



adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 26-28 SEPTEMBER 2016

Staging by Versioning
Ralf Wunsch, IST GmbH Dresden.

the Idea

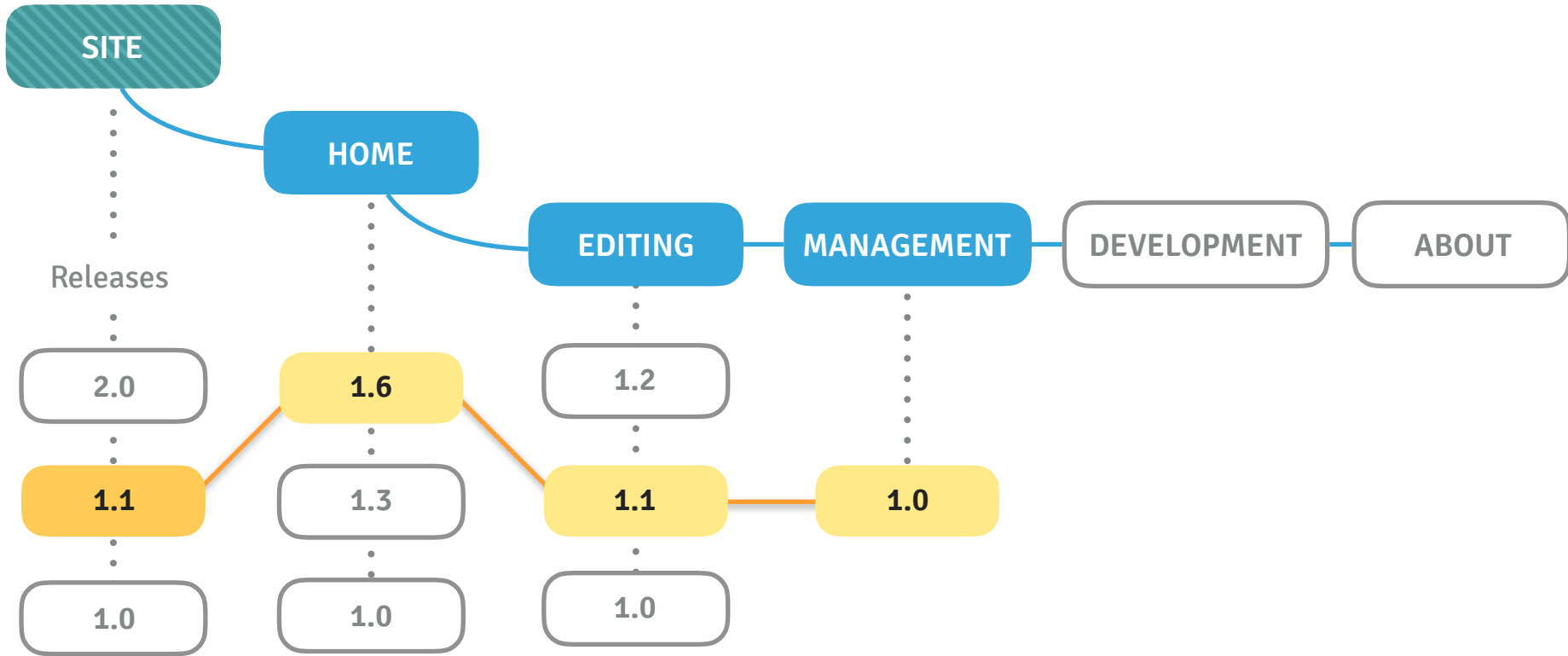
a simple open source application management platform
starting with a small CMS nucleus

- **AEM**
 - huge, heavyweight, expensive
- **but (1) - a nice template**
- **but (2) - needs simplification...**
 - no ,staging' for I18N (no MSM)
 - no different instances (author, publish), no replication
- ...

- Versions
 - JCR defines a versioning api
 - not really used by sling based applications
 - let's use this for content staging...
- use the 'versionable' mixing type
 - each page content node is 'versionable'
 - each asset original node is 'versionable'

- there is a new object bundling the pages:
the ‚Site‘
 - the site manages releases
(and some other things like mappings rules and languages)
 - ‚versionables‘ can be assigned to releases
 - one release is marked as the ‚public‘ release

an example



Homepage - versions



The screenshot shows the Composum Pages editor interface. On the left is a file tree for the project 'jcr.root', with 'home' selected. The main area is divided into sections: 'Development' (listing Sling components and editing capabilities), 'About' (describing the project as open source and available on GitHub), and a footer for 'adaptTo() 2016 staging demo'. On the right, a version control panel displays a list of versions:

Version	Timestamp
1.6	2016-09-28 07:51:15
no custom label	
1.3	2016-09-28 07:36:54
no custom label	
1.0	2016-09-27 20:48:56
composum-release-1.0	

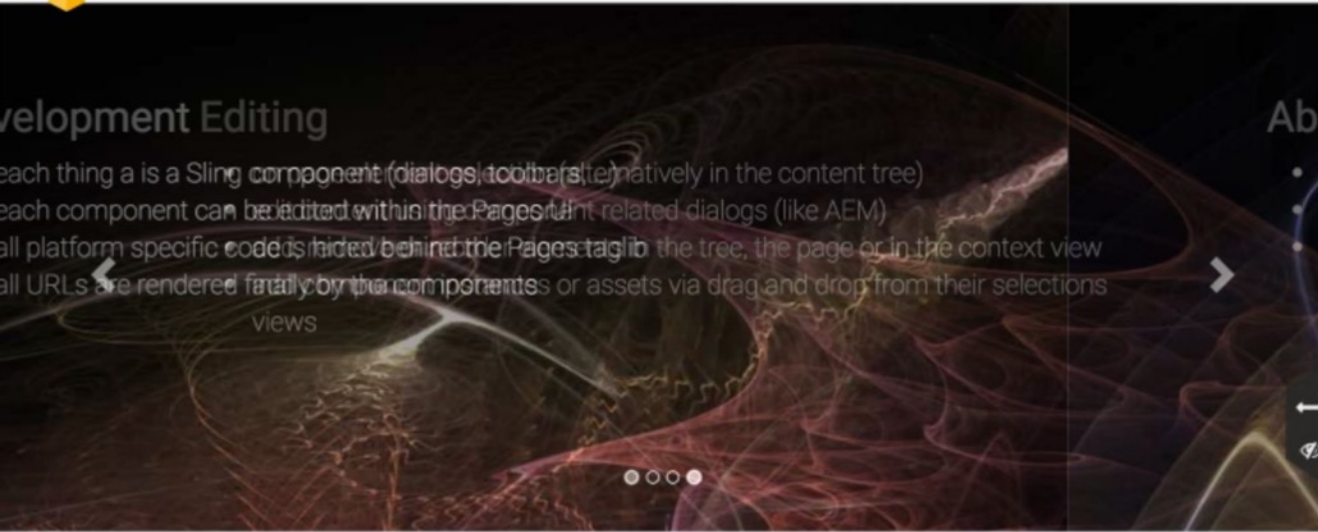
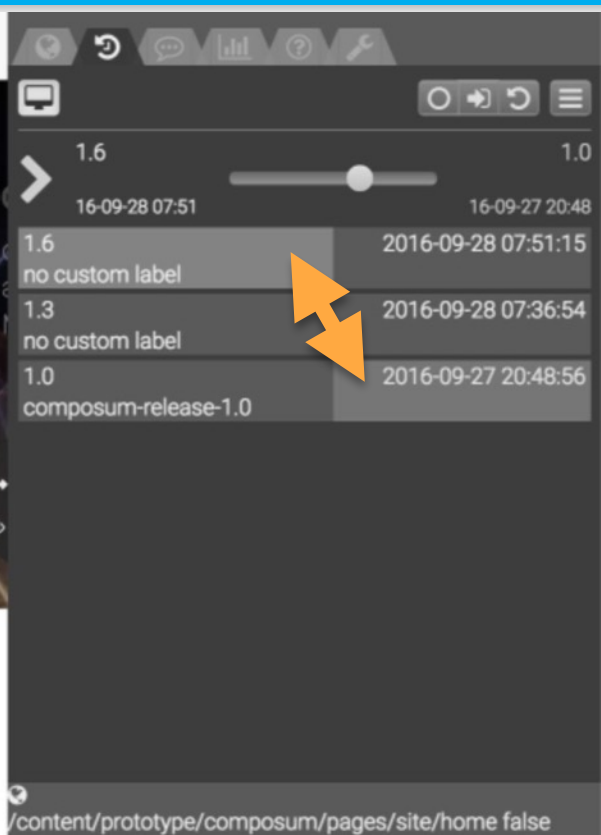
An orange arrow points to the '1.0' version entry in the list.

Homepage - comparing

Composum Pages

Development Editing

each thing a is a Sling component (dialogs, toolbars, etc) alternatively in the content tree
 each component can be edited within the Pages unit related dialogs (like AEM)
 all platform specific code is hidden behind the Pages tag in the tree, the page or in the context view
 all URLs are rendered finally by the components or assets via drag and drop from their selections

Version	Label	Timestamp
1.6	no custom label	2016-09-28 07:51:15
1.3	no custom label	2016-09-28 07:36:54
1.0	composum-release-1.0	2016-09-27 20:48:56

URL: /content/prototype/composum/pages/site/home false

Composum Platform 2016

The Composum Platform is a Web application platform based on the Apache Sling framework. A simple demo to illustrate the support content management, asset management and fast application development.



Composum Pages prototype

Pages is the brand new module which provides a CMS UI to manipulate your content in an intuitive way. the content is organized in a structured way.



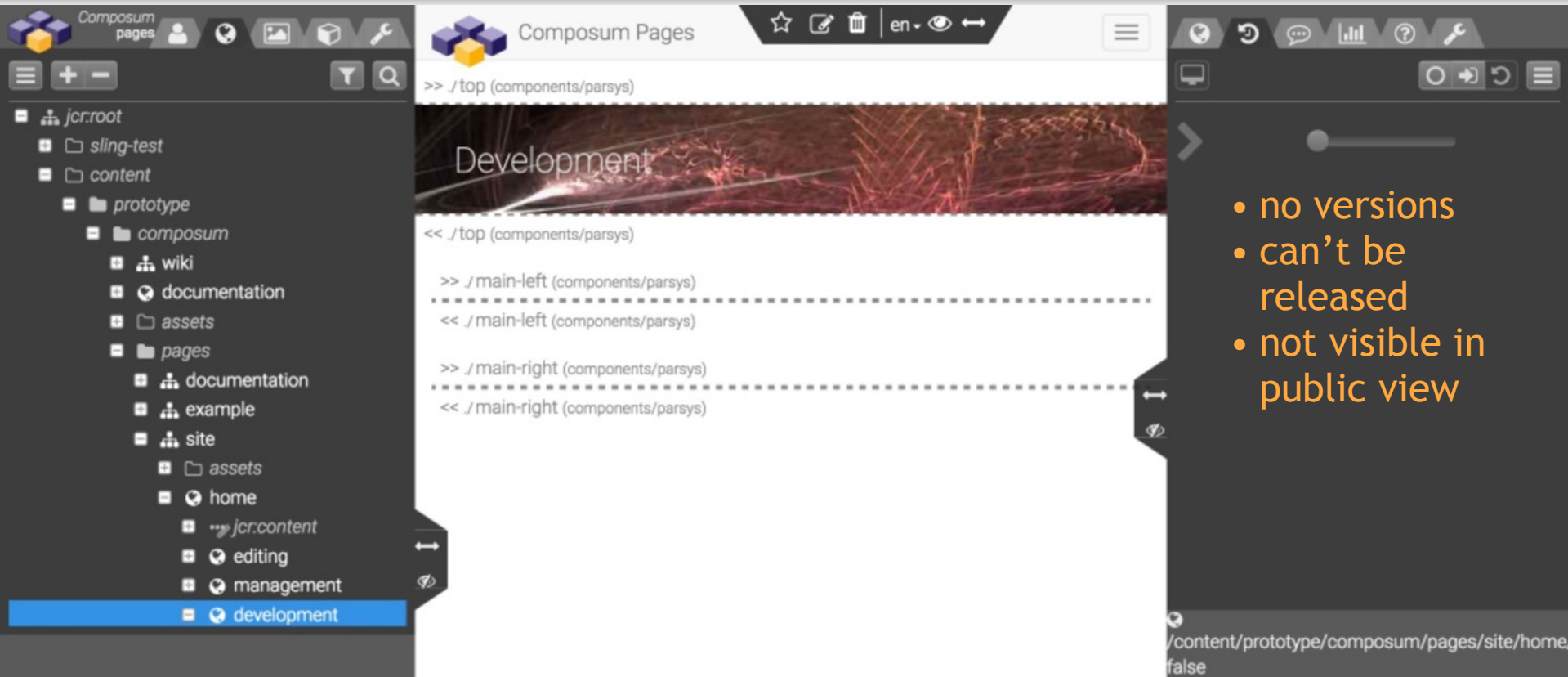
About

- open source project
- available on GitHub: <https://github.com/ist-dresden/composum>
- MIT License - contributors welcome

all pages in the navigation



Dev page - no versions



The screenshot shows the Composum Pages interface. On the left is a sidebar with a tree view of the project structure. The main area displays a page titled 'Development' with a red and black background image. The page content is a simple layout with several components, each with a path like '>> ./top (components/parsys)'. The 'development' page is highlighted in the sidebar. On the right, there is a dark overlay with a list of bullet points in orange text.

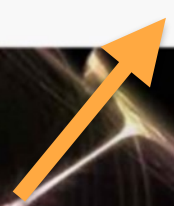
- no versions
- can't be released
- not visible in public view

Homepage - public view

Development

- each thing is a Sling component (dialogs, toolbars, ...)
- each component can be edited within the Pages UI
- all platform specific code is hidden behind the Pages taglib
- all URLs are rendered finally by the components

navigation contains released pages only
(transparent for the navigation component)



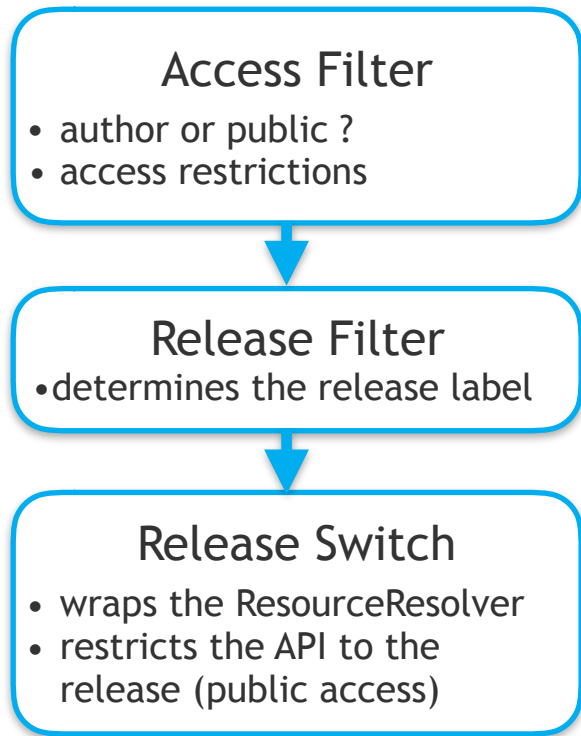
Composum Platform

The Composum Platform is a Web application platform based on the

Composum Pages

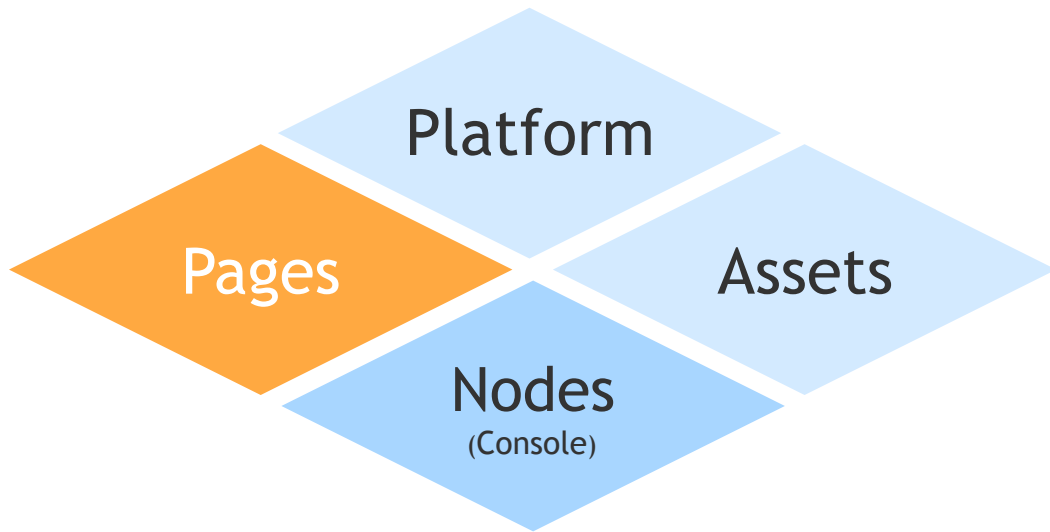
Pages is the brand new module which provides a CMS UI to manipulate your

the request handling



- for each request
 - autor or public request ?
 - restrict the access appropriate
 - determine the right release (public only)
 - restrict delivered artifacts to the selected release content

What's going on?



- first prototype
- will be public in a few months
- available on GitHub

<https://github.com/ist-dresden/composum>