

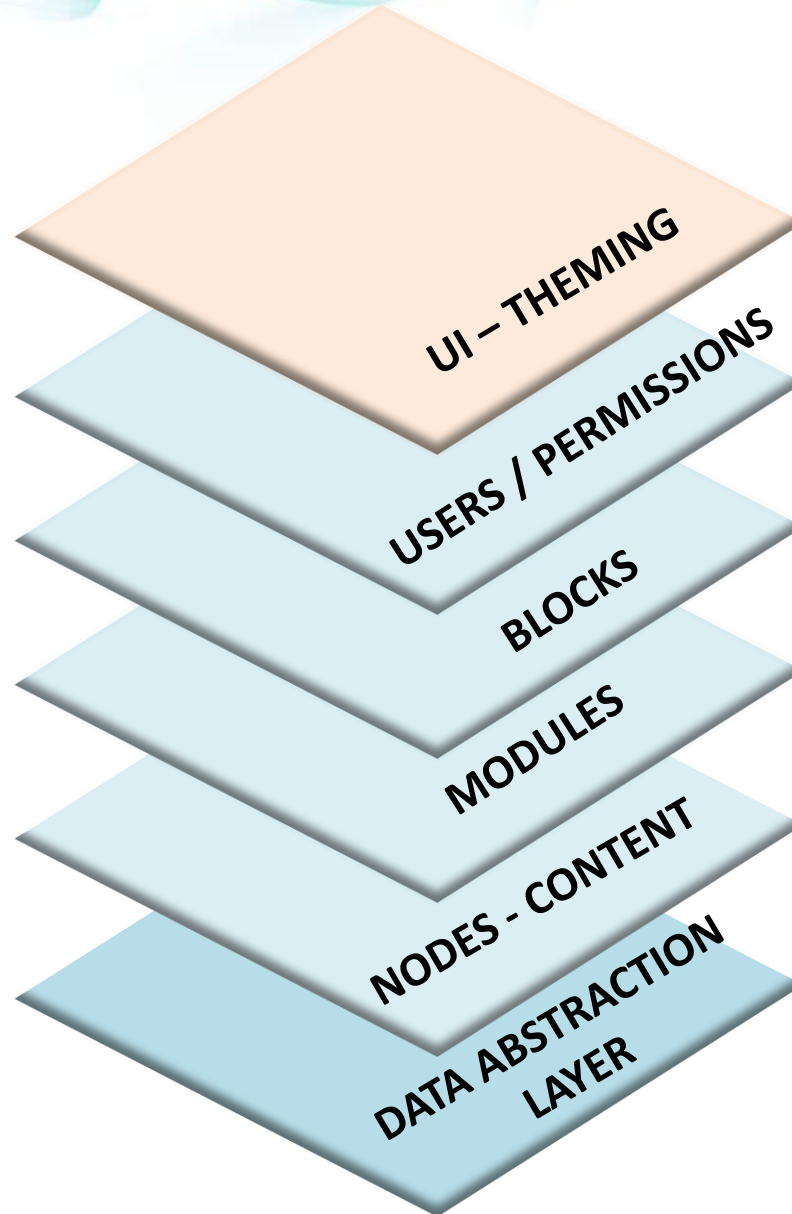
xurface ><

Desarrollo de módulos en Drupal e integración con dispositivos móviles

Luis Curo Salvatierra
CTO Xurface



DRUPAL ARCHITECTURE

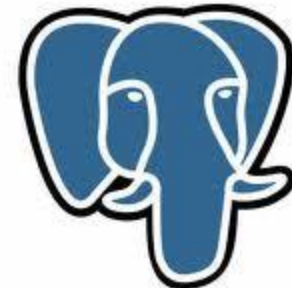


DATABASE ABSTRACTION LAYER

- Schema Definition
- Insert
- Update
- Delete



PostgreSQL



NODES

- Basic unit of information
- Content
- Object

Descriptive Label for Your Fund Name

Your story tells why you are championing this fund and should provide information about the people associated with this fund, such as the fund honoree. Writing a compelling story may help inspire people to donate to this cause. This story displays on your fund page, which is published and visible to people as soon as you create your fund (unless the organization requires a system administrator to approve your page first).

In the Champion Center, you can edit your story as well as add more sections to it and separate your sections with up to three headlines.

Guest Book

If you would like, you can add your name and a short message to our Guest Book. Thank you.

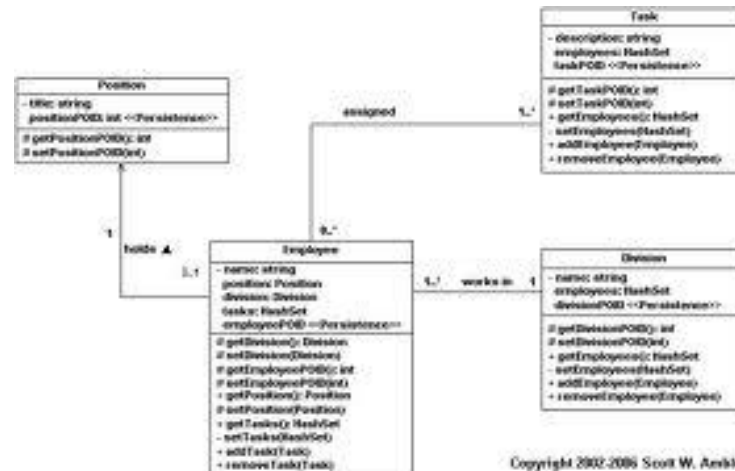
[Sign the Guest Book](#)

Pat Smith

[Tell a friend Pat Smith](#)
[Create a page for Pat](#)
[Bookmarks for Pat](#)

If you think this page contains objectionable content, please [notify the system administrator](#):
[Home](#) | [Help](#) | [Contact Us](#) | [Home](#)

[Enter Family Vitals](#)



MODULES

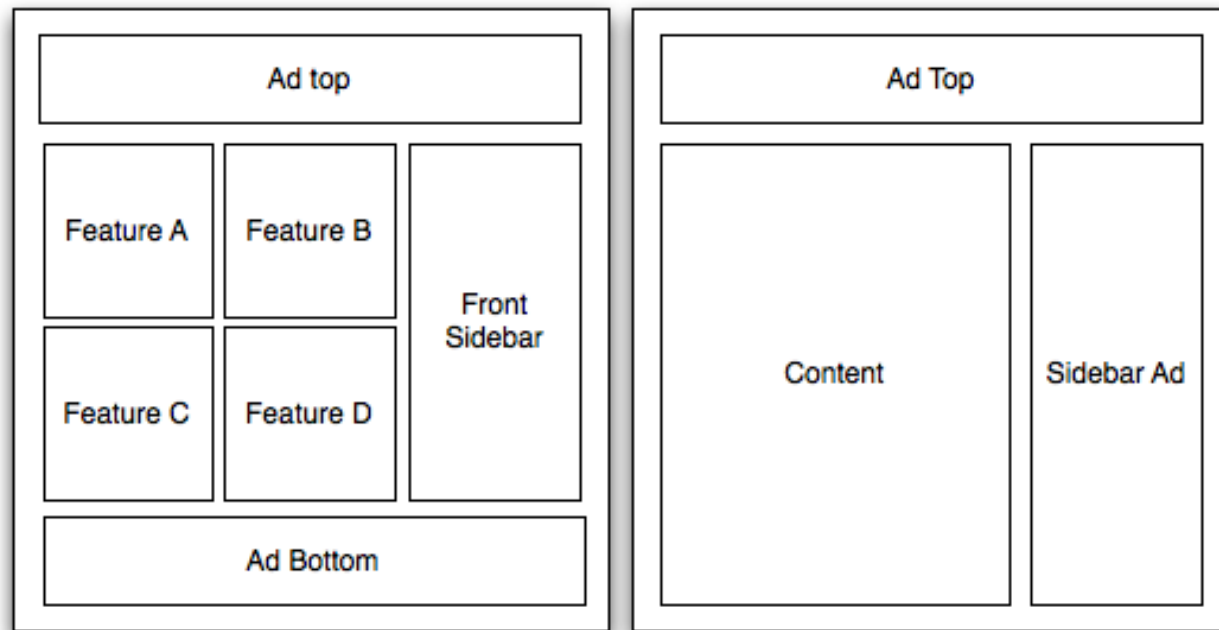
- Control of Drupal environment
- Control of Drupal Layers



**7, 587
modules
till today**

BLOCKS

- Piece of structural view



USER - PERMISSIONS

- Manage user authentication
- Manage user authorization
- Manage user privileges



UI - THEMING

- Manage design layouts
- Manage UI

HORIZON
Software Consultants Ltd.

Horizon's Sapphire
Jewelry Management Information System

Logon To Sapphire

Connector:

User Name:

Password:

LOGON RESET

Horizon POS
For Jewelry

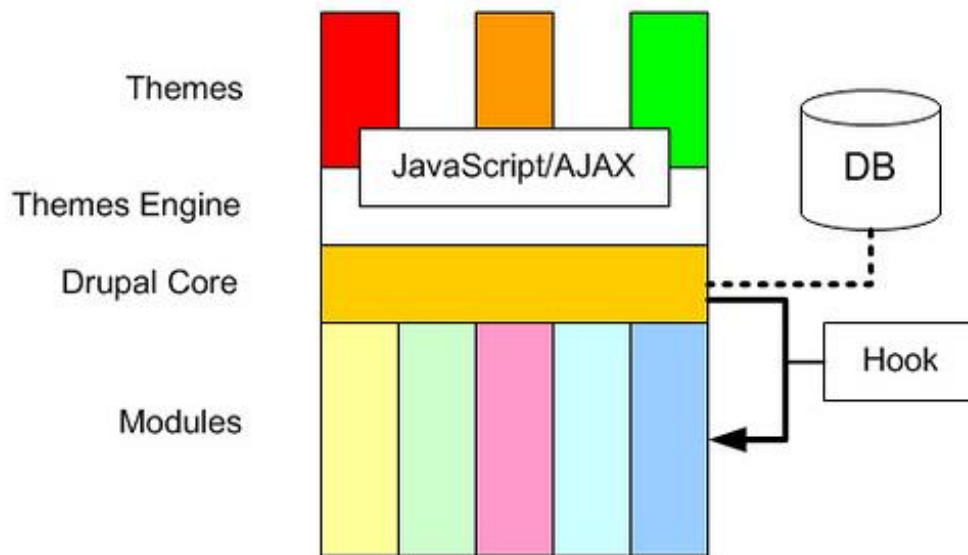
Coming Soon
Hotline: (86)020-34807750

Copyright © 2006-2008 Horizon Software Consultants Ltd. All rights reserved.

HORIZON

Hooks

- Way to interact with drupal layers



V. Hooks

5 57

6 80

7 267

<http://api.drupal.org/api/drupal/includes--module.inc/group/hooks>

Hooks samples

- `hook_block()`
- `hook_node_info()`
- `hook_form(&$node)`
- `hook_access ($op, $node, $account)`
- `hook_load ($node)`
- `hook_view($node, $teaser = FALSE, $page = FALSE)`

Working with Blocks



```
function MyCustomBlock_block($op='list', $delta=0, $edit=array())
{
    switch ($op)
    {
        case 'list':
            $blocks[0]['info'] = 'This is my custom Block';
            return $blocks;
        case 'view':
            $blocks['subject'] = 'My Block';
            $blocks['content'] = 'Hey My Block is Here!';
            return $blocks;
    }
}
```

Working with Nodes

```
function MyCustomNode_node_info()
{
    return array(
        'MyCustomNode' => array(
            'name' => 'Custom Node',
            'module' => 'MyCustomNode',
            'description' => 'This is my custom node',
            'has_title' => TRUE,
            'title_label' => 'Title of my node',
            'has_body' => TRUE,
            'body_label' => 'Body of my node',
        )
    );
}
```

Working with Forms

```
function MyCustomNode_form(&$node) {
    $type = node_get_types('type', $node);

    if ($type->has_title) {
        $form['title'] = array(
            '#type' => 'textfield',
            '#title' => $type->title_label,
            '#required' => TRUE,
            '#default_value' => $node->title,
            '#weight' => -5,
        );
    }

    if ($type->has_body) {
        $form['body_field'] = node_body_field(
            $node,
            $type->body_label,
            $type->min_word_count
        );
    }
}
```

Working with Forms

```
$form['code'] = array(  
    '#type' => 'textfield',  
    '#size' => 50,  
    '#maxlength' => 127,  
    '#title' => ('Code'),  
    '#description' => 'Code',  
    '#default_value' => isset($node->code) ? $node->code :  
    "",  
    );  
  
return $form;  
}
```

Working with Users - Permissions



```
function MyCustomNode_perm() {
    return array(
        'create MyCustom node',
        'edit MyCustom node',
        'delete MyCustom node',
    );
}
```

```
function MyCustomNode_access($op, $node, $account) {
    switch ($op) {
        case 'create':
            return user_access('create MyCustom node', $account);
        case 'update':
            return user_access('edit MyCustom node', $account);
        case 'delete':
            return user_access('delete MyCustom node', $account);
    }
}
```

Working with Data



```
function MyCustomNode_schema() {  
  
    $schema['MyCustomNode'] = array(  
        'description' => 'My Custom node table',  
        'fields' => array(  
            'vid' => array(  
                'type' => 'int',  
                'unsigned' => TRUE,  
                'not null' => TRUE,  
                'default' => 0,  
                'description' => 'version id',  
            ),  
        ),  
    ),  
}
```

Working with Data



```
'nid' => array(  
'type' => 'int',  
'unsigned' => TRUE,  
'not null' => TRUE,  
'default' => 0,  
'description' => 'node id'  
)
```

Working with Data



```
'code' => array(  
'description' => 'code',  
'type' => 'varchar',  
'length' => 127,  
'not null' => TRUE,  
'default' => 'varchar'  
)  
,),  
'primary key' => array(  
'vid', 'nid' ), );  
return $schema;  
}
```

Working with Data



```
function MyCustomNode_load($node) {
    $result = db_query(
        'SELECT code FROM {MyCustomNode}
        WHERE vid = %d',
        $node->vid
    );
    return db_fetch_object($result);
}
```

Working with Data



```
function MyCustomNode_insert($node) {
    if (!isset($node->life)) {
        $node->life = "";
    }
    if (!isset($node->works)) {
        $node->works = "";
    }
    db_query(
        'INSERT INTO {MyCustomNode} (vid, nid, code) '
        ."VALUES (%d, %d, '%s')",
        $node->vid,
        $node->nid,
        $node->code );
}
```

Working with Data

```
function MyCustomNode_update($node) {  
    if ($node->revision) {  
        MyCustomNode_insert($node); }  
    else {  
        db_query("UPDATE {MyCustomNode} "  
            ."SET code = '%s'"  
            ."WHERE vid = %d",  
            $node->code,  
            $node->vid  
            );  
    }  
}
```

Working with Data



```
function MyCustomNode_delete($node) {
    db_query( 'DELETE FROM {MyCustomNode}
    WHERE nid = %d',$node->nid
    );
}
function MyCustomNode_nodeapi(&$node, $op, $teaser, $page)
{
if ($op == 'delete revision') {
    db_query(
    'DELETE FROM {MyCustomNode}
    WHERE vid = %d',
    $node->vid );
}
}
```

Working with XML

- Drupal uses XML-RPC
- Define the XMP-RPC method
- Use `hook_xmlrpc()`



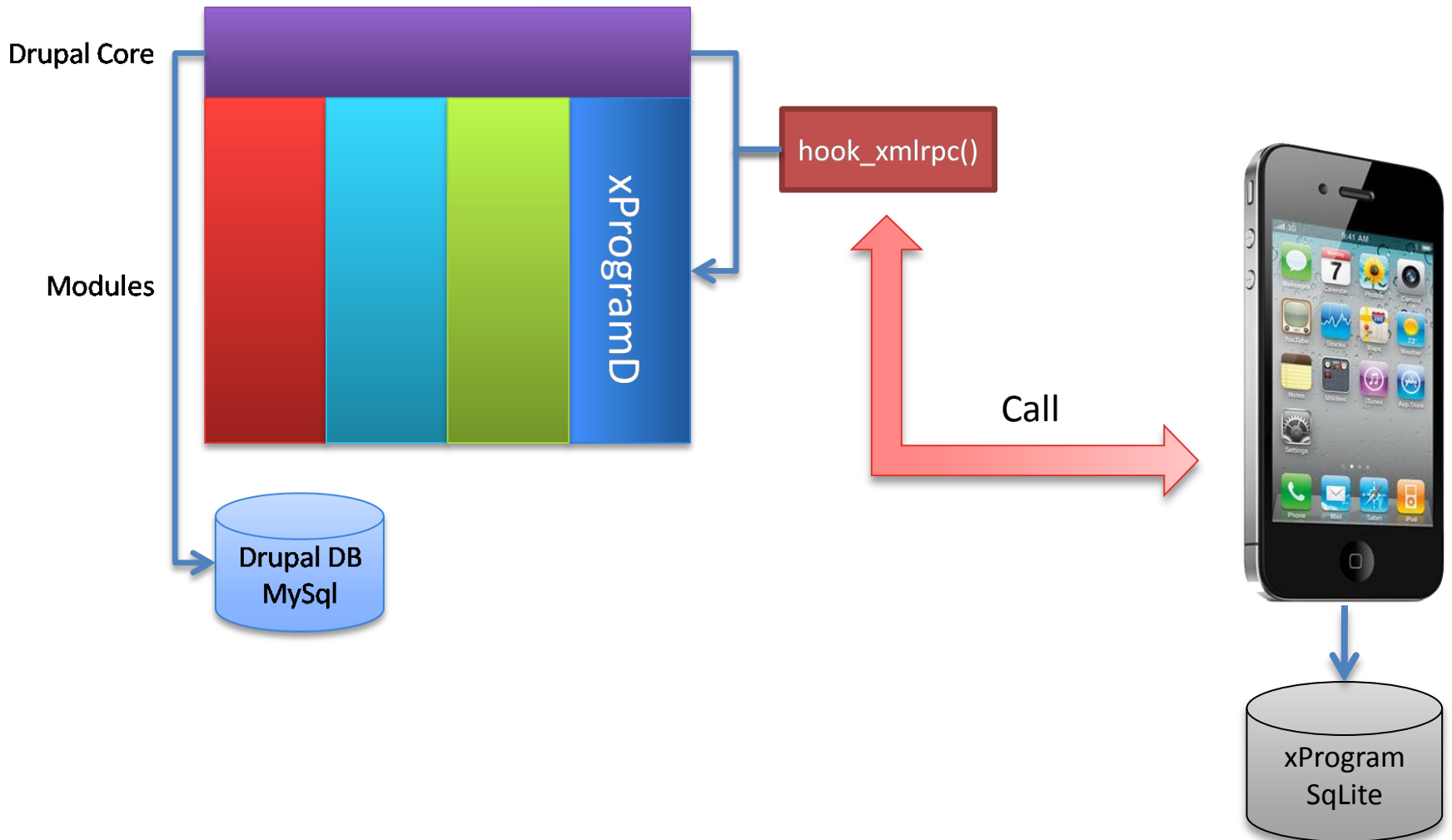
XML-RPC



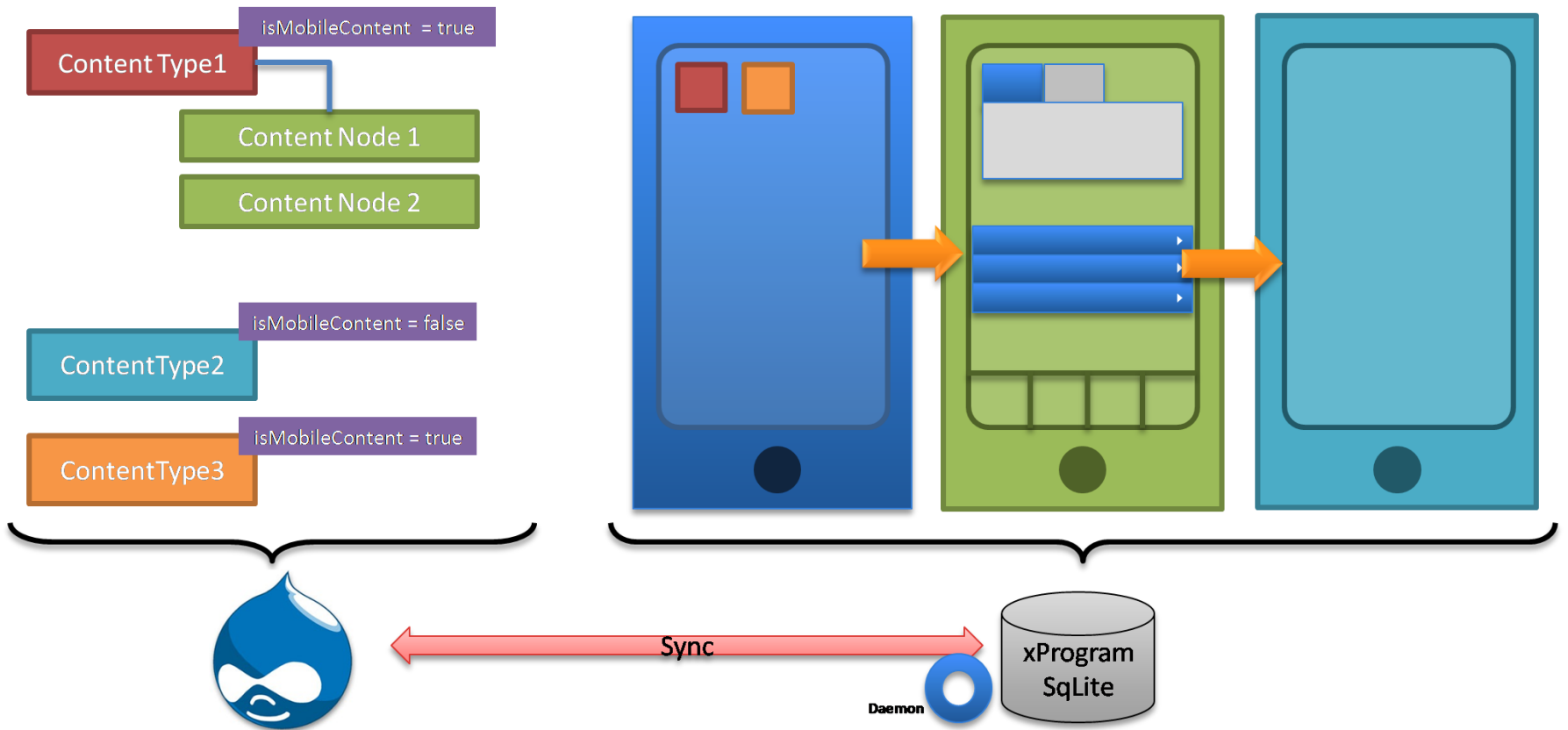
xProgramD ARCHITECTURE



XProgramD Architecture



XProgramD Architecture



Practical Case



Questions

Thank you