

Website Development Kit

Initial Training

Sascha Wildgrube
www.websitedevkit.com
Version 1.5 2010-04-02

Website Development Kit

Definition

Website Development Kit is a PHP5/mysql based framework for the fast creation of complex and secure web application systems

Website Development Kit

At the end of this presentation you will...

- ...know the benefits of applying WDK
- ...understand the architecture approach
- ...be able to design, plan, and implement application systems using a large diversified team
- ...be capable of using the predefined building blocks
- ...want to use it on a daily basis

Website Development Kit

Audience

- **Business Stakeholders**
- Architects and Project Managers
- Developers

Website Development Kit

Principles

- Service Oriented Architecture
- Independency of Logic, Layout, Text, and Data
- Test-Driven Development
- Cross Application Code Transfer and Re-use
- Support Segregation of Duties

Website Development Kit

Reason Why

- More efficient development process
 - Reduced lead time
 - Reduced efforts
- Quality improvement

Website Development Kit

Nice try but...

- WDK based application projects are...
 - ...faster because larger teams can work simultaneously on the same application
 - ...less expensive because of a built-in re-use strategy
 - ...more robust due to the test driven development approach
 - ...easier and less risky to plan because of a clear architecture approach and predefined building blocks

Website Development Kit

At the end of this presentation you will...

- ...know the benefits of applying WDK ✓
- ...understand the architecture approach
- ...be able to design, plan, and implement application systems using a large diversified team
- ...be capable of using the predefined building blocks
- ...want to use it on a daily basis

Website Development Kit

Audience

- Business Stakeholders
- Architects and Project Managers
- Developers

Website Development Kit

Agenda for Architects and Project Managers

- Key Questions
- Service Oriented Architecture
- WDK Components
- Planning a WDK based Application Project
- Test-Driven Development and Operations

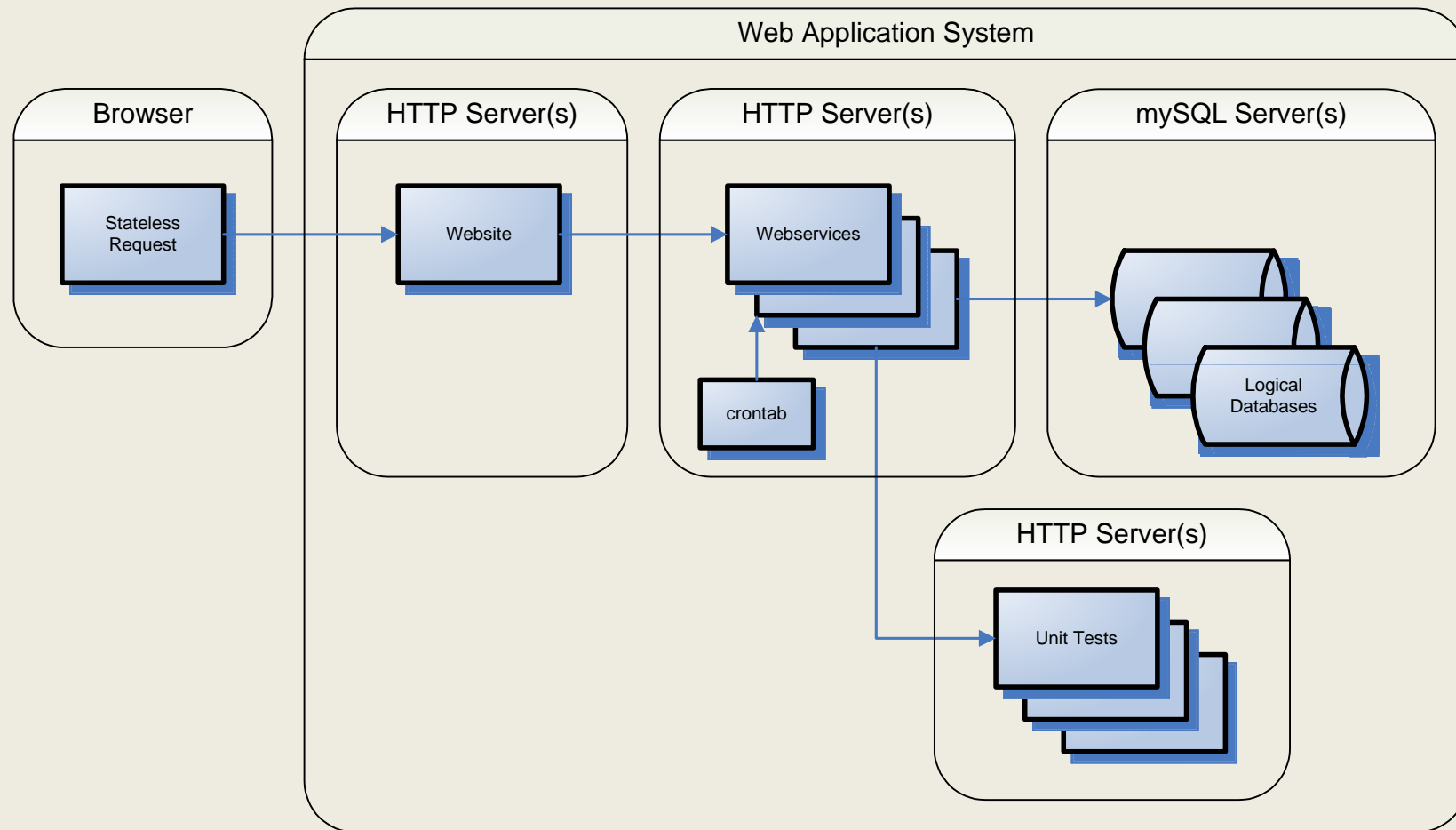
Website Development Kit

Answers to key questions

- How to design an application system?
- How to define work packages?
- How to develop logic and layout independently?
- How to ensure quality?
- How to manage local text?
- How to encapsulate the database?

Website Development Kit


Service Oriented Architecture



Website Development Kit

WDK Components

2009-10-13 22:24:39 (UTC) [Login](#) 

WDK 

Website Development Kit

Home | References | Maintenance | Log-in | Sign-in | Experimental

→ [Home](#)
→ [References](#)
→ [Maintenance](#)
→ **Log-in**
→ [Sign-in](#)
→ [Experimental](#)


You are here: [Log-in](#)

Log-in

User name:

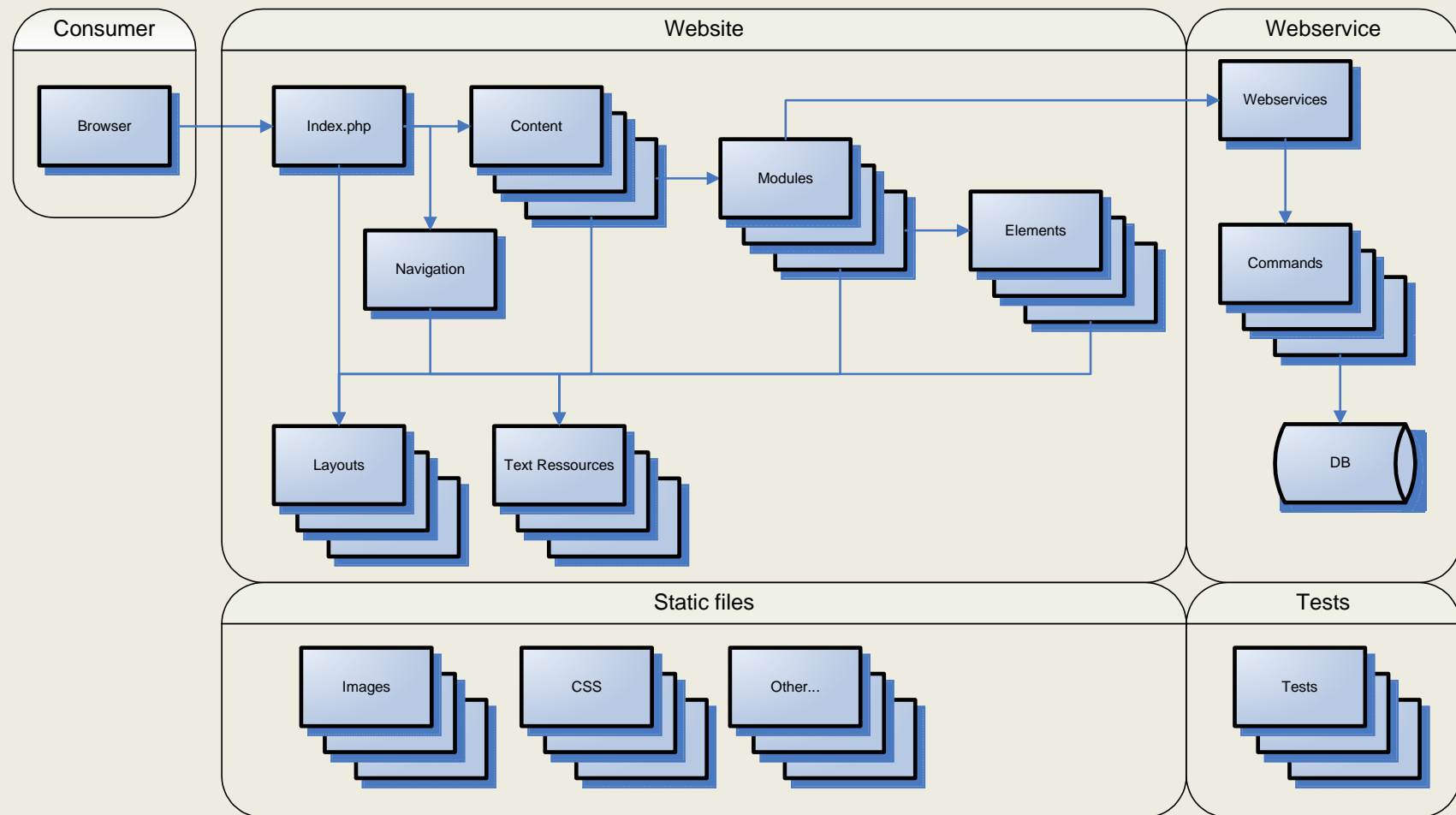
Password:

[Imprint](#) [Contact](#) [License](#) [Options](#) [Validate](#)

Copyright © 2007-2009 by Sascha Wildgrube 

Website Development Kit

WDK Components



Website Development Kit

WDK Components

2009-10-13 22:24:39 (UTC) [Login](#)

WDK

Website Development Kit

[Home](#) [References](#) [Maintenance](#) [Log-in](#) [Sign-in](#) [Experimental](#)

[→ Home](#)
[→ References](#)
[→ Maintenance](#)
[→ **Log-in**](#)
[→ Sign-in](#)
[→ Experimental](#)

You are here: [Log-in](#)

Log-in

User name:

Password:

[Log-in](#)

[Imprint](#) [Contact](#) [License](#) [Options](#) [Validate](#)

Copyright © 2007-2009 by Sascha Wildgrube

Website Development Kit

At the end of this presentation you will...

- ...know the benefits of applying WDK ✓
- ...understand the architecture approach ✓
- ...be able to design, plan, and implement application systems using a large diversified team
- ...be capable of using the predefined building blocks
- ...want to use it on a daily basis.

Website Development Kit

Planning a WDK based Application Project

- Requirements
- Quality Criteria
 - Define Tests
- Feature List
 - Create a Product Breakdown Structure
- Appearance
 - Define Layout and Text

Website Development Kit

Product Breakdown Structure - Application System

- Application System
 - Website and setup
 - Navigation
 - Content
 - Framework Layout
 - Features
 - Feature A
 - Feature B
 - Feature C...
 - Installer

Website Development Kit

Product Breakdown Structure - Feature

- Feature
 - Tests
 - Modules
 - Module A
 - Module B...
 - Elements
 - Element A
 - Element B...
 - Webservices
 - Webservice A
 - Webservice B...

Website Development Kit

Product Breakdown Structure – Modules and Elements

- Module
 - Tests
 - States
 - State A
 - Layout
 - State B...
 - Layout
 - Text
 - Translations
- Element
 - Implementation
 - Layout
 - Tests

Website Development Kit

Product Breakdown Structure – Webservices

- Webservice
 - Tests
 - Database Model
 - Commands
 - Command A
 - Command B...

Website Development Kit

Exercise

A simple wiki...

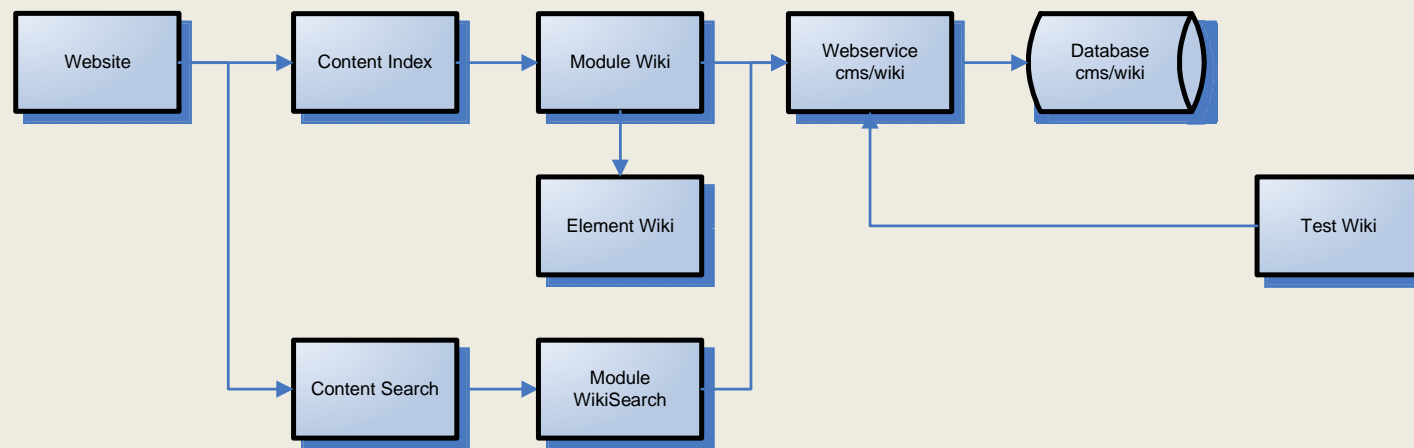
Website Development Kit

Exercise

- Requirements
 - Users can create and edit wiki articles
 - Users can search for articles

Website Development Kit

Exercise



Website Development Kit

Exercise

- Wiki Website – Product Breakdown Structure

- Navigation
- Content
 - Index
 - UserSettings
- Framework Layout
- Features
 - “Wiki”
 - Tests
 - Modules
 - » Wiki
 - State View
 - State Edit
 - State Save
 - » WikiSearch
 - State Form
 - State Search
 - Elements
 - » Wiki
 - [...]
 - Webservices
 - » cms/wiki
 - Test
 - Database Model
 - Command
 - Get
 - Set
 - Search
- Installer

Website Development Kit

Test-Driven Development

- Define Feature
- Write Test
- Develop against Test until OK
- If necessary extend or adapt test
- A Quality Review validates the Test Results

Website Development Kit

Test-Driven Operations

- Functional Automated Tests being triggered on a permanent basis
- Regressions in new versions will cause existing tests to fail
- Warnings can be produced

Website Development Kit

At the end of this presentation you will...

- ...know the benefits of applying WDK ✓
- ...understand the architecture approach ✓
- ...be able to design, plan, and implement application systems using a large diversified team ✓
- ...be capable of using the predefined building blocks
- ...want to use it on a daily basis

Website Development Kit

Audiences

- Business Stakeholders
- Architects and Project Managers
- **Developers**

Website Development Kit

Agenda for Developers

- Features
- Frontend Components
- Backend Components
- Code Organization
- Debugging
- Security
- Coding Guidelines

Website Development Kit

Features

- HTTP Response Frameworks
- Input Parameter Control
- Input Validation
- Configuration Reader
- Template Engine / Renderer
- Testing Framework
- Framework for Webservices and Database Access Encapsulation
- Date / Time / Locale / Currency Management
- Debugging / Profiling
- Error Handling
- Encryption / Random Data Generation / Password Policy Management
- Emailing
- Advanced String Operations / Regular Expressions / Array Functions / UTF8 support
- List Searching, Sorting, and Filtering
- File System / File IO Functions / Shell Functions
- Character separated file handling
- Event Dispatching Engine
- XML Functions

Website Development Kit

Frontend Components

- Website
- Navigation
- Content
- Modules
- Elements
- Layout
- Resources

Website Development Kit

Backend Components

- Webservices
- Tests
- Installer

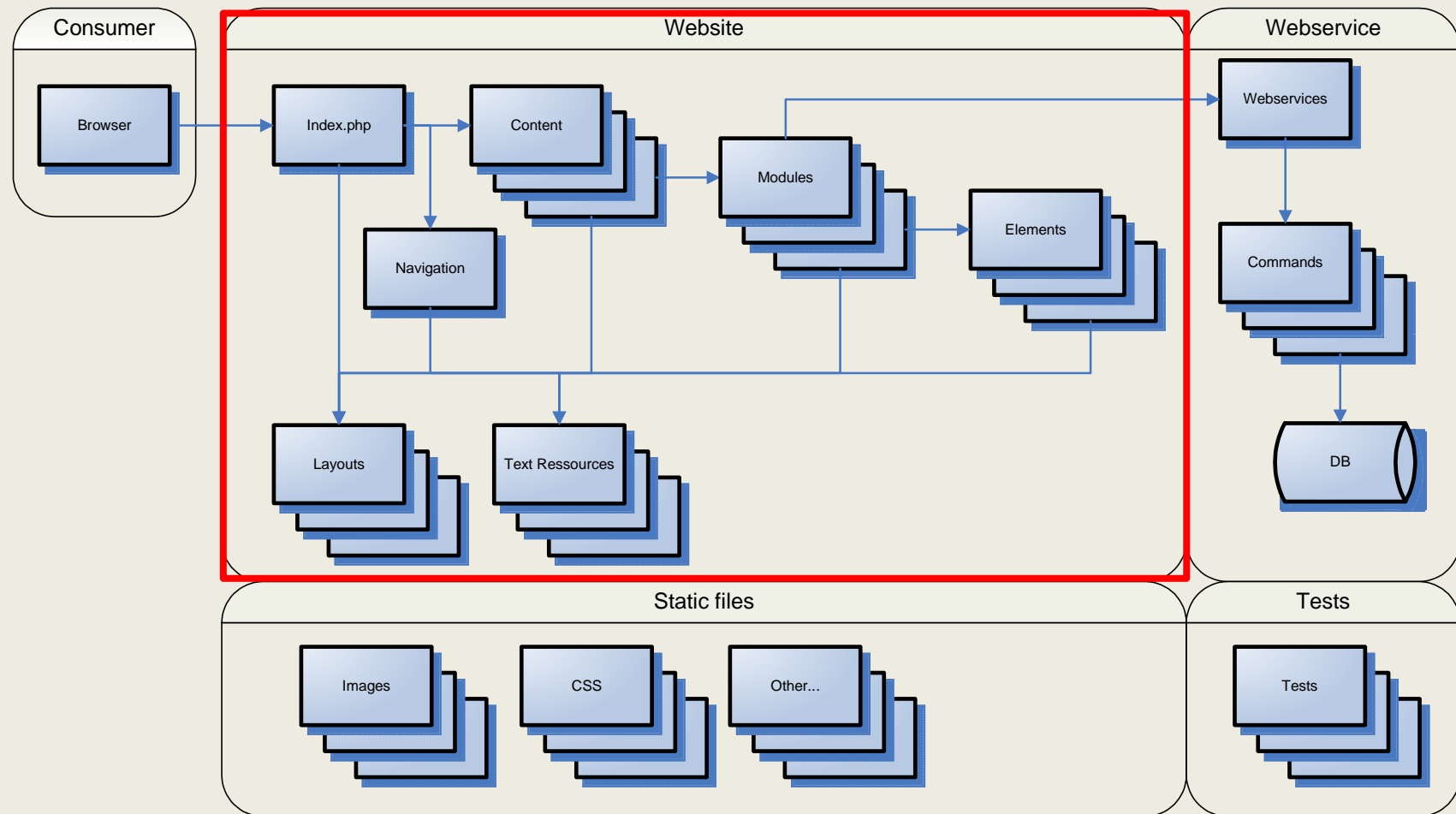
Website Development Kit

Code Organization

- Assemblies
- Assembly Line

Website Development Kit

Frontend Components / Website



Website Development Kit

Frontend Components / Website

- **Website**
 - Is a class derived from CWebApplication
 - The central dispatcher
 - Loads navigation trees and content
 - Manages parameters and session states

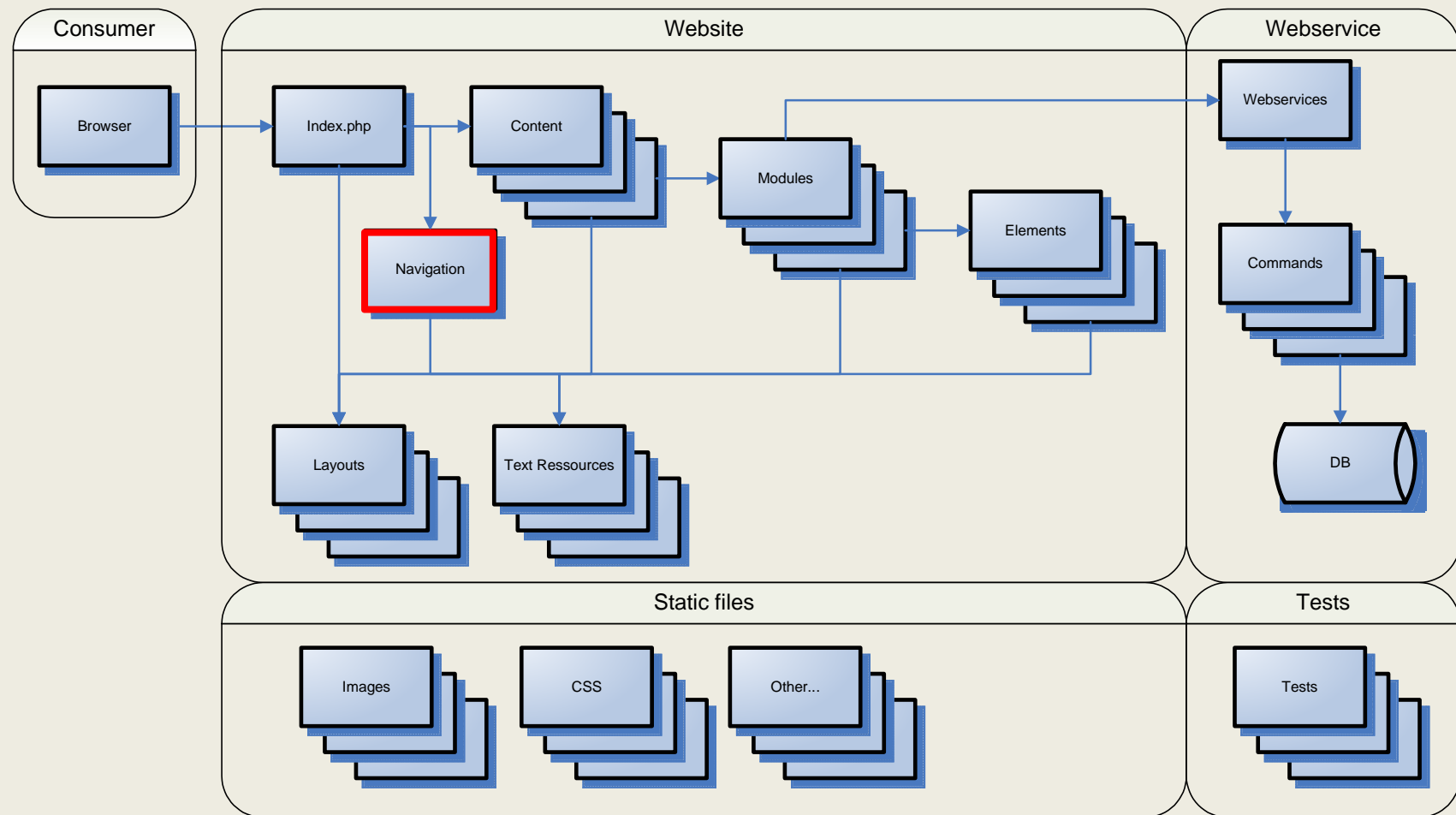
Website Development Kit

Frontend Components / Website

- Callback Functions
 - CallbackInitAssemblyLine()
 - CallbackInit()
 - CallbackCommand()
 - CallbackUserLogin()
 - CallbackPreContent()
 - CallbackPostContent()
 - CallbackGetDateLocalFormat()
 - Many more...

Website Development Kit

Frontend Components / Navigation



Website Development Kit

Frontend Components / Navigation

- Navigation
 - Directory tree
 - Defined by CSV files
 - Can be rendered by the navigation element
 - Multiple navigation trees can be merged into one

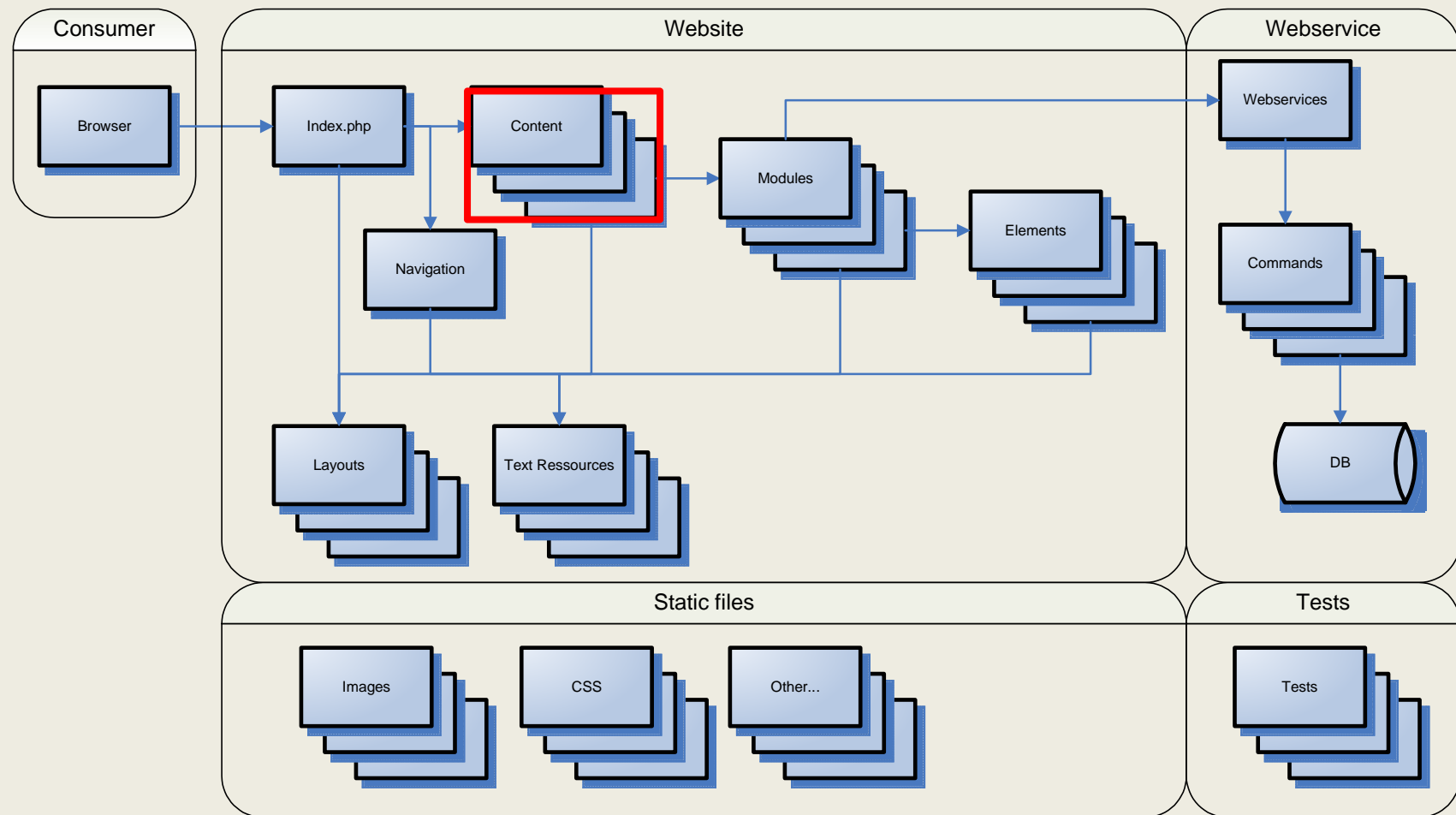
Website Development Kit

Frontend Components / Navigation / File Example

LEVEL	;CONTEXT	;LABEL	;CONDITIONS
0	;index	;?TID_NAVIGATION_HOME?	;
0	;references	;?TID_NAVIGATION_REFERENCES?	;
1	;reference-a	;?TID_NAVIGATION_REFERENCE-A?	;
1	;reference-b	;?TID_NAVIGATION_REFERENCE-B?	;
2	;reference-b-moreinfo	;?TID_NAVIGATION_REFERENCE-B-MOREINFO?	;
0	;features	;?TID_NAVIGATION_FEATURES?	;
0	;include:maintenance;		;login
0	;login	;?TID_NAVIGATION_LOGIN?	;nlogin
0	;signin	;?TID_NAVIGATION_SIGNIN?	;nlogin
0	;usersettings	;?TID_NAVIGATION_USERSETTINGS?	;login

Website Development Kit

Frontend Components / Content



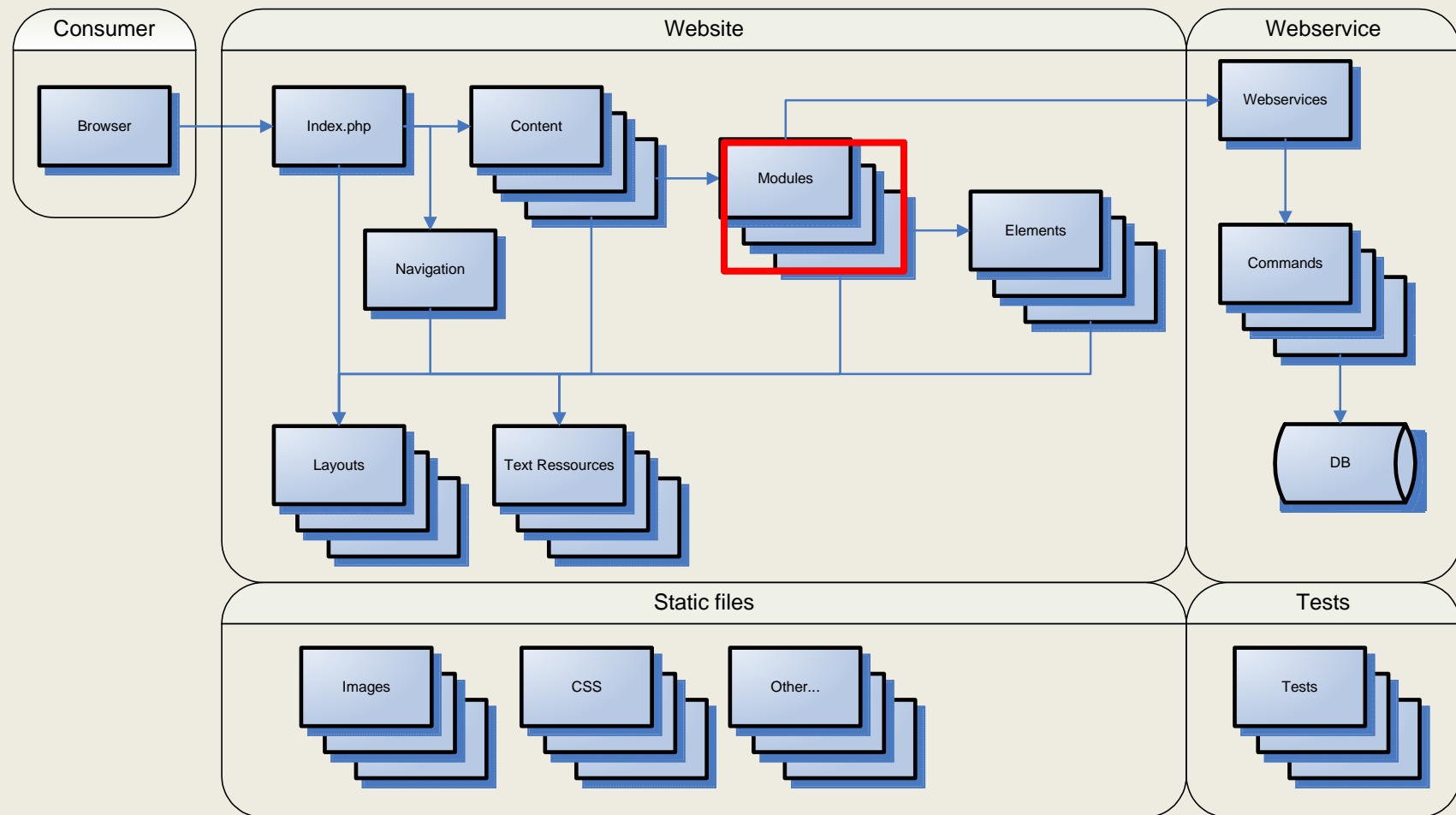
Website Development Kit

Frontend Components / Content

- Content
 - Is a class derived from CWebsiteSatellite
 - Is closest to the traditional „pages“ of a website
 - Can be language independent
 - Arranges functional modules
 - Is selected by the website instance

Website Development Kit

Frontend Components / Modules



Website Development Kit

Frontend Components / Modules

- Modules
 - A class Derived from CModule (which inherits CWebsiteSatellite)
 - Represent functionality and frontend logic
 - Are layout independent
 - Communicate to webservice
 - Go from one „state“ to another
 - Examples
 - CLoginModule
 - CTestManagerModule
 - CFeedbackModule



Website Development Kit

Frontend Components / Modules

```
Class CContent extends CContentDefault
{
    function CallbackRenderContent(&$arrayItems)
    {
        $this->GetWebsite()->IncludeModule("crm/feedback");
        $feedback = new CFeedbackModule($this->GetWebsite(),"");
        $arrayItems["MODULE_FEEDBACK"] = $feedback->RenderModule();
    }
}
```

Website Development Kit

Frontend Components / Modules

2009-10-19 21:39:15 (UTC) [Login](#) 

Website Development Kit

Home | References | Maintenance | Log-in | Sign-in | Experimental | Feedback

[→ Home](#)
[→ References](#)
[→ Maintenance](#)
[→ Log-in](#)
[→ Sign-in](#)
[→ Experimental](#)
[→ Feedback](#)

You are here: [Feedback](#)

Feedback

Please provide your feedback on this website! Tell us what you like and what you miss!

Your name:

Your email address:

Your Feedback:

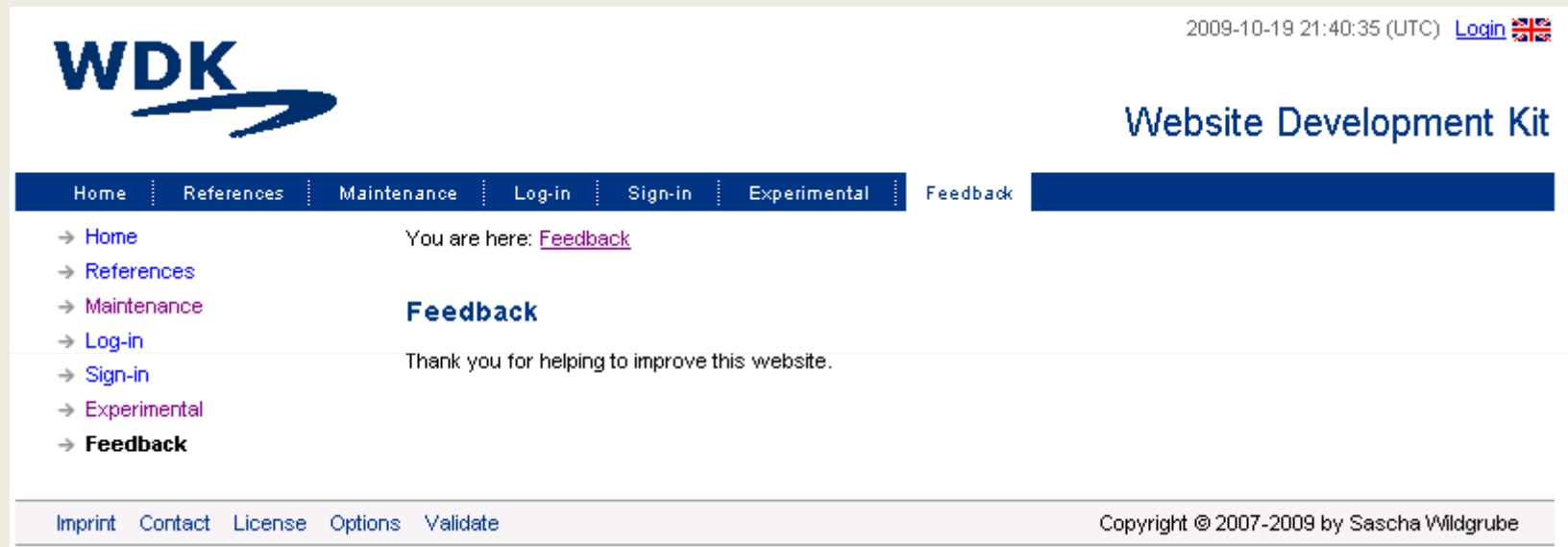
Send Feedback

[Imprint](#) [Contact](#) [License](#) [Options](#) [Validate](#)

Copyright © 2007-2009 by Sascha Wildgrube


Website Development Kit

Frontend Components / Modules



The screenshot displays the Website Development Kit (WDK) frontend. At the top left is the WDK logo, and at the top right is the date and time '2009-10-19 21:40:35 (UTC)' followed by a 'Login' link and a flag icon. Below this is a dark blue navigation bar with links: Home, References, Maintenance, Log-in, Sign-in, Experimental, and Feedback. The main content area shows a breadcrumb trail 'You are here: Feedback' and a list of links with arrows: Home, References, Maintenance, Log-in, Sign-in, Experimental, and Feedback. The title 'Feedback' is prominently displayed, followed by the text 'Thank you for helping to improve this website.' At the bottom, there is a footer with links: Imprint, Contact, License, Options, and Validate, and a copyright notice: 'Copyright © 2007-2009 by Sascha Wildgrube'.

WDK

2009-10-19 21:40:35 (UTC) [Login](#) 

Website Development Kit

Home References Maintenance Log-in Sign-in Experimental Feedback

→ Home You are here: [Feedback](#)

→ References

→ Maintenance

→ Log-in

→ Sign-in

→ Experimental

→ **Feedback**

Feedback

Thank you for helping to improve this website.

Imprint Contact License Options Validate

Copyright © 2007-2009 by Sascha Wildgrube

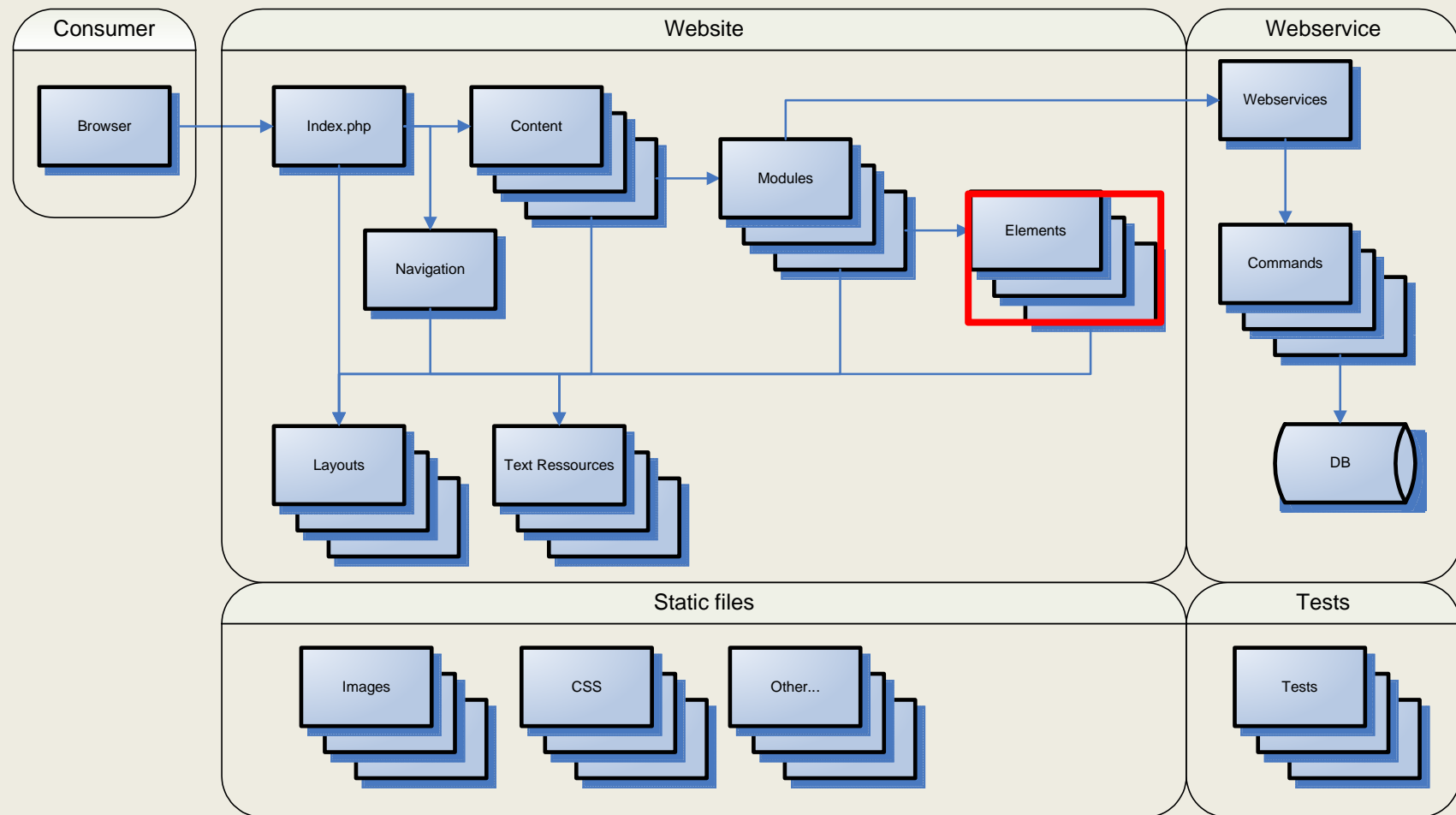
Website Development Kit

Frontend Components / Modules

- Singletons vs. Individual Instances
- Instance IDs

Website Development Kit

Frontend Components / Elements



Website Development Kit

Frontend Components / Elements

- Elements
 - Produce HTML output
 - Do not contain functional logic
 - Are language independent
 - Examples
 - CTableElement
 - CNavigationElement
 - CListElement
 - CLinkElement

Website Development Kit

Frontend Components / Elements

Code Example:

```
$icon = new CIconElement($this->GetWebsite(), "default");  
$strOutput = $icon->RenderIcon("link", "An icon without a link");
```

Layout:

```

```

Output:

```

```

Website Development Kit

Frontend Components / Elements

Styles

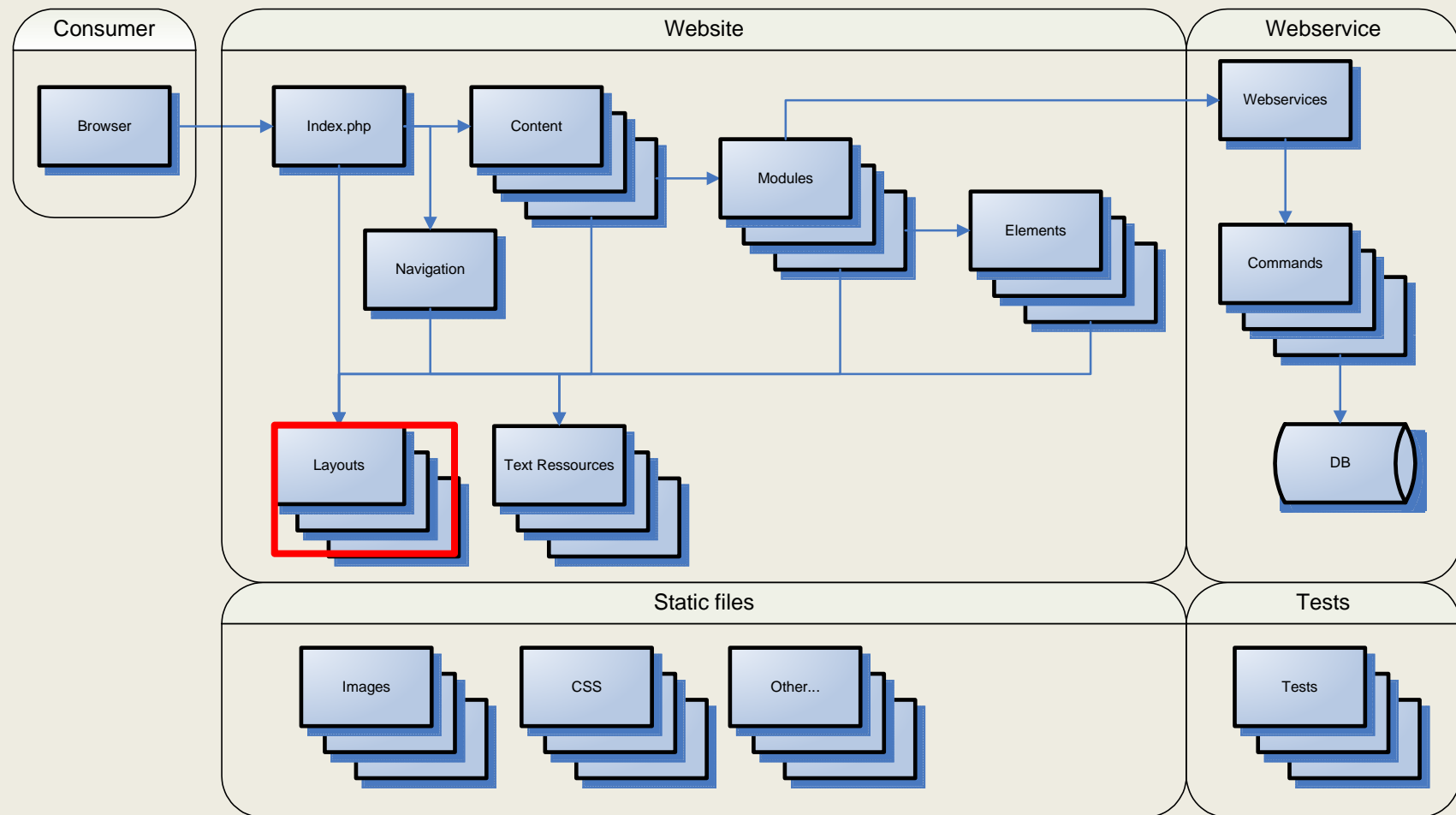
```
$format_red = new CFormatElement($this->GetWebsite(),"red");  
$format_strong = new CFormatElement($this->GetWebsite(),"strong");  
$text = new CTextElement($this->GetWebsite(),"default");  
$strOutput = $text->RenderText(  
    "A Header with some ".$format_red->RenderFormat("red colored")." words",  
    "And this is some text containing ".$format_strong->RenderFormat("significantly emphasized")." words. ");
```

A Header with some **red colored** words

And this is some text containing **significantly emphasized** words.

Website Development Kit

Frontend Components / Layout



Website Development Kit

Frontend Components / Layout

- Layouts
 - Are defined in layout definition files (txt and htm)
 - Define the optical appearance of elements, modules, forms, and website frameworks
 - Can be developed separately from the logic part

Website Development Kit

Frontend Components / Layout / File Name Convention

- layout_<layoutid>_<type>[_<subjectid>][_<stateid>][_<styleid>].(txt|htm)
- layoutid is the name of the layout (e.g. „default“ or „newbranding2010“)
- Types
 - framework
 - module
 - element
 - form
- Subject ID is the id of a module, element, or form
- State Ids are specific to modules
- Style ID defines a variant of an optical representation (e.g. big, small, etc)
- Examples
 - layout_default_framework.htm
 - layout_default_element_table.txt
 - layout_newbranding2010_module_login_loginform_big.htm
 - layout_newbranding2010_module_login_loginform_small.htm
 - layout_default_element_link_default.htm
 - layout_default_element_link_external.htm

Website Development Kit

Frontend Components / Layout / File Examples

- layout_default_element_link_default.htm

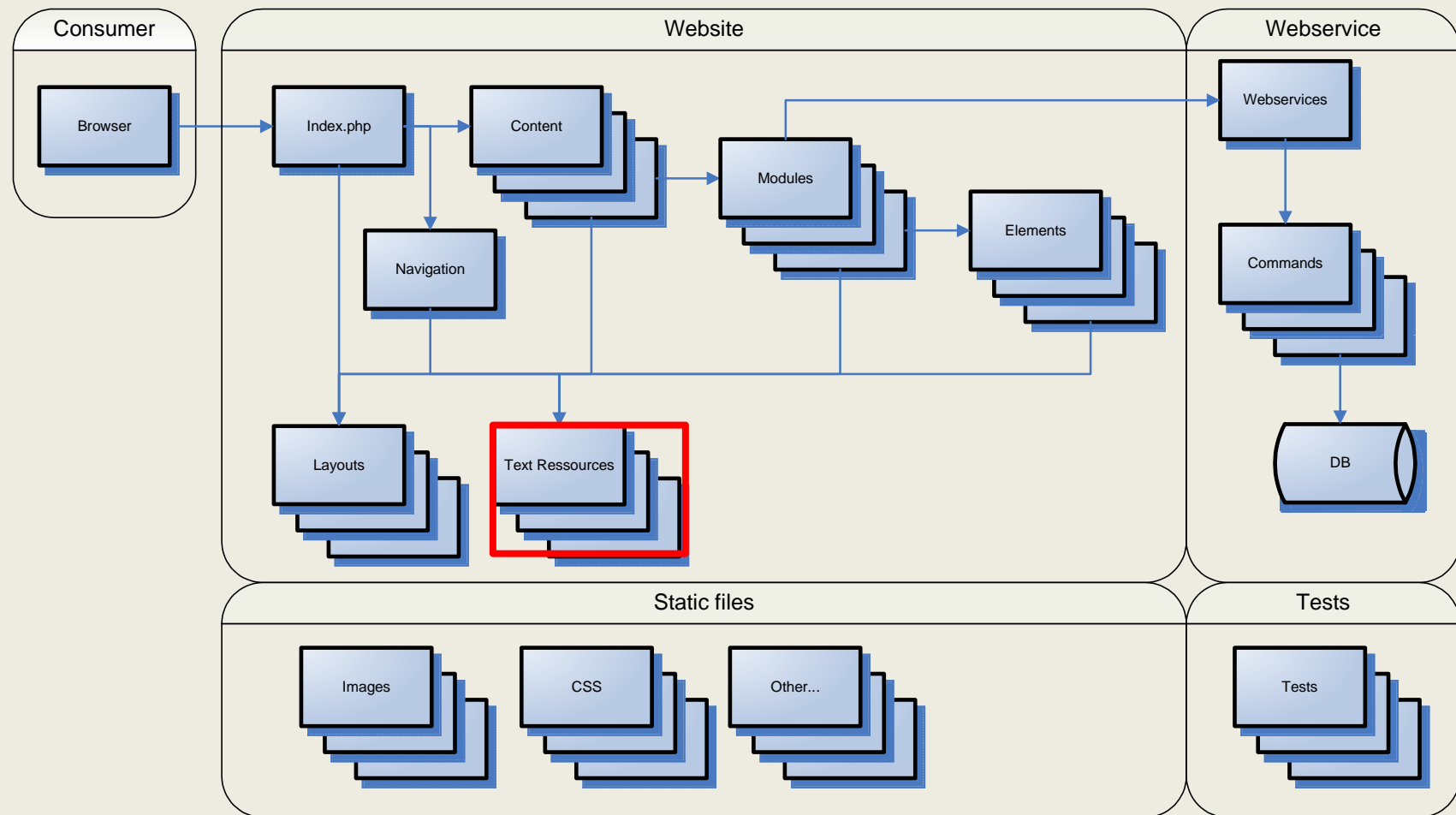
```
<a href="{URL}">{TEXT}</a>
```

- layout_default_element_table.txt

```
LAYOUT_TABLE_BEGIN
<table cellpadding="2" cellspacing="2" width="100%"><tbody>
LAYOUT_TABLE_END
</tbody></table>
LAYOUT_TABLE_ROW_BEGIN
<tr>
LAYOUT_TABLE_ROW_END
</tr>
LAYOUT_TABLE_DATACELL_BEGIN
<td class="data">
LAYOUT_TABLE_DATACELL_END
</td>
LAYOUT_TABLE_HEADERCELL_BEGIN
<th class="data">
LAYOUT_TABLE_HEADERCELL_END
</th>
```


Website Development Kit

Frontend Components / Ressources



Website Development Kit

Frontend Components / Ressources

- Ressources
 - Contain localized text
 - Are identified by text identifiers (?TID_...?)
 - Can be global or module specific
 - Each language is in a separate file

Website Development Kit

Frontend Components / Ressources / TID Naming Convention

- TID_[<MODULE>_]<TYPE>_NAME
- Types
 - TITLE (e.g. TID_TITLE_MAIN)
 - TEXT (e.g. TID_TEXT_DISCLAIMER)
 - NAVIGATION (e.g. TID_NAVIGATION_ABOUT)
 - TABLEHEADER
(e.g. TID_SOMEMODULE_TABLEHEADER_DATETIME)
 - ERROR/WARNING/INFO
(e.g. TID_ERROR_WRONGPASSWORD)
 - LINK (e.g. TID_LINK_BACK)
 - BUTTON (e.g. TID_BUTTON_NEXT)
 - LABEL (e.g. TID_SOMEMODULE_LABEL_NAME)
 - CHECKBOX
(e.g. TID_DISCLAIMER_CHECKBOX_CONFIRMDISCLAIMER)

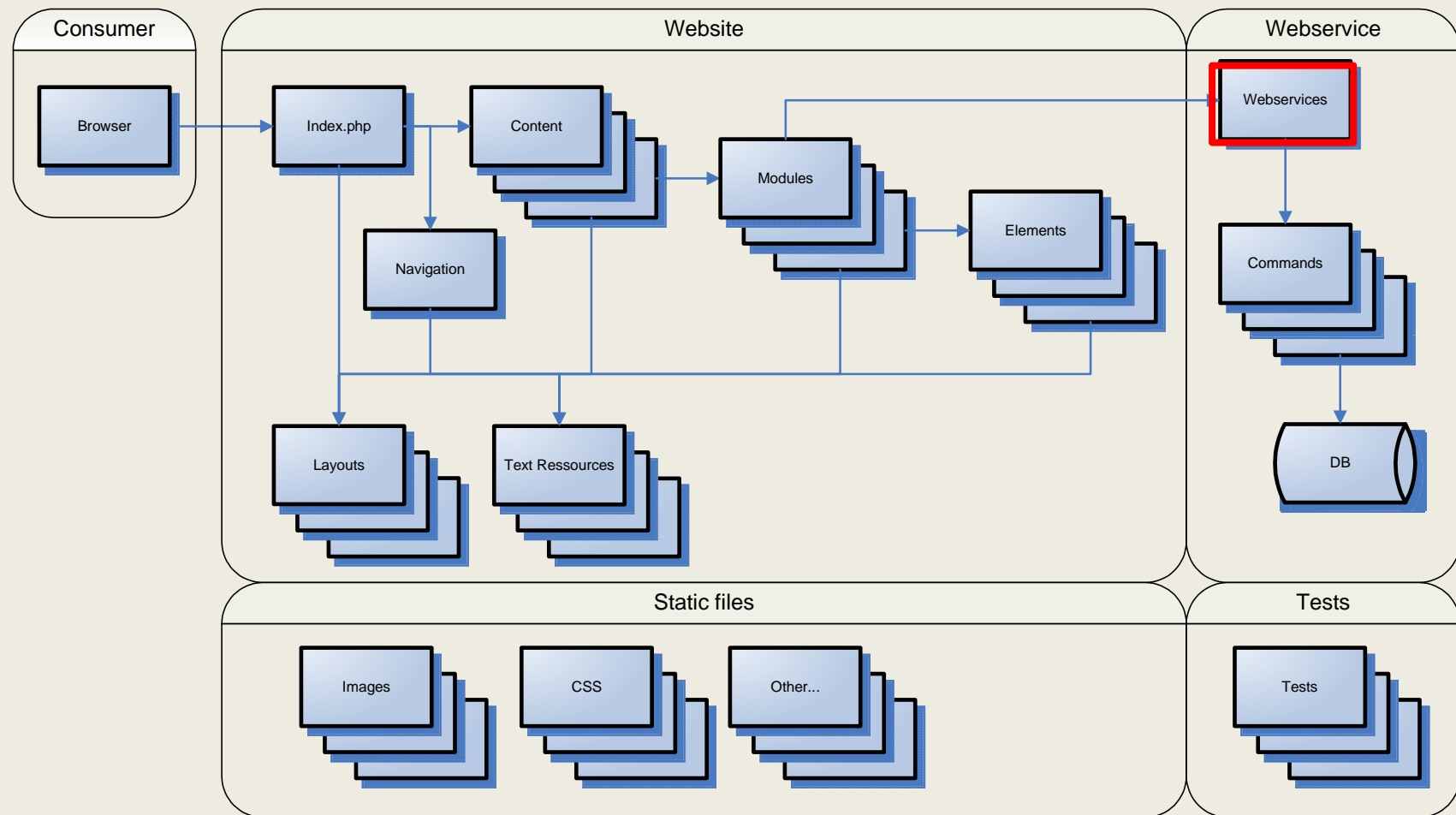
Website Development Kit

Frontend Components / Ressources

- LoadString() vs. Tag Replacements
 - Use LoadString()
 - ...if local text needs to be sorted (e.g. alphabetically)
 - ...if the existence of a specific TID must be checked dynamically
 - ...if tags WITHIN the local text must be replaced by local runtime variables
 - Use Tag Replacements
 - ...in all other cases

Website Development Kit

Backend Components / Webservice



Website Development Kit

Backend Components / Webservice

- **Webservice**
 - Is a class derived from CWebApplication
 - Interfaces with databases or other external services
 - Produces XML output
 - Has a Command Dispatcher

Website Development Kit

Backend Components / Webservice

```
<?xml version="1.0" encoding="utf-8" ?>
<WEBSERVICE NAME="demo/demo" VERSION="1.4">
  <META>
    <NAME><![CDATA[ demo/demo ]]></NAME>
    <VERSION><![CDATA[ 1.4 ]]></VERSION>
    <AUTHOR><![CDATA[ Sascha Wildgrube ]]></AUTHOR>
  </META>
  <REQUEST>
    <GET><![CDATA[ http://www.websitedevkit.com/webservices/demo/demo/?trace=1&command=demo&format=xml&accesscode=12345 ]]></GET>
    <PARAMETERS>
      <PARAMETER NAME="trace"><![CDATA[ 1 ]]></PARAMETER>
      <PARAMETER NAME="command"><![CDATA[ demo ]]></PARAMETER>
      <PARAMETER NAME="format"><![CDATA[ xml ]]></PARAMETER>
      <PARAMETER NAME="accesscode"><![CDATA[ * ]]></PARAMETER>
    </PARAMETERS>
  </REQUEST>
  <RESULT COUNT="6">
    <LIST COUNT="3">
      <ELEMENT INDEX="0" COUNT="2">
        <TAG><![CDATA[ Tag0 ]]></TAG>
        <VALUE><![CDATA[ Value0 ]]></VALUE>
      </ELEMENT>
      [...]
    </LIST>
    <TEST1><![CDATA[ Test ]]></TEST1>
    <TEST2><![CDATA[ Text is encapsulated like this &quot;&lt;![CDATA[This is a test]]&gt;&quot;. ]]></TEST2>
  </RESULT>
```

[to be continued...]

Website Development Kit

Backend Components / Webservice

[...continued]

```
<PROFILING>
  <RUNTIME_SECONDS><![CDATA[ 0.0050101280212402  ]]></RUNTIME_SECONDS>
  <MEMORY_USAGE><![CDATA[ 1309904  ]]></MEMORY_USAGE>
  <REQUEST_DATETIME><![CDATA[ 2009-10-18 20:42:15  ]]></REQUEST_DATETIME>
  <RESPONSE_DATETIME><![CDATA[ 2009-10-18 20:42:15  ]]></RESPONSE_DATETIME>
</PROFILING>
<TRACE>
  <![CDATA[
    WEBSERVICE: Webservice: demo/demo
    WEBSERVICE: Version : 1.4
    WEBSERVICE: Server : www.websitedevkit.com
    WEBSERVICE: InstanceToken: 4adb7da7efa42
    WEBSERVICE: Current time: 2009-10-18 20:42:15 UTC
    WEBSERVICE: Trace activated by parameter.
    WEBSERVICE: m_arrayParams:
    WEBSERVICE:
    Array (
      [trace] => 1
      [command] => demo
      [format] => xml
      [accesscode] => *
    )
    WEBSERVICE: An Access Code is required to use this webservice.
    WEBSERVICE: Lets add some bogus output!
    WEBSERVICE: Profiling: Seconds elapsed BEFORE profiling and rendering: 0.0050101280212402
  ]]>
</TRACE>
</WEBSERVICE>
```

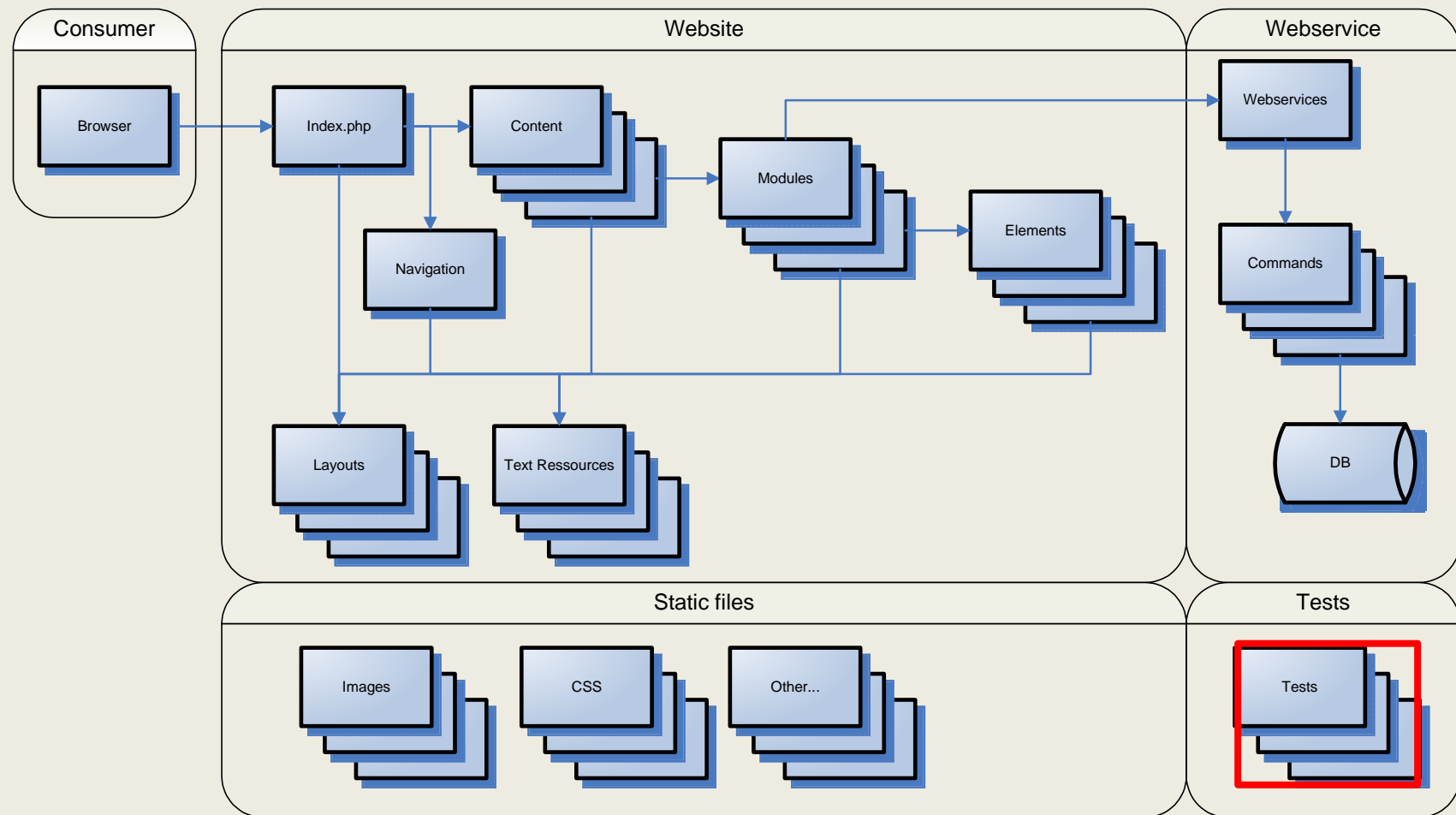

Website Development Kit

Backend Components / Webservice

- System Webservices
 - system/session
 - system/scheduler
 - system/test
 - system/log
 - system/data
 - system/user
 - system/entitlement

Website Development Kit

Backend Components / Tests



Website Development Kit

Backend Components / Tests

- Tests
 - Are classes derived from CUnitTest
 - Can result in OK, FAILED, or INACTIVE
 - Initialize and Cleanup test data if required

Website Development Kit

Backend Components / Tests

```
<script language="PHP">
    class CTest extends CUnitTest
    {
        function __construct()
        {
            parent::__construct("Test CUnitTest");
        }
        function CallbackInit()
        {
            parent::CallbackInit();
            return true;
        }
        function CallbackTest()
        {
            parent::CallbackTest();
            $this->Trace("This is a successful test.");
            $this->SetResult(true);
        }
        function CallbackCleanup()
        {
            parent::CallbackCleanup();
            return true;
        }
    }
    $test = new CTest();
</script>
```

Website Development Kit

Backend Components / Tests

UNIT TEST "Test CUnitTest"

URL www.websitedevkit.com/quality/?test=wdk/test/test_passed.php

INIT BEGIN

INIT END

TEST BEGIN

This is a successful test.

TEST END

CLEANUP BEGIN

CLEANUP END

OK

Website Development Kit

Backend Components / Installer

- **Installer**
 - Is a class derived from CInstaller
 - Invokes Webservices to setup the database, create initial users, privileges, scheduled jobs, etc

Website Development Kit

Code Organization / env.inc

- GetDocumentRootDir()
- GetSourceDir()
- GetWDKDir()
- GetConfigDir()
- GetWebservicesDir()
- GetQualityDir()
- GetRootURL()
- GetWebservicesURL()
- GetAdminEmail()
- GetSystemEmail()
- GetSystemEmailSenderName()
- GetMailDomain()
- GetStage()
- GetDatabaseConfigID()
- GetOperationMode()

Website Development Kit

Code Organization / Assemblies

- Assemblies
 - Are collections of content, navigation, layout, resources, modules, elements, and webservices
 - Can be shared across application systems
 - Are defined by their root directory
 - Can be combined using the assembly line
 - Components within assemblies can „override“ each other
 - The WDK itself is an assembly

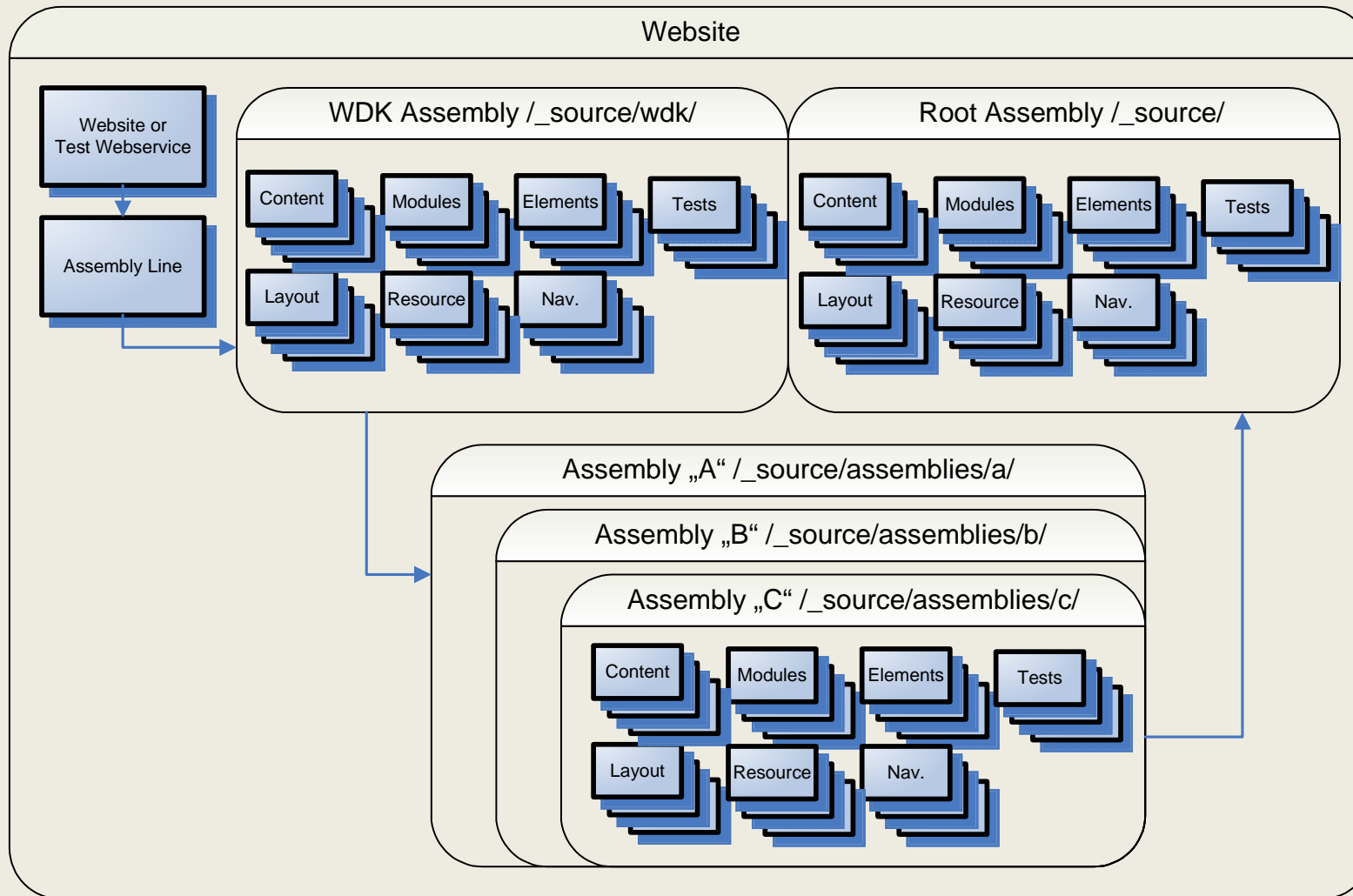
Website Development Kit

Code Organization / Defined Directory Tree

```
/www/  
/www/images/  
/www/css/  
/www/quality/  
/www/install/  
...  
/www/webservices/system/user/  
...  
/www/_source/  
/www/_source/content/  
/www/_source/layout/  
/www/_source/res/  
/www/_source/navigation/  
/www/_source/modules/  
/www/_source/wdk/  
/www/_source/wdk/layout/  
...  
/www/_source/assemblies/example/  
/www/_source/assemblies/example/quality/tests/group1/  
...  
/www/_source/assemblies/example/layout/  
...
```

Website Development Kit

Code Organization / Assembly Line



Website Development Kit

Modules in the WDK Assembly

- Features (Modules)
 - Log-In, Sign-In
 - User Management
 - Scheduler Management
 - Test Center
 - User Settings, Change Passwords
 - Feedback
 - Disclaimer
 - More...

Website Development Kit

Debugging

- `&trace=1`
 - Activates trace output
 - Trace output is generated via `CWebApplication::Trace()` and `CWebApplication::TraceArray()`
 - The trace parameter is inherited to all invoked webservices
 - Trace output is forwarded from webservices into the frontend website

Website Development Kit

Security

- Attacks
 - Cross Site Scripting (XSS)
 - SQL Injection
 - HTTP Parameter Pollution (HPP)
- Defence
 - GetVar() vs. GetVarInsecure()
 - Input Parameter Control
 - Webservices encapsulate database access

Website Development Kit

Coding Guidelines / Hungarian Notation

```
$bCondition = true;
```

```
$strName = "John";
```

```
$nAge = 23;
```

```
$playerBlue = new CPlayer("blue"); // one of many class instances
```

```
$arrayPlayers = array();
```

```
$arrayPlayers[] = $playerBlue;
```

```
$website = new CMyWebsite();           // singleton or global class instances
```

Website Development Kit

Coding Guidelines / Curly Bracet Convention

```
if ($bCondition == true)
{
    DoSomething();
}
```

```
for ($nIndex = 0; $nIndex < $nTotal; $nIndex++)
{
    DoSomething();
}
```

```
function DoSomething()
{
    print("I did it!");
}
```

```
// EOM!
```

Website Development Kit

Coding Guidelines / Functions , Classes, and Interfaces

```
function DoSomething()  
{  
    [...]  
}
```

```
class CMyClass extends CRootClass implements IMyInterface  
{  
    private $m_strSomeMember;  
    function __construct()  
    {  
        $this->m_strSomeMember = "";  
    }  
    function InitRandom()  
    {  
        $this->m_strSomeMember = uniqid();  
    }  
}
```


Website Development Kit

At the end of this presentation you will...

- ...know the benefits of applying WDK ✓
- ...understand the architecture approach ✓
- ...be able to design, plan, and implement application systems using a large diversified team ✓
- ...be capable of using the predefined building blocks ✓
- ...want to use it on a daily basis ?

Website Development Kit

Initial Training

Q&A

Sascha Wildgrube
www.websitedevkit.com
Version 1.5 2010-04-02