

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

API PLATFORM CONFERENCE





01

Password Issues

02

What about Api Platform?

03

How to solve password issues?

04

How Webauthn Works?

05

Application Integration?

About me

Florent Morselli
aka Spomky

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



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

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

Password Issues

What **size**?

What **characters**?

Storage?

Hashing function?

Password Issues

Bruteforce protection?

Keyloggers detection?

Expiration policy?

Ban sharing?

...

Password Issues



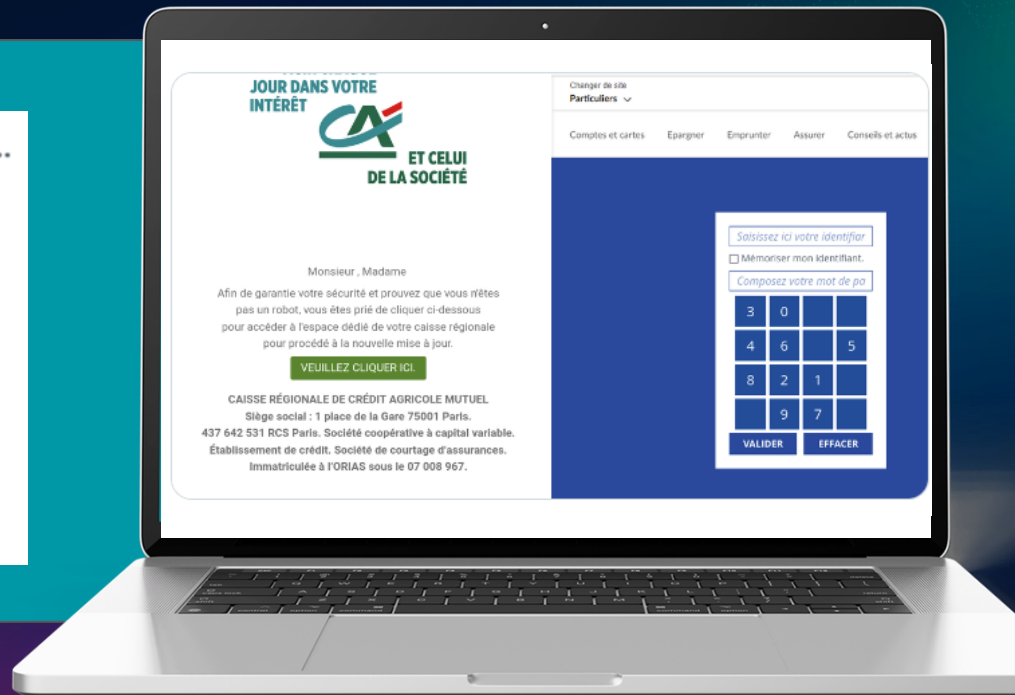
Phishing



Mikolajek
@_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



Phishing



Mikolajek
@_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

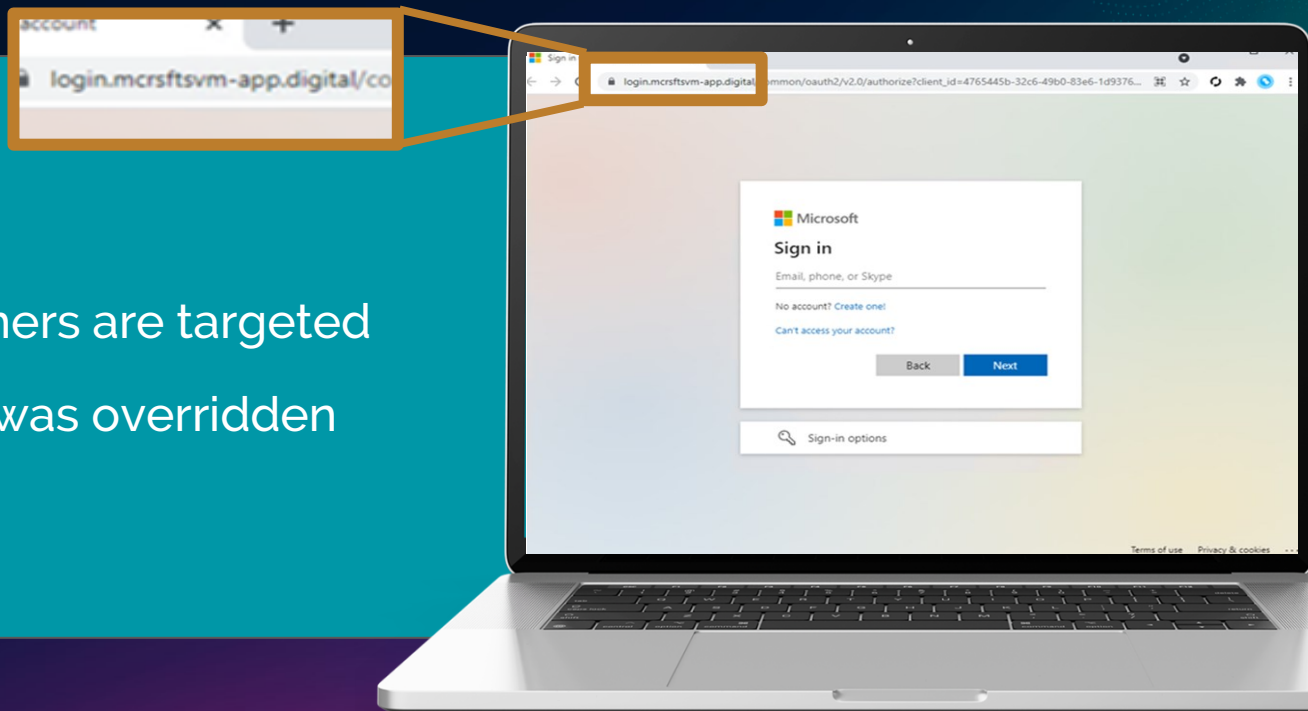
Phishing



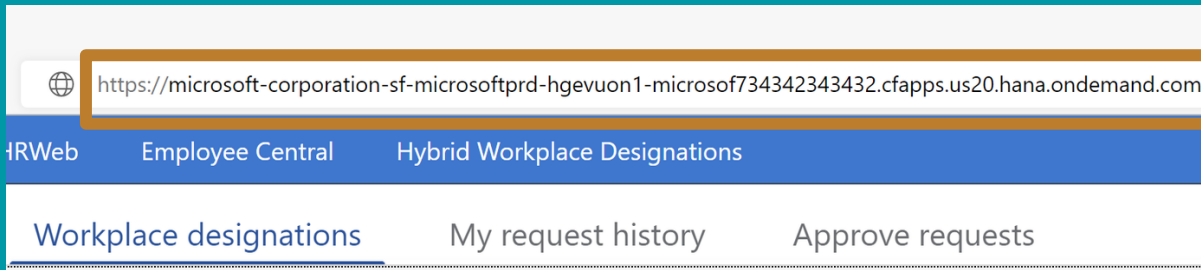
7 Retweets 3 Tweets cités 47 J'aime

Phishing

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



Phishing



Phishing



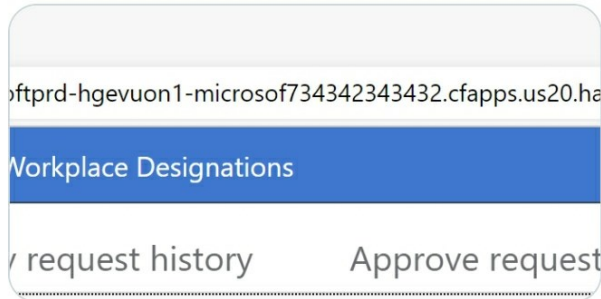
Eric Lawrence 🎸

@ericlaw

BigCorp: You should recognize a phishing attack.

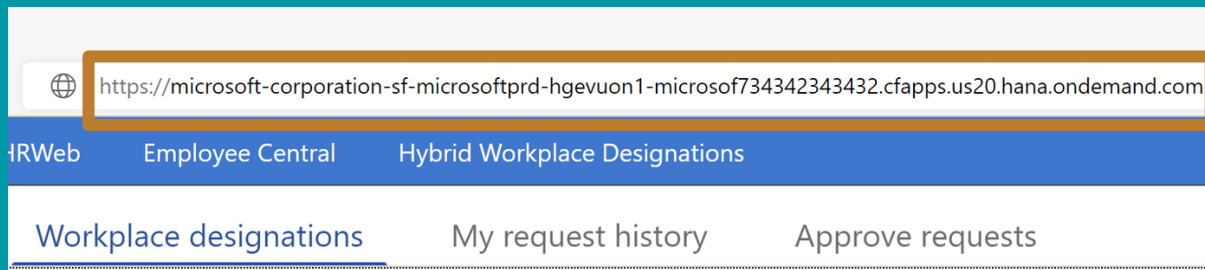
BigCorp: This is a legitimate and mandatory URL.

[Traduire le Tweet](#)



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

1 063 Retweets 55 Tweets cités 5 591 J'aime



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*.

You should really change your password if 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."

03 – How to solve password issues?

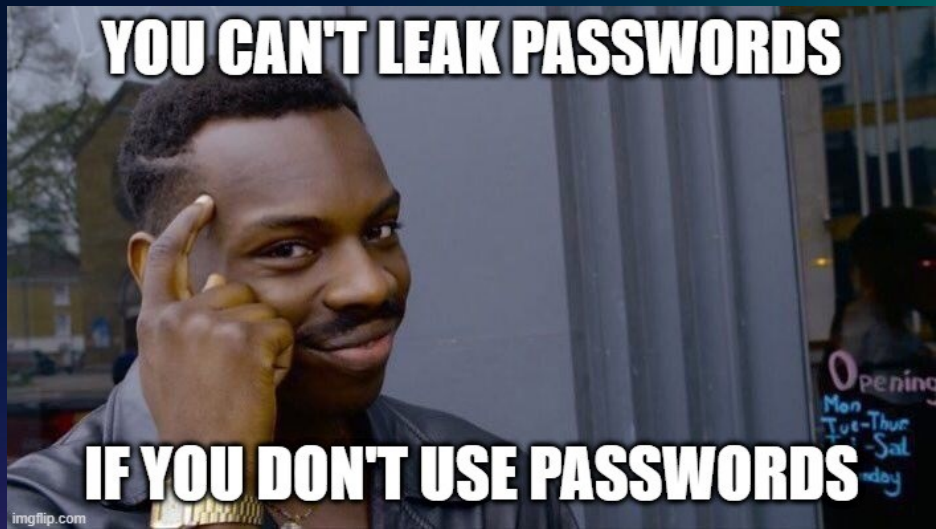
How to solve password issues?

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

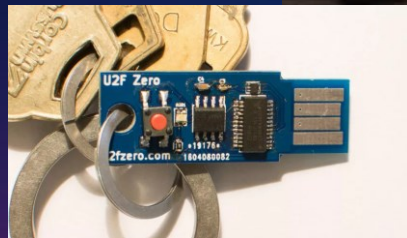
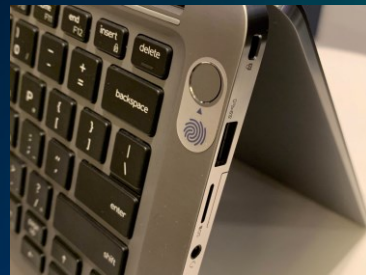
API that enables the creation and use of **strong public key-based** credentials by web applications.

Webauthn

Cryptographic operations are delegated to **authenticators**

Webauthn Authenticators

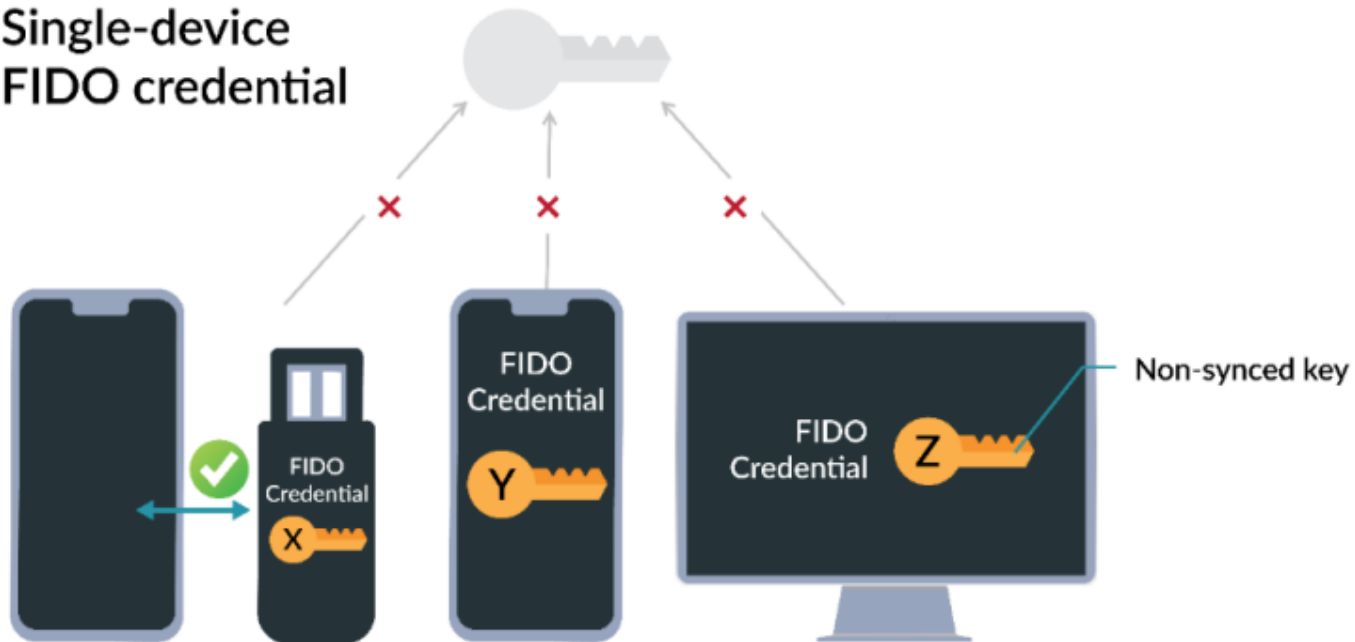
- ✓ **Platform:** embedded in the device
- ✓ **Roaming:** NFC, USB, Lightning, Bluetooth LE, caBLE



Old U2F Security Keys are compatible!

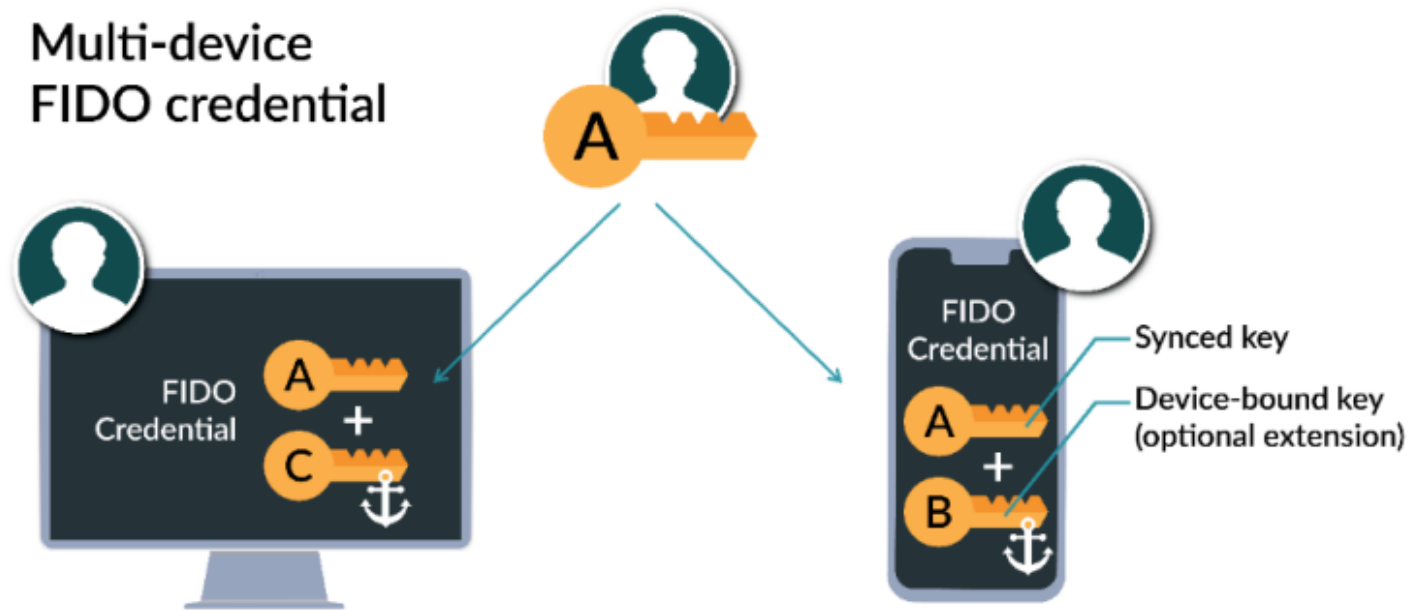
FIDO Credentials

Single-device
FIDO credential



Multi-device FIDO Credentials

Multi-device
FIDO credential



« 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


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.

Usage

% of all users  ?

Global 93.15% + 0.69% = 93.84%

unprefixed: 93.15% + 0.69% = 93.84%

Current aligned Usage relative Date relative Filtered All 

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									

04 – How Webauthn Works?

Webauthn Ceremonies

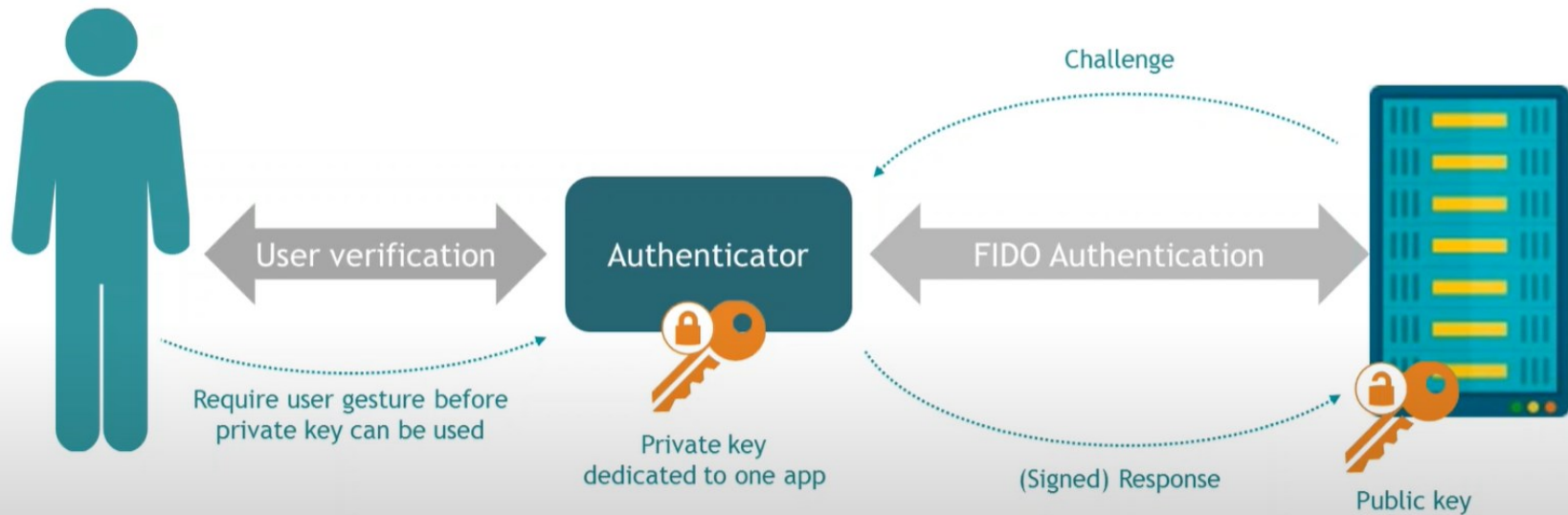
- ✓ **Creation:** registration of an authenticator (new or existing account)
- ✓ **Request:** use of an existing authenticator

Webauthn Ceremonies

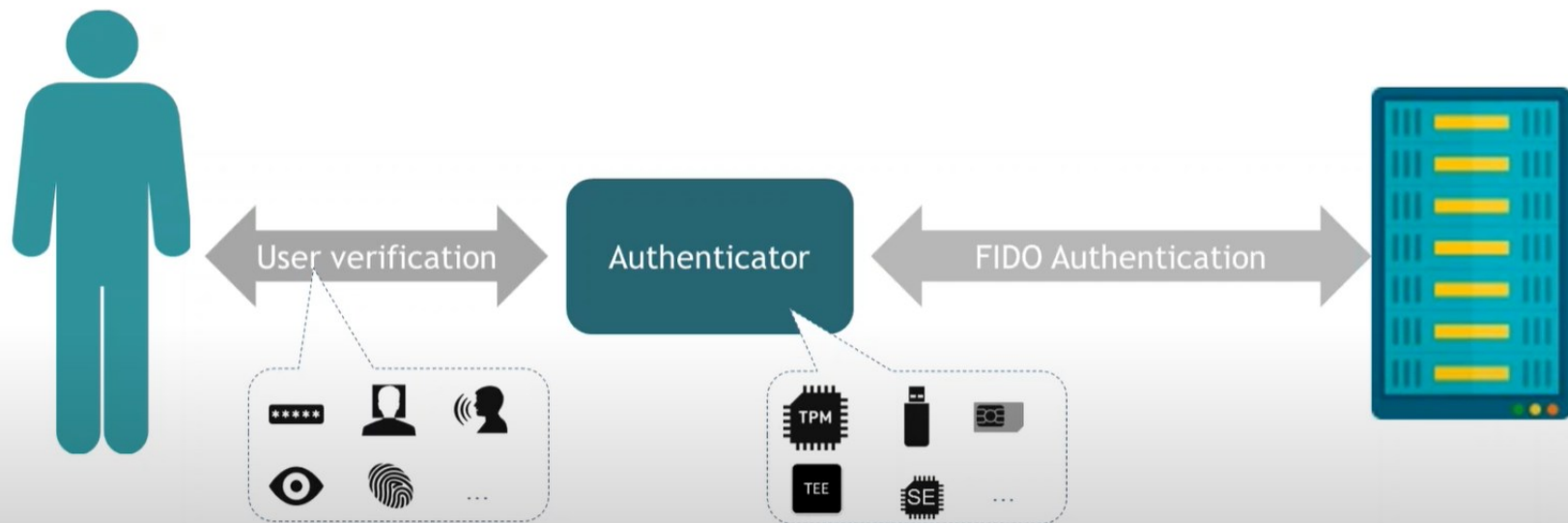
Each ceremony requires **2 HTTP requests**:

- ✓ **Obtaining options** (challenge + security policy)
- ✓ **Sending the result** of the challenge (+ credentials)

Webauthn Ceremonies



Webauthn Ceremonies



Webauthn Ceremonies

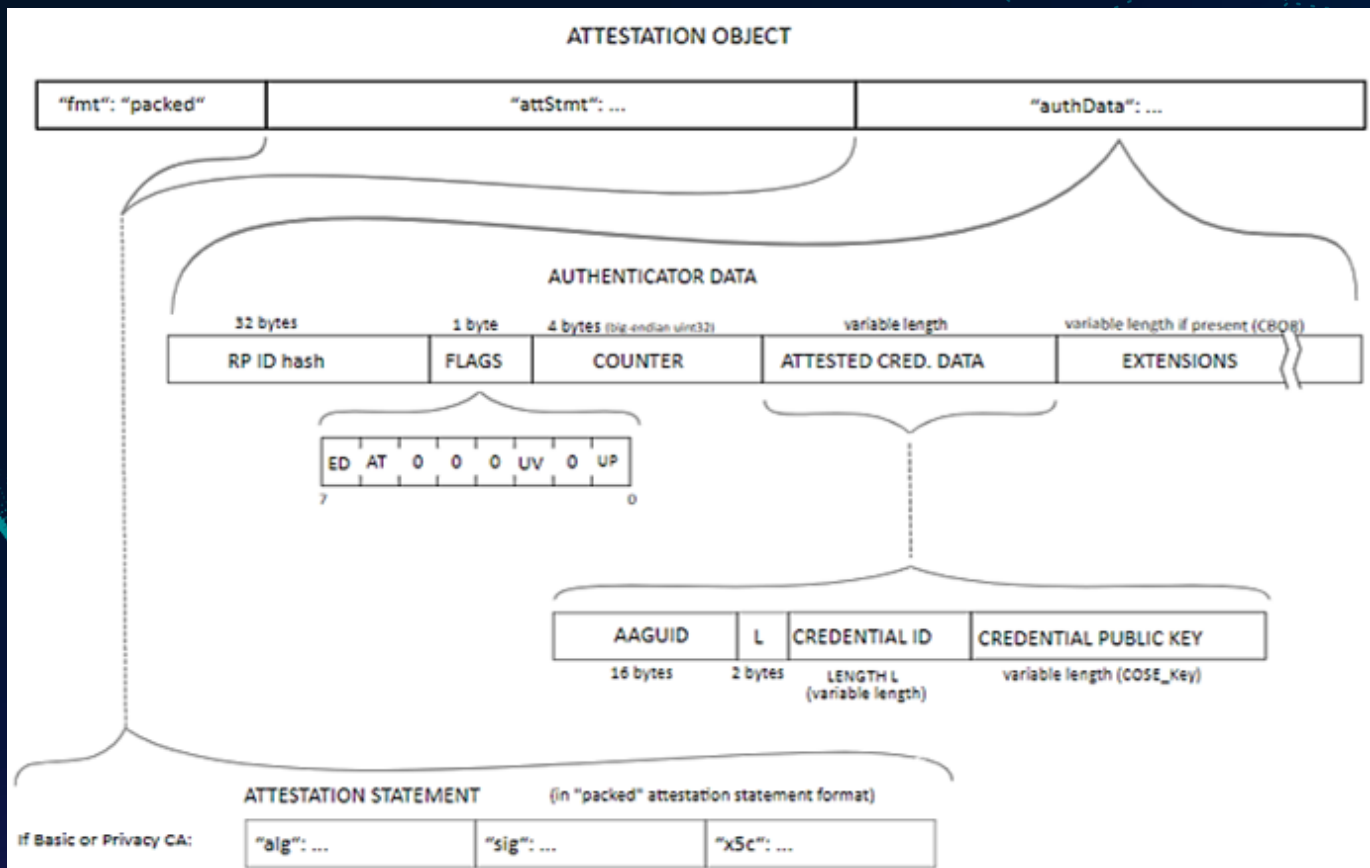
```
1 {  
2   "username": "spomky",  
3   "displayName": "Spomky"  
4 }
```

```
1 {  
2   "challenge": "scIBUcV9TTCd2SFskRCYH+o4RIDHrqSOaBs0IHQ9xwo",  
3   "rp": {  
4     "name": "My API",  
5     "id": "example.com"  
6   },  
7   "user": {  
8     "name": "spomky",  
9     "displayName": "Spomky",  
10    "id": "ABCDEFGH-0123456789"  
11  },  
12  "pubKeyCredParams": [  
13    {  
14      "type": "public-key",  
15      "alg": -7  
16    },  
17    {  
18      "type": "public-key",  
19      "alg": -35  
20    }  
21  ],  
22  "attestation": "none"  
23 }
```

Webauthn Ceremonies

```
1 {  
2   "id": "EvqdkRD3eNWnByCzKCZaRlwun85x...kphDcXCnsDoWSkdUdkDaYm33o9fuuRuMg",  
3   "rawId": "EvqdkRD3eNWnByCzKCZaRlwu...OdeCXBGZyCPMokphDcXCnsDoWSkdUdkDaYm33o9fuuRuMg",  
4   "type": "public-key",  
5   "response": {  
6     "attestationObject": "o2NmbXRkbn9uZWdhdHRTdG10oGhhdXRoRGF0...CYqlMGpUKA5",  
7     "clientDataJSON": "eyJ0eXB1Ijoid2ViYXV0aG4uY3JlYXRlI...bmdlIjoiUE1VRWlrd"  
8   }  
9 }
```

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 🗝️
- ✓ Custom Flex Recipes Server

Application Integration

Glue Code

**Bundle and firewall
configuration**

Frontend

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 unique random value** and associate it to your users.

Application Integration

A diagram illustrating the components of Application Integration. It features three white rectangular boxes arranged horizontally. The central box is the largest and contains the text 'Glue Code'. To its left is a smaller box containing 'Bundle and firewall configuration', and to its right is another smaller box containing 'Frontend'. The boxes are set against a dark blue background with a teal wave pattern on the right and a teal dotted pattern on the left. A teal horizontal bar is positioned below the boxes.

**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 \Leftrightarrow Symfony User

- ✓ Webauthn User Entity class is **provided by the bundle**
- ✓ Conversion is easy

Webauthn User Repository

Webauthn User Repository \Leftrightarrow Symfony User Repository

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

Webauthn User Repository

```
1 <?php
2
3 declare(strict_types=1);
4
5 namespace Webauthn\Bundle\Repository;
6
7 use Webauthn\PublicKeyCredentialUserEntity;
8
9 interface PublicKeyCredentialUserEntityRepository
10 {
11     public function findOneByUsername(string $username): ?PublicKeyCredentialUserEntity;
12
13     public function findOneByUserHandle(string $userHandle): ?PublicKeyCredentialUserEntity;
14
15     public function generateNextUserEntityId(): string;
16
17     public function saveUserEntity(PublicKeyCredentialUserEntity $userEntity): void;
18 }
```

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

```
1 <?php
2
3 namespace App\Entity;
4
5 use App\Repository\PublicKeyCredentialSourceRepository;
6 use Doctrine\ORM\Mapping as ORM;
7 use Webauthn\PublicKeyCredentialSource;
8
9 #[ORM\Table(name: 'webauthn_credentials')]
10 #[ORM\Entity(repositoryClass: PublicKeyCredentialSourceRepository::class)]
11 class WebauthnCredential extends PublicKeyCredentialSource
12 {
13     ...
14 }
```

Webauthn Credential Repository

```
1 <?php
2
3 declare(strict_types=1);
4
5 namespace Webauthn;
6
7 interface PublicKeyCredentialSourceRepository
8 {
9     public function findOneByCredentialId(string $publicKeyCredentialId): ?PublicKeyCredentialSource;
10
11     /**
12      * @return PublicKeyCredentialSource[]
13      */
14     public function findAllForUserEntity(PublicKeyCredentialUserEntity $publicKeyCredentialUserEntity): array;
15
16     public function saveCredentialSource(PublicKeyCredentialSource $publicKeyCredentialSource): void;
17 }
```

Application Integration

Glue Code

**Bundle and firewall
configuration**

Frontend

Bundle/Firewall Configuration

```
1 # config/packages/webauthn.yaml
2 webauthn:
3     credential_repository: 'App\Repository\PublicKeyCredentialSourceRepository'
4     user_repository: 'App\Repository\PublicKeyCredentialUserEntityRepository'
5     creation_profiles:
6         default:
7             rp:
8                 name: '%env(RELAYING_PARTY_NAME) %'
9                 id: '%env(RELAYING_PARTY_ID) %'
10    request_profiles:
11        default:
12            rp_id: '%env(RELAYING_PARTY_ID) %'
```

Bundle/Firewall Configuration

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

Bundle/Firewall Configuration

Depends on the type of application.

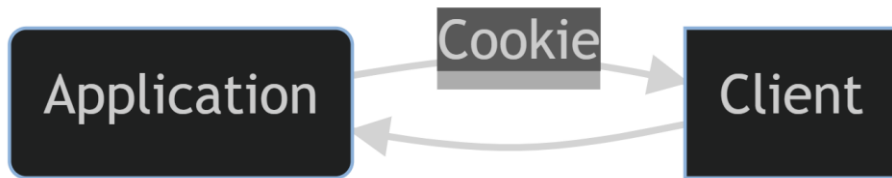
Stateful and **stateless** configurations are supported



Bundle/Firewall Configuration

- ✓ Traditional application with sessions
- ✓ Sessions shall be configured in your application

<https://symfony.com/doc/current/session.html>



Bundle/Firewall Configuration

```
1 security:
2   enable_authenticator_manager: true
3   providers:
4     default:
5       id: App\Security\UserProvider
6   firewalls:
7     main:
8       pattern: ~/
9       webauthn: ~
```

Bundle/Firewall Configuration

```
1 security:
2   enable_authenticator_manager: true
3   providers:
4     default:
5       id: App\Security\UserProvider
6   firewalls:
7     main:
8       pattern: ~/
9       webauthn:
10         registration:
11           enabled: true
12           routes:
13             options_path: '/api/register/options'
14             result_path: '/api/register'
15         authentication:
16           enabled: true # true dy default
17           routes:
18             options_path: '/api/login/options'
19             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

```
1 security:
2     firewalls:
3         login:
4             pattern: ~/api/(login|register)
5             webauthn:
6                 success_handler: 'lexik_jwt_authentication.handler.authentication_success'
7                 failure_handler: 'lexik_jwt_authentication.handler.authentication_failure'
8
9     api:
10         pattern: ^/
11         stateless: true
12         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

```
1 security:
2     firewalls:
3         login:
4             pattern: ~/api/(login|register)
5             stateless: true
6             webauthn:
7                 success_handler: 'lexik_jwt_authentication.handler.authentication_success'
8                 failure_handler: 'lexik_jwt_authentication.handler.authentication_failure'
9                 options_storage: 'Webauthn\Bundle\Security\Storage\CacheStorage'
10
11     api:
12         pattern: ^/
13         stateless: true
14         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

```
1 webauthn:
2   controllers:
3     enabled: true
4     creation:
5       from_user_account: # Endpoints accessible by the user itself
6         options_path: '/profile/security/devices/add/options'
7         result_path: '/profile/security/devices/add'
8         user_entity_guesser: 'Webauthn\Bundle\Security\Guesser\CurrentUserEntityGuesser'
9       from_admin_dashboard: # Endpoint accessible by an administrator
10        options_path: '/admin/security/user/{user_id}/devices/add/options'
11        result_path: '/admin/security/user/{user_id}/devices/add'
12        user_entity_guesser: 'App\Guesser\FromQueryParameterGuesser'
```

Bundle/Firewall Configuration

```
1 <?php
2
3 declare(strict_types=1);
4
5 namespace App\Guesser;
6
7 use Assert\Assertion;
8 use Symfony\Component\HttpFoundation\Request;
9 use Webauthn\Bundle\Repository\PublicKeyCredentialUserEntityRepository;
10 use Webauthn\Bundle\Security\Guesser\UserEntityGuesser;
11 use Webauthn\PublicKeyCredentialUserEntity;
12
13 final class FromQueryParameterGuesser implements UserEntityGuesser
14 {
15     public function __construct(
16         private PublicKeyCredentialUserEntityRepository $userEntityRepository
17     ) {
18     }
19
20     public function findUserEntity(Request $request): PublicKeyCredentialUserEntity
21     {
22         $userHandle = $request->query->get('user_id');
23         Assertion::string($userHandle, 'User entity not found. Invalid user ID');
24         $user = $this->userEntityRepository->findOneByUserHandle($userHandle);
25         Assertion::assertInstanceOf($user, PublicKeyCredentialUserEntity::class, 'User entity not found.');
```

```
26
27         return $user;
28     }
29 }
```

Bundle/Firewall Configuration

```
1 security:
2   access_control:
3     - { path: ^/profile, roles: ROLE_USER, requires_channel: 'https' }
4     - { path: ^/admin, roles: ROLE_ADMIN, requires_channel: 'https' }
5     - ...
```

Application Integration

A diagram illustrating the components of application integration. It features three white rectangular blocks on a dark blue background. A large central block is labeled 'Frontend'. To its left is a smaller block labeled 'Glue Code'. To its right is a block labeled 'Bundle and firewall configuration'. The 'Glue Code' and 'Bundle and firewall configuration' blocks are positioned at the same vertical level, while the 'Frontend' block is slightly higher. The background includes decorative elements: a teal wavy line with a dot pattern in the top right and teal dotted triangular shapes in the bottom left and right corners of the side blocks.

Glue Code

Frontend

**Bundle and firewall
configuration**

Frontend

The diagram illustrates a frontend transition for a login form. On the left, a form with 'Email address' and 'Password' fields and a 'SIGN IN' button is shown. An orange arrow points from this form to a new form on the right. The new form has a 'Username' field and a 'SIGN IN' button. A second orange arrow points from the new form to a third form below it, which only contains a 'SIGN-IN' button. The background features a blue and green gradient with a faint spiderweb pattern.

Email address

Password

SIGN IN

+ second factors (if any)

Username

SIGN IN

SIGN-IN

You should plan a smooth transition for you users!

Frontend

Javascript packages:

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

Frontend

```
1 // Import the tool(s) ou need
2 import {useRegistration} from '@web-auth/webauthn-helper';
3
4 // We want to register new authenticators
5 const register = useRegistration({
6   actionUrl: '/api/register',
7   optionsUrl: '/api/register/options'
8 });
9
10 register({
11   username: 'spomky',
12   displayName: 'Spomky'
13 })
14   .then((response)=> console.log('Registration success'))
15   .catch((error)=> console.log('Registration failure'))
16 ;
```

Frontend

Webauthn + Symfony UX = ❤️

✓ **web-auth/webauthn-stimulus**

Go discover Symfony UX on ux.symfony.com!

Frontend

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

```
1 <form {{ stimulus_controller('@web-auth/webauthn-stimulus/webauthn') }}>
2   <button type="submit" {{ stimulus_action('@web-auth/webauthn-stimulus/webauthn', 'signin') }}>
3     Sign in
4   </button>
5 </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

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!

Any questions?



@FlorentMorselli



github.com/Spomky

Try it:

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

Special thanks to contributors and sponsors!

API PLATFORM
CONFERENCE