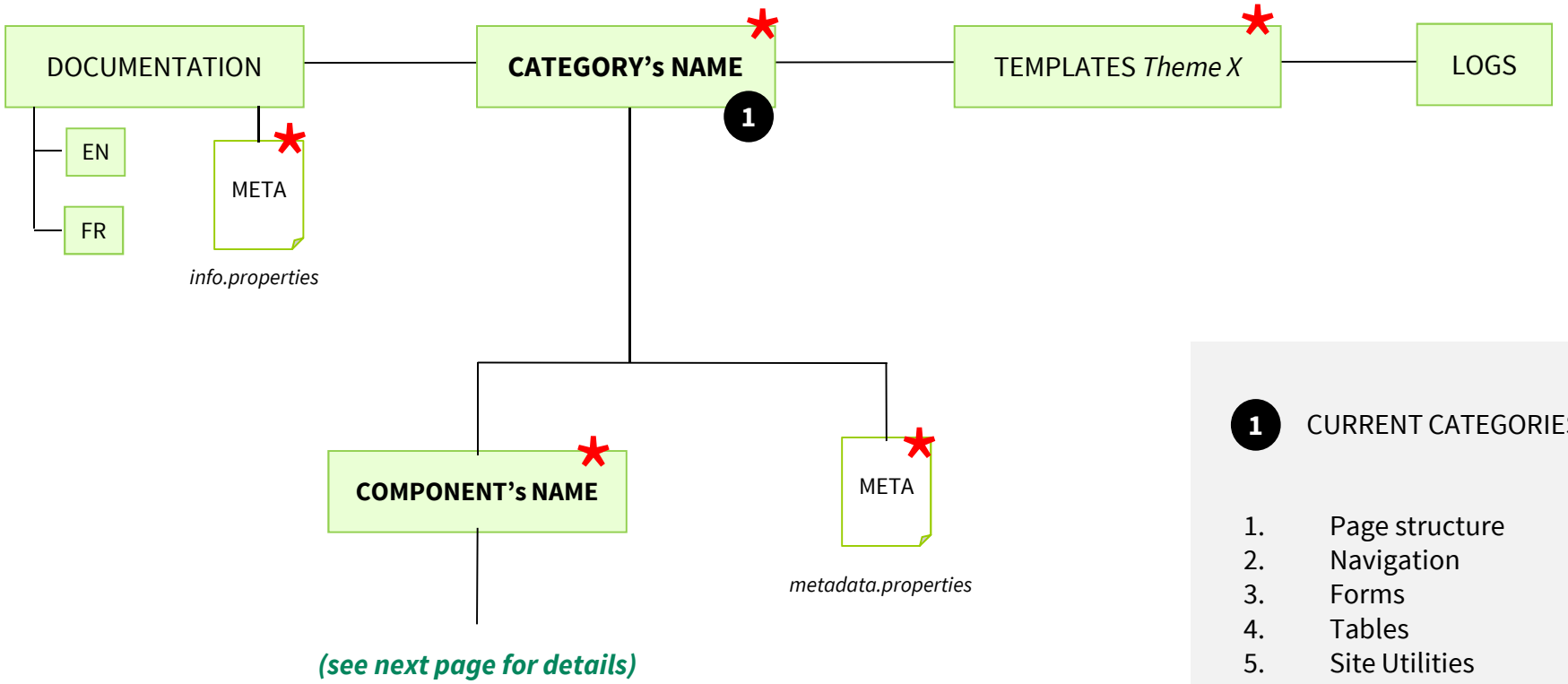
KAWWA PORTAL
Reference

Date : April 06, 2012 / January 2012
Author : Angela RICCI/Emmanuel DEMEY

The Anatomy of Components

Component's parent folder

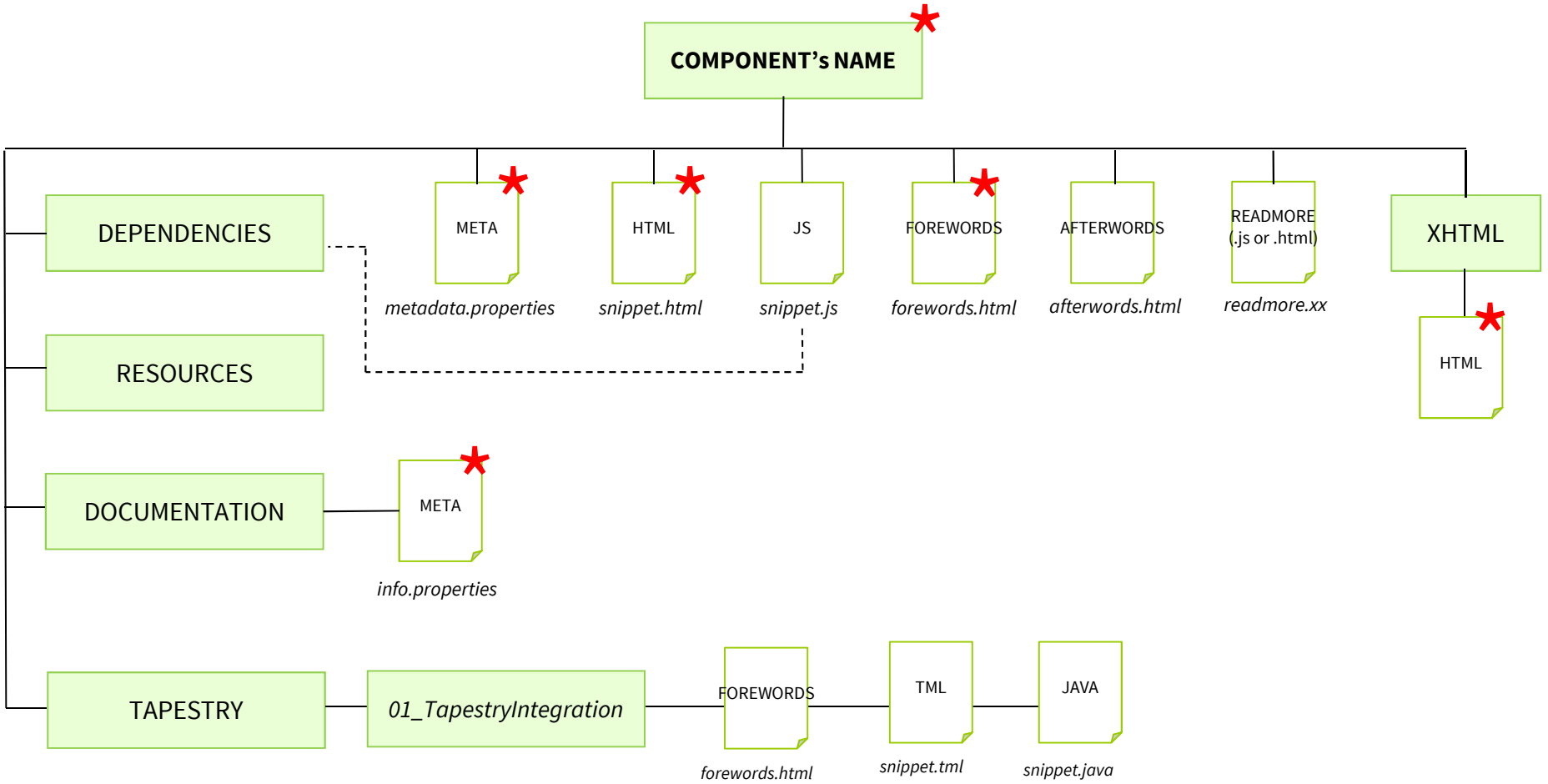


1 CURRENT CATEGORIES

1. Page structure
2. Navigation
3. Forms
4. Tables
5. Site Utilities
6. Blocks
7. Editorial
8. eCommerce

* Mandatory elements

Component's folder content



Description of the Component's folder content

THE FOLDERS

- ***dependencies****:* this folder must be present only in the case of "rich" components that need javascript plugins or jQuery files

- ***resources***: add this folder if your component needs specific images or additional files for its demo (besides the images that depend of the graphical theme). For example, the "jpg" images that illustrate the "k-article" component

- ***documentation***: this optional folder must contain the "*info.properties*" file with information about documents related to the component. See the "Contextual Documentation Scheme" chapter for more information.

- ***tapestry***: this optional folder is present only in the case of Kawwa Components that are part of the Tapestry5-jQuery Kawwa library. Please refer to the "Tapestry Integration Scheme" chapter for more information.

- ***xhtml***: this optional folder will contain a XHTML version of the component's snippet. Please refer to the "Version Scheme" chapter for more information.

★ Mandatory elements

Description of the Component's folder content

THE FILES

- ***metadata.properties*** *: this file contains all meta keywords used to assemble the component's page (see the « Components Metadata » chapter for more information)

- ***snippet.html*** *: this file contains the HTML template for the component.

- ***snippet.js***: this file contains javascript functions definitions and, above all, the « call » for the js function or plugin on page load.

- ***forewords.html*** *: this file contains the text that presents the component and explains when and how to use it. **You must use the HTML syntax here**.

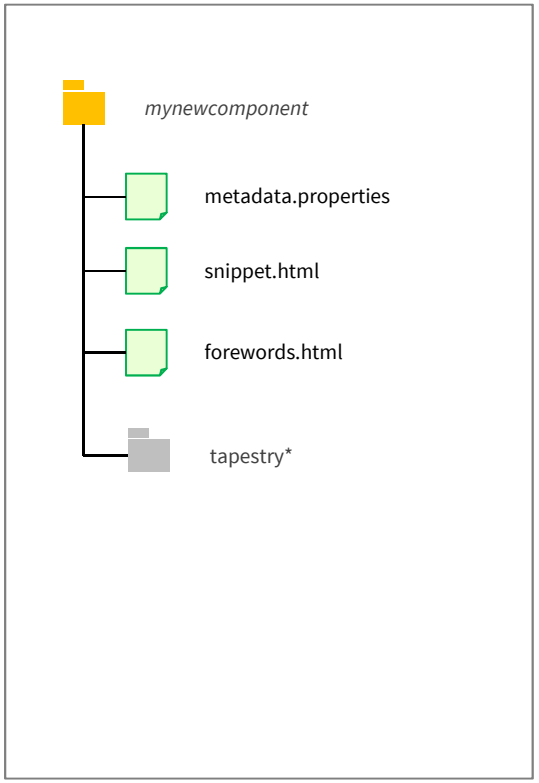
- ***afterwords.html***: this file contains some optional afterword text. **You must use the HTML syntax here**.

- ***readmore.xx*** (« *.js* » or « *.html* »): this file contains a single paragraph with additional instructions about the HTML and/or javascript code. It will be added in the end of the « HTML Plain Code » or « How to Apply » blocks. Do not add the <p> tag in this file.

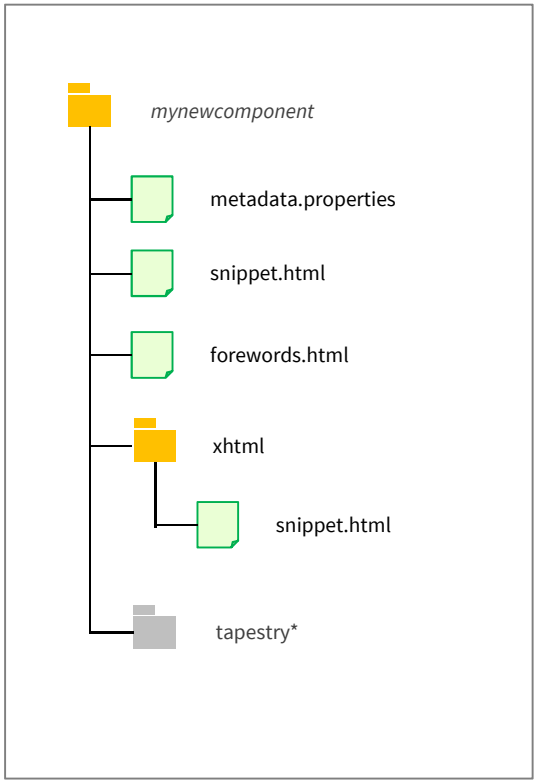
★ Mandatory elements

MINIMAL STRUCTURE FOR A NEW COMPONENT

HTML5 version only



HTML5 & XHTML versions



** Tapestry integration is shown here as a reference as it is conditional to the type of component*

Components Metadata

The Metadata of a component is stored in the « *metadata.properties* » file, which must be found on the component's root folder.

In this file, you will find the following data:

- The « ***name*** » keyword: contains the « visible » name of the component, that will be used to present the component in the index, menu and component's page. This is a mandatory keyword.

- The « ***type*** » keyword: this keyword may have one of the two values, « *component* » or « *group* ». In the case of a component, the only valid value is « *component* ». The « group » value must be used only to categories' folders. This is a mandatory keyword.

COMPONENTS' METADATA (*CONTINUED*)

- The « ***urlParam*** » keyword: contains the name of the component as it must appear in the component's URL. For example, if the value of « urlParam » is « **modal** », the URL to the modal window components will be https://kawwa.atosworldline.com/component/**modal**
  This keyword is also used as a search pattern in the style-sheet to comment the style declarations related to the component.
  This is a mandatory keyword.

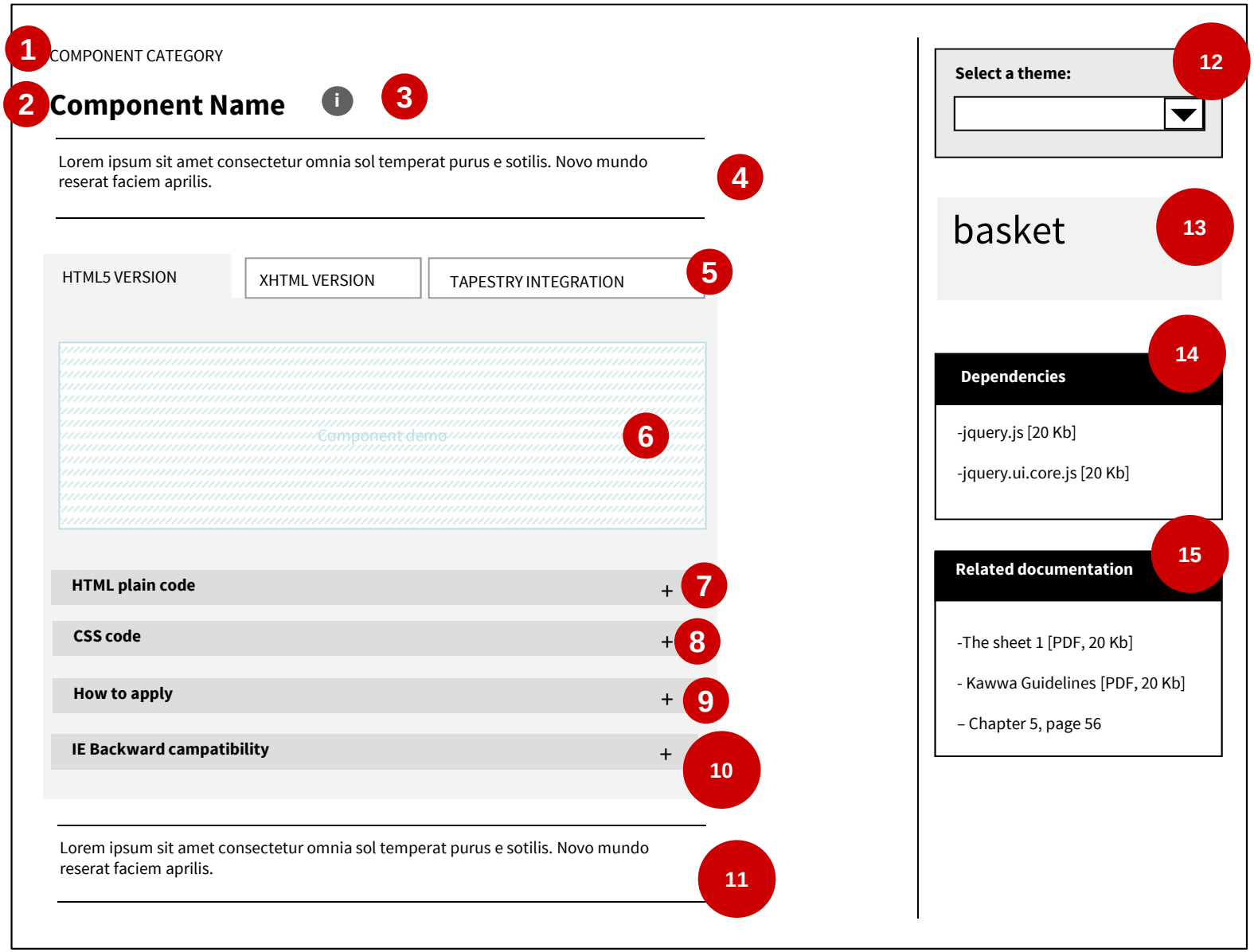
- The « ***tags*** » keyword: contains a list of predefined, case-sensitive values that will identify the component. Possible values are « *jquery* » or « *tapestry* ». Although this is a mandatory keyword, there's no mandatory value (it can be empty), and the order in which the values appear is not important.

- The « ***css*** » keyword: contains the name of the components' css class (or classes). This is a mandatory keyword.

- The « ***version*** » keyword: contains the component's version. If this keyword is missing, it implies version 1.0.

Component data sources on the Kawwa Portal

COMPONENT DATA SOURCES ON KAWWA PORTAL



Hereafter we explain the **sources of content** used to feed the component's page on the Kawwa Portal:

1. **Component's category name**: comes from the meta keyword « ***name*** » defined on « *metadata.properties* » file on category's root folder

2. **Component's name**: comes from the meta keyword « ***name*** » defined on « *metadata.properties* » file on the component's root folder

3. **Component's info** (class name, version, interactive feature, Tapestry Integration): meta keywords « ***css*** », « ***name*** », « ***tags*** », « ***version*** », defined on « *metadata.properties* » file on the component's root folder

4. **Forewords**: comes from the HTML text from the « *forewords.html* » file

5. **Versions tabs**: HTML5 is the default tab. If the meta keyword « ***tags*** » has the value « tapestry » (defined on « *metadata.properties* » file, on the component's root folder), the "Tapestry Integration" tab will be present. The XHTML tab will be present if there's a "xhtml" folder inside the component's root folder.

6. **Component's demo**: the component's demo is assembled using the source snippet files present in the component's root folder. The style of the demo is defined by the current graphical theme of the portal

7. **HTML plain code block**: the content of the « *snippet.html* » file. If a file « *readmore.html* » exists, some information will be added in the end of this block.

8. **CSS code block**: the css declarations of the « CSS Code » block are directly taken from the style-sheet of the current graphical theme of the portal. In order to find the matching declarations for the component, the system searches for an opening and closing comment (see "Style-sheets management" chapter for more information).

9. **How to apply block**: this block is optional. Its presence is conditioned by the existence or not of the « *snippet.js* » file. If this file exists, the content of the block will be read from it. If a file « *readmore.js* » exists, some information will be added in the end of the block.

10. **Afterwords**: HTML text from « *afterwords.html* » file.

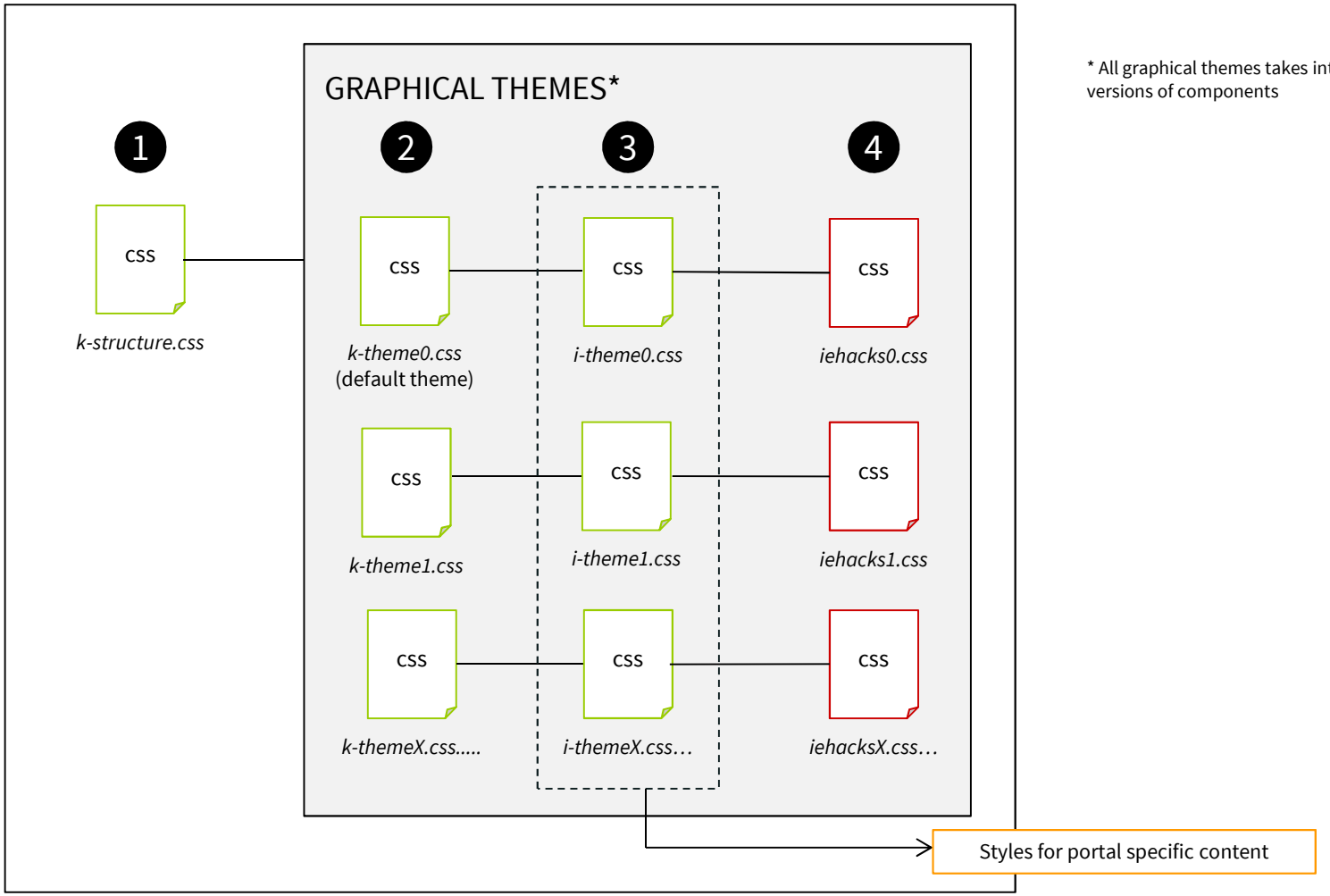
11. **Dependencies**: The content of the « dependencies » folder. It contains all jQuery plugins and/or javascript functions needed, in the case of a rich component

12. **Documentation block**: The content of this block is read from the "*info.properties*" file, contained in the « documentation » folder.

13. **IE backwards compatibility:** The content of this block is **static**. It presents links to the two files needed in order to support IE versions <9, stored on the portal server. Please refer to "Backwards Compatibility Scheme" chapter for more information.

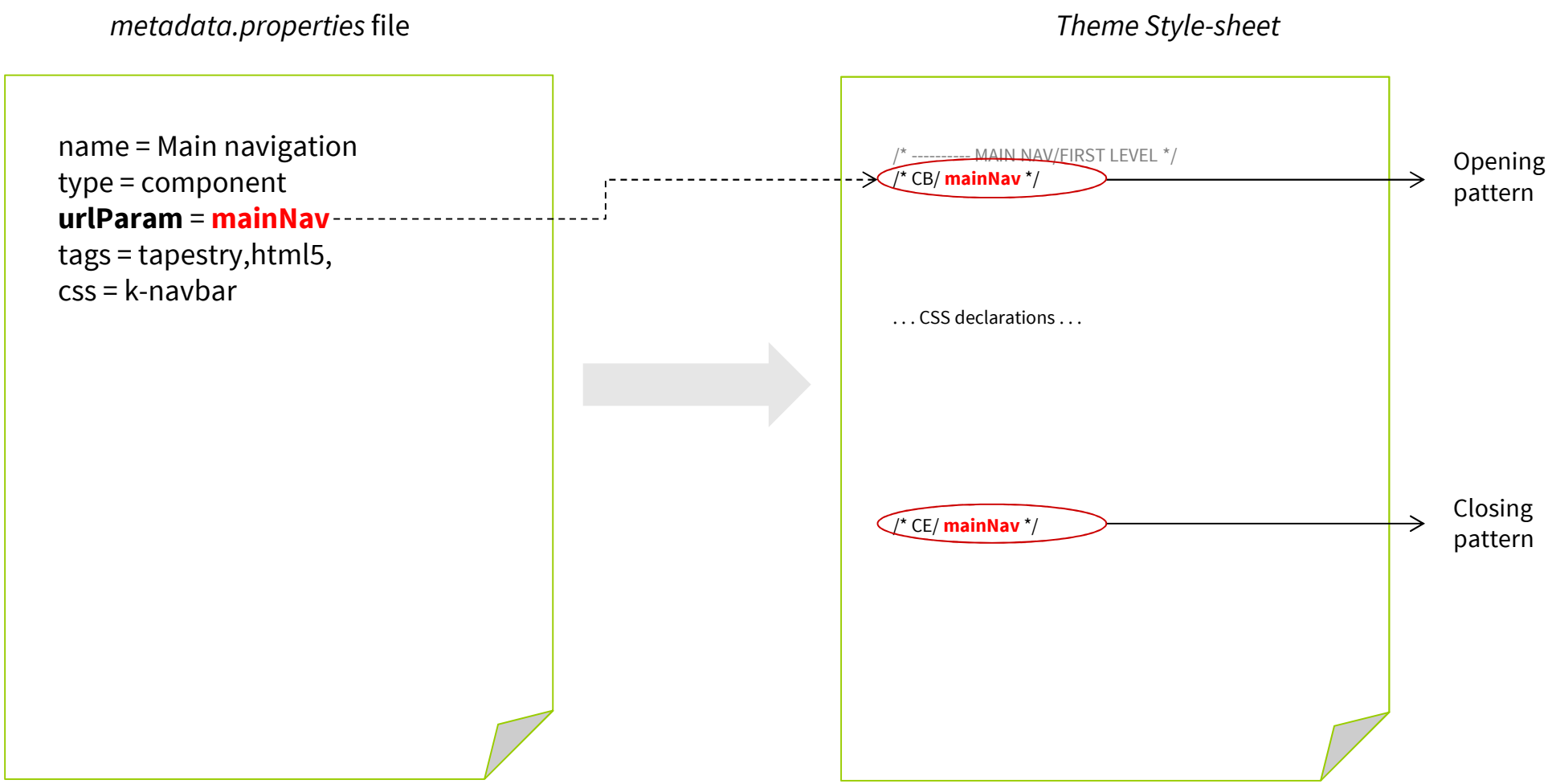
Style-sheets Management

Style-sheets Stacks

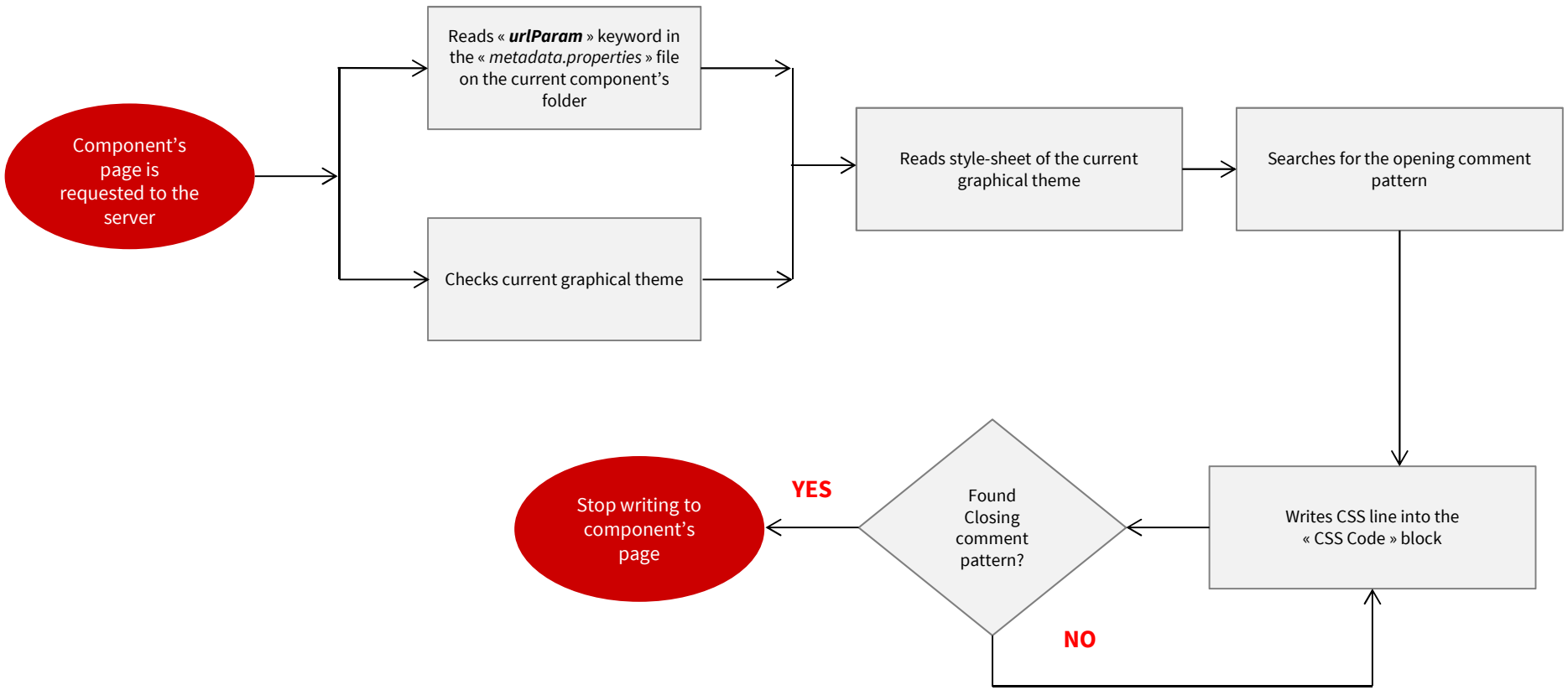


* All graphical themes takes into account both HTML and XHTML versions of components

Style declarations pattern



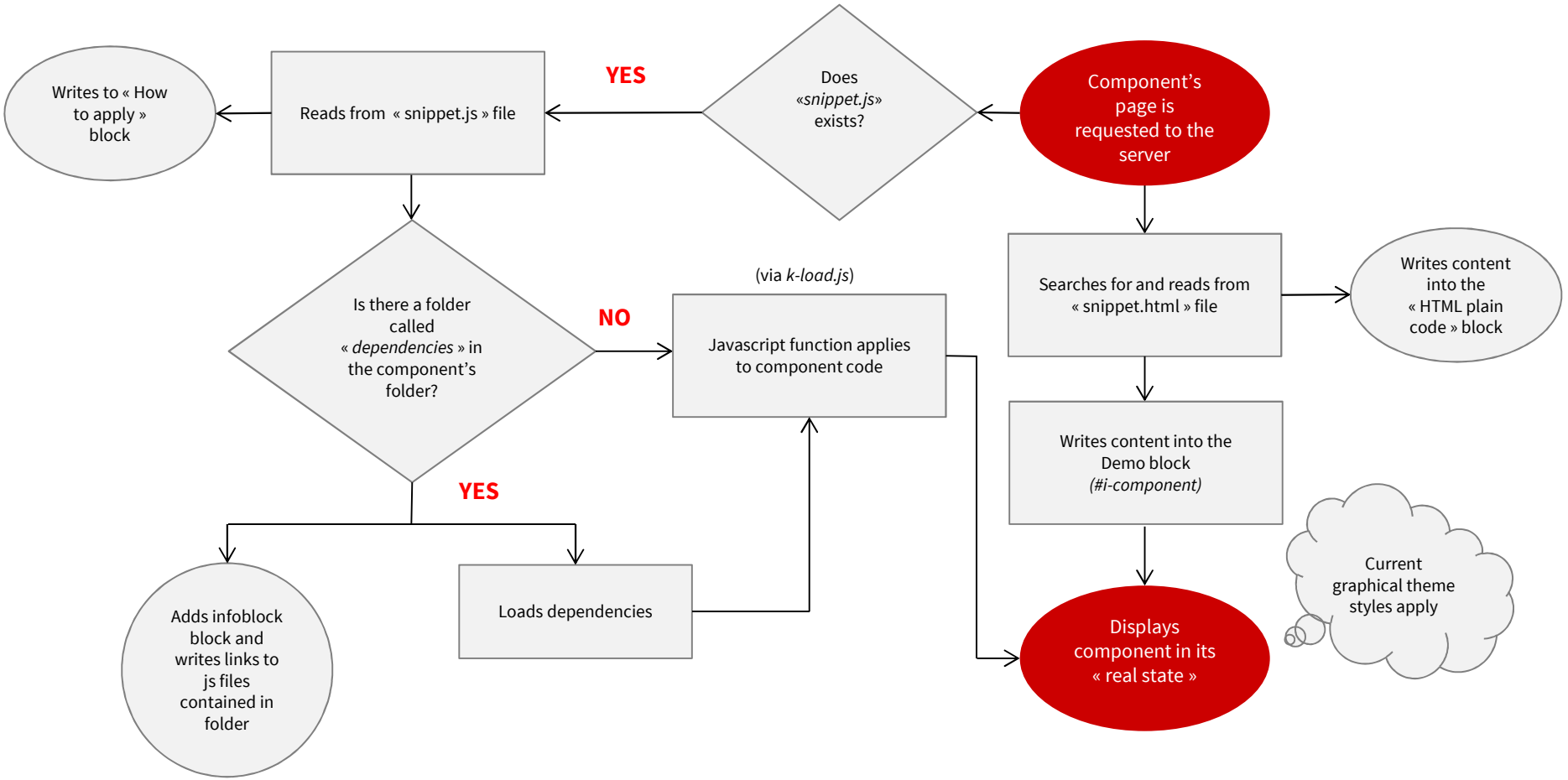
Feeding the component's page with style declarations



Demo Scheme

COMPONENT'S DEMO SCHEME

In each component's page the user can see and test the component in its « real state » (design, interaction, etc.). Hereafter we explain how the component's demo is assembled:



Backwards Compatibility Scheme

Supporting Internet Explorer Versions <9

In order to support older versions of Internet Explorer, we propose the use of two main files:

- **ieGeneralFix.js**: this JavaScript file defines fallback functions for unsupported CSS selectors used in the main style-sheets

- **iehacks*X*.css** (where « *X* » is the number of the graphical theme): this style-sheet will fix display problems and style definitions called by the fallback functions in « *ieGeneralFix.js* ». This style-sheet is graphical theme's dependent (one per proposed graphical theme)

Both files are called using *Conditional Comments*, in the form:

```
<!--[if lt IE 9]>
... Code ...
<![endif]-->
```

Those files are proposed in the components page.

NOTE : On the Templates pack, IE fallback functions are defined in the "*k-general.js*" file

Contextual Documentation Scheme

The “info.properties” file

A component may present one or more links to specific documents that will explain its use.
The « *info.properties* », located in a folder called « *documentation* », is a file that lists those documents titles and references.

There may be a local « *documentation* » folder in the component’s folder listing documentation for a given component, or a documentation folder in a category’s folder, listing contextual documentation for a whole category of components.

The syntax for this file is (one line per referenced document):

```
Usability_Guidelines_Navigation.doc = fiche 4.9 | sheet 4.9
```

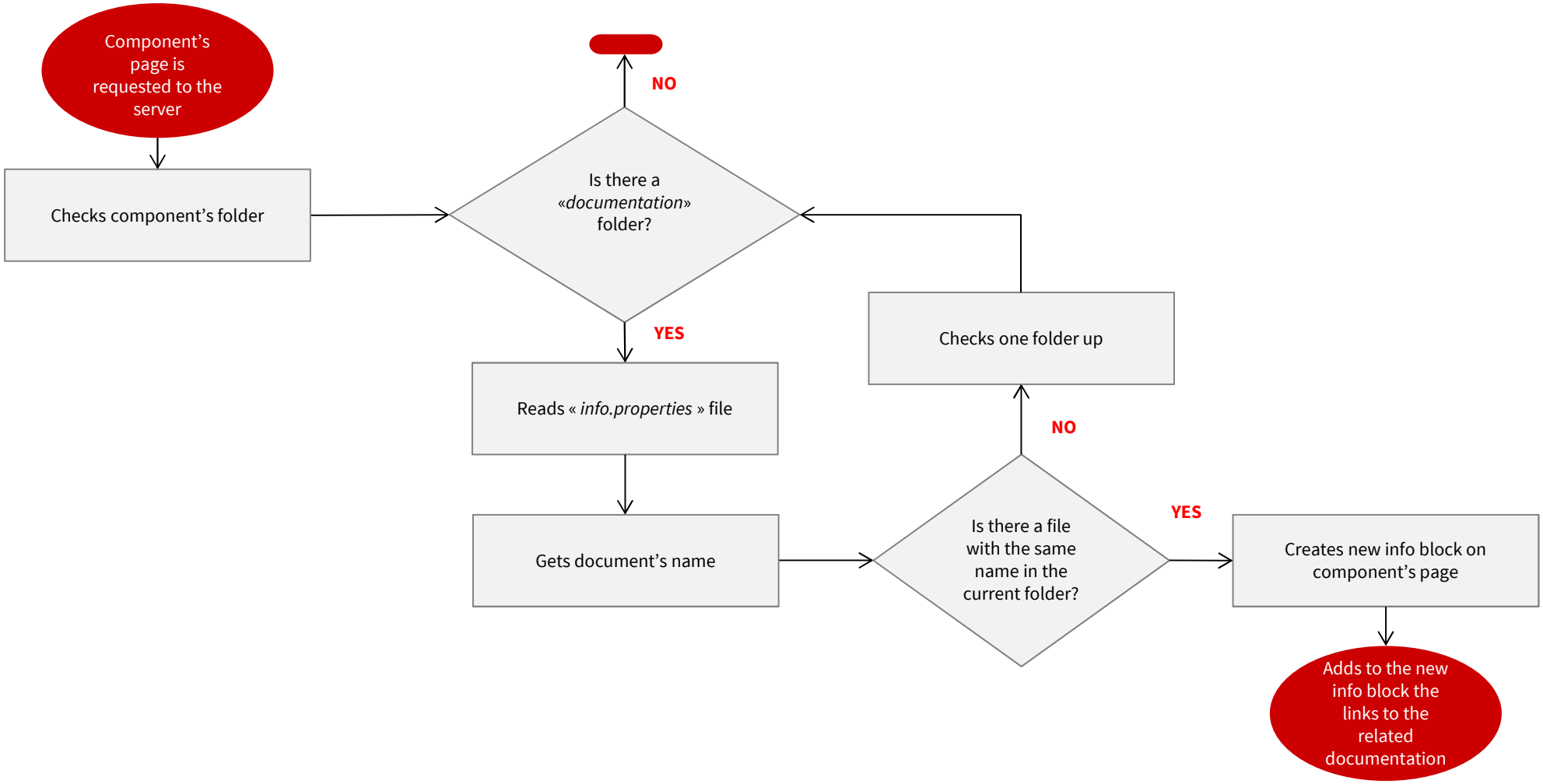
Document file name

Page, sheet or chapter reference (french version) *- not used today*

Page, sheet or chapter reference (English version)

CONTEXTUAL DOCUMENTATION SCHEME

Documentation links at component's page load



8

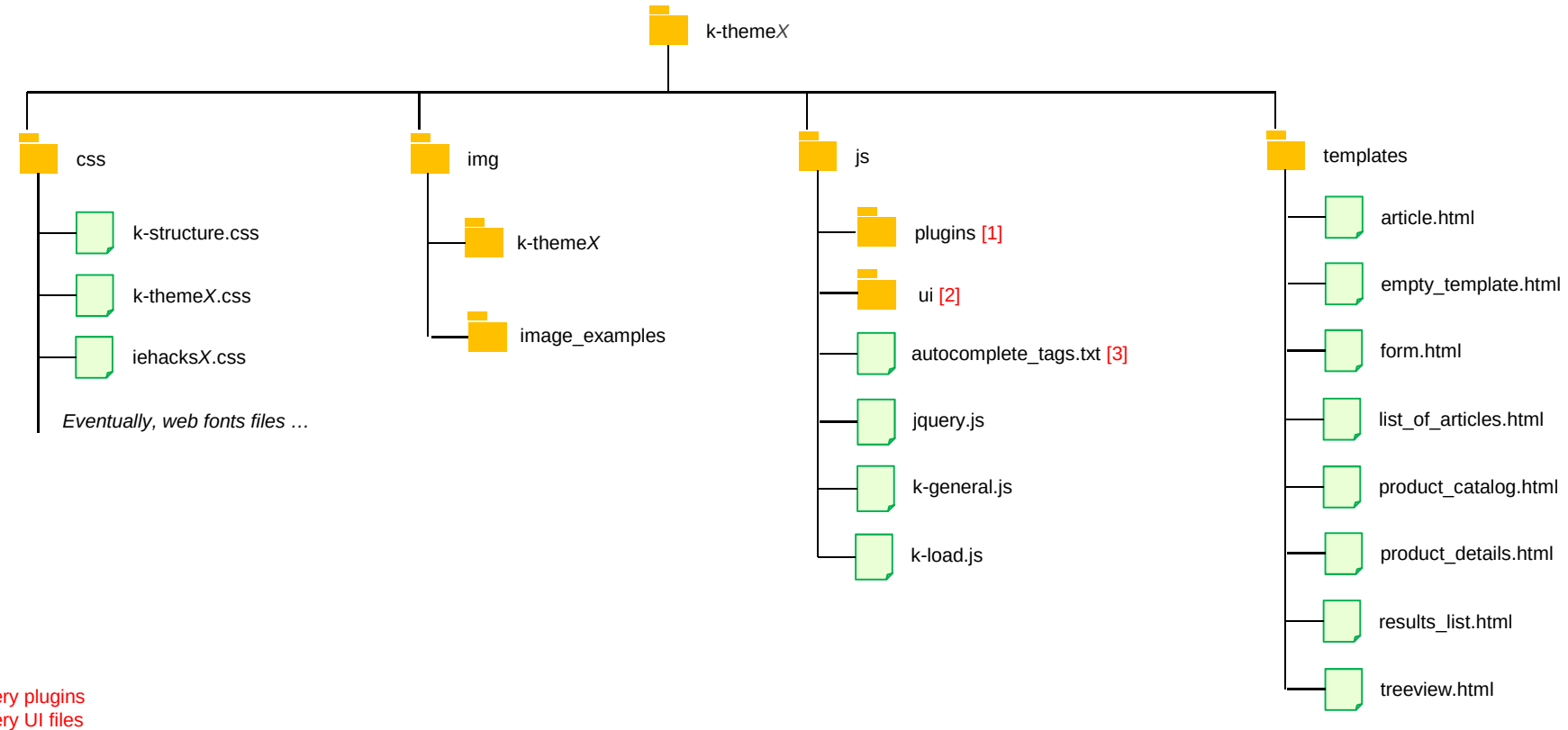
The Templates Pack

THE TEMPLATES PACK

The user may choose to download the Templates Pack with 7 predefined templates and 1 empty page.
At download, it will be the current graphical theme's style-sheets and images that will be packed along with the templates.

Notice that, for each graphical theme, we will have two packs: one pack for the XHTML version and one pack for the HTML5 version of templates and related files.

Here below you present the content of a HTML5 Template pack :



[1] jQuery plugins
[2] jQuery UI files
[3] Autocomplete search source example