

Welcome!

# PHP Extensions - What and Why

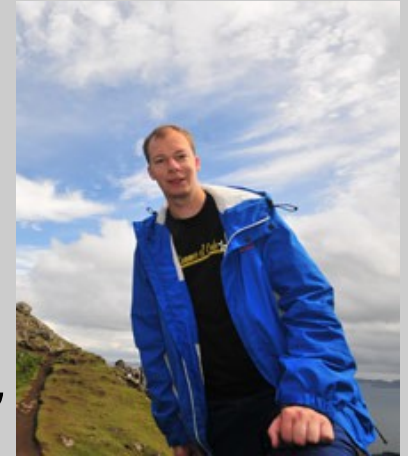
PHP London - London, UK - Sep 1st, 2011

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

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

## Derick Rethans

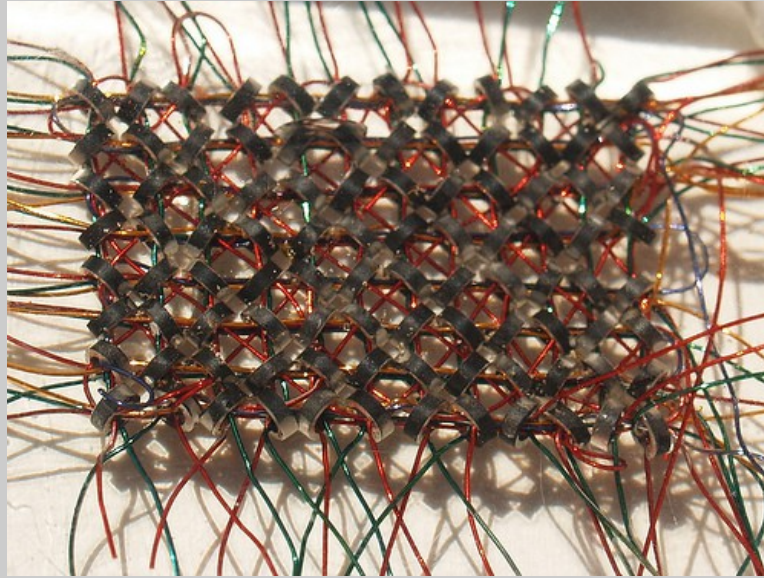
- Dutchman living in London
- PHP development
- Author of the `mcrypt`, `input_filter`, `dbus`, and `date/time` extensions
- Author of `Xdebug`
- Contributor to the Apache Zeta Components Incubator project (formerly eZ Components)
- Freelancer doing PHP (internals) development



# 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
- that's it really



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



## Zend Extensions

- Provide a few more (very specialized hooks)
- Are not necessarily made by Zend (the company)
- Xdebug is one of them!

# 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 courieraauth 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 framegrab 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 mdhtools memcache memcached memsession memtrack ming

# 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

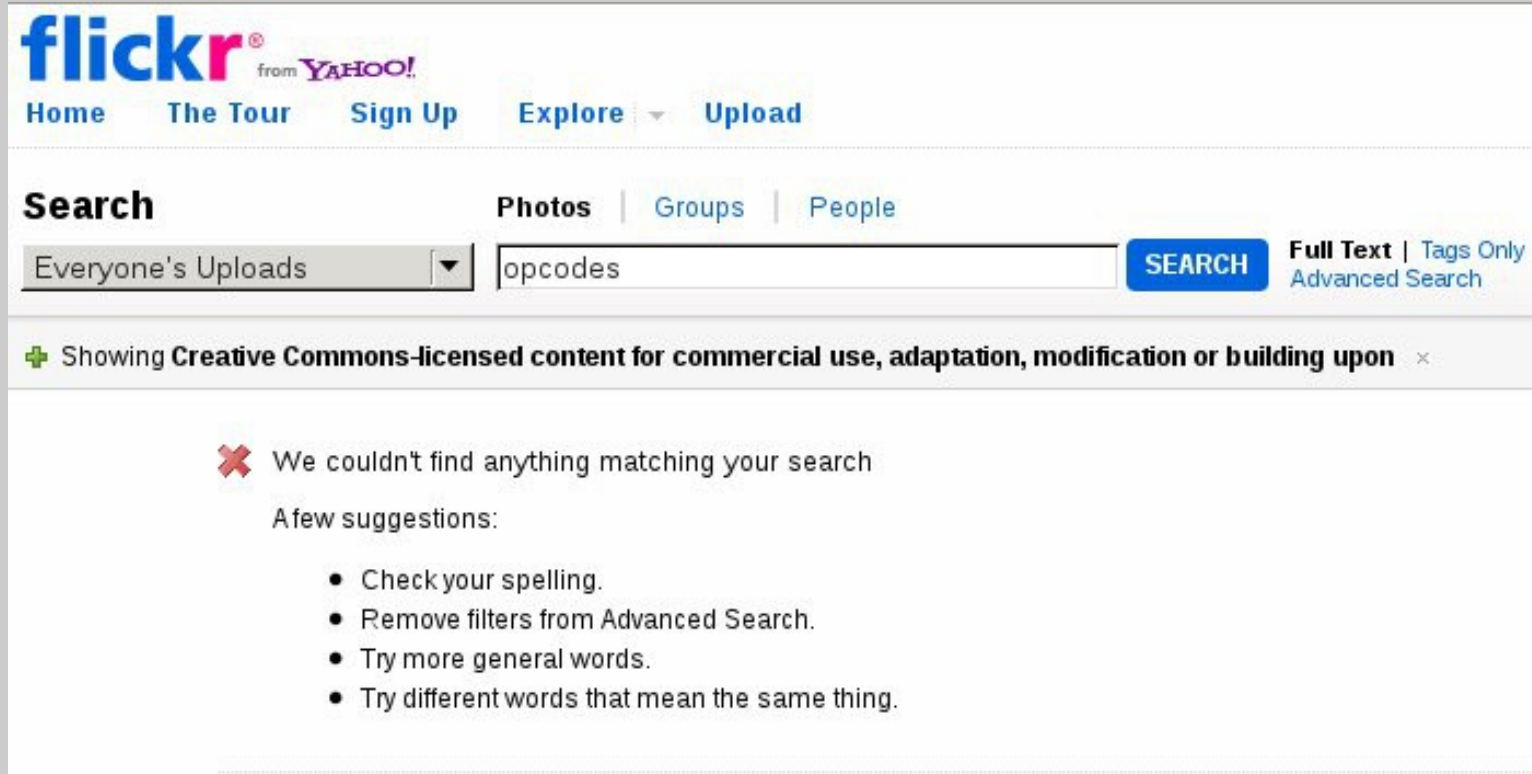
- statement calls
- all sorts of internal handlers: activate/deactivate functions, `op_array` handler, `op_array` constructor and destructor

# 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

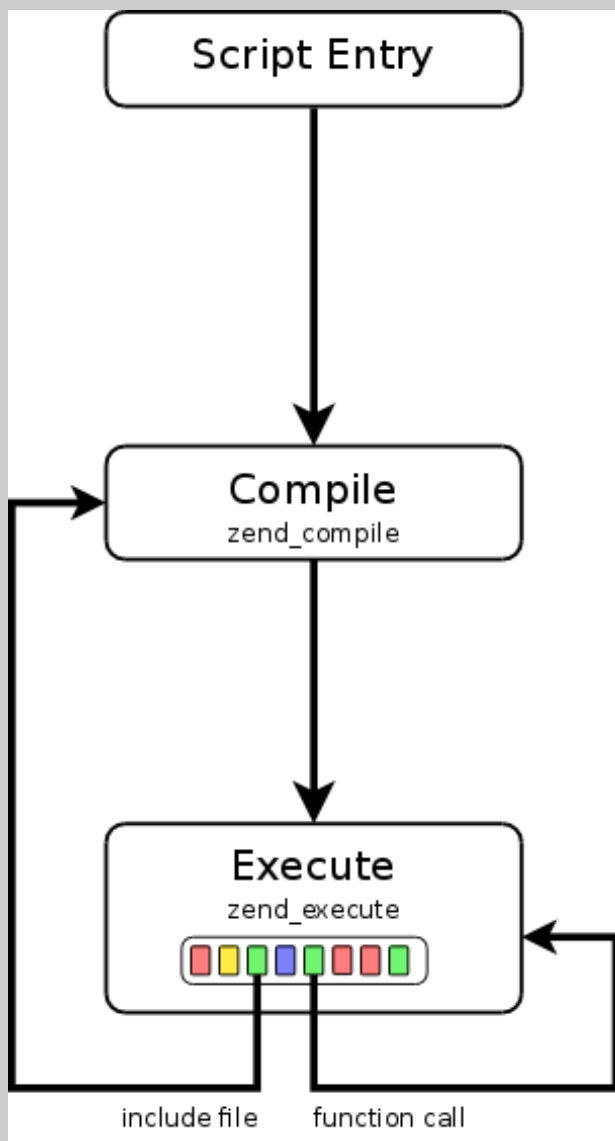


The screenshot shows the Flickr website's search interface. At the top left is the Flickr logo with the text "from YAHOO!". Navigation links include "Home", "The Tour", "Sign Up", "Explore", and "Upload". The search bar is set to "Everyone's Uploads" and contains the text "opcodes". A blue "SEARCH" button is to the right of the input field. Further right are 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 main content area displays 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.", and "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



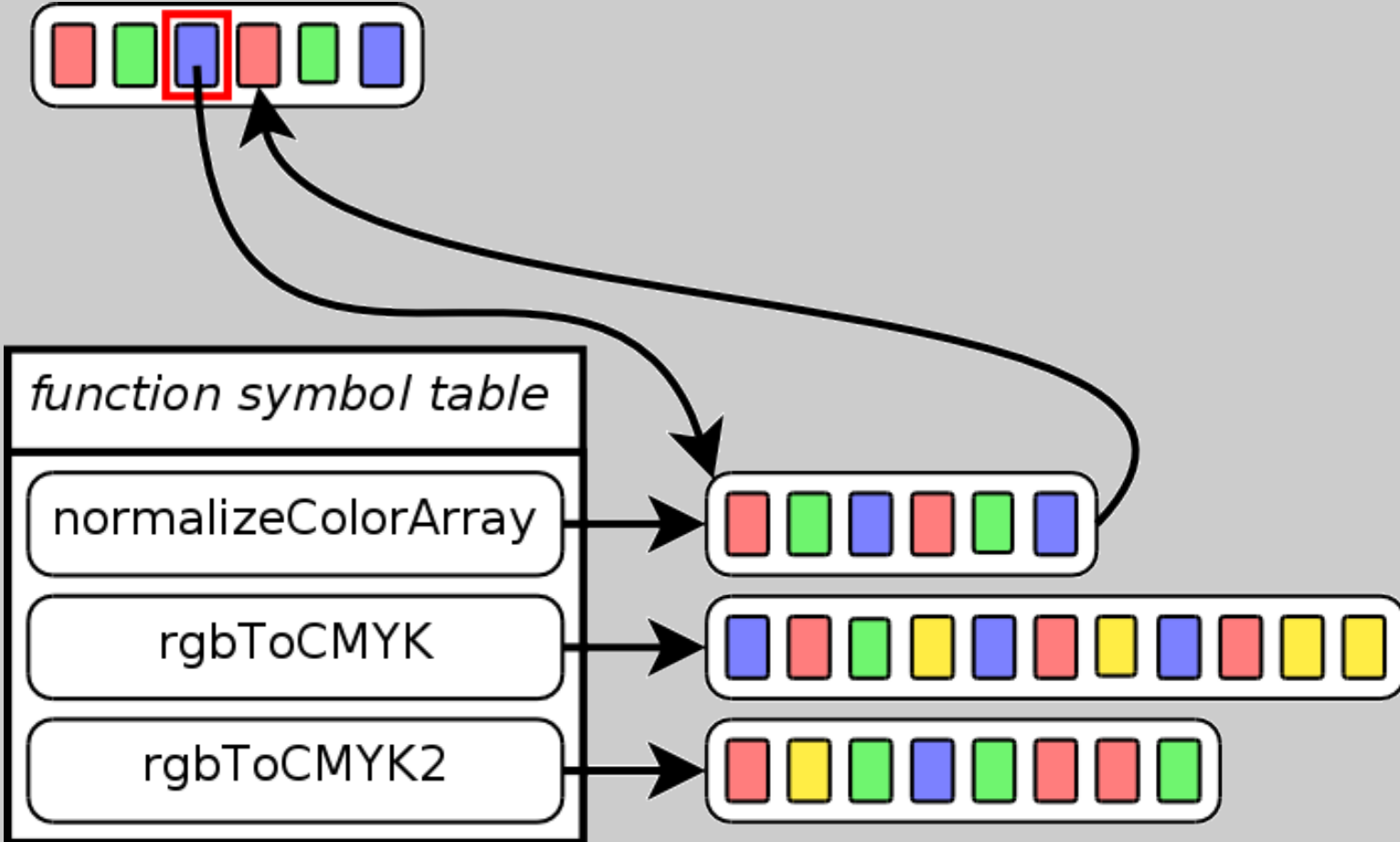
class symbol table

function symbol table

- normalizeColorArray
- rgbToCMYK
- rgbToCMYK2

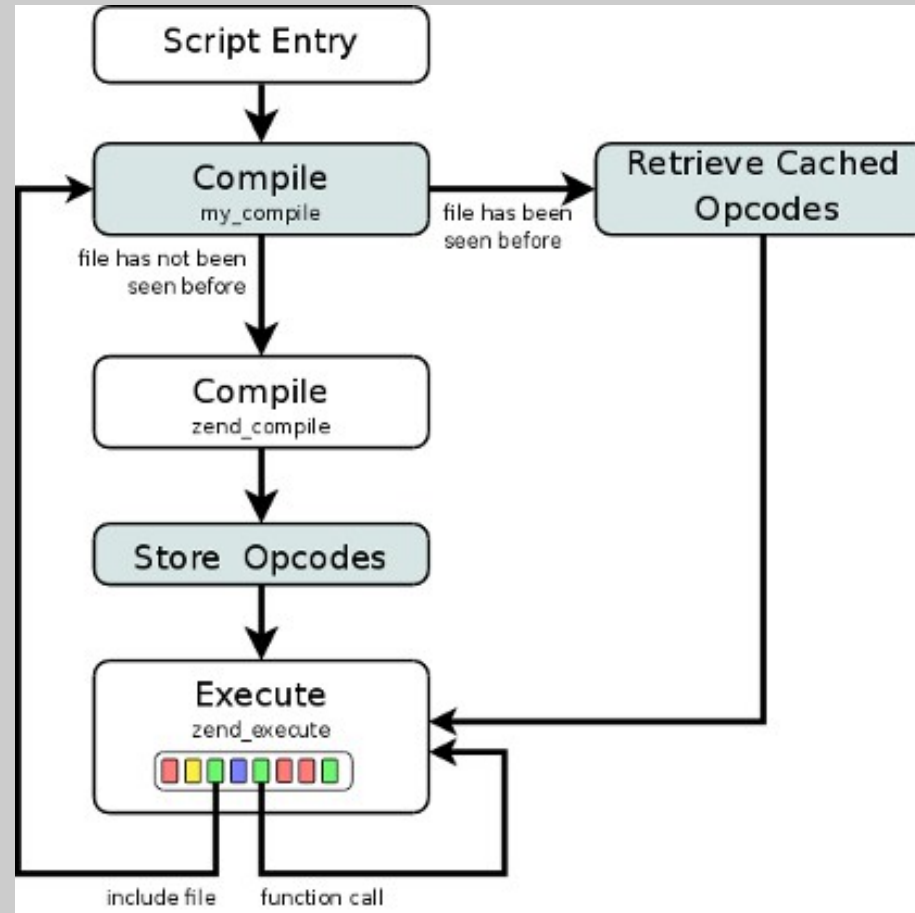


# Executing: Diagram



# 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



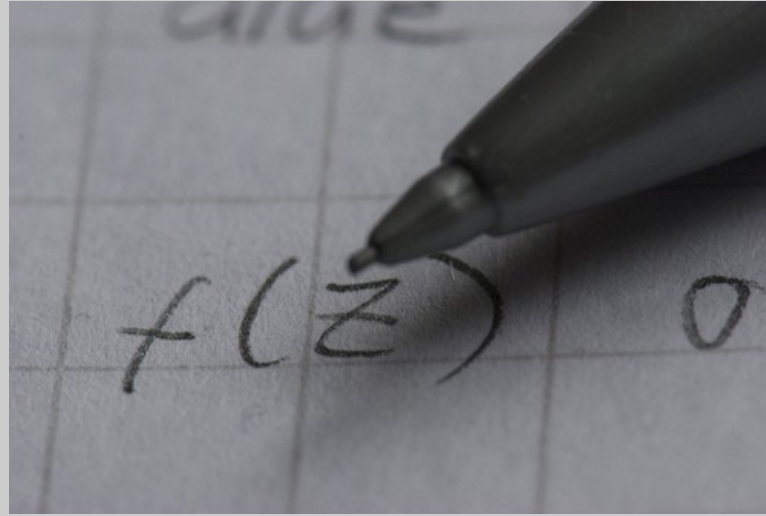
Good:

- **Scream:** For making PHP ignore the @ operator
- **Xdebug:** For code coverage, and for making PHP ignore the @ operator

Bad:

- **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 and you'd like some pain)



- 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.
- However...

# Why extensions?

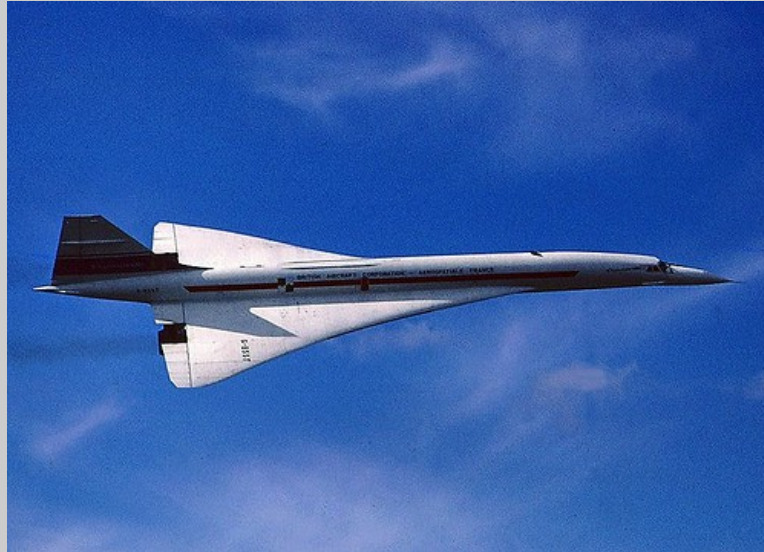
## Wrapping libraries



- 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

# 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

?

Thanks!

Derick Rethans - [derick@derickrethans.nl](mailto:derick@derickrethans.nl) - twitter:  
@derickr  
<http://derickrethans.nl/talks.html>