

# Hands on eZ Components – A practical guide

- International PHP Conference
- Möhrfelden, 4<sup>th</sup> of November 2007
- Tobias Schlitt <ts@ez.no>

# Agenda

- What are eZ Components?
- Overview
- Setup
- Architecture
- Licensing & Contributing
- Support

# Agenda

- What are eZ Components?
- Overview
- Setup
- Architecture
- Licensing & Contributing
- Support

# What are eZ Components?

- Library for PHP 5 applications
- Loosely coupled components
- General purpose
- Enterprise directed
- Open Source

# Project target

- Migration to PHP 5.X (5.1.1)
- Thoroughly plan to get a top notch API
- Test driven development approach
- Proper docs from the start
- Flexibility and openness for the future

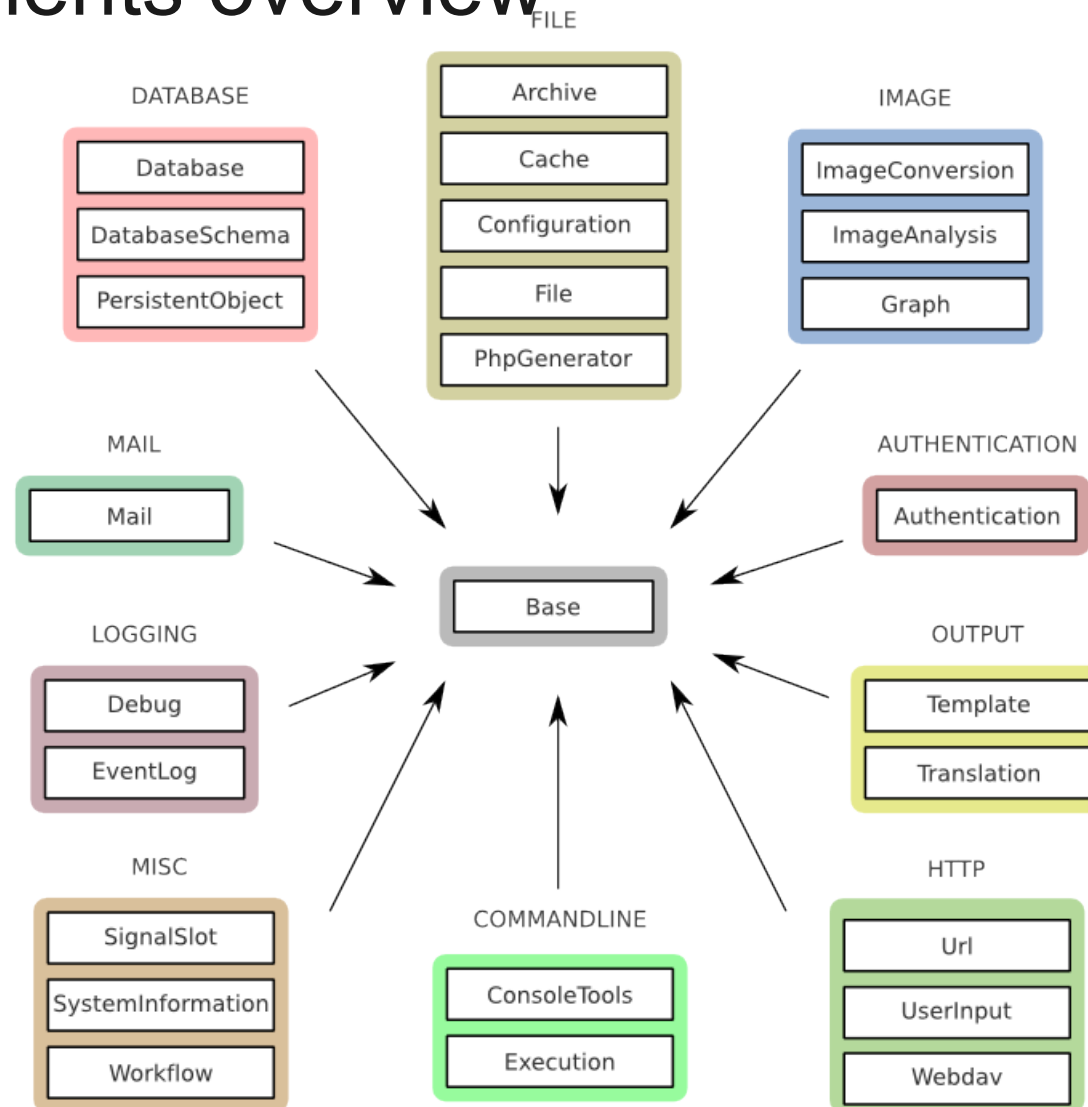
# Project goals

- Solid platform for PHP application development
- Clean and simple API
- Excellent docs
- Few regressions
- Clean IP
- Open Source friendly

# Agenda

- What are eZ Components?
- Overview
- Setup
- Architecture
- Licensing & Contributing
- Support

# Components overview





# Status

- Latest stable
  - 2007.1.1
- Upcoming stable
  - 2007.2
  - around X-mas
- Alpha releases
  - Authentication, Base, File, Graph, Mail, Url, Webdav, Workflow

# What is next

- New components in 2007.2
  - Webdav
  - Tree
- Additions in 2007.2
  - A lot...
  - We can browse the change logs, later

# Agenda

- What are eZ Components?
- Overview
- Setup
- Architecture
- Licensing & Contributing
- Support

# Obtaining

- 3 possible methods
  - Download bundle package
    - Convenient
    - Needs manual updates
  - Use PEAR Installer
    - Very convenient
    - Updates can be automated
  - Checkout from SVN
    - Convenient
    - Possibly contains unstable code

# Bundle package

- <http://ez.no/ezcomponents/download>
  - Download the package you desire
  - Stable version recommended
  - Unstable releases found at the bottom (marked as unsupported)
- Extract the package
- Follow common steps...

# PEAR Installer

- Check that PEAR is working
  - `$ pear`
- Discover ezc PEAR channel
  - `$ pear channel-discover components.ez.no`
- Choose your installation and install
  - `$ pear install -a ezc/eZComponents`
  - `$ pear install -a ezc/<component name>`
- Follow common steps...

# SVN checkout 1/2

- Checkout component repository
  - `$ svn co http://svn.ez.no/svn/ezcomponents/trunk`
- Checkout scripts
  - `$ svn co http://svn.ez.no/svn/ezcomponents/scripts`
- Perform setup
  - `$ scripts/setup-env.sh`
  - `$ scripts/setup-env.bat`  
Should be re-run on every update on Windows!
- Follow common steps...

# SVN checkout 2/2

- More interesting stuff in SVN
  - Upcoming components
    - <http://svn.ez.no/svn/ezcomponents/experimental>
  - Articles, internal documentation,...
    - <http://svn.ez.no/svn/ezcomponents/docs>
  - All components releases, by component
    - <http://svn.ez.no/svn/ezcomponents/releases>



# Common steps

- Check your `include_path`
  - Either in `php.ini`
  - Or in the script using `ini_set()`
- Include Base component
  - Beware: PEAR structure differs!
- Setup `__autoload()`

# Agenda

- What are eZ Components?
- Overview
- Setup
- **Architecture**
- Licensing & Contributing
- Support

# A component

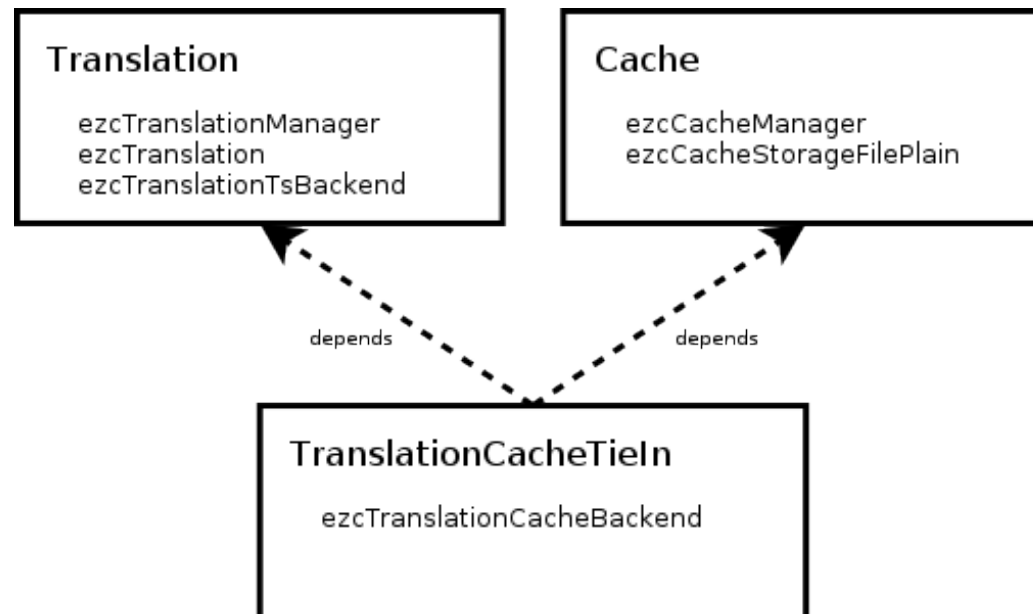
- Fulfills a certain purpose
- Collection of PHP Classes
- Important classes marked as @mainclass
- Directory structure
  - design/
  - docs/
  - src/
  - tests/

# Dependencies

- PHP version
  - At last 5.1.1
  - 5.2.x recommended
- Extensions / 3<sup>rd</sup> party apps
  - General
    - SPL
    - DOM
  - Component specific
    - PDO (Database components)
    - GD / ImageMagick (Image\* components)

# Tie-Ins

- All components depend on Base
- Inter-component deps discouraged
- Integration of components through Tie-Ins



# Base component

- Basic functionalities
  - Autoload
  - Dependency checks
  - File operations
- Common features
  - Common base classes
    - Structs
    - Options
    - Exceptions
  - Lazy initialization

# Struct classes

- Extending `ezcBaseStruct`
- No functionality
- Just public properties
- Used instead of hash tables
  - Readability
  - Type check possibilities using `instanceof`

# Settings and options

- Settings

- Only provided during construction of an object
- Cannot be changed later
- Define global behavior of the object

- Options

- Can be provided during construction of an object
- Can be changed at any time
- Access through `$options` property of the object
- Options container class extending `ezcBaseOptions` using virtual properties



# Virtual properties

- Used in every component
- Access of properties through interceptors
  - `__get()` / `__set()` / `__isset()`
- Provided type and value checks on the fly
- Exceptions instead of warnings/notices/nothing
  - `ezcBaseValueException`
  - `ezcBasePropertyNotFoundException`
  - `ezcBasePropertyPermissionException`

# Lazy initialization

- Configure objects without instantiation
- Instantiate objects when needed
- Saves resources
- Supported by several components



# Agenda

- What are eZ Components?
- Overview
- Setup
- Architecture
- Support
- Licensing & Contributing

# Documentation 1/2

- API docs
  - Online
    - <http://ez.no/doc/components/overview/latest>
  - For download
    - <http://ez.no/ezcomponents/download>
  - Generate yourself
    - phpDocumentor (<http://phpdoc.org>)
    - Restructured Text (<http://docutils.sourceforge.net/rst.html>)

# Documentation 2/2

- Tutorials
  - Shipped with each component
  - In the online API docs
- Articles online
  - <http://ez.no/developer/articles>
- Magazine articles
  - PHP Magazin (German / international)
  - php|arch
  - PHP Solutions

...



# Support

- Community

- Mailinglist

- [<components@lists.ez.no>](mailto:components@lists.ez.no)

- Forum

- [http://ez.no/developer/forum/ez\\_components](http://ez.no/developer/forum/ez_components)

- IRC

- #ezcomponents@Freenode

- Professional

- [http://ez.no/support/total\\_product\\_responsibility\\_from\\_ez](http://ez.no/support/total_product_responsibility_from_ez)

# Agenda

- What are eZ Components?
- Overview
- Setup
- Architecture
- Support
- Licensing & Contributing



# License

- New BSD License
  - <http://ez.no/eZcomponents/license>
- Very permissive
  - Even ship eZ Components with your product
- GPL friendly



# Contributing

- Open Source
  - Feel invited to contribute!
- Contributor License Agreement (CLA)
  - Signed by every contributor
  - Secures all involved parties
    - Customer
    - Contributor
    - eZ
  - Leaves copyright untouched

# Next step

- Now it's your decision
- What would you like to do?