U2F Tutorial: Authentication Tokens for Enterprise and Consumers

MIGUEL RODRIGUEZ, EXPERT SERVICES, YUBICO

yubico

Agenda

- About FIDO U2F and Security Keys
- Exploring FIDO U2F
- FIDO U2F Demos
- Integrating FIDO U2F

© 2018 Yubico

About FIDO U2F and Security Keys

© 2017 Yubico

yubico

FIDO U2F / Security Keys

Co-developed by Yubico and Google, completed and published as a standard by the FIDO Alliance in 2014

Google salesforce facebook Dropbox

One Authenticator **Any** # of Services **No** Shared Secrets

© 2017 Yubico

yubico

U2F Value

Secure
Eliminates Phishing & Prevents MitM

Simple
Insert and Touch Gold Contact

Scalable
Use the Same Key for Unlimited Services

Private
No Shared Secrets Between Services
No Shared Secrets with Yubico

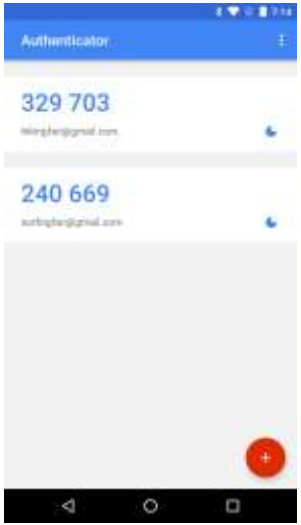
© 2018 Yubico

Google Case Study

- **YubiKey mandatory for all Google staff and contractors**
- **Support for Google end users**

U2F YubiKey
vs Google Authenticator

- 4x faster to login
- Significant fraud reduction
- Support reduced by 92%



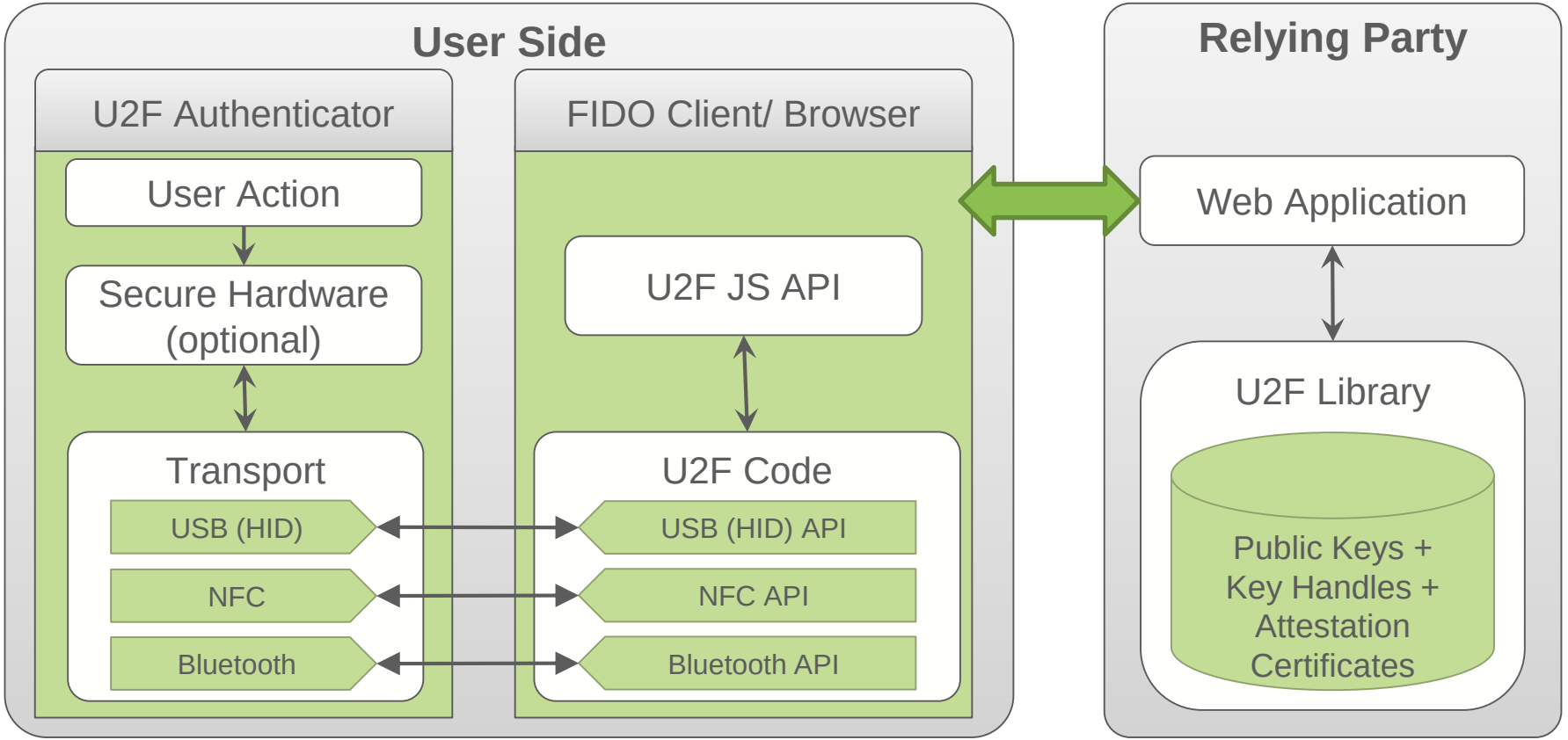
Exploring FIDO U2F

Protocol flows

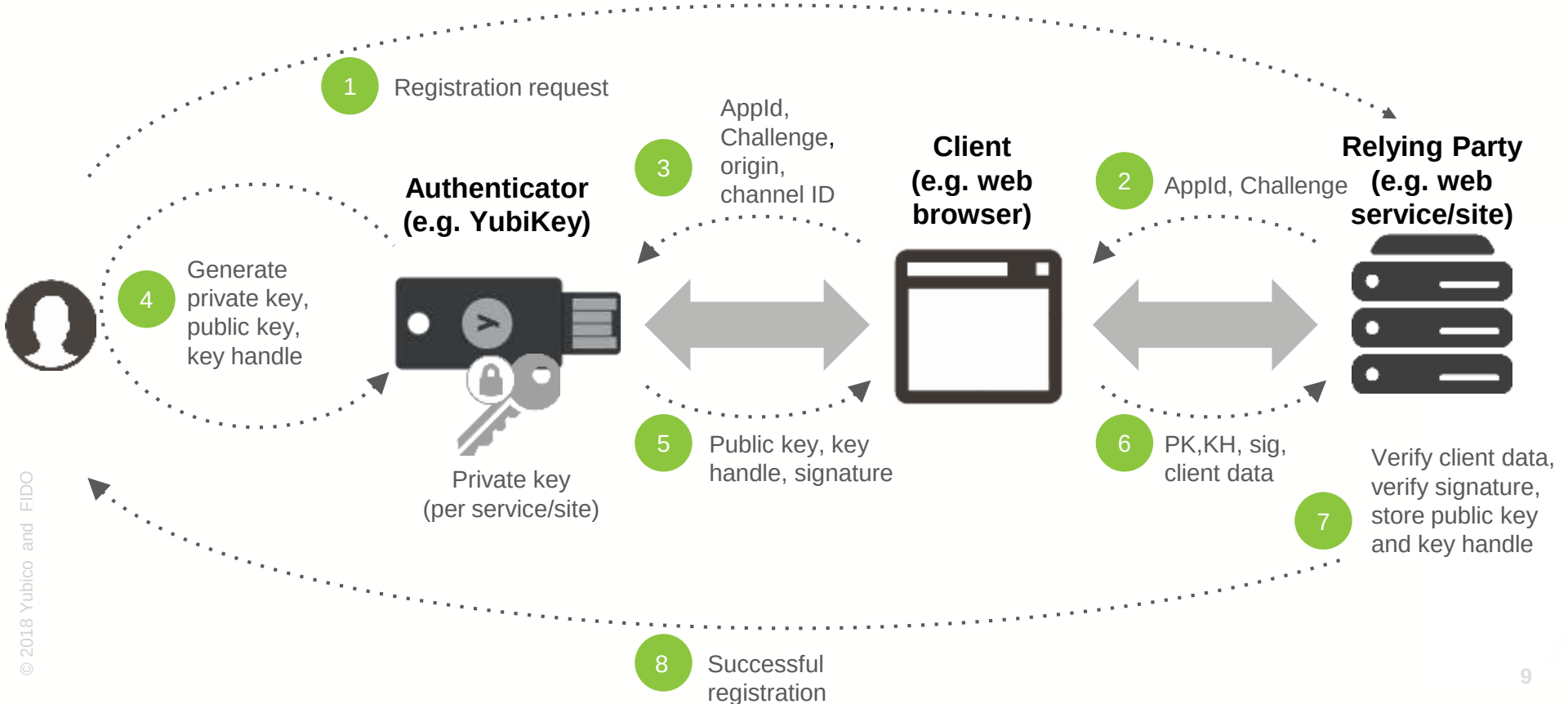
© 2017 Yubico

yubico

U2F Entities

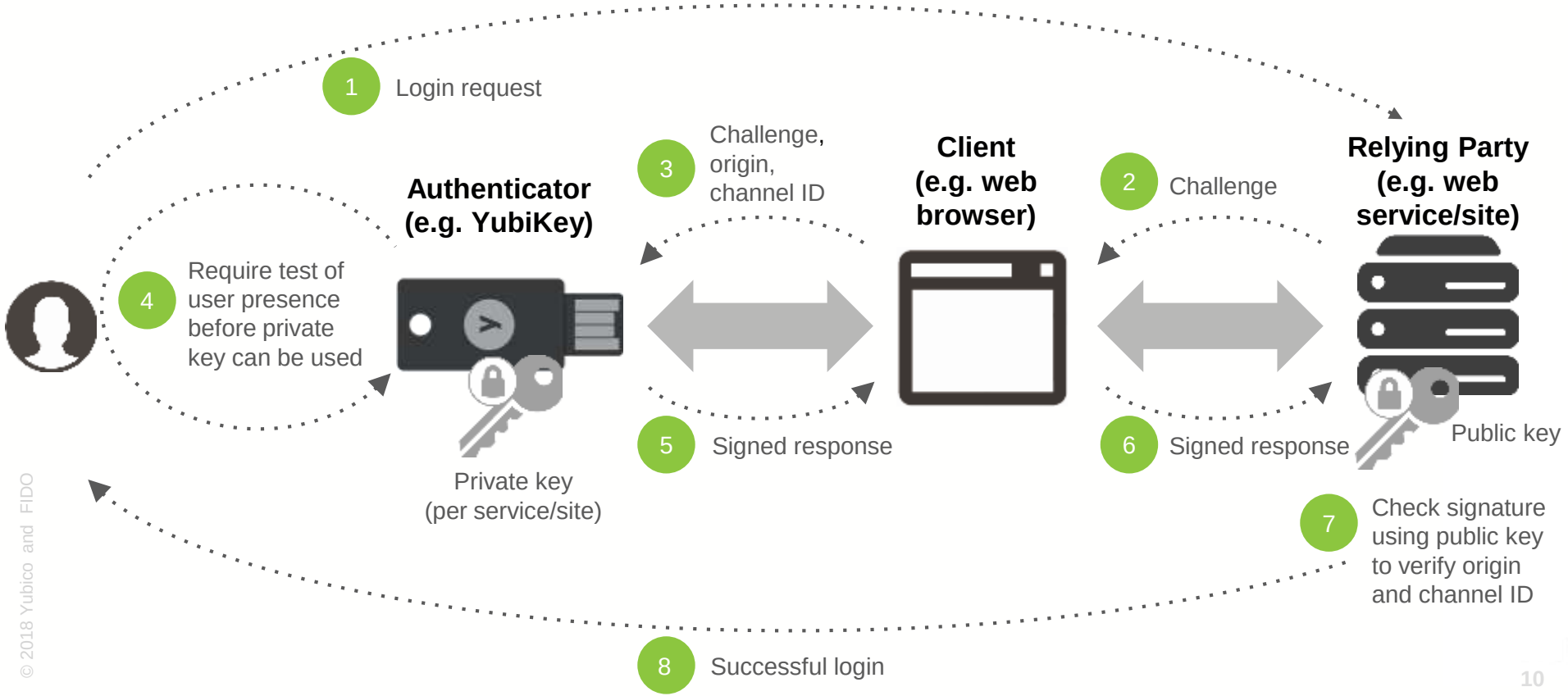


How FIDO Registration Works



© 2018 Yubico and FIDO

How FIDO Authentication Works



© 2018 Yubico and FIDO

Demo 1

FIDO U2F and Firefox

© 2017 Yubico

yubico

Demo 2

FIDO U2F and Android

© 2017 Yubico

yubico

Integrating FIDO U2F

© 2017 Yubico

yubico

Integrating FIDO U2F- client side

- Typically a U2F enabled browser
- Use the u2f-api.js library (high level API)
  - Simplest way to get started
- Call two functions
  - u2f.register
  - u2f.sign
- If not a browser use a U2F host library (Python, C)

© 2018 Yubico

yubico

Integrating FIDO U2F- client side

Errors

1. **OTHER_ERROR**: An error otherwise not enumerated here.
2. **BAD_REQUEST**: One of the following reasons:
   - The visited URL doesn't match the App ID.
   - The App ID does not conform with the rules for App ID's.
   - The U2F API is called with bad parameters (e.g. calling u2f.register with the parameters in the wrong order).
3. **CONFIGURATION_UNSUPPORTED**: Client configuration is not supported.
4. **DEVICE_INELIGIBLE**: The presented device is not eligible for this request. For a registration request this may mean that the token is already registered, and for a sign request it may mean that the token does not know the presented key handle.
5. **TIMEOUT**: Timeout reached before request could be satisfied.

© 2018 Yubico

Integrating FIDO U2F- server side

- Some servers are tightly coupled with the web application
- Others are decoupled (U2F as REST or SOAP API)
- Pros and cons dependent on the web application
- Messages between the U2F Authenticator and the U2F Server are standardized
- Server implementation based on typical considerations
  - Cost
  - Ease-of-integration
  - Support
  - Scalability
  - Etc.

© 2018 Yubico

Integrating FIDO U2F- server side

Tools for implementation (open source)

- Server libraries, in several programming languages:
  - C (libu2f-server)
  - Java (java-u2flib-server)
  - PHP (php-u2flib-server
  - Python (python-u2flib-server)
- Stand alone server (demo only):
  - Yubico U2FVAL server (REST, Python)

© 2018 Yubico

Integrating FIDO U2F- server side

- Server-side U2F library has 4 basic functions:
  - Start registration
  - Finish registration
  - Start authentication
  - Finish authentication

© 2018 Yubico

yubico

Integrating FIDO U2F- server side

Registration

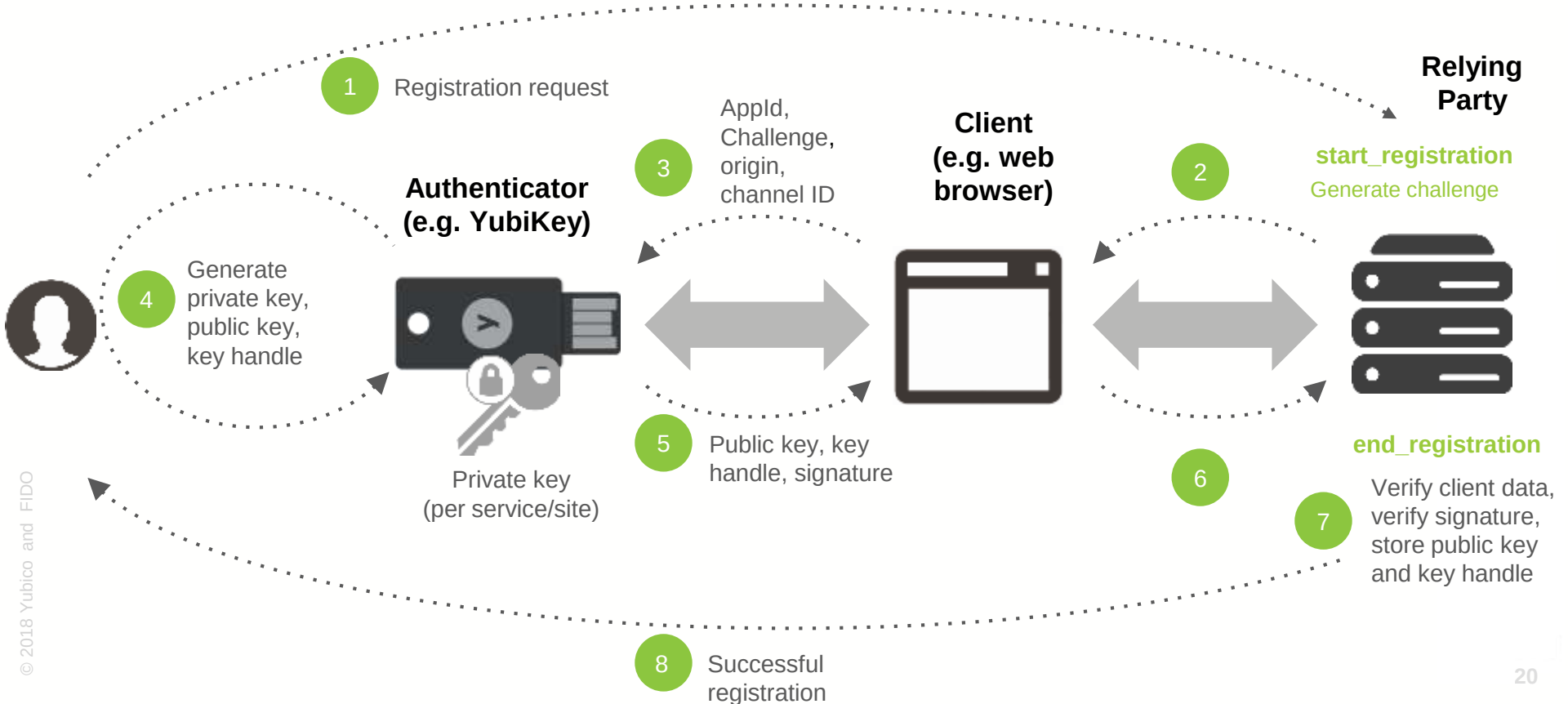
```
# handles HTTPS requests to /start_registration
def start_registration(username):
  challenge = u2f_lib.start_registration(APP_ID)
  challenge_store.set(username, challenge)
  return challenge
```

```
# handles HTTPS requests to /finish_registration
def finish_registration(username, device_response):
  challenge = challenge_store.pop(username)
  registered_device = u2f_lib.finish_registration(challenge, device_response)
  device_store.set(username, registered_device)
  return "Success!"
```

© 2018 Yubico

yubico

Integrating FIDO U2F - server side



© 2018 Yubico and FIDO

Integrating FIDO U2F- server side

Authentication

```
# handles HTTPS requests to /start_authentication
def start_authentication(username, password):
  verify_user_pass(username, password)
  registered_devices = device_store.get(username)
  challenge = u2f_lib.start_authentication(registered_devices, APP_ID)
  challenge_store.set(username, challenge)
  return challenge
```

```
# handles HTTPS requests to /finish_authentication
def finish_authentication(username, password, device_response):
  challenge = challenge_store.pop(username)
  u2f_lib.finish_authentication(challenge, device_response, registered_devices)
  return "Success!"
```

© 2018 Yubico

yubico

Integrating FIDO U2F- server side

Authentication: create challenges for all registered devices

```
private abstract Iterable<DeviceRegistration> getRegistrations(String username);

@GET
public View startAuthentication(String username) throws NoEligibleDevicesException {

    // Generate a challenge for each U2F device that this user has registered
    SignRequestData requestData
        = u2f.startSignature(SERVER_ADDRESS, getRegistrations(username));

    // Store the challenges for future reference
    requestStorage.put(requestData.getRequestId(), requestData.toJson());

    // Return an HTML page containing the challenges
    return new AuthenticationView(requestData.toJson(), username);
}
```

© 2018 Yubico

yubico

Integrating FIDO U2F- server side

Authentication: successful for one registered device

```
@POST
public String finishAuthentication(SignResponse response, String username) throws
        DeviceCompromisedException {

    // Get the challenges that we stored when starting the authentication
    SignRequestData signRequest
        = requestStorage.remove(response.getRequestId());

    // Verify the that the given response is valid for one of the registered devices
    u2f.finishSignature(signRequest,
                        response,
                        getRegistrations(username));

    return "Successfully authenticated!";
}
```

© 2018 Yubico

FIDO U2F - Learn More

Get Started

- Read the specifications: fidoalliance.org/specifications/overview/
- Go through a MiniTwit U2F tutorial: MiniTwit training video

Implement

- Google reference code: github.com/google/u2f-ref-code
- Build your own U2F server: dev.yubi.co/U2F/libraries
- Use Yubico standalone U2F server: dev.yubi.co/u2fval

Test

- Yubico U2F demo server: demo.yubico.com/u2f
- Google U2F demo server: u2fdemo.appspot.com

yubico

© 2017 Yubico