Advanced PHP

Peter Green and Joel Boonstra
Agathon Group
Originally presented at Gospelcon 2006

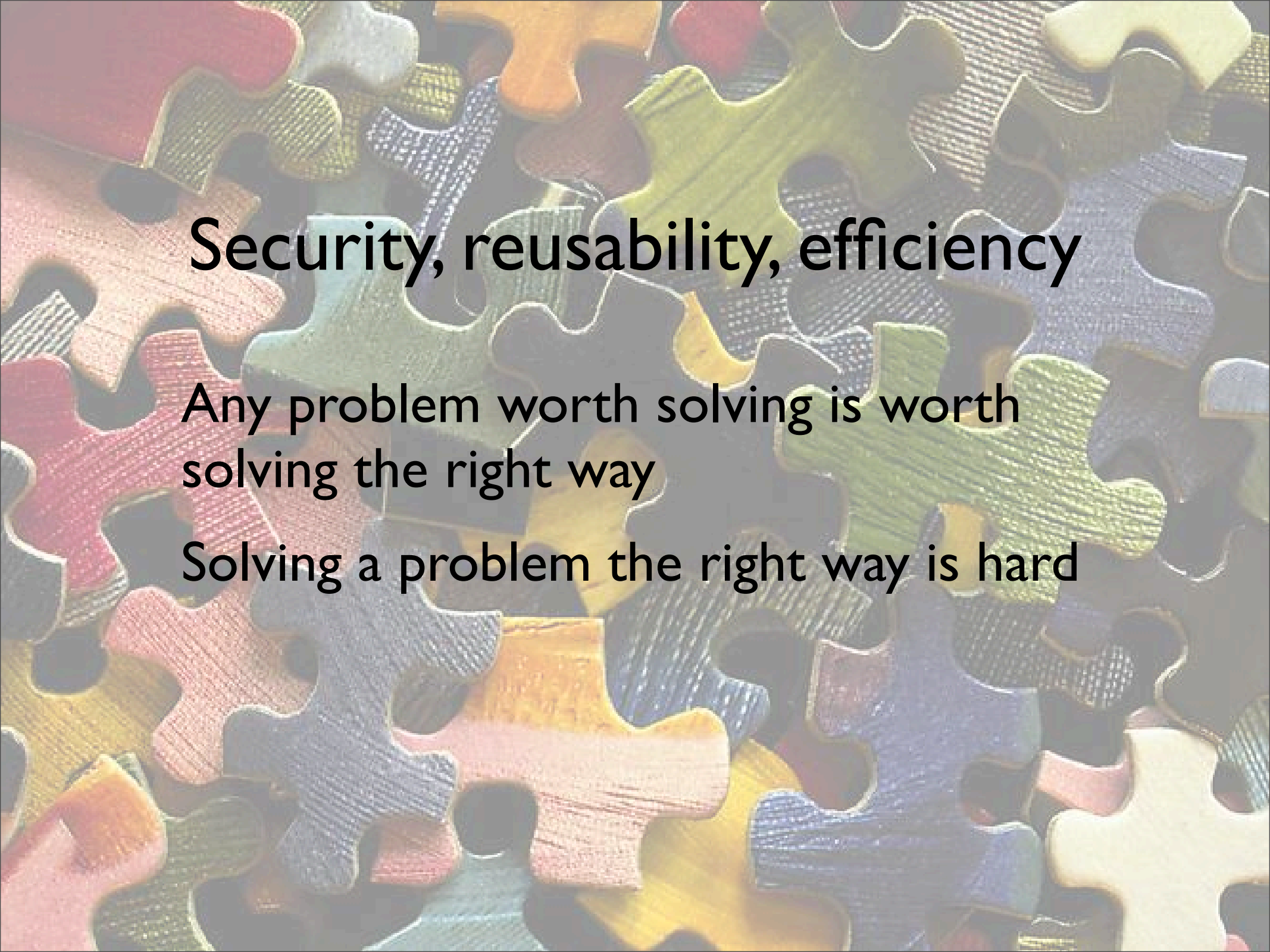
http://www.agathongroup.com/talks/gospelcon06/advancedphp/

Introduction


Security, reusability, efficiency

Security, reusability, efficiency

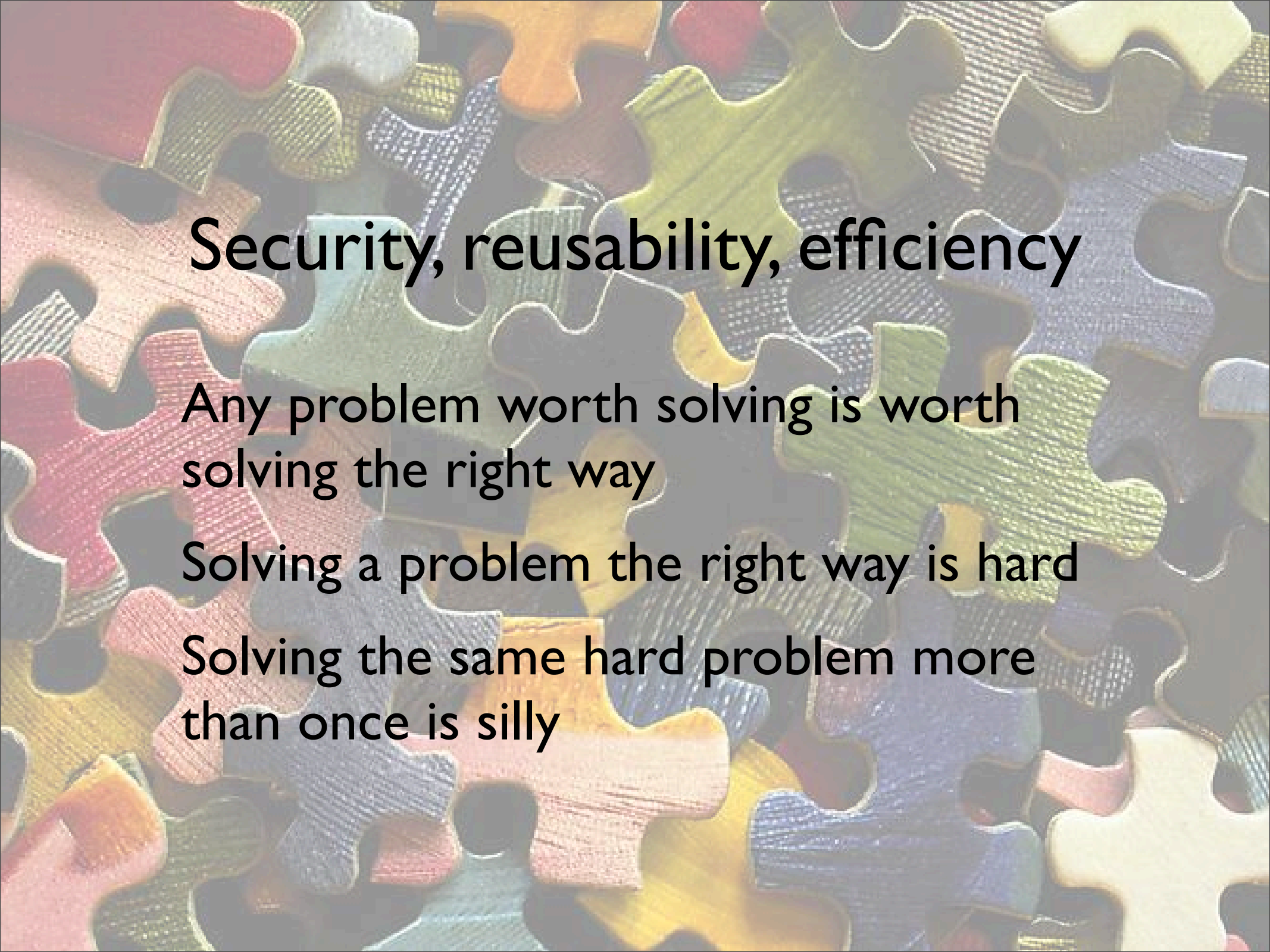
Any problem worth solving is worth solving the right way



Security, reusability, efficiency

Any problem worth solving is worth solving the right way

Solving a problem the right way is hard

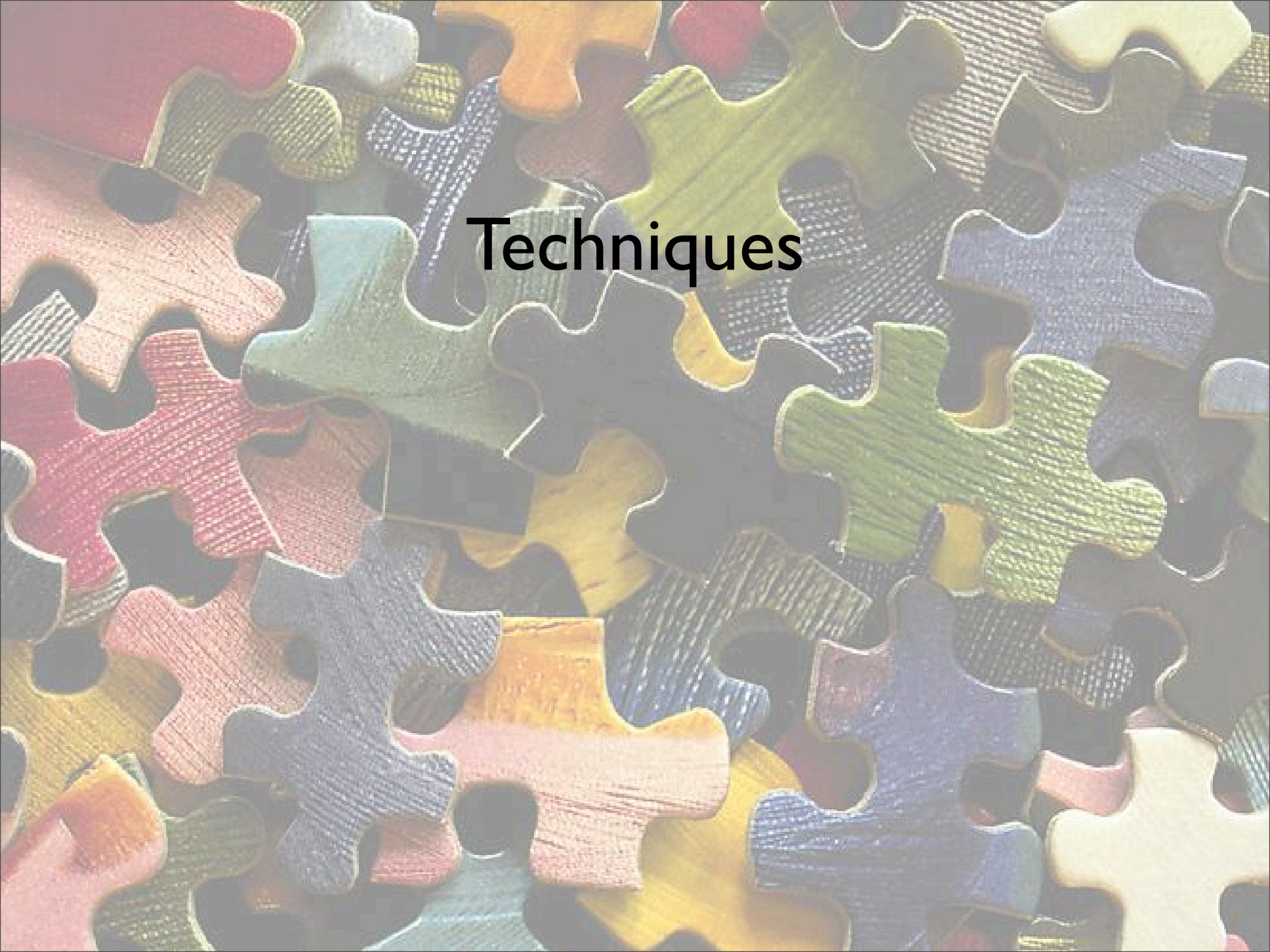


Security, reusability, efficiency

Any problem worth solving is worth solving the right way

Solving a problem the right way is hard

Solving the same hard problem more than once is silly


Techniques

Techniques

Directory organization

Techniques

Directory organization

Using pre-written libraries

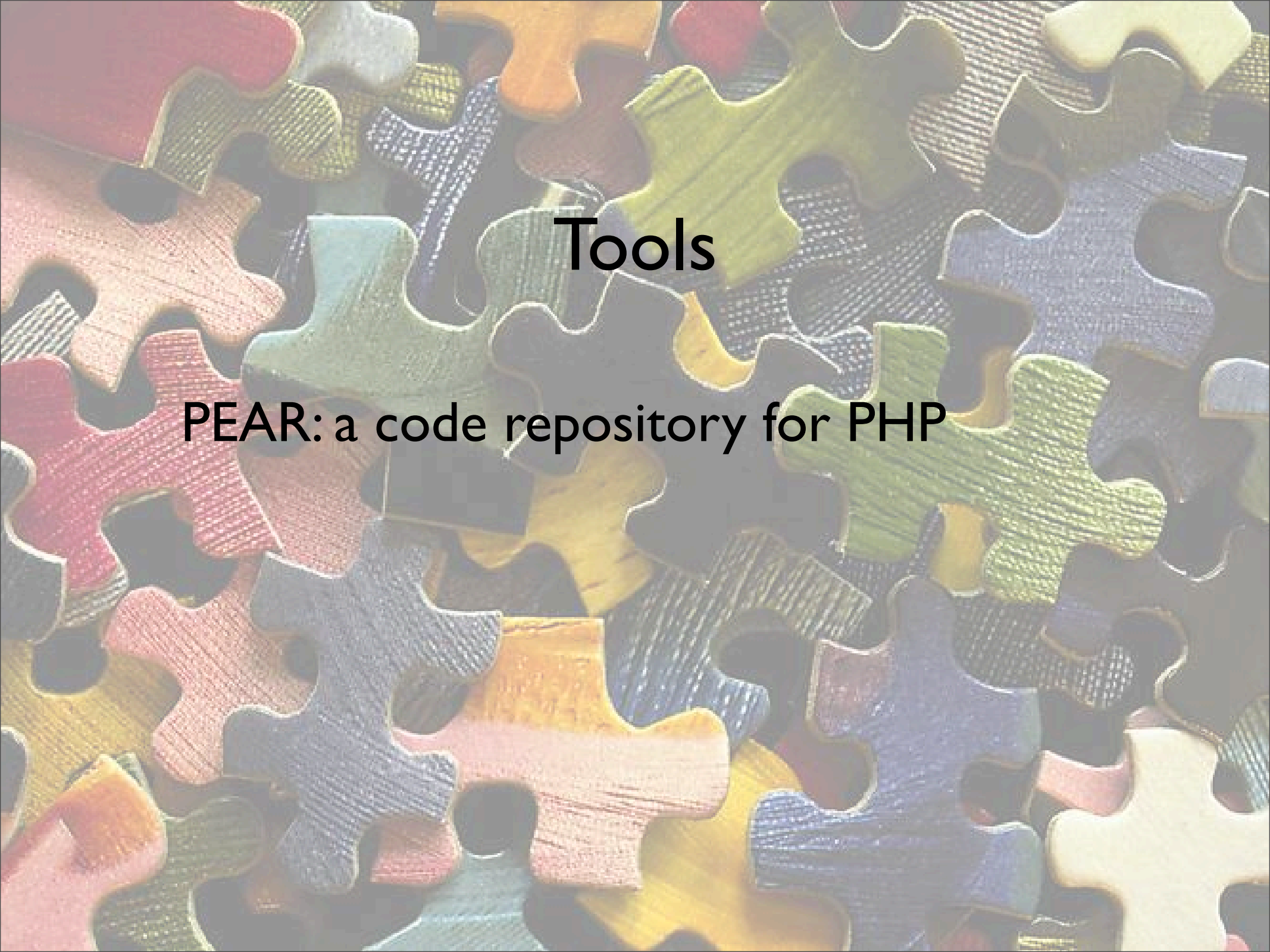
Techniques

Directory organization

Using pre-written libraries

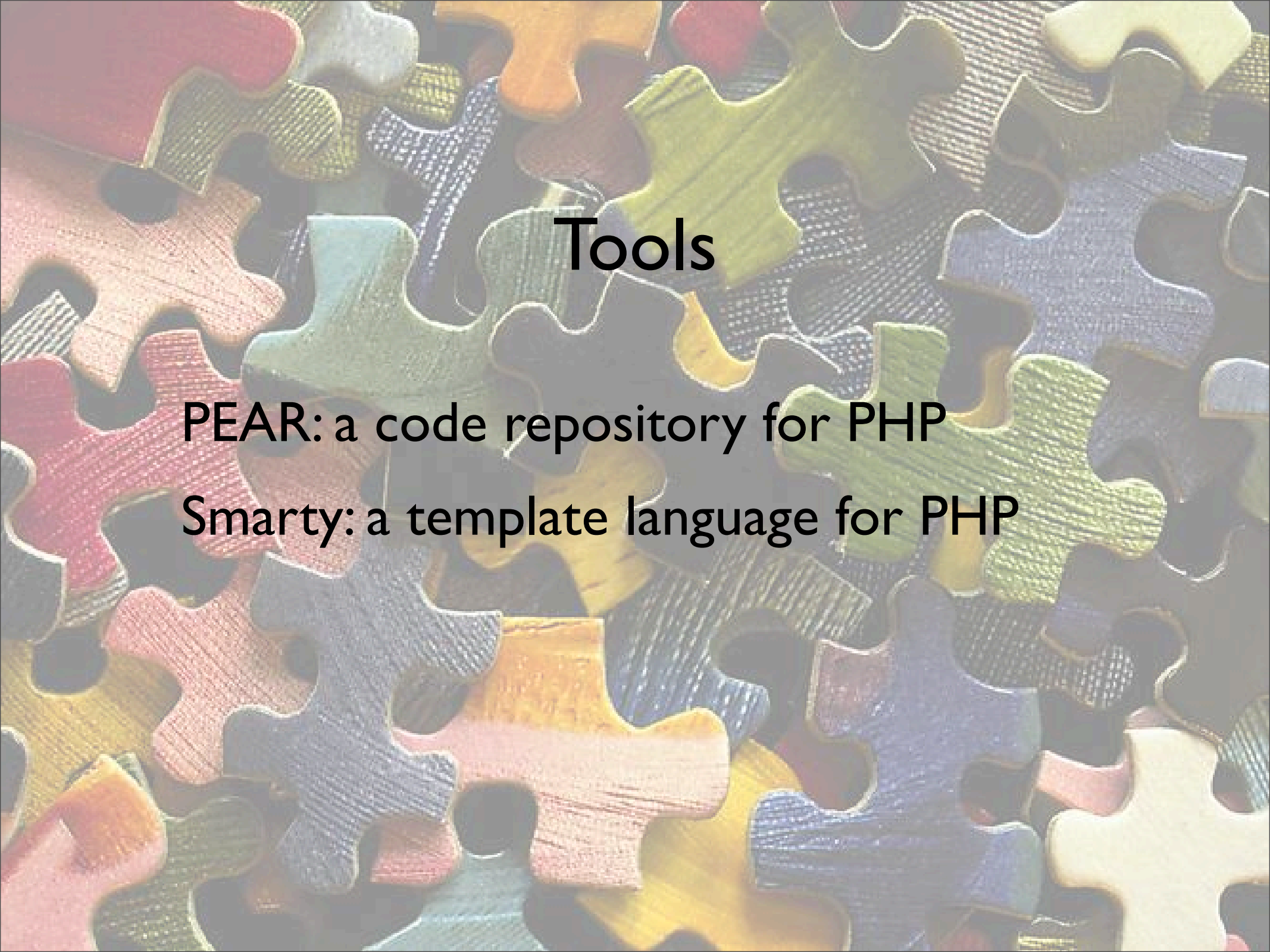
Writing your own libraries


Tools



Tools

PEAR: a code repository for PHP



Tools

PEAR: a code repository for PHP

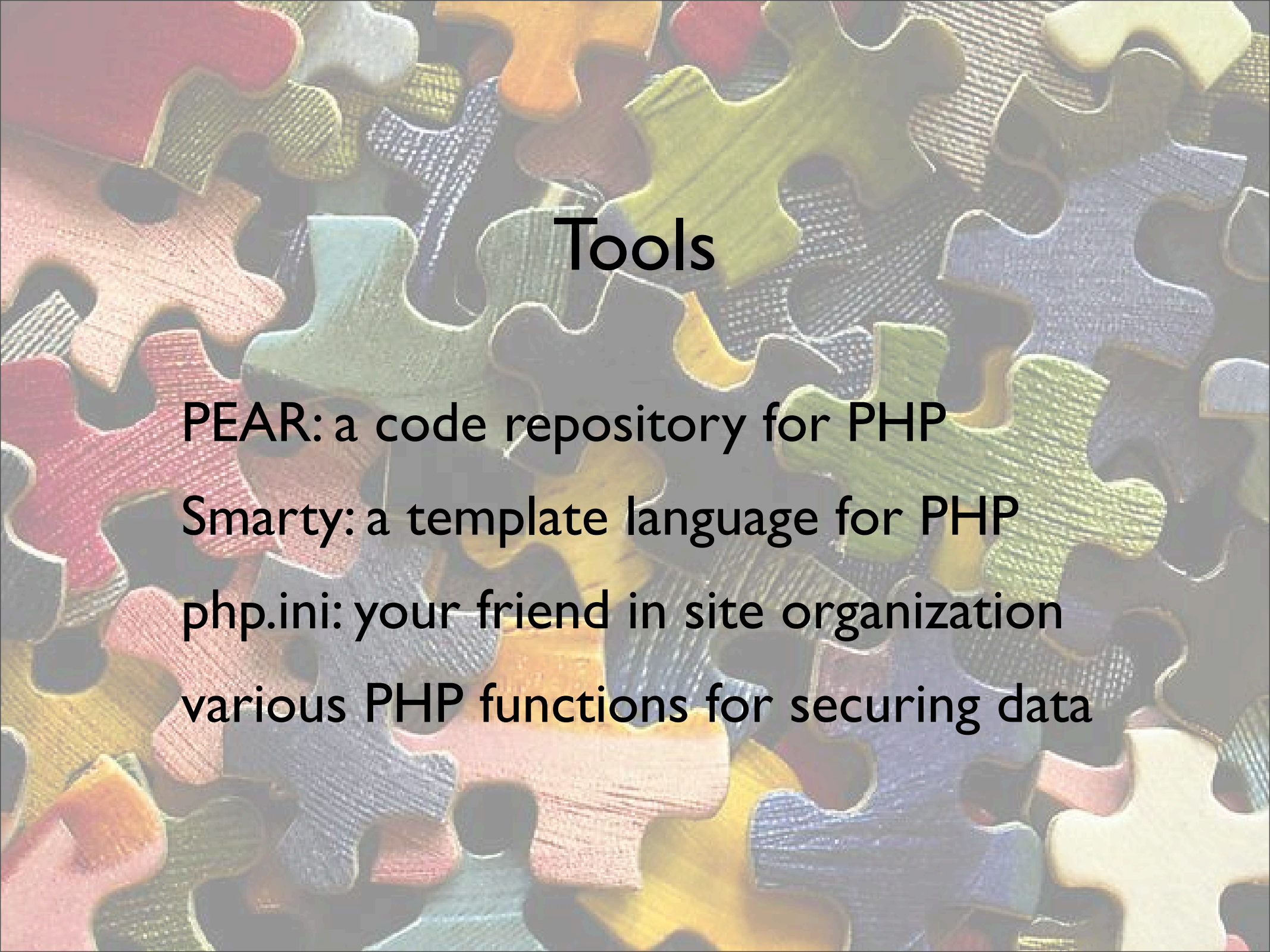
Smarty: a template language for PHP

Tools

PEAR: a code repository for PHP

Smarty: a template language for PHP

php.ini: your friend in site organization



Tools

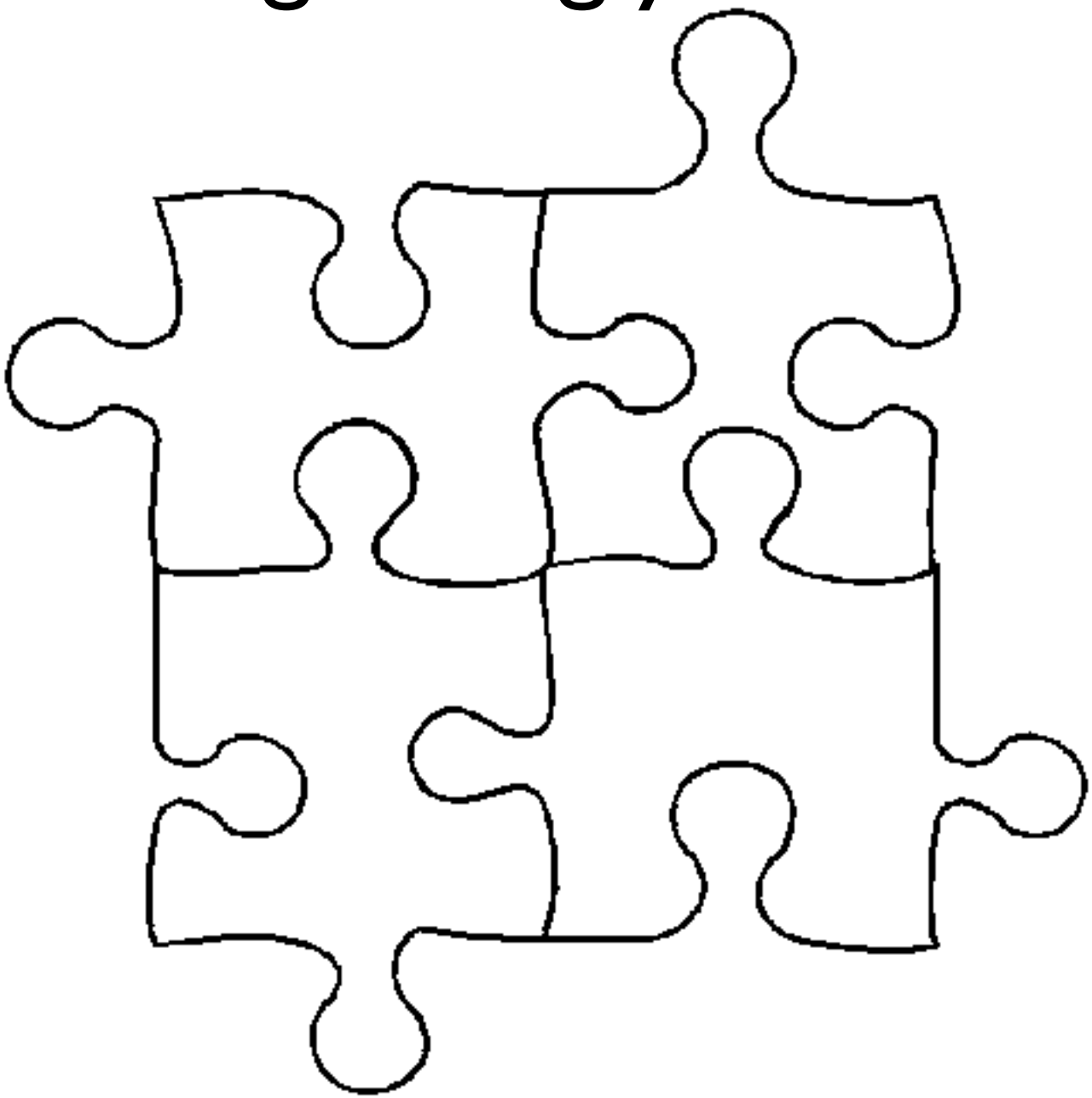
PEAR: a code repository for PHP

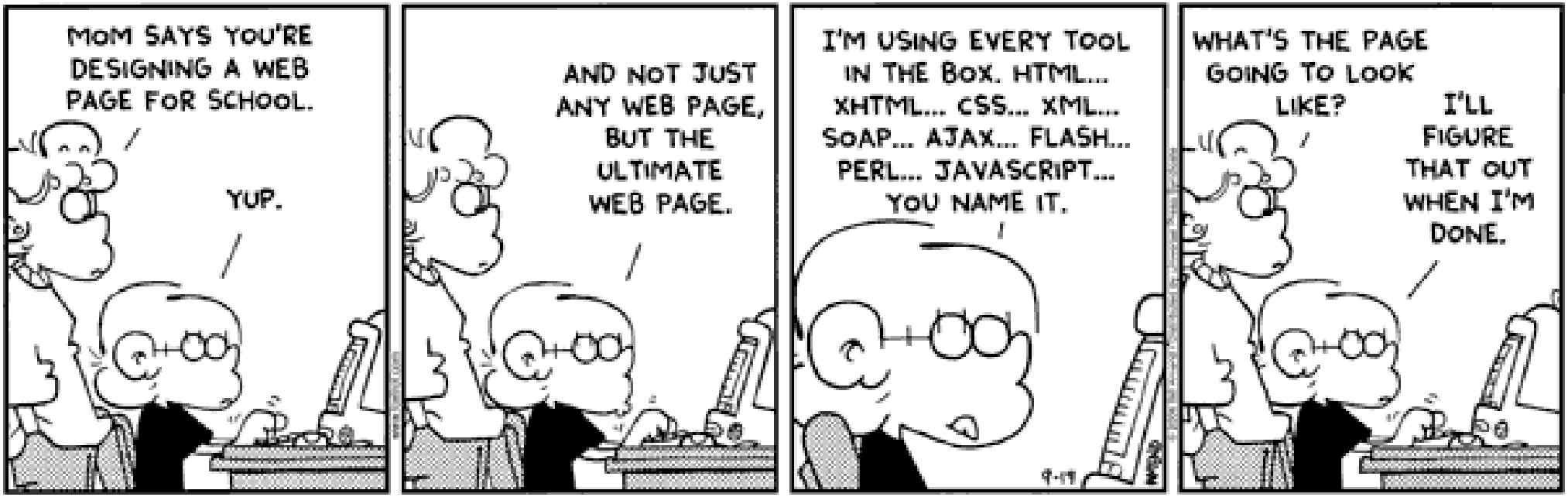
Smarty: a template language for PHP

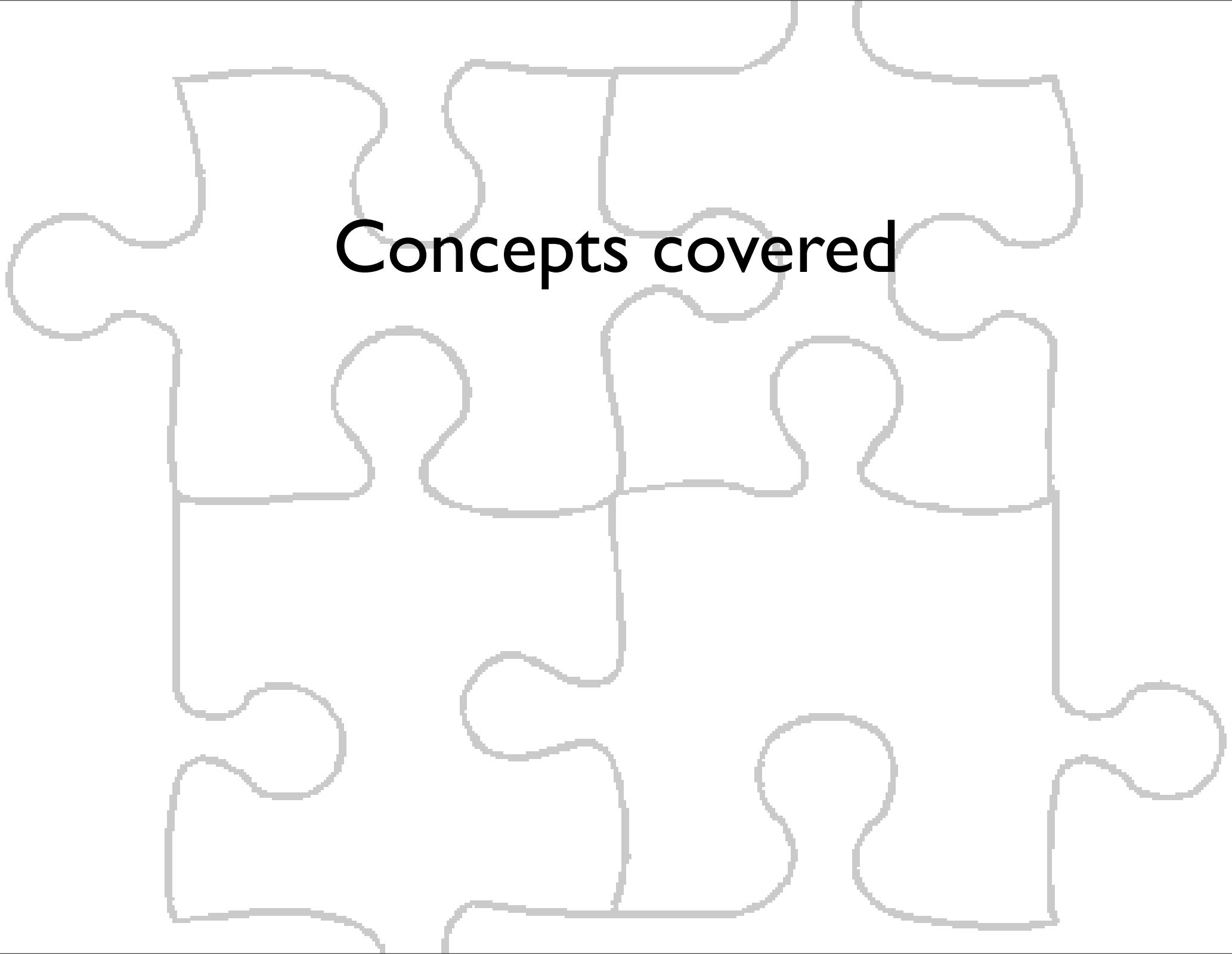
php.ini: your friend in site organization

various PHP functions for securing data

Step 0: Organizing your directory




MOM SAYS YOU'RE DESIGNING A WEB PAGE FOR SCHOOL.
YUP.
AND NOT JUST ANY WEB PAGE, BUT THE ULTIMATE WEB PAGE.
I'M USING EVERY TOOL IN THE BOX. HTML... XHTML... CSS... XML... SOAP... AJAX... FLASH... PERL... JAVASCRIPT... YOU NAME IT.
WHAT'S THE PAGE GOING TO LOOK LIKE?
I'LL FIGURE THAT OUT WHEN I'M DONE.


Concepts covered

Concepts covered

"Slow down, you move too fast!"

Concepts covered

"Slow down, you move too fast!"

Separation of logic from presentation

Concepts covered

"Slow down, you move too fast!"

Separation of logic from presentation

Prevent access to sensitive files

Concepts covered

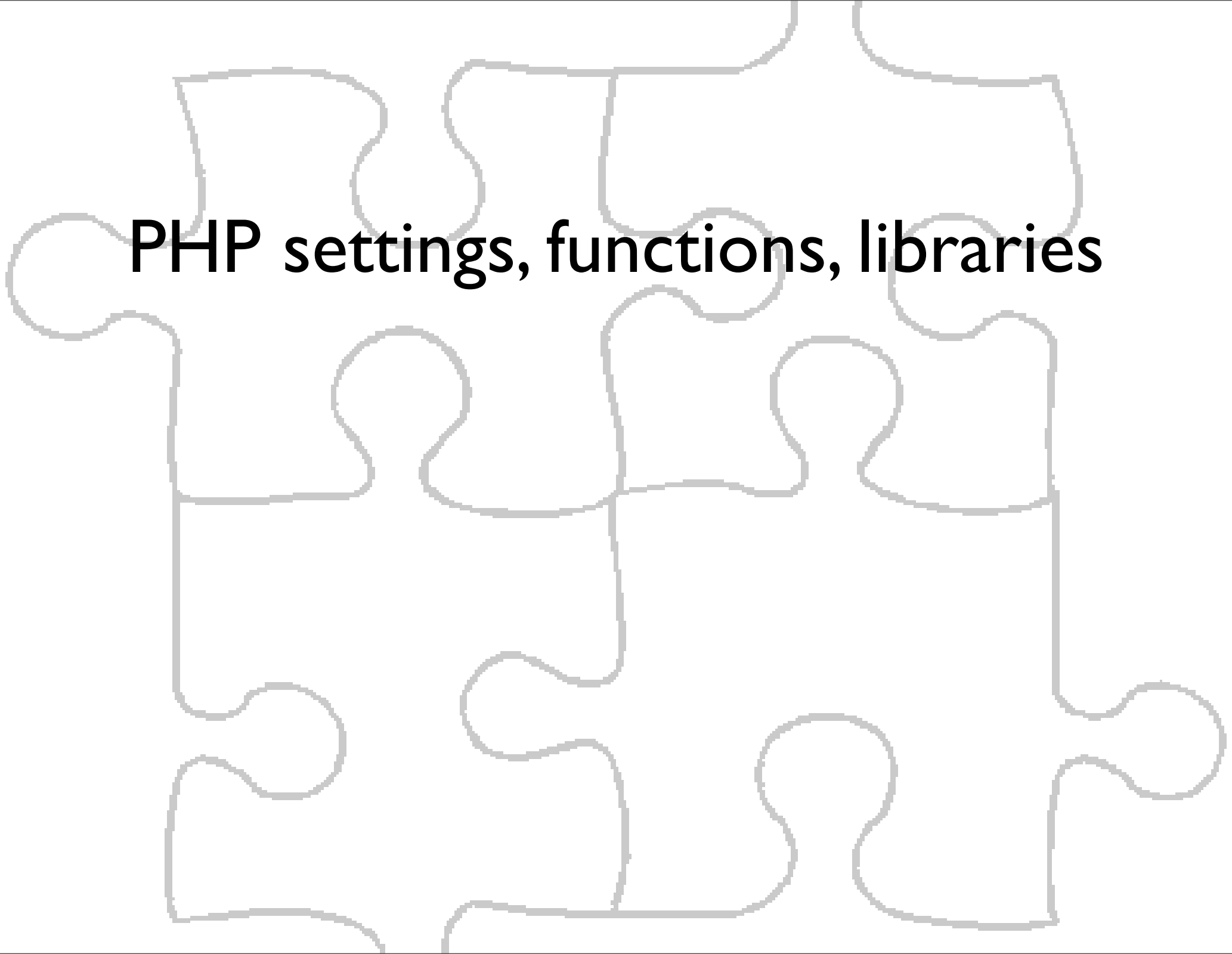
"Slow down, you move too fast!"

Separation of logic from presentation

Prevent access to sensitive files

Self-contained, portable development

PHP settings, functions, libraries



PHP settings, functions, libraries

php.ini: `include_path`: normalize includes

PHP settings, functions, libraries

php.ini: `include_path`: normalize includes

php.ini: `auto_prepend_file`: apply a common "settings" file to all PHP files

PHP settings, functions, libraries

php.ini: `include_path`: normalize includes

php.ini: `auto_prepend_file`: apply a common "settings" file to all PHP files

function: `ini_set()`: access php.ini values

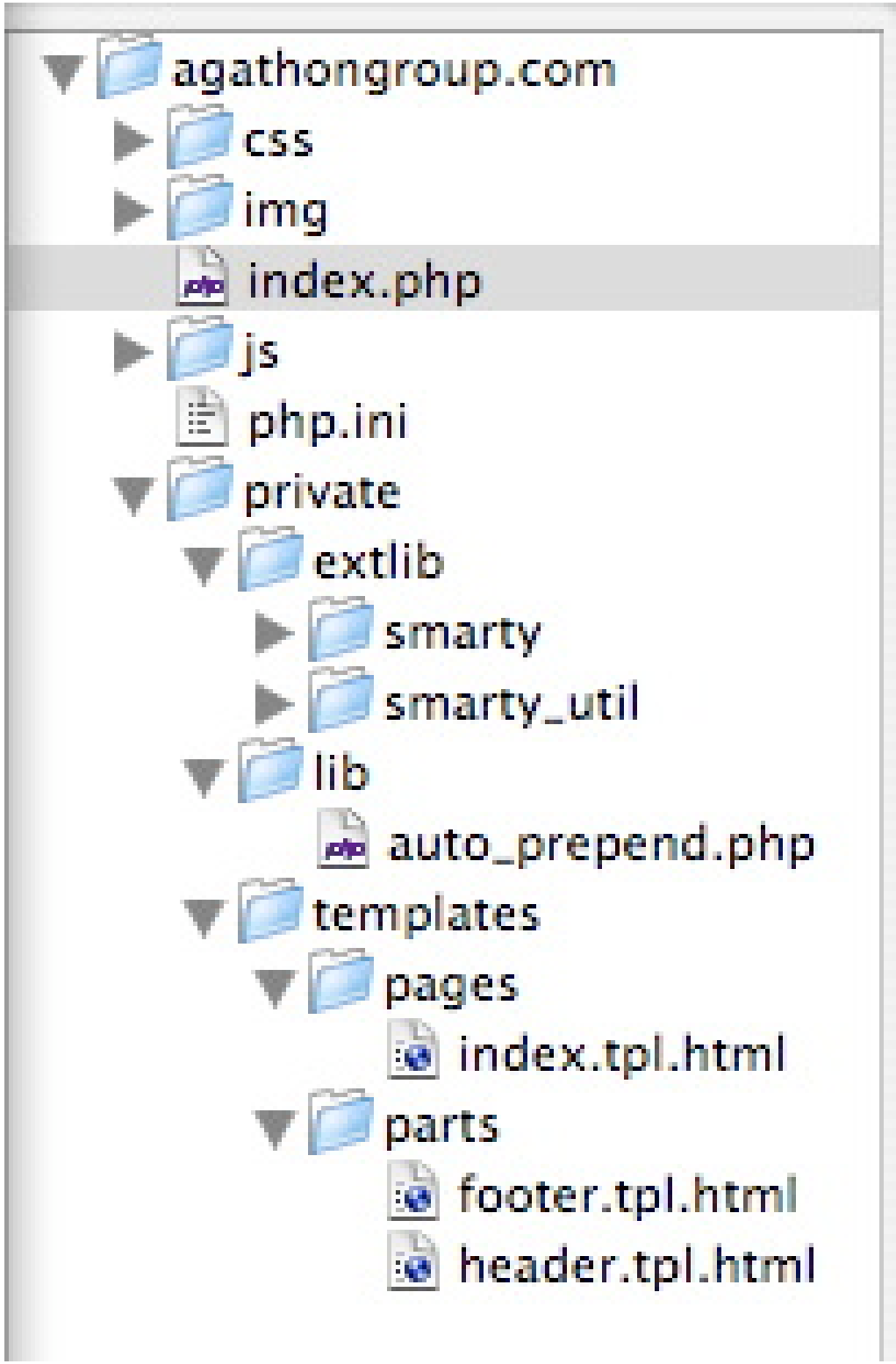
PHP settings, functions, libraries

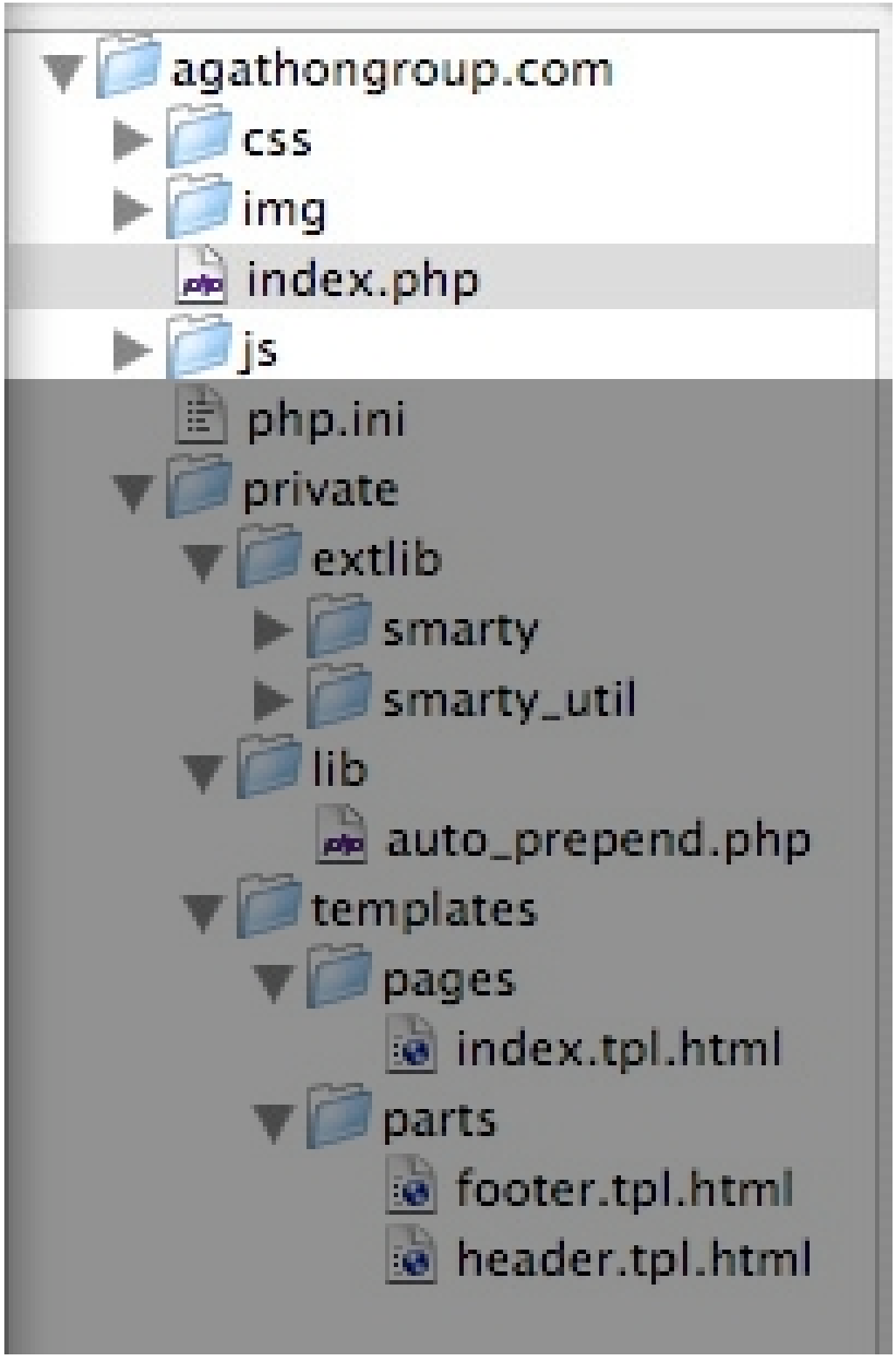
php.ini: `include_path`: normalize includes

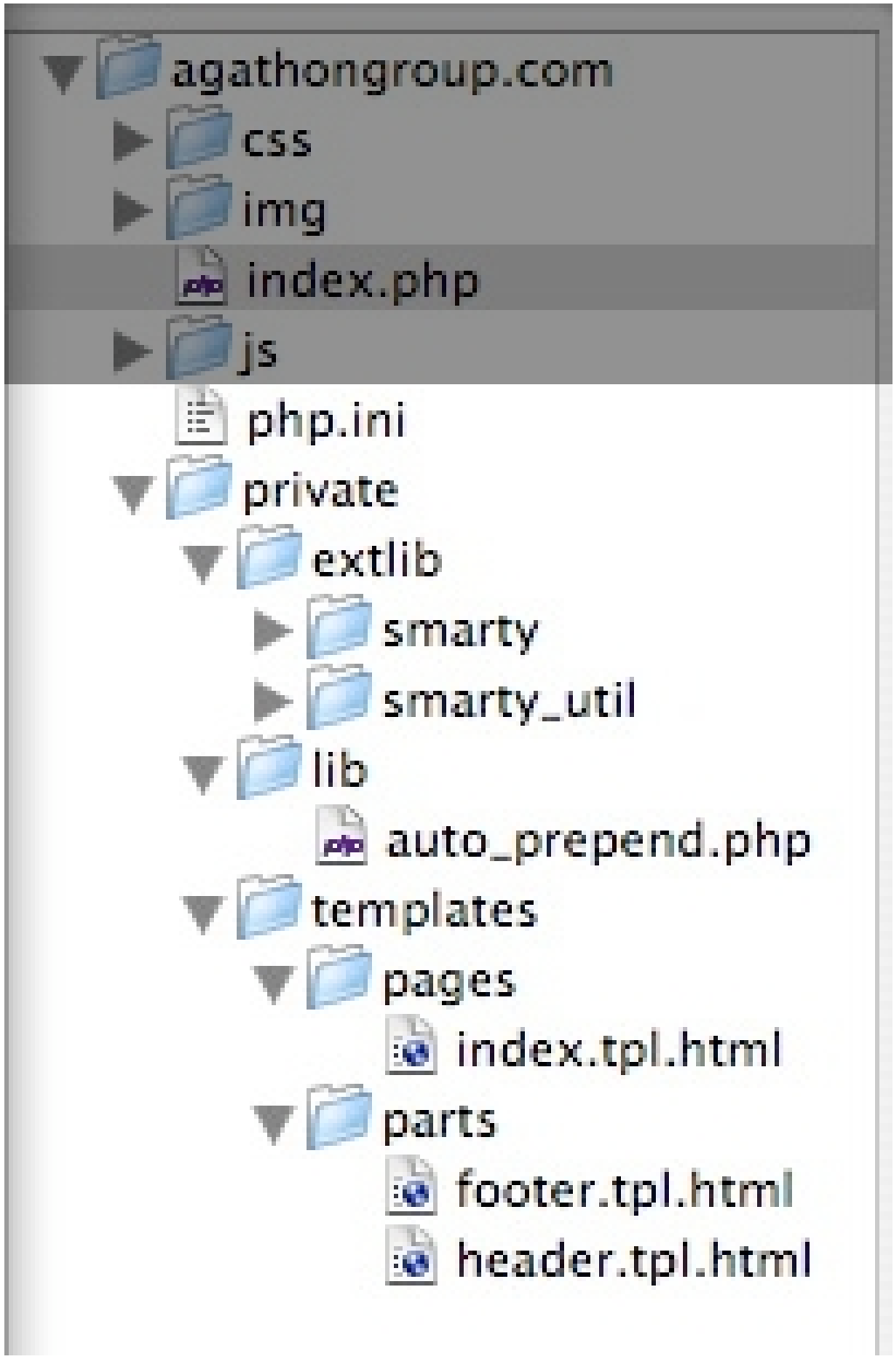
php.ini: `auto_prepend_file`: apply a common "settings" file to all PHP files

function: `ini_set()`: access php.ini values

library: Smarty: templating


agathongroup.com
css
img
index.php
js
php.ini
private
extlib
smarty
smarty_util
lib
auto_prepend.php
templates
pages
index.tpl.html
parts
footer.tpl.html
header.tpl.html


agathongroup.com
css
img
index.php
js
php.ini
private
extlib
smarty
smarty_util
lib
auto_prepend.php
templates
pages
index.tpl.html
parts
footer.tpl.html
header.tpl.html


agathongroup.com
css
img
index.php
js
php.ini
private
extlib
smarty
smarty_util
lib
auto_prepend.php
templates
pages
index.tpl.html
parts
footer.tpl.html
header.tpl.html

php.ini

```
;; automatically add a file to all .php files
auto_prepend_file = private/lib/auto_prepend.php
```

private/lib/auto_prepend.php

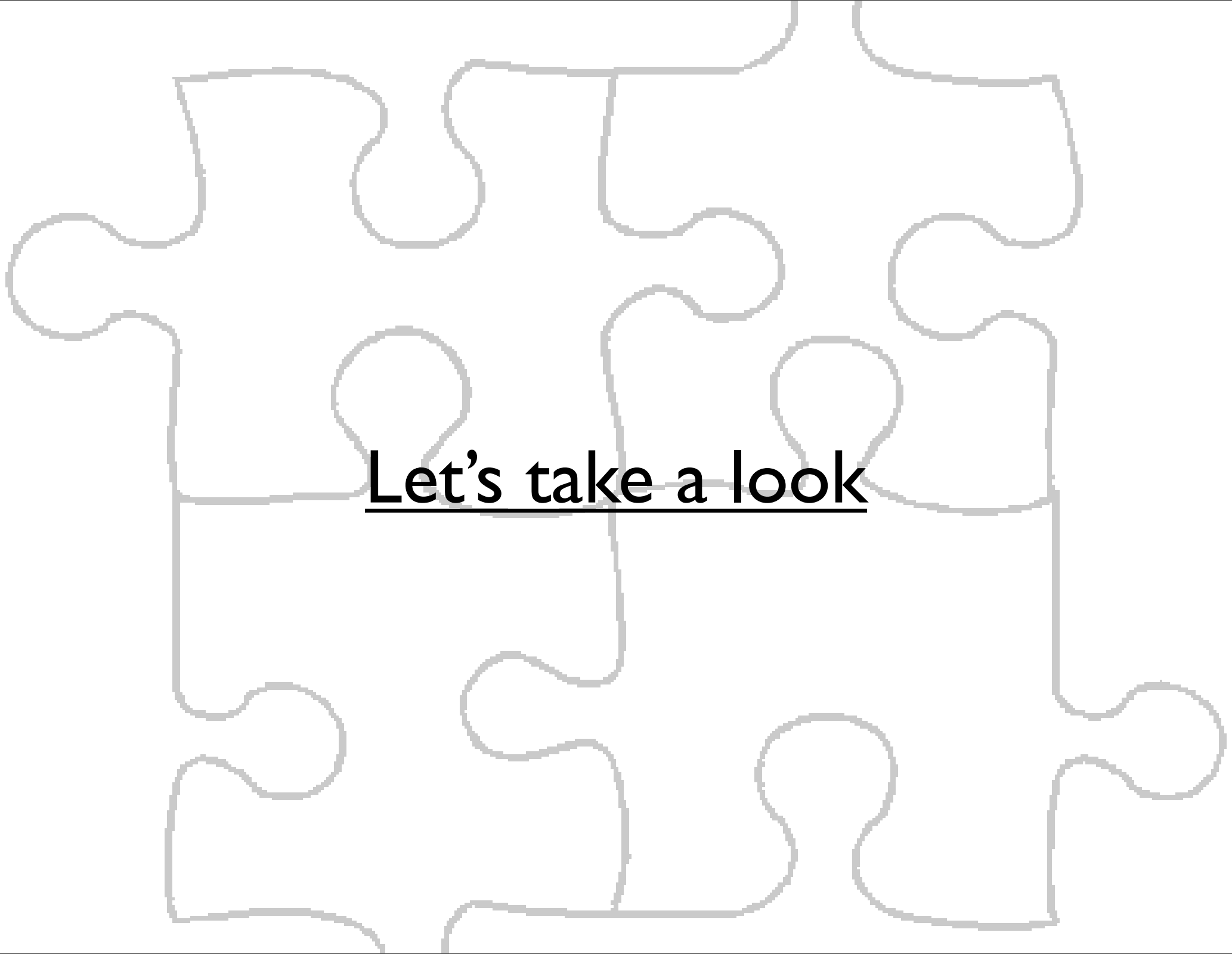
```php
<?php
/*==
 | file    : private/lib/auto_prepend.php
 | author  : Agathon Group
 | contact : support@agathongroup.com
 |
 | purpose : set common things that all of our pages need
 |
 | $Id$
 +==*/
// set a nice include_path for all pages
$base_dir = $_SERVER['DOCUMENT_ROOT'];
ini_set('include_path', "$base_dir/private");

// use Smarty
require_once('extlib/smarty/Smarty.class.php');
$smarty = new Smarty();
$smarty->template_dir = "$base_dir/private/templates";
$smarty->compile_dir  = "$base_dir/private/extlib/smarty_util/templates_c";
$smarty->cache_dir    = "$base_dir/private/extlib/smarty_util/cache";
?>
```

index.php

```php
<?php
/*==
 | file     : index.php
 | author   : Agathon Group
 | contact  : support@agathongroup.com
 |
 | purpose : Our main index page
 |
 | $Id$
 +==*/

// $smarty is available to us automatically
$smarty->assign(array(
    'title' => 'Agathongroup.com',
    'description' => 'Agathon Group provides hosting, consulting, and
programming.',
));
$smarty->display('pages/index.tpl.html');
?>
```

private/templates/pages/index.tpl.html

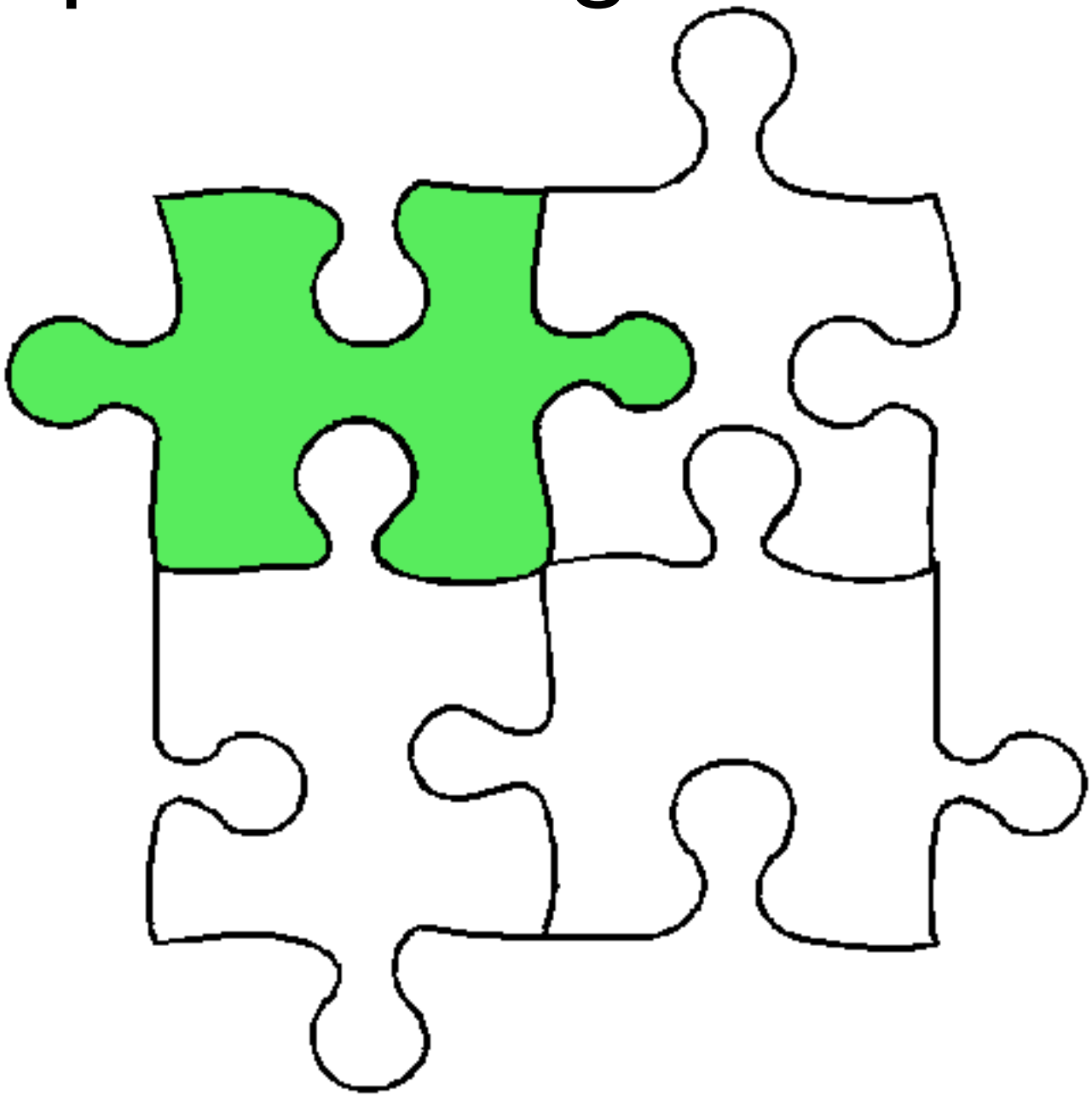
```
{include file="parts/header.tpl.html"}

<h3>Hello, world!</h3>

{include file="parts/footer.tpl.html"}
```

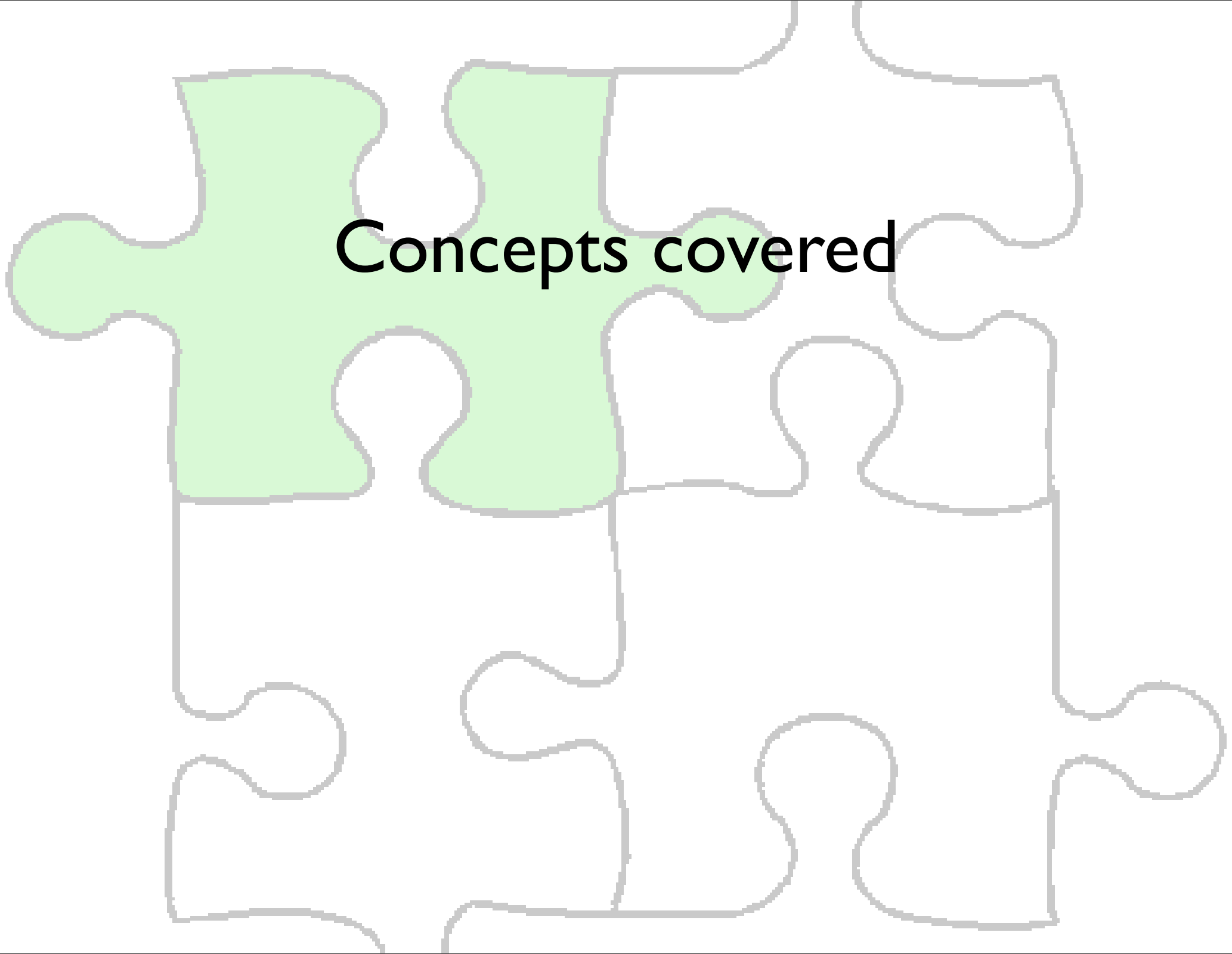

Let's take a look

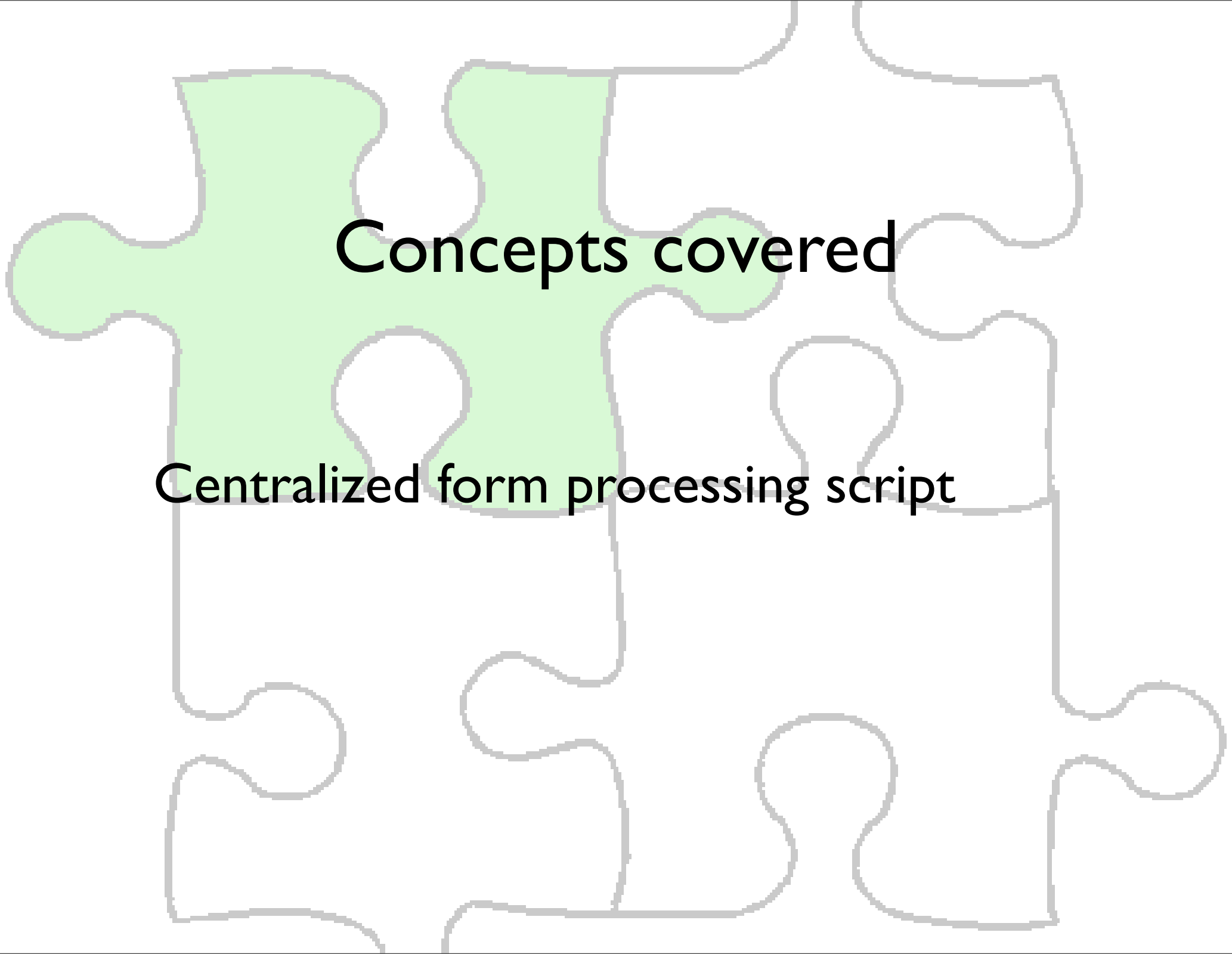
[page source]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <title>Agathongroup.com</title>
        <meta name="description" content="Agathon Group provides hosting,
consulting, and programming." />
        <link rel="stylesheet" href="/css/styles.css" type="text/css" />
</head>
<body><h3>Hello, world!</h3></body>
</html>
```

Step 1: Creating email forms




Concepts covered



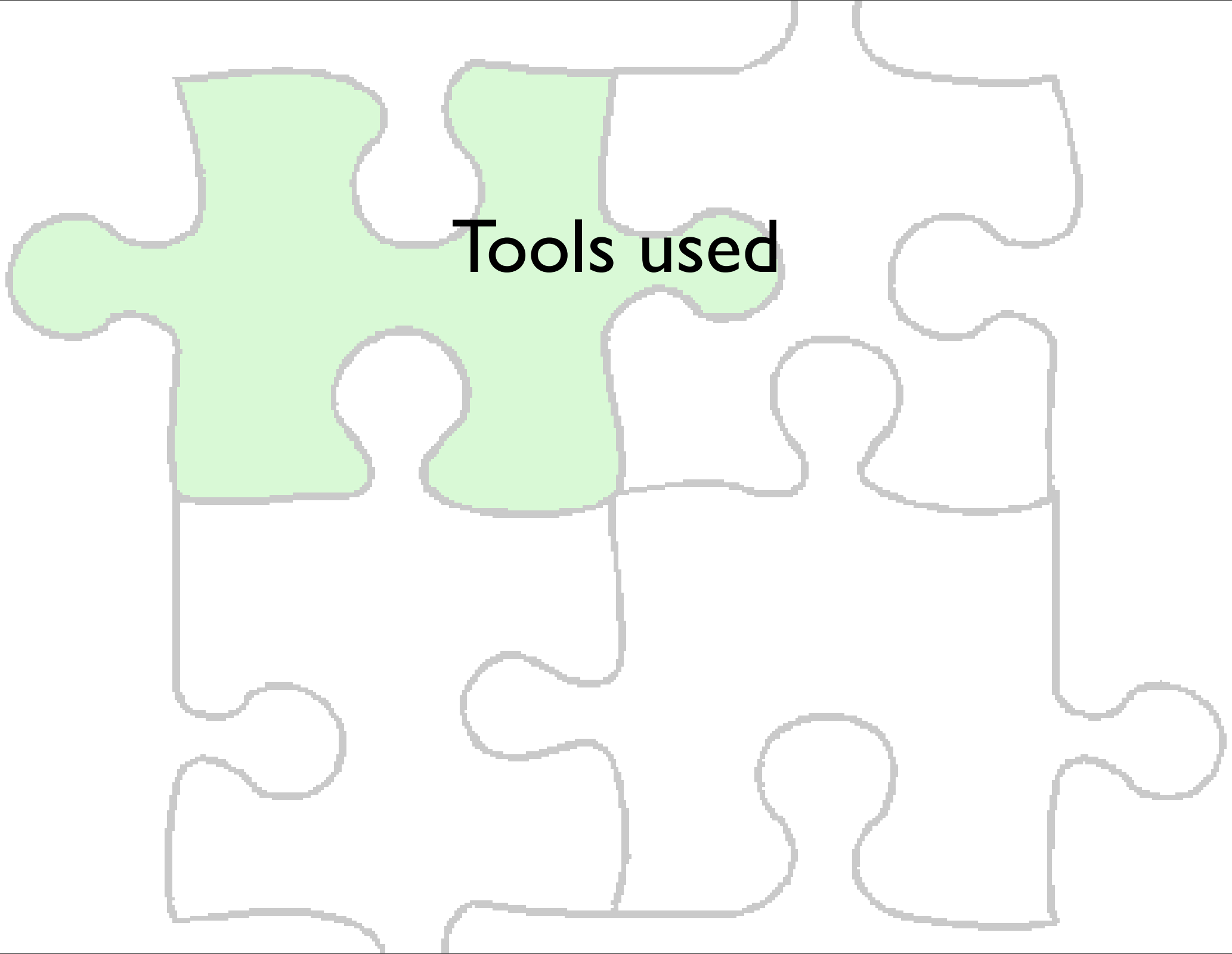
Concepts covered

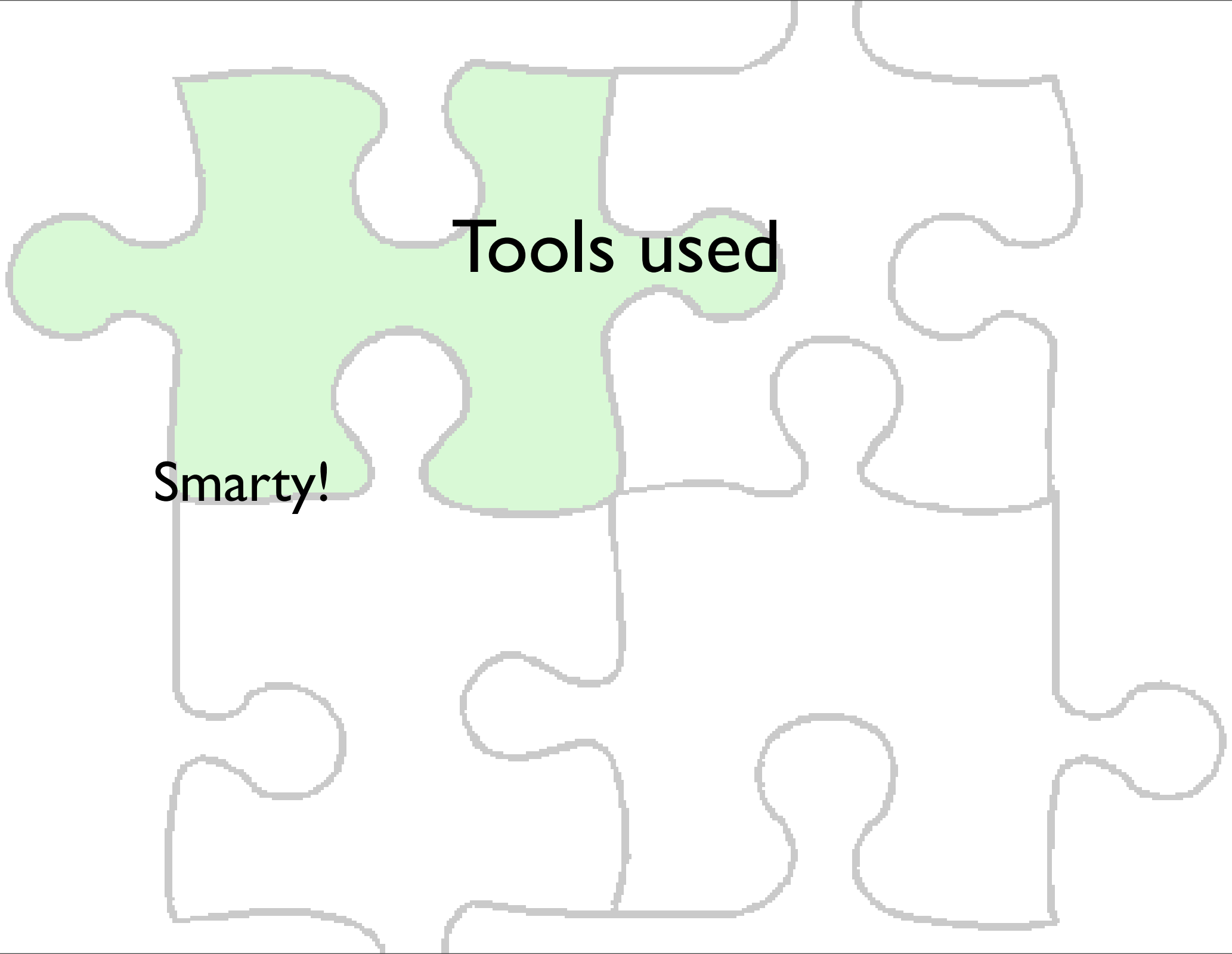
Centralized form processing script

Concepts covered

Centralized form processing script

Modular form processing, based on function

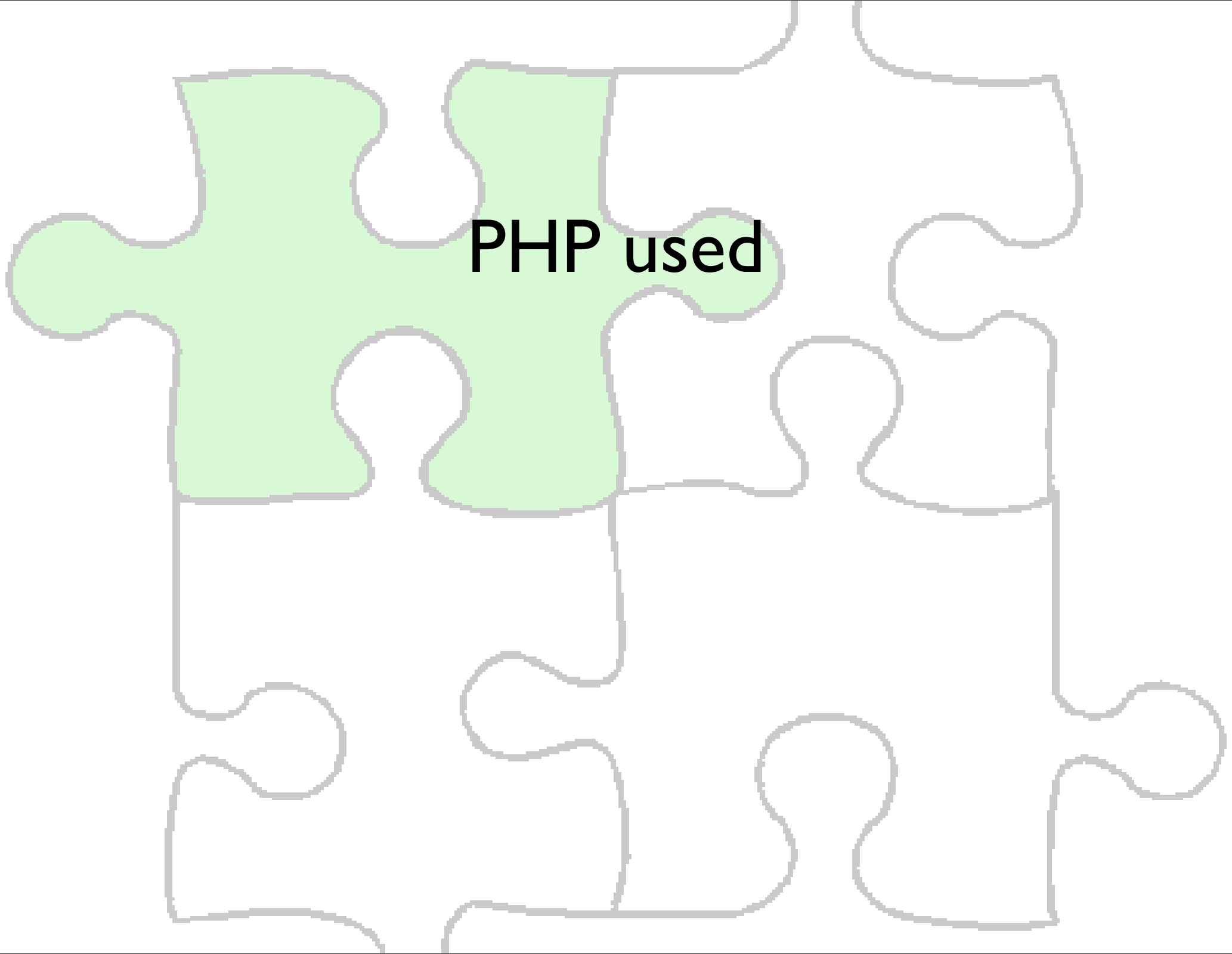

Tools used


Tools used
Smarty!

Tools used

Smarty!

Smarty's `html_options` and related functions for quickly creating forms


PHP used

PHP used

`mail()`: basic function to send email

PHP used

`mail()`: basic function to send email

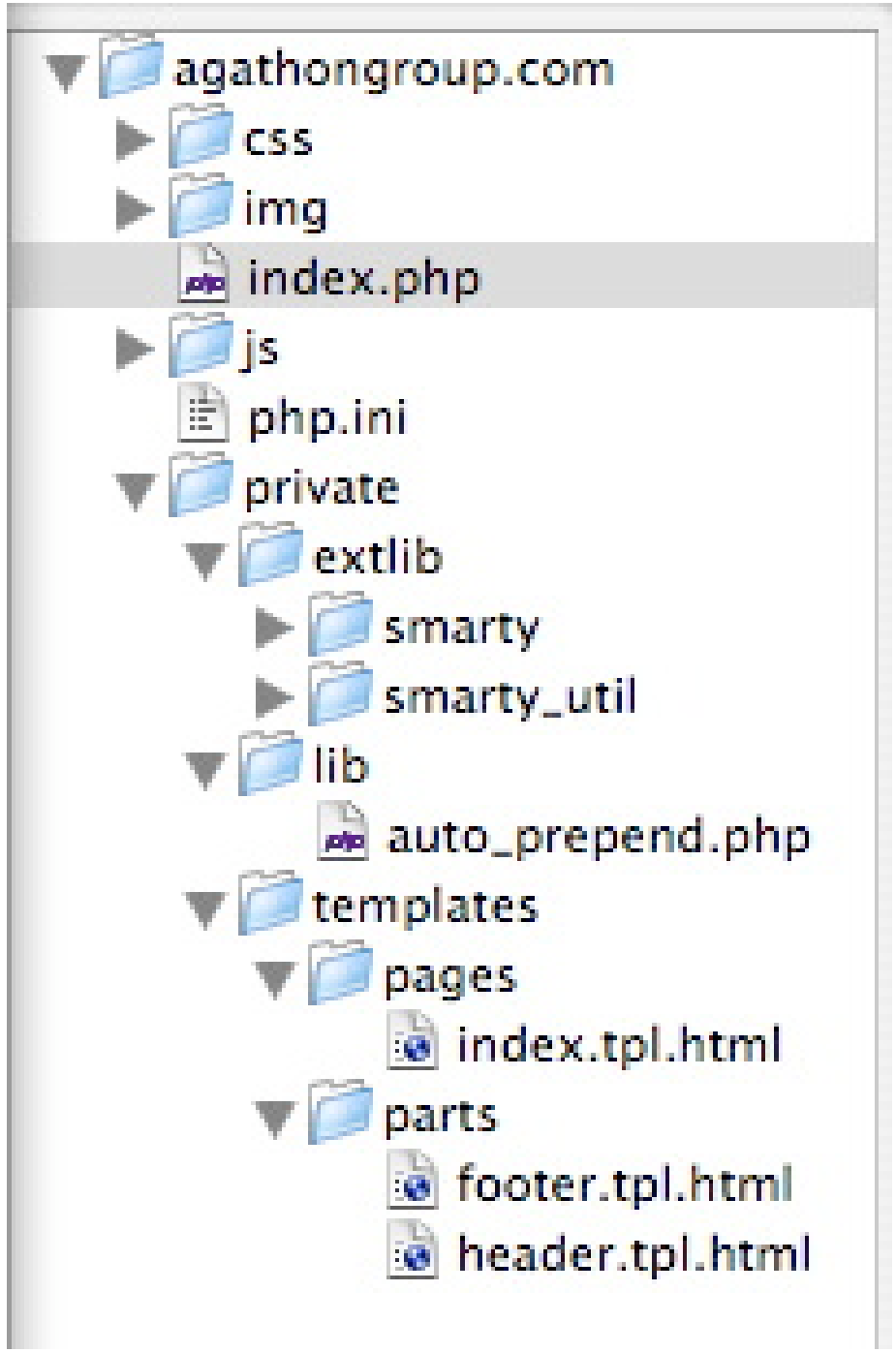
`include_once()`, `require_once()`: load in a common set of functions

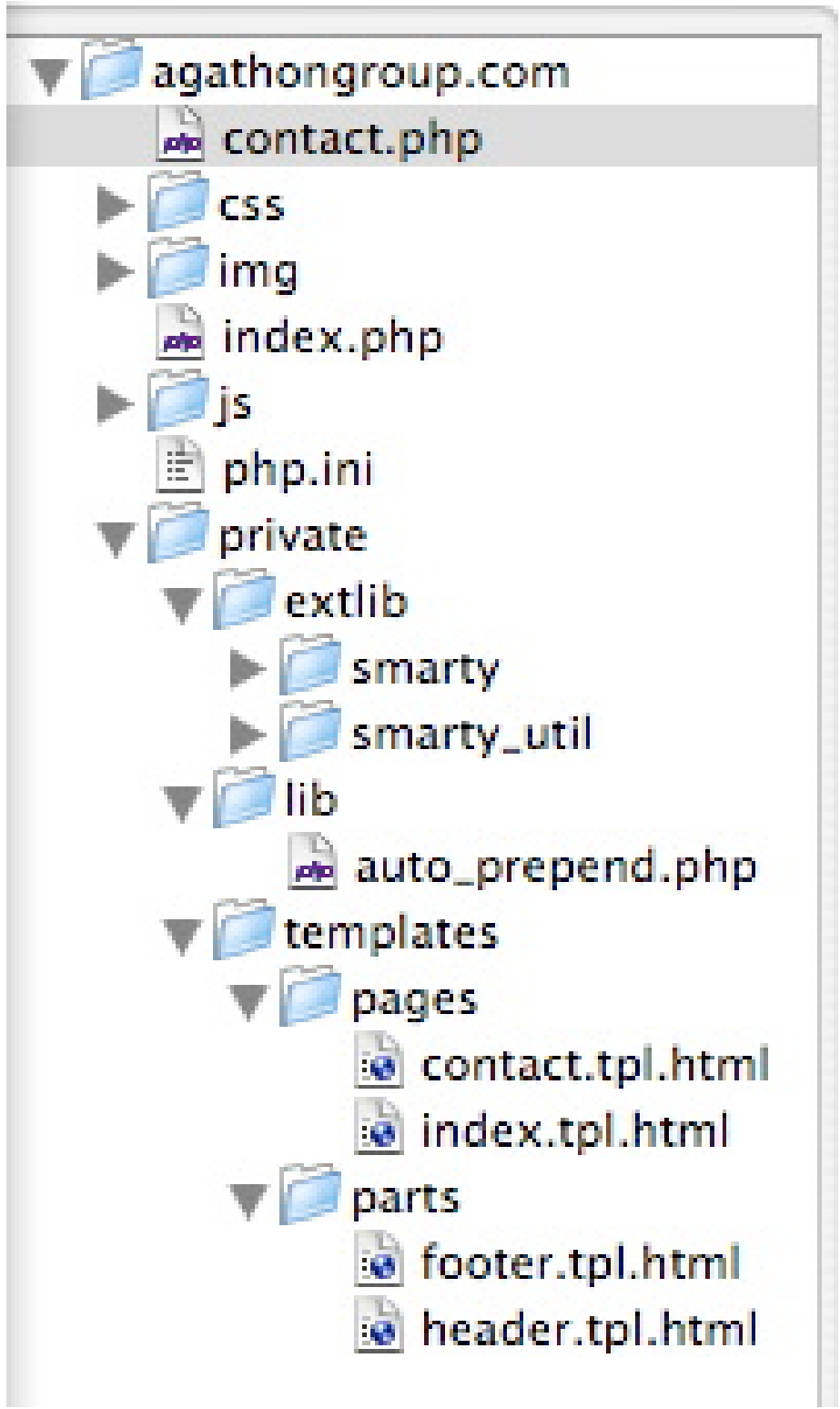
PHP used

`mail()`: basic function to send email

`include_once()`, `require_once()`: load in a common set of functions

Superglobals (`$_GET`, `$_POST`, `$_REQUEST`)


agathongroup.com
css
img
index.php
js
php.ini
private
extlib
smarty
smarty_util
lib
auto_prepend.php
templates
pages
index.tpl.html
parts
footer.tpl.html
header.tpl.html


agathongroup.com
contact.php
css
img
index.php
js
php.ini
private
extlib
smarty
smarty_util
lib
auto_prepend.php
templates
pages
contact.tpl.html
index.tpl.html
parts
footer.tpl.html
header.tpl.html

contact.php

```php
// common settings are auto-included by auto_prepend.php

// Set up variables to be used in the website
$page_vars = array(

    // page title, description
    'title'       => 'Contact us',
    'description' => 'How to contact us.',

    // values for our contact form
    'email_recipients'  => array(
        'alan'   => 'Alan Ritari',
        'joel'   => 'Joel Boonstra',
        'peter'  => 'Peter Green',
        'rachel' => 'Rachel Ramirez',
        'ron'    => 'Ron Veenstra',
    ),

    // form variables
    'form' => array(
        'errors_to'  => $_SERVER['REQUEST_URI'],
        'success_to' => '/thanks.php',
        'action'     => 'email',
    ),
);

// send our variables to Smarty
$smarty->assign($page_vars);

// display the page
display_full_page('contact.tpl.html');
?>
```

private/templates/pages/contact.tpl.html

```
<h3>Contact us!</h3>

{if $error_msg}
<p><strong>Error:</strong> {$error_msg}</p>
{/if}

<form action="/bin/form_processor.php" method="post">
    <p>Your email address: <input type="text" name="email" size="30" /></p>
    <p>Send message to:
    <select name="recipient">
    {html_options options=$email_recipients}
    </select>
    </p>
    <p><textarea name="message" rows="5" cols="20"></textarea></p>
    <input type="hidden" name="errors_to" value="{$form.errors_to}" />
    <input type="hidden" name="success_to" value="{$form.success_to}" />
    <input type="hidden" name="action" value="{$form.action}" />
    <p><input type="submit" value="Send us email!" /></p>
</form>
```

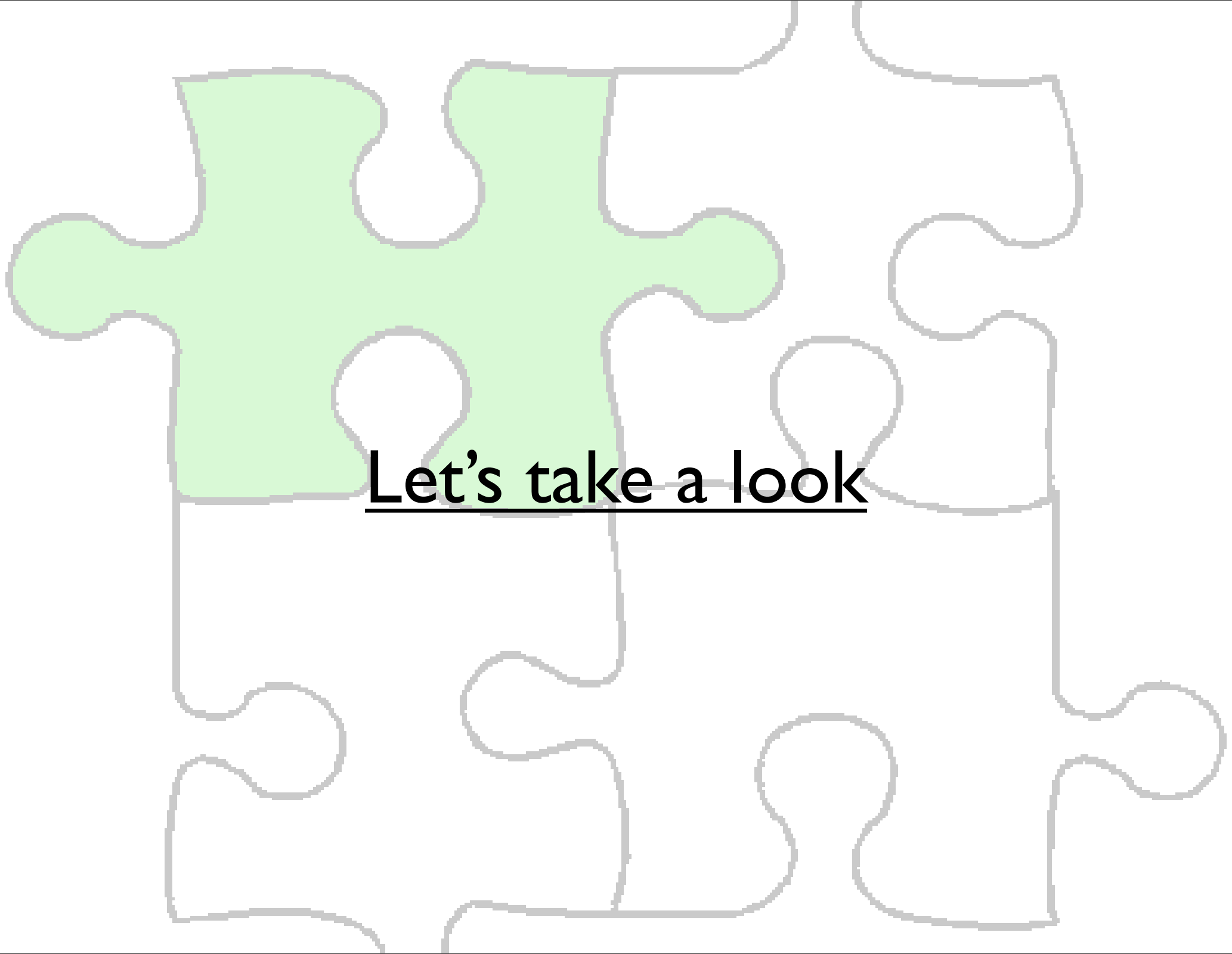
bin/form_processor.php

```php
<?php
/*==
 | file     : bin/form_processor.php
 | author   : Agathon Group
 | contact  : support@agathongroup.com
 |
 | purpose : processes a form from the web
 |
 | $Id$
 +==*/
require_once('lib/form_functions.inc.php');
switch ($_REQUEST['action']) {
    case 'email' :
        handle_email_form();
        break;
    default :
        handle_bad_action();
        break;
}
?>
```

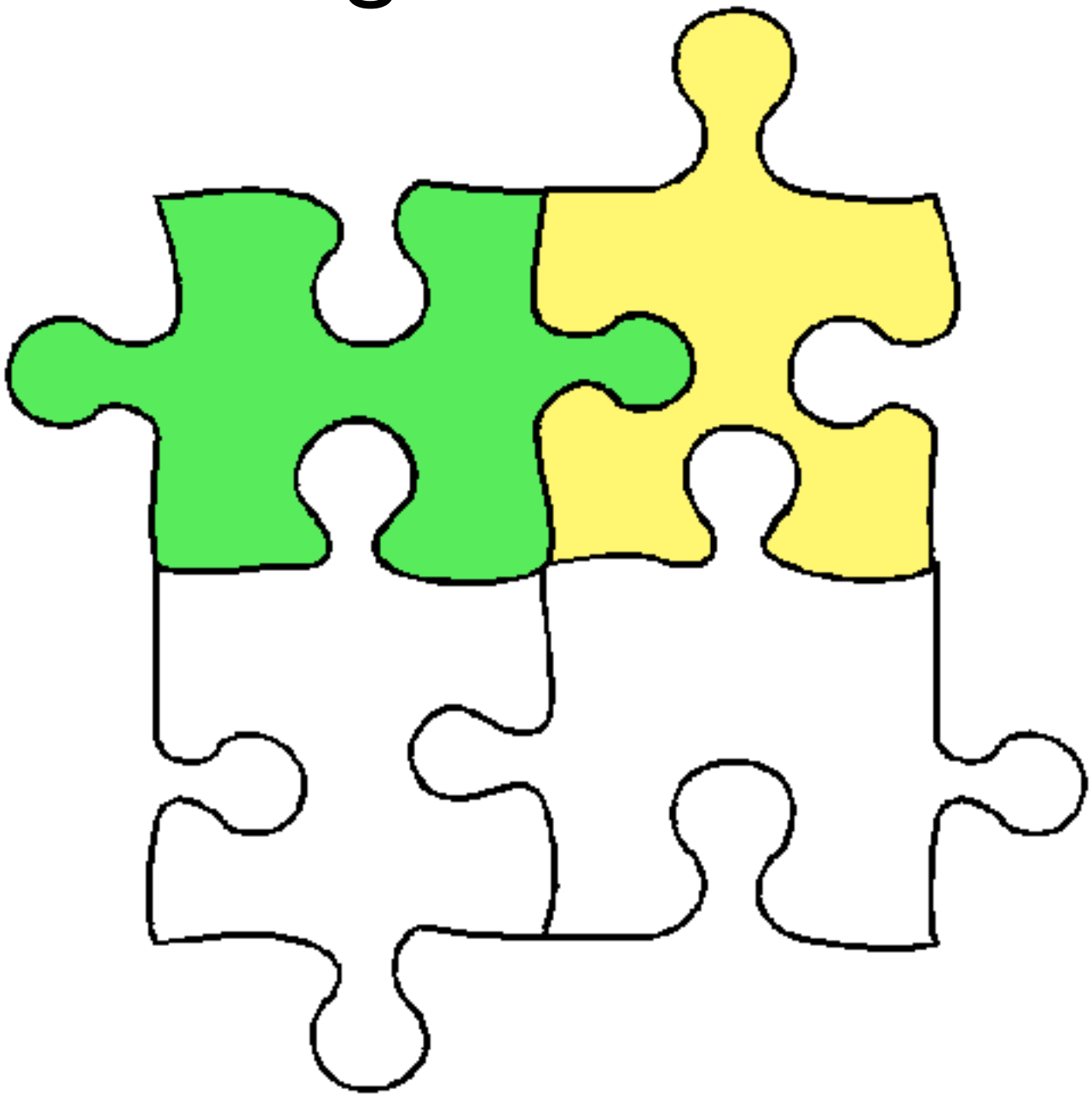
private/lib/form_functions.inc.php

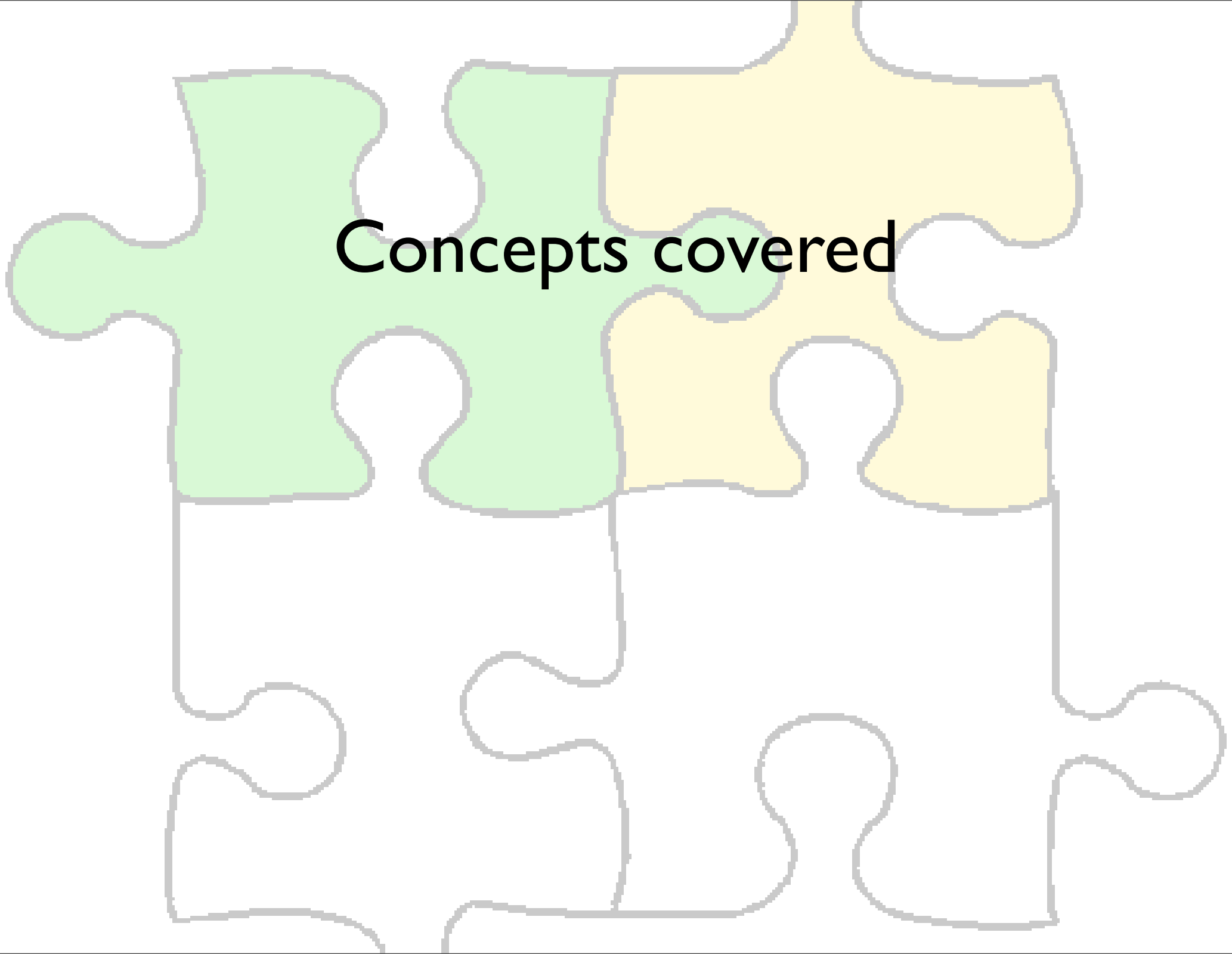
```php
<?php
/*==
 | file    : private/lib/form_functions.inc.php
 | author  : Agathon Group
 | contact : support@agathongroup.com
 |
 | purpose : functions for handling form input
 |
 | $Id$
 +==*/
// handles sending of email from an email form
function handle_email_form() {
    # our email recipients
    $recipients = array(
        'alan'   => 'alan@agathongroup.com',
        'joel'   => 'joel@agathongroup.com',
        'peter'  => 'pcg@agathongroup.com',
        'rachel' => 'rachel@agathongroup.com',
        'ron'    => 'ron@agathongroup.com',
    );

    # our incoming data
    $email     = $_POST['email'];
    $message   = $_POST['message'];
    $recipient = $recipients[$_POST['recipient']];
```

```
    # first, check that we have a valid message and email address
    if (!($email && $message)) {
        $error = 'Need an email address and a message.';
        redirect_and_exit($_POST['errors_to'], $error);
    }

    # next, check that we have a valid email recipient
    if (!$recipient) {
        $error = 'That recipient is invalid.';
        redirect_and_exit($_POST['errors_to'], $error);
    }

    # we have a good recipient, message, and email address.
    $success = send_email($email, $recipient, $message,
                          "Message from the website");
    if (!$success) {
        $error = 'Error sending email.';
        redirect_and_exit($_POST['errors_to'], $error);
    }
    else {
        redirect_and_exit($_POST['success_to']);
    }
}
```


Let's take a look

Step 2: Adding database interaction




Concepts covered

Concepts covered

Extending the centralized form processing script to handle a new function

Concepts covered

Extending the centralized form processing script to handle a new function

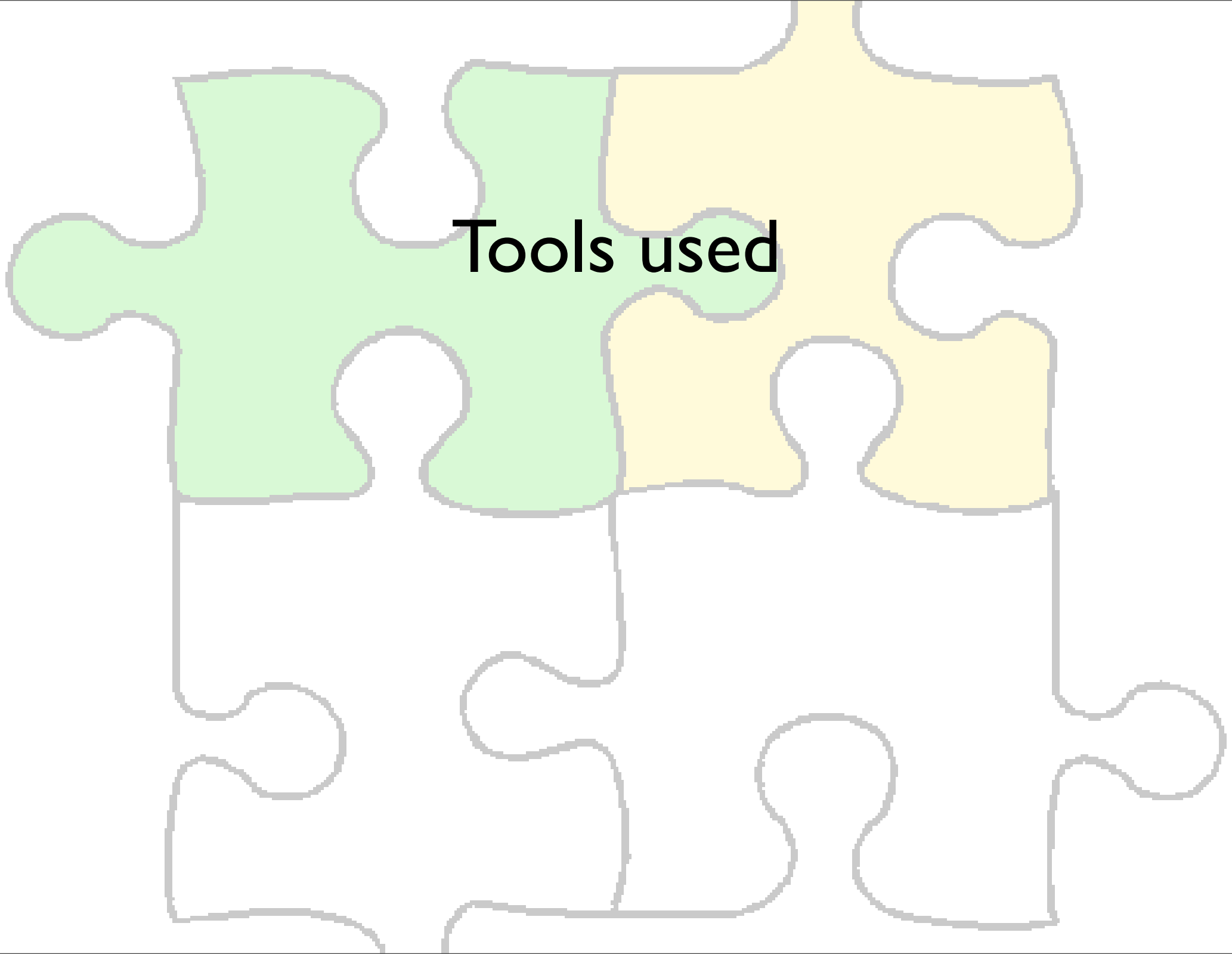
Checking database error statuses

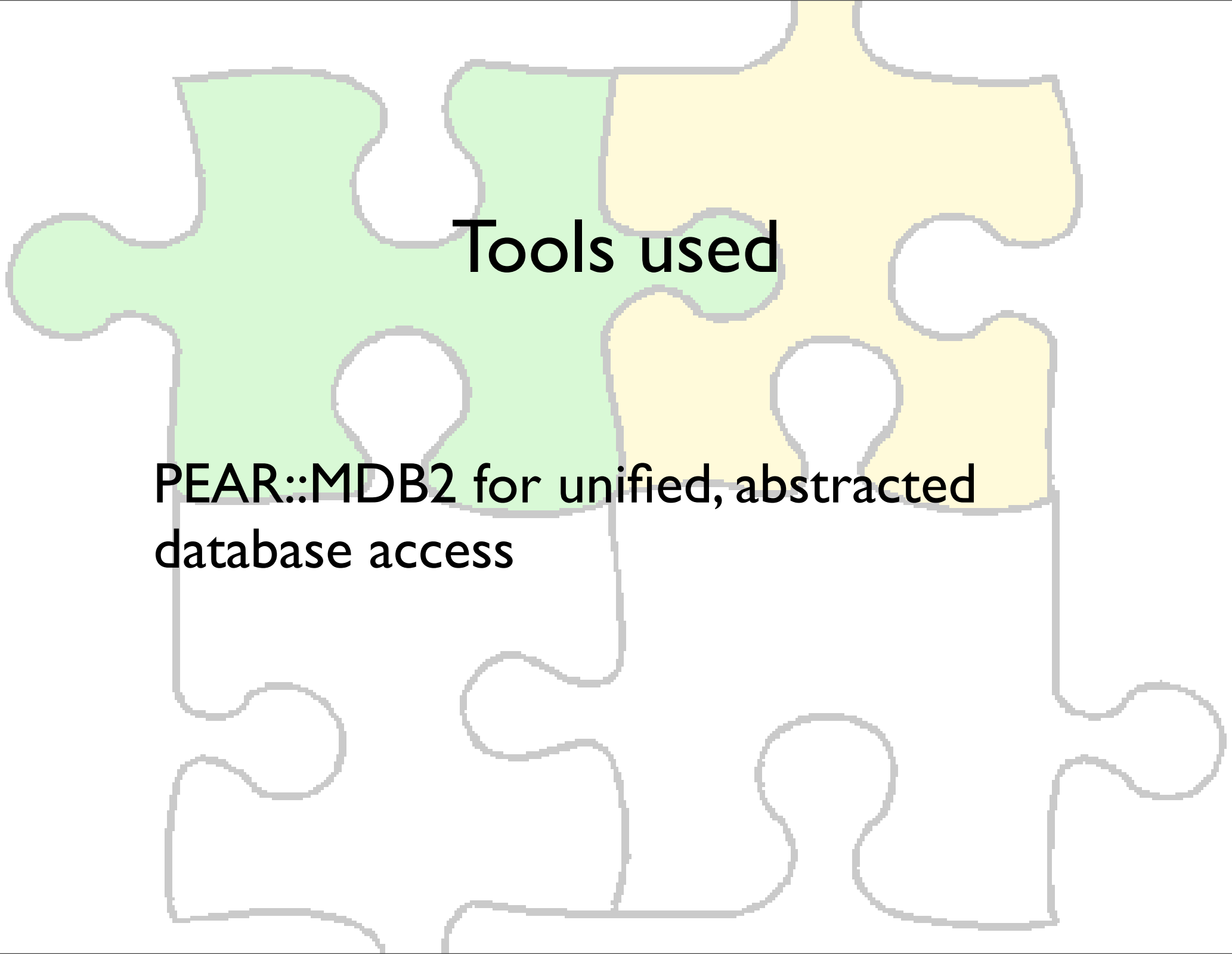
Concepts covered

Extending the centralized form processing script to handle a new function

Checking database error statuses

Centralized, separate config file for DB settings


Tools used



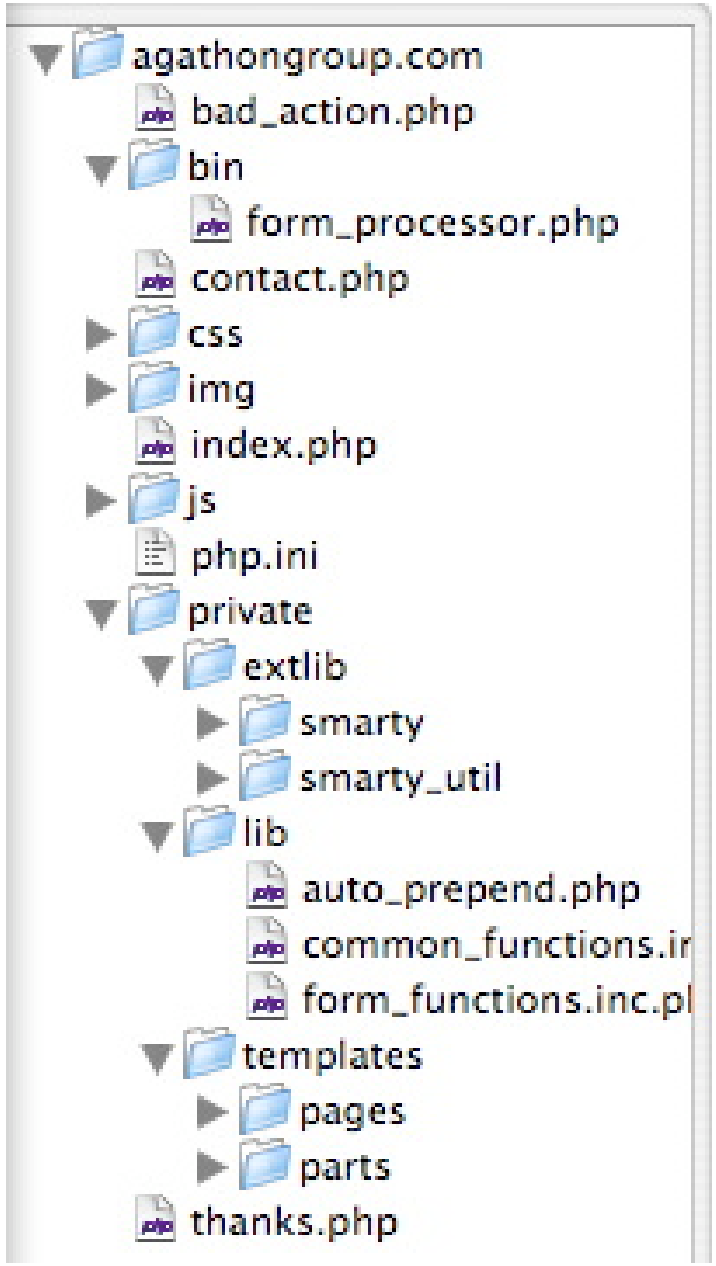
Tools used

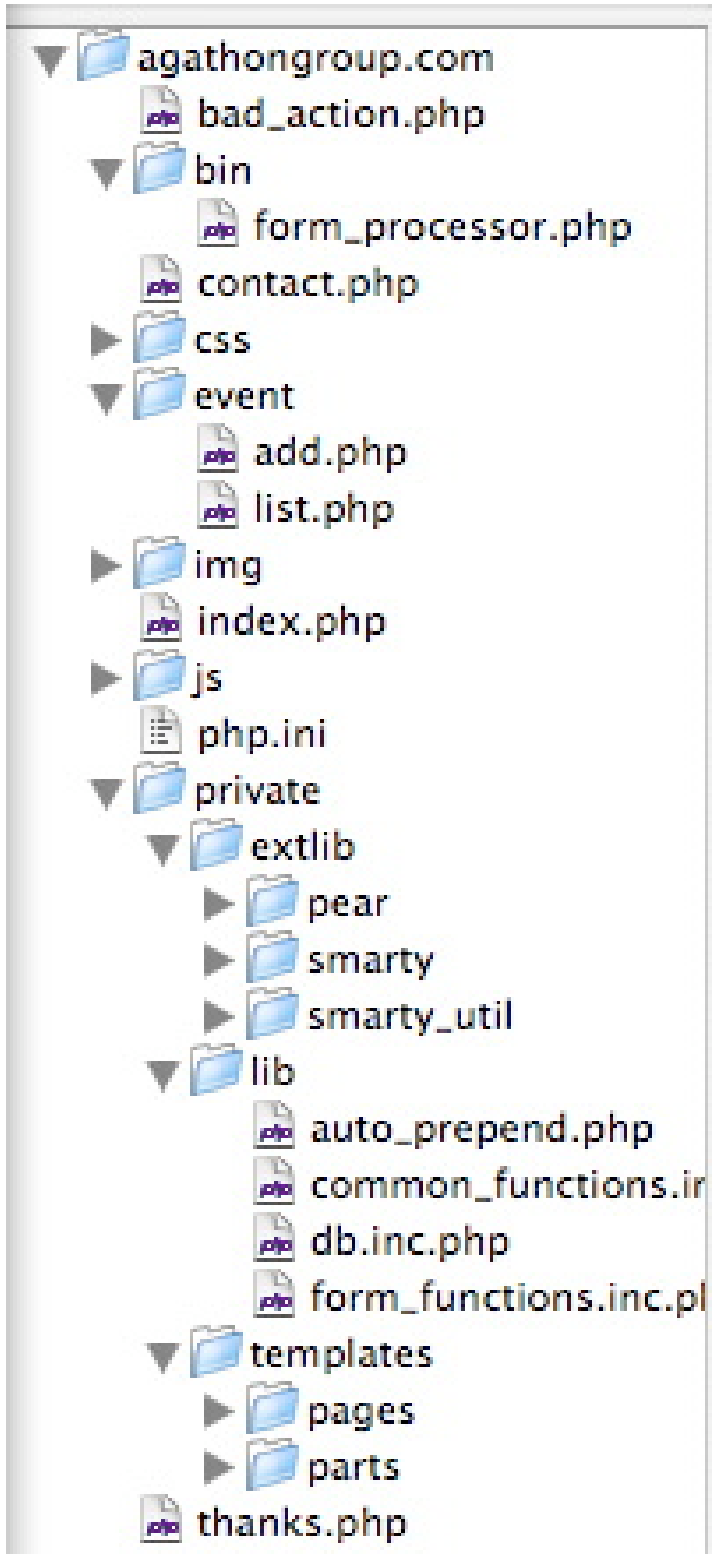
PEAR::MDB2 for unified, abstracted database access

Tools used

PEAR::MDB2 for unified, abstracted database access

Superior to mysql_* and mysqli_* calls!


agathongroup.com
bad_action.php
bin
form_processor.php
contact.php
css
img
index.php
js
php.ini
private
extlib
smarty
smarty_util
lib
auto_prepend.php
common_functions.ir
form_functions.inc.pl
templates
pages
parts
thanks.php


agathongroup.com
bad_action.php
bin
form_processor.php
contact.php
css
event
add.php
list.php
img
index.php
js
php.ini
private
extlib
pear
smarty
smarty_util
lib
auto_prepend.php
common_functions.ir
db.inc.php
form_functions.inc.pl
templates
pages
parts
thanks.php

private/templates/pages/event/add.tpl.html

```
<h3>Add an event to the schedule</h3>

{include file="parts/form_error.tpl.html"}

<form action="/bin/form_processor.php" method="post">
    <p>Event name:
       <input type="text" name="event_name" value="" id="event_name" /></p>
    <p>Event date:
       {html_select_date day_value_format="%02d"}
    </p>
    <p>Event description:</p>
    <p><textarea name="event_description" rows="8" cols="40"></textarea></p>

    {include file="parts/form_input.tpl.html"}

    <input type="submit" value="Add event" />
</form>
```

bin/form_processor.php

```php
<?php
/*==
 | file     : bin/form_processor.php
 | author   : Agathon Group
 | contact  : support@agathongroup.com
 |
 | purpose : processes a form from the web
 |
 | $Id$
 +==*/
require_once('lib/form_functions.inc.php');
switch ($_REQUEST['action']) {
    case 'email' :
        handle_email_form();
        break;
    default :
        handle_bad_action();
        break;
}
?>
```

bin/form_processor.php

```php
<?php
/*==
 | file    : bin/form_processor.php
 | author  : Agathon Group
 | contact : support@agathongroup.com
 |
 | purpose : processes a form from the web
 |
 | $Id$
 +==*/
require_once('lib/form_functions.inc.php');
switch ($_REQUEST['action']) {
    case 'email' :
        handle_email_form();
        break;
    case 'add_event' :
        handle_event_form();
        break;
    default :
        handle_bad_action();
        break;
}
?>
```

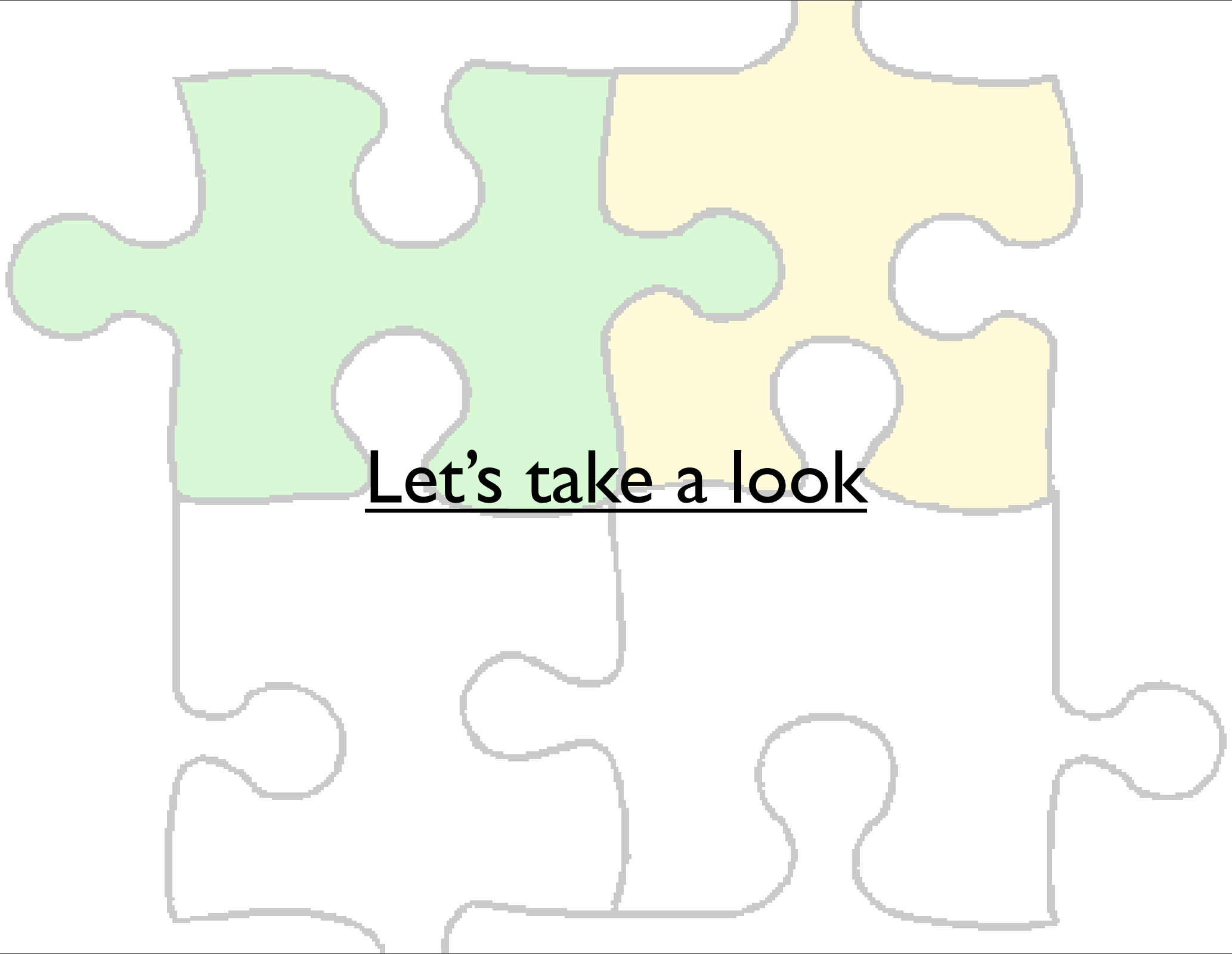
private/lib/form_functions.inc.php

```php
// handles the event form submission
function handle_event_form() {
    $dbh =& get_dbh();

    # can we connect to the database?
    if (!is_object($dbh)) {
        redirect_and_exit($_POST['errors_to'], $dbh);
    }

    # validate name/description
    $e_name     = $_POST['event_name'];
    $e_descript = $_POST['event_description'];
    if (!($e_name && $e_descript)) {
        $error = 'Need an event name and description.';
        redirect_and_exit($_POST['errors_to'], $error);
    }

    # validate date
    $e_date = $_POST['Date_Year'] . $_POST['Date_Month'] . $_POST['Date_Day'];
    if ($e_date < date('Ymd')) {
        $error = 'Date cannot be in the past.';
        redirect_and_exit($_POST['errors_to'], $error);
    }
```

```
    # insert into database
    $insert = "
        INSERT
          INTO events (event_name, event_date, event_descript)
        VALUES ('$e_name', '$e_date', '$e_descript')
    ";
    $result =& $dbh->query($insert);

    # if DB error, redirect
    if (PEAR::isError($result)) {
        $error = $result->getMessage();
        redirect_and_exit($_POST['errors_to'], $error);
    }

    # else, success
    else {
        redirect_and_exit($_POST['success_to']);
    }
}
```

private/lib/db.inc.php

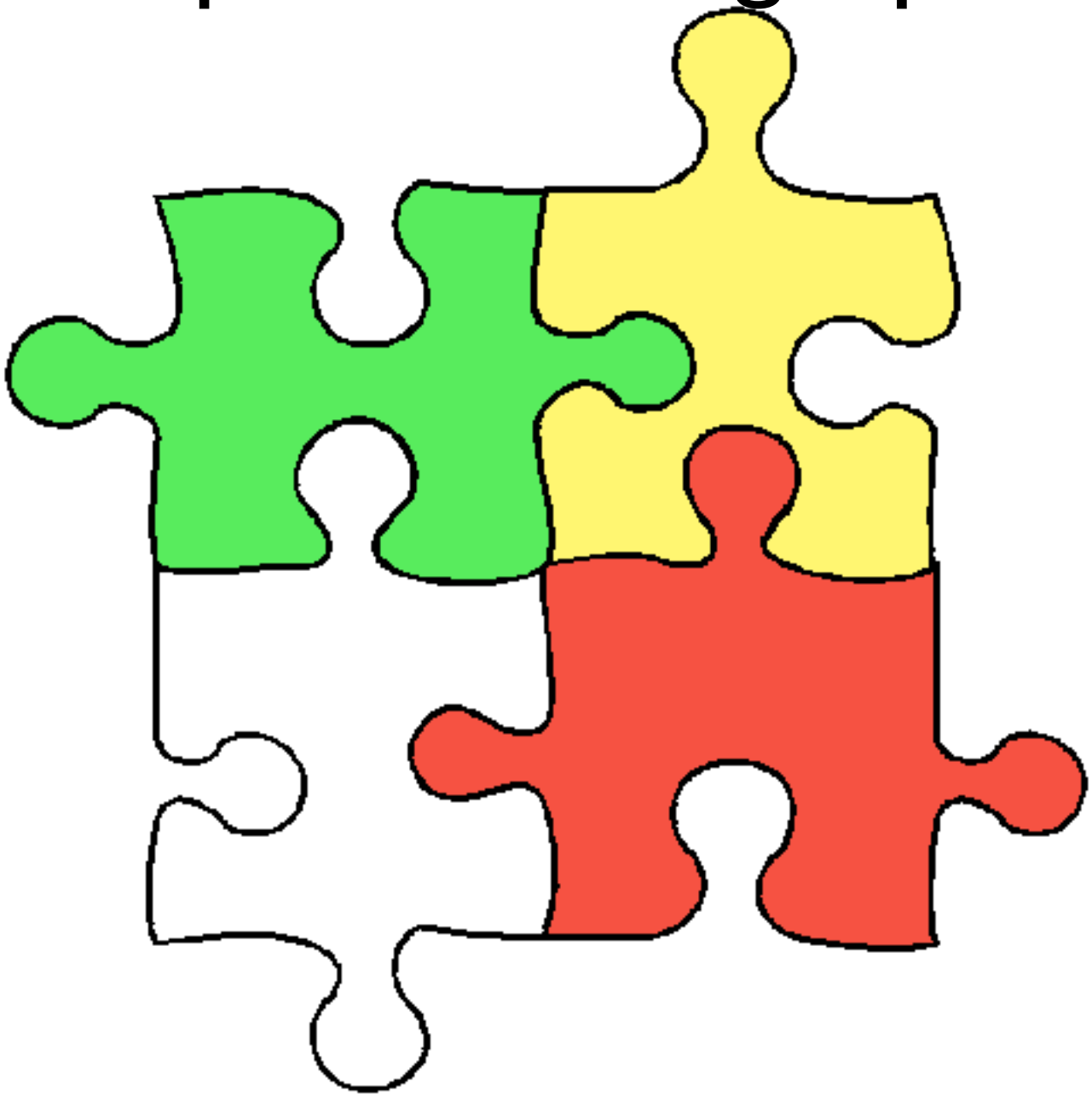
```php
<?php
/*==
 | file    : private/lib/db.inc.php
 | author  : Agathon Group
 | contact : support@agathongroup.com
 |
 | purpose : provides a DB connection using PEAR::MDB2
 |
 | $Id$
 +==*/
require_once 'MDB2.php';

// create a DB handle, only if one has not already been created
function get_dbh() {
    $dsn = array(
        'phptype'  => 'mysql',
        'username' => 'gospelcon',
        'password' => 'abc123',
        'hostspec' => 'localhost',
        'database' => 'gospelcon',
    );
    $dbh =& MDB2::singleton($dsn);
    if (PEAR::isError($dbh)) { return $dbh->getMessage(); }
    return $dbh;
}
?>
```

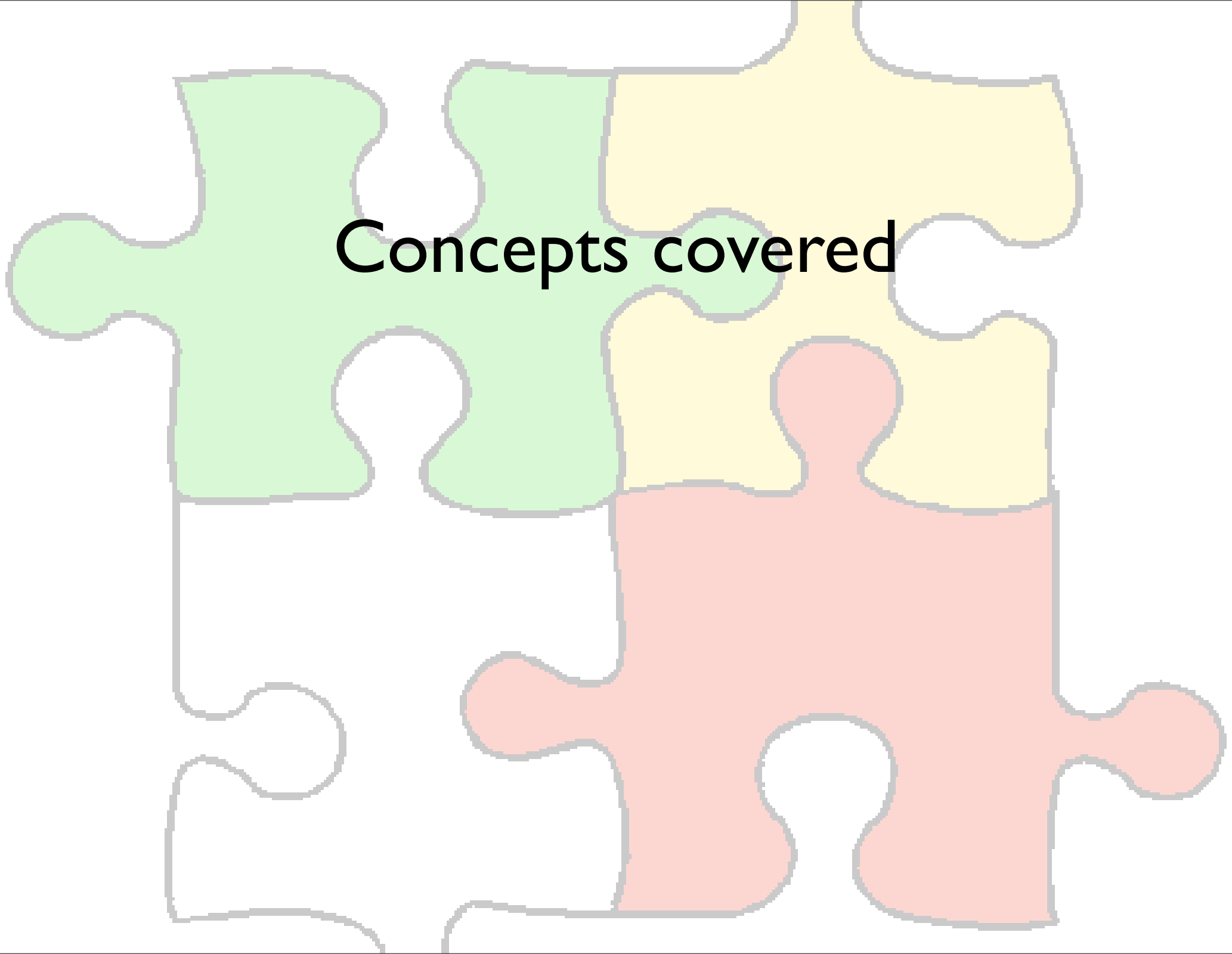

Let's take a look

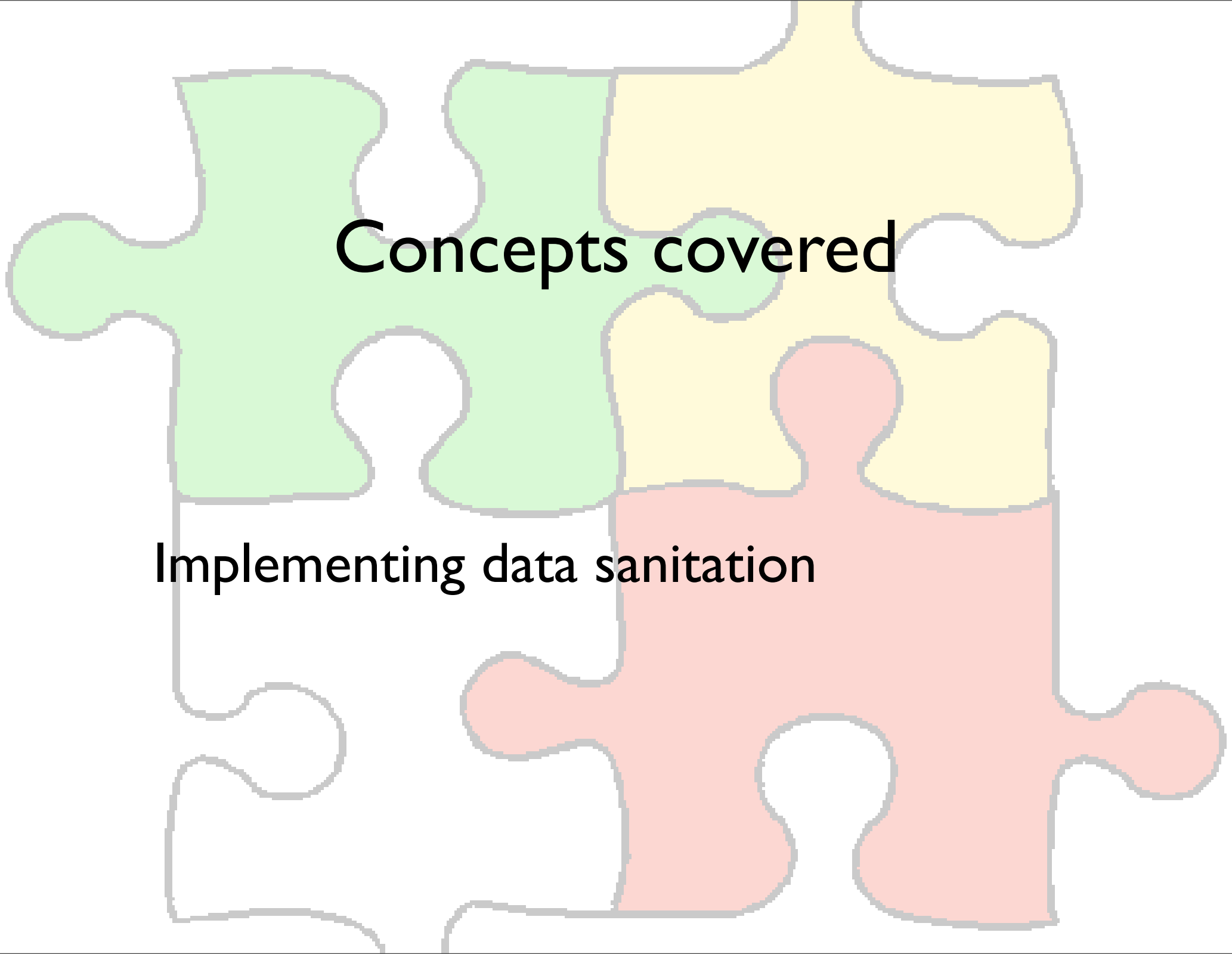
private/templates/pages/event/add.tpl.html

```
<h3>Add an event to the schedule</h3>

{include file="parts/form_error.tpl.html"}

<form action="/bin/form_processor.php" method="post">
    <p>Event name:
       <input type="text" name="event_name" value="" id="event_name" /></p>
    <p>Event date:
       {html_select_date day_value_format="%02d"}
    </p>
    <p>Event description:</p>
    <p><textarea name="event_description" rows="8" cols="40"></textarea></p>

    {include file="parts/form_input.tpl.html"}

    <input type="submit" value="Add event" />
</form>
```

Step 3: Sanitizing input

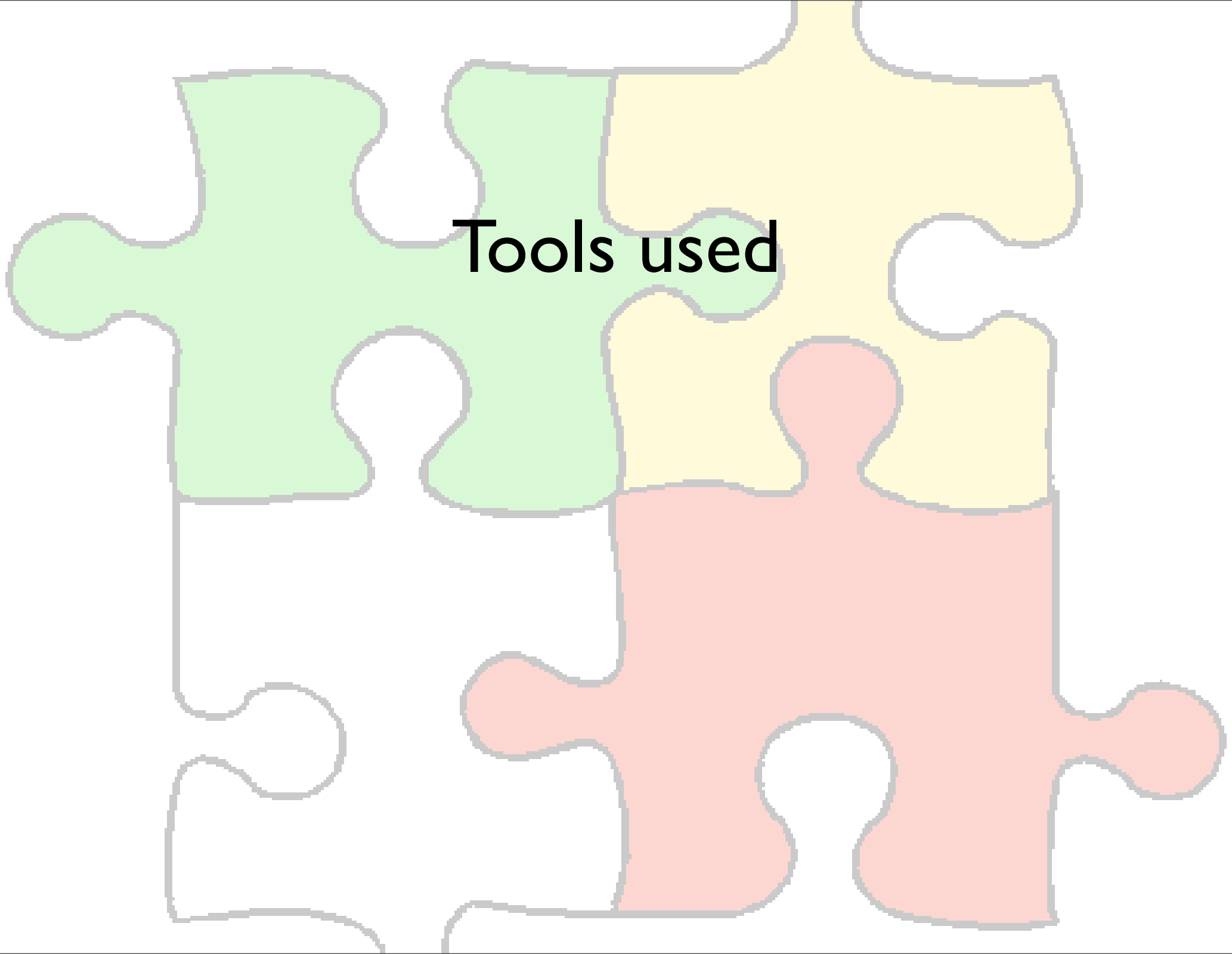


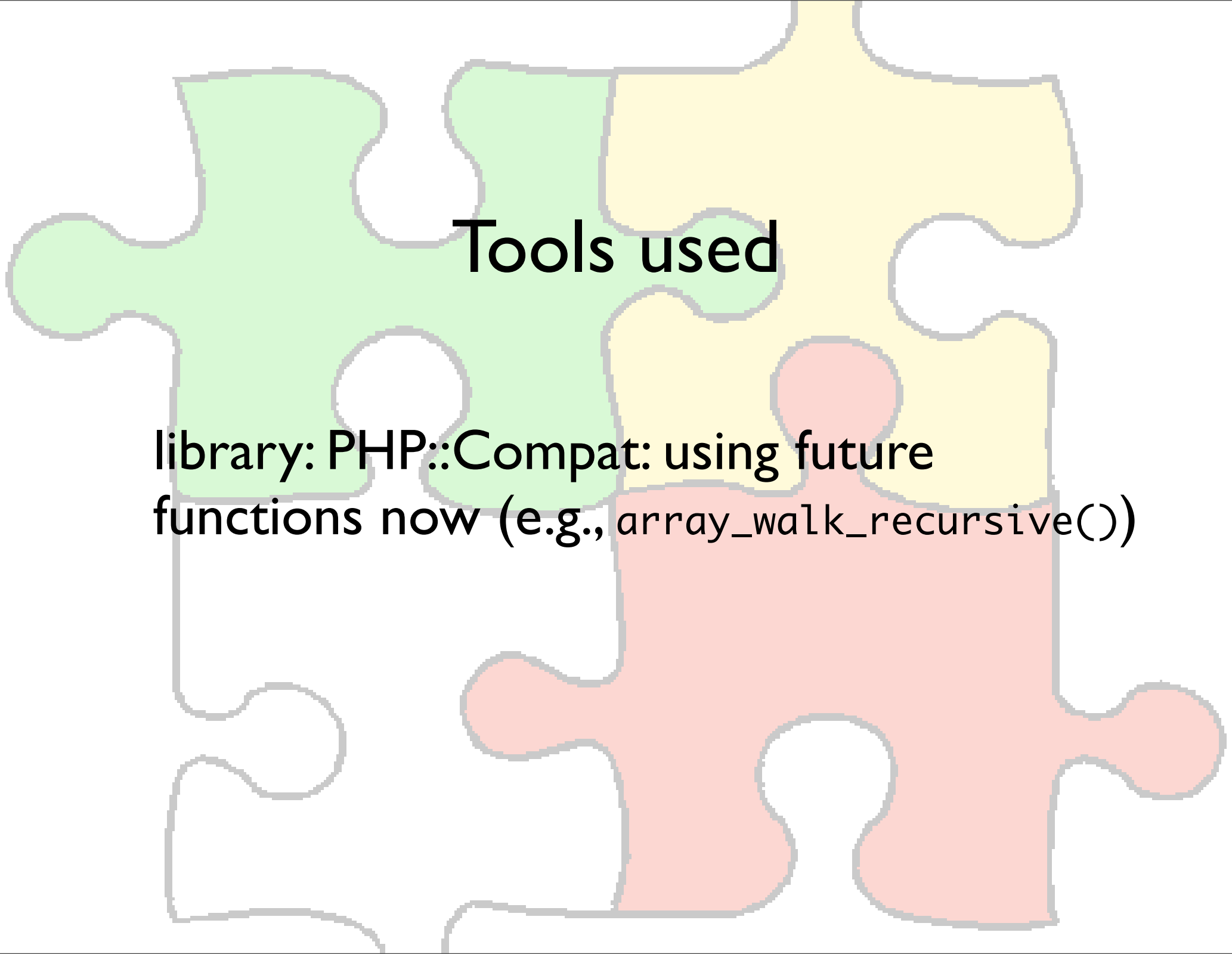

Concepts covered



Concepts covered

Implementing data sanitation


Tools used



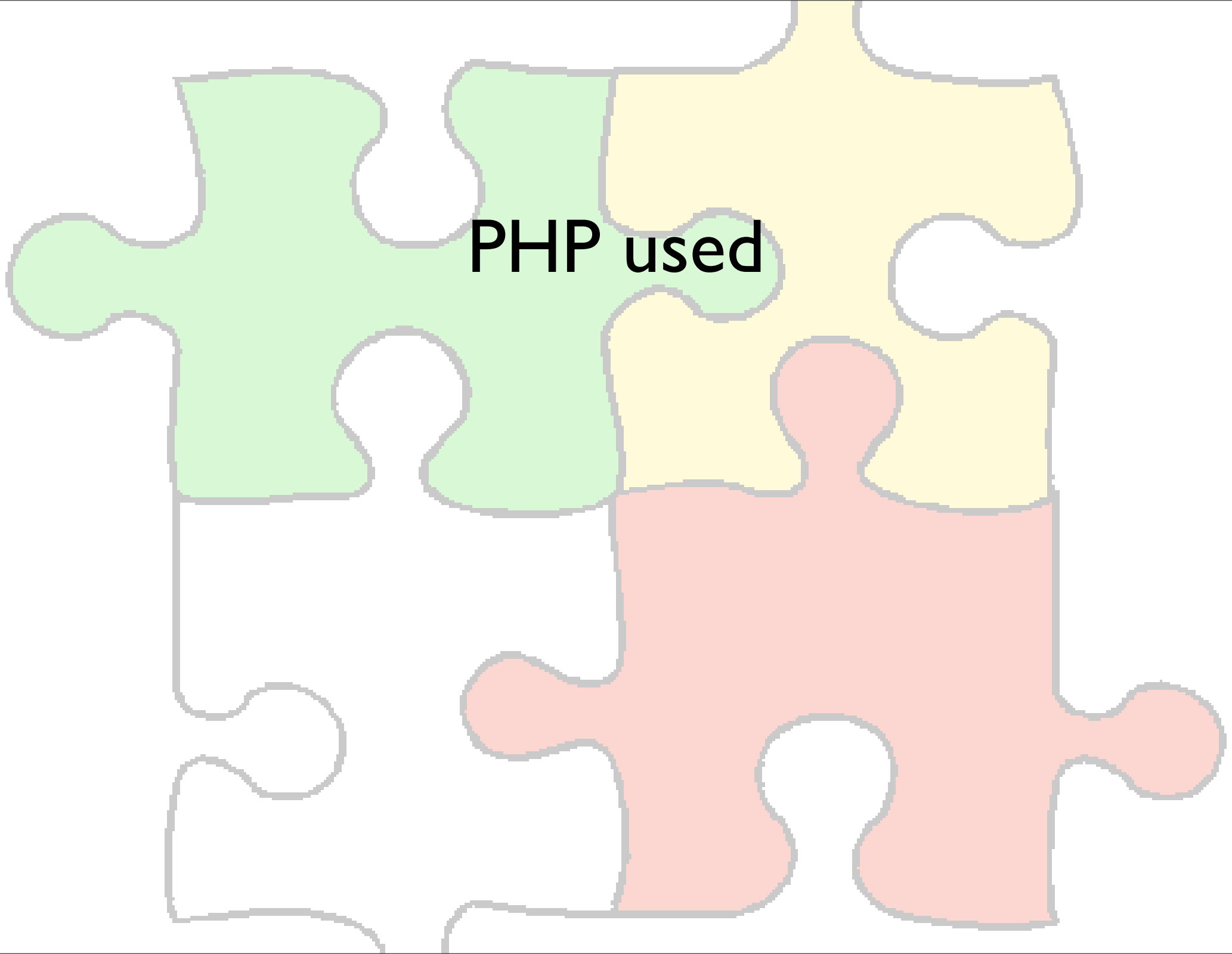
Tools used

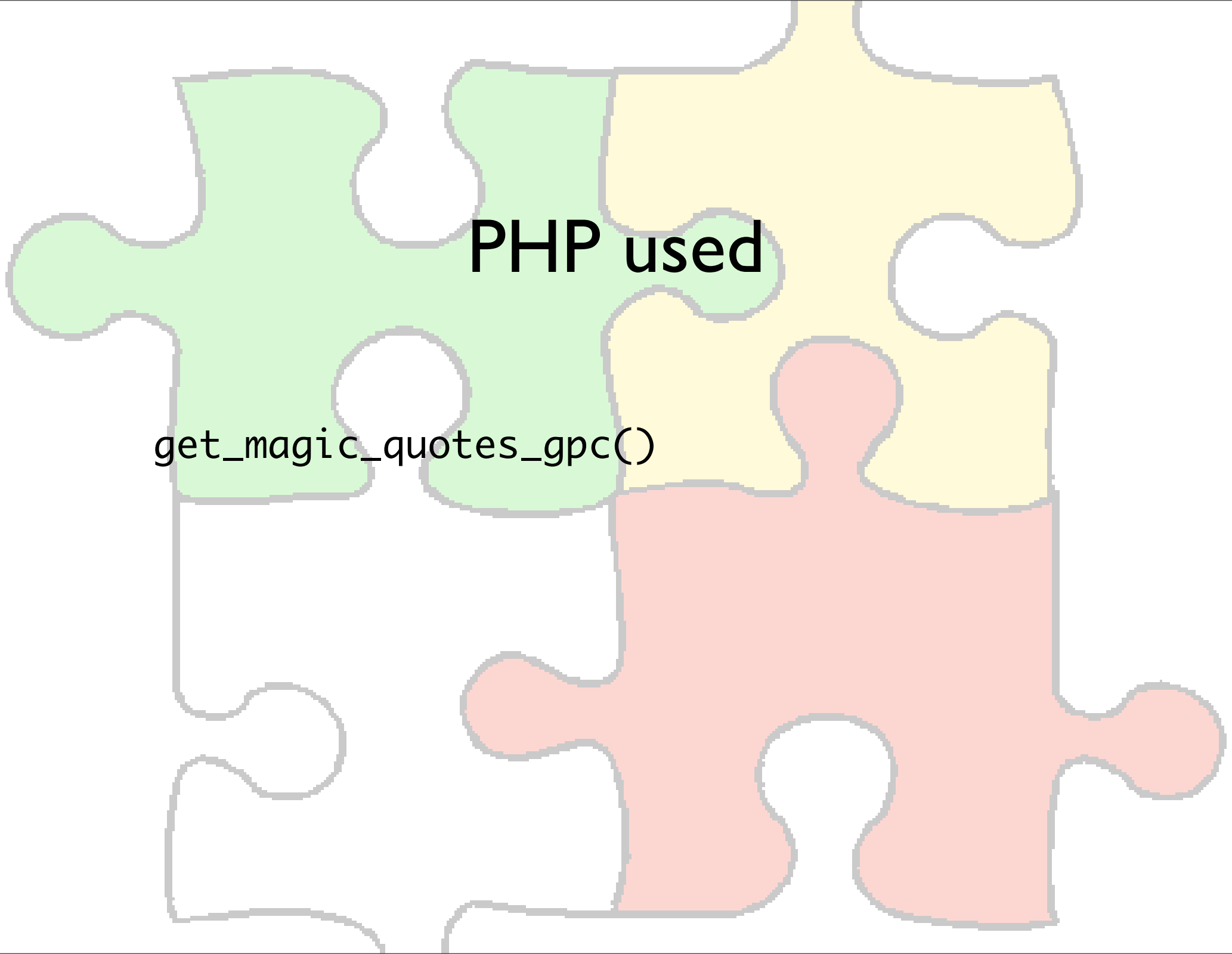
library: PHP::Compat: using future functions now (e.g., `array_walk_recursive()`)

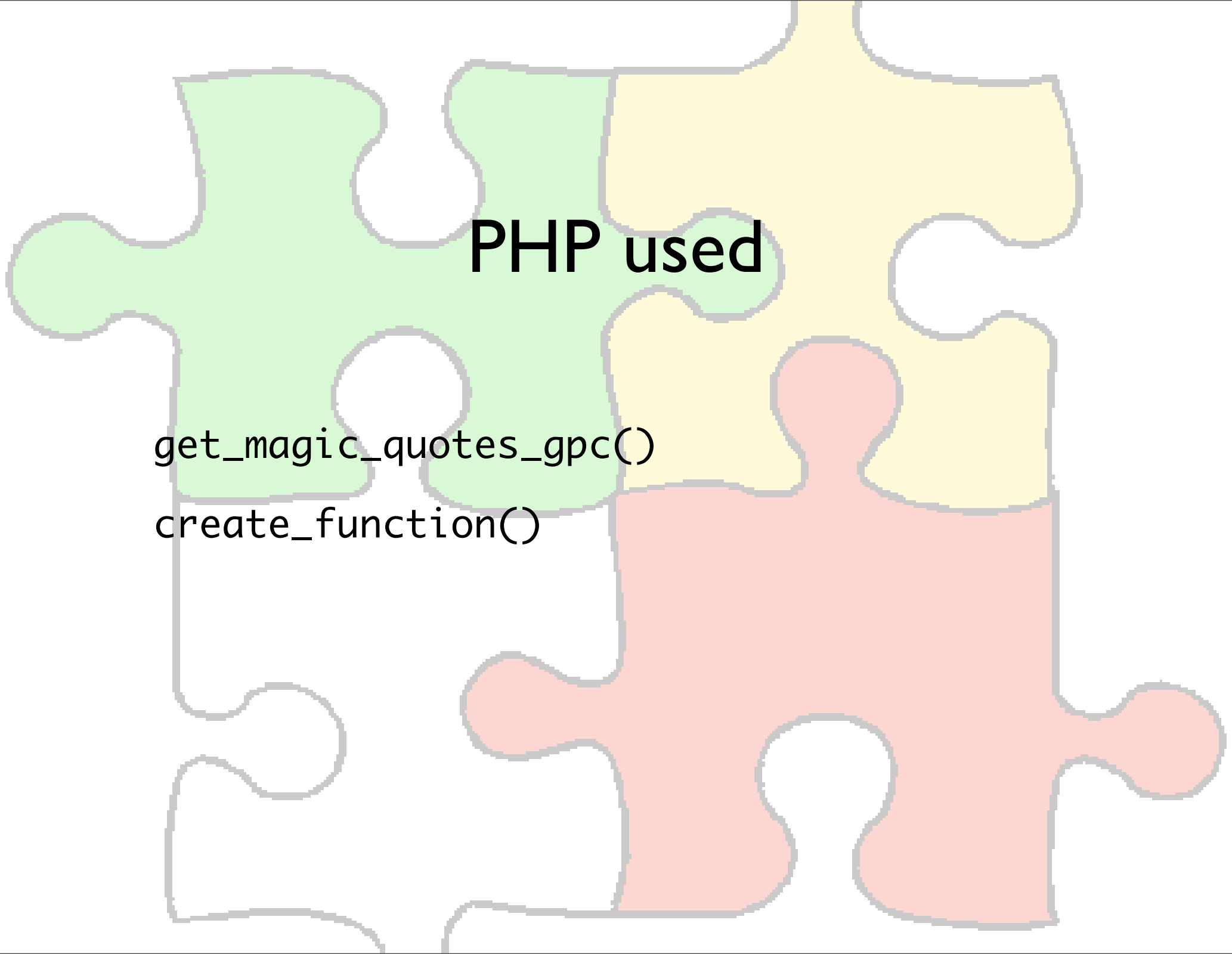
Tools used

library: PHP::Compat: using future functions now (e.g., `array_walk_recursive()`)

library: PEAR::Mail: send email without worrying about spammers

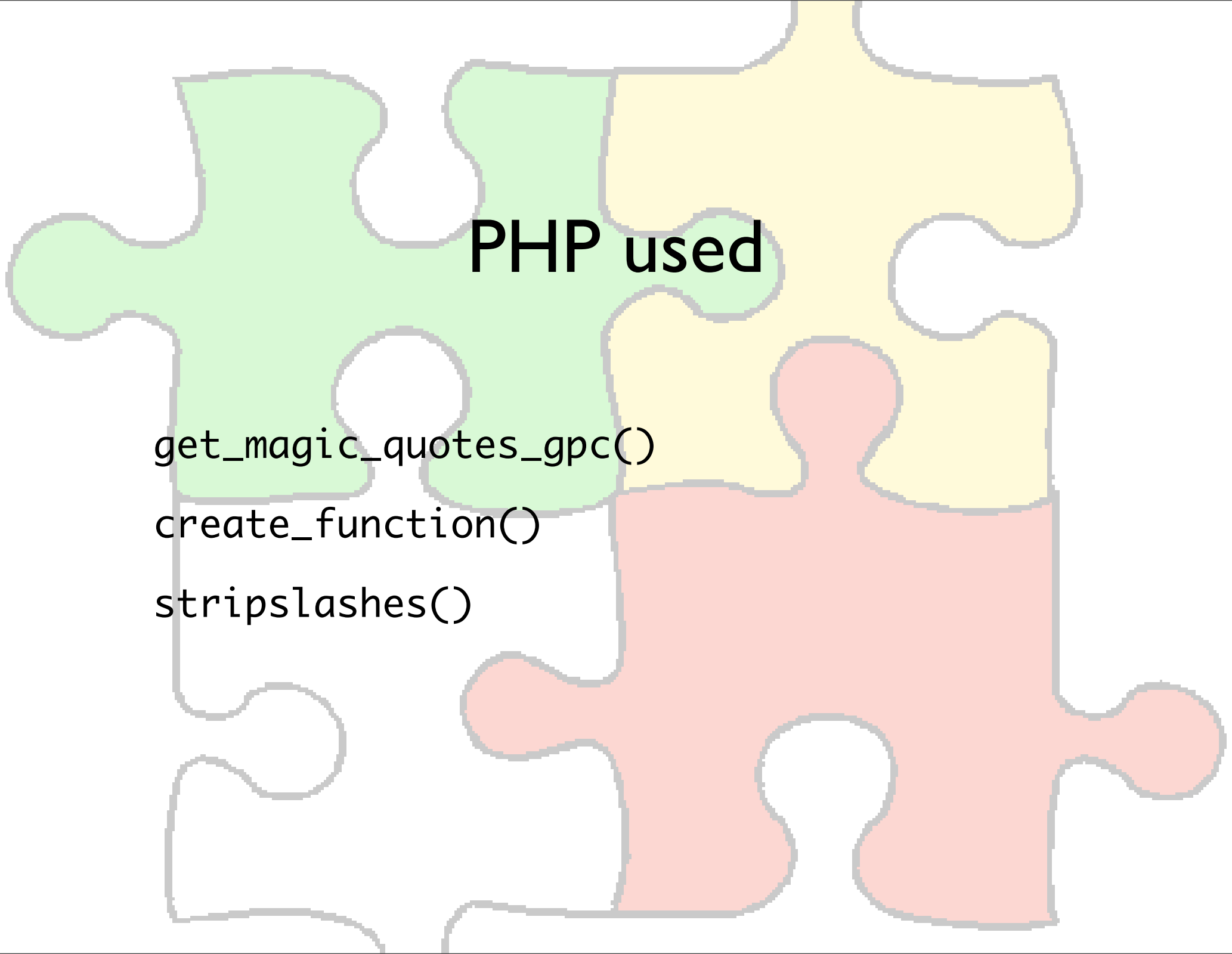

PHP used


PHP used
get_magic_quotes_gpc()



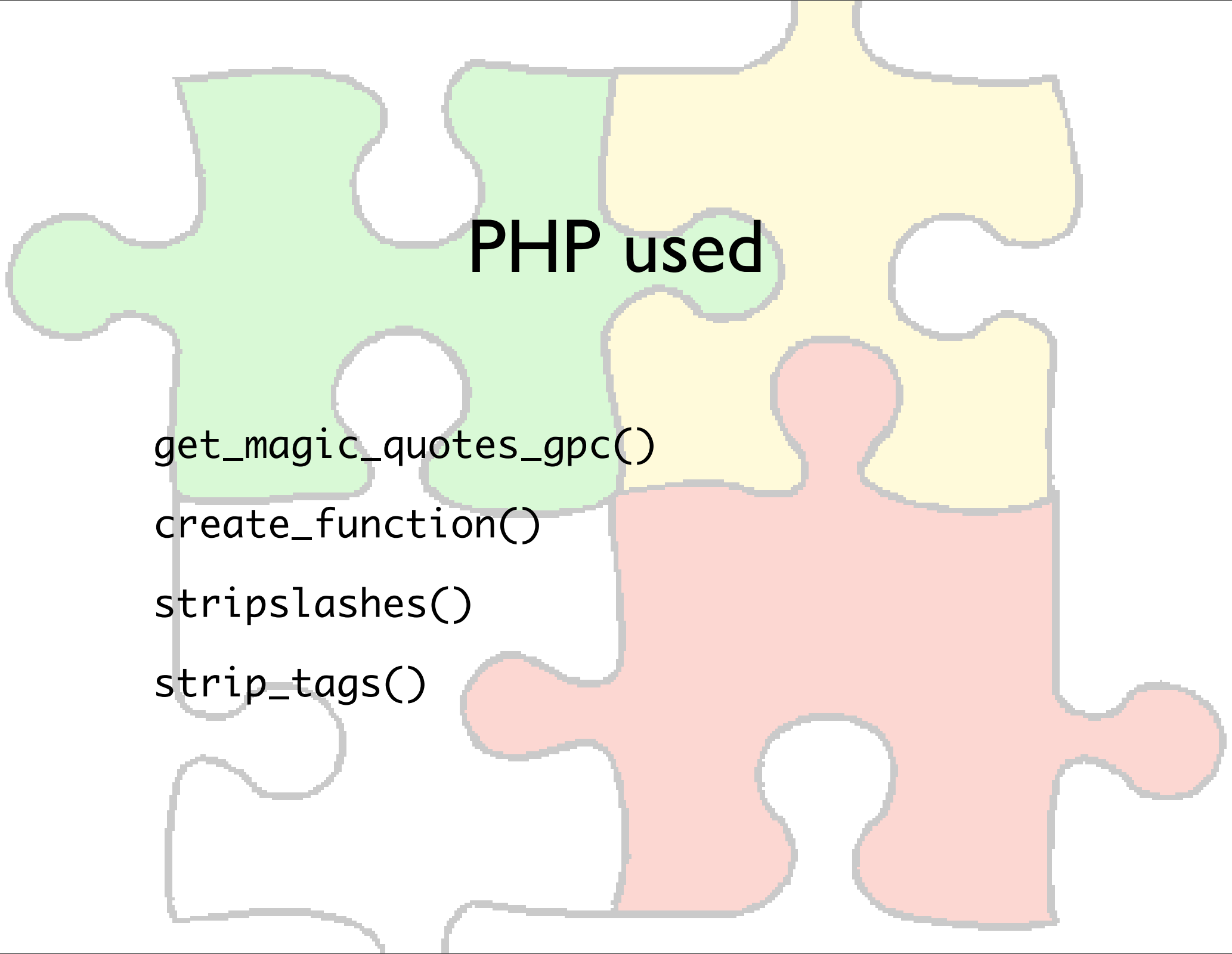
PHP used

```
get_magic_quotes_gpc()

create_function()
```



PHP used

```
get_magic_quotes_gpc()

create_function()

stripslashes()
```



PHP used

```
get_magic_quotes_gpc()

create_function()

stripslashes()

strip_tags()
```

private/lib/auto_prepend.php (excerpt)

```php
// need to un-do PHP's magical quote-escaping?
if (get_magic_quotes_gpc()) {
    require_once('PHP/Compat.php');
    PHP_Compat::loadFunction('array_walk_recursive');
    $fn = create_function('&$v,$k', '$v=stripslashes($v);');
    array_walk_recursive($_GET,     $fn);
    array_walk_recursive($_REQUEST, $fn);
    array_walk_recursive($_POST,    $fn);
}
```

private/lib/common_functions.inc.php (old & busted)

```php
// sends email from $from, to $to, with message $message.  returns TRUE or
// FALSE depending on success
function send_email($from, $to, $message, $subject) {
    $headers = "From: $from";
    return mail($to, $subject, $message, $headers);
}
```

private/lib/common_functions.inc.php (new hotness)

```php
// sends email from $from, to $to, with message $message.  returns TRUE or
// FALSE depending on success
function send_email($from, $to, $message, $subject) {
    # use PEAR's Mail class to safely send email
    require_once('Mail.php');
    $mailer =& Mail::factory('mail');
    $headers = array(
        'From'    => $from,
        'Subject' => $subject
    );
    $rc = $mailer->send($to, $headers, $message);
    return (!PEAR::isError($rc));
}
```

private/lib/form_functions.inc.php

old & busted:

```php
# validate name/description
$e_name     = $_POST['event_name'];
$e_descript = $_POST['event_description'];
if (!($e_name && $e_descript)) {
    $error = 'Need an event name and description.';
    redirect_and_exit($_POST['errors_to'], $error);
}
```

private/lib/form_functions.inc.php

old & busted:

```php
# validate name/description
$e_name     = $_POST['event_name'];
$e_descript = $_POST['event_description'];
if (!($e_name && $e_descript)) {
    $error = 'Need an event name and description.';
    redirect_and_exit($_POST['errors_to'], $error);
}
```

new hotness:

```php
# validate name/description, stripping potentially harmful data out
$e_name     = strip_tags($_POST['event_name']);
$e_descript = strip_tags($_POST['event_description']);
if (!($e_name && $e_descript)) {
    $error = 'Need an event name and description.';
    redirect_and_exit($_POST['errors_to'], $error);
}
```

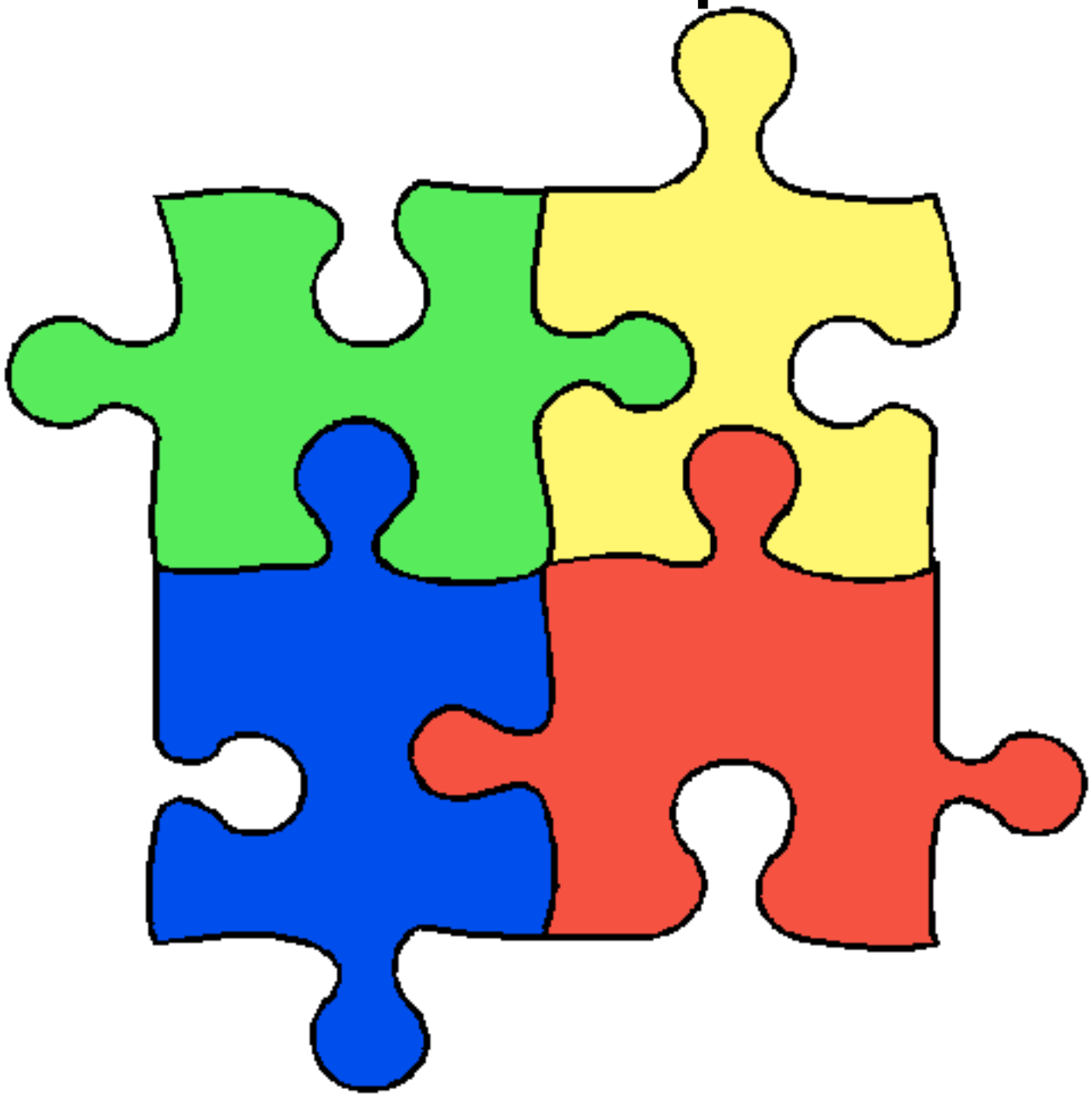
private/lib/form_functions.inc.php (old & busted)

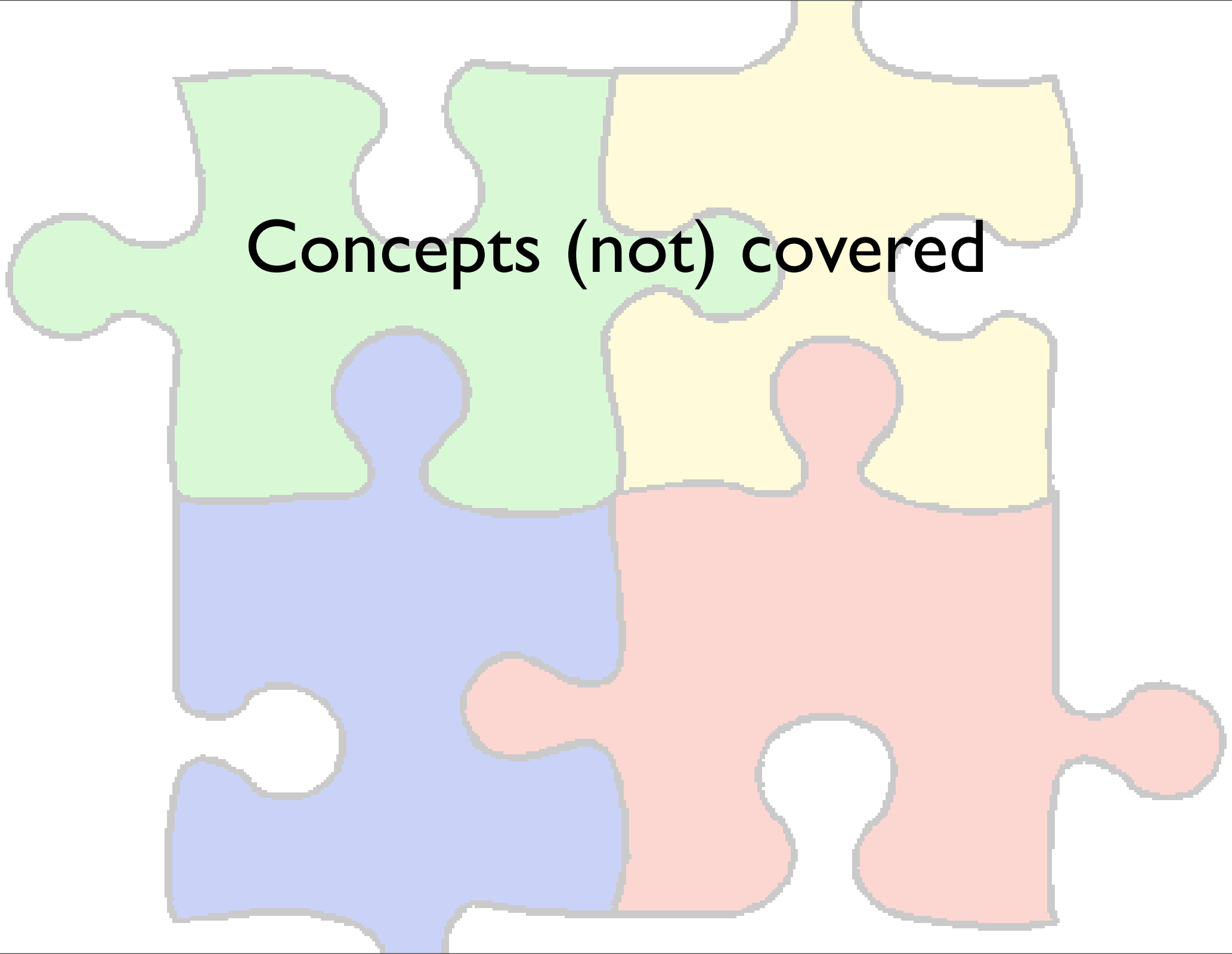
```
# insert into database
$insert = "
    INSERT
      INTO events (event_name, event_date, event_descript)
    VALUES ('$e_name', '$e_date', '$e_descript')
";
$result =& $dbh->query($insert);
```

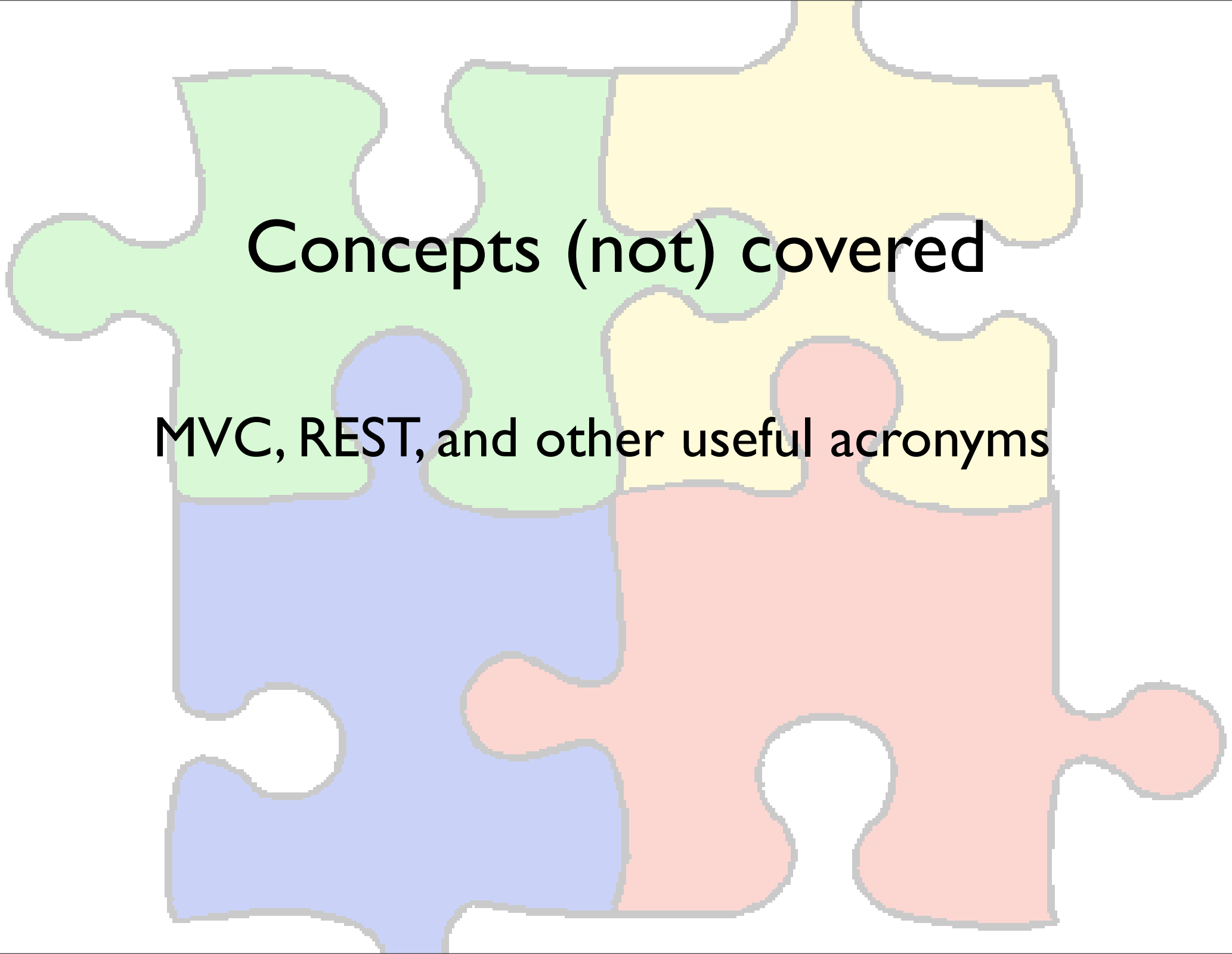
private/lib/form_functions.inc.php (new hotness)

```
# prepare the statement, letting MDB2 handle quotes
$sth =& $dbh->prepare("
    INSERT
      INTO events (event_name, event_date, event_descript)
    VALUES (:e_name, :e_date, :e_descript)
");
if (PEAR::isError($sth)) {
    $error = $sth->getMessage();
    redirect_and_exit($_POST['errors_to'], $error);
}

# run the query, handling errors appropriately
$result =& $sth->execute(array(
    'e_name'     => $e_name,
    'e_date'     => $e_date,
    'e_descript' => $e_descript,
));
```

Next steps

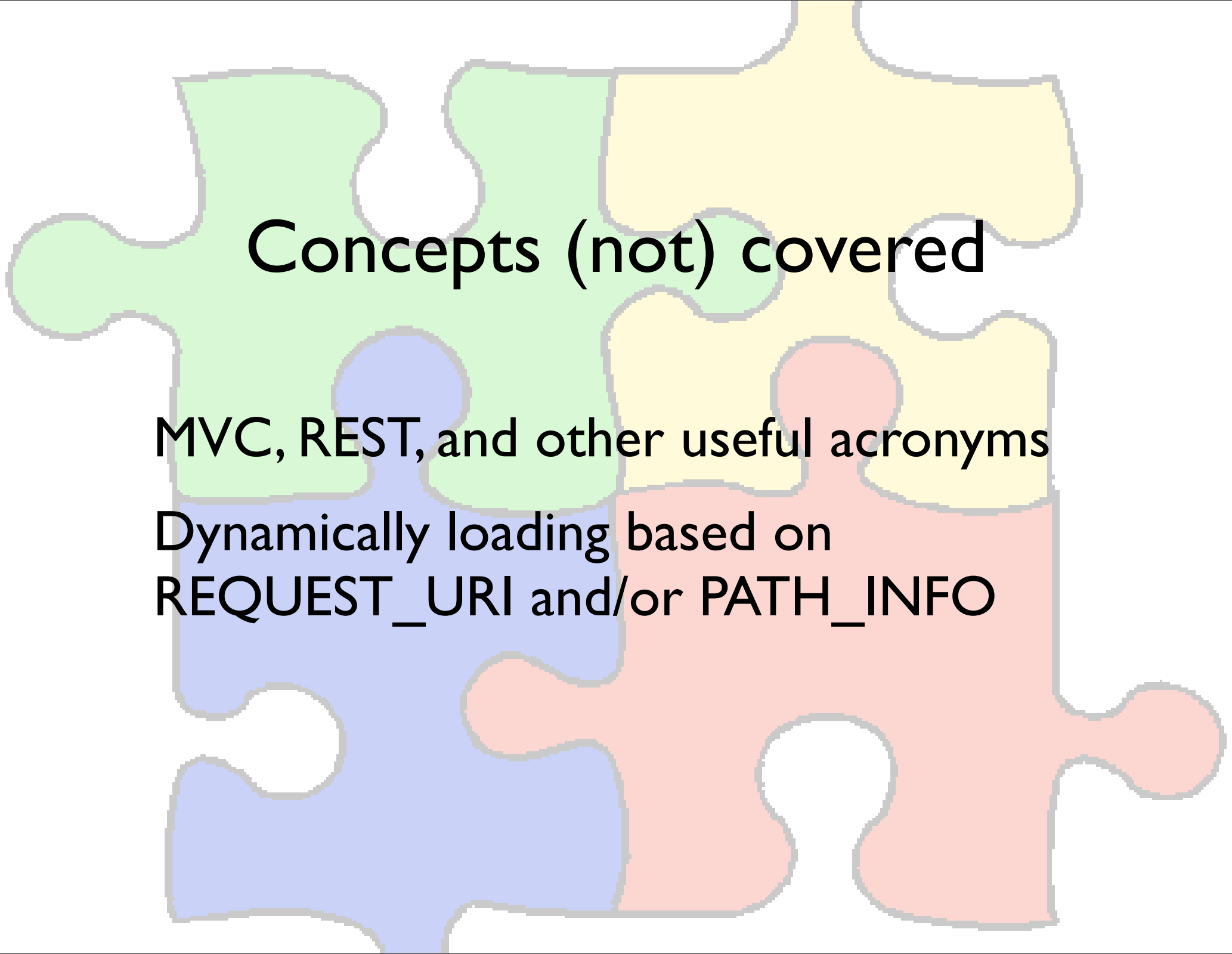



Concepts (not) covered



Concepts (not) covered

MVC, REST, and other useful acronyms



Concepts (not) covered

MVC, REST, and other useful acronyms

Dynamically loading based on REQUEST_URI and/or PATH_INFO

Concepts (not) covered

MVC, REST, and other useful acronyms

Dynamically loading based on REQUEST_URI and/or PATH_INFO

Live and development sites: one codebase

Advanced PHP

Peter Green and Joel Boonstra
Agathon Group
Originally presented at Gospelcon 2006

http://www.agathongroup.com/talks/gospelcon06/advancedphp/