Dependency Injection component. What where why?

Petr Sergeev, EvercodeLab

http://www.tv.net.ua/news/na_teleekrane/1050664487-zvezdnye-odessity-v-yetu-pyatnicu-proveryat-svoj-intellekt-v-programme-chto-gde-kogda-zvezdnye-vojny.html

About me

- Web-developer since 2005
- EvercodeLab co-founder and CTO
- Ruby Php Python, over 9000 frameworks
- ~~Genius Billionaire Playboy Philanthropist~~

- Basics
- Dependency injection
- In Symfony2
- In Drupal 8

Basics about injecting, good code and other things

Good code is good

- Clean
- Reusable
- Clutter-free
- Testable

Bad policeman

```
class Notifier
{
    private $mailer;

    public function __construct() {
        $this->mailer = new Mailer();
    }

    public function notify() {
        ...
        $this->mailer->send($from, $to, $msg);
        ...
    }
}
```

Good policeman

```
class Notifier
{
	private $mailer;

	public function __construct(Mailer $m) {
		$this->mailer = $m;
	}

	public function notify() {
		...
		$this->mailer->send($from, $to, $msg);
		...
	}
}

$mailer = new Mailer();
$notifier = new Notifier($mailer);
```

Superman

```
class Notifier
{
	private $mailer;

	public function __construct(MailerInterface $m) {
		$this->mailer = $m;
	}

	public function notify() {
		...
		$this->mailer->send($from, $to, $msg);
		...
	}
}

$mailer = new Mailer();
$notifier = new Notifier($mailer);
```

http://www.flickr.com/photos/gkyc/with/43902217/

Injection types

- Property injection
- Contructor injection
- Setter injection

Property injection

```
class Notifier
{
    public $mailer;

    ...
}

$mailer = new Mailer();
$notifier = new Notifier($mailer);
$notifier->mailer = $mailer;
```

Constructor injection

```
class Notifier
{
    private $mailer;

    public function __construct(Mailer $m) {
        $this->mailer = $m;
    }
}
```

Setter injection

```
class Notifier
{
    private $mailer;

    public function setMailer($mailer) {
        $this->mailer = $mailer;
    }
}

$mailer = new Mailer();
$notifier = new Notifier($mailer);
$notifier->setMailer($mailer);
```

Dependency Injection

=

Inversion of control

http://www.flickr.com/photos/begumf/with/7916700382/

«Do not instantiate the dependencies explicitly in your class. Instead, declaratively express dependencies in your class definition.»

MSDN

How it all goes

- Manual injection
- Factory
- From container/injector

Dependency Injection in action

- System based on abstractions
- Independent low and high level modules
- Abstraction doesn't know about implementation

What does component do

- Construct objects graph
- Instantiatiate services
- Application logic and infrastructure logic are kept separately
- Cache dependencies

Service

Bunch of code responsible some particular

global task

Separation of concerns

http://www.luminescentphoto.com/blog/2011/04/27/concert-going-with-the-p7000/

To be or not to be

- Database
- Cache
- Mailer
- Logger
- Model
- Controller(?)

How it works

- $container->getService('some_service')
- Read services graph
- Instantiate service
- ...
- Profit

Scopes

```
app.form.handler.registration:
    class: App\UserBundle\Form\Handler\FormHandler
    arguments: [@fos_user.registration.form, @request]
    scope: request

app.listener.location:
    class: App\DefaultBundle\Listener\LocationListener
    arguments: [@request, @session, @security.context, @router]
   scope: request
```

Tags

```
app.listener.accountNumber:
    class: App\UserBundle\Listener\AccountNumberListener
    calls:
        - [ setContainer, [ @service_container ] ]
    tags:
        - { name: doctrine.event_listener, event: postUpdate }
```

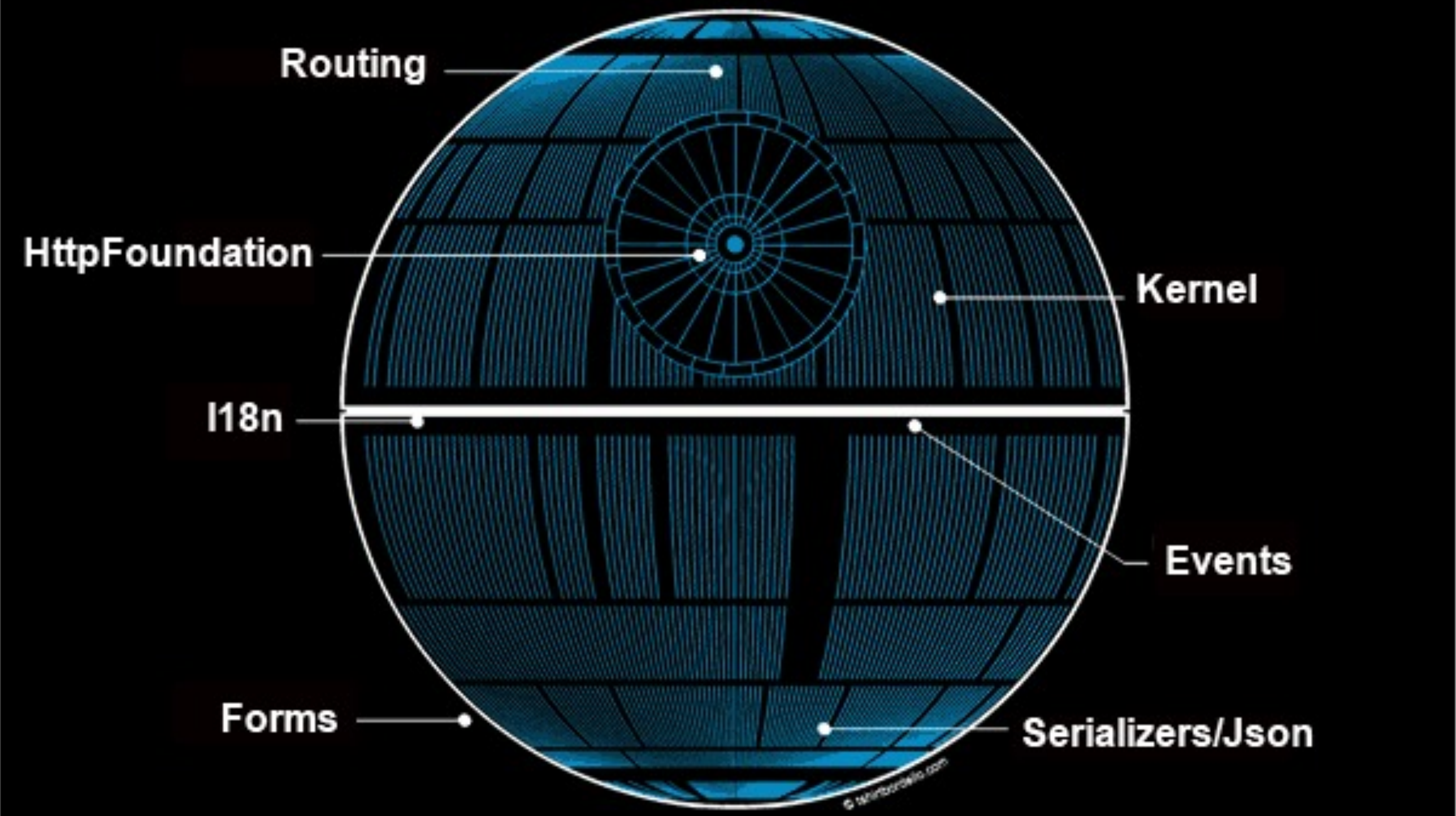
Pros and cons

http://www.flickr.com/photos/jayashree-shankar/with/5180430923/

- Been loaded service cannot be redefined
- Low coupling
- Type Hinting
- Localization for debug and testing
- Caching

- Blackbox
- Hard to understand, huh?

THE DEATH STAR



Symfony 2

HttpKernel

```
/**
 * Boots the current kernel.
 * @api
 */
public function boot()
{
    // init container
    $this->initializeContainer();

    foreach ($this->getBundles() as $bundle) {
        $bundle->setContainer($this->container);
        $bundle->boot();
    }

    $this->booted = true;
}
```

Symfony2

- Compiling
- Tagging, event subscribers and listeners
- Bundles

```php
$em = $this->container->get('doctrine.orm.entity_manager');
```

Drupal 8

Services available

- Database
- Module handler
- Request object

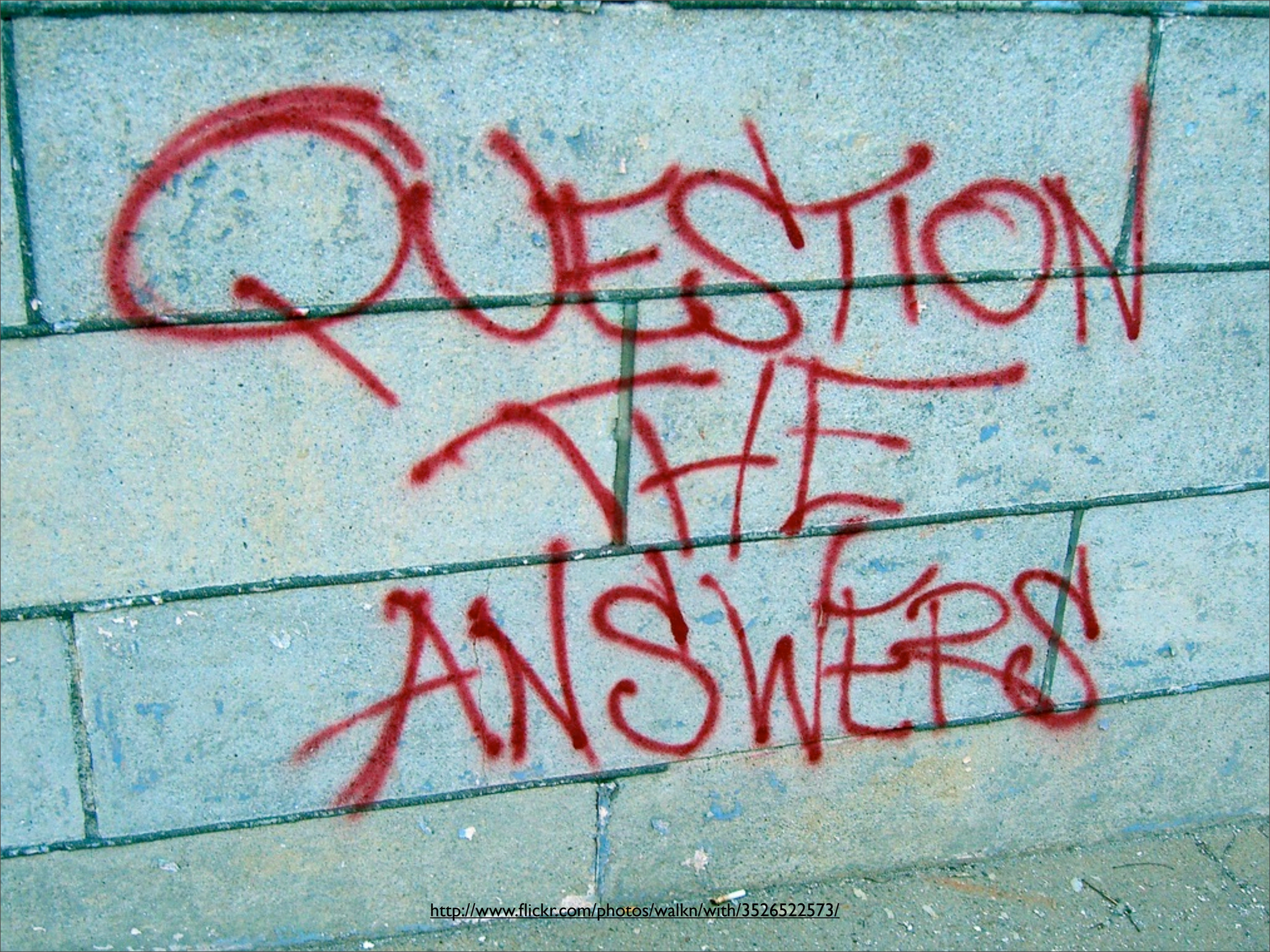
2 ways of usage

- Drupal::service(‘db’)
- OOP style

What's that all about

Sources list

- Wikipedia
- Drupal.org
- Symfony.com
- Katbailey.github.io



http://www.flickr.com/photos/walkn/with/3526522573/

Thanks

Petr Sergeev

EVERCODE

@i_feya

petr@evercodelab.com