

# PHPCR

JCR meets scripting languages

# Henri Bergius



Director of R&D  
Nemein Berlin

@bergie

bergie.iki.fi

*"The Midgard Project is born."*

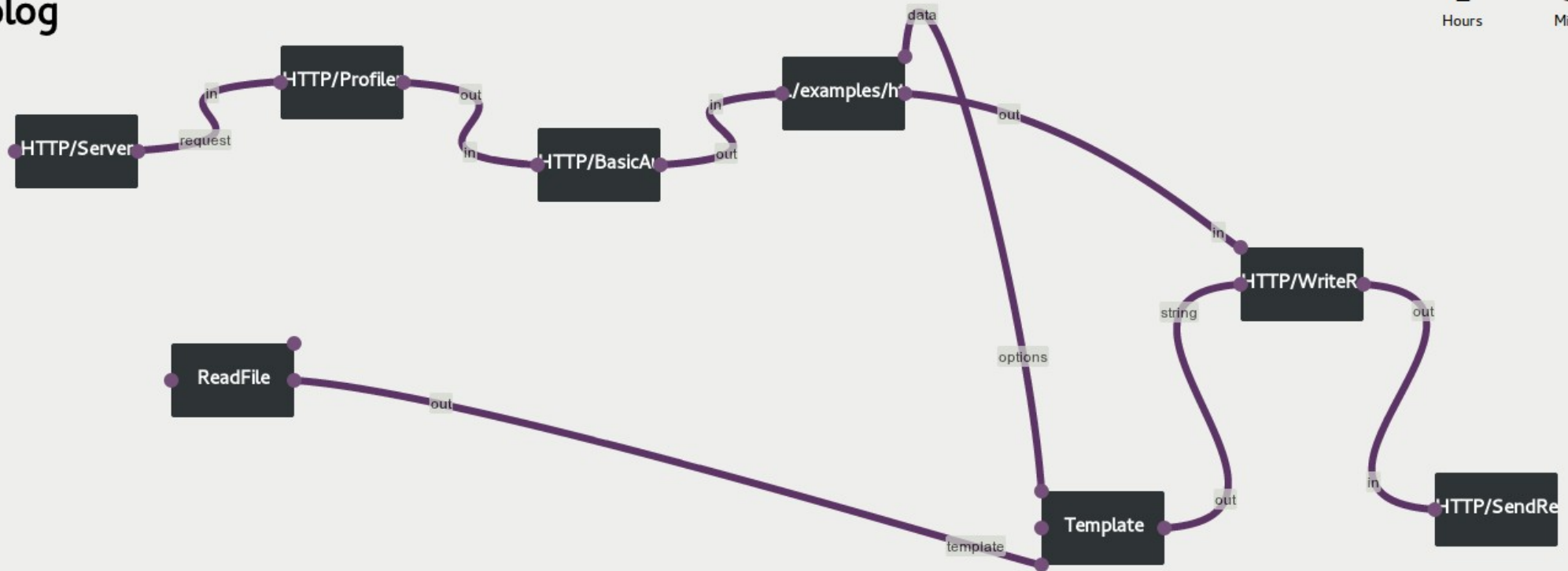
Launching our first Open Source CMS effort  
bergie.iki.fi, April 25th 1998



# NoFlo

blog

2 Hours 37 Minutes



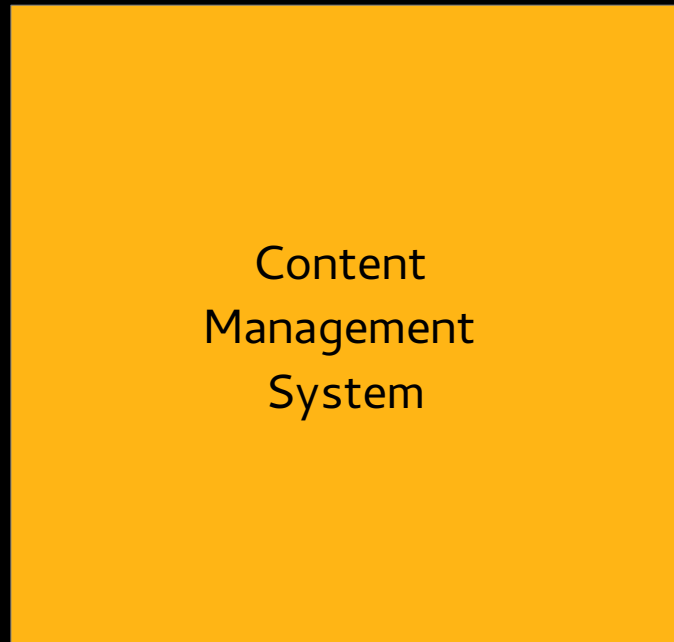


PHPCR

*The PHP Content Repository is an adaption of the Java Content Repository (JCR) standard, an open API specification defined in JSR-283. The API defines how to handle hierarchical semi-structured data in a consistent way.*

# Decoupling Content Management

Monolithic approach



Decoupled approach





# Why PHP?

# fuzzy notepad

this is really hard to write on

Blog | Archives

RSS

APR 9TH, 2012

## PHP: a fractal of bad design

---

### Preface

I'm cranky. I complain about a lot of things. There's a lot in the world of technology I don't like, and that's really to be expected—programming is a hilariously young discipline, and none of us have the slightest clue what we're doing. Combine with [Sturgeon's Law](#), and I have a lifetime's worth of stuff to gripe about.

### Recent Posts

- [Heteroglot: #15 in COBOL](#)
- [Stripe CTF 2.0](#)
- [Quick doesn't have to mean dirty](#)
- [Flora](#)
- [Python FAQ: Passing](#)
- [Python FAQ: Descriptors](#)
- [Python FAQ: Webdev](#)

# PHP runs practically anywhere

...from your LAMP server to cloud providers

Lots and lots of PHP developers

...especially in digital agencies

Modern PHP isn't actually  
that bad

# Standards

PSR-0: Autoloading interoperability

PSR-1: Coding standards

PSR-2: Style guide

[github.com/php-fig/fig-standards](https://github.com/php-fig/fig-standards)


# Namespaces








```
<?php
namespace Sandbox\MainBundle\Document;

use Doctrine\ODM\PHPCR\Mapping\Annotations as PHPCRODM;
use Symfony\Cmf\Bundle\MultilangContentBundle\Document\MultilangStaticContent;
use Liip\VieBundle\FromJsonLdInterface;


/**
 * @PHPCRODM\Document(referenceable=true,translator="child")
 */
class EditableStaticContent extends MultilangStaticContent implements FromJsonLdInterface
{
    /**
     * @PHPCRODM\String(multivalue=true)
     */
    public $tags;
```

# Shared libraries

Firefox ▾ Packagist 

←      psr-1   

Create a new account | Login



# Packagist

The PHP package archivist.

[Submit Package](#)

Packagist is the main **Composer** repository. It aggregates all sorts of PHP packages that are installable with Composer.  
[Browse packages](#) or [submit your own](#).

## Getting Started

### Define Your Dependencies

Put a file named `composer.json` at the root of your project, containing your project dependencies:

```
{
  "require": {
    "vendor/package": "1.3.2",
    "vendor/package2": "1.*",
    "vendor/package3": ">=2.0.3"
  }
}
```

## Publishing Packages

### Define Your Package

Put a file named `composer.json` at the root of your package, containing this information:

```
{
  "name": "your-vendor-name/package-name",
  "require": {
    "php": ">=5.3.0",
    "another-vendor/package": "1.*"
  }
}
```



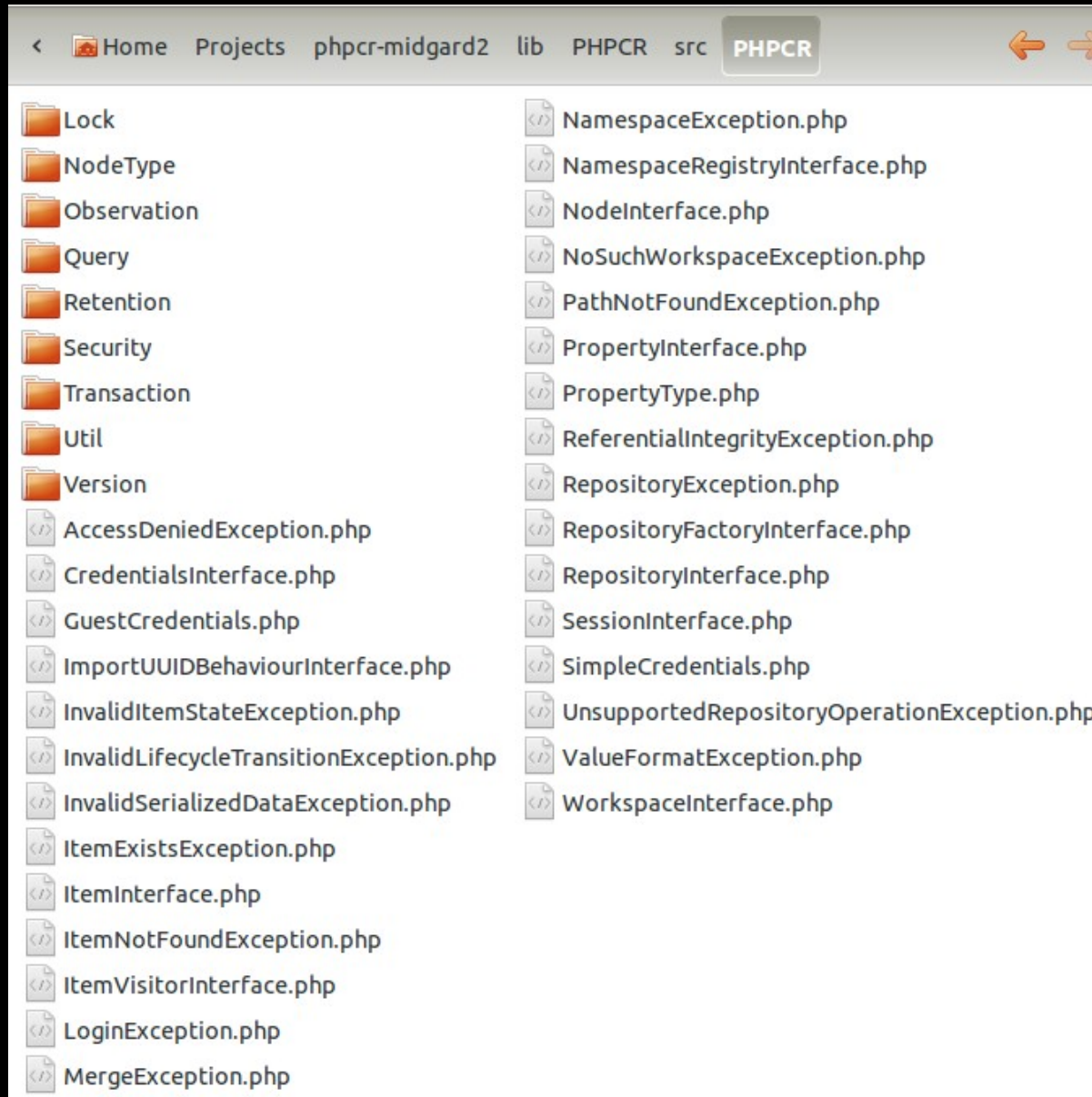
*"Bad programmers worry about the code. Good programmers worry about data structures and their relationships."*

- Linus Torvalds



PHPCR

# PHPCR is a collection of interfaces





# Same API, multiple repositories

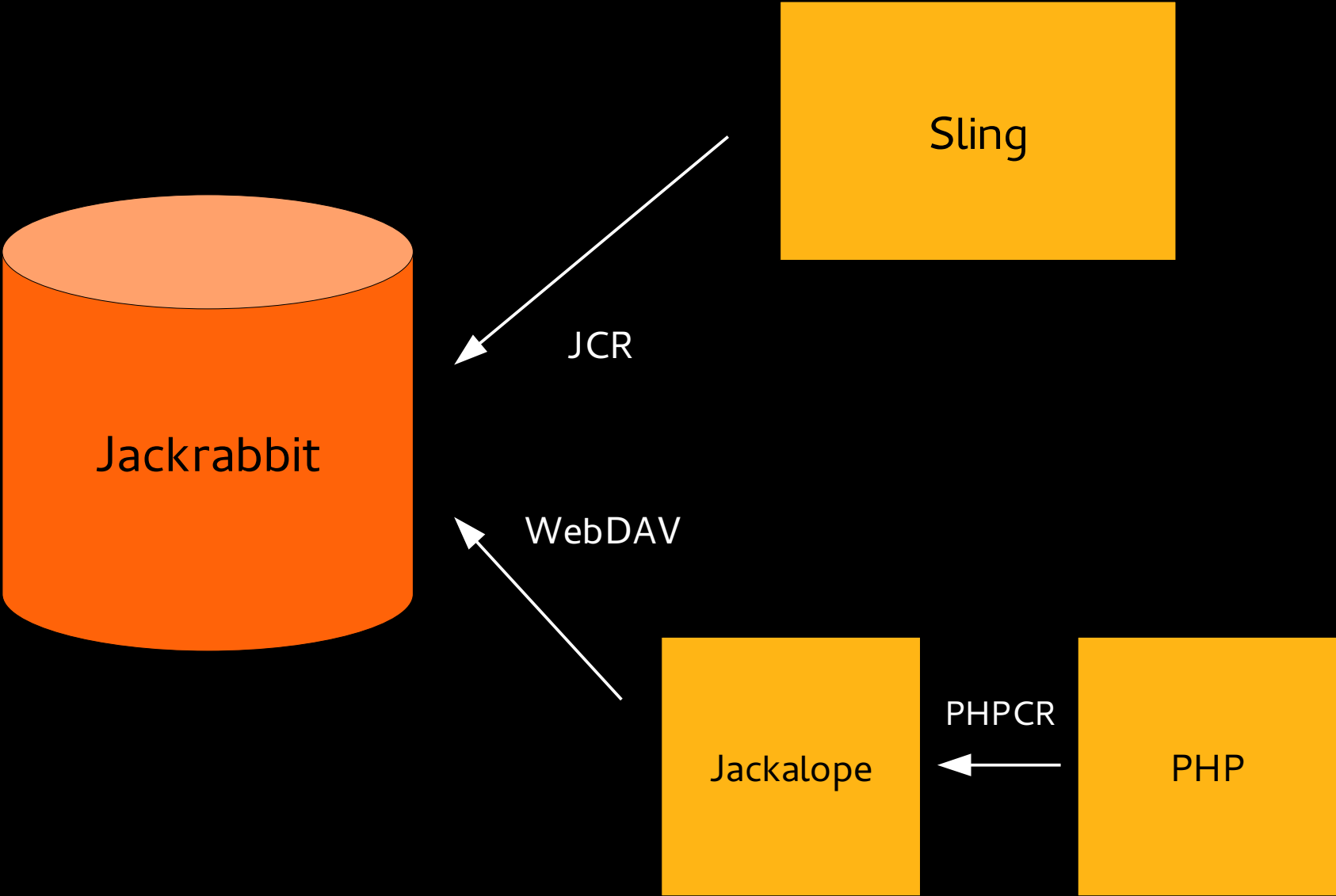
- Jackalope
  - Apache Jackrabbit: accessed through WebDAV
  - Doctrine DBAL: pure-PHP repository library
- [jackalope.github.com](http://jackalope.github.com)
- Midgard2
  - GObject-oriented repository library accessible through PHP extension
  - Python, JavaScript, Java, ...
- [midgard-project.org/phpcr](http://midgard-project.org/phpcr)

*A digital agency is building a  
campaign site for you.*

*They know PHP, but need some of  
your JCR-managed content*

# Scenario 1

Connect PHP to your existing Jackrabbit repository via Jackalope

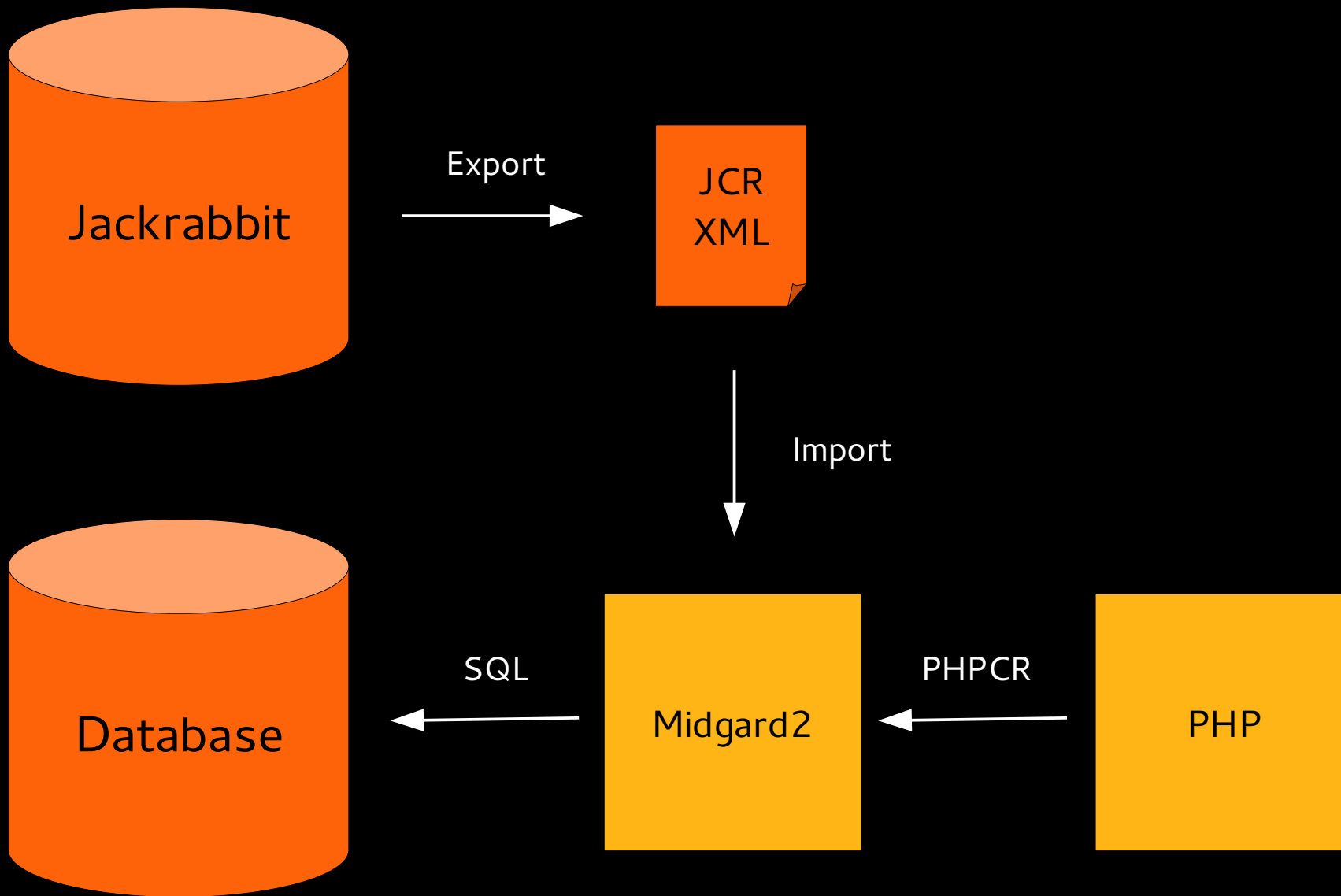




# Scenario 2

Run Jackalope DBAL or Midgard2 on LAMP

Export-import content from Jackrabbit



# Connecting to a repository: Jackrabbit

- Start Jackrabbit:  
\$ java -jar jackrabbit-standalone-2.4.1.jar
- Connect to repository:

```
4 $params = array(  
5     'jackalope.jackrabbit_uri' => 'http://localhost:8080/server',  
6 );  
7  
8 // Get a Jackalope repository  
9 $repository = Jackalope\RepositoryFactoryJackrabbit::getRepository($params);  
0  
1 // Log in  
2 $credentials = new \PHPCR\SimpleCredentials('admin', 'admin');  
3 $session = $repository->login($credentials, 'default');
```

# This is the only repository-specific part

```
// Set up Midgard2 repository configs
$parameters = array(
    // Use local SQLite file for storage
    'midgard2.configuration.db.type' => 'SQLite',
    'midgard2.configuration.db.name' => 'midgard2cr',
    'midgard2.configuration.db.dir' => __DIR__,
    'midgard2.configuration.blobdir' => __DIR__ . '/blobs',
    // Let Midgard2 initialize the DB as needed
    'midgard2.configuration.db.init' => true,
    // Enable this if you want to see the actual database queries
    // 'midgard2.configuration.loglevel' => 'debug',
);

// Get a Midgard repository
$repository = Midgard\PHPCR\RepositoryFactory::getRepository($parameters);

// Log in
$credentials = new \PHPCR\SimpleCredentials('admin', 'admin');
$session = $repository->login($credentials, 'default');
```

```
4 $params = array(
5     'jackalope.jackrabbit_uri' => 'http://localhost:8080/server',
6 );
7
8 // Get a Jackalope repository
9 $repository = Jackalope\RepositoryFactoryJackrabbit::getRepository($params);
10
11 // Log in
12 $credentials = new \PHPCR\SimpleCredentials('admin', 'admin');
13 $session = $repository->login($credentials, 'default');
```

# Working with the tree

```
5 // The PHPCR tree starts from a root node
6 $root = $session->getRootNode();
7
8 // Add node unless it already exists
9 if (!$session->nodeExists('/example')) {
0     $node = $root->addNode('example', 'nt:unstructured');
1     $node->setProperty('property', 'value');
2     $session->save();
3 }
4
5 // Get a node by path
6 $node = $session->getNode('/example');
7 echo $node->getPropertyValue('property'); // "value"
```

# Queries

```
9 // Get the Query Manager
10 $qm = $session->getWorkspace()->getQueryManager();
11
12 // Prepare a SQL2 query
13 $query = $qm->createQuery('SELECT * FROM [nt:unstructured]
14                             WHERE property IS NOT NULL
15                             ORDER BY property ASC',
16                             \PHPCR\Query\QueryInterface::JCR_SQL2);
17
18 // Execute
19 $results = $query->execute();
20
21 // Loop through results
22 $nodes = $results->getNodes();
23 foreach ($nodes as $node) {
24     var_dump($node->getPath());
25     var_dump($node->getPropertyValue('property'));
26 }
```

# Versioning: commit

```
0 $node->setProperty('property', 'value');  
1 $session->save();  
2 // Commit current version of the node  
3 $vm->checkpoint('/example');  
4 // Change property value  
5 $node->setProperty('property', 'other value');  
6 $session->save();  
7 // Commit again  
8 $vm->checkpoint('/example');
```

# Export and import

```
25 // Export a node
26 $session->exportSystemView('/example',
27     fopen('/tmp/exportedNode.xml', 'w'),
28     true, true);
29
30 // Check that XML import is supported
31 if (!$repository->getDescriptor(\PHPCR
    \RepositoryInterface::OPTION_XML_IMPORT_SUPPORTED)) {
32     die("XML import not supported by repository\n");
33 }
34
35 // Import the node into a new path
36 $session->importXML('/test',
37     fopen('/tmp/exportedNode.xml', 'r'),
38     \PHPCR\ImportUUIDBehaviourInterface::IMPORT_UUID_CREATE_NEW);
```



# Status

5507 installs from Packagist

Used in production

PHPCR 2.1 in beta

# Built on PHPCR



The image shows a screenshot of a web browser displaying the homepage of the Symfony2 Content Management Framework (CMF). The browser's address bar shows the URL `cmf.symfony.com`. The page features a green header with the text "Symfony2 CMF" and "Symfony2 Content Management Framework". A dark grey navigation bar contains links for "Home", "News", "Get Started", "Get Involved", "About", "Docs", "Wiki", and "Demo". The main content area includes a section titled "The Symfony CMF Project" with a welcome message and a "Mission Statement" section.

Firefox Home | Symfony2 CMF

cmf.symfony.com

psr-1

Symfony2 + Doctrine2

# Symfony2 CMF

Symfony2 Content Management Framework

Home News Get Started Get Involved About Docs Wiki Demo

## The Symfony CMF Project

Welcome to the home of the [Symfony2](#) Content Management Framework. The project is organized by the Symfony community and has several sponsoring companies and prominent open source leaders implementing the philosophy of the [decoupled CMS](#). You can read learn more about the project on the [about](#) page.

## Mission Statement

The Symfony CMF project makes it easier for **developers** to add **CMS functionality** to applications built with the **Symfony2 PHP** framework. Key development principles for the provided **set of bundles** are **scalability, usability, documentation** and **testing**.

Considering to adopt



TYP03



*Fork*  
— CMS —



# Questions?

Fork me on GitHub

@bergie

[phpcr.github.com](https://phpcr.github.com)  
[decoupledcms.org](https://decoupledcms.org)

FreeNode #jackalope

