

# A W3C Widget implementation: The Palette Portal

*Jérôme Bogaerts, Alain Vagner*  
*{firstname.lastname}@tudor.lu*

*W3C WAFWG meeting, 02-26-2009*

## ➤ The Palette Project (EU FP6)

- **Target:** Communities of Practice in the learning sector
- Lots of heterogeneous tools and services  
=> **need of integration**

### ➤ Proposed integration services

- Unified search
- Unified authentication
- Unified storage
- ...
- **Web Portal**



- In Palette:
  - Wide range of services available
    - Web Applications, Desktop Applications, Web Services
  - Different Communities of Practice with specific needs
    - Selection and adaptation needed in the proposed services
- The Palette Services Portal provides:
  - Central access point to the services
  - Customizable view
  - Global view of the group activity
  - Interaction between these services

- **Lightweight**
  - Easy and fast development of new web widgets
  
- **Well-known technologies**
  - No need to learn a new environment
  
- **Integration in the web architecture**
  - Rest web services, ROA
  
- **Interoperable and open solution**
  - Not be tied to a specific vendor / proprietary platform

- **Evaluated specs**
  - Google gadgets, Netvibes Universal Widgets, Yahoo! Widgets, Microsoft Gadgets, Apple Dashboard Widgets, and Opera Widgets
  
- **Common denominator**
  - Widget configuration section
    - Meta-data, settings
  - Main Widget file
    - Logic, layout and structure
  
- **Similar to MVC**
  - View: widget layout and structure
  - Controller: programming logic of the widget
  - Model: widget settings and user preferences

- All widgets formats very similar
- Only one open standard matching the requirements
- Some extensions needed
  - provided by none of these specs

=> choice of the W3C Widget format spec as a basis

- Moving target
  - Freeze of the version used for implementing
  - Last version of the portal aligned with the last spec
  
- Partial implementation
  - Implementation in priority
    - Base of the spec
  - Not implemented (packaging & API)
    - Window Modes
    - Updates
    - Features
    - Widget signature
    - I18n

- Main problem
  - Same origin policy on XHR
  - W3C Widget spec oriented towards desktop widgets
    - > out of scope
  - Currently: XHR proxying
    - => W3C Cross Origin Resource Sharing spec ?

## ➤ A prototype

- Tested in the project on 2 groups of users, real data
- Security, performance not really tested

## ➤ Open source

- Freely available on Google Code
- Free software: GPL v2
- Community: external contributions welcome
- Presentation & demos:
  - <http://palette.tudor.lu/content/index.php?page=palette-services-portal>
- Sources:
  - <http://code.google.com/p/mywiwall/source/checkout>
- Mailing List « Talk about widgets »
  - <http://groups.google.com/talk-about-widgets>

- Innovation
  - Extensions as a mean to explore new use cases
- Clean separation of extensions
  - Extensions spec in a specific document
  - Manifest extensions in a separate namespace
- Compatibility
  - widget compliant with the W3C spec without our extensions should work
- No unnecessary extension
  - what can be done with the strict spec should not be extended by laziness

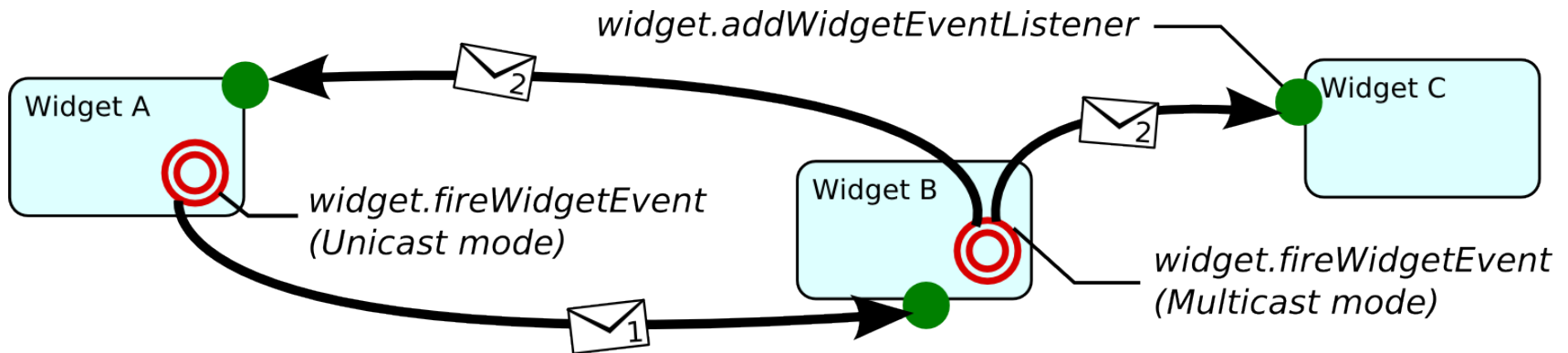
- Main extensions categories
  - Preferences handling
  - Remote connexions
  - Widget authentication
  - **Inter-widget messaging**
  - **Drag & drop between widgets**

- Prefs declaration in the manifest file
  - Used by the portal to serve a configuration form
- Optional prefs typing
  - Basic types
  - Enum

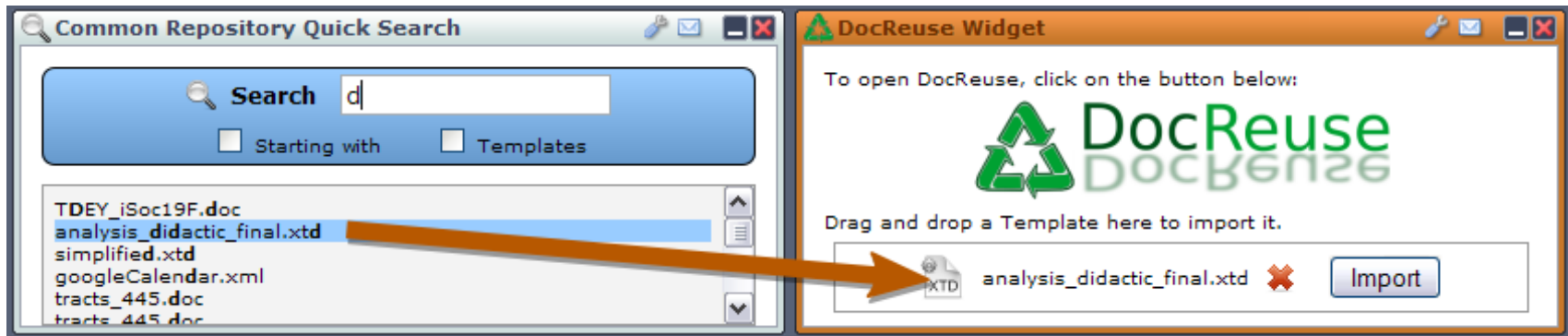
- Necessity to bypass same origin policy on XHR
  - Simple proxy, Rest compatible
  
- Extension of the widget object
  - Widget.httpGet .httpPost, .httpPut, .httpDelete, .httpGetJson
  - Widget.setHttpCredentials
  - Widget.setContentProxy
  
- Rewrite needed
  - Homogenization with Tencompetences APIs
  - Implementation of W3C Cross Origin Resource Sharing spec ?

- Use case: log in to the portal and be logged in all widgets
- Trust between widgets and external web services
- Authentication on the Portal propagated to web services
- Encryption: 256bit AES
- Need to be coupled to a distributed identity system (OpenId)

- Several communications modes possible between widgets in a same container
  - Unicast
  - Multicast
  - Broadcast
- Event type as URI
- No server-side event propagation



- Exploits the inter-widget messaging system
  - Widget.bindWidgetToDropType()
  - Widget.addDragData()



# Demonstration

- Ongoing work
  - Security
    - Google-caja ?
  - Compatibility with other widget formats
    - Google gadgets, Netvibes UWA
  
- Future work
  - I18n
  - Multi-container messaging system
    - Comet
  - Manifest for groups of widgets
  - RDA-based desktop widget platform
    - Mozilla Prism, Adobe Air ?

- Possible to use the W3C widget spec for Web Widgets
- Inter-widget messaging, DnD
  - Different from the Tencompetences solution
  - Convergence possible
  - Interesting functionalities
  - Probably need to be standardized
- Palette extensions spec:
  - <http://palette.tudor.lu/files/portal/specifications/palette-widget-format-specification-january-2009.pdf>
- Palette Services Portal Website
  - <http://palette.tudor.lu/content/index.php?page=palette-services-portal>

# Thanks for your attention!

*<http://palette.tudor.lu>*

*Jérôme Bogaerts, Alain Vagner*  
*{firstname.lastname}@tudor.lu*