

Yii, frameworks and where PHP is heading to

Alexander Makarov
Yii core team, Stay.com

Is PHP still there?

- Developers are migrating to
 - Ruby
 - Python
 - Java
 - C++
- Developers are migrating...
- Because... **PHP sucks?!**

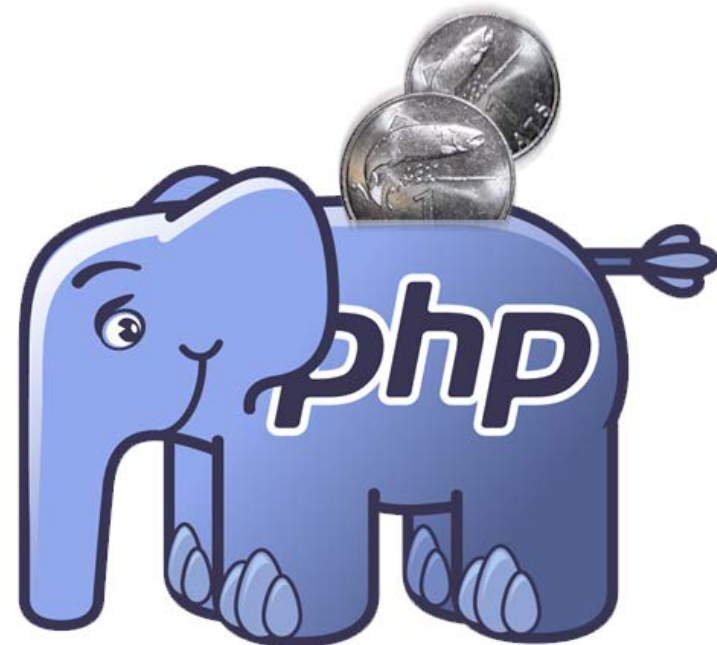


Well, it sucks a bit. Not **that** much!

- Very good choice for web development.
- Will be there for a long time.
- Programming language is just a tool... one of the tools.
- With a good framework bad parts aren't that visible.

PHP is good for business

- It's **web-focused**.
- **Lots** of ready to use stuff.
- PHP makes **everything** easy.
- Easy to find and replace developers.
- Less risky.
- Project costs less.



Good choice for business → Good
choice for developer.

PHP itself is heading towards **simplicity and usability**

- Password hashing API.
- Generators.
- More consistency.
- Built-in webserver.
- More performance.
- Better syntax.
- etc.

And that's the way to go!

What about **frameworks**?

Good things first

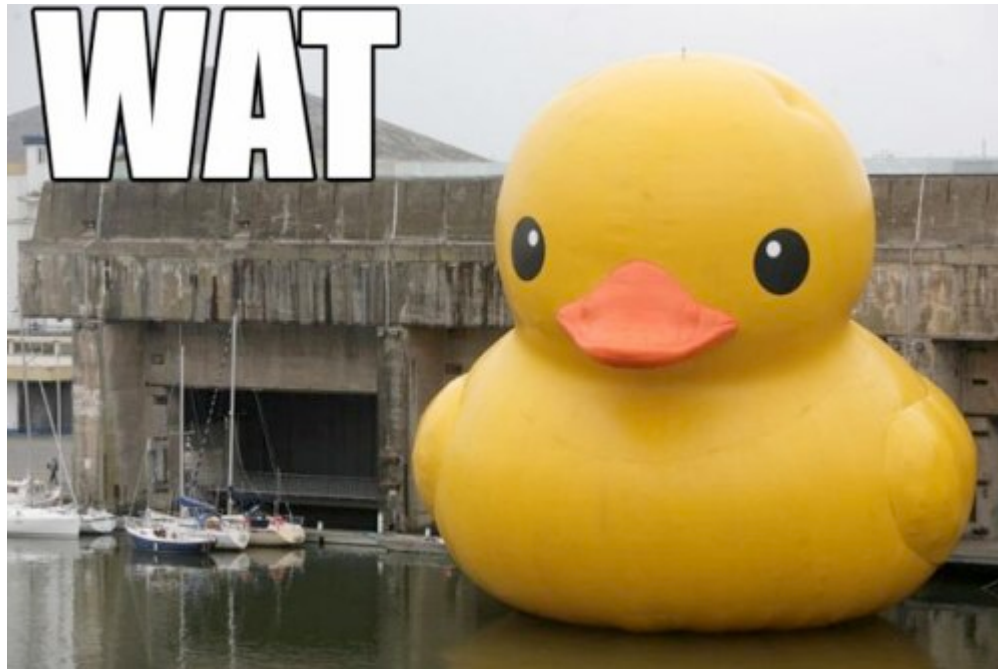
- github = standard for hosting cool PHP stuff.
- PSR-0 = interoperable & faster class loading.
- At least some **stable frameworks** are there and definitely ready for business: **BC and maintained.**



Now not that good things...

“Enterprise”

- ZF 2 and Symfony2 are very “**enterprise**”.
- Much more “**enterprise**” than ZF1 and symfony were.



This “enterprise” shit is too complex

“ZF2 is too "scientific". You can write a thesis about ZF2 and its design. Don't get me wrong, I love best practices, standards, design patterns etc., but ZF2 devs dive too deep into "science".
Symfony2 is better but not enough.”

But **why** becoming **complex** instead of
being simple?

Why ZF2 is what it is

- Zend is the **PHP company**, they'll have enough customers no matter what they'll release.
- **Main customers** are of **enterprise** type.
- Not really interested in 80% projects out there.
- **Consulting isn't profitable if product isn't complex enough ;)**
- Heading into J2EE direction they have a chance to get former EE-companies on board.

It works for Zend very well.

What about Symfony2?

- Personally I think they're **trying to be “enterprise”** (=complex) as well.
- Sensio labs is a **commercial company** earning for framework support and consulting.
- <http://fabien.potencier.org/article/65/why-symfony>
- Good thing is that framework is **a bit more practical than ZF2**.
- Still **very complex**.

What's **bad** in these “enterprise” things?

- Design-patterns oriented instead of practically oriented. Emphasizing on patterns.
- Easier to unit-test, **harder to develop and learn.**
- Almost impossible to delegate routine work to less competent developers w/o spending lots of time teaching them first.
- **High risk** for project owner.

That reminds me... Java past

- “Their **main thesis to support that complexity** is... hold your breath... fasten your seatbelts: **if it were easier, more stupid people would be using it!. Ta-da!!**”
- “J2EE is no way simple. However **the reality is simple:** for J2EE to survive - we have to **make it simple to build, deploy and manage**”

Java is simpler now compared to what it was

- Play
- Apache Wicket
- Even Spring is much more usable and simple than what it was before

So are SF2 and ZF2 evil?

Not really. **Just a different niche.**

Enterprise isn't **only** complexity

- Stable API.
- Well-tested code.
- Backwards-compatible regular releases.
- Guarantee that vendor will support product for at least X years.
- Well-documented.
- Commercial support, trainings.
- And more...

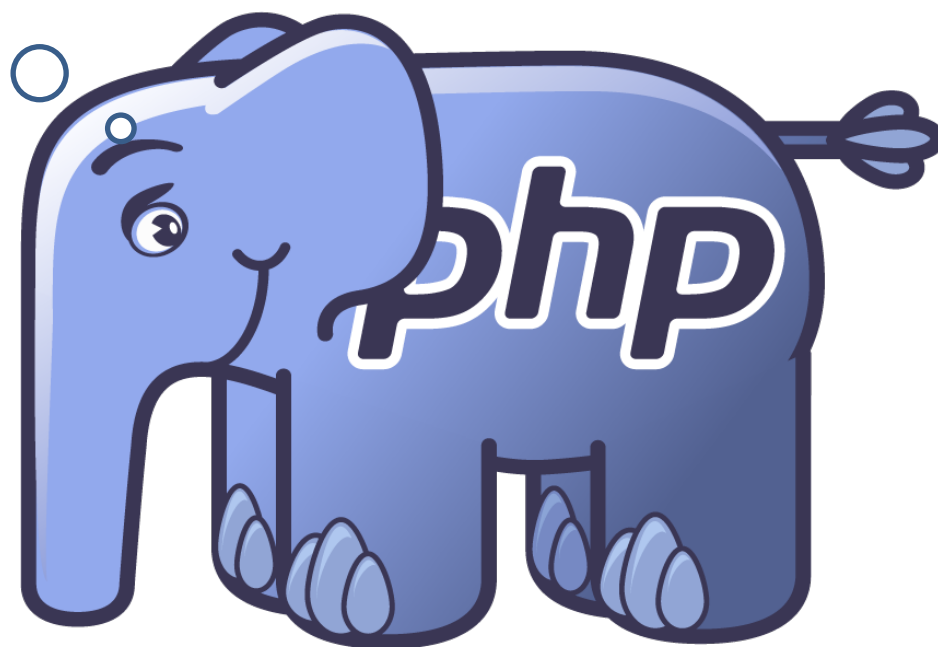
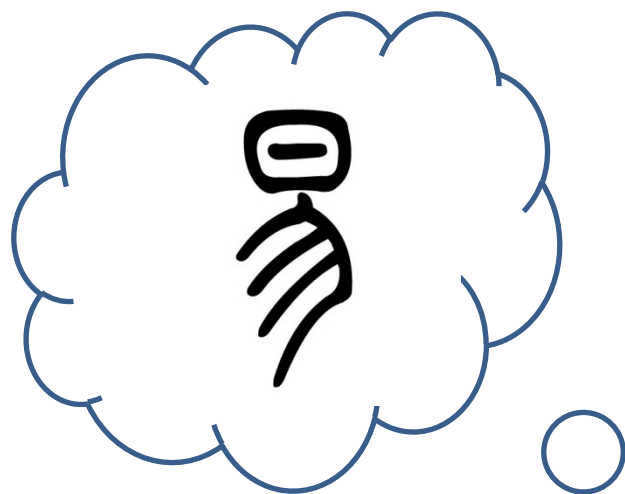
So what's left for practical fellows?

Practical frameworks

- **Development should be straightforward.**
We're not doing complex stuff for webapps most of the time.
- **Easy learning.**
- **Less magic.**
- **Less configuration.**
- **As simple API as possible.**

If **good** parts of enterprise are still
there it's perfect

Yii — a **practical** framework



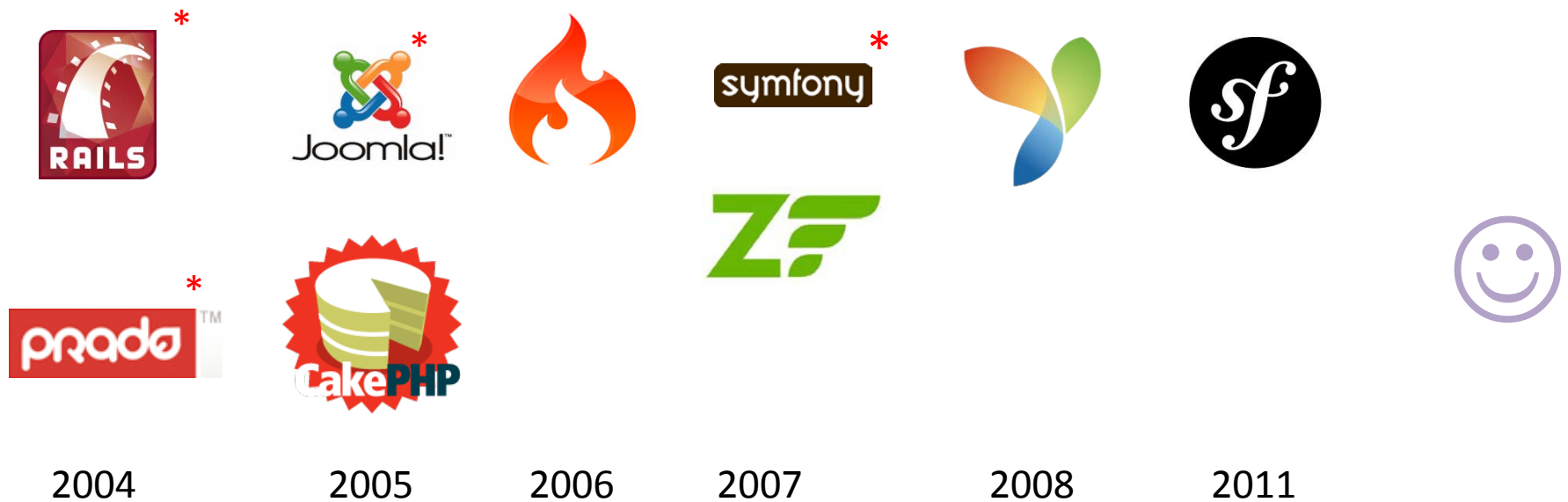
Let's bust some myths first

Framework **myths**

- If framework **reinvented the wheel**, it's a **bad** one.
- If there is **DI container** and **design patterns** all over the place, it's a **good** one.
- If framework X **implemented feature Y first**, it's **better**.
- **Loosely coupled** framework is always a **good** choice.
- If there's **no feature X** it's a **bad** one.

What's really important

- Easy to **learn**.
- Easy to **debug and fix**.
- Doesn't mess with your code or any good **third-party code**.
- Friendly active **community**.
- **BC** and **stable**.
- No roadblocks when trying to **extend** or **customize** it.
- Feels good ;)



Prado, c 2004 → Yii 1.0, 2008 → Yii 1.1, 2010

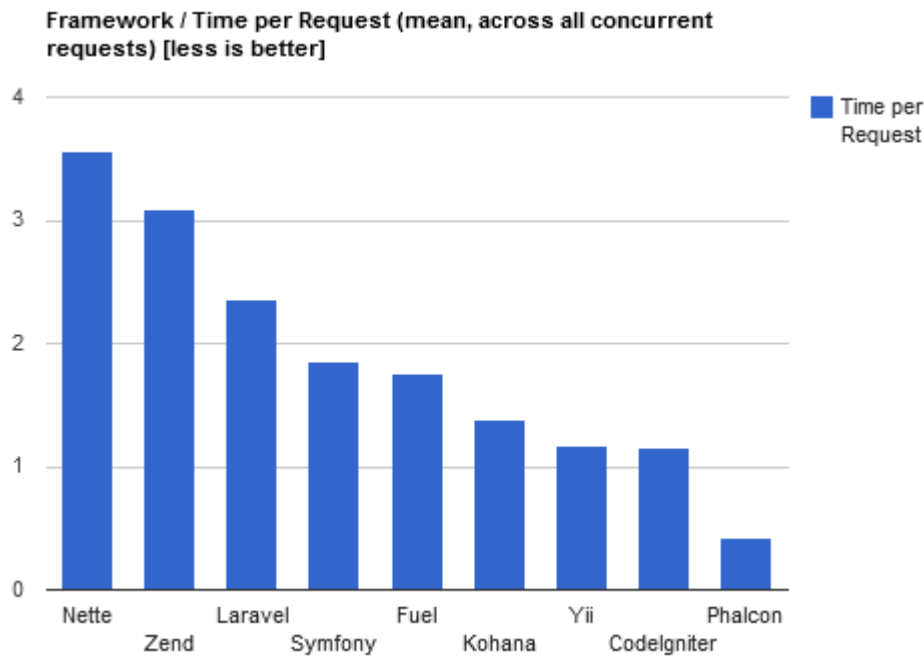
What is Yii?

- ✓ **PHP5 MVC.**
- ✓ **Nice API.**
- ✓ **DAO, AR, migrations.**
- ✓ **Form builder.**
- ✓ **Themes.**
- ✓ **Test framework.**
- ✓ **Well documented.**
- ✓ **Friendly community.**
- ✓ **Powerful cache.**
- ✓ **RBAC, auth framework.**
- ✓ **Console.**
- ✓ **error handler, log.**
- ✓ **Gii.**
- ✓ **CLDR based I18n.**
- ✓ **Widgets.**
- ✓ **BSD license.**

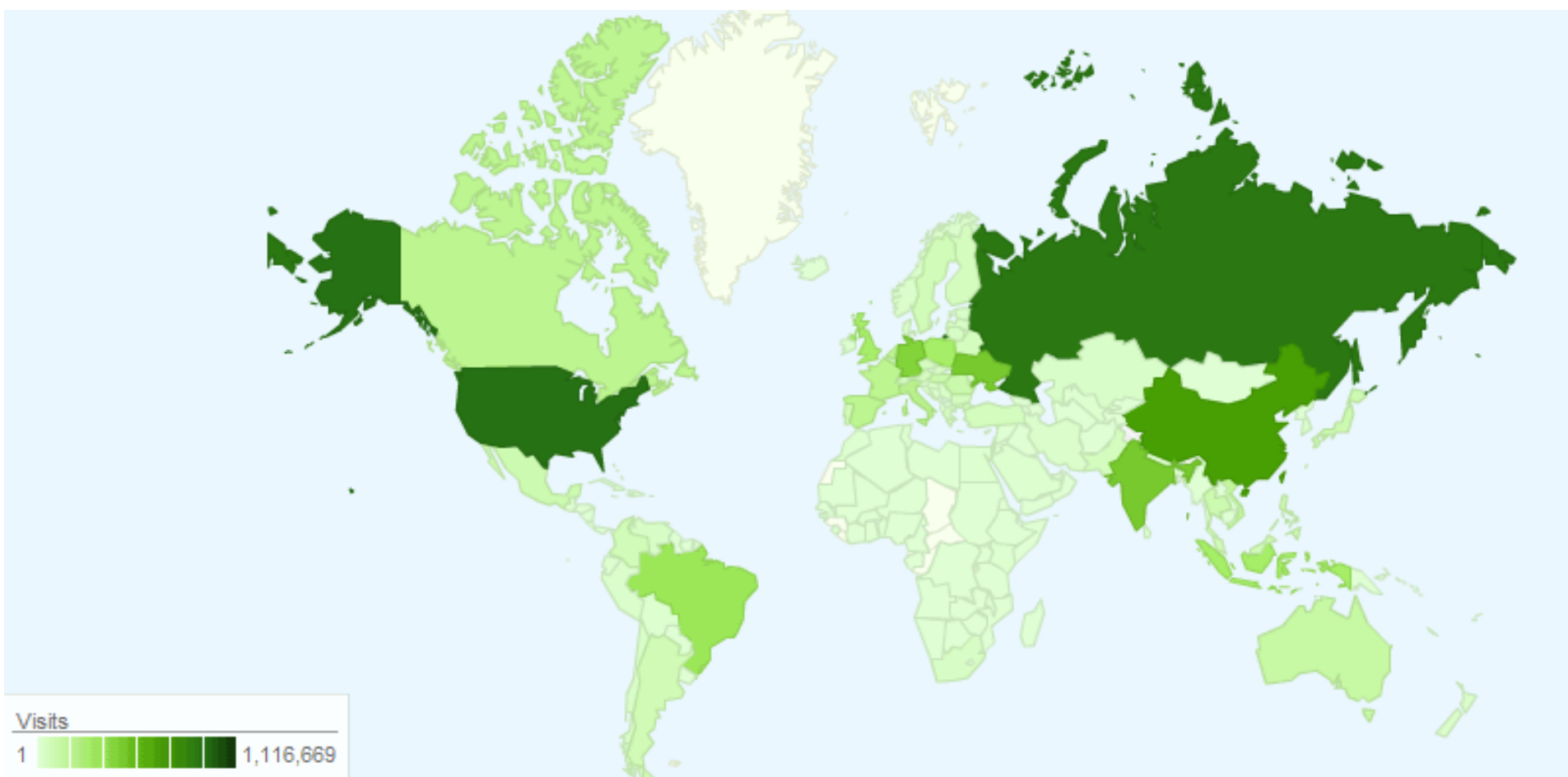
Still stays **the fastest modern PHP framework** (excluding some microframeworks and PECL ones).

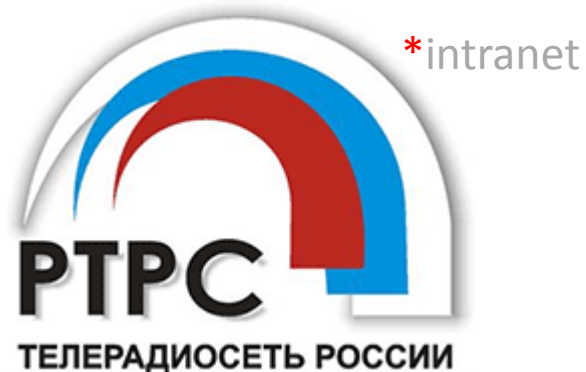
- That's **not** ours benchmark.

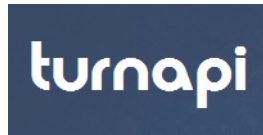
<http://docs.phalconphp.com/en/latest/reference/benchmark/hello-world.html>




Getting even faster and less memory-hungry thanks to our GitHub community.





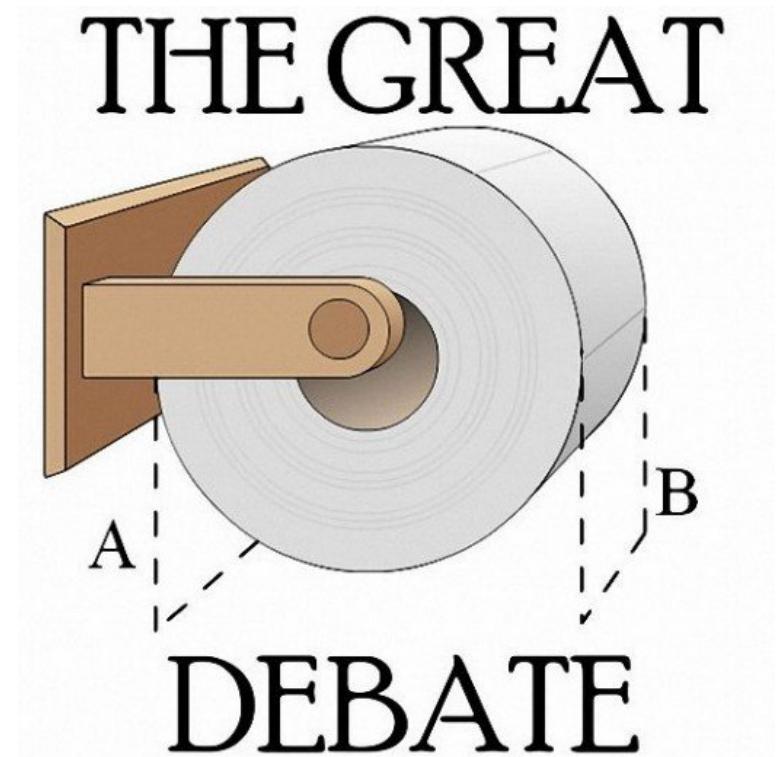




- 
1. Balanced
 2. Stable
 3. Flexible
 4. Well-documented
 5. Doesn't mess with
your stuff

Balance isn't easy to achieve

- Theoretically correct architecture or practical experience?
- 20% or 80%
- Features or core?

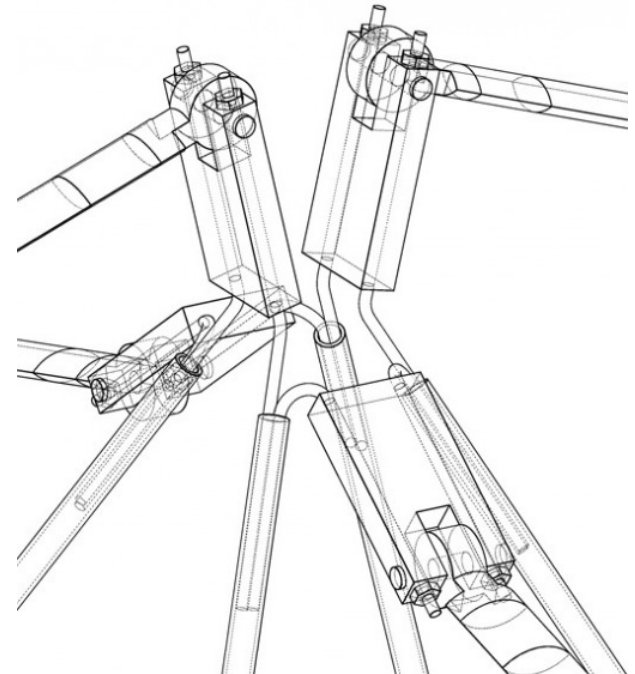


Stable API and BC = good & evil



Flexibility should not mess with simplicity

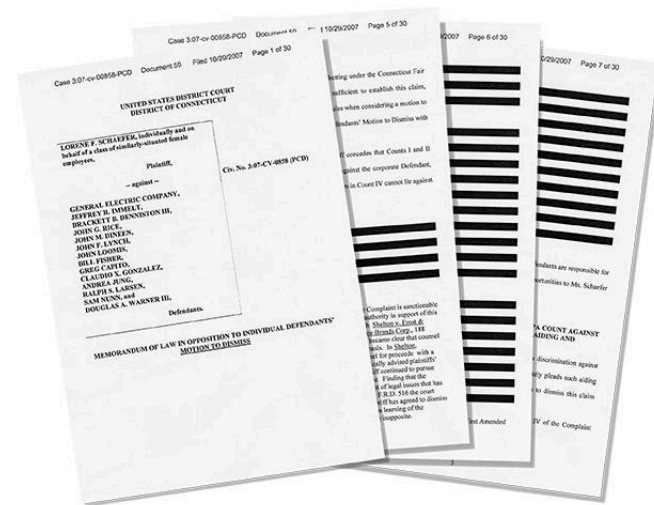
- As **less abstraction** as possible.
- It's simple to make things complex. Making things simple isn't.
- Conventions.
- Hide complexity.



There's not a single
setMyCoolFactoryDependencyInjection
Container()
in Yii

Docs are **very** important

- We're documenting as soon as writing code.
- Definitive guide and blog tutorial translated into 16 languages.
- phpdoc.
- Examples.
- Great books.



Yii can work with components from
Symfony 2 and ZF2.
That's what these are for, right? ;)

Yii is **practice oriented** framework

The end?

- Questions?
- Suggestions?
- sam@rmcreative.ru
- @sam_dark
- <http://yiiframework.com/>

I knew you would ask about it ;)

- PHP **5.3.8+**
- **Namespaced** classes
- **PSR-0**

- Better **structured**
- Less entities
- All the **good things are still there**

v2

Yii2: base

- New aliases
@yii/base/Component
- CComponent →
Object + Component
- SPL instead of custom
collections
- Killed CFormModel.
Model should be
used instead.
- Better rules and
scenarios.

```
class MyComponent extends
\yii\base\Object
{
    public $x;
    public function __construct($a, $b)
    {
        //...
    }
}

$component = MyComponent::newInstance(
    array('x'=>10),
    'a', 'b'
);
```

Yii2: View Object

- render(), widget(), beginCache() → viewObject
- View: \$owner = class that called render
- \$this = View.
- No need for renderers.
- Can be used in console.
- CHtml is still there.

Yii2: events

```
$post->on('add',  
function($event) { ...  
});
```

```
$post->trigger('add',  
new Event($this));
```

```
$post->off('add',  
$callback);
```

```
$handlers = $post->  
getEventHandlers('add'  
);
```

- No need to declare
- jQuery-like syntax
- Behaviors instead of filters

Yii2: Query object

```
// Query object
$query = new Query;
$query->select('id')->from('tbl_customer')->limit(10);
$command = $db->createCommand($query);
$this->assertEquals("SELECT `id` FROM `tbl_customer`
LIMIT 10", $command->sql);

// array
$command = $db->createCommand(array(
    'select' => 'name',
    'from' => 'tbl_customer',
));
```


Yii2: Even better ActiveRecord

```
$customer = Customer::find(2)
    ->active()
    ->one();
$customer->name = 'Qiang';
$customer->save();

$customers = Customer::find()
    ->order('id')
    ->asArray(true)
    ->all();
```

- Finder / Model
- Can create your own finder
- ~~::model()~~
- Autoquotes.
- Method chains.
- Dirty attributes.
- Less memory.
- Can get arrays from AR.

Yii2: AR

```
$postFinder = Post::find()
    ->where(array(
        'active' => true
    ));
if($isPrivate) {
    $postFinder->addWhere(array(
        'createdBy' => $userId,
    ));
}
$posts = $postFinder
    ->mergeWith($anotherFinder)
    ->all();
```

~~Criteria~~

- You can merge finders
- You can add conditions on the fly

Yii2: AR

- tableName(), relations(), scopes() = static.
- HAS_ONE, HAS_MANY relations.
- link = FKs
- via = through
- Scopes via anonymous functions.
- "@" and "?" tokens. Automatic alias. Own table. Foreign table.

```
class Customer extends ActiveRecord {
    const STATUS_ACTIVE = 1;

    public static function tableName() {
        return 'tbl_customer';
    }

    public static function relations() {
        return array(
            'orders:Order[]' => array(
                'link' => array('customer_id'
=> 'id'),
            ),
        );
    }

    public static function scopes() {
        return array(
            'active' => function($q) {
                return $q-
>andWhere('@.`status` = ' . self::STATUS_ACTIVE);
            },
        );
    }
}
```

Yii2: AR

```
$customers =  
Customer::find()->  
asArray()->all();  
  
foreach (Customer::find()  
as $customer)  
  
$count = Customer::count()  
->value();
```

```
$customers =  
Customer::find()->active()  
->all();  
  
$customers =  
Customer::find()  
->where('name like :name',  
array(  
    ':name' => '%customer%'  
))->order('id')->all();
```

TODO (if there will be enough time)

- HTTP (CURL) wrapper
- Package manager
- Mailer
- Twitter Bootstrap
- Debug toolbar
- Console requirements
- More helpers
- jQueryUI-based widgets (not just jQueryUI widgets)
- Commercial support (most probably not by Yii core team members)

1 or 2?

Work with the stable one. **1.1 will be supported till December 31, 2015.**

When?

Before we'll push alpha code to github repo we need to finish at least →

- i18n
- Controller + webapp
- A solid base for widgets
- URL manager

The end (really)

- Questions?
- Suggestions?
- sam@rmcreative.ru
- @sam_dark
- <http://yiiframework.com/>

