

Advanced project tracking with Arbit

International PHP Conference 2010

Kore Nordmann <kore@qafoo.com>

October 12, 2010



About me

- ▶ Kore Nordmann (<kore@php.net>, <kore@apache.org>, <kore@qafoo.com>)
- ▶ Each more than 10 years of professional PHP
- ▶ Open source enthusiasts
- ▶ Contributing to various FLOSS projects
- ▶ Founder of Qafoo GmbH
 - ▶ Provides training & consulting on PHP software quality tools & processes
 - ▶ Provides commercial support for Arbit

Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

The future

QA



What is arbit?

- ▶ An issue tracker on steroids.

[http://arbit.qafoo.com](#)



What is arbit?

- ▶ An issue tracker on steroids.
 - ▶ A “project tracker”



Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

The future

QA



Motivation

- ▶ Yet another issue tracker - why?
 - There are already so many out there...
 - ▶ Trac
 - ▶ Redmine
 - ▶ Mantis
 - Why develop another one?
 - Anybody got an idea?
 - What do you hate most about them?

Motivation

- ▶ Yet another issue tracker - why?
- ▶ There are already so many out there...
 - ▶ Trac
 - ▶ Redmine
 - ▶ Mantis
 - ▶ Jira
 - ▶ Bugzilla
 - ▶ Flyspray
 - ▶ ...

Why develop another one?

Why anybody got an idea?

What do you hate most about them?

Motivation

- ▶ Yet another issue tracker - why?
- ▶ There are already so many out there...
 - ▶ Trac
 - ▶ Redmine
 - ▶ Mantis
 - ▶ Jira
 - ▶ Bugzilla
 - ▶ Flyspray
 - ▶ ...
- ▶ So, why develop another one?

anybody got an idea?

what do you hate most about them?

Motivation

- ▶ Yet another issue tracker - why?
- ▶ There are already so many out there...
 - ▶ Trac
 - ▶ Redmine
 - ▶ Mantis
 - ▶ Jira
 - ▶ Bugzilla
 - ▶ Flyspray
 - ▶ ...
- ▶ So, why develop another one?
 - ▶ Anybody got an idea?
 - ▶ What do you hate most about them?



Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals

- ▶ Clean, extensible PHP code
- ▶ Support for easy multi-project support
- ▶ Support for custom extensions
- ▶ Well defined feature set
- ▶ Support for issue tracker, wiki, source browsing, notifications, ...
- ▶ Support for integrate Continuous Integration (CI) with issue tracking

Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals
 - ▶ Clean extensible PHP code

▶ Multi-project support

▶ Easy to extend, for custom extensions

▶ Defined feature set

▶ Issue tracker, wiki, source browsing, notifications, ...

▶ Support Continuous Integration (CI) with issue tracking

Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals
 - ▶ Clean extensible PHP code
 - ▶ Native, easy multi-project support

▶ Easy to extend, for custom extensions

▶ Well defined feature set

▶ Rich feature set: tracker, wiki, source browsing, notifications, ...

▶ Support for popular Continuous Integration (CI) with issue tracking

Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals
 - ▶ Clean extensible PHP code
 - ▶ Native, easy multi-project support
 - ▶ Modularized, for custom extensions

Designed feature set

Issue tracker, wiki, source browsing, notifications, ...

Supports Continuous Integration (CI) with issue tracking

Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals
 - ▶ Clean extensible PHP code
 - ▶ Native, easy multi-project support
 - ▶ Modularized, for custom extensions
- ▶ Originally planned feature set

Issue tracker, wiki, source browsing, notifications, ...
Integrate Continuous Integration (CI) with issue tracking

Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals
 - ▶ Clean extensible PHP code
 - ▶ Native, easy multi-project support
 - ▶ Modularized, for custom extensions
- ▶ Originally planned feature set
 - ▶ Issue tracker, wiki, source browsing, notifications, ...
 - ▶ Support Continuous Integration (CI) with issue tracking

Initial roadmap

- ▶ Started in February 2008
- ▶ Primary development goals
 - ▶ Clean extensible PHP code
 - ▶ Native, easy multi-project support
 - ▶ Modularized, for custom extensions
- ▶ Originally planned feature set
 - ▶ Issue tracker, wiki, source browsing, notifications, ...
 - ▶ Integrate Continuous Integration (CI) with issue tracking

Integrating CI

► What does CI mean?

- Running tests on each commit / every hour
- Analyze source code for defects
- Build releases on build success
- Report failures

► Build notification pending

► Build notification in Hudson, phpUnderControl, ...

► Build notification integration with issue trackers

► Combine knowledge about bug statistics with source metrics

► Get a good overview of a project in one tool

Integrating CI

- ▶ What does CI mean?
 - ▶ Running tests on each commit / every hour

- ▶ Analyze source code for defects

- ▶ Build releases on build success

- ▶ Build on build failures

- ▶ Build on build notification pending

- ▶ Integrate with build systems in Hudson, phpUnderControl, ...

- ▶ Integrate with issue trackers

- ▶ Combine knowledge about bug statistics with source metrics

- ▶ Get a good overview of a project in one tool

Integrating CI

- ▶ What does CI mean?

- ▶ Running tests on each commit / every hour
- ▶ Analyze source code for defects

- ▶ Build releases on build success

- ▶ Build on test failures

- ▶ Build on code conversation pending

- ▶ Integrate with Jenkins, Hudson, phpUnderControl, ...

- ▶ Integrate with issue trackers

- ▶ Combine knowledge about bug statistics with source metrics

- ▶ Get a good overview of a project in one tool

Integrating CI

- ▶ What does CI mean?
 - ▶ Running tests on each commit / every hour
 - ▶ Analyze source code for defects
 - ▶ Build releases on build success

Continuous Integration

Build automation pending

Integration in Hudson, phpUnderControl, ...

Integration with issue trackers

Combine knowledge about bug statistics with source metrics

Get a good overview of a project in one tool

Integrating CI

- ▶ What does CI mean?
 - ▶ Running tests on each commit / every hour
 - ▶ Analyze source code for defects
 - ▶ Build releases on build success
 - ▶ Report failures
 - ▶ Implementation pending

▶ Integrate CI in Hudson, phpUnderControl, ...

▶ Integration with issue trackers

▶ Combine knowledge about bug statistics with source metrics

▶ Get a good overview of a project in one tool

Integrating CI

- ▶ What does CI mean?
 - ▶ Running tests on each commit / every hour
 - ▶ Analyze source code for defects
 - ▶ Build releases on build success
 - ▶ Report failures
 - ▶ Implementation pending
- ▶ We got all of this in Hudson, phpUnderControl, ...

▶ Integration with issue trackers

▶ Combine knowledge about bug statistics with source metrics

▶ Get a good overview of a project in one tool

Integrating CI

- ▶ What does CI mean?
 - ▶ Running tests on each commit / every hour
 - ▶ Analyze source code for defects
 - ▶ Build releases on build success
 - ▶ Report failures
 - ▶ Implementation pending
- ▶ We got all of this in Hudson, phpUnderControl, ...
- ▶ Benefits of integration with issue trackers
 - ▶ Combine knowledge about bug statistics with source metrics
 - ▶ Get a good overview of a project in one tool

Dashboard



Arbit - project tracking

PQI*

Arbit aims to provide a decent modern extensible multi project tracking tool. Features start with issue tracking and wiki and do not stop before code analysis and translation management.

- `issue_tracker`: 62 open issues of 163 issues in issue tracker.
- `browse_source`: Checked out revision #1705 of project source.
- `documentation`: 6 pages available in wiki.
- `phpunit`: 630 tests run with PHPUnit, 0 failures (163.67 seconds, 864 assertions).
- `development_wiki`: 11 pages available in wiki.



PHPillow - PHP CouchDB connector

PQI*

PHPillow is a PHP CouchDB wrapper, which lays on top of the Couch and offers even more comfort laying anything into the database.

- `issue_tracker`: 5 open issues of 9 issues in issue tracker.
- `browse_source`: Checked out revision #152 of project source.
- `phpunit`: 165 tests run with PHPUnit, 2 failures (3.44 seconds, 238 assertions).



CI in Arbit



Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

The future

QA



Dependencies

- ▶ PHP \geq 5.3
- ▶ CouchDB \geq 0.9

▶ Alternative RDBMS backend (MySQL, PostgreSQL)
▶ Can be installed on Windows
▶ Can optionally use Doctrine 2

- ▶ Installation
 - ▶ Download PHAR or archive
 - ▶ Only for testing, not recommended for production
 - ▶ Installation instructions: <http://tracker.arbitracker.org/arbit/documentation/view/InstallationGuide>

Dependencies

- ▶ PHP \geq 5.3
- ▶ CouchDB \geq 0.9
 - ▶ Alternative RDBMS backend (MySQL, PostgreSQL, ...) is being worked on
 - ▶ Will probably use Doctrine 2
- ▶ Installation
 - ▶ Download PHAR or archive
 - ▶ Only for testing, not recommended for production
 - ▶ Installation instructions: <http://tracker.arbitracker.org/arbit/documentation/view/InstallationGuide>

Projects using arbit

- ▶ PHP_Depend (<http://tracker.pdepend.org/pdepend>)
- ▶ PHPMD
(http://tracker.pdepend.org/php_mess_detector)
- ▶ Dwoo (<http://bugs.dwoo.org/>)
- ▶ Arbit (<http://tracker.arbitracker.org>)
- ▶ Different companies for their customers

Contribute

- ▶ Already contributing:
 - ▶ Kore Nordmann (kore)
 - ▶ Jordi Boggiano (seldaek)
 - ▶ Manuel Pichler (mapi)
 - ▶ Arne Nordmann (norro)
 - ▶ Jakob Westhoff (jakob)
 - ▶ Hans-Christian Otto (hco)
 - ▶ Tobias Schlitt (toby)
 - ▶ Tobias Tom (tobias)
 - ▶ ... more contributors in sub projects like VCS-Wrapper
- ▶ We would love to welcome you in the community around arbit:
<http://arbitracker.org/arbit/participate.html>

Already working

- ▶ Issue tracker
- ▶ Wiki
- ▶ Source browser
- ▶ Source metrics rendering (phpDepend)
- ▶ API documentation rendering
- ▶ Continuous Integration (CI) (*experimental*)
 - ▶ PHPUnit
 - ▶ PHPCodeSniffer
 - ▶ phploc

Already working

- ▶ Issue tracker
- ▶ Wiki
- ▶ Source browser
- ▶ Source metrics rendering (phpDepend)
- ▶ API documentation rendering
- ▶ Continuous Integration (CI) (*experimental*)
 - ▶ PHPUnit
 - ▶ PHPCodeSniffer
 - ▶ phploc
- ▶ Demo!

Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

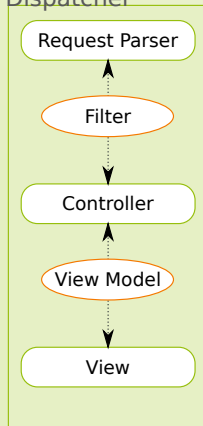
The future

QA

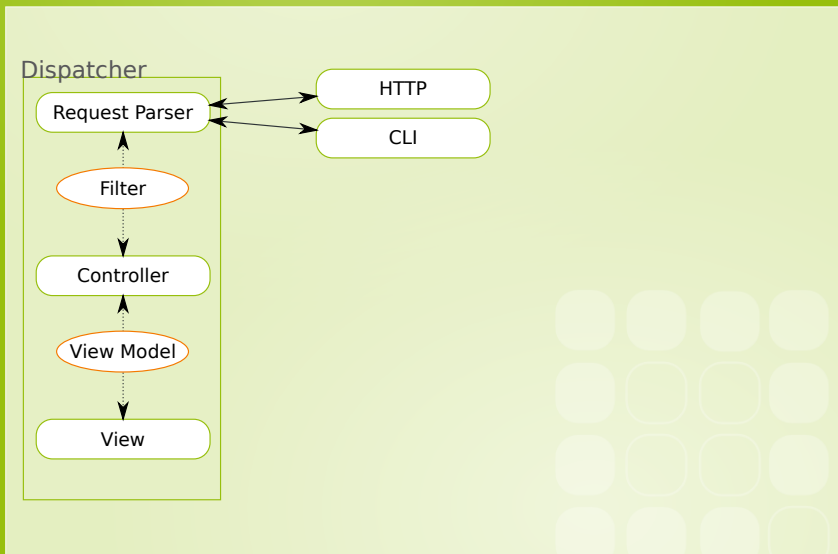


Arbit basic architecture

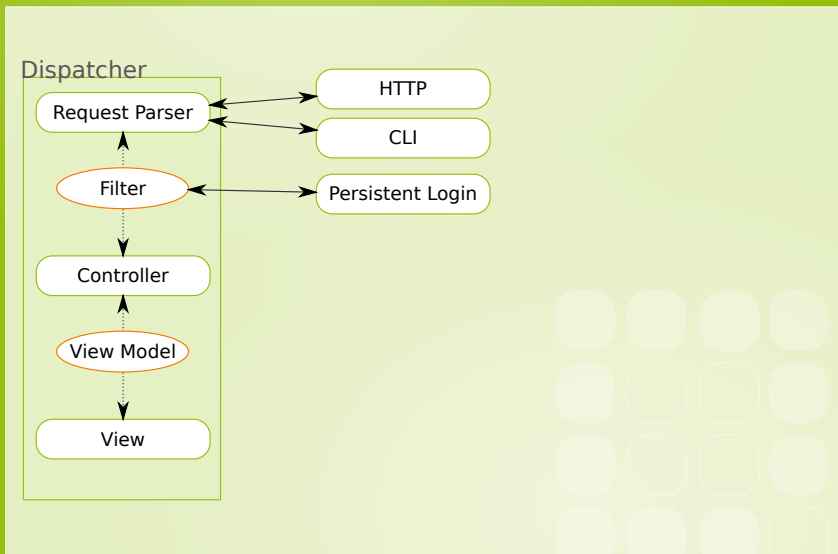
Dispatcher



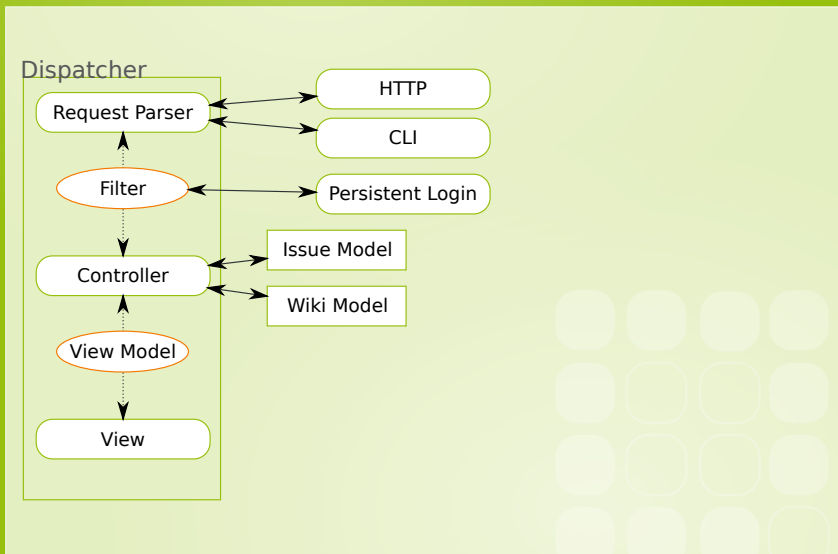
Arbit basic architecture



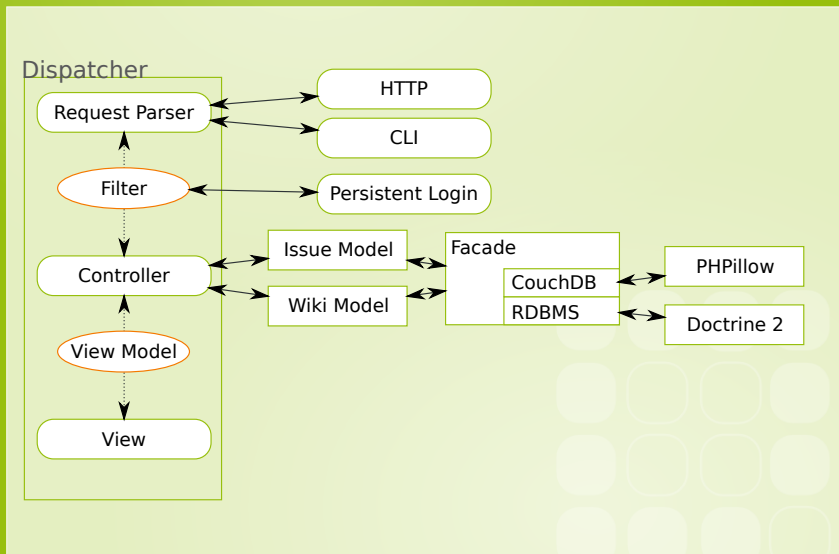
Arbit basic architecture



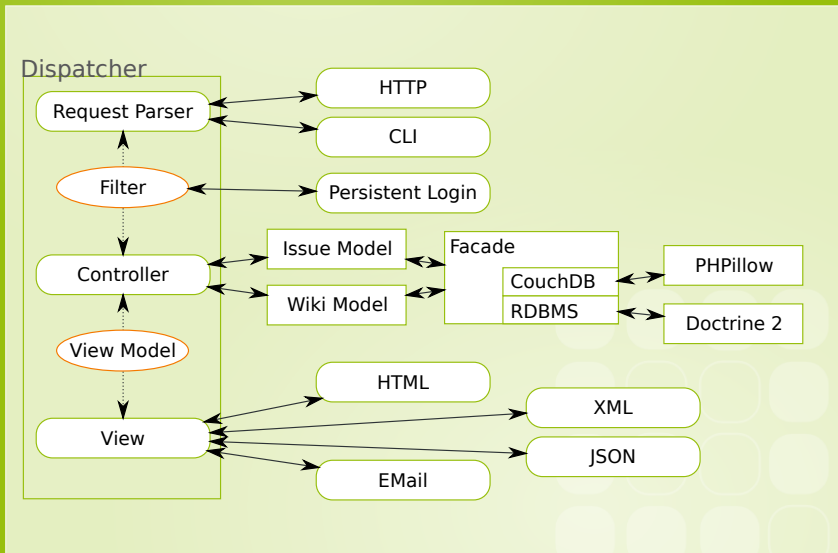
Arbit basic architecture



Arbit basic architecture



Arbit basic architecture



Module communication

- ▶ How can modules communicate?
 - ▶ No hard dependencies.
 - ▶ New modules should be able to interact with existing ones.



Module communication

- ▶ How can modules communicate?
 - ▶ No hard dependencies.
 - ▶ New modules should be able to interact with existing ones.
- ▶ Signal Slot



Signal slot

```
1 <?php
2
3 $handler = new arbitSignalSlot();
4
5 $handler->register( 'signalA', 'myModule::handleSignal' );
6 $handler->register( 'signalA', 'yourModule::handleSignal' );
7
8 // In module c
9 $handler->emit( 'signalA', array( /* data */ ) );
10
11 // Now all modules registered for this signal are called with the provided data
12
13 ?>
```

Practical example

► Inform other modules about source code updates

```
1 // modules/source/classes/controller.php +300
2 if ( $checkout->update( /* ... */ ) )
3 {
4     // ...
5     arbitSignalSlot::emit(
6         'sourceUpdated',
7         new arbitSourceUpdatedStruct( $checkout )
8     );
9 }
```

Practical example

- ▶ Other modules can specify signals to listen for

```
1 // modules/phpunit/definition.php
2 class arbitModulePhpunitDefintion extends arbitModuleDefintion
3 {
4     // ...
5     protected $properties = array(
6         // ...
7         'slots' => array(
8             'sourceUpdated' => 'arbitModulePhpunitController::sourceUpdated',
9         ),
10        // ...
11    );
```

Signals can also be converted into notifications (mail, ...)

Notifications are sent for "everything"

Practical example

- ▶ Other modules can specify signals to listen for

```
1 // modules/phpunit/definition.php
2 class arbitModulePhpunitDefintion extends arbitModuleDefintion
3 {
4     // ...
5     protected $properties = array(
6         // ...
7         'slots' => array(
8             'sourceUpdated' => 'arbitModulePhpunitController::sourceUpdated',
9         ),
10    // ...
11 );
```

- ▶ All signals can also be converted into notifications (mail, jabber, ...)

Notifications are sent for "everything"

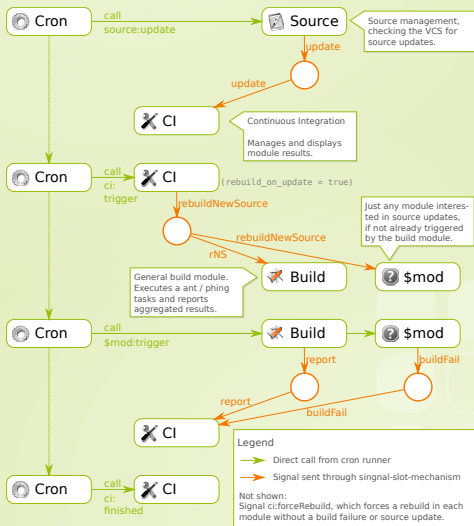
Practical example

- ▶ Other modules can specify signals to listen for

```
1 // modules/phpunit/definition.php
2 class arbitModulePhpunitDefintion extends arbitModuleDefintion
3 {
4     // ...
5     protected $properties = array(
6         // ...
7         'slots' => array(
8             'sourceUpdated' => 'arbitModulePhpunitController::sourceUpdated',
9         ),
10    // ...
11 );
```

- ▶ All signals can also be converted into notifications (mail, jabber, ...)
- ▶ Signals are sent for “everything”

Signal handling of the CI module



Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

The future

QA



Periodic

- ▶ Task runner implemented in PHP
 - ▶ Can parse cron-tables (vixie-cron dialect)
 - ▶ Can handle parallel running cron tasks
- ▶ Design document:
`http://arbitracker.org/periodic/design.html`
- ▶ Website: `http://arbitracker.org/periodic.html`

VCS-Wrapper

- ▶ Abstraction layer for version control system (VCS) read access
 - ▶ Implements support for: SVN, CVS, Git, Mercurial, Bazaar, (Archives)
 - ▶ Interfaces reflect which features are supported in which VCS
 - ▶ Supports: History, diffs, blaming, listings, ...
 - ▶ Even works on Windows! (except for CVS)
- ▶ Website: http://arbitracker.org/vcs_wrapper.html

PHPillow

- ▶ Lightweight PHP CouchDB library

- ▶ Features

- ▶ Simple document validation constraints
- ▶ Automatic synchronization of views
- ▶ Automatic versioning of documents
- ▶ Python compatible tool for dump and import
- ▶ Connection handlers
- ▶ HTTP stream wrapper
- ▶ Standalone HTTP protocol implementation

- ▶ Website: <http://arbitracker.org/phpillow.html>

PHPillow

- ▶ Lightweight PHP CouchDB library
- ▶ Features
 - ▶ Simple document validation constraints
 - ▶ Automatic synchronization of views
 - ▶ Automatic versioning of documents
 - ▶ couchdb-python compatible tool for dump and import
- ▶ `PHPillow` features:
 - ▶ `PHPillow` session handlers
 - ▶ `PHPillow` HTTP stream wrapper
 - ▶ `PHPillow` HTTP protocol implementation
- ▶ Website: <http://arbitracker.org/phpillow.html>

PHPillow

- ▶ Lightweight PHP CouchDB library
- ▶ Features
 - ▶ Simple document validation constraints
 - ▶ Automatic synchronization of views
 - ▶ Automatic versioning of documents
 - ▶ couchdb-python compatible tool for dump and import
- ▶ Different connection handlers
 - ▶ PHP HTTP stream wrapper
 - ▶ Custom HTTP protocol implementation
- ▶ Website: <http://arbitracker.org/phpillow.html>

Arbit installer

- ▶ Web installer for PHP projects
- ▶ Design phase finished
 - ▶ Results at: http://tracker.arbitracker.org/arbit/development_wiki/view/Installer
 - ▶ Please verify that it would work for *your* application
 - ▶ Proof of concept in implementation phase

`svn://arbitracker.org/arbit/projects/installer/trunk`

- ▶ Major feature of next arbit release

Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

The future

QA



The future

- ▶ Current development

- ▶ DIC branch of arbit:

- <http://github.com/kore/arbit-dic-branch>

- ▶ Next important tasks

- ▶ Implement easy-to-use web installer

- ▶ Improve and stabilize CI related modules

- ▶ Implement a RDBMS backend

- ▶ Roadmap: <http://tracker.arbitracker.org/>

- ▶ qa_tracker/roadmap

- ▶ Provides commercial support / consulting / training for

- ▶ Will always still stay a fully community-driven Open-Source

- ▶ Project

The future

- ▶ Current development
 - ▶ DIC branch of arbit:
<http://github.com/kore/arbit-dic-branch>
- ▶ Next important tasks
 - ▶ Implement easy-to-use web installer
 - ▶ Refactor and stabilize CI related modules
 - ▶ Implement a RDBMS backend

Project Tracker: <http://tracker.arbitracker.org/>

Project Tracker Roadmap: http://wiki.arbitracker.org/wiki/qa_tracker/roadmap

Qafoo provides commercial support / consulting / training for

Arbit will always still stay a fully community-driven Open-Source

project.

The future

- ▶ Current development
 - ▶ DIC branch of arbit:
<http://github.com/kore/arbit-dic-branch>
- ▶ Next important tasks
 - ▶ Implement easy-to-use web installer
 - ▶ Refactor and stabilize CI related modules
 - ▶ Implement a RDBMS backend
- ▶ Current roadmap: http://tracker.arbitracker.org/arbit/issue_tracker/roadmap

Qafoo provides commercial support / consulting / training for

Arbit will always still stay a fully community-driven Open-Source

project.

The future

- ▶ Current development
 - ▶ DIC branch of arbit:
<http://github.com/kore/arbit-dic-branch>
- ▶ Next important tasks
 - ▶ Implement easy-to-use web installer
 - ▶ Refactor and stabilize CI related modules
 - ▶ Implement a RDBMS backend
- ▶ Current roadmap: http://tracker.arbitracker.org/arbit/issue_tracker/roadmap
- ▶ Qafoo provides commercial support / consulting / training for arbit
 - ▶ Will always still stay a fully community-driven Open-Source project

Arbit feedback

- ▶ What would *you* like to see in arbit?
 - ▶ Special requirements
 - ▶ Important features



Outline

Introduction

Motivation

Current state

Architectural gems

Sub-Projects

The future

QA



Resources

- ▶ Website: <http://arbitracker.org/news.html>
- ▶ Issue tracker: <http://tracker.arbitracker.org/>
- ▶ CI: <http://tracker.arbitracker.org/arbit/ci>

The end

- ▶ Open questions?
- ▶ Further remarks?
- ▶ Contact
 - ▶ Mail: <kore@qafoo.com>
 - ▶ Web: <http://kore-nordmann.de> / <http://qafoo.com>
 - ▶ Twitter: @koredn / @qafoo