

Apache Zeta Components

PHP Benelux 2011

Tobias Schlitt <toby@qafoo.com>

January 29, 2011



About me

► Tobias Schlitt (Toby)

- Degree in computer science
- More than 10 years of professional PHP
- Open source enthusiast

► Contributed to various FLOSS projects

► Member of Apache Zeta Components

► Founder of Qafoo GmbH (<http://qafoo.com>)

► Provides all around high quality PHP

► Includes support / development for Zeta

About me

- ▶ Tobias Schlitt (Toby)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiast
- ▶ Contributing to various FLOSS projects
- ▶ PMC members of Apache Zeta Components

CEO of Qafoo GmbH (<http://qafoo.com>)

provides all around high quality PHP

includes support / development for Zeta

About me

- ▶ Tobias Schlitt (Toby)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiast
- ▶ Contributing to various FLOSS projects
- ▶ PMC members of Apache Zeta Components
- ▶ Co-founder of **Qafoo GmbH** (<http://qafoo.com>)
 - ▶ services all around high quality PHP

▶ includes support / development for Zeta

About me

- ▶ Tobias Schlitt (Toby)
- ▶ Degree in computer science
- ▶ More than 10 years of professional PHP
- ▶ Open source enthusiast
- ▶ Contributing to various FLOSS projects
- ▶ PMC members of Apache Zeta Components
- ▶ Co-founder of **Qafoo GmbH** (<http://qafoo.com>)
 - ▶ services all around high quality PHP
 - ▶ includes support / development for Zeta

What to expect

- ▶ Some words on history
- ▶ Current project state
- ▶ Some useful code examples



What to expect

- ▶ Some words on history
- ▶ Current project state
- ▶ Some useful code examples

Shaked well . . .



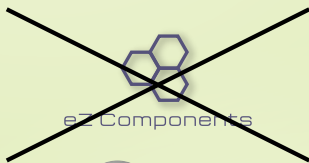
eZ Components



eZ Components



Apache Zeta Components



History: eZ Components

- ▶ Developed since 2004 by eZ Systems AS
 - ▶ Makers of enterprise OSS CMS eZ Publish
 - ▶ <http://ez.no>
- ▶ Initial goal
 - ▶ Refactor and cleanup core libs of eZ Publish
 - ▶ Pull out standalone, general purpose library
- ▶ Objectives
 - ▶ High quality code
 - ▶ Well thought out API
 - ▶ Backwards compatibility
 - ▶ Extensive documentation

Statistics

► 46 components

• 2,347 classes

• 1,051 test classes

• 1,000,000 lines / method

• 100,000,000 cycl. complex.

• 1,000,000 non-test LOC

• 100,000 CLOC



Statistics

▶ 46 components

▶ 2,347 classes

▶ 1,051 test classes

▶ 1,000,000 lines / method

▶ 100,000 cycl. complex.

▶ 1,000 non-test LOC

▶ 10,000 CLOC

Statistics

- ▶ 46 components
- ▶ 2,347 classes
 - ▶ 651 test classes

▶ 1,696 classes / method
▶ 1,000 cycl. complex.

▶ 1,000 non-test LOC
▶ 1,000 CLOC

Statistics

- ▶ 46 components
- ▶ 2,347 classes
 - ▶ 651 test classes
 - ▶ 18 avg. lines / method
 - ▶ 2.86 avg. cycl. complex.

1,696 non-test LOC

651 test LOC

Statistics

- ▶ 46 components
- ▶ 2,347 classes
 - ▶ 651 test classes
 - ▶ 18 avg. lines / method
 - ▶ 2.86 avg. cycl. complex.
- ▶ 759,092 LOC

▶ 500,000 non-test LOC

▶ 259,092 CLOC

Statistics

- ▶ 46 components
- ▶ 2,347 classes
 - ▶ 651 test classes
 - ▶ 18 avg. lines / method
 - ▶ 2.86 avg. cycl. complex.
- ▶ 759,092 LOC
 - ▶ 344,077 non-test LOC

344,077 LOC



Statistics

- ▶ 46 components
- ▶ 2,347 classes
 - ▶ 651 test classes
 - ▶ 18 avg. lines / method
 - ▶ 2.86 avg. cycl. complex.
- ▶ 759,092 LOC
 - ▶ 344,077 non-test LOC
 - ▶ 190,240 CLOC



The components

- ▶ Archive
- ▶ ConsoleTools
- ▶ Database
- ▶ Document
- ▶ Feed
- ▶ Graph
- ▶ Mail
- ▶ MvcTools
- ▶ SignalSlot
- ▶ Template
- ▶ Webdav
- ▶ Workflow

...

Intermission ...

The Graph component



The Zeta Graph component

```
1 require_once 'Base/src/ezc_bootstrap.php';
2
3 $graph = new ezcGraphPieChart();
4 $graph->palette = new ezcGraphPaletteEzRed();
5 $graph->title = 'Access_statistics';
6 $graph->options->label = '%2$d_(%3$.1f%%)';
7
8 $graph->data['Access_statistics'] = new ezcGraphArrayDataSet( array(
9     'Mozilla' => 19113,
10    'Explorer' => 10917,
11    'Opera' => 1464,
12    'Safari' => 652,
13    'Konqueror' => 474,
14 ) );
15 $graph->data['Access_statistics']->highlight['Explorer'] = true;
```

The Zeta Graph component

```
1 require_once 'Base/src/ezc_bootstrap.php';
2
3 $graph = new ezcGraphPieChart();
4 $graph->palette = new ezcGraphPaletteEzRed();
5 $graph->title = 'Access_statistics';
6 $graph->options->label = '%2$d_(%3$.1f%%)';
7
8 $graph->data['Access_statistics'] = new ezcGraphArrayDataSet( array(
9     'Mozilla' => 19113,
10    'Explorer' => 10917,
11    'Opera' => 1464,
12    'Safari' => 652,
13    'Konqueror' => 474,
14 ));
15 $graph->data['Access_statistics']->highlight['Explorer'] = true;
```

The Zeta Graph component

```
1 require_once 'Base/src/ezc_bootstrap.php';
2
3 $graph = new ezcGraphPieChart();
4 $graph->palette = new ezcGraphPaletteEzRed();
5 $graph->title = 'Access_statistics';
6 $graph->options->label = '%2$d_(%3$.1f%)';
7
8 $graph->data['Access_statistics'] = new ezcGraphArrayDataSet( array(
9     'Mozilla' => 19113,
10    'Explorer' => 10917,
11    'Opera' => 1464,
12    'Safari' => 652,
13    'Konqueror' => 474,
14 ) );
15 $graph->data['Access_statistics']->highlight['Explorer'] = true;
```

The Zeta Graph component

```
1 require_once 'Base/src/ezc_bootstrap.php';
2
3 $graph = new ezcGraphPieChart();
4 $graph->palette = new ezcGraphPaletteEzRed();
5 $graph->title = 'Access_statistics';
6 $graph->options->label = '%2$d_(%3$.1f%%)';
7
8 $graph->data['Access_statistics'] = new ezcGraphArrayDataSet( array(
9     'Mozilla' => 19113,
10    'Explorer' => 10917,
11    'Opera' => 1464,
12    'Safari' => 652,
13    'Konqueror' => 474,
14 ) );
15 $graph->data['Access_statistics']->highlight['Explorer'] = true;
```

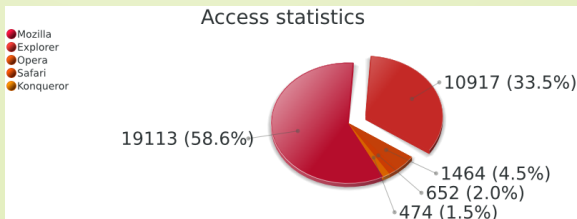
The Zeta Graph component

```
1 $graph->renderer = new ezcGraphRenderer3d();
2
3 $graph->renderer->options->dataBorder = 0.;
4 $graph->renderer->options->moveOut = .2;
5
6 $graph->renderer->options->pieChartOffset = 63;
7
8 $graph->renderer->options->pieChartGleam = .3;
9 $graph->renderer->options->pieChartGleamColor = '#FFFFFF';
10
11 // ...
12 $graph->render( 800, 300, 'pie_chart_3d.svg' );
```


The Zeta Graph component

```
1 $graph->renderer = new ezcGraphRenderer3d();
2
3 $graph->renderer->options->dataBorder = 0.;
4 $graph->renderer->options->moveOut = .2;
5
6 $graph->renderer->options->pieChartOffset = 63;
7
8 $graph->renderer->options->pieChartGleam = .3;
9 $graph->renderer->options->pieChartGleamColor = '#FFFFFF';
10
11 // ...
12 $graph->render( 800, 300, 'pie_chart_3d.svg' );
```

The Zeta Graph component



Back to the topic ...

History ...



History: The break

March 2010

- ▶ All eZ Components developers left eZ Systems

◀ Still, we wanted to ...

◀ ... put voluntary efforts into the project

◀ ... together

◀ ... so the code is not broken

◀ ... with eZ Systems

◀ ... and get the project to become independent

◀ ... or fork for or create a foundation

◀ ... and propose code to Apache Software Foundation

History: The break

March 2010

- ▶ All eZ Components developers left eZ Systems
- ▶ Still, we wanted to ...
 - ▶ put voluntary efforts into the project
 - ▶ push it further
 - ▶ ensure it is not broken

→ talk with eZ Systems

→ ask them to let the project to become independent

→ ask them for or create a foundation

→ propose code to Apache Software Foundation

History: The break

March 2010

- ▶ All eZ Components developers left eZ Systems
- ▶ Still, we wanted to ...
 - ▶ put voluntary efforts into the project
 - ▶ push it further
 - ▶ ensure it is not broken
- ▶ Discussion with eZ Systems
 - ▶ Spin off the project to become independent
 - ▶ Look for or create a foundation

proposed code to Apache Software Foundation

History: The break

March 2010

- ▶ All eZ Components developers left eZ Systems
- ▶ Still, we wanted to ...
 - ▶ put voluntary efforts into the project
 - ▶ push it further
 - ▶ ensure it is not broken
- ▶ Discussion with eZ Systems
 - ▶ Spin off the project to become independent
 - ▶ Look for or create a foundation
- ▶ Result: Proposed code to Apache Software Foundation

Apache Software Foundation



- ▶ <http://apache.org>
- ▶ Well known OSS foundation
- ▶ Home of OSS projects like
 - ▶ Apache HTTP server
 - ▶ Subversion
 - ▶ CouchDB

Incubation

- ▶ Before becoming an Apache project
- ▶ Goals:
 - ▶ Adjust to the ASF principles
 - ▶ Gather stable community
 - ▶ Sort out legal issues
 - ▶ Setup infrastructure

Incubation on 2010-05-21

Length of incubation phase unknown



Incubation

- ▶ Before becoming an Apache project
- ▶ Goals:
 - ▶ Adjust to the ASF principles
 - ▶ Gather stable community
 - ▶ Sort out legal issues
 - ▶ Setup infrastructure
- ▶ Entered incubation on 2010-05-21
- ▶ Duration of incubation phase unknown



The ASF way

- ▶ Meritocracy
 - ▶ Power through contribution
 - ▶ Project management comitee (PMC)

Open community

Open source
Open engineers
Openly votes
Open to the community

Apache License

Open source

The ASF way

- ▶ Meritocracy
 - ▶ Power through contribution
 - ▶ Project management comitee (PMC)
- ▶ Transparency
 - ▶ Public mailinglists
 - ▶ Decisions by votes
 - ▶ Involve the community

Apache License

The ASF way

- ▶ Meritocracy
 - ▶ Power through contribution
 - ▶ Project management comitee (PMC)
- ▶ Transparency
 - ▶ Public mailinglists
 - ▶ Decisions by votes
 - ▶ Involve the community
- ▶ Legal stuff
 - ▶ Apache License
 - ▶ CLAs



Intermission ...

The Mail component



Sending mail

```
1 $mail = new ezcMail();
2
3 $mail->from = new ezcMailAddress( 'sender@example.com', 'Bernard_Bernoulli' );
4 $mail->addTo( new ezcMailAddress( 'receiver@example.com', 'Wendy' ) );
5 $mail->subject = "This_is_the_subject_of_the_example_mail";
6
7 $textPart = new ezcMailText( "This_is_the_body_of_the_example_mail." );
8
9 $fileAttachment = new ezcMailFile( "~/myfile.jpg" );
10
11 $mail->body = new ezcMailMultipartMixed( $textPart, $fileAttachment );
12
13 $transport = new ezcMailMtaTransport();
14
15 $transport->send( $mail );
```

Sending mail

```
1 $mail = new ezcMail();
2
3 $mail->from = new ezcMailAddress( 'sender@example.com', 'Bernard_Bernoulli' );
4 $mail->addTo( new ezcMailAddress( 'receiver@example.com', 'Wendy' ) );
5 $mail->subject = "This_is_the_subject_of_the_example_mail";
6
7 $textPart = new ezcMailText( "This_is_the_body_of_the_example_mail." );
8
9 $fileAttachment = new ezcMailFile( "~/myfile.jpg" );
10
11 $mail->body = new ezcMailMultipartMixed( $textPart, $fileAttachment );
12
13 $transport = new ezcMailMtaTransport();
14
15 $transport->send( $mail );
```


Sending mail

```
1 $mail = new ezcMail();
2
3 $mail->from = new ezcMailAddress( 'sender@example.com', 'Bernard_Bernoulli' );
4 $mail->addTo( new ezcMailAddress( 'receiver@example.com', 'Wendy' ) );
5 $mail->subject = "This_is_the_subject_of_the_example_mail";
6
7 $textPart = new ezcMailText( "This_is_the_body_of_the_example_mail." );
8
9 $fileAttachment = new ezcMailFile( "~/myfile.jpg" );
10
11 $mail->body = new ezcMailMultipartMixed( $textPart, $fileAttachment );
12
13 $transport = new ezcMailMtaTransport();
14
15 $transport->send( $mail );
```

Sending mail

```
1 $mail = new ezcMail();
2
3 $mail->from = new ezcMailAddress( 'sender@example.com', 'Bernard_Bernoulli' );
4 $mail->addTo( new ezcMailAddress( 'receiver@example.com', 'Wendy' ) );
5 $mail->subject = "This_is_the_subject_of_the_example_mail";
6
7 $textPart = new ezcMailText( "This_is_the_body_of_the_example_mail." );
8
9 $fileAttachment = new ezcMailFile( "~/myfile.jpg" );
10
11 $mail->body = new ezcMailMultipartMixed( $textPart, $fileAttachment );
12
13 $transport = new ezcMailMtaTransport();
14
15 $transport->send( $mail );
```

Sending mail

```
1 $mail = new ezcMail();
2
3 $mail->from = new ezcMailAddress( 'sender@example.com', 'Bernard_Bernoulli' );
4 $mail->addTo( new ezcMailAddress( 'receiver@example.com', 'Wendy' ) );
5 $mail->subject = "This_is_the_subject_of_the_example_mail";
6
7 $textPart = new ezcMailText( "This_is_the_body_of_the_example_mail." );
8
9 $fileAttachment = new ezcMailFile( "~/myfile.jpg" );
10
11 $mail->body = new ezcMailMultipartMixed( $textPart, $fileAttachment );
12
13 $transport = new ezcMailMtaTransport();
14
15 $transport->send( $mail );
```

Sending mail

```
1 $mail = new ezcMail();
2
3 $mail->from = new ezcMailAddress( 'sender@example.com', 'Bernard_Bernoulli' );
4 $mail->addTo( new ezcMailAddress( 'receiver@example.com', 'Wendy' ) );
5 $mail->subject = "This_is_the_subject_of_the_example_mail";
6
7 $textPart = new ezcMailText( "This_is_the_body_of_the_example_mail." );
8
9 $fileAttachment = new ezcMailFile( "~/myfile.jpg" );
10
11 $mail->body = new ezcMailMultipartMixed( $textPart, $fileAttachment );
12
13 $transport = new ezcMailMtaTransport();
14
15 $transport->send( $mail );
```

Retrieving mail

```
1 $imap = new ezcMailmapTransport( "imap.example.com" );
2
3 $imap->authenticate( "user", "password" );
4 $imap->selectMailbox( 'Inbox' );
5
6 $imap->delete( 1 );
7 $imap->expunge();
8
9 $imap->noop();
10
11 $set = $imap->fetchByFlag( "DELETED" );
12
13 $parser = new ezcMailParser();
14
15 $mails = $parser->parseMail( $set );
```

Retrieving mail

```
1 $imap = new ezcMailmapTransport( "imap.example.com" );
2
3 $imap->authenticate( "user", "password" );
4 $imap->selectMailbox( 'Inbox' );
5
6 $imap->delete( 1 );
7 $imap->expunge();
8
9 $imap->noop();
10
11 $set = $imap->fetchByFlag( "DELETED" );
12
13 $parser = new ezcMailParser();
14
15 $mails = $parser->parseMail( $set );
```

Retrieving mail

```
1 $imap = new ezcMailmapTransport( "imap.example.com" );
2
3 $imap->authenticate( "user", "password" );
4 $imap->selectMailbox( 'Inbox' );
5
6 $imap->delete( 1 );
7 $imap->expunge();
8
9 $imap->noop();
10
11 $set = $imap->fetchByFlag( "DELETED" );
12
13 $parser = new ezcMailParser();
14
15 $mails = $parser->parseMail( $set );
```

Retrieving mail

```
1 $imap = new ezcMailmapTransport( "imap.example.com" );
2
3 $imap->authenticate( "user", "password" );
4 $imap->selectMailbox( 'Inbox' );
5
6 $imap->delete( 1 );
7 $imap->expunge();
8
9 $imap->noop();
10
11 $set = $imap->fetchByFlag( "DELETED" );
12
13 $parser = new ezcMailParser();
14
15 $mails = $parser->parseMail( $set );
```


Retrieving mail

```
1 $imap = new ezcMailmapTransport( "imap.example.com" );
2
3 $imap->authenticate( "user", "password" );
4 $imap->selectMailbox( 'Inbox' );
5
6 $imap->delete( 1 );
7 $imap->expunge();
8
9 $imap->noop();
10
11 $set = $imap->fetchByFlag( "DELETED" );
12
13 $parser = new ezcMailParser();
14
15 $mails = $parser->parseMail( $set );
```

Retrieving mail

```
1 $imap = new ezcMailmapTransport( "imap.example.com" );
2
3 $imap->authenticate( "user", "password" );
4 $imap->selectMailbox( 'Inbox' );
5
6 $imap->delete( 1 );
7 $imap->expunge();
8
9 $imap->noop();
10
11 $set = $imap->fetchByFlag( "DELETED" );
12
13 $parser = new ezcMailParser();
14
15 $mails = $parser->parseMail( $set );
```

Back to the topic ...

The ASF way ...



Everything new ...

- ▶ ... website
 - ▶ <http://incubator.apache.org/zetacomponents/>
 - ▶ (<http://zetacomponents.org>)
- ▶ ... SVN
 - ▶ <http://svn.apache.org/repos/asf/incubator/zetacomponents>
- ▶ ... mailinglists
 - ▶ Developers: zeta-dev@incubator.apache.org
 - ▶ Users: zeta-users@incubator.apache.org
 - ▶ Commits: zeta-commits@incubator.apache.org
- ▶ ... issue tracker
 - ▶ <https://issues.apache.org/jira/browse/ZETACOMP>
 - ▶ (<http://s.apache.org/zeta-issues>)

Everything new ...

<http://zetacomponents.org>

Current state

- ▶ In the works ...
 - ▶ First release
 - ▶ PEAR channel

▶ Accepting contributions

▶ Configuration override

▶ Support for MySQL Database

▶ Support for session component (Midgard)

▶ Support for PHP5 backend for Workflow

▶ Support for different users

▶ Support for PHP 3 Association

▶ Support for Midgard CMS

Current state

- ▶ In the works ...
 - ▶ First release
 - ▶ PEAR channel
- ▶ Pending contributions
 - ▶ Configuration override
 - ▶ Profiler for Database
 - ▶ Geolocation component (Midgard)
 - ▶ CouchDB backend for Workflow

Component users
→ PHP User Association
→ Midgard CMS



Current state

- ▶ In the works ...
 - ▶ First release
 - ▶ PEAR channel
- ▶ Pending contributions
 - ▶ Configuration override
 - ▶ Profiler for Database
 - ▶ Geolocation component (Midgard)
 - ▶ CouchDB backend for Workflow
- ▶ New prominent users
 - ▶ Typo 3 Association
 - ▶ Midgard CMS



Participate

- ▶ We need you!
- ▶ Getting involved:
 - ▶ Join the developer mainlinglist
 - ▶ Help fixing bugs
 - ▶ Check and write docs
 - ▶ Propose features / components

Quality PHP 5 code
Backwards compatibility
Generality / flexibility

Participate

- ▶ We need you!
- ▶ Getting involved:
 - ▶ Join the developer mainlinglist
 - ▶ Help fixing bugs
 - ▶ Check and write docs
 - ▶ Propose features / components
- ▶ Our goals:
 - ▶ High quality PHP 5 code
 - ▶ Backwards compatibility
 - ▶ Generality / flexibility



Thanks for listening

Are there any questions left?
(open part)



Thanks for listening

Please rate this talk at
<http://joind.in/2416>

Stay in touch

- ▶ Tobias Schlitt
- ▶ toby@qafoo.com
- ▶ @tobySen / @qafoo

Rent a PHP quality expert:
<http://qafoo.com>