Advanced project tracking with Arbit

FrOSCon 2010

Kore Nordmann

August 23, 2010

About me

- Kore Nordmann (<kore@php.net>, <kore@apache.org>, <kore@qafoo.com>)
- Long time PHP developer
- Founder of Qafoo GmbH
  - Provides training & consulting on PHP software quality tools & processes
  - Provides commercial support for Arbit
- Active open source developer:
  - Apache Zeta Components (Graph, WebDav, Document), *Arbit*, PHPUnit, Torii, *PHPillow*, KaForkL, Image 3D, WCV, ...

Outline

What is arbit?

- An issue tracker on steroids.

What is arbit?

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

Outline

Motivation

- Yet another issue tracker - why?
- There are already so many out there, ...
  - Trac
  - Redmine
  - Mantis
  - ...
- ... 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
  - ...

...develop another one?

...ybody got an idea?

...hat 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?

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 Februrary 2008
- Primary development goals
  - Clean extensible PHP code
  - Native, easy multi-project support
  - Well-defined for custom extensions
  - Planned feature set
    - Tracker, wiki, source browsing, notifications, ...
    - Integrate Continuous Integration (CI) with issue tracking

Initial roadmap

- Started in Februrary 2008
- Primary development goals
  - Clean extensible PHP code

Initial roadmap

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

Initial roadmap

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

Initial roadmap

- Started in Februrary 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 Februrary 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 Februrary 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
  - Deployment pending
- Support for this in Hudson, phpUnderControl, ...
- Lots of 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
  - Test implementation pending
- Support for this in Hudson, phpUnderControl, ...
- Plugins for 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
  - Use 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

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 prnding

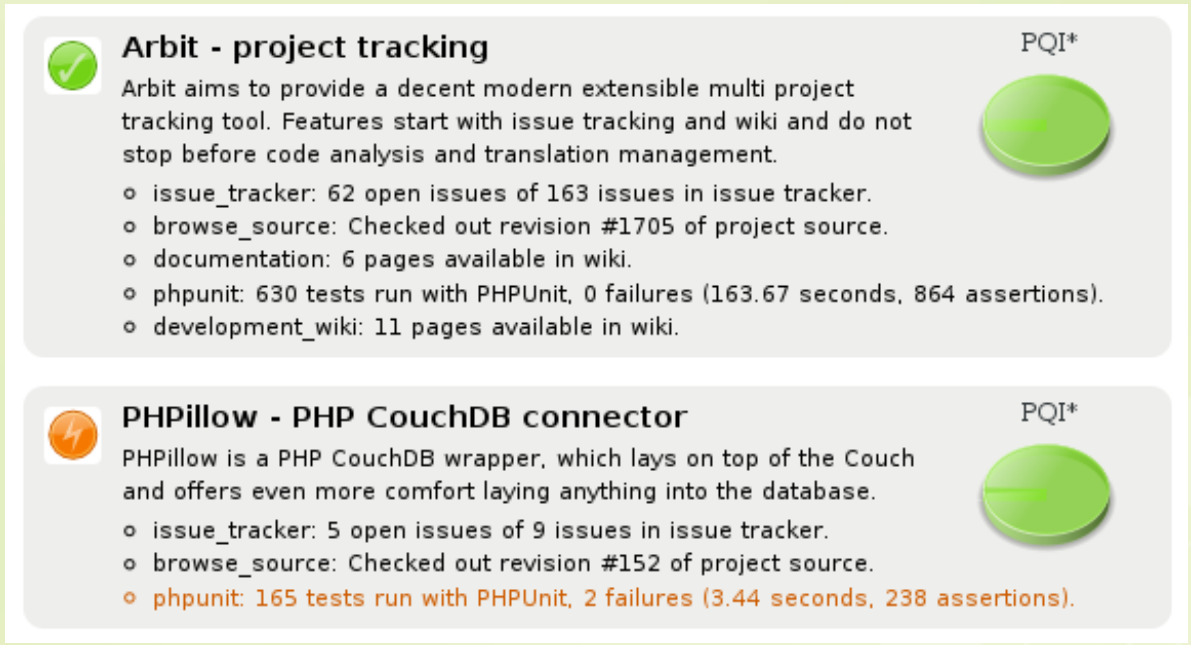
Integrating CI

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

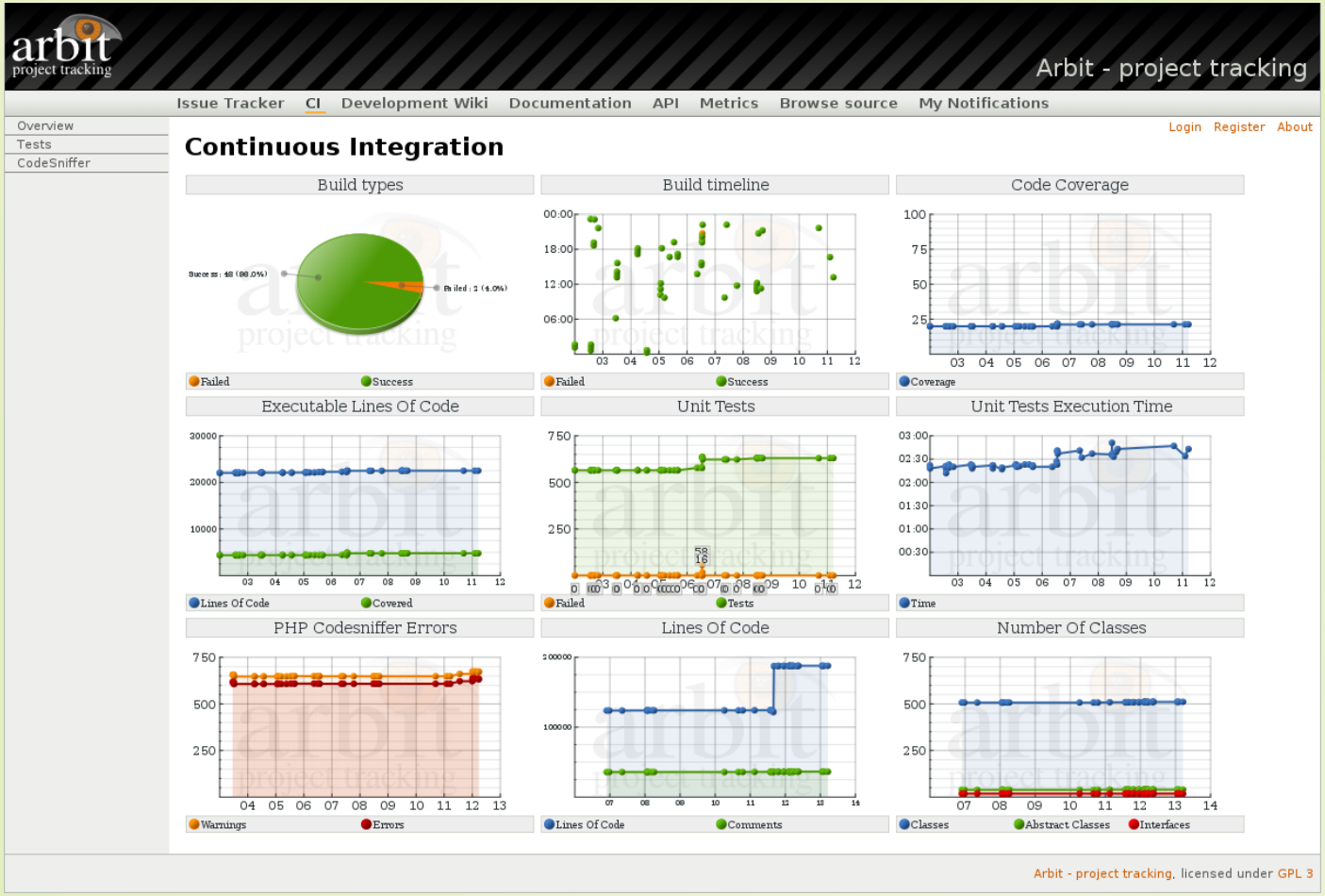
Integrating CI

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


arbit
project tracking
Arbit - project tracking
Issue Tracker
CI
Development Wiki
Documentation
API
Metrics
Browse source
My Notifications
Overview
Tests
CodeSniffer
Login
Register
About
Continuous Integration
Build types
Build timeline
Code Coverage
Failed
Success
Coverage
Executable Lines Of Code
Unit Tests
Unit Tests Execution Time
Lines Of Code
Covered
Tests
Time
PHP Codesniffer Errors
Lines Of Code
Number Of Classes
Warnings
Errors
Comments
Classes
Abstract Classes
Interfaces
Arbit - project tracking, licensed under GPL 3

Outline

Dependencies

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

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

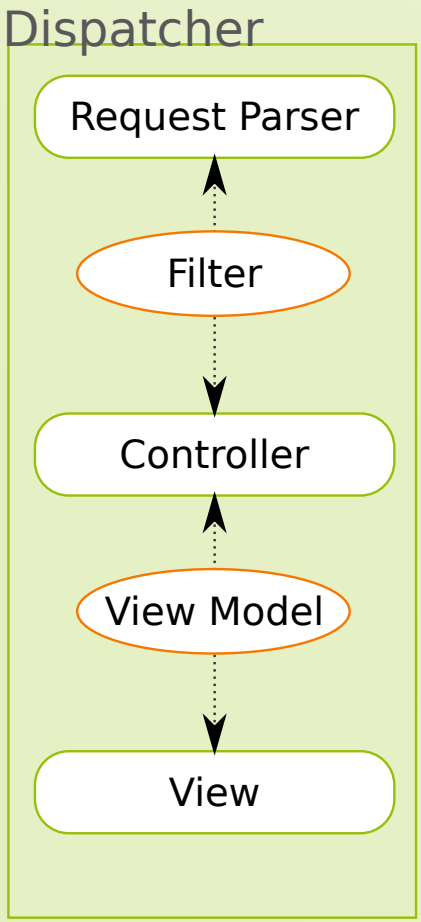
- Issue tracker
- Wiki
- Source browser
- Source metrics rendering (phpDepend)
- API documentation rendering
- Continuous Integration (CI) (*experimental*)
  - PHPUnit
  - PHPCodeSniffer
  - phploc

Already working

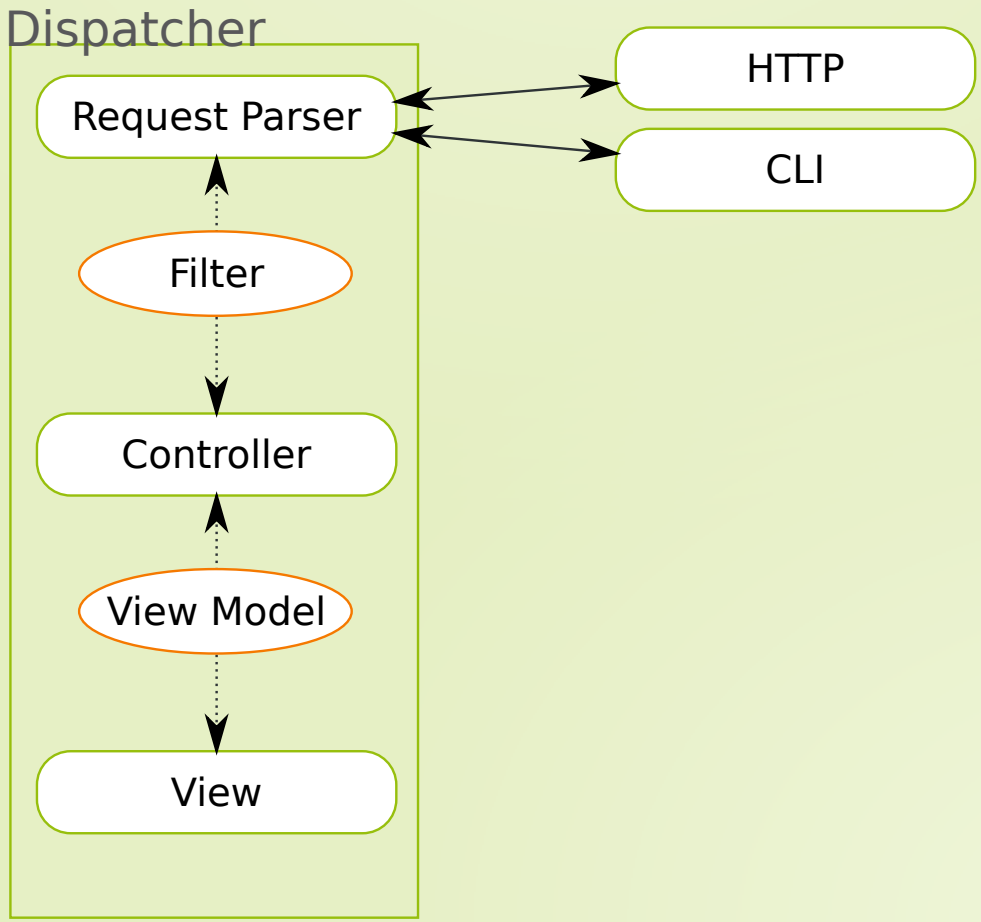
- Issue tracker
- Wiki
- Source browser
- Source metrics rendering (phpDepend)
- API documentation rendering
- Continuous Integration (CI) (*experimental*)
  - PHPUnit
  - PHPCodeSniffer
  - phploc
- Demo!

Outline

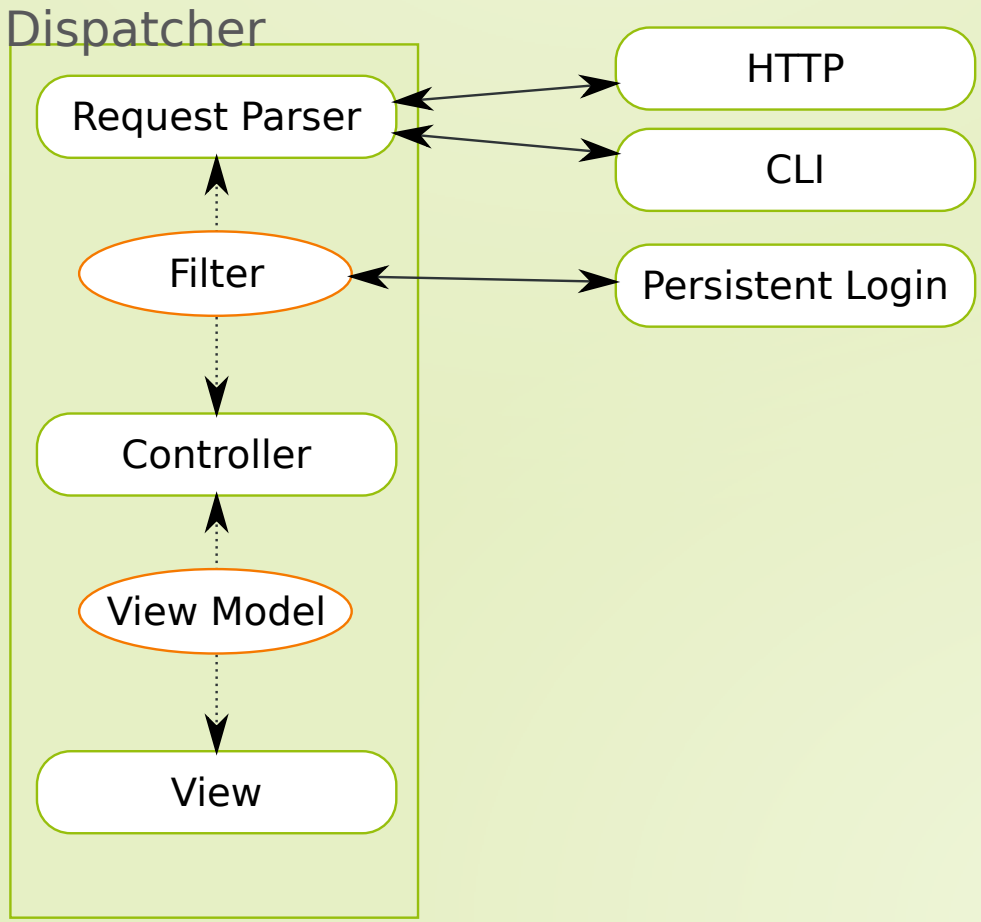
Arbit basic architecture



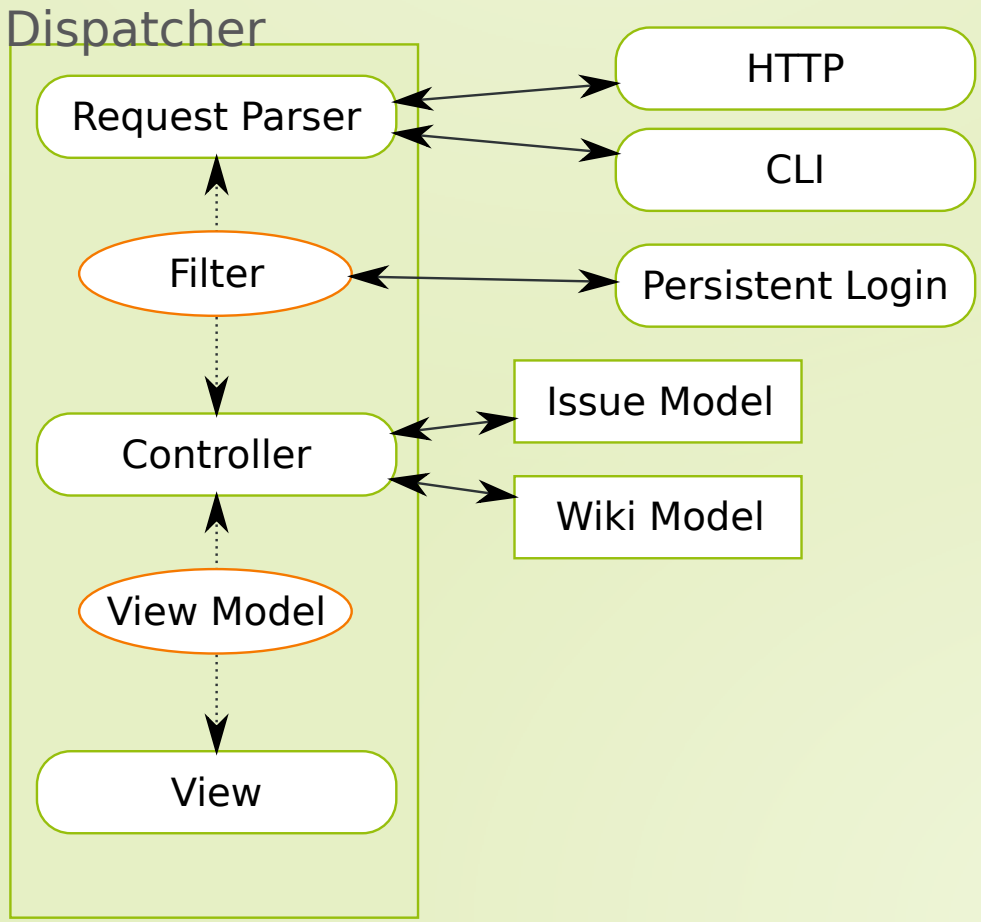
Arbit basic architecture



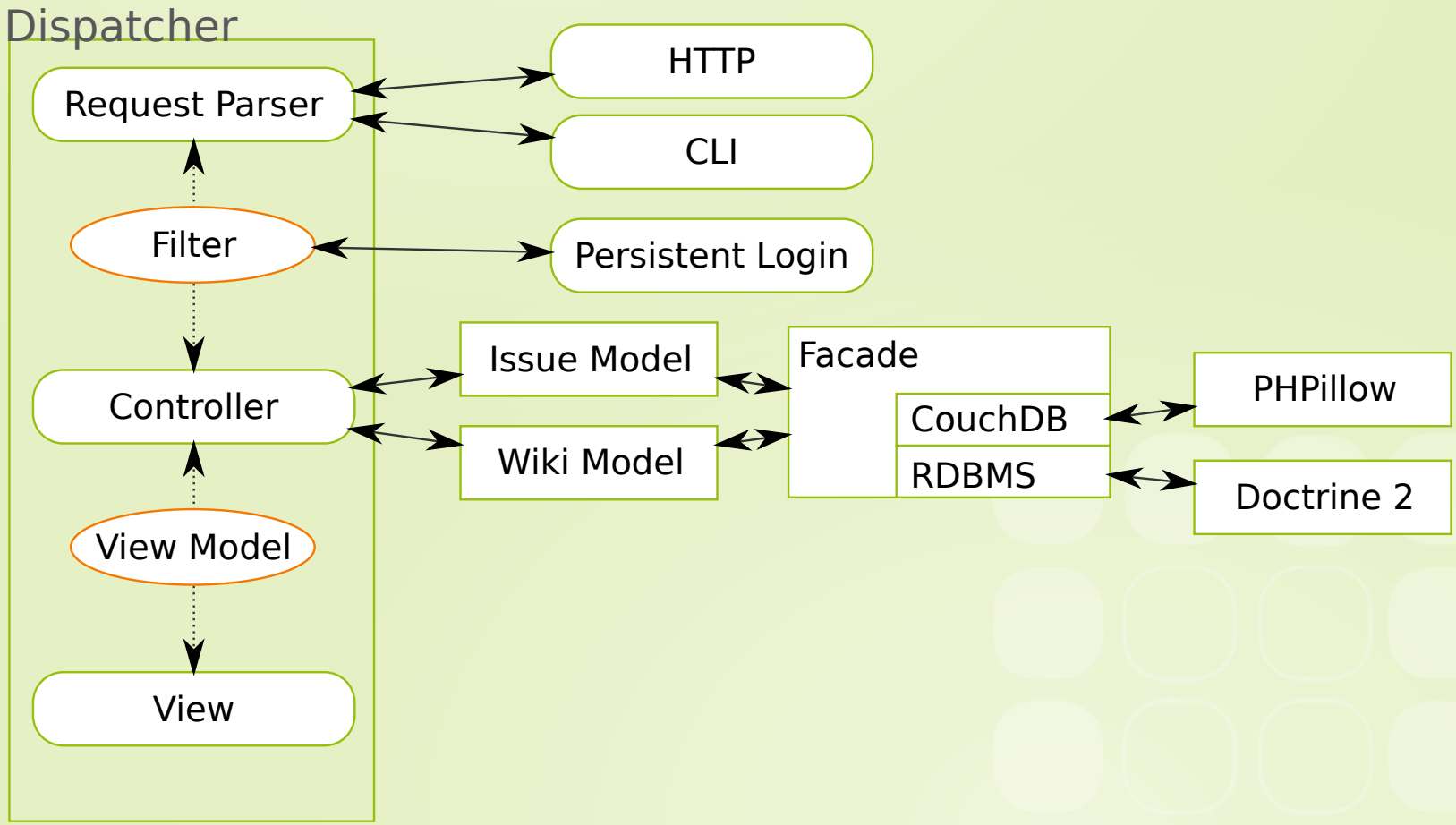
Arbit basic architecture



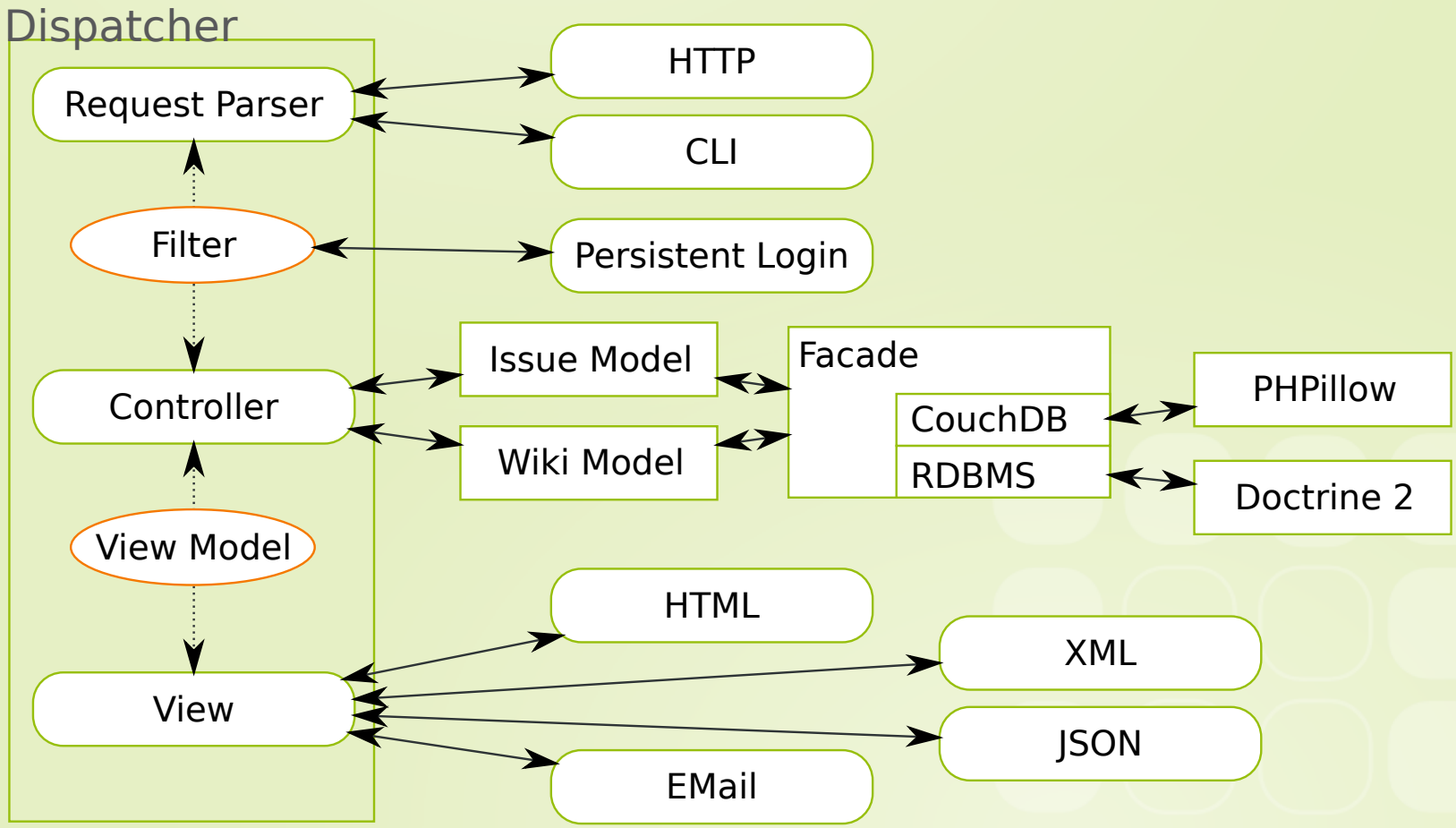
Arbit basic architecture



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

```php
<?php

$handler = new arbitSignalSlot();

$handler->register( 'signalA', 'myModule::handleSignal' );
$handler->register( 'signalA', 'yourModule::handleSignal' );

// In module c
$handler->emit( 'signalA', array( /* data */ ) );

// Now all modules registerd for this signal are called with the provided data

?>
```

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     );
```

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, ...)

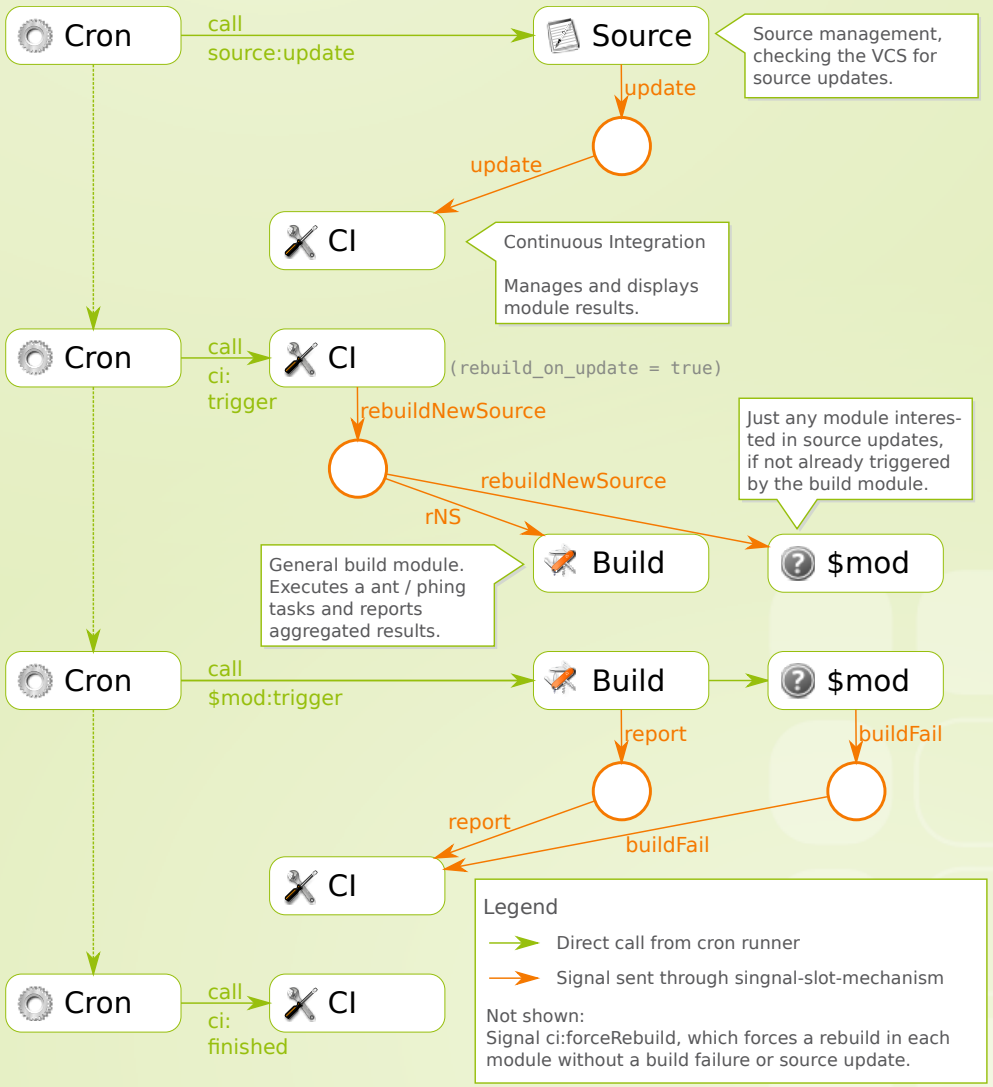
Practical example

- 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

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, ...
- 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
  - Couchdb-python compatible tool for dump and import
  - Connection handlers
  - HTTP stream wrapper
  - Custom 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
- 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

The future

- 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 cummunity-driven Open-Source project

Arbit feedback

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

Outline

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