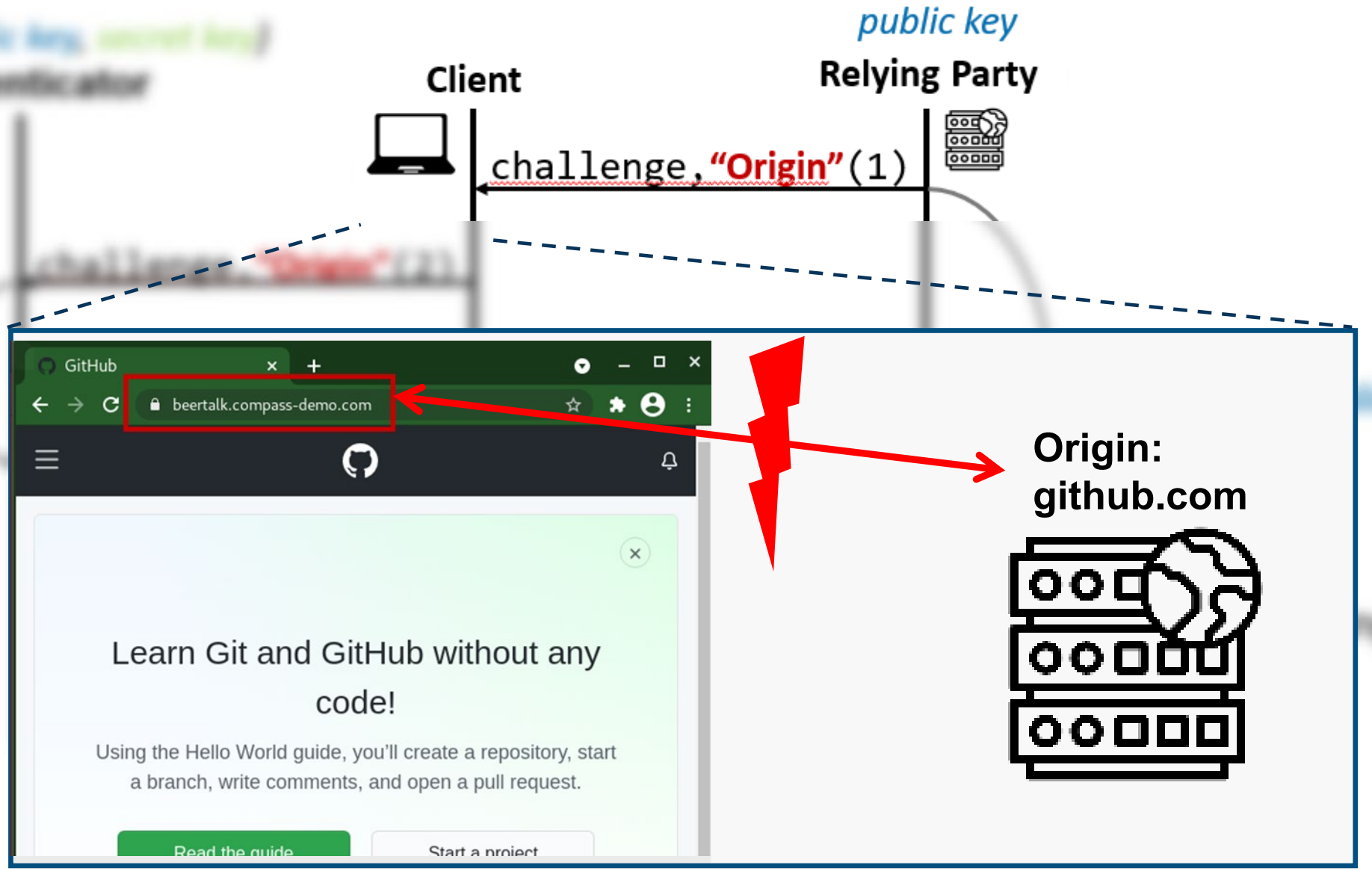
Source: https://www.w3.org/TR/webauthn/

Authentication Protocol

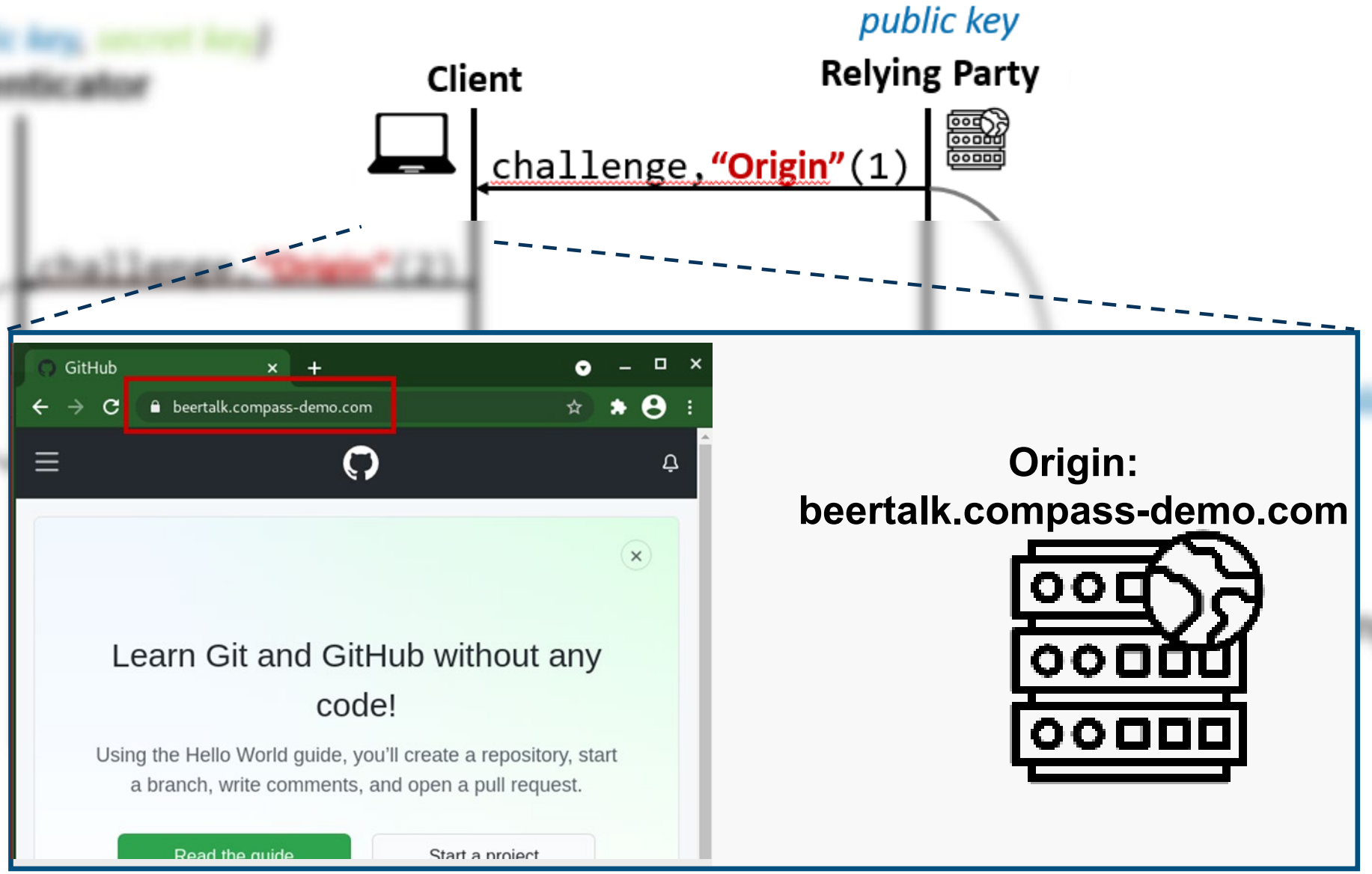
FIDO2 Challenge Response Protocol



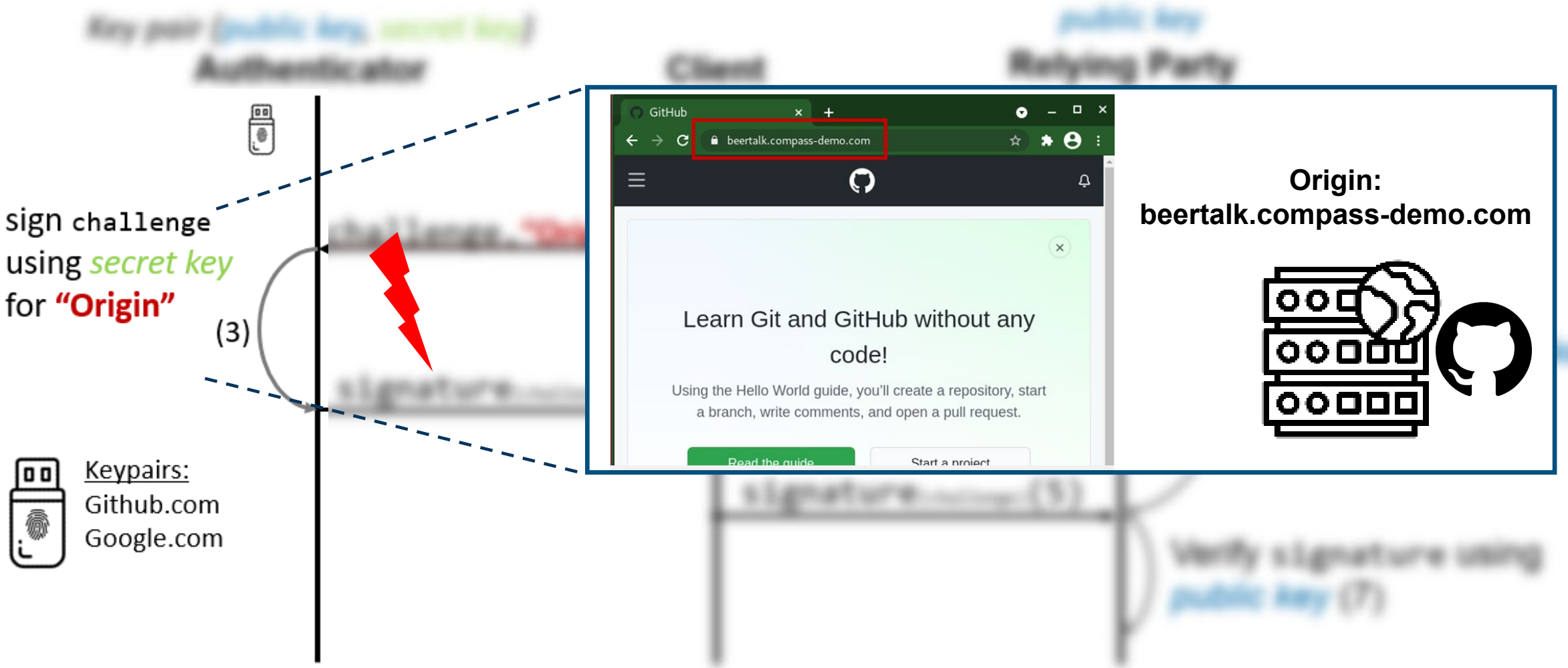
Phishing Protection (1) - Github



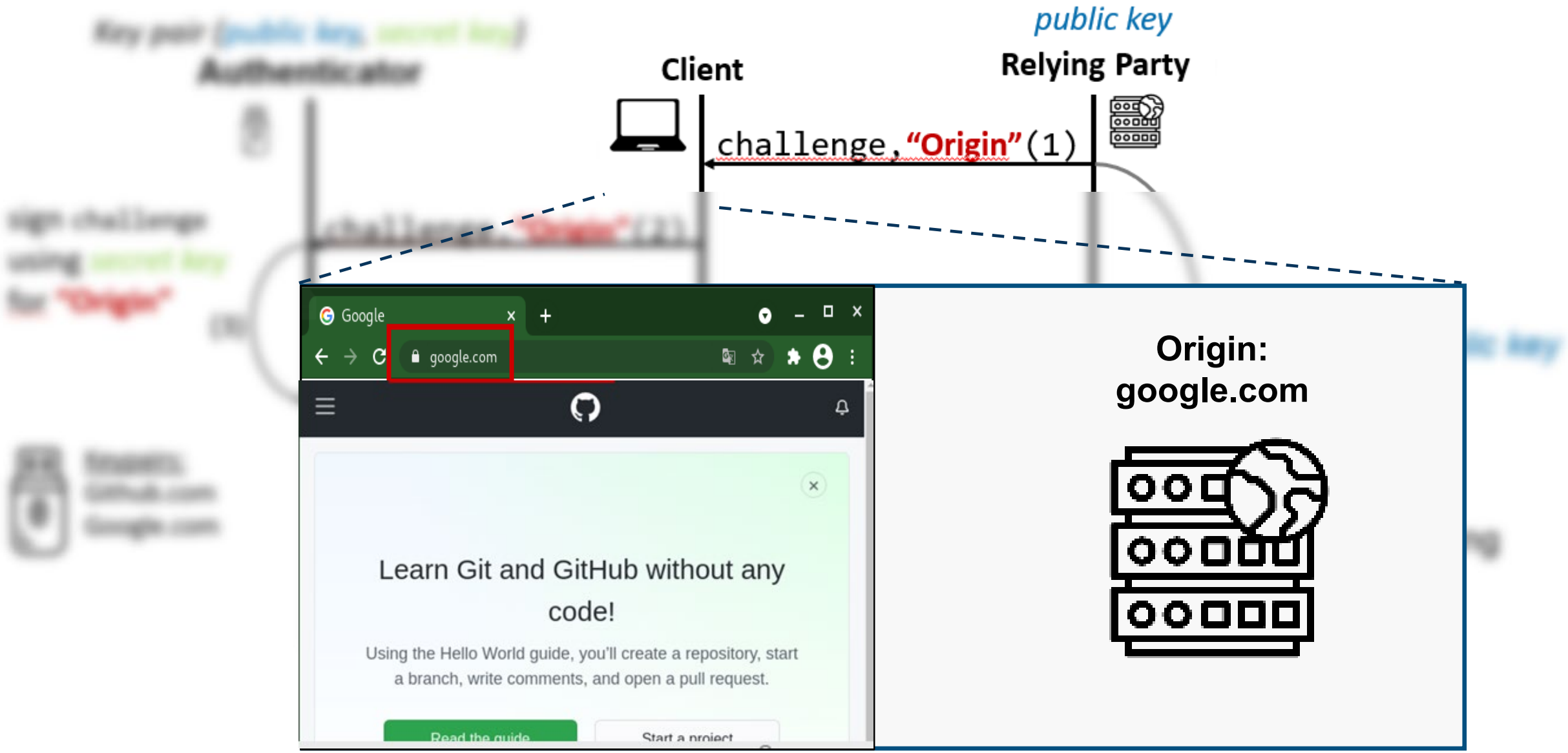
Phishing Protection (2) - Github



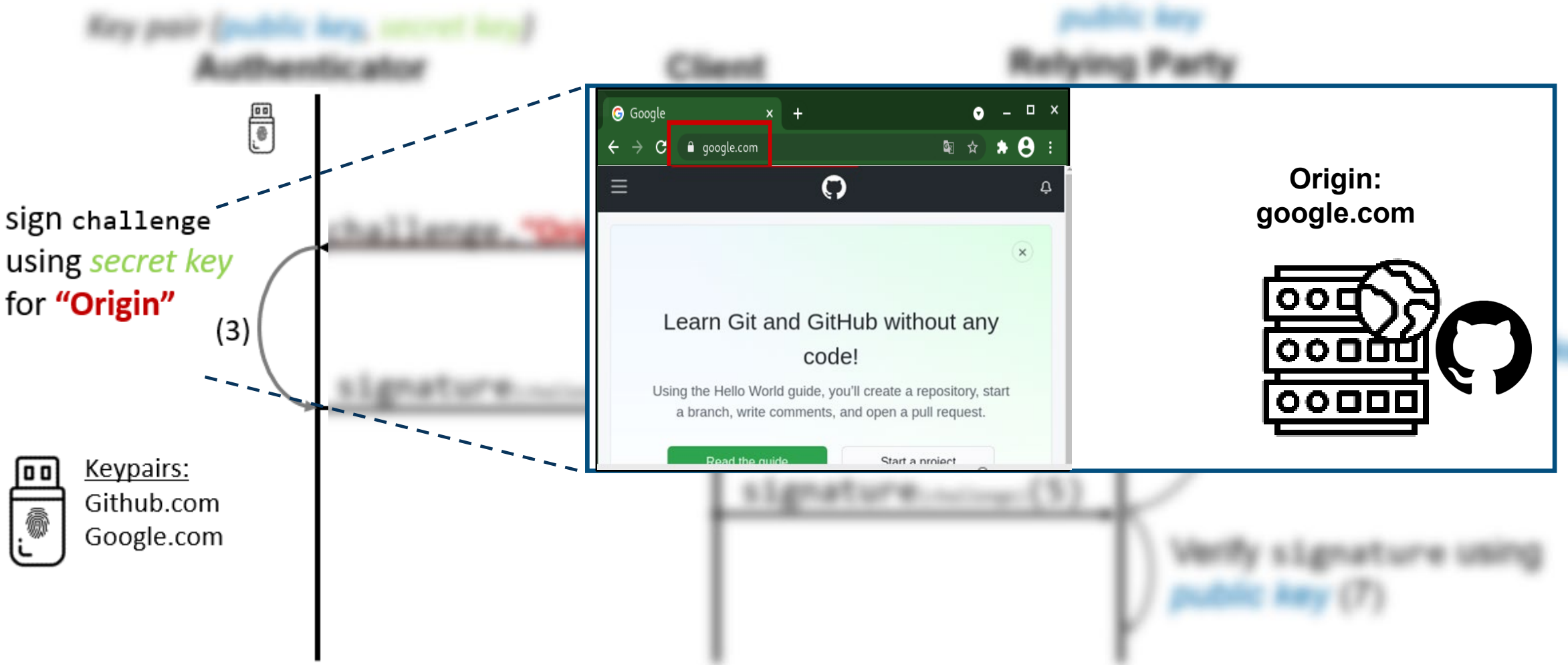
Phishing Protection (2) - Github



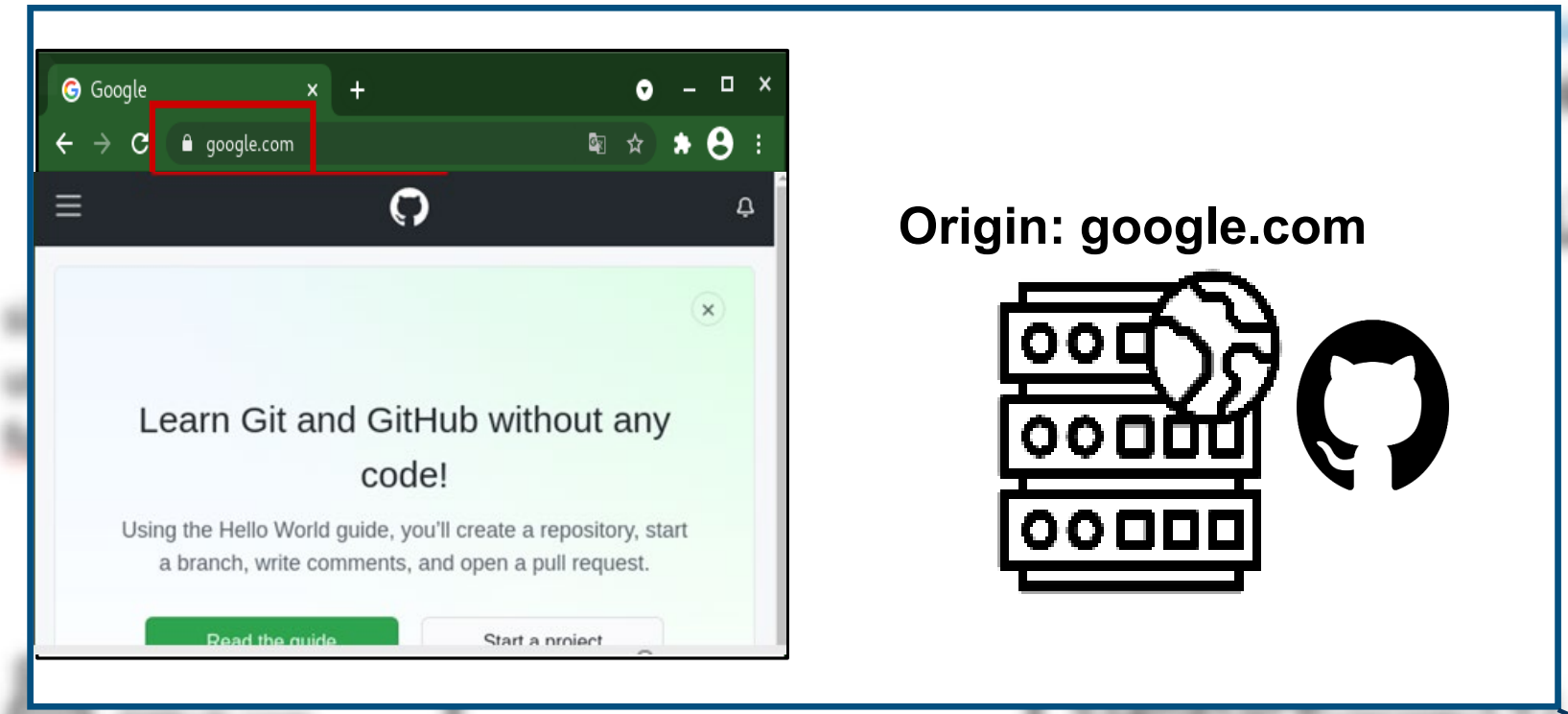
Phishing Protection (3) - Github



Phishing Protection (3) - Github



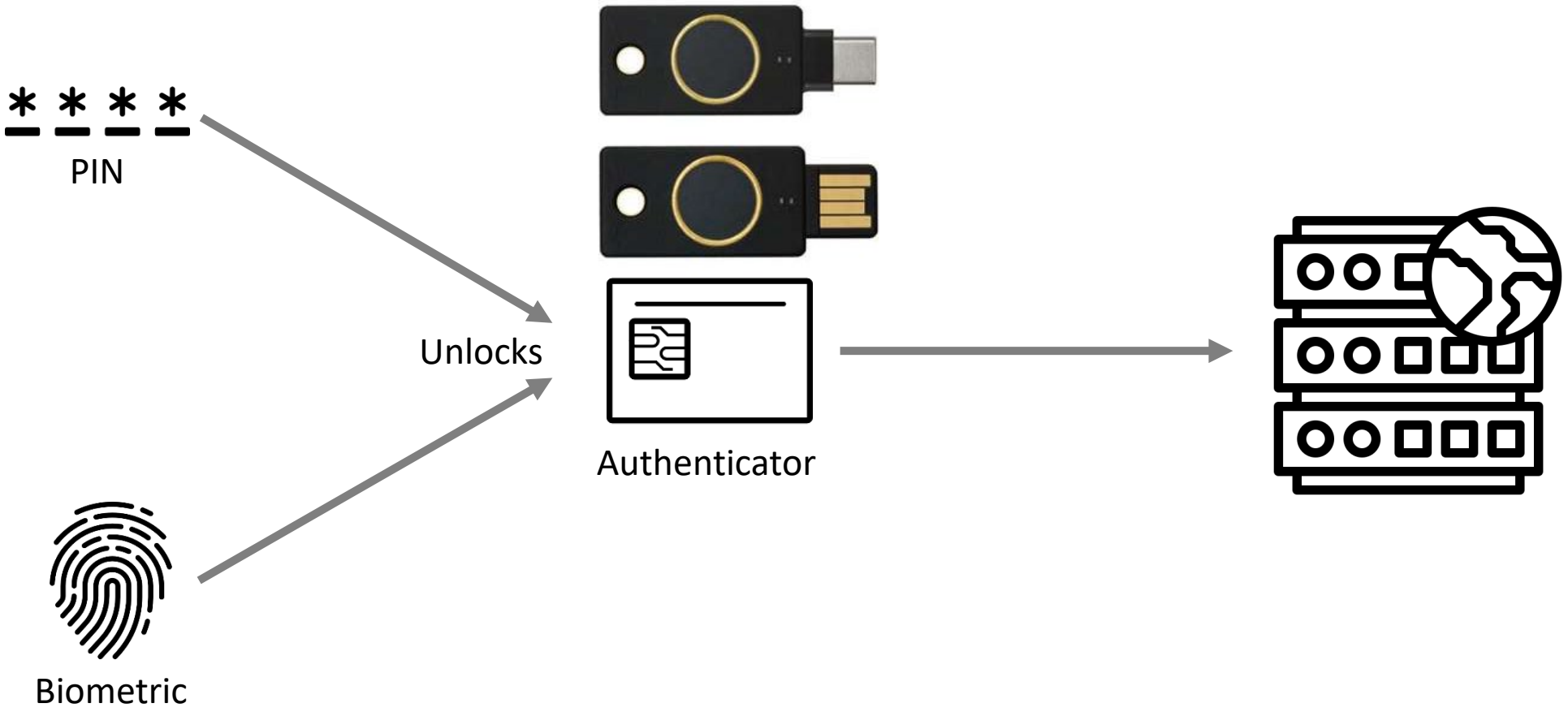
Phishing Protection (3) - Github



Verify `signature` using *public key* (7)

Password-Less Authentication

Password-Less Authentication



Source: https://pages.yubico.com/YubiKey-Bio-Updates-de.html

Conclusion

Conclusion

Traditional authentication mechanisms are not sufficient anymore

- Inconvenient
- Not resistant to phishing

FIDO2

- is a phishing-resistant authentication protocol
- is simple to use (also for non-technical people)
- has strong platform & industry support

Additional Material

References

Demo Pages

https://webauthn.io
https://demo.yubico.com/webauthn
https://webauthn.me

Developer Information

https://developers.yubico.com/WebAuthn/WebAuthn_Developer_Guide/

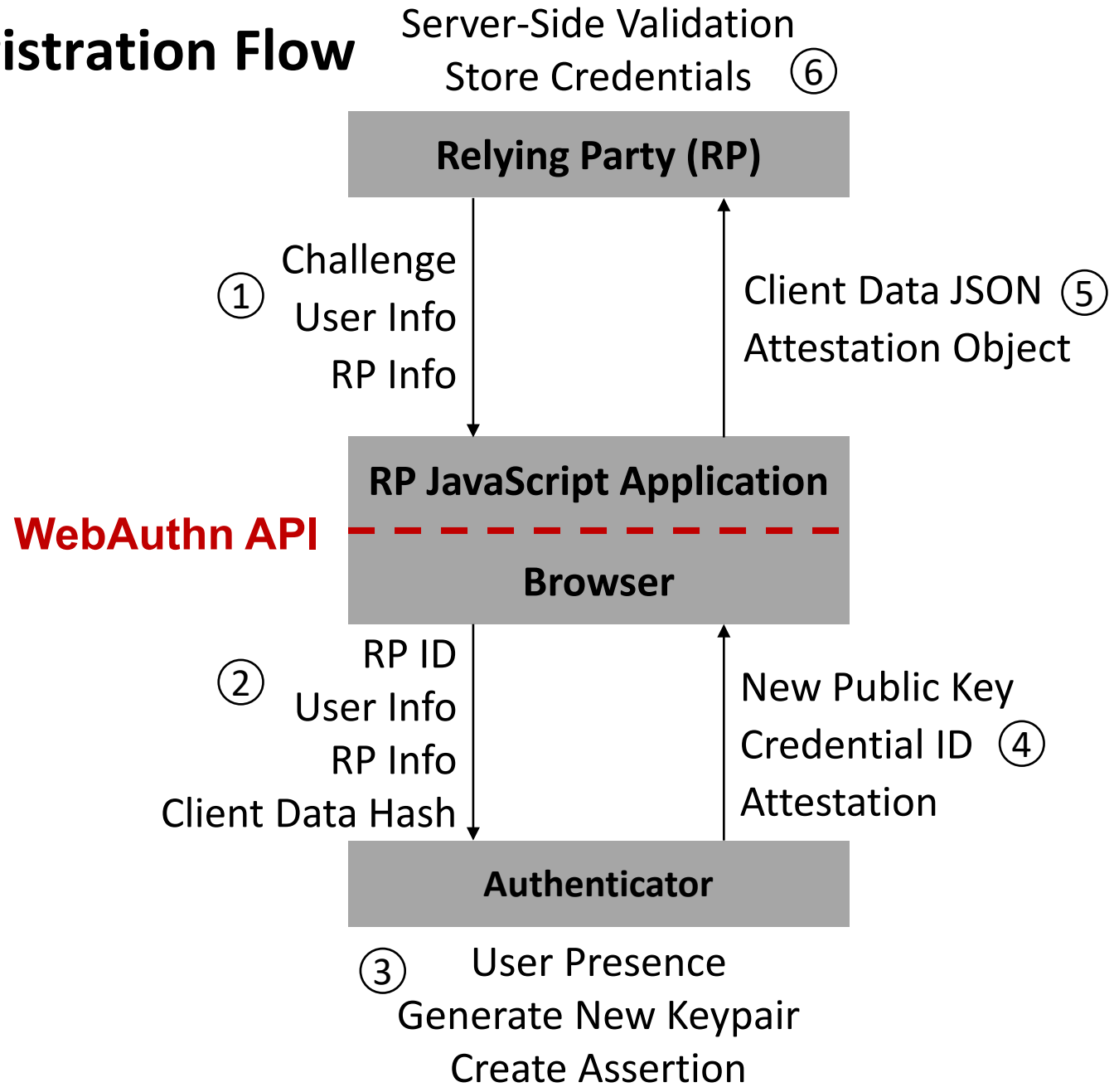
Specifications

https://www.w3.org/TR/webauthn/
https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-client-to-authenticator-protocol-v2.0-id-20180227.html

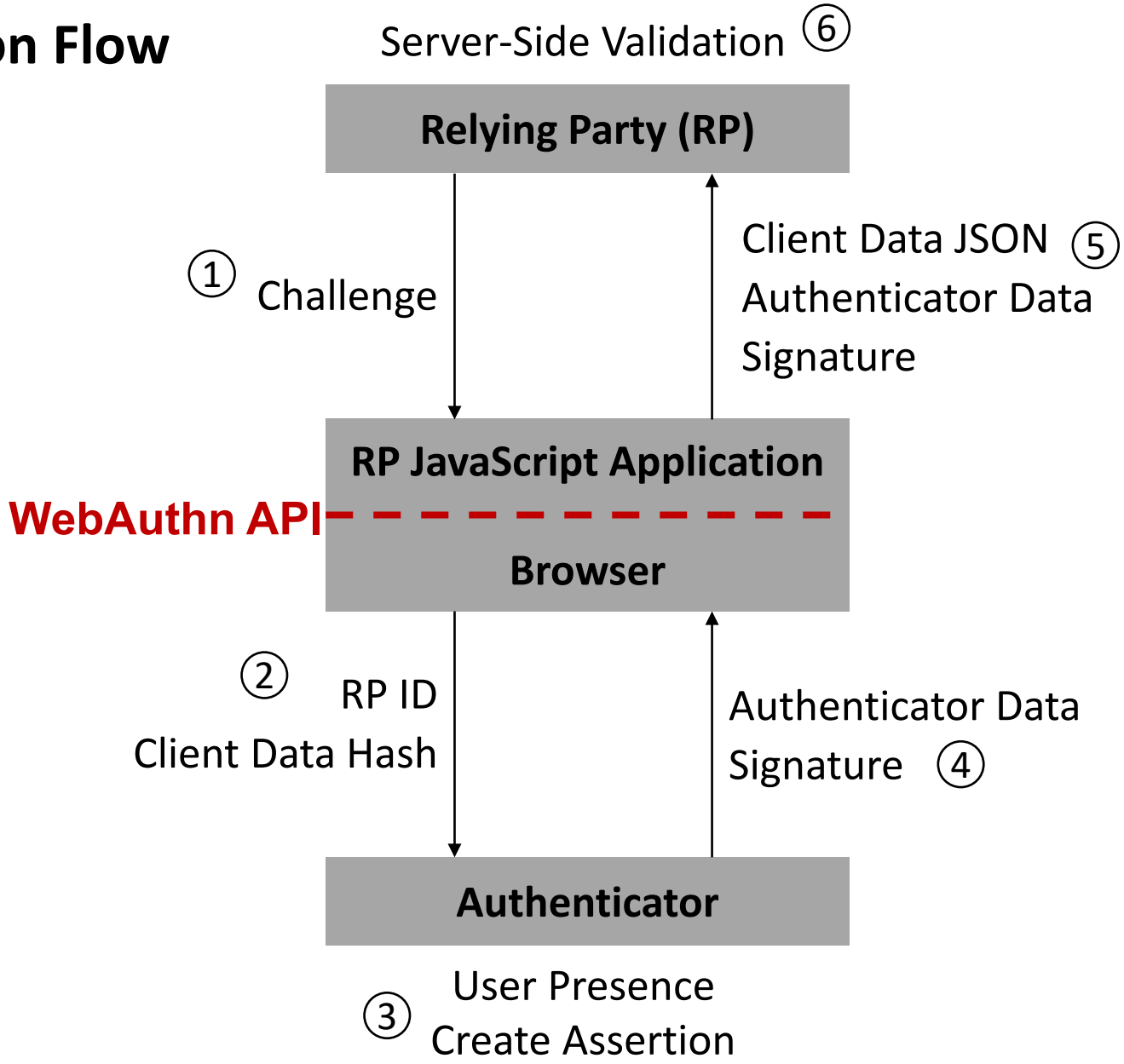
Talks

https://www.youtube.com/watch?v=J53Ya7E5HGQ

Credential Registration Flow



Authentication Flow



Scoped Credentials - Example

RP with origin https://login.example.com:1337

RP ID	Validity
login.example.com	**Valid**
example.com	**Valid**
foo.example.com	**Not valid**
m.login.example.com	**Not valid**
.com	**Not valid**

Credential Registration – JS API

```
var publicKey = {
    challenge: {challenge},
    rp: {
        name: "Example server",
        id: "example.com"     // optional
    },
    user: {
        id: {user_id},
        name: "user@example.com",
        displayName: "A User",
    },
    pubKeyCredParams: [{
      type: "public-key",
      alg: -7                 // ES256
    }],
    excludeCredentials: [],
    attestation: "direct",
    timeout: 60000,
    extensions: {"loc": true}
};
```

```
if (!window.PublicKeyCredential)
{ /* Platform not capable. Handle error. */ }

navigator.credentials.create({ publicKey })
  .then(function (attestation) {
    // Send new credential info to server
    // for verification and registration.
  }).catch(function (err) {
    // No acceptable authenticator or user
    // refused consent. Handle appropriately.
  });
```

Authentication – WebAuthn JavaScript API

```
if (!window.PublicKeyCredential) { /* Platform not capable. Handle error. */ }
navigator.credentials.get({
    publicKey: {
        rpId: document.domain,
        challenge: {challenge}, // The challenge must be produced by the server
        allowCredentials: [
            {
                type: 'public-key',
                id: {credential_id} // The credential_id may be provided by the server
            }
        ],
        timeout: 60000
     }
}).then(function (assertion) {
    // Send signed challenge and credential info to server for verification and registration.
}).catch(function (err) {
    // No acceptable authenticator or user refused consent. Handle appropriately.
});
```