SEPTEMBER 15 -16, 2022 - LILLE, FRANCE & ONLINE

API PLATFORM CONFERENCE

Slides template

01 Password Issues

02 What about Api Platform?

03 How to solve password issues?

04 How Webauthn Works?

05 Application Integration?

API PLATFORM CONFERENCE

About me

Florent Morselli

aka Spomky

- ✓ **Web developer**, self-taught
- ✓ Actively maintain libraries **OTP, JWT, CBOR**...
- ✓ OAuth2.x, OIDC and **Webauthn enthusiast**

API PLATFORM CONFERENCE

01 – Password Issues

What is a good password?

Password Issues

Most of web applications rely on **username/password** authentication

Password Issues

Even your API: tokens are usually issued after **username/password** authentication and consent

Password Issues

But **passwords** are the root cause of a lot of **troubles**

Password Issues

Good passwords **do not exist**

Password Issues

A password is a **shared secret difficult to choose** and very **difficult to protect**!

Password Issues

- ✓ uJ72S2!5jC&AET2*
- ✓ armful-jalapeno-onlooker

Password Issues

It is not only shared

- with the **browser**
- during the **transport**
- with the **recipient**

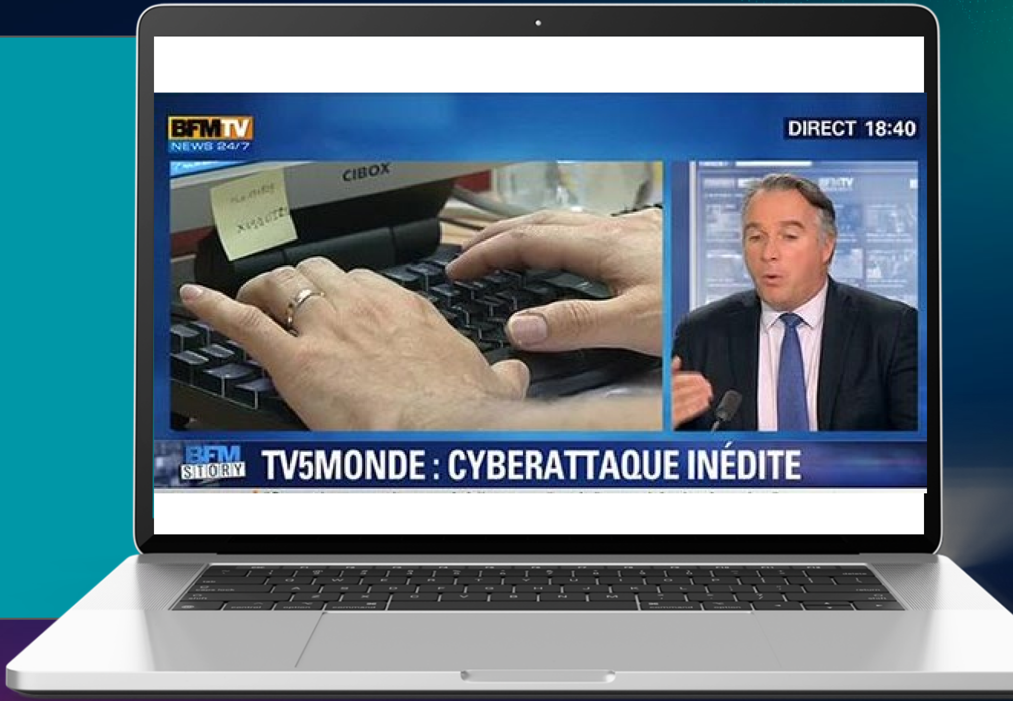
Password Issues

It is also shared

- with **third parties**
- With **colleagues**, **family** and **friends**
- within **emails**, **SMS** and more…

Password Issues

- ✓ This picture was made without rigging in 2015 😱



Password Issues

Too much questions for both users and developers

API PLATFORM CONFERENCE

Password Issues

What **size**?

What **characters**?

Storage?

Hashing function?

Password Issues

Bruteforce protection?

Keyloggers detection?

Expiration policy?

Ban sharing?

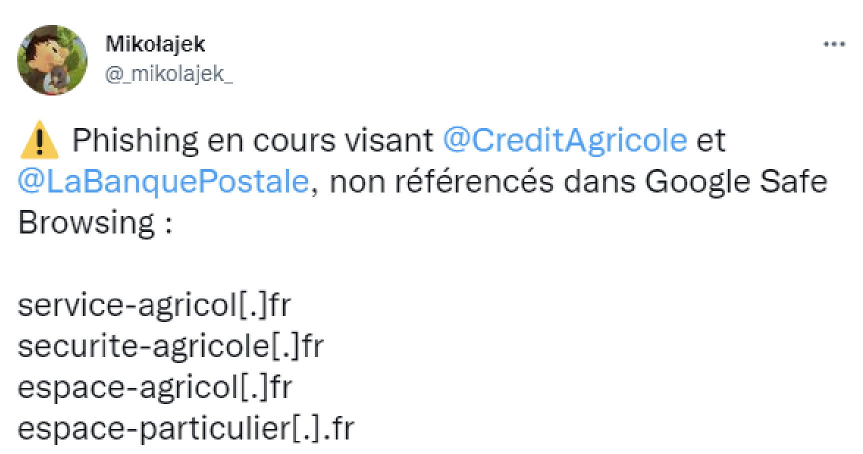
...

API PLATFORM CONFERENCE

Password Issues

Phishing?

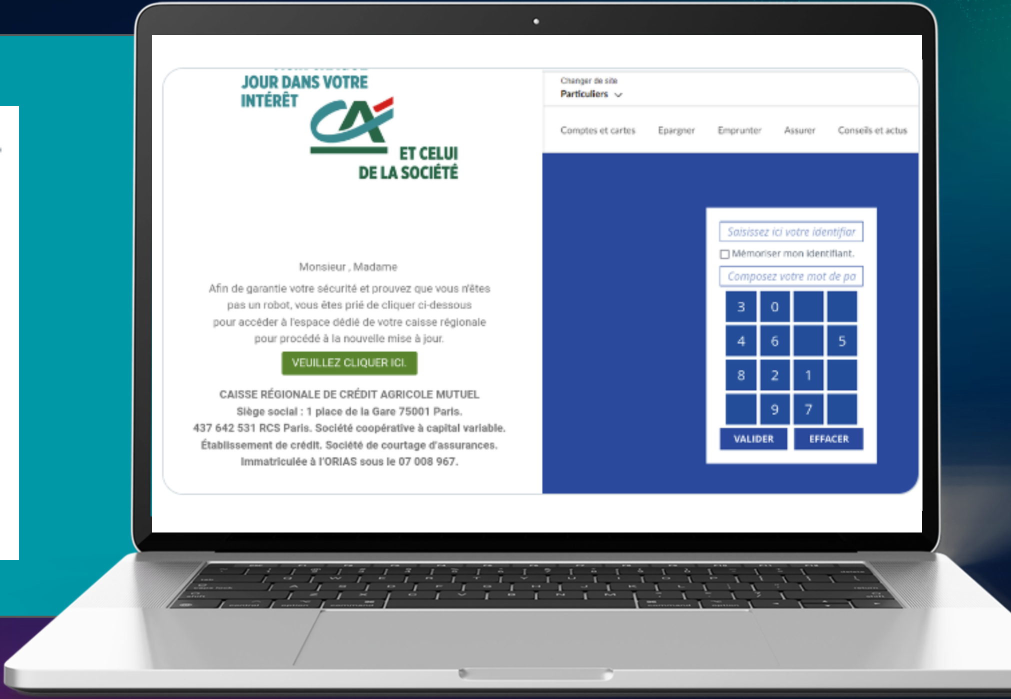
Phishing



Mikołajek
@_mikolajek_

⚠️ Phishing en cours visant @CreditAgricole et @LaBanquePostale, non référencés dans Google Safe Browsing :

service-agricol[.]fr
securite-agricole[.]fr
espace-agricol[.]fr
espace-particulier[.].fr



☐ Mémoriser mon identifiant.

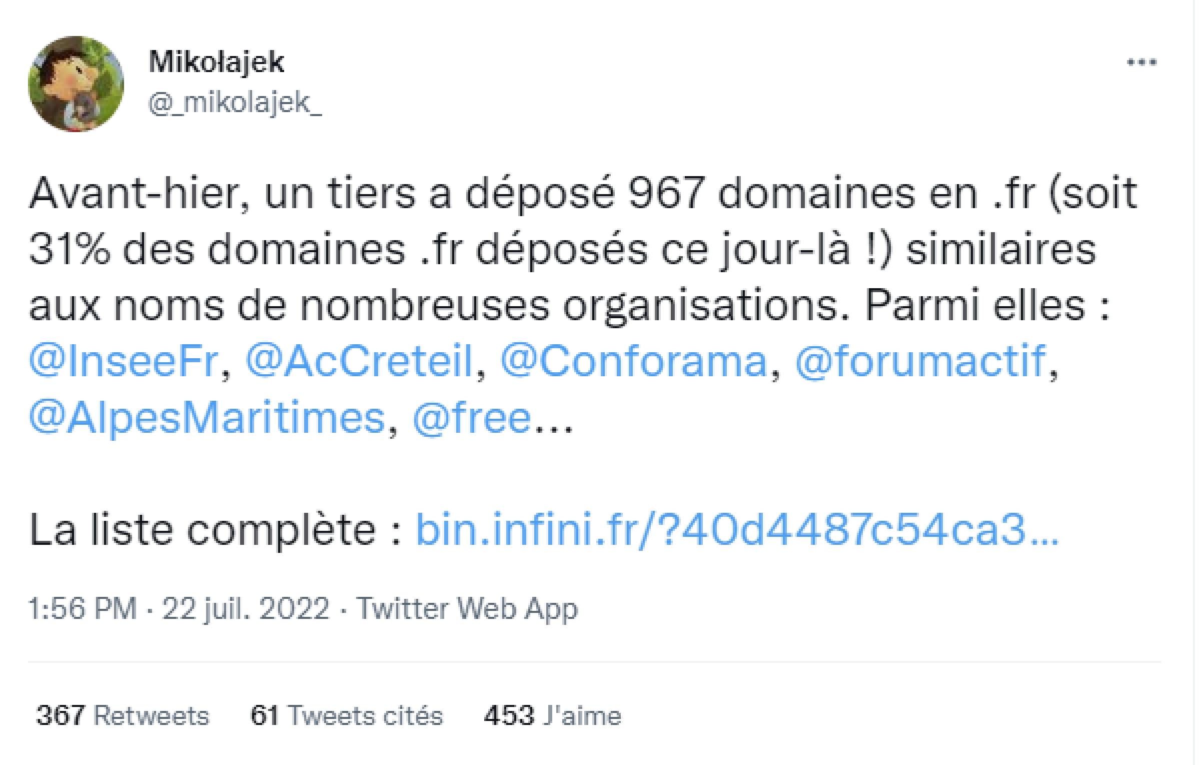
Monsieur , Madame

Afin de garantie votre sécurité et prouvez que vous n'êtes pas un robot, vous êtes prié de cliquer ci-dessous pour accéder à l'espace dédié de votre caisse régionale pour procédé à la nouvelle mise à jour.

VEUILLEZ CLIQUER ICI.

CAISSE RÉGIONALE DE CRÉDIT AGRICOLE MUTUEL
Siège social : 1 place de la Gare 75001 Paris.
437 642 531 RCS Paris. Société coopérative à capital variable.
Établissement de crédit. Société de courtage d'assurances.
Immatriculée à l'ORIAS sous le 07 008 967.

Phishing



Mikołajek
@_mikolajek_

Avant-hier, un tiers a déposé 967 domaines en .fr (soit 31% des domaines .fr déposés ce jour-là !) similaires aux noms de nombreuses organisations. Parmi elles : @InseeFr, @AcCreteil, @Conforama, @forumactif, @AlpesMaritimes, @free…

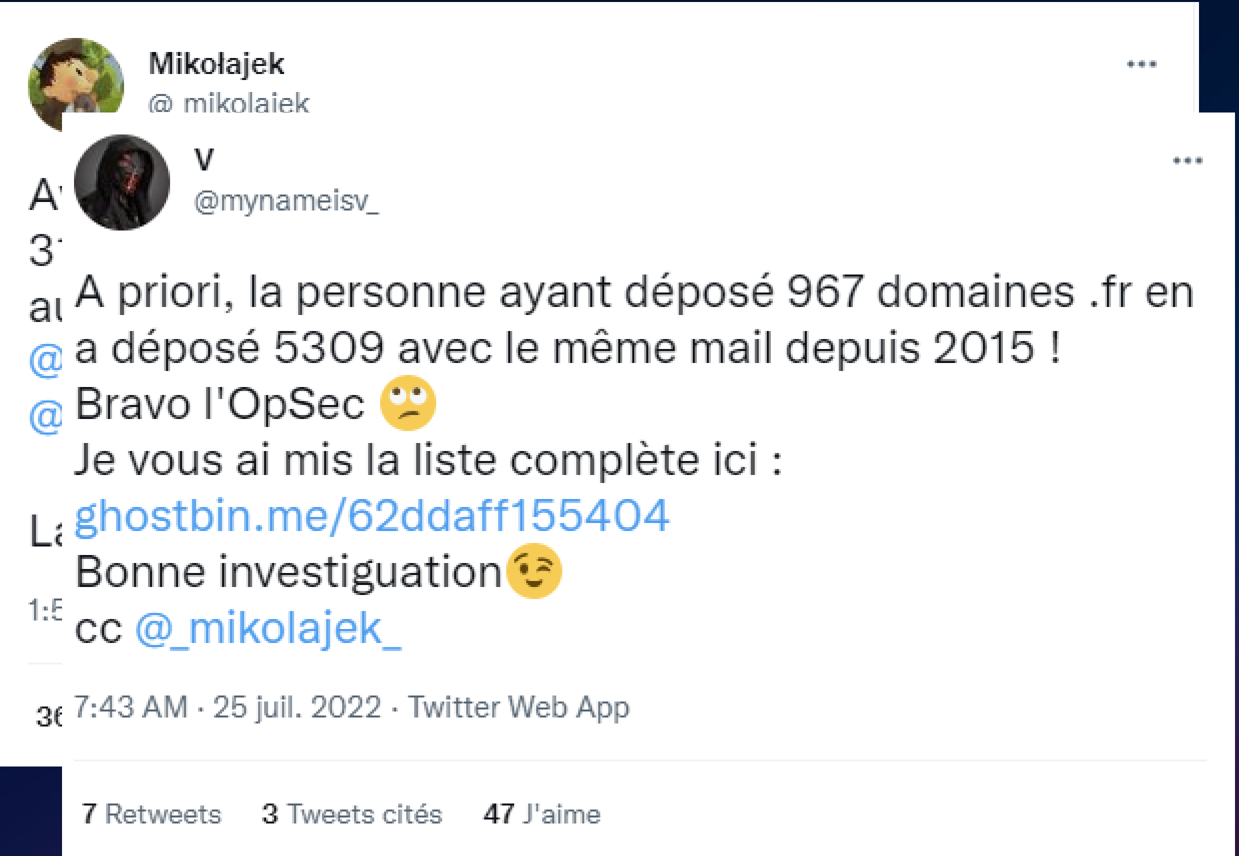
La liste complète : bin.infini.fr/?40d4487c54ca3…

1:56 PM · 22 juil. 2022 · Twitter Web App

367 Retweets 61 Tweets cités 453 J'aime

Any application can be **targeted**

Phishing



Mikołajek
@_mikolajek

A
3
au
@
@

La

1:5

36

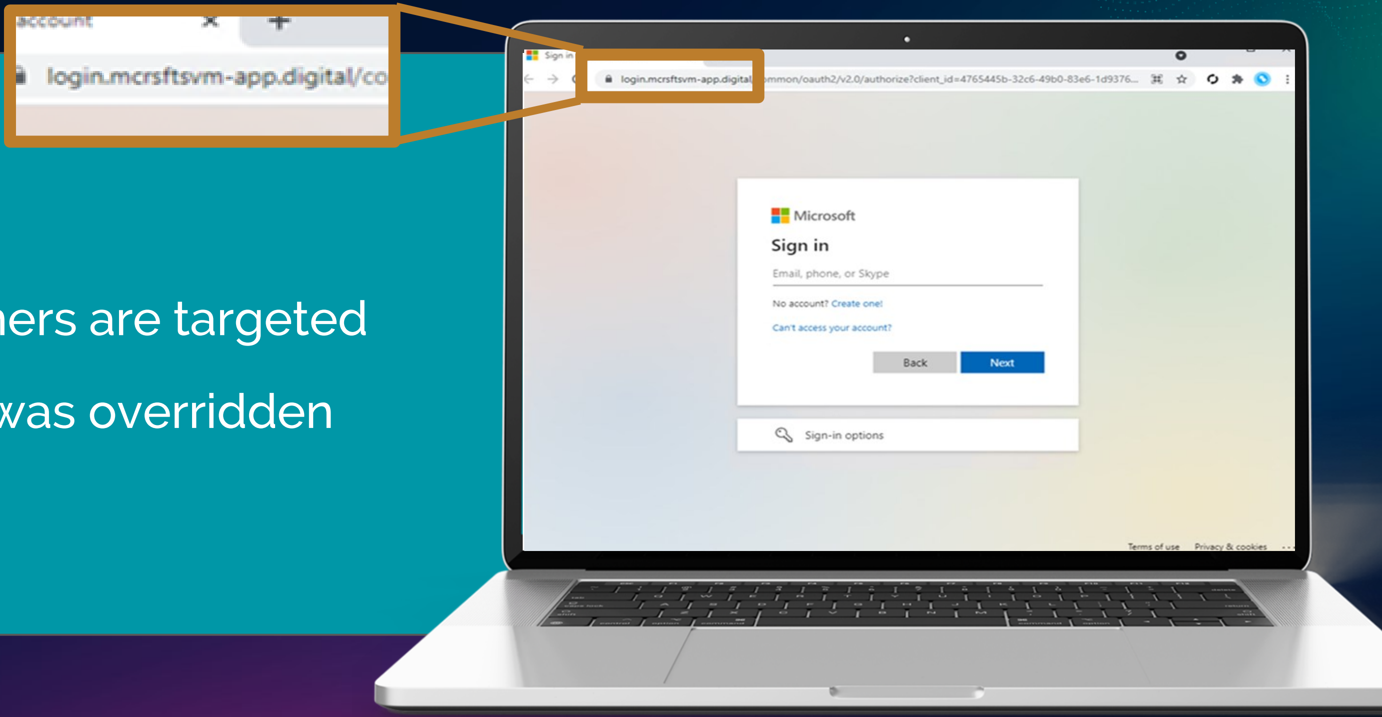
V
@mynameisv_

A priori, la personne ayant déposé 967 domaines .fr en a déposé 5309 avec le même mail depuis 2015 !
Bravo l'OpSec 🙄
Je vous ai mis la liste complète ici :
ghostbin.me/62ddaff155404
Bonne investiguation😉
cc @_mikolajek_

7:43 AM · 25 juil. 2022 · Twitter Web App

7 Retweets 3 Tweets cités 47 J'aime

Phishing



account

login.mcrsftsvm-app.digital/co

Sign in

login.mcrsftsvm-app.digital/common/oauth2/v2.0/authorize?client_id=4765445b-32c6-49b0-83e6-1d9376...

Microsoft

Sign in

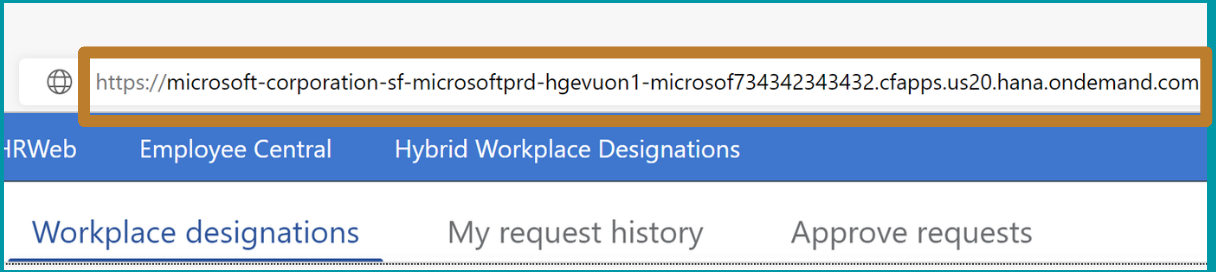
Email, phone, or Skype

No account? Create one!

Can't access your account?

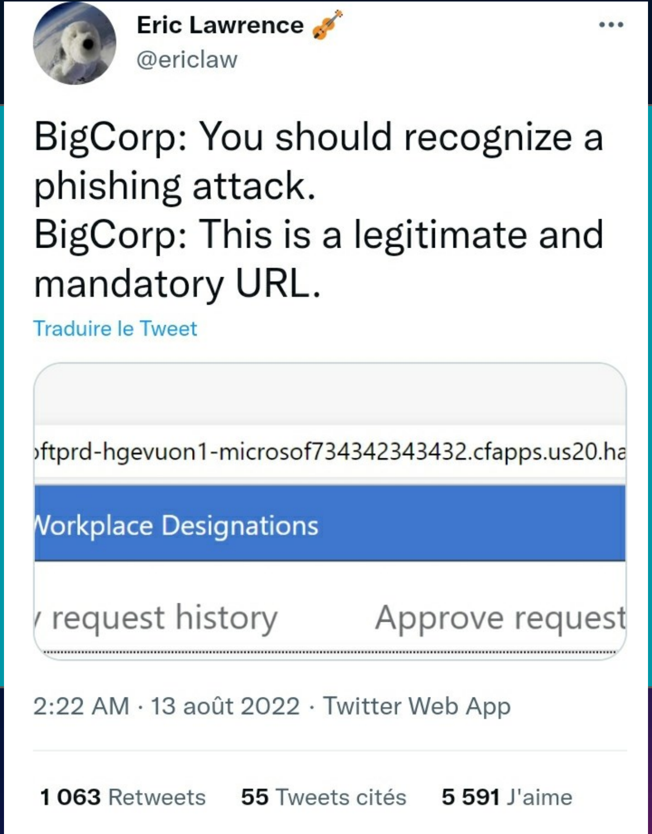
- ✓ **July 2022**
- ✓ Mirosoft customers are targeted
- ✓ Even 2nd factor was overridden

Phishing



https://microsoft-corporation-sf-microsoftprd-hgevuon1-microsof734342343432.cfapps.us20.hana.ondemand.com

Phishing



Eric Lawrence 🎻
@ericlaw

BigCorp: You should recognize a phishing attack.
BigCorp: This is a legitimate and mandatory URL.

Traduire le Tweet

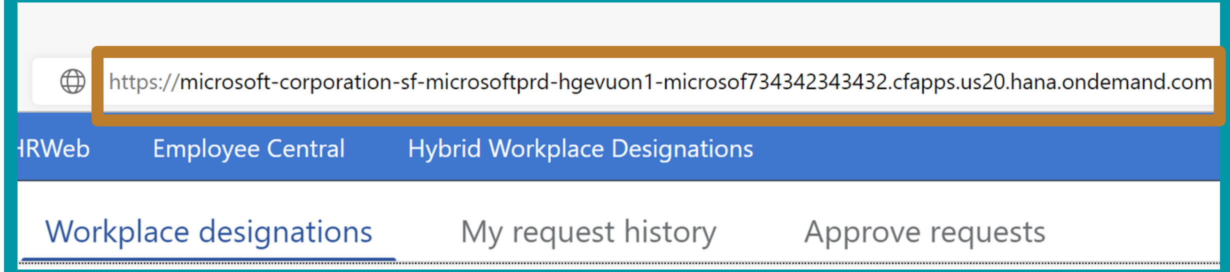
ɔftprd-hgevuon1-microsof734342343432.cfapps.us20.ha

Vorkplace Designations

/ request history   Approve request

2:22 AM · 13 août 2022 · Twitter Web App

1 063 Retweets   55 Tweets cités   5 591 J'aime



https://microsoft-corporation-sf-microsoftprd-hgevuon1-microsof734342343432.cfapps.us20.hana.ondemand.com

Phishing

Collection #1

Wikipedia: Collection #1 is the name of **a set of email addresses and passwords** that appeared on the dark web around January 2019.

Phishing

Have I Been Pwned ?

https://haveibeenpwned.com/

Phishing

Collection #2!

#3, #4, #5 and many more

Phishing

A total of

11.9+ billion accounts

records 🤯

02 – What about Api Platform?

Api Platform

Relies on the

PHP / Symfony

ecosystem

PHP / Symfony Ecosystem

bjeavons/zxcvbn-php

Zxcvbn-PHP is a **password strength estimator** using pattern matching and minimum entropy calculation

- ✓ Allows you to obtain a score **from 0** (very weak) **to 4** (very strong)
- ✓ Avoids **repetition**, **short** passwords, **dictionary** words, low **character variation**, etc.

PHP / Symfony Ecosystem

Symfony Constraints

- ✓ **Length**
- ✓ **NotCompromisedPassword**: validates by checking that it is not included in any of the public data breaches tracked by *haveibeenpwned.com*.

No need for password change if not compromised!

PHP / Symfony Ecosystem

Symfony Security

- ✓ **Automatically selected** hashing functions
- ✓ **Update / Migrate** if more reliable hashing function
- ✓ **Session** management
- ✓ And **many other features** constantly revised

PHP / Symfony Ecosystem

NelmioSecurityBundle

Provides additional security features for your Symfony application

- ✓ Content Security Policy,
- ✓ Signed Cookies,
- ✓ XSS Protection,
- ✓ …

PHP / Symfony Ecosystem

NelmioCorsBundle

- ✓ Allows you to **send CORS headers**.
- ✓ **Shipped by default** with Api Platform

PHP / Symfony Ecosystem

scheb/2fa

Multi-factor Authentication

- ✓ One-time passwords (TOTP)
- ✓ SMS
- ✓ Email
- ✓ …

Is it enough?

OWASP’s TOP 10

https://owasp.org/www-project-top-ten/

“OWASP is a nonprofit foundation that works to improve the security of software.”

API PLATFORM
CONFERENCE

03 – How to solve password issues?

API PLATFORM CONFERENCE

How to solve password issues?

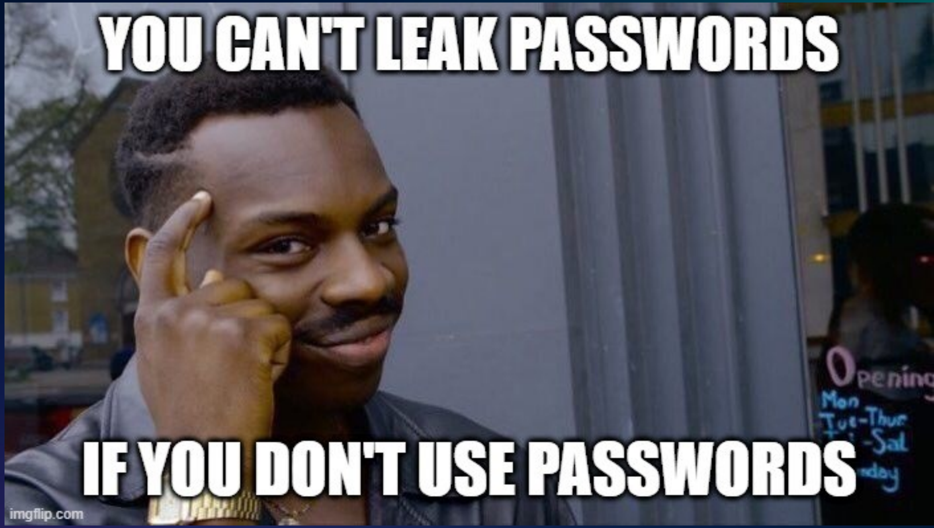
The main problem remains the very **concept of the password**:

✓ It **can be reused** across contexts/applications

✓ It is **shared**

How to solve password issues?

Well,
get rid of them!



FIDO Alliance

An **open industry association** with a focused **mission**:

*Authentication **standards** to help **reduce** the world's over-reliance on **passwords**.*

Web Standards to the rescue

Webauthn: Scoped and strong authentication made easy

Webauthn

API that enables the creation and use of **strong public key-based** credentials by web applications.

Webauthn

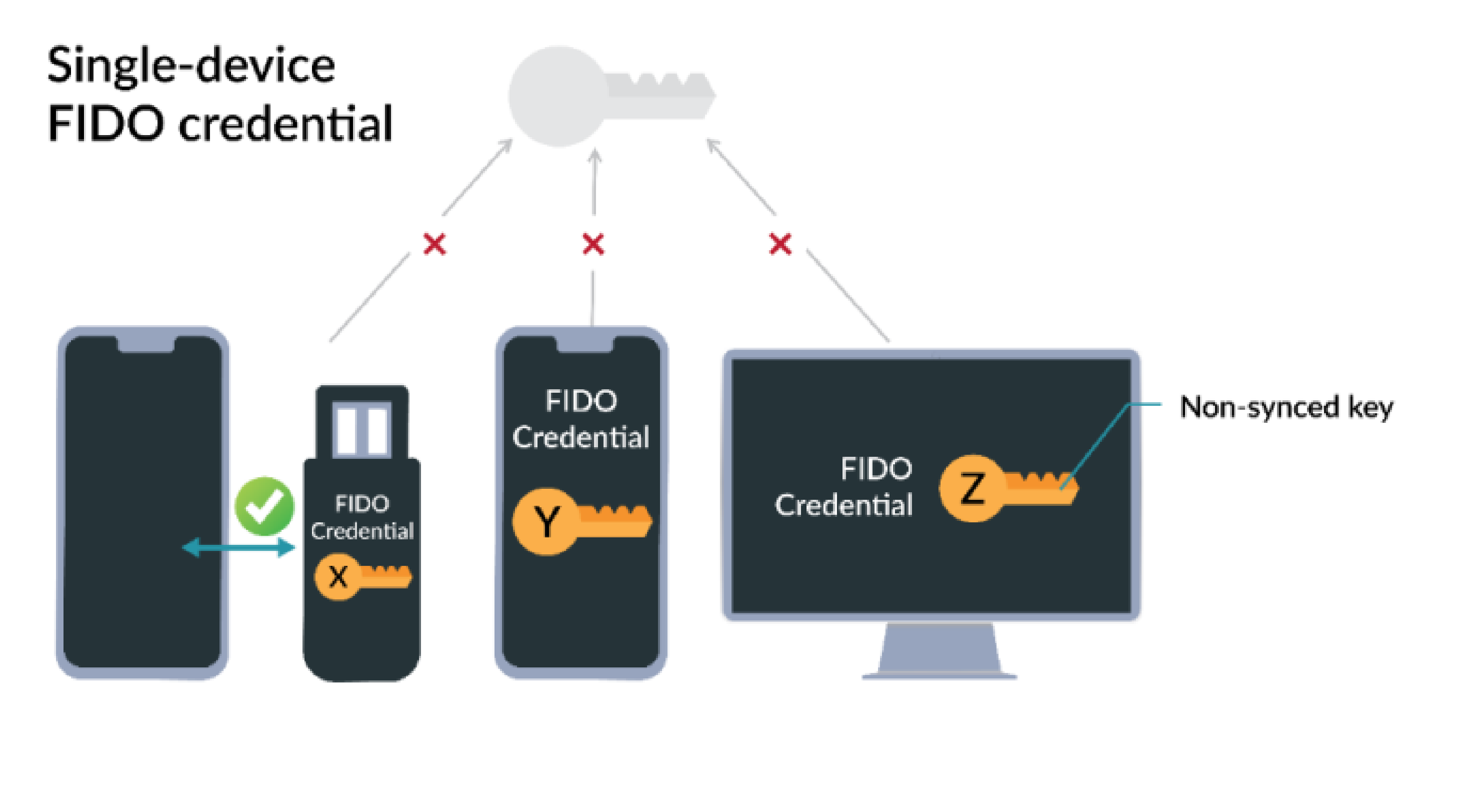
Cryptographic operations are delegated to **authenticators**

Webauthn Authenticators

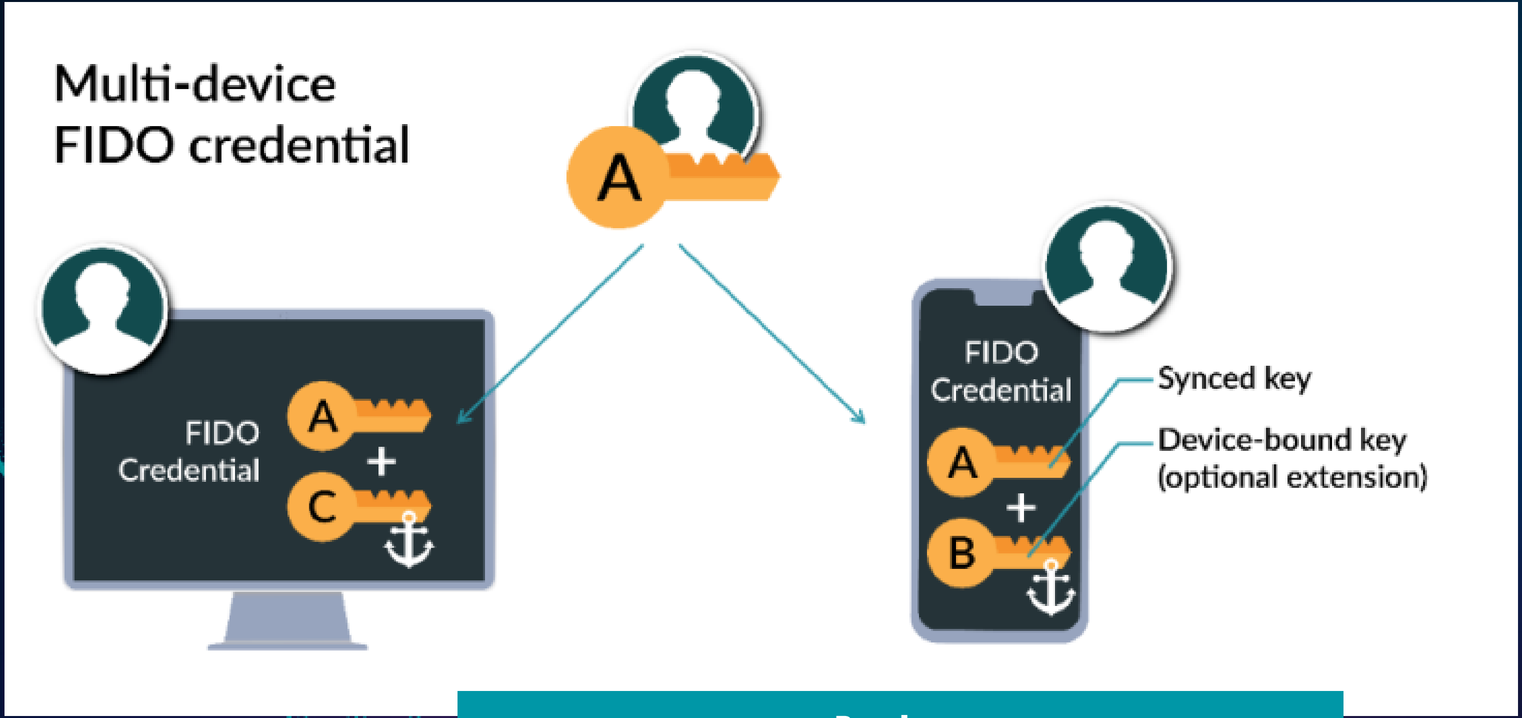
- ✓ **Platform**: embedded in the device
- ✓ **Roaming**: NFC, USB, Lightning, Bluetooth LE, caBLE

Old U2F Security Keys are compatible!

FIDO Credentials



Multi-device FIDO Credentials

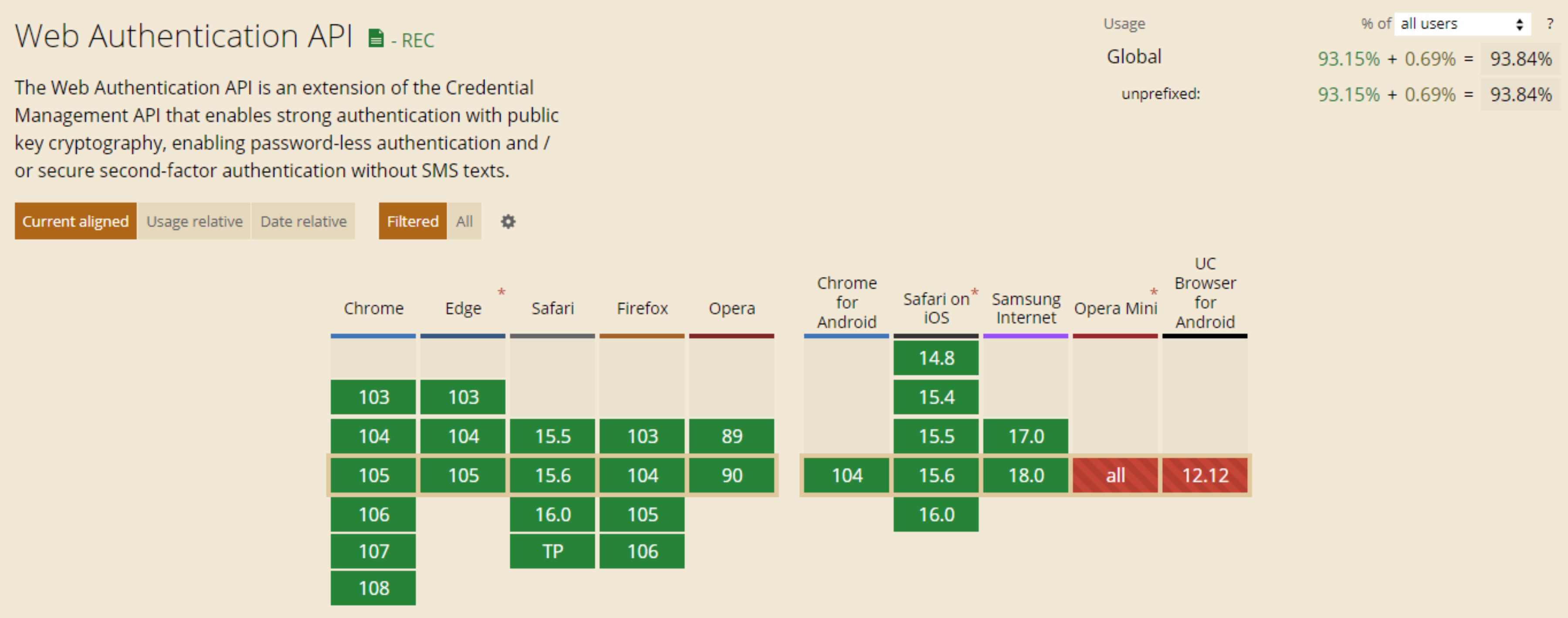


« Passkeys »

Webauthn & Operating Systems

- ✓ **Windows** 10+,
- ✓ **Android** 7+,
- ✓ **MacOS** 11, **iOS**14.2, **iPadOS** 15.5
- ✓ **Linux**: only via web browsers

Webauthn & Browsers



Web Authentication API - REC

Usage   % of all users   ?

Global   93.15% + 0.69% = 93.84%

unprefixed: 93.15% + 0.69% = 93.84%

The Web Authentication API is an extension of the Credential Management API that enables strong authentication with public key cryptography, enabling password-less authentication and / or secure second-factor authentication without SMS texts.

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	UC Browser for Android
						14.8			
103	103					15.4			
104	104	15.5	103	89		15.5	17.0		
105	105	15.6	104	90	104	15.6	18.0	all	12.12
106		16.0	105			16.0			
107		TP	106						
108									

API PLATFORM CONFERENCE

04 – How Webauthn Works?

Webauthn Ceremonies

- ✓ **Creation**: registration of an authenticator (new or existing account)
- ✓ **Request**: use of an existing authenticator

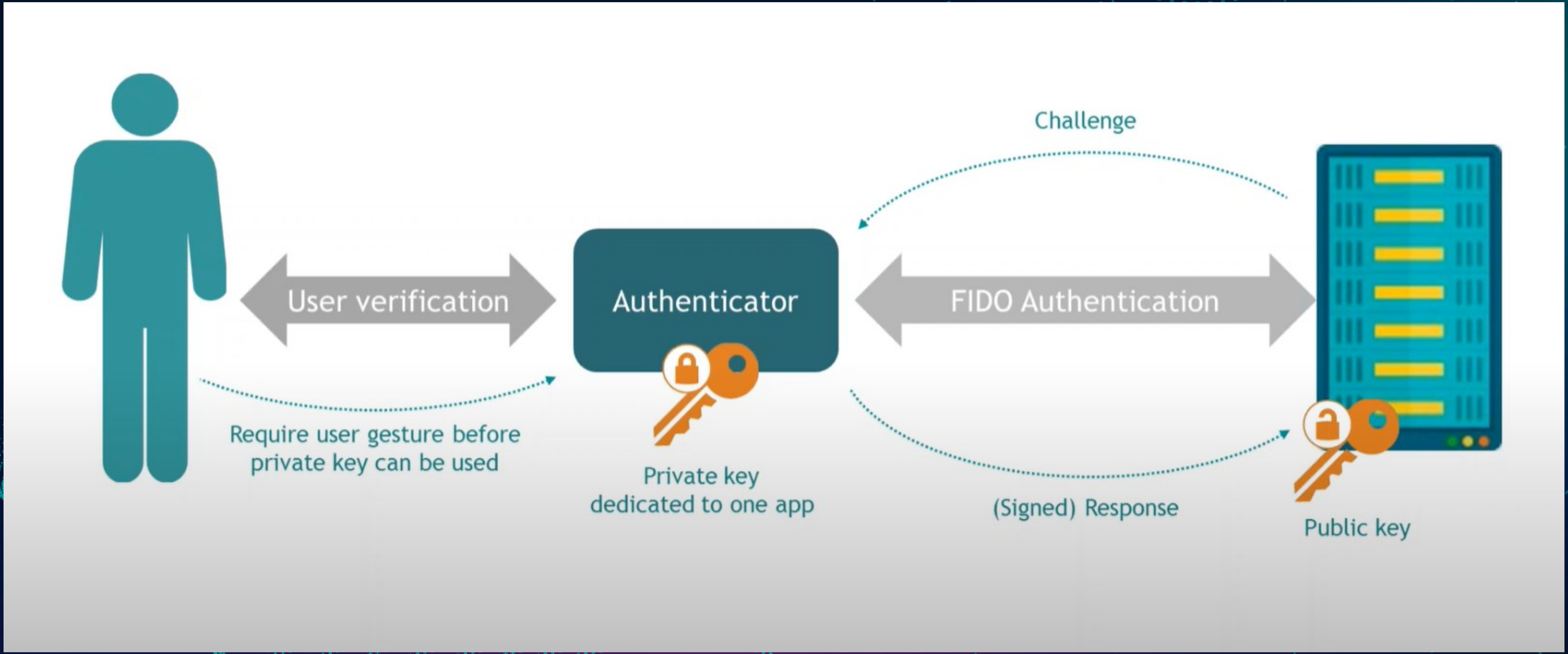
Webauthn Ceremonies

Each ceremony requires **2 HTTP requests**:

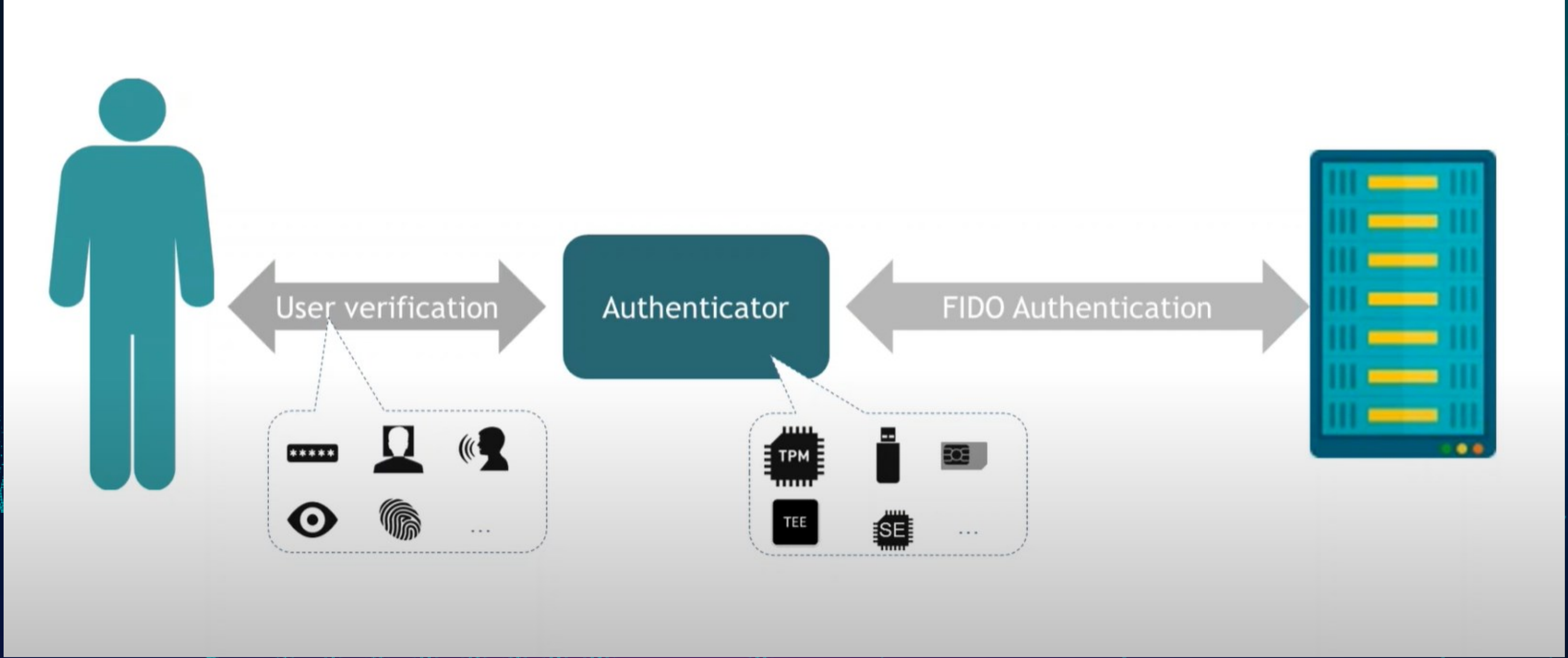
✓ **Obtaining options** (challenge + security policy)

✓ **Sending the result** of the challenge (+ credentials)

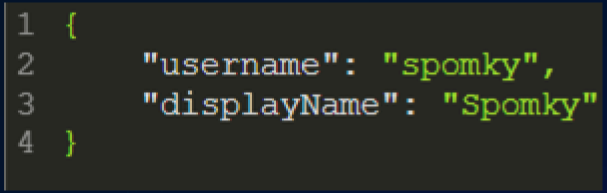
Webauthn Ceremonies



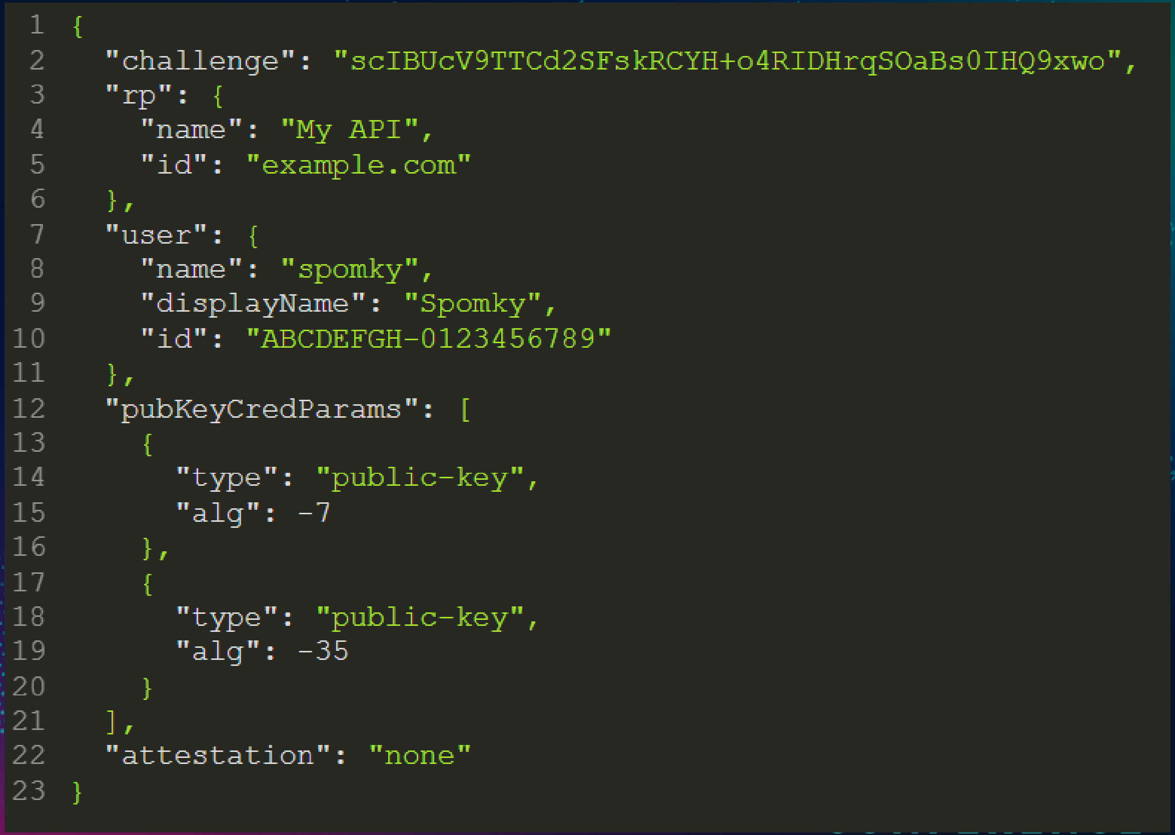
Webauthn Ceremonies



Webauthn Ceremonies

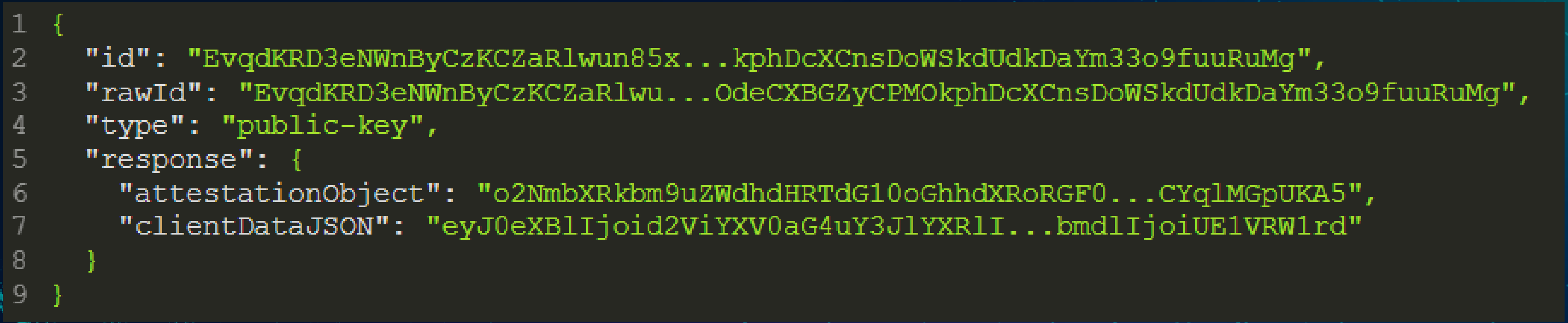


```
{
	"username": "spomky",
	"displayName": "Spomky"
}
```



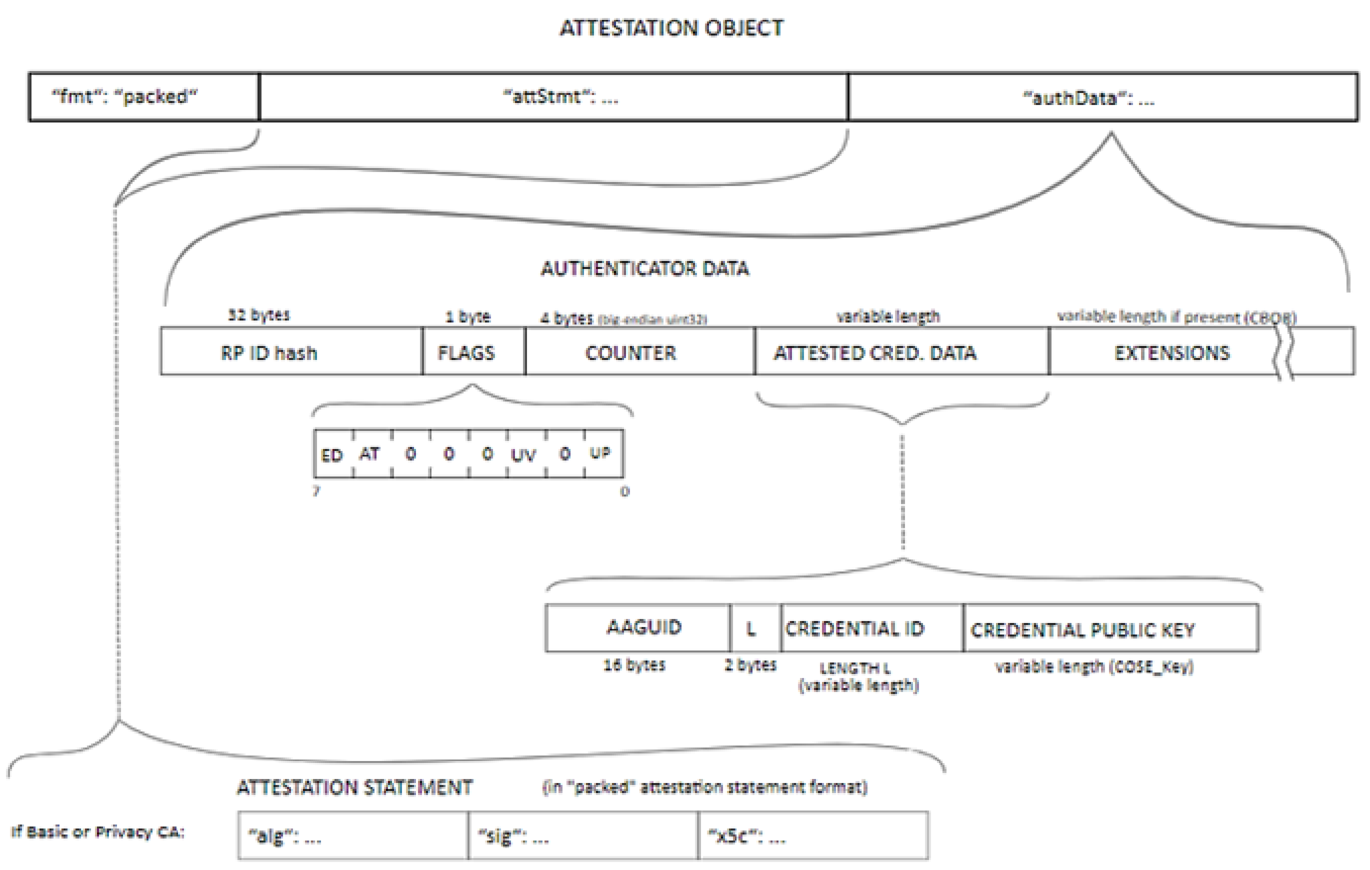
```
{
  "challenge": "scIBUcV9TTCd2SFskRCYH+o4RIDHrqSOaBs0IHQ9xwo",
  "rp": {
    "name": "My API",
    "id": "example.com"
  },
  "user": {
    "name": "spomky",
    "displayName": "Spomky",
    "id": "ABCDEFGH-0123456789"
  },
  "pubKeyCredParams": [
    {
      "type": "public-key",
      "alg": -7
    },
    {
      "type": "public-key",
      "alg": -35
    }
  ],
  "attestation": "none"
}
```

Webauthn Ceremonies



```
{
  "id": "EvqdKRD3eNWnByCzKCZaRlwun85x...kphDcXCnsDoWSkdUdkDaYm33o9fuuRuMg",
  "rawId": "EvqdKRD3eNWnByCzKCZaRlwu...OdeCXBGZyCPMOkphDcXCnsDoWSkdUdkDaYm33o9fuuRuMg",
  "type": "public-key",
  "response": {
    "attestationObject": "o2NmbXRkbm9uZWdhdHRTdG10oGhhdXRoRGF0...CYqlMGpUKA5",
    "clientDataJSON": "eyJ0eXBlIjoid2ViYXV0aG4uY3JlYXRlIi...bmdlIjoiUE1VRW1rd"
  }
}
```

Webauthn Ceremonies



05 –Application Integration?

Application Integration

web-auth/webauthn-symfony-bundle

- ✓ A bundle to allow developers **integrating Webauthn** in Symfony-based applications.
- ✓ Based on the library web-auth/webauthn-lib
- ✓ FIDO2 Conformant 🍾

Application Integration

Glue Code

Bundle and firewall configuration

Frontend

API PLATFORM CONFERENCE

Application Integration

Requirements

- ✓ You have a **User class, repository and provider**
- ✓ The User Repository can fetch users from their ID and their username
- ✓ The **User ID shall be a string**

Application Integration

Requirements

The User ID shall be a string:

✓ If you have an **integer** as ID, no worries: you can **generate a random value** and associate it to your users.

Application Integration

Bundle and firewall
configuration

Glue Code

Frontend

Webauthn Entity

The bundle requires:

- ✓ **User Entity** Class and associated Repository
- ✓ **Credential Entity** Class and associated Repository

Webauthn User Entity

Webauthn User Entity ⇎ Symfony User

✓ Webauthn User Entity class is **provided by the bundle**

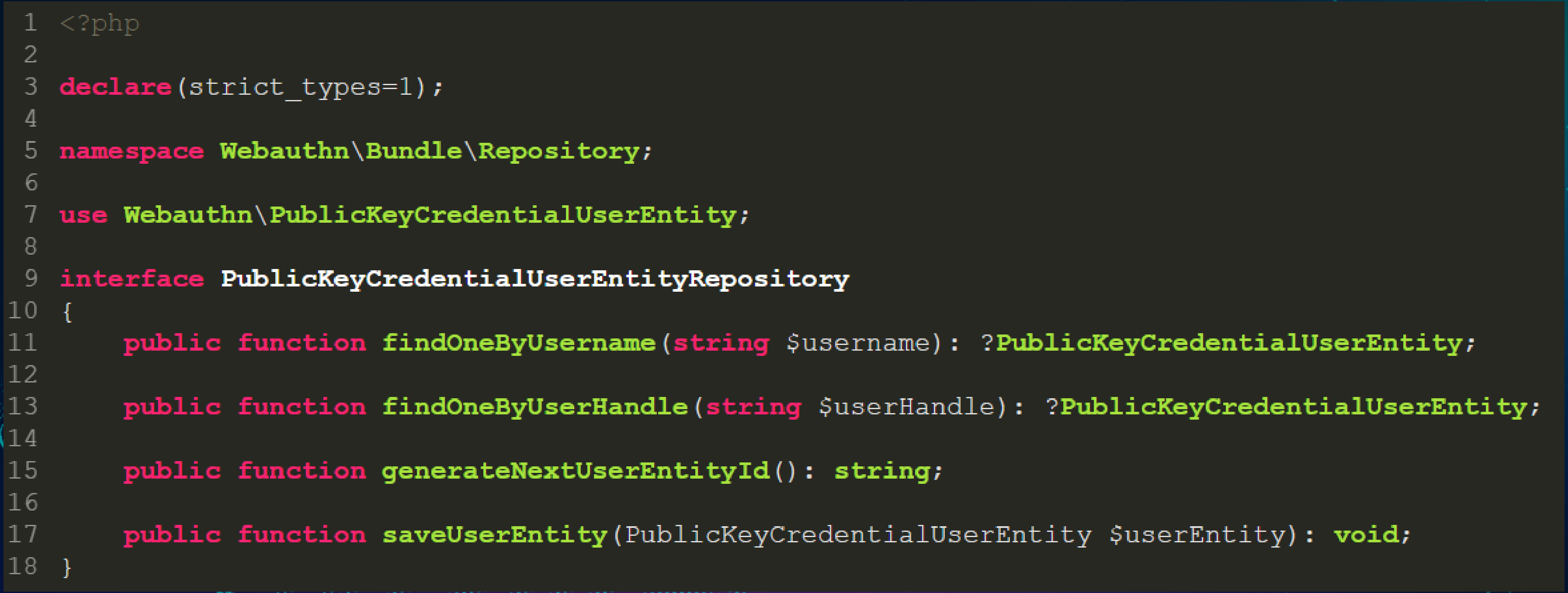
✓ Conversion is easy

Webauthn User Repository

Webauthn User Repository ⇎ Symfony User Repository

- ✓ We can **leverage on the Symfony User Repository**
- ✓ Webauthn User Entity object can be created on demand

Webauthn User Repository

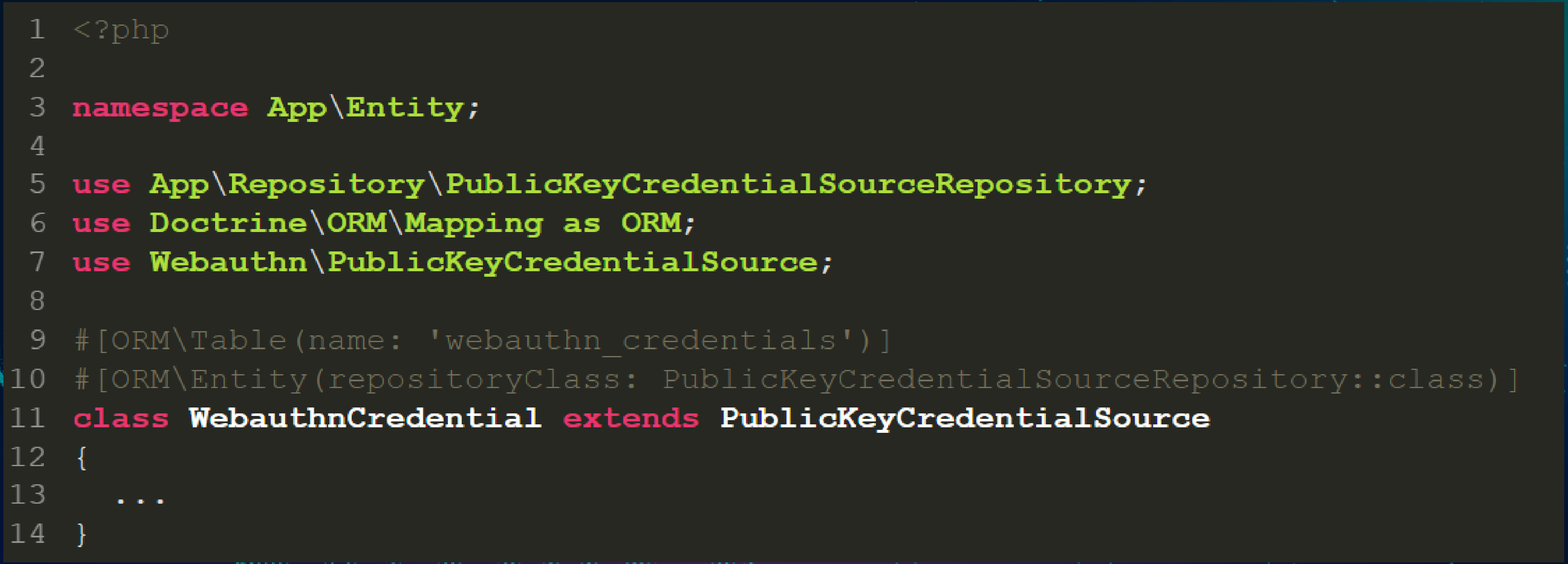


```php
<?php

declare(strict_types=1);

namespace Webauthn\Bundle\Repository;

use Webauthn\PublicKeyCredentialUserEntity;

interface PublicKeyCredentialUserEntityRepository
{
    public function findOneByUsername(string $username): ?PublicKeyCredentialUserEntity;

    public function findOneByUserHandle(string $userHandle): ?PublicKeyCredentialUserEntity;

    public function generateNextUserEntityId(): string;

    public function saveUserEntity(PublicKeyCredentialUserEntity $userEntity): void;
}
```

Webauthn Credential

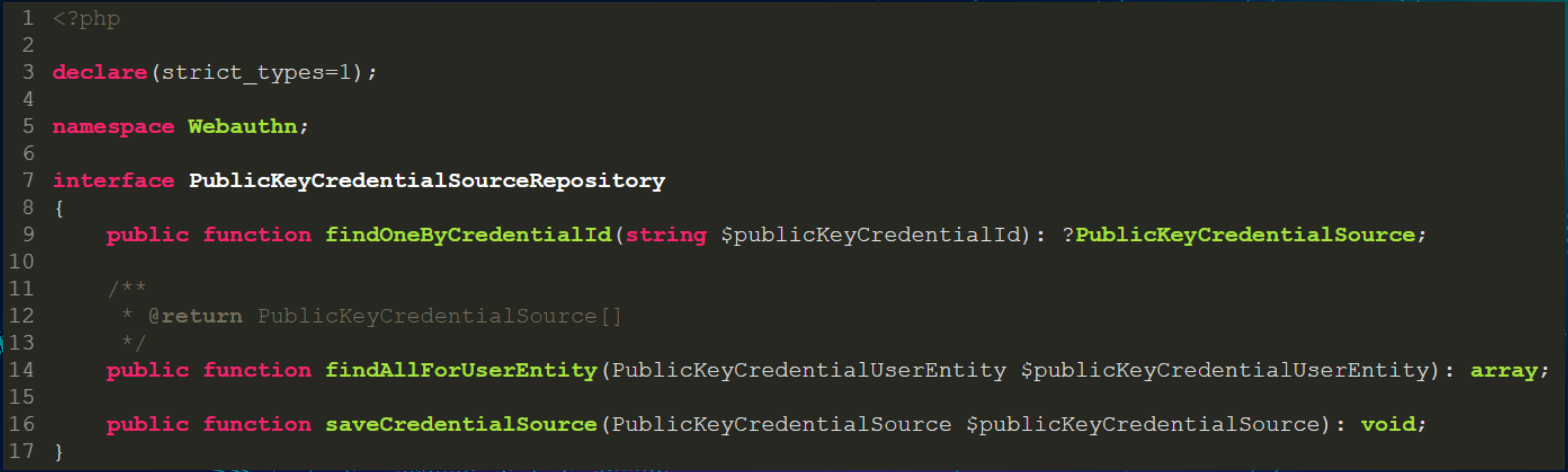
- ✓ A class is **provided by the bundle**
- ✓ Can be converted into **JSON** (e.g. filesystem storage)
- ✓ Can be stored via your **DBMS** (e.g. Doctrine)

Webauthn Credential Entity



```php
<?php

namespace App\Entity;

use App\Repository\PublicKeyCredentialSourceRepository;
use Doctrine\ORM\Mapping as ORM;
use Webauthn\PublicKeyCredentialSource;

#[ORM\Table(name: 'webauthn_credentials')]
#[ORM\Entity(repositoryClass: PublicKeyCredentialSourceRepository::class)]
class WebauthnCredential extends PublicKeyCredentialSource
{
    ...
}
```

Webauthn Credential Repository



```php
<?php

declare(strict_types=1);

namespace Webauthn;

interface PublicKeyCredentialSourceRepository
{
    public function findOneByCredentialId(string $publicKeyCredentialId): ?PublicKeyCredentialSource;

    /**
     * @return PublicKeyCredentialSource[]
     */
    public function findAllForUserEntity(PublicKeyCredentialUserEntity $publicKeyCredentialUserEntity): array;

    public function saveCredentialSource(PublicKeyCredentialSource $publicKeyCredentialSource): void;
}
```

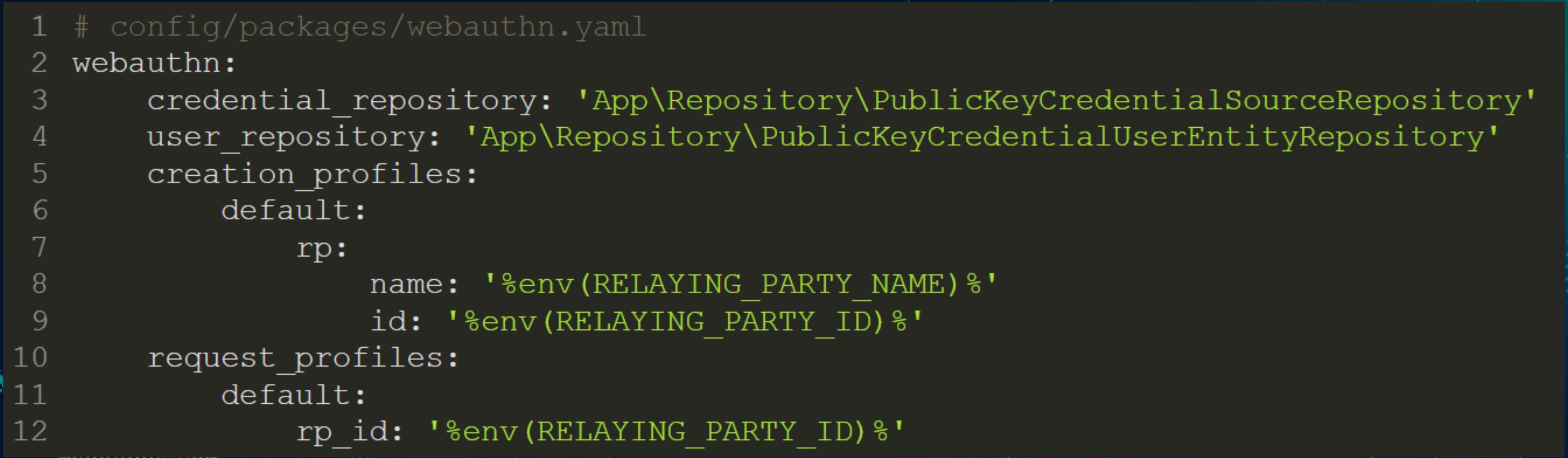
Application Integration

Glue Code

Bundle and firewall
configuration

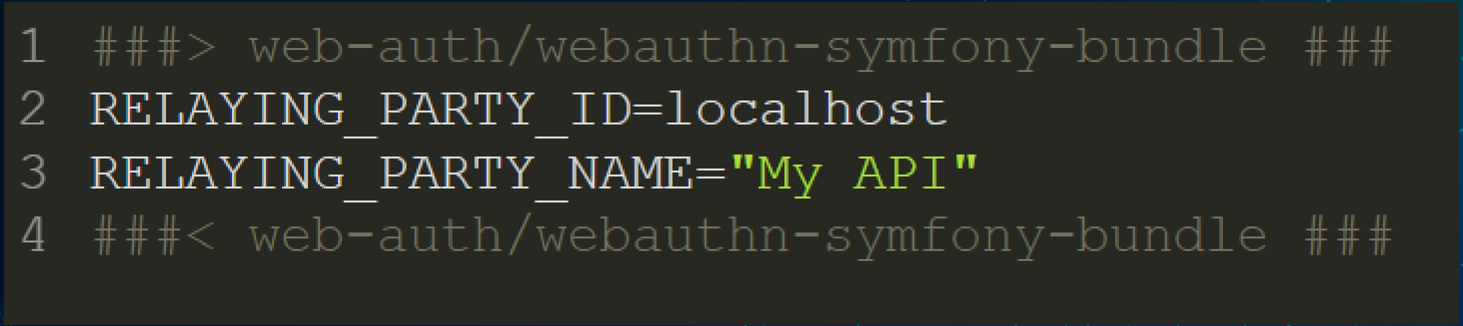
Frontend

Bundle/Firewall Configuration



```yaml
# config/packages/webauthn.yaml
webauthn:
    credential_repository: 'App\Repository\PublicKeyCredentialSourceRepository'
    user_repository: 'App\Repository\PublicKeyCredentialUserEntityRepository'
    creation_profiles:
        default:
            rp:
                name: '%env(RELAYING_PARTY_NAME)%'
                id: '%env(RELAYING_PARTY_ID)%'
    request_profiles:
        default:
            rp_id: '%env(RELAYING_PARTY_ID)%'
```

Bundle/Firewall Configuration



```
###> web-auth/webauthn-symfony-bundle ###
RELAYING_PARTY_ID=localhost
RELAYING_PARTY_NAME="My API"
###< web-auth/webauthn-symfony-bundle ###
```

Bundle/Firewall Configuration

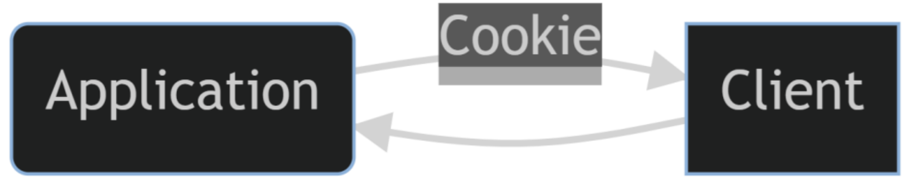
Depends on the type of application.

Stateful and **stateless** configurations are supported

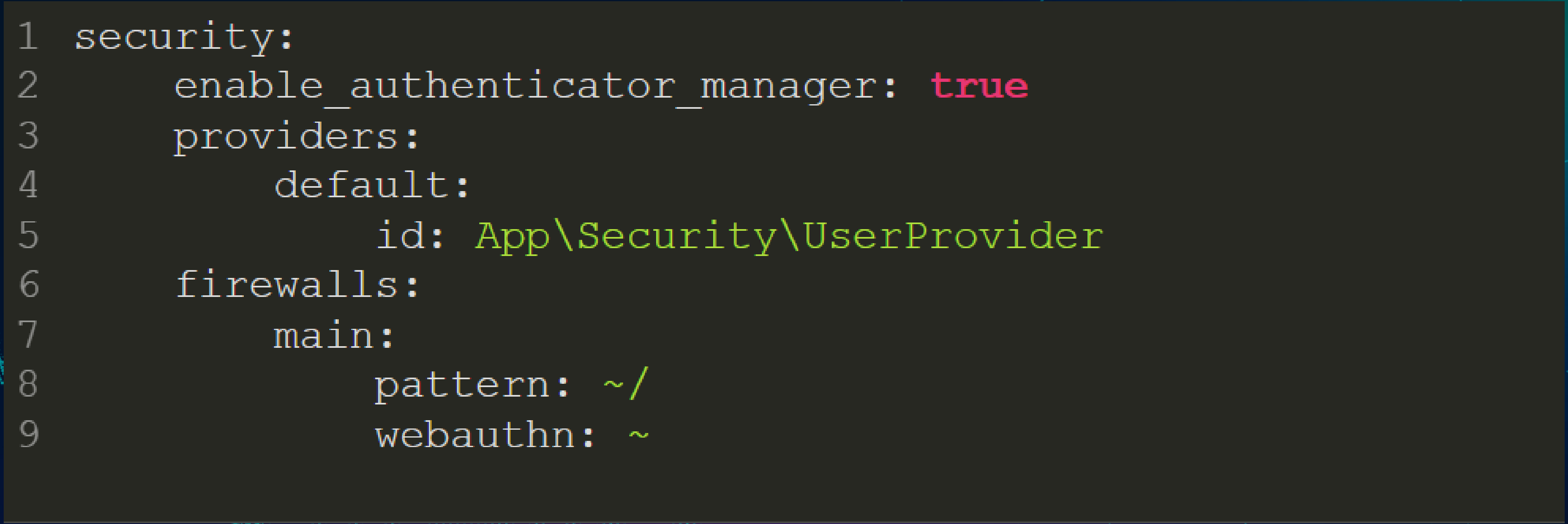
🤠

Bundle/Firewall Configuration

- ✓ Traditional application with sessions
- ✓ Sessions shall be configured in your application
  https://symfony.com/doc/current/session.html

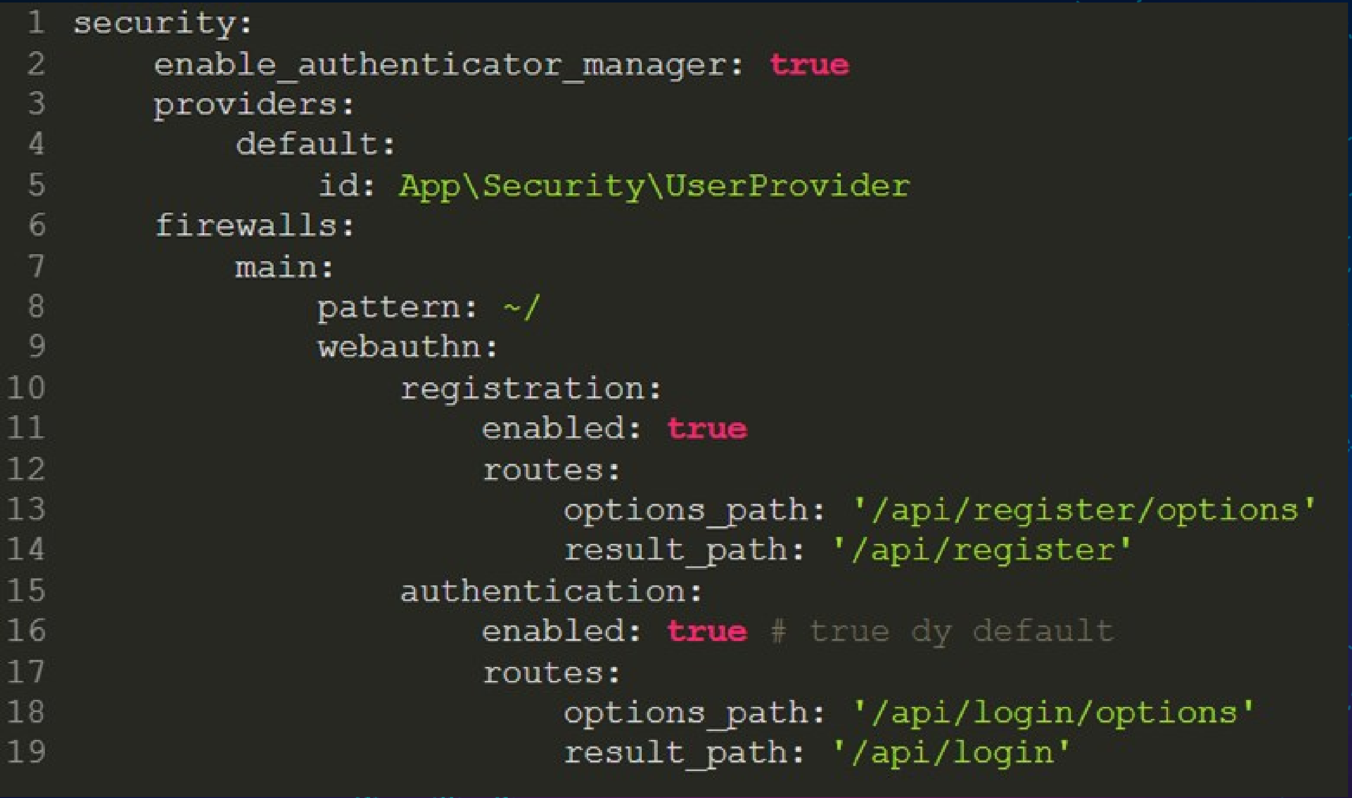


Bundle/Firewall Configuration



```yaml
security:
    enable_authenticator_manager: true
    providers:
        default:
            id: App\Security\UserProvider
    firewalls:
        main:
            pattern: ~/
            webauthn: ~
```

Bundle/Firewall Configuration



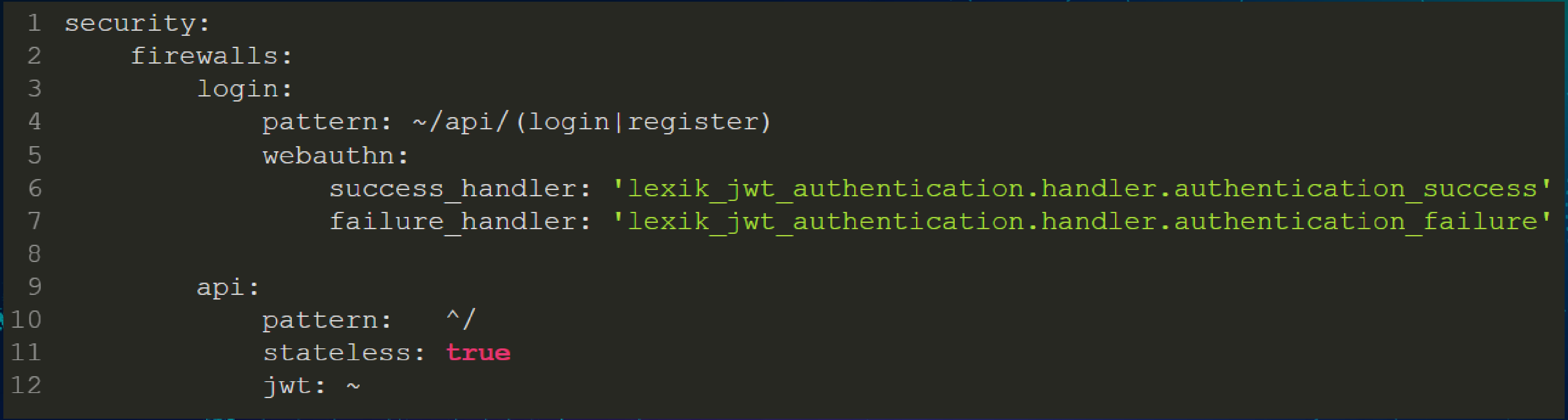
```yaml
security:
    enable_authenticator_manager: true
    providers:
        default:
            id: App\Security\UserProvider
    firewalls:
        main:
            pattern: ~/
            webauthn:
                registration:
                    enabled: true
                    routes:
                        options_path: '/api/register/options'
                        result_path: '/api/register'
                authentication:
                    enabled: true # true dy default
                    routes:
                        options_path: '/api/login/options'
                        result_path: '/api/login'
```

Bundle/Firewall Configuration

Webauthn + LexikJWTBundle = 🖤

- ✓ **Login and API endpoints** have different firewalls.
- ✓ The login endpoint is stateful
- ✓ The API endpoint is stateless

Bundle/Firewall Configuration



```yaml
security:
    firewalls:
        login:
            pattern: ~/api/(login|register)
            webauthn:
                success_handler: 'lexik_jwt_authentication.handler.authentication_success'
                failure_handler: 'lexik_jwt_authentication.handler.authentication_failure'

        api:
            pattern:   ^/
            stateless: true
            jwt: ~
```

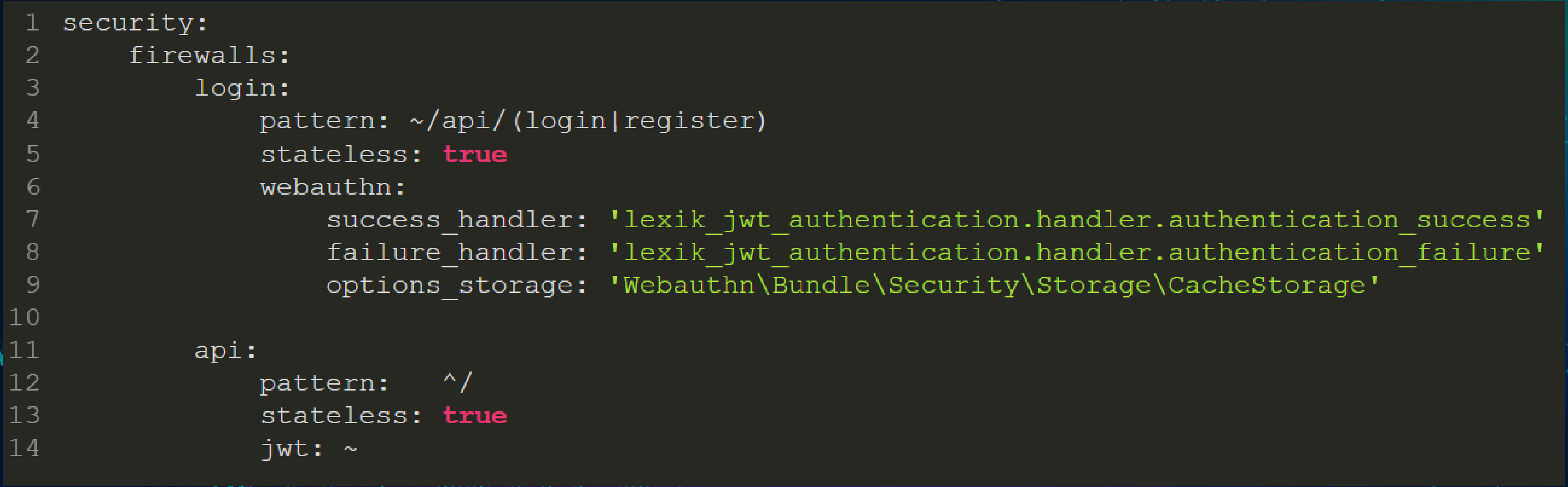
Bundle/Firewall Configuration

Stateless:

- ✓ Webauthn + LexikJWTBundle
- ✓ No Sessions, no cookies
- ✓ Each request shall carry a token

User shall login again when tab is closed

Bundle/Firewall Configuration



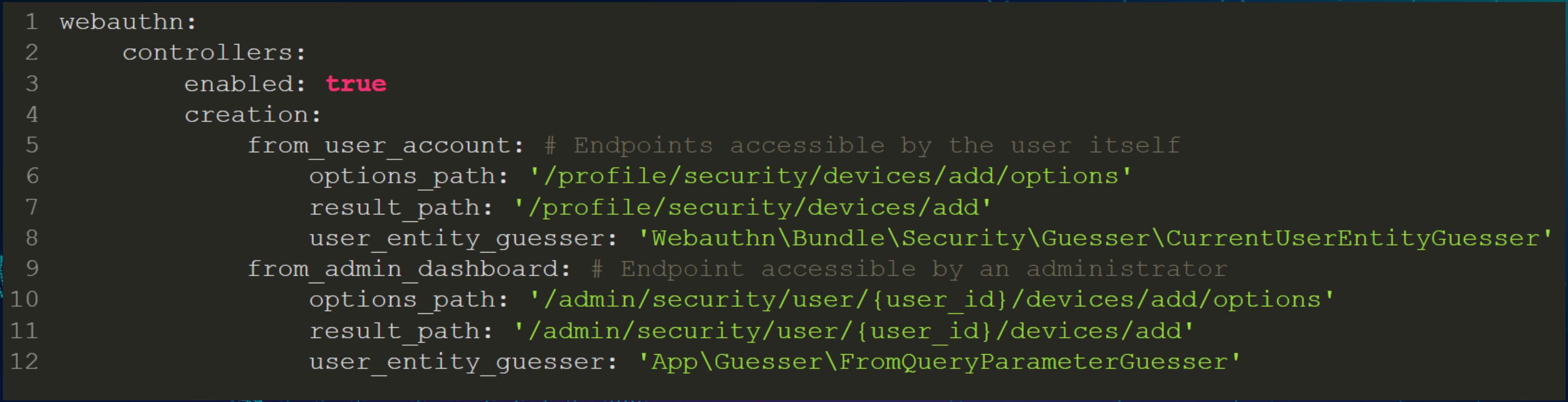
```yaml
security:
    firewalls:
        login:
            pattern: ~/api/(login|register)
            stateless: true
            webauthn:
                success_handler: 'lexik_jwt_authentication.handler.authentication_success'
                failure_handler: 'lexik_jwt_authentication.handler.authentication_failure'
                options_storage: 'Webauthn\Bundle\Security\Storage\CacheStorage'

        api:
            pattern:   ^/
            stateless: true
            jwt: ~
```

Bundle/Firewall Configuration

Additional Webauthn Authenticators:

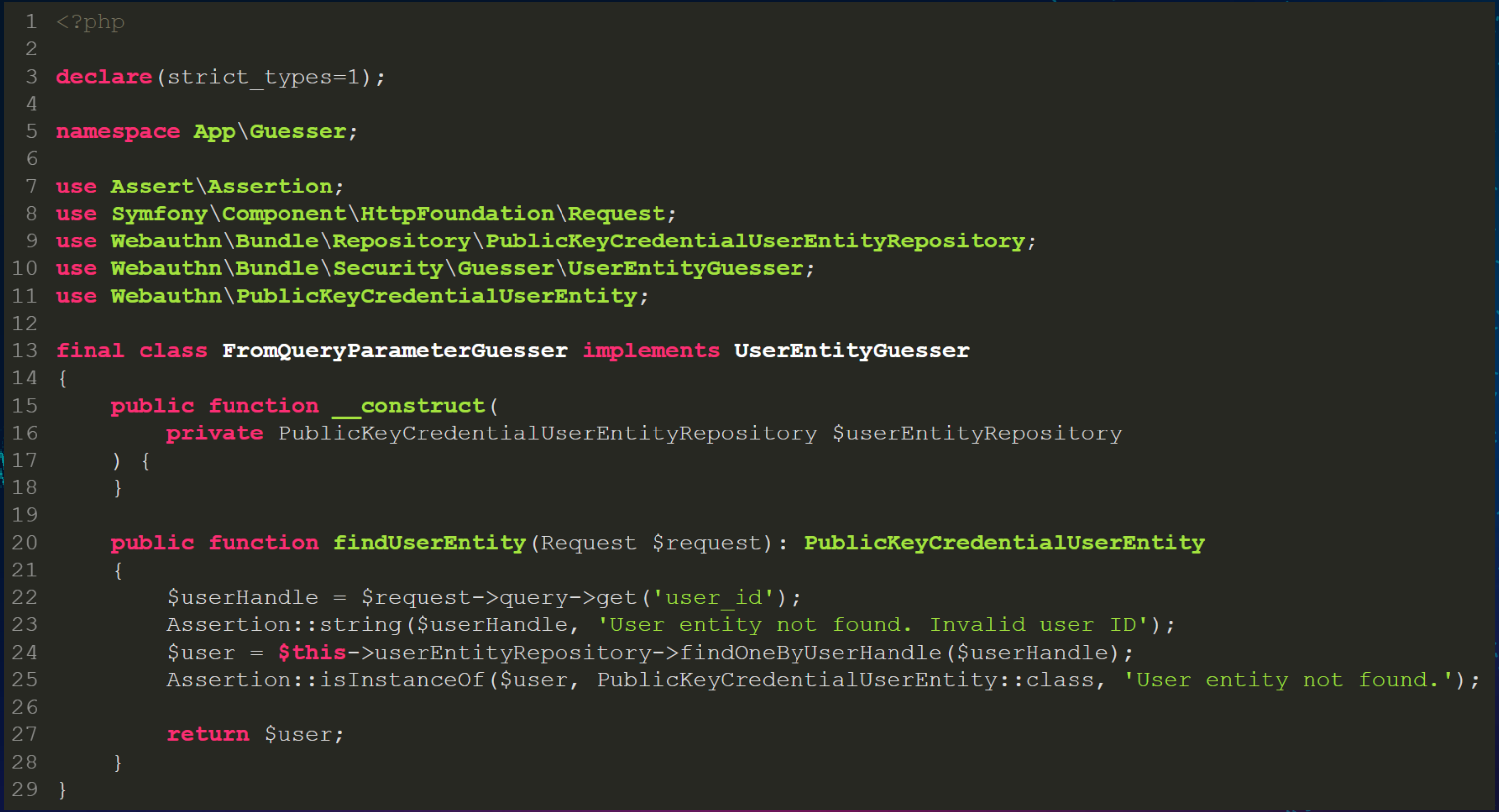
- ✓ Registering multiple authenticators is **highly recommended**
- ✓ Just a few lines of configuration
- ✓ Routes and Controllers are automatically created

Bundle/Firewall Configuration



```yaml
webauthn:
    controllers:
        enabled: true
        creation:
            from_user_account: # Endpoints accessible by the user itself
                options_path: '/profile/security/devices/add/options'
                result_path: '/profile/security/devices/add'
                user_entity_guesser: 'Webauthn\Bundle\Security\Guesser\CurrentUserEntityGuesser'
            from_admin_dashboard: # Endpoint accessible by an administrator
                options_path: '/admin/security/user/{user_id}/devices/add/options'
                result_path: '/admin/security/user/{user_id}/devices/add'
                user_entity_guesser: 'App\Guesser\FromQueryParameterGuesser'
```

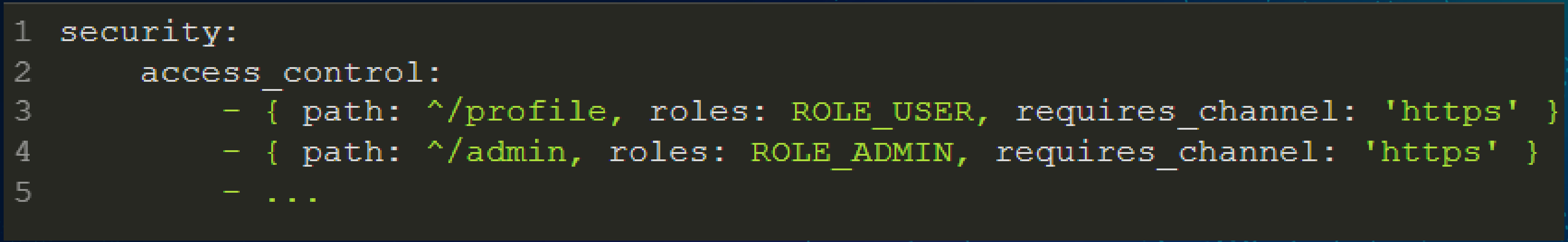
Bundle/Firewall Configuration



```php
<?php

declare(strict_types=1);

namespace App\Guesser;

use Assert\Assertion;
use Symfony\Component\HttpFoundation\Request;
use Webauthn\Bundle\Repository\PublicKeyCredentialUserEntityRepository;
use Webauthn\Bundle\Security\Guesser\UserEntityGuesser;
use Webauthn\PublicKeyCredentialUserEntity;

final class FromQueryParameterGuesser implements UserEntityGuesser
{
    public function __construct(
        private PublicKeyCredentialUserEntityRepository $userEntityRepository
    ) {
    }

    public function findUserEntity(Request $request): PublicKeyCredentialUserEntity
    {
        $userHandle = $request->query->get('user_id');
        Assertion::string($userHandle, 'User entity not found. Invalid user ID');
        $user = $this->userEntityRepository->findOneByUserHandle($userHandle);
        Assertion::isInstanceOf($user, PublicKeyCredentialUserEntity::class, 'User entity not found.');

        return $user;
    }
}
```

API PLATFORM CONFERENCE

Bundle/Firewall Configuration



```yaml
security:
    access_control:
        - { path: ^/profile, roles: ROLE_USER, requires_channel: 'https' }
        - { path: ^/admin, roles: ROLE_ADMIN, requires_channel: 'https' }
        - ...
```

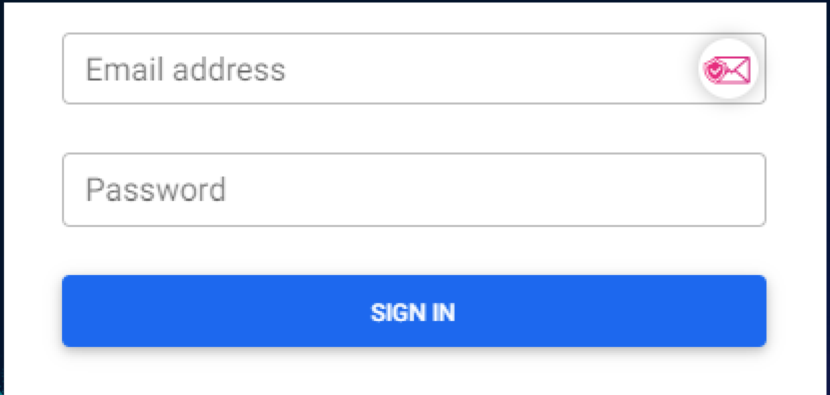
Application Integration

Glue Code

Frontend

Bundle and firewall
configuration

Frontend

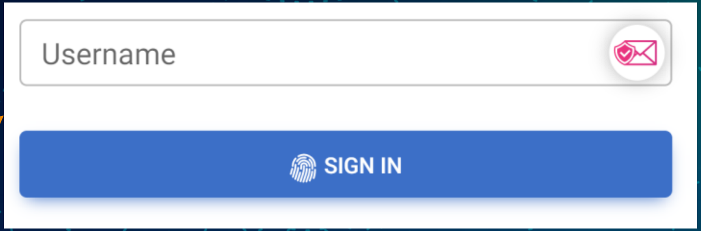


Email address

Password

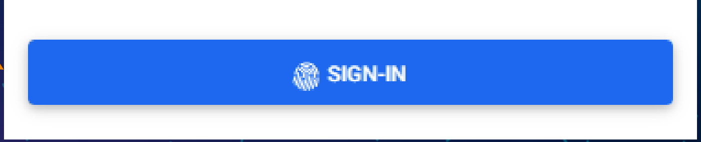
SIGN IN

+ optional second factors



Username

SIGN IN



SIGN-IN

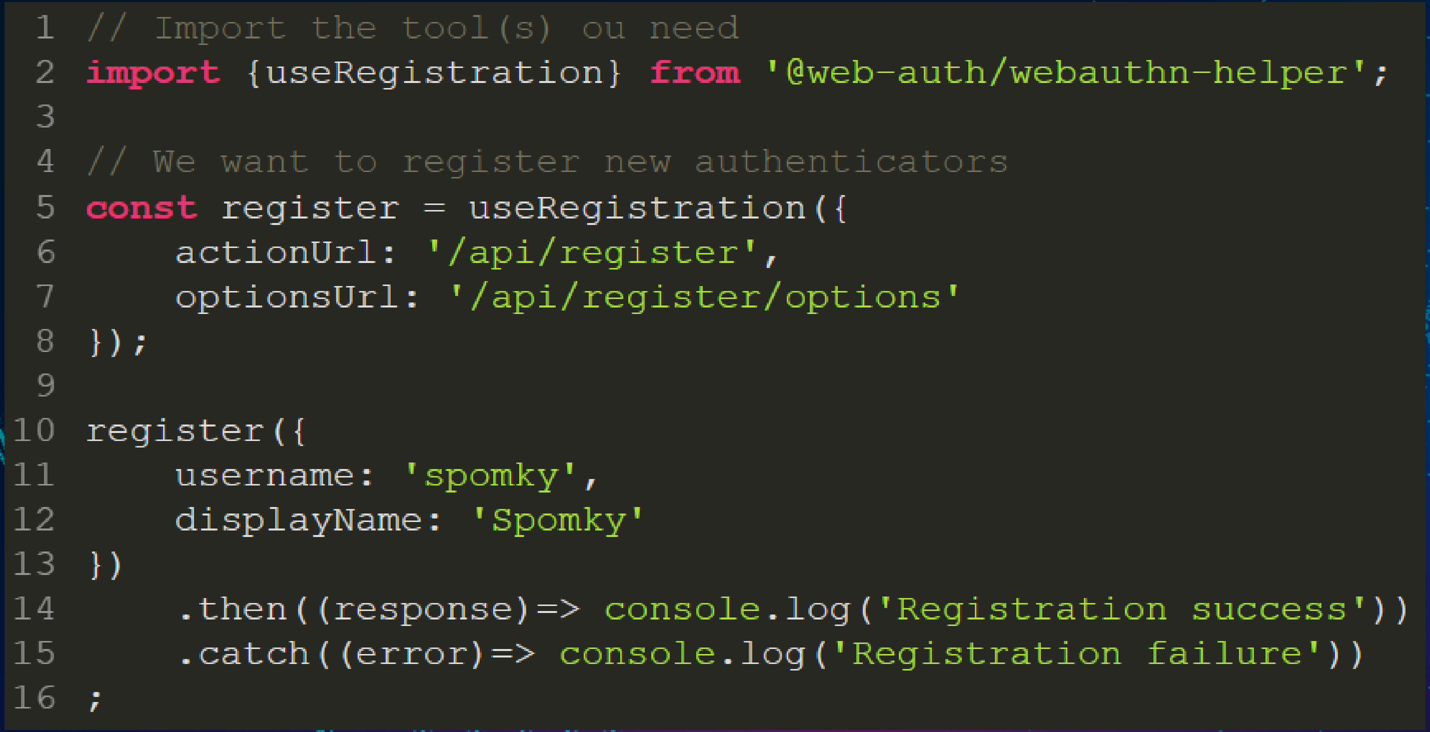
You should plan a smooth transition for you users!

Frontend

Javascript packages:

- ✓ @web-auth/webauthn-helper
- ✓ @simplewebauthn/browser

Frontend



```javascript
// Import the tool(s) ou need
import {useRegistration} from '@web-auth/webauthn-helper';

// We want to register new authenticators
const register = useRegistration({
    actionUrl: '/api/register',
    optionsUrl: '/api/register/options'
});

register({
    username: 'spomky',
    displayName: 'Spomky'
})
    .then((response)=> console.log('Registration success'))
    .catch((error)=> console.log('Registration failure'))
;
```

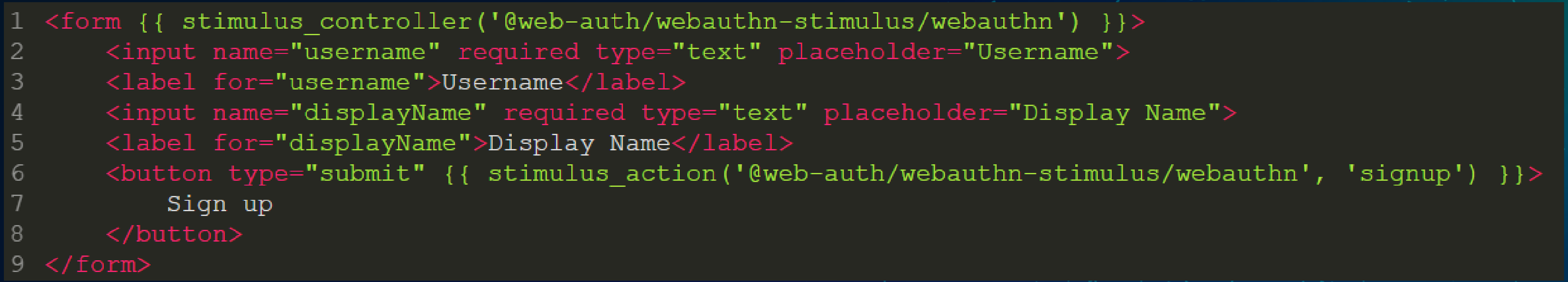
Frontend

Webauthn + Symfony UX = ♥

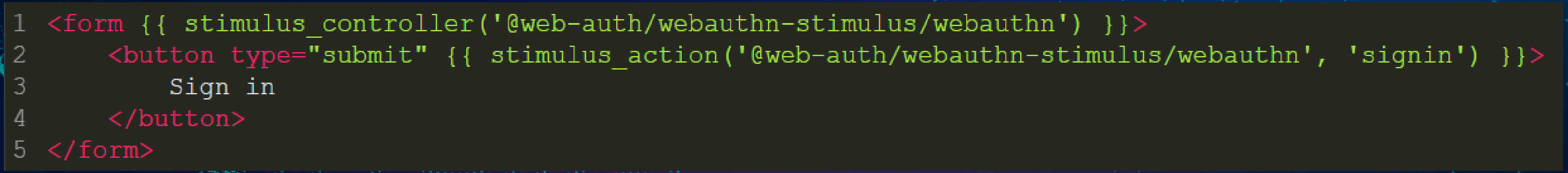
✓ **web-auth/webauthn-stimulus**

Go discover Symfony UX on ux.symfony.com!

Frontend



```
<form {{ stimulus_controller('@web-auth/webauthn-stimulus/webauthn') }}>
    <input name="username" required type="text" placeholder="Username">
    <label for="username">Username</label>
    <input name="displayName" required type="text" placeholder="Display Name">
    <label for="displayName">Display Name</label>
    <button type="submit" {{ stimulus_action('@web-auth/webauthn-stimulus/webauthn', 'signup') }}>
        Sign up
    </button>
</form>
```



```
<form {{ stimulus_controller('@web-auth/webauthn-stimulus/webauthn') }}>
    <button type="submit" {{ stimulus_action('@web-auth/webauthn-stimulus/webauthn', 'signin') }}>
        Sign in
    </button>
</form>
```

Many other features

- ✓ Extensions
- ✓ Attestation formats and statements
- ✓ Authenticator Selection Criteria
- ✓ User verification modes
- ✓ User Attributes **IS_USER_PRESENT** and **IS_USER_VERIFIED** (Symfony)

06 – Wrap up

API PLATFORM CONFERENCE

Wrap up

- ✓ No password, but **asymmetric key pairs and digital signatures**
- ✓ IMPOSSIBLE reuse on several applications: **1 domain = 1 pair of keys**
- ✓ Completely **ineffective phishing**
- ✓ Important information **never leaves the authenticator**
- ✓ **No sensitive data**, you store **public keys** and basic metadata
- ✓ Challenges and counters are **anti-replay protections**

Wrap up

Improved user experience for login

- ✓ **Less keyboard input** (username and possibly PIN code)
- ✓ **No need to remember** anything
- ✓ **No need for third-party** applications
- ✓ TOTP, SMS or email as 2nd factor become **useless**
- ✓ Multi-device (**Passkey**) and possibility to **register several authenticators**

Conclusion

Easy to implement.

Easy to adapt.

Easy to use.

Conclusion

Hard to hack.

Thank you!

Follow me on social media

🐦 @FlorentMorselli

🌐 github.com/Spomky

Special thanks to contributors and sponsors!

Try it:

https://webauthn.spomky-labs.com/