

Welcome!

# PHP Extensions - What and Why

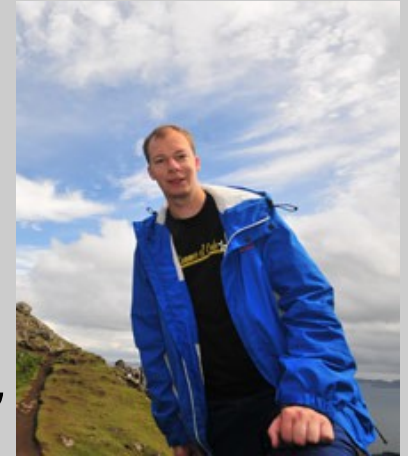
PHP North West - Manchester, UK - Oct 8th, 2011  
Derick Rethans - [derick@derickrethans.nl](mailto:derick@derickrethans.nl) - twitter:  
[@derickr](https://twitter.com/derickr)

<http://derickrethans.nl/talks.html>

<http://joind.in/3586>

## Derick Rethans

- Dutchman living in London
- PHP development
- Author of the `mcrypt`, `input_filter`, `dbus`, and date/time extensions
- Author of Xdebug
- Contributor to the Apache Zeta Components Incubator project (formerly eZ Components)
- Freelancer doing PHP (internals) development
- Anderskor



# Extension on a slide

```
helloworld.c (~/dev/php/hello-world) - VIM - Terminal
PHP_ARG_ENABLE(helloworld, whether to enable helloworld support,
  --enable-helloworld      Enable helloworld support)

if test "$PHP_helloworld" != "no"; then
  PHP_NEW_EXTENSION(helloworld, helloworld.c, $ext_shared)
fi
config.m4
2,1 All
#ifndef PHP_HELLOWORLD_H
#define PHP_HELLOWORLD_H

#include "php.h"

extern zend_module_entry helloworld_module_entry;
#define phpext_helloworld_ptr &helloworld_module_entry

#ifdef PHP_WIN32
#define PHP_HELLOWORLD_API __declspec(dllexport)
#else
#define PHP_HELLOWORLD_API
#endif

#ifdef ZTS
#include "TSRM.h"
#endif

PHP_FUNCTION(helloworld);

#endif
~
~
~
~
~
~
php_helloworld.h 17,1 All
#ifdef HAVE_CONFIG_H
#include "config.h"
#endif

#include "php.h"
#include "php_helloworld.h"
#include "ext/standard/info.h"

#ifdef COMPILE_DL_HELLOWORLD
ZEND_GET_MODULE(helloworld)
#endif

PHP_FUNCTION(helloworld)
{
    php_printf("Hello World!\n");
}

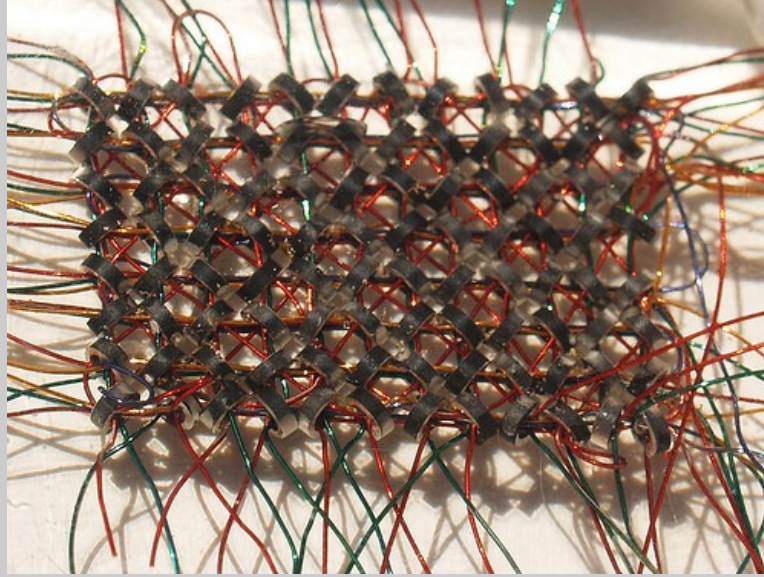
zend_function_entry helloworld_functions[] = {
    PHP_FE(helloworld, NULL)
    {NULL, NULL, NULL}
};

zend_module_entry helloworld_module_entry = {
    STANDARD_MODULE_HEADER,
    "helloworld",
    helloworld_functions,
    NULL, NULL, NULL, NULL,
    "0.0.1",
    STANDARD_MODULE_PROPERTIES
};
helloworld.c 30,2 All
```

# What?



- Extensions add functionality to PHP
- Extensions replace functionality in PHP
- PHP's extension mechanism is easy
- PHP's extension mechanism is (too) powerful



ext/standard

- Contains all the default PHP functions
- Can not be disabled

Other core extensions are:

- ext/date
- ext/ereg
- ext/pcre
- ext/reflection



Everything in ext/

- Are compiled into PHP
- Can most of the time be compiled as shared extensions as well ← distributions like to do that (too much)



Everything not in the core distribution

- Are often not compiled into PHP, but instead are compiled as shared objects
- Sometimes make their way into the core distribution, and sometimes they are moved away



## PECL: PHP Extension C Library

- <http://pecl.php.net>
- A whole load of extensions for various interesting (and odd things)
- Are installed with the PECL tool
- Used to be called "Siberia"
- Don't really have to be hosted in PHP's SVN repository, on the PECL website

# A few examples...

- Core distribution: bcmath bz2 calendar com\_dotnet ctype curl date dba dom enchant ereg exif fileinfo filter ftp gd gettext gmp hash iconv imap interbase intl json ldap libxml mbstring mcrypt mssql mysql mysqli mysqlnd oci8 odbc openssl pcntl pcre pdo pdo\_dblib pdo\_firebird pdo\_mysql pdo\_oci pdo\_odbc pdo\_pgsql pdo\_sqlite pgsql phar posix pspell readline recode reflection session shmop simplexml skeleton snmp soap sockets spl sqlite sqlite3 standard sybase\_ct sysvmsg sysvsem sysvshm tidy tokenizer wddx xml xmlreader xmlrpc xmlwriter xsl zip zlib
- PECL: activescript adt amfext amqp apache\_accessor apc apd apm archive ares automap axis2 bbcode bcompiler bitset blenc bloomy bz2\_filter cairo chasen classkit colorer courierauth cpdf crack cubrid cvsclient cybercash cybermut cyrus daffodildb date dazuko db dbase dbdo dbplus dbus dbx demoext dio docblock domxml doublemetaphone drizzle dtrace ecasound enchant esmtp etpan event expect fam fbsql fdf ffi filepro flitetts framgrab freeimage fribidi fuse gdchart gearman gender geoip gmagick gnupg gupnp haru hidef html\_parse htscanner http hwapi hyperwave ibm\_db2 id3 idn iisfunc imagick ims included informix ingres inotify intercept intl iptcdata ixsfunc judy kadm5 kakasi krb5 ktaglib lchash libevent libexif litespeed llvm lua lzf mailparse markdown maxdb mcal mcrypt\_filter mcve mdbtools memcache memcached memsession memtrack ming mnogosearch mono mqseries msession msqldb muscat myphp mysqlnd\_mc mysqlnd\_ms mysqlnd\_pscache mysqlnd\_qc mysqlnd\_sip mysqlnd\_uh namazu ncurses Net\_Gopher netools newt notes ntuser oauth ocal ogg Vorbis opal opendir openssl operator optimizer oracle ovrimos pam paradox params parsekit parse\_tree pdf pdo\_4d pdo\_ibm pdo\_ids pdo\_informix pdo\_user perl pfpro phar philter phk phpdoc picosql pop3 prephp preprocessor printer proctitle ps python qtdom radius rar rb rpc rpmreader rrd rsvg rsync runkit sam sasl satellite scream sdo selinux servlet session\_pgsql shp simple\_cvs smbc smtp sndfile solr sphinx spl\_types spplus spread sqlite ssdeep ssh2 statgrab stats stem stomp svn swf swish sybase tcc tclink tcpwrap tdb tests textcat threads timezonedb translit txforward unicode uploadprogress usblib uuid v8js valkyrie varnish vpopmail w32api wbxm weakref win32ps win32scheduler win32service win32std wincache wxwidgets xattr xdiff xdom xhprof xmgrace xml xmlrpc xmms xquery xrange xslcache xslt yaf yami yaml yaz yp zeroconf zip zlib\_filter

# A quick overview...

## Internal Extensions

### Core Extensions

ext/standard  
ext/date  
ext/ereg  
ext/pcre  
ext/reflection  
ext/spl

### Bundled Extensions

ext/bcmath  
ext/bz2  
ext/calendar  
ext/com\_dotnet  
ext/ctype  
ext/curl  
ext/dba  
ext/dom  
ext/enchant  
ext/exif  
ext/fileinfo  
ext/filter  
ext/ftp  
...

## External Extensions

### PECL Extensions

pecl/activestscript  
pecl/adt  
pecl/amfext  
pecl/amqp  
pecl/apache\_accessor  
pecl/apc  
pecl/apd  
pecl/apm  
...  
pecl/yaz  
pecl/yp  
pecl/zeroconf  
pecl/zip  
pecl/zlib\_filter

### Other Extensions

github.com/facebook/xhprof  
github.com/facebook/xhp  
github.com/fhoenig/Kellner  
github.com/fujimoto/php-fuse  
github.com/mkoppanen/php-zmq  
github.com/mgdm/php-pango  
...

# Traditional parts



- Module init/shutdown
- Request init/shutdown
- Functions
- Configuration (INI) settings

# Modern parts



- Classes and methods
- Argument description
- Dependencies on other extensions



## Normal extensions

- Module and request init and shutdown; and post module deinitialisation
- `phpinfo()`

## Zend extensions

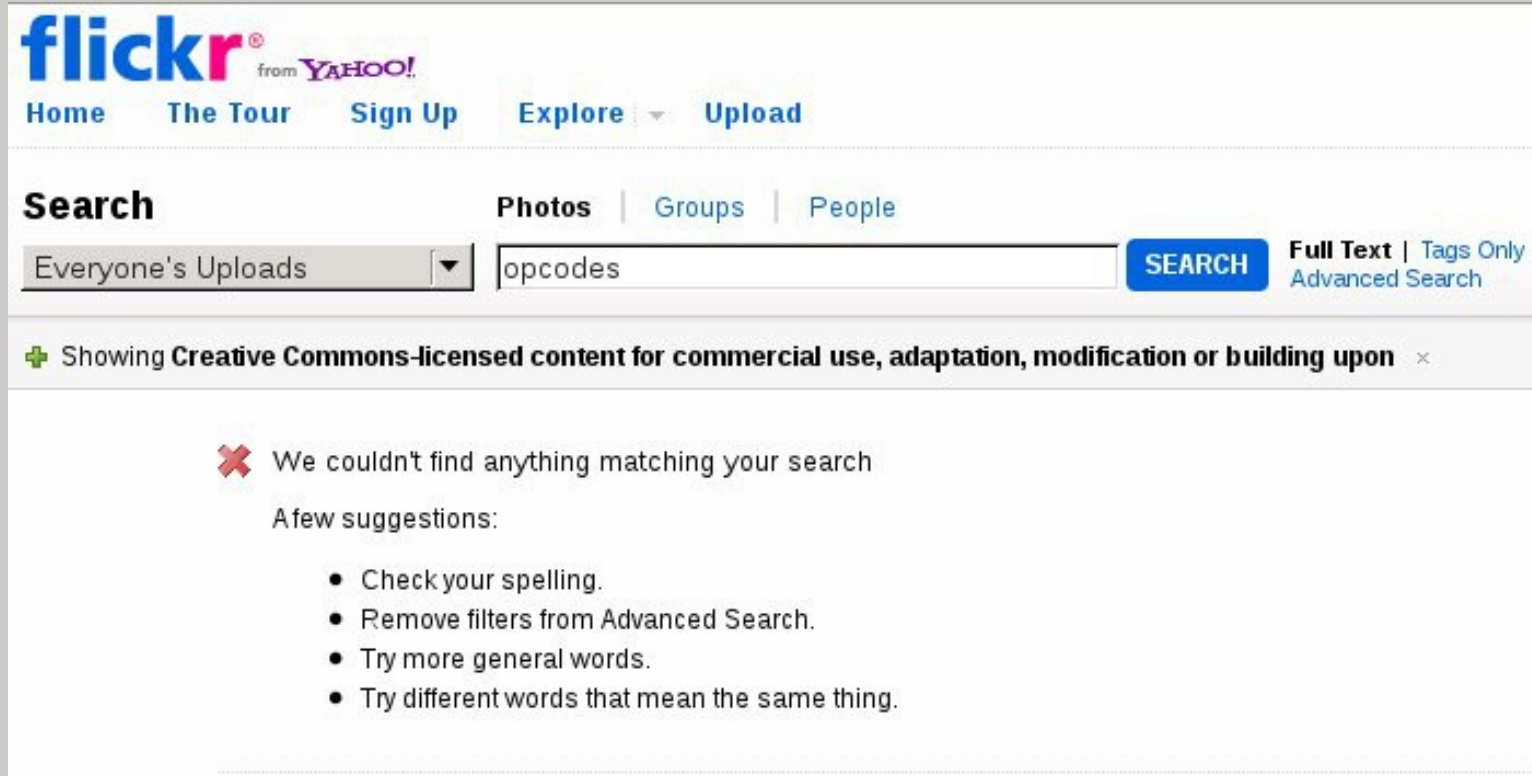
- statement calls
- all sorts of internal handlers: activate/deactivate functions, `op_array` handler, `op_array` constructor and destructor

# Overloading



- Error handler: Xdebug, SOAP
- Exception throwing handler
- Compiler and executor: APC and other caches, funcall, Xdebug
- Setting of headers
- Opcodes: Xdebug, operator

# Opcode overloading



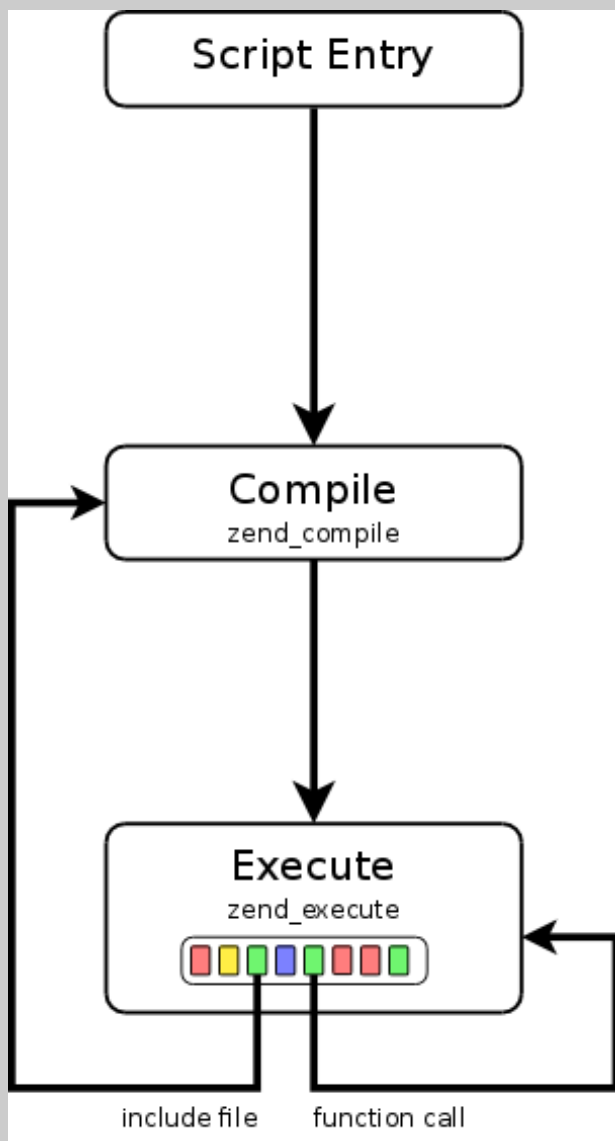
The screenshot shows the Flickr website's search interface. At the top left is the Flickr logo with the text "from YAHOO!". Navigation links include "Home", "The Tour", "Sign Up", "Explore", and "Upload". The search bar is titled "Search" and has tabs for "Photos", "Groups", and "People". A dropdown menu is set to "Everyone's Uploads". The search input field contains the text "opcodes". To the right of the input is a blue "SEARCH" button and links for "Full Text | Tags Only" and "Advanced Search". Below the search bar, a green plus icon indicates "Showing Creative Commons-licensed content for commercial use, adaptation, modification or building upon". The search results area shows a red "X" icon and the message "We couldn't find anything matching your search". Below this, it says "A few suggestions:" followed by a bulleted list:

- Check your spelling.
- Remove filters from Advanced Search.
- Try more general words.
- Try different words that mean the same thing.

Opcodes you say?

# Executing

In a diagram



# Compiling: Diagram

```
<?php
function normalizeColorArray($array)
{
    foreach (array_keys($array) as $key)
    {
        $array[$key] = (float) $array[$key]/255;
    }

    return $array;
}

function rgbToCMYK($rgbArray)
{
    $cya = 1 - min(1, max((float) $rgbArray['r'], 0));
    $mag = 1 - min(1, max((float) $rgbArray['g'], 0));
    $yel = 1 - min(1, max((float) $rgbArray['b'], 0));

    $min = min($cya, $mag, $yel);
    if (1 - $min == 0)
    {
        return array('c' => 1, 'm' => 1,
    }

    return array('c' => ($cya - $min) / (1 - $min),
                'm' => ($mag - $min) / (1 - $min),
                'y' => ($yel - $min) / (1 - $min),
                'k' => $min );
}

function rgbToCMYK2($r, $g, $b)
{
    return e2Nacht::rgbToCMYK(array('r' => $r, 'g' => $g, 'b' => $b));
}
?>
```

Parse

Compile  
zend\_compile



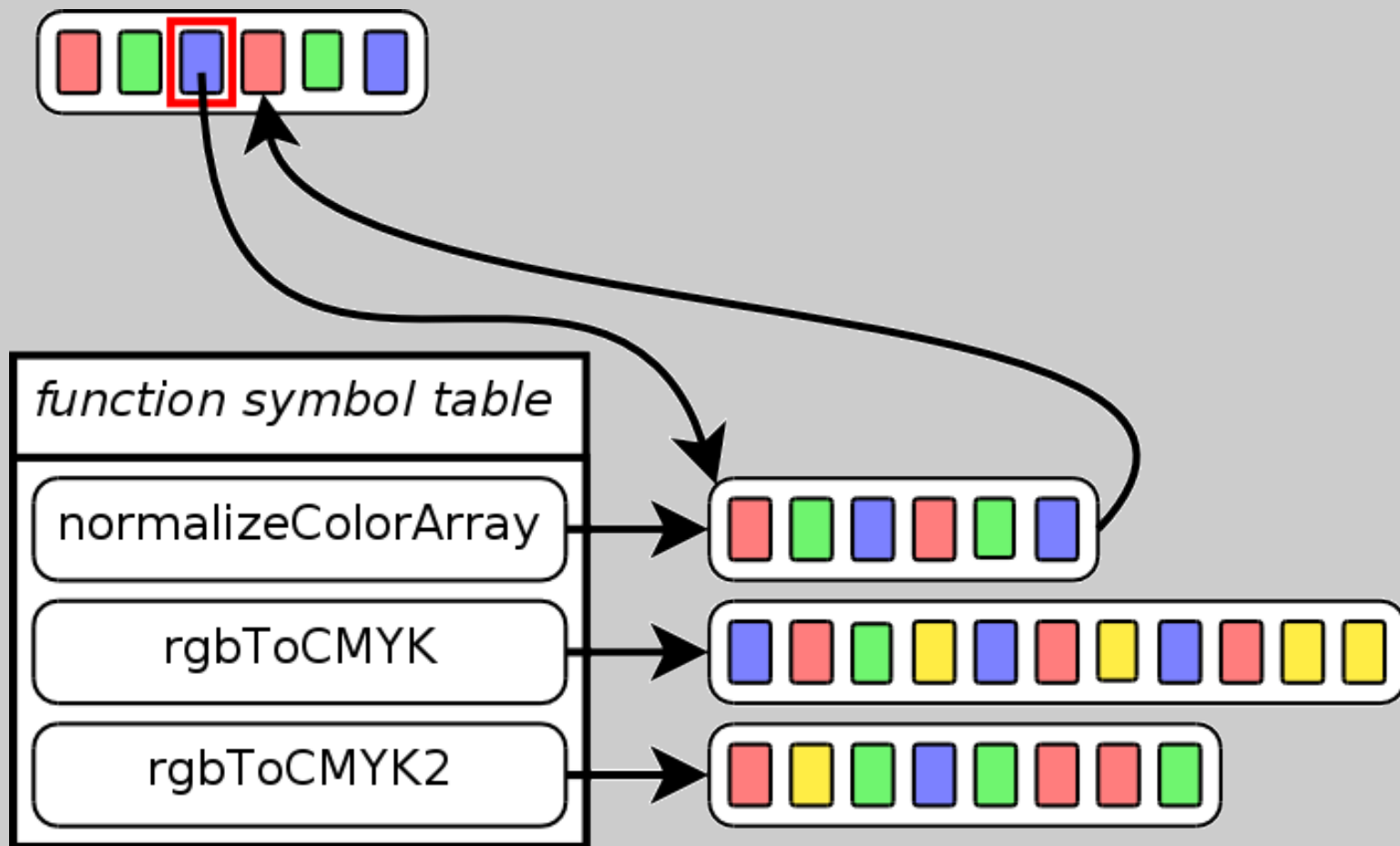
class symbol table

function symbol table

- normalizeColorArray
- rgbToCMYK
- rgbToCMYK2

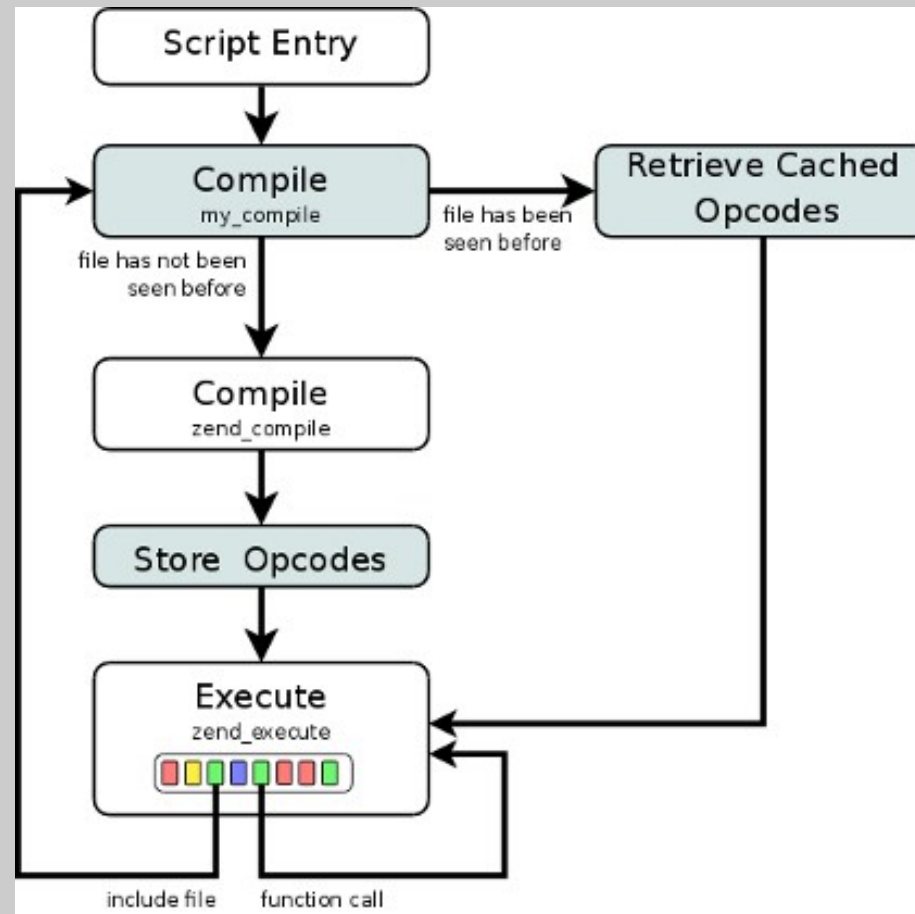


# Executing: Diagram



# Compiler Caches

## How it works



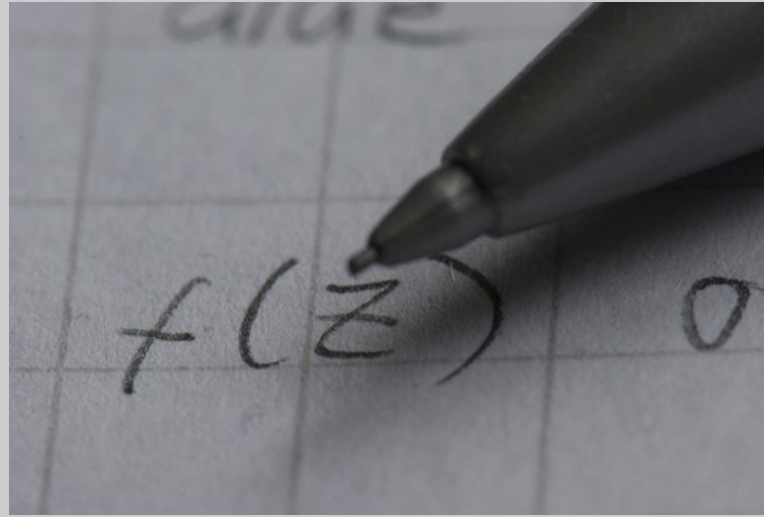
- In general, each source file is compiled once
- Compilation overhead becomes inconsequential
- Cache introduces its own overhead due to dynamic nature of includes

# Opcode overloading



- **Scream:** For making PHP ignore the @ operator
- **Xdebug:** For code coverage, and for making PHP ignore the @ operator
- **Operator:** Allows for operator overloading, but also demands that you hand over your first-born

# Replacing functions



- Replaces a function pointer in the symbol table with a new one
- You can also use it for methods
- Xdebug uses it to replace `var_dump` with `xdebug_var_dump`
- You can only replace internal functions (unless you're `runkit/classkit`)



- Wrappers: Wrap around file reading and writing operations with a "protocol" (f.e.: `http://compress.bzip2`)
- Filters: On the fly encrypting/decrypting filter (f.e.: `mcrypt_filter`)



```
<?php  
$marmiteIsFor = 'trashcan';  
echo 'Hello World.';  
?>
```



```
<?php  
£marmiteIsFor = 'biscuits';  
announce 'Good morrow, fellow subjects of the Crown.';  
?>
```

- Sadly, this you can't do.
- Or can you?
- Facebook's xhp works as a pre-processor...

# Why extensions?

## Wrapping libraries



- General use libraries: re2
- Specific use libraries: cybermut, pam
- Libraries that you have written yourself, or your company
- Missing database support, or new webscale things like mongodb

# Why extensions?

They are faster than raw PHP code



- It's just too slow in normal PHP: ssh2, openssl
- Specific uses: QuickHash, StumbleCache

# Why extensions?

## Performance bottlenecks



- Your code is really slow, and need something quicker: twig

# Why extensions?

Impossible in just PHP



- Not everything is possible in pure PHP: apc, com, xdebug, xhprof

# Questions

?

# Resources

- These Slides: <http://derickrethans.nl/talks.php>
- VLD: <http://www.derickrethans.nl/vld.php>
- Xdebug: <http://xdebug.org>
- Included: <http://t3.dotgnu.info/blog/tags/included/>
- Extending and Embedding PHP ('Sara's Book'):  
<http://www.amazon.com/Extending-Embedding-PHP-Sara-Golemon/dp/067232704X>
- Tutorial on Zend's devzone:  
<http://devzone.zend.com/article/1021>

Thanks!

Derick Rethans - [derick@derickrethans.nl](mailto:derick@derickrethans.nl) - twitter:  
@derickr

<http://derickrethans.nl/talks.html>

<http://joind.in/3586>