



How to integrate **FOSUserBundle** in **eZ Platform** to externalize Users in **MongoDB**



<https://github.com/Novactive/NovaeZFOSUserBundle>

Novactive

Premier Sponsor

Novactive Group

- **Novactive France**

- Paris ~ 19 years old
- Toulon ~ 2 years old

- **Novactive Tunis** ~ 8 years old

- **Novactive Canada Inc.**

- Montréal ~ 4 years

- **Novactive Corporation**

- San Francisco ~ 1 year old

Worldwide Team 80 employees

- Digital Consulting
- Web Development
- e-Commerce
- Web Marketing
- Mobile Apps

Novactive and eZ Systems

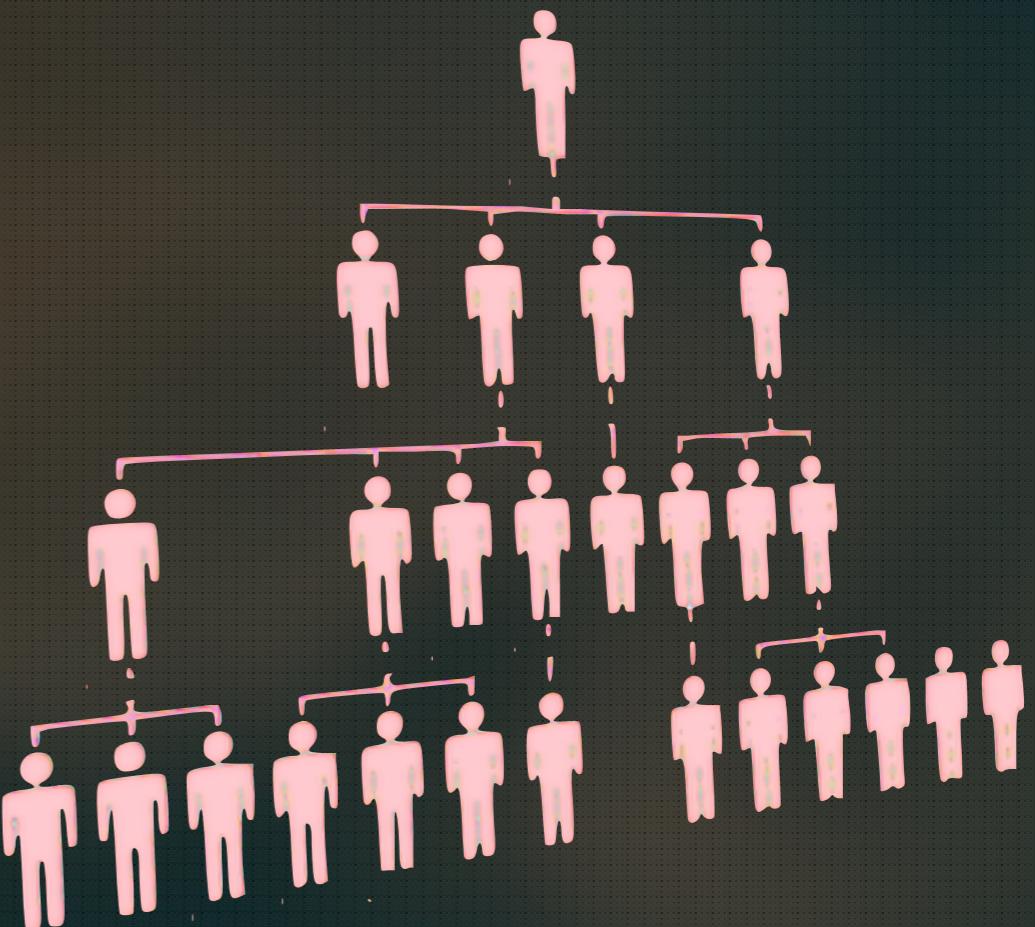
- Over 10 years of partnership
- First eZ project developed in 2005
- 70% of our projects are eZ Publish based technology
- Over 150 projects delivered with eZ Publish
- More than 40 websites currently in ongoing maintenance on eZ Publish
- Provider of professional extensions for the eZ Market



Let's get started!

User management in ez

- Simple
- Extensible
- Robust
- Extremely powerful roles and policies system
- Almost no limit
- Managed like Contents



User Group 1

Role 1

User A

Policy 10

User B

Policy 11

Policy 12

Policy 13

User Group 2

Role 2

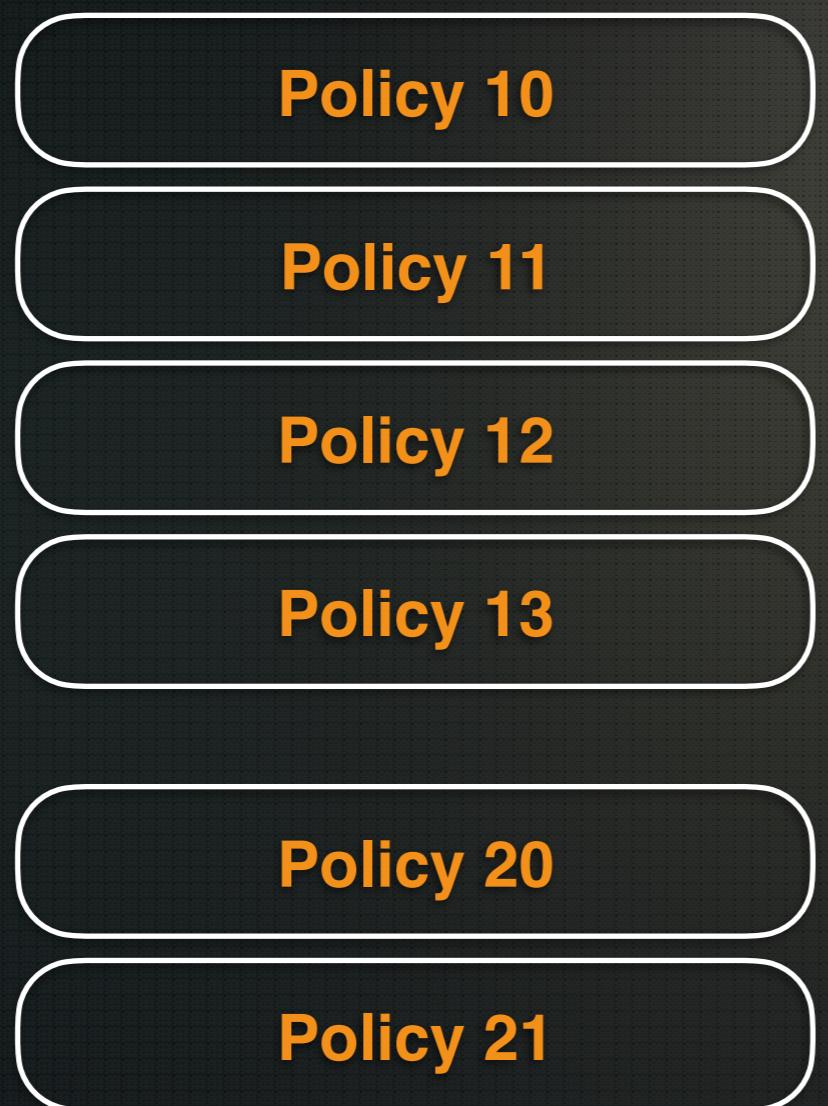
User C

Policy 20

User D

Policy 21

User B



Role 1

Role 2

A User is not a Content

- Performance issue
- Dependency between
 - User repository
 - Content repository
- Hard to manage a website with
 - 300 000 articles
 - 7 millions Subscribers/Members



A good approach

Users externalization
keeping the policy system

A good approach

→ Keeping only one UserAccount Field Type in eZ

→ And “join” on an *email* or on an *id*

→ **Pros**

→ All the built-in features of eZ stay functional

→ **Cons**

→ Need to extend the Administration Interface

→ Still using the famous ***ezcontentobject_attribute*** table



Do we have a better
approach with
eZ Platform?

“Symfony provides native support for multiple user providers. This makes it easy to integrate any kind of login handlers, including SSO and existing 3rd party bundles”

– The Amazing eZ Documentation



eZ Platform

DESIGN AND DEVELOP CONTENT-DRIVEN
WEBSITES AND APPLICATIONS

A Top-Notch Developer Experience Built
on the Symfony Full-Stack Framework



Thanks eZ Platform

Thanks Symfony and the community !



FOS User Bundle

Friends Of Symfony User Bundle

FOS User Bundle

- The **best** and **most useful** Symfony2 bundle

<http://symfony.com/blog/the-30-most-useful-symfony-bundles-and-making-them-even-better>

- More than **3.1 million installations**

<https://packagist.org/packages/friendsofsymfony/user-bundle>

- Ready and simple to use

- Reliable and unit tested

- Documentation on [symfony.com](#)



FOS User Bundle

- ✓ Registration, with an optional email confirmation
- ✓ Password reset
- ✓ Profile edition over the Symfony Forms
- ✓ Group of users
- ✓ Invitations are coming
- ✓ Custom storage layer



MongoDB ready

- ✓ Through Doctrine ODM
- ✓ CouchDB support available



We have
Everything we need!

eZ

+

Symfony

+

External User provider

<https://doc.ez.no/display/EZP/How+to+authenticate+a+user+with+multiple+user+providers>



Some inputs

- **API User**

- Stored into the eZ database
- Contains the UserAccount Fields
 - Is connected to the Role and Permissions

- **UserWrapped**

- A combination of
 - 1 API User
 - 1 External User (also called the WrappedUser)



For Instance

UserWrapped

API User

In eZ

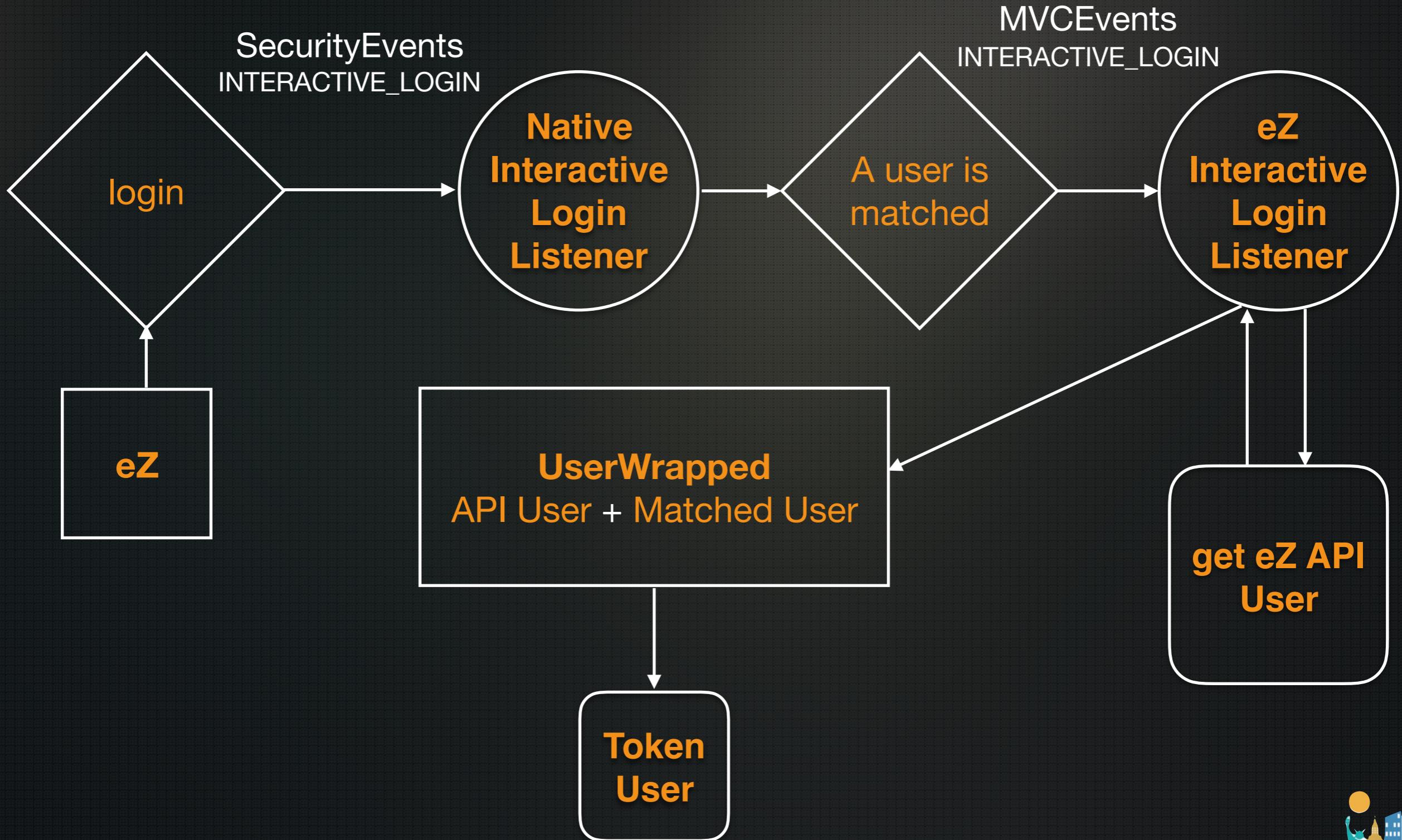
linked to the
Policies

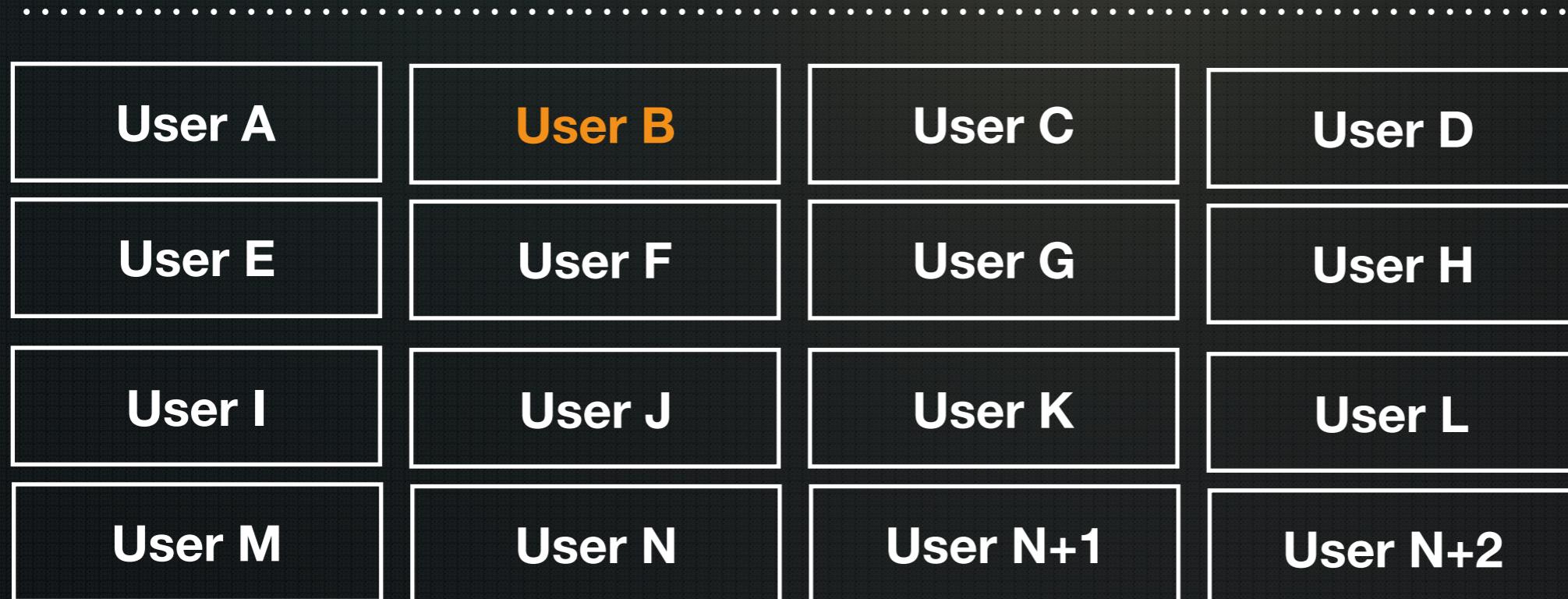
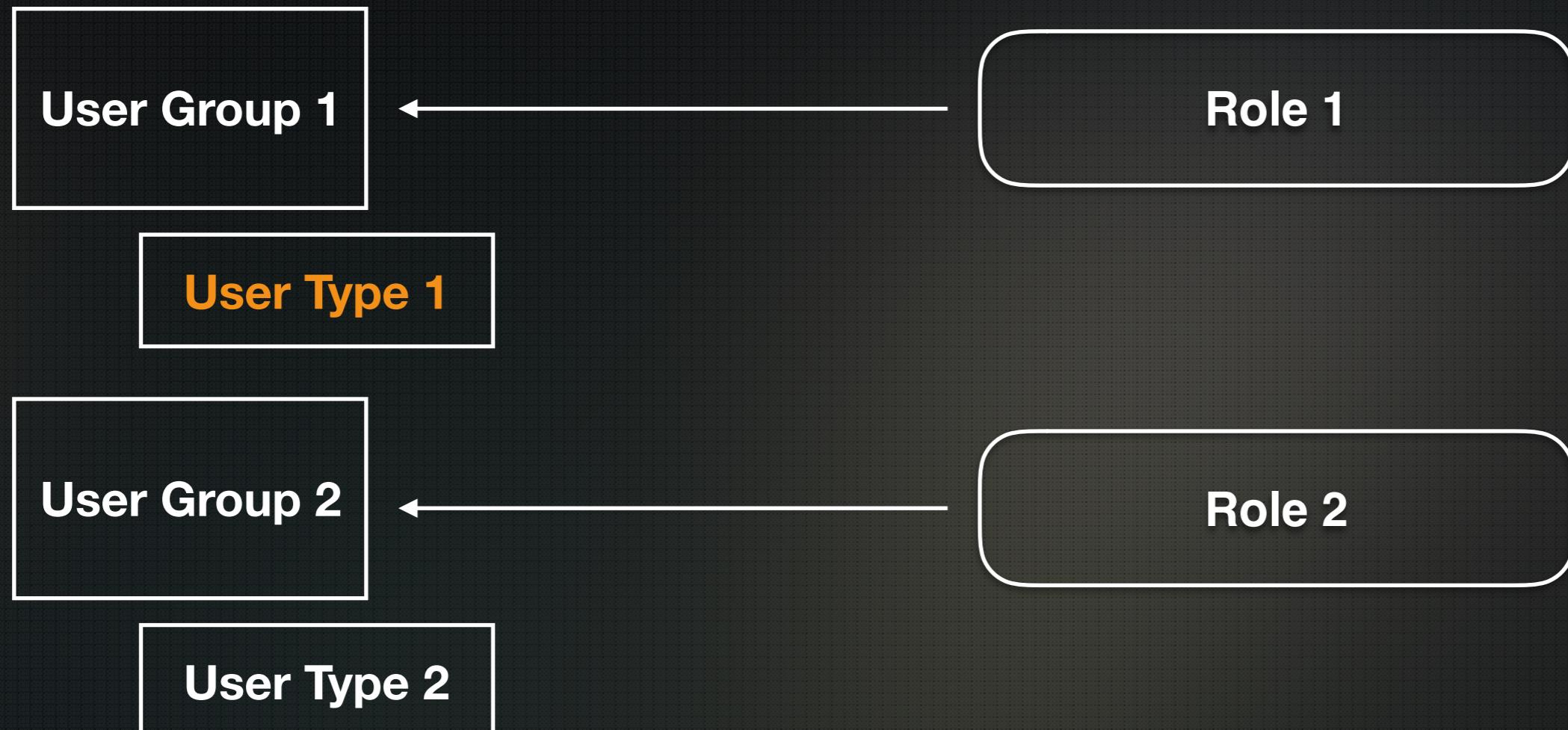
WrappedUser

In Mongo DB



Simplified process





E
Z
D
B
M
O
N
G
O
D
B

Let's do it

Assumptions

- ✓ An eZ Platform installation working
- ✓ Composer installed
- ✓ A MongoDB instance working

Installation

```
$ php composer.phar require doctrine/mongodb-odm-bundle:'3.0.*@dev'  
$ php composer.phar require friendsofsymfony/user-bundle:~2.0@dev
```

```
// ezpublish/EzPublishKernel.php  
  
public function registerBundles()  
{  
    $bundles = array(  
        ...  
        new FOS\UserBundle\FOSUserBundle(),  
        new Doctrine\Bundle\MongoDBBundle  
\DoctrineMongoDBBundle(),  
    );  
}
```



FOS Configuration

```
# app/config/routing.yml
fos_user:
    resource: "@FOSUserBundle/Resources/config/
routing/all.xml"
```

```
# ezpublish/config/config.yml

fos_user:
    db_driver: mongodb
    firewall_name: ezpublish_front
    user_class: Novactive\Bundle\eZF0SUserBundle
\Document\User
```



Extend FOS Bundle

```
class NovaefOSUserBundle extends Bundle
{
    /**
     * Extends FOS
     *
     * @return string
     */
    public function getParent()
    {
        return 'FOSUserBundle';
    }
}
```



ODM Configuration

```
# ezpublish/config/config.yml

doctrine_mongodb:
    connections:
        default:
            server: 'mongodb://localhost:27017'
            options: {}
    default_database: 'EZCONFNYC2015'
    document_managers:
        default:
            auto_mapping: true
```



User Document

```
namespace Novactive\Bundle\eZFOSUserBundle\Document;

use FOS\UserBundle\Model\User as BaseUser;
use Doctrine\ODM\MongoDB\Mapping\Annotations as MongoDB;

/**
 * Class User
 * @MongoDB\Document
 */
class User extends BaseUser
{
    /**
     * Identifier
     *
     * @var int Identifier
     *
     * @MongoDB\Id(strategy="auto")
     */
    protected $id;

    /**
     * @var bool
     *
     * @MongoDB\Boolean
     */
    protected $isCertified;

    /**
     * @var array
     *
     * @MongoDB\Collection
     */
    protected $certifiedDates;

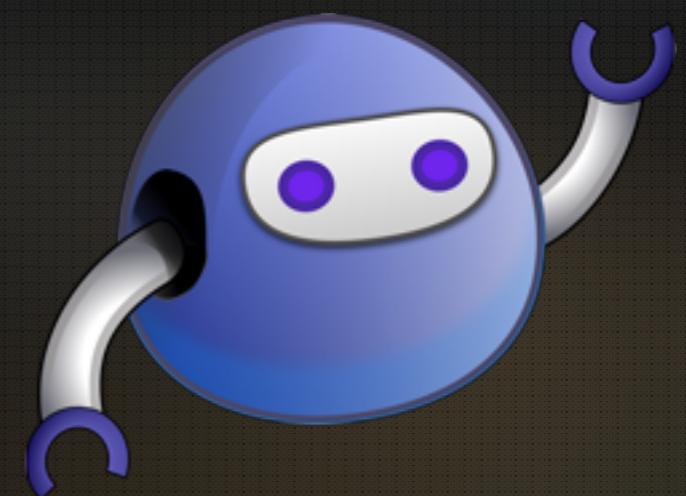
    .....
}
```

User Document

```
php ezpublish/console doctrine:mongodb:schema:create --index
```

The screenshot shows the RoboMongo interface displaying the database structure for the 'User' collection. The database is named 'EZCONFNYC2015'. The 'User' collection contains three indexes: '_id_', 'usernameCanonical_1', and 'emailCanonical_1'. The indexes are represented by magnifying glass icons.

- EZCONFNYC2015
 - Collections (2)
 - System
 - User
 - Indexes (3)
 - _id_
 - usernameCanonical_1
 - emailCanonical_1

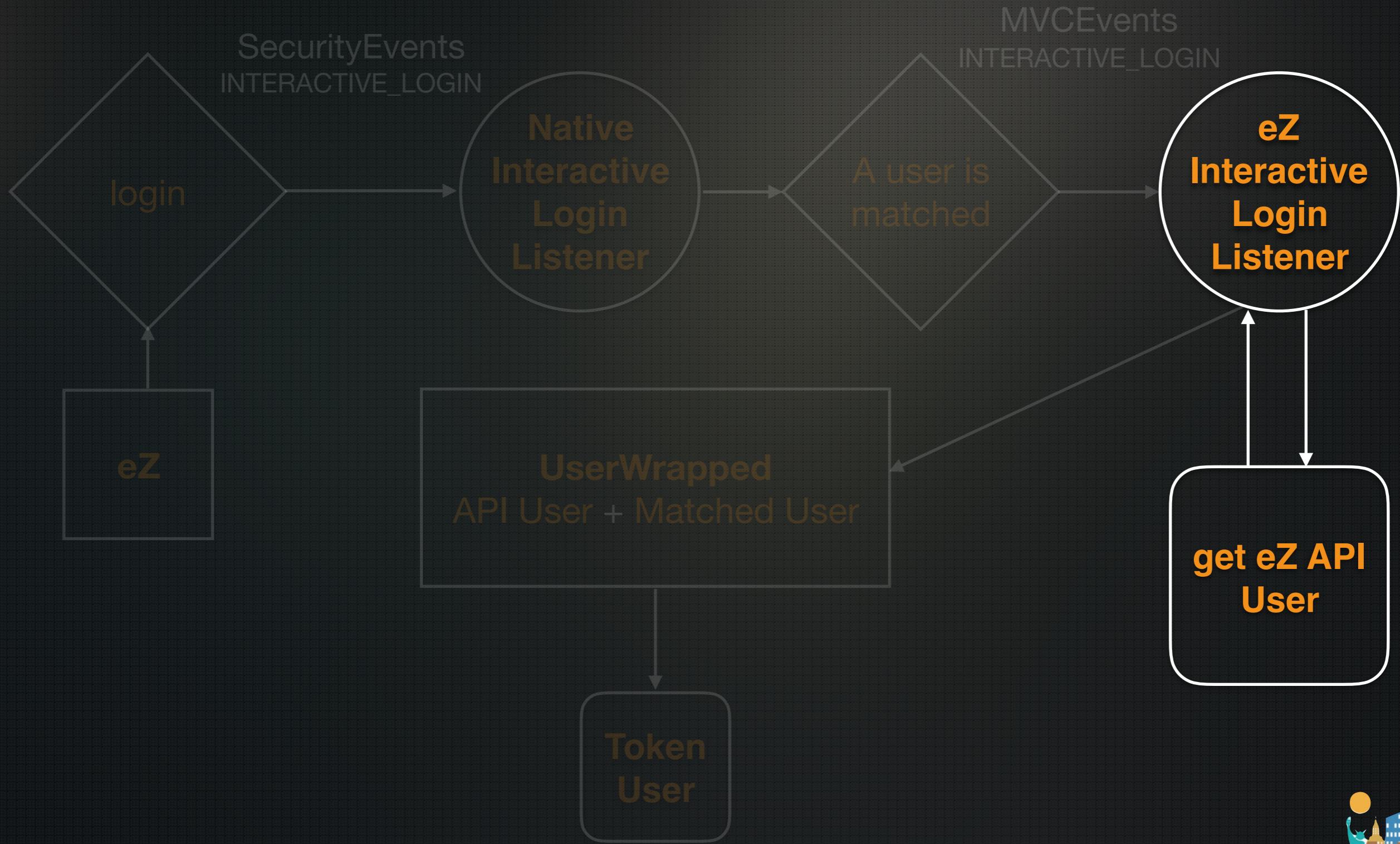


RoboMongo

Security Configuration

```
security:  
  
    encoders:  
        FOS\UserBundle\Model\UserInterface: bcrypt  
  
    role_hierarchy:  
        ROLE_ADMIN:                 ROLE_USER  
        ROLE_SUPER_ADMIN:          ROLE_ADMIN  
  
    providers:  
        chain_provider:  
            chain:  
                providers: [novaefos_provider, ezpublish]  
        novaefos_provider:  
            id: fos_user.user_provider.username  
        ezpublish:  
            id: ezpublish.security.user_provider
```

Simplified process



Mapping the API User

- ✓ Listen the **MVCEvents::INTERACTIVE_LOGIN**

services:

```
novactive.ezfosuser.interactive_event_listener:  
    class: Novactive\Bundle\eZF0SUserBundle\Listener  
    \InteractiveLoginListener  
    arguments: [@ezpublish.api.service.user]  
    tags:  
        - { name: kernel.event_subscriber }
```



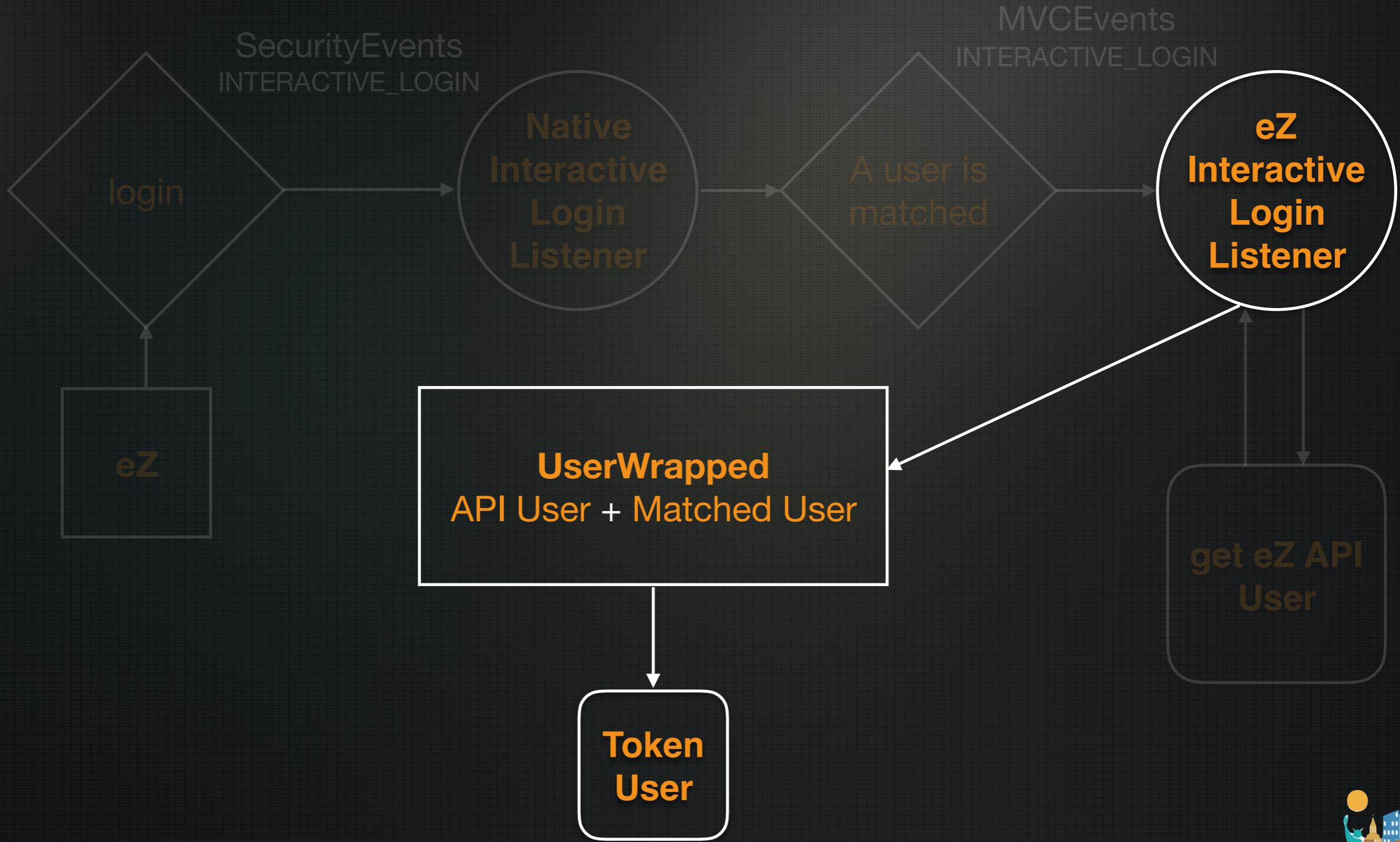
Set the API User

```
class InteractiveLoginListener implements EventSubscriberInterface
{
    .....
    public function onInteractiveLogin(InteractiveLoginEvent $event)
    {
        // We just load a generic user and assign it back to the event.
        // You may want to create users here, or even load predefined users depending on
        your own rules.

        $event->setApiUser(
            $this->userService->loadUserByLogin('UserType1')
        );

    }
}
```

Simplified process



Layout

A screenshot of a web browser window displaying a registration form. The URL in the address bar is `http://ezconfnyc...5.dev/register/`. The form fields are labeled "Email", "Username", "Password", and "Repeat password", each with an associated input field. A "Register" button is at the bottom. The browser interface includes standard navigation buttons (back, forward, search) and a tab bar.

[Log in](#)

Email

Username

Password

Repeat password

Register

Layout

- Thanks to the FOS Bundle overriding

```
{# eZFOSUserBundle/Resources/views/layout.html.twig #}
{% extends "eZDemoBundle::pagelayout.html.twig" %}

{% block content %}
<div>
    {% if is_granted("IS_AUTHENTICATED_REMEMBERED") %}
        {{ 'layout.logged_in_as'|trans({'%username%': app.user.username}, 'FOSUserBundle') }} |
        <a href="{{ path('fos_user_security_logout') }}">
            {{ 'layout.logout'|trans({}, 'FOSUserBundle') }}
        </a>
    {% else %}
        <a href="{{ path('fos_user_security_login') }}">{{ 'layout.login'|trans({}, 'FOSUserBundle') }}</a>
    {% endif %}
</div>
{% if app.request.hasPreviousSession %}
```

ez Layout

A screenshot of a web browser showing the eZ Platform registration page. The URL in the address bar is `ezconfmcy2015.dev/register/`. The page features a large eZ logo in the top left corner. A search bar with the placeholder "Search text" is positioned in the top right. A navigation bar at the top includes links for "eZ Mountains", "Getting Started", "Shopping", "Blog", and "Contact Us". Below the navigation bar is a "Log in" form with fields for "Email", "Username", "Password", and "Repeat password", followed by a "Register" button. At the bottom of the page, a blue footer bar states "Powered by eZ Platform™ CMS Open Source Web Content Management." Below this, a "Get in touch" section provides information about eZ's global offices and encourages users to contact them for more information.

Powered by eZ Platform™ CMS Open Source Web Content Management.

Get in touch

eZ is a global company with offices around the world including Skien (Norway), New York (USA), Oslo (Norway), Cologne (Germany), Paris (France), Lyon (France), Tokyo (Japan), Katowice (Poland)

[Contact us](#) for more information!



Recap

- ✓ Our MongoDB database ready
- ✓ FOS configured
- ✓ You can log in with an **API User**
 - ✓ You will get a **API User** authenticated
- ✓ You can log in with a **FOS User**
 - ✓ You will get a **User Wrapped** authenticated



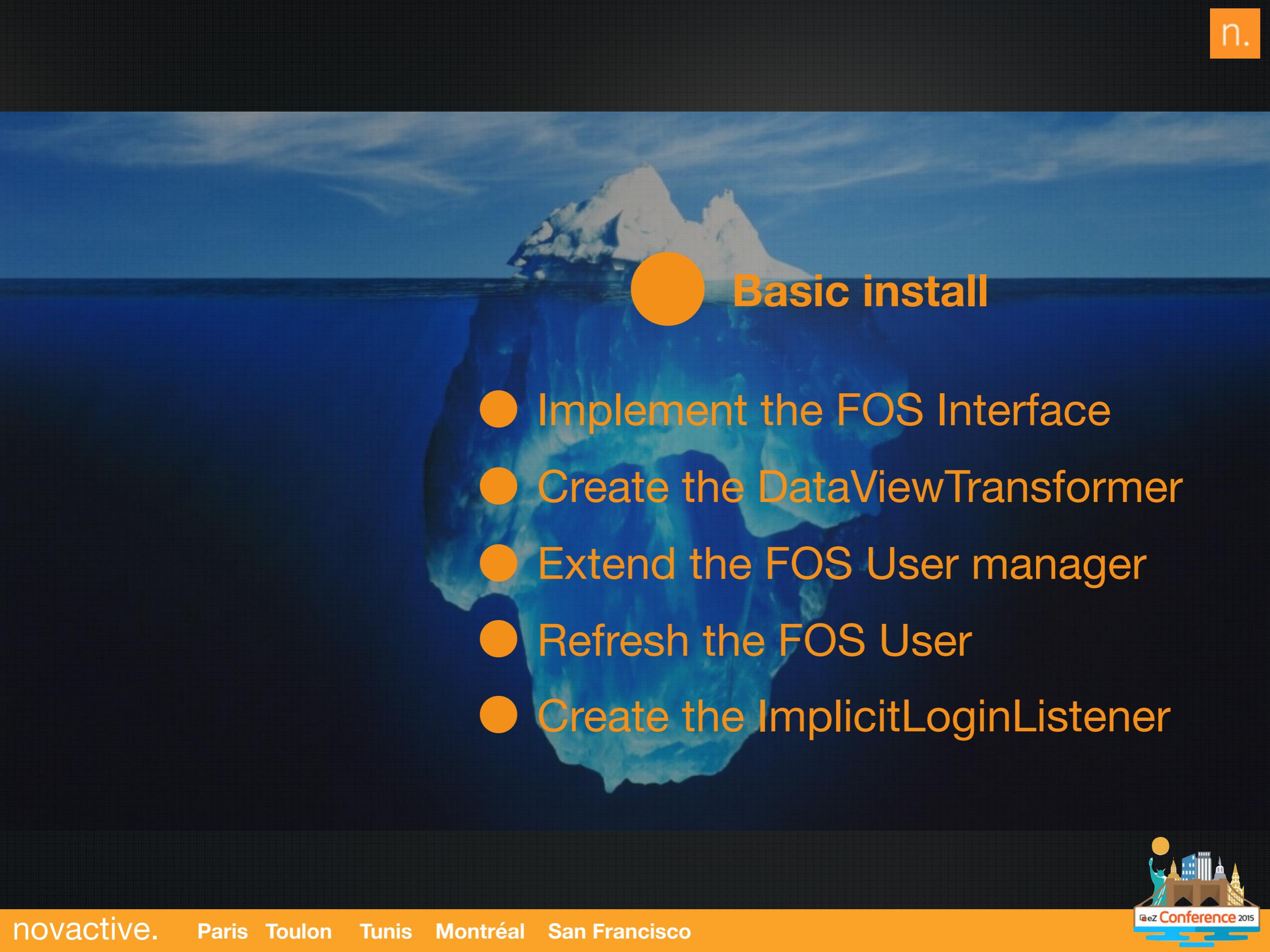
Are we done?

Why ?

- FOS User Bundle doesn't know what a **UserWrapped** is
- You need to be compliant to its Interface:
 - FOS\UserBundle\Model\UserInterface

// FOS Controller action - first lines

```
$user = $this->getUser();
if (!is_object($user) || !$user instanceof UserInterface) {
    throw new AccessDeniedException('This user does not have
access to this section.');
```

A large iceberg is visible in the background, partially submerged in dark blue water under a cloudy sky.

Basic install

- Implement the FOS Interface
- Create the DataViewTransformer
- Extend the FOS User manager
- Refresh the FOS User
- Create the ImplicitLoginListener

“It is possible to customize the `user class` used by `extending ezpublish.security.login_listener` service,[...] and `override getUser()` to return whatever user class you want.”

–The Amazing eZ Documentation



Wrapped User

- ✓ You need to be compliant to:

- ✓ *FOS\UserBundle\Model\UserInterface*

- ✓ Override

```
parameters:
```

```
    ezpublish.security.login_listener.class: TheEconomist
\Bundle\UserBundle\EventListener\SecurityListener
```

```
class SecurityListener extends BaseSecurityListener {
    protected function getUser(UserInterface $originalUser,
APIUser $apiUser) {
        return new UserWrapped($originalUser, $apiUser);
    }
}
```



“The form's view data is expected to be an instance of class Document\User, but is an instance of class Core\UserWrapped.”

–Our next error

Data Transformer

```
public function transform($value)
{
    if ($value instanceof UserWrapped) {
        return $value->getWrappedUser();
    }

    return $value;
}

public function reverseTransform($value)
{
    return $value;
}
```

- ✓ To use this transformer you need to add it to the form type
- ✓ Extend the Form Type



Create the profile Form Type

```
novactive.ezfosuser.novactive_ezfosuser_profile.form.type:  
    class: Novactive\Bundle\eZF0SUserBundle\Form\Type\Profile  
    tags:  
        - { name: form.type, alias:  
novactive_ezfosuser_profile}
```

```
use Symfony\Component\Form\AbstractType;  
class Profile extends AbstractType {  
  
    public function buildForm(FormBuilderInterface $builder, array $options) {  
        $builder->add('firstname')->add('lastname')->add('isCertified');  
        $builder->addViewTransformer( new UserWrappedToUser());  
    }  
  
    public function getParent() {  
        return 'fos_user_profile';  
    }  
  
    public function getName() {  
        return 'novactive_ezfosuser_profile';  
    }  
}
```

“The class '**Novactive\Bundle\\eZFOSUserBundle\Core\UserWrapped**' was not found in the **chain configured namespaces** **Novactive\Bundle\\eZFOSUserBundle\Document, FOS\\UserBundle\Model** ”

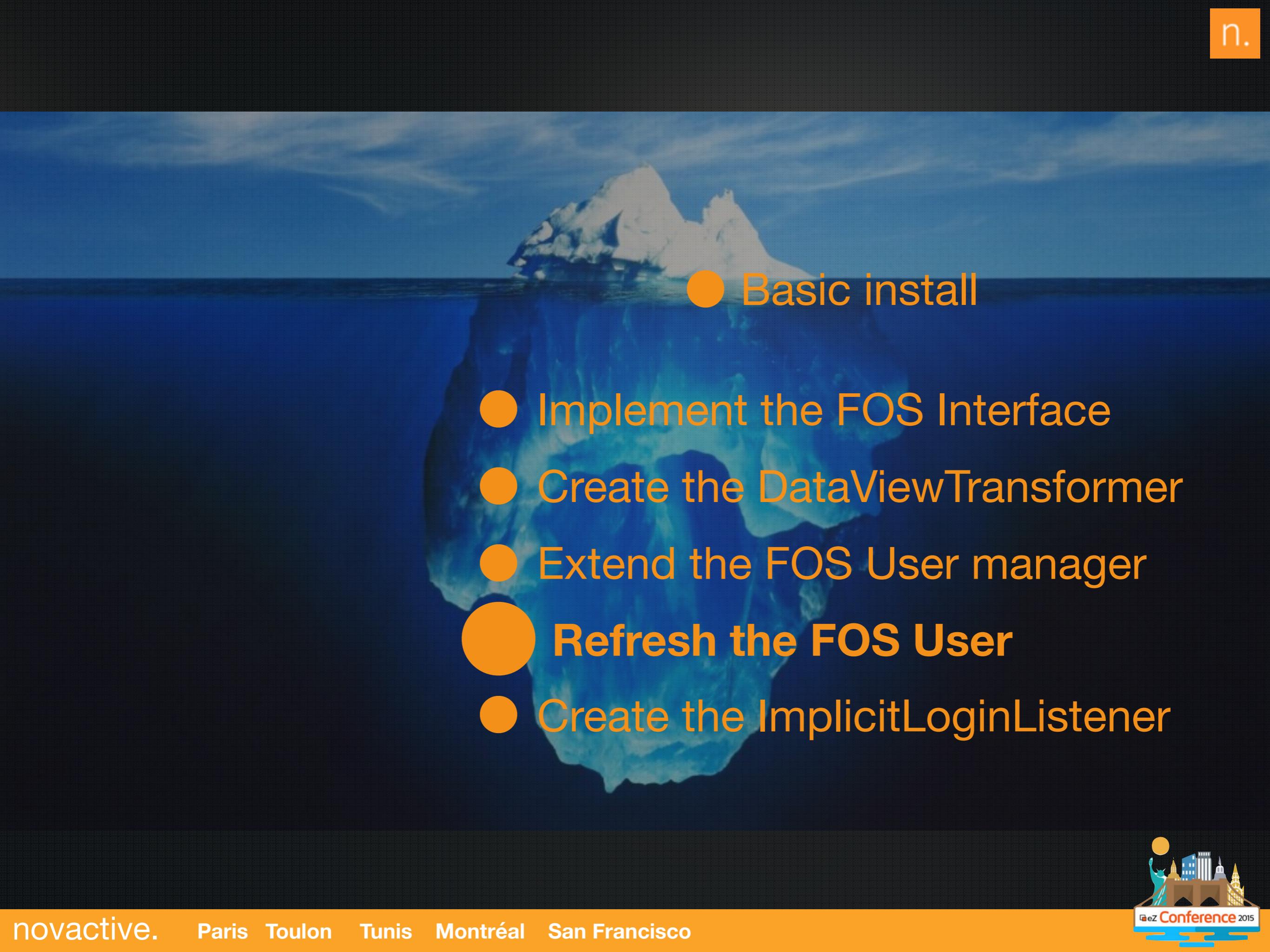
–in vendor/doctrine/common/lib/Doctrine/Common/Persistence/Mapping/MappingException.php at line 37



Extend the User manager

```
novactive.ezfosuser.fos_user_manager:  
    class: Novactive\Bundle\eZFOSUserBundle\Doctrine  
\UserManager  
        parent: fos_user.user_manager.default
```

```
public function __method__User(UserInterface $user, $andFlush = true)  
{  
    if ($user instanceof UserWrapped) {  
        parent::__method__User($user->getWrappedUser(), $andFlush);  
  
        return;  
    }  
    parent::__method__User($user, $andFlush);  
}
```

A large iceberg is visible in the background, partially submerged in dark blue water under a cloudy sky.

- Basic install

- Implement the FOS Interface
- Create the DataViewTransformer
- Extend the FOS User manager
- Refresh the FOS User
- Create the ImplicitLoginListener

Refresh user

```
novactive.ezfosuser.user_provider.username_email:  
    class: Novactive\Bundle\eZFOSUserBundle\Security  
\UserProvider  
    arguments: [ @fos_user.user_manager ]
```

```
providers:  
    chain_provider:  
        chain:  
            providers: [novaefos_provider, ezpublish]  
    novaefos_provider:  
        id: fos_user.user_provider.username  
        id: novactive.ezfosuser.user_provider.username_email  
    ezpublish:  
        id: ezpublish.security.user_provider
```



Refresh user

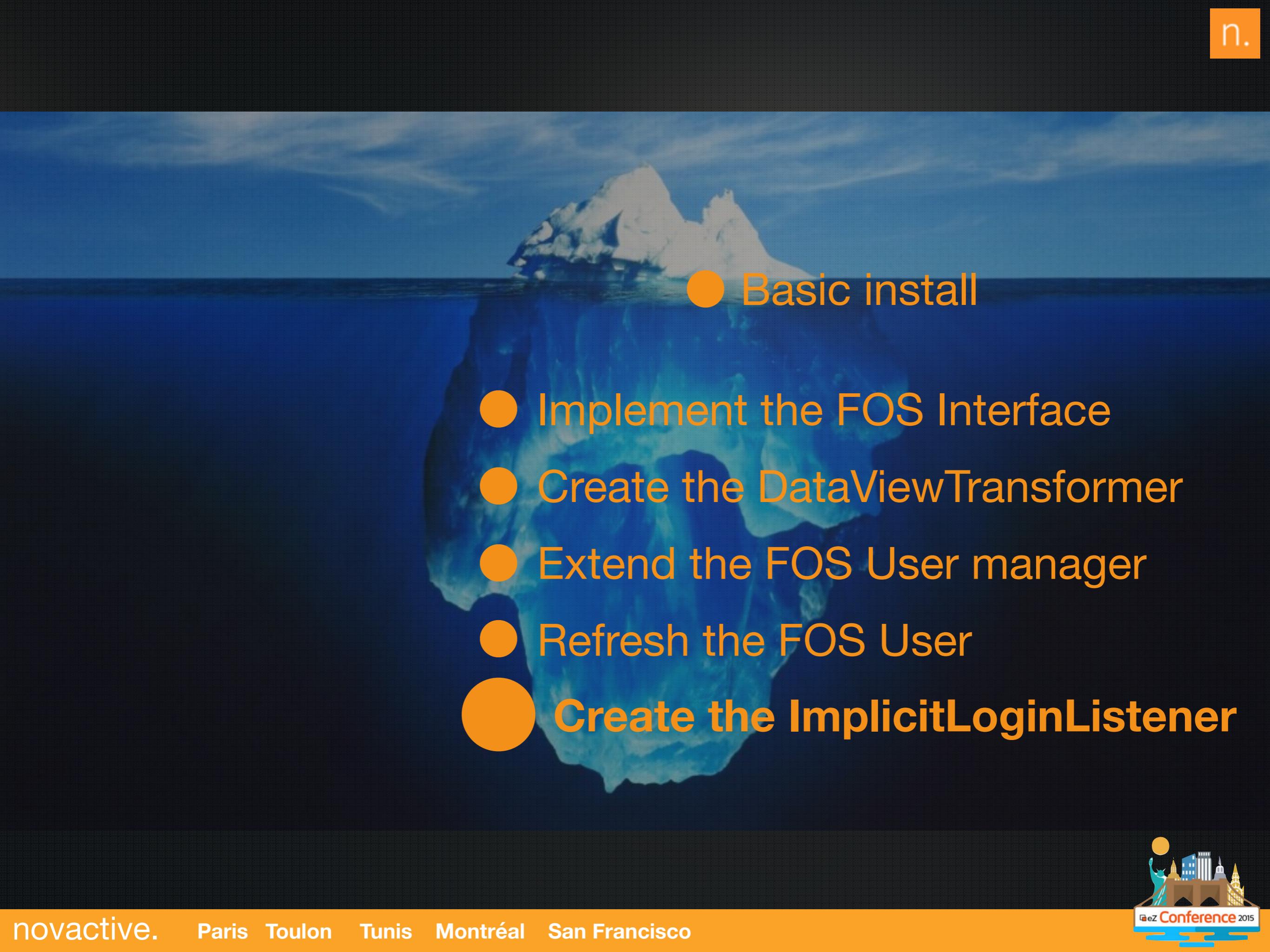
```
class UserProvider extends EmailUserProvider {

    public function refreshUser(SecurityUserInterface $user)
    {
        if (!$user instanceof UserWrapped) {
            throw new UnsupportedUserException(...);
        }
        if (null ===
            $reloadedUser = $this->userManager->findUserBy(array( 'id' => $user-
>getWrappedUser()->getId() ))
        ) {
            throw new UsernameNotFoundException(...);
        }
        $user->setWrappedUser($reloadedUser);
        throw new UnsupportedUserException("Forward to the
ezpublish provider");
    }
}
```



Final step



A large iceberg is visible in the background, partially submerged in dark blue water under a cloudy sky.

- Basic install

- Implement the FOS Interface
- Create the DataViewTransformer
- Extend the FOS User manager
- Refresh the FOS User
- Create the ImplicitLoginListener

Implicit Login

- Goal
 - **Rely on** the eZ Interactive Login
- Steps
 - 1 - Dispatch the event.
 - 2 - If no eZ user is returned, load Anonymous user.
 - 3 - Inject eZ user in repository.
 - 4 - Create the UserWrapped user object
 - 5 - Create new token with UserWrapped user
 - 6 - Inject the new token in security context



Implicit Login

```
novactive.ezfosuser.implicit_event_listener:  
    class: Novactive\Bundle\eZFOSUserBundle\Listener  
\ImplicitLoginListener  
arguments:  
    - @ezpublish.api.repository  
    - @ezpublish.config.resolver  
    - @event_dispatcher  
    - @security.token_storage  
tags:  
    - { name: kernel.event_subscriber }
```



```

$originalUser = $event->getUser();
if ($originalUser instanceof eZUser || !$originalUser instanceof UserInterface) {
    return;
}

$token = $this->tokenStorage->getToken();

$subLoginEvent = new InteractiveLoginEvent($event->getRequest(), $token);
$this->eventDispatcher->dispatch(MVCEvents::INTERACTIVE_LOGIN, $subLoginEvent);
if ($subLoginEvent->hasAPIUser()) {
    $apiUser = $subLoginEvent->getAPIUser();
}
else {
    $apiUser = $this->repository->getUserService()->loadUser(
        $this->configResolver->getParameter('anonymous_user_id')
    );
}
$this->repository->setCurrentUser($apiUser);
$providerKey      = method_exists($token, 'getProviderKey') ? $token-
>getProviderKey() : __CLASS__;
$interactiveToken = new InteractiveLoginToken(
    new UserWrapped($originalUser, $apiUser),
    get_class($token),
    $token->getCredentials(),
    $providerKey,
    $token->getRoles()
);
$interactiveToken->setAttributes($token->getAttributes());
$this->tokenStorage->setToken($interactiveToken);

```

You did it
Congratulations

Wanna Go further?

Install HWI OAuth for

Social Connectors !



Sign in with Facebook



Sign in with Google+



Sign in with Twitter



Sign in with LinkedIn



Sign in with Yahoo



Sign in with Foursquare



Sign in with Windows Live

Thank you!



<https://twitter.com/NovactiveSF>

<https://twitter.com/Plopix>



<https://www.facebook.com/NovactiveSF>

<http://www.novactive.us>

s.morel@novactive.us

<https://github.com/Novactive/NovaefZFOSUserBundle>

