A JCR view of the world

📁 a-jcr-folder
　　📁 a-subfolder
　　　　🔶 a-jcr-node
　　　　　　propertyA : Nodes have 0..N properties
　　　　　　anotherProperty : Another value

Everything is content, everything is a tree

Bertrand Delacretaz, Senior Developer, Adobe CQ5 team.
Apache Software Foundation member and (current) director
@bdelacretaz - http://grep.codeconsult.ch

Berlin Buzzwords 2012
Powered by Apache Sling (content and rendering) and deck.js
(presentation)

JCR?
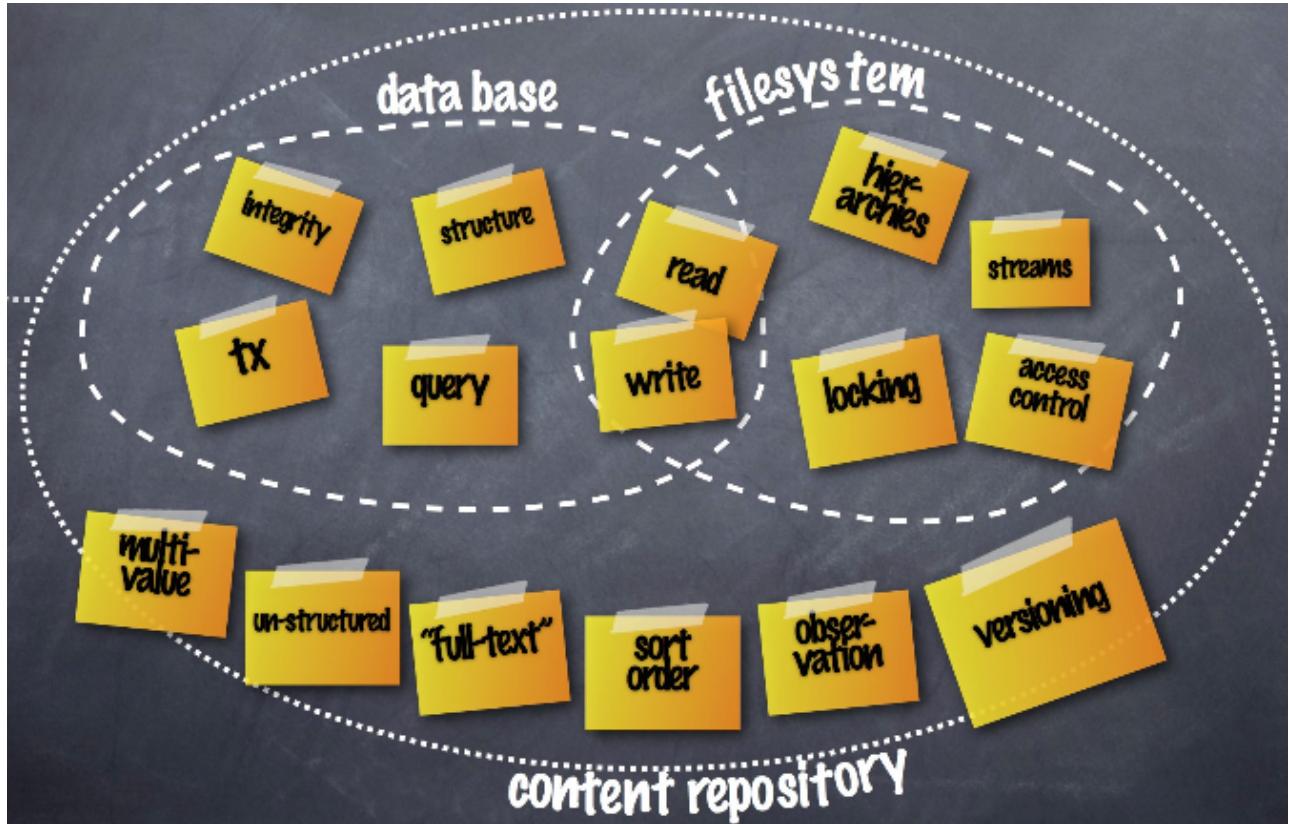
Java API for content repositories (JSR 283)

A big tree of nodes and properties.

Rich infrastructure for content-based applications.

JCR Explorer

Java API for content repositories (JSR 283)

A big tree of nodes and properties.

Rich infrastructure for content-based applications.

A basic website page

📁 my-page
   ◆ jcr:content
      jcr:text : This is the *text* of the page.
      jcr:title : Title of my page
      📁 images
         ◆ image_1.jpg

       ◆ image_2.jpg
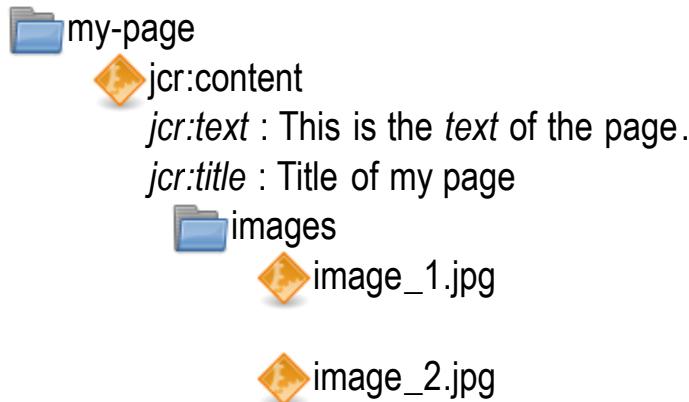
The jcr:content subnode will later act as a container for more complex content structures.

A more realistic page

📁 /content/my-site/my-page
    🔶 jcr:content
        jcr:title : Title of my page
            📁 content-blocks
                🔶 block_1

                🔶 block_2

            📁 images
                📁 first-image.jpg
                    🔶 original.jpg

                    📁 renditions
                        🔶 first-image-212x43.jpg

                        🔶 first-image-512x83.jpg
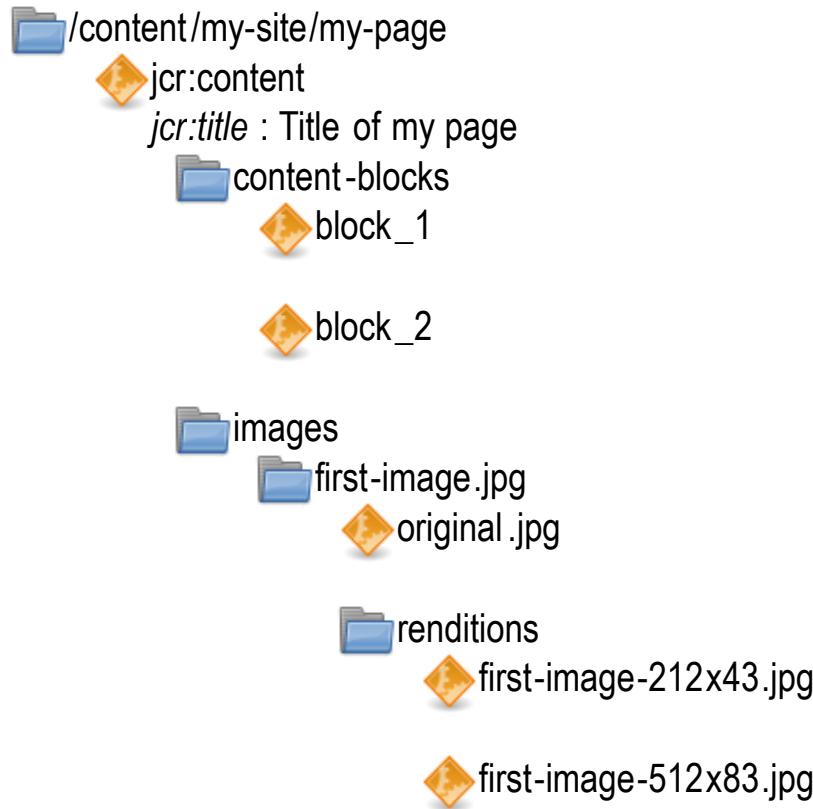
Uses JCR observation to generate renditions when original image is uploaded.

The first-image node is a folder, not just a file. A "micro-tree" of content.

A print queue

📁 print-queue-example
    📁 incoming
        🔶 printjob_20120524_1502

    📁 printers
        📁 canonMP640
            📁 done

            🔶 printjob_20120524_1458

        📁 Laserjet 5M

    📁 rejected
        🔶 printjob_20120524_1458
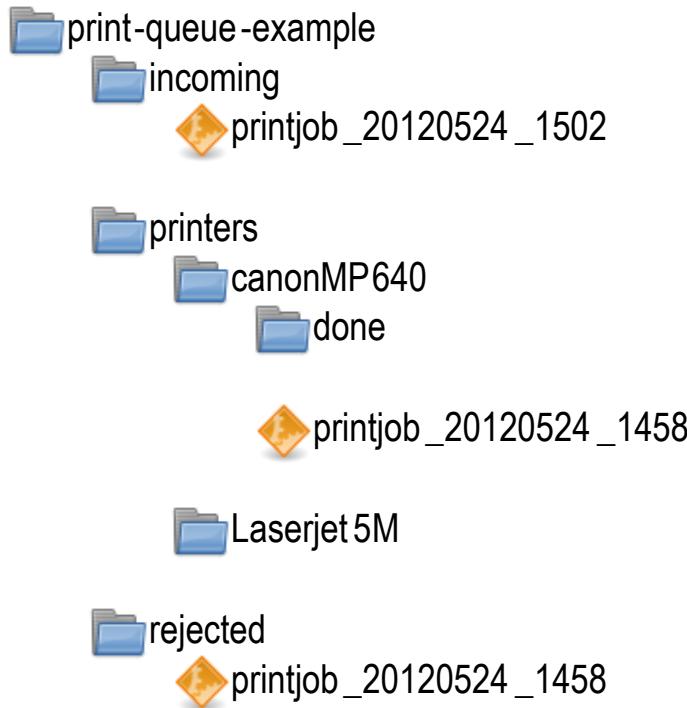
Jobs arrive under "incoming" and move under a printer's node while they are executed. Remember unix spooling?

The "rejected" folder holds jobs that could not be executed.

Defining virtual hosts

📁 /etc/map/http

    🔶 www_geometrixx_de

       sling:internalRedirect : /content/geometrixx/de

       sling:match : www.geometrixx.de/$

    🔶 www_geometrixx_fr

       sling:internalRedirect : /content/geometrixx/fr

       sling:match : www.geometrixx.fr/$

Obvious, transparent…no explanation needed

Adobe CQ5's content structure

📁 content
  📁 geometrixx_en

📁 libs
  📁 wcm/mobile
    📁 config

    📁 install

📁 apps
  📁 wcm/mobile
    📁 install

📁 etc
  📁 virtual-hosts

📁 var
  📁 compiled-scripts
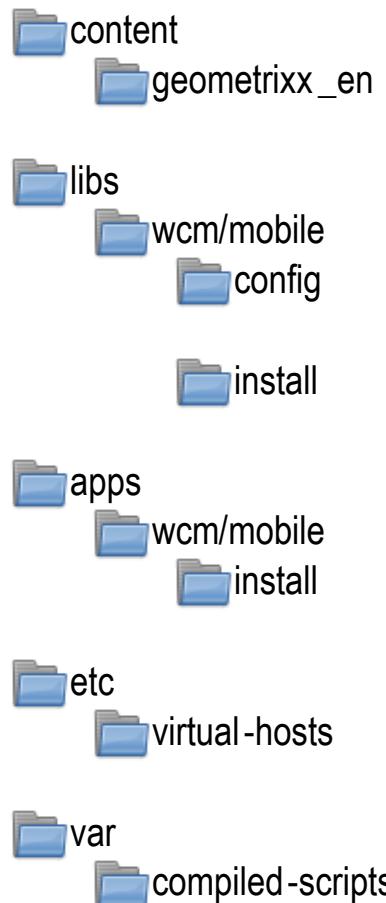
Looks familiar?

Code and configurations

📁 /apps/my-app
    📁 install
        🔶 a-bundle .jar

        🔶 sling.get.servlet .jar

    📁 config
        🔶 o.a.sling.GET.config

    📁 config.dev
        🔶 o.a.sling.logging .config
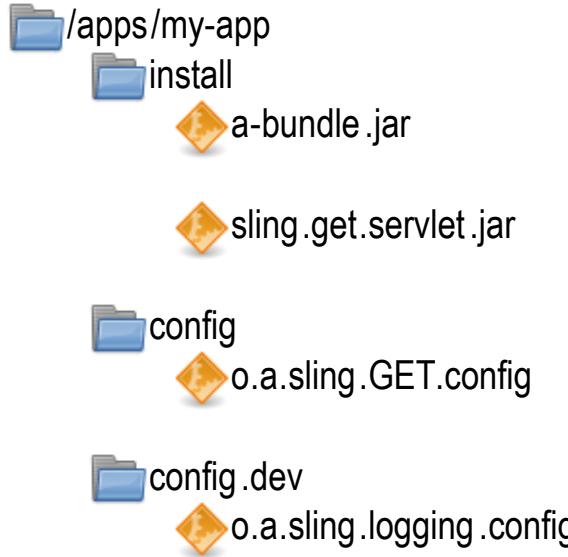
Sling's JCR installer detects and installs bundles, configs and other resources.

Scary? Extremely useful - with access controls of course.

A workflow model

📁 /etc/workflow/models/my-model
    jcr:description : Please approve ...
    jcr:title : Content Approval
    📁 states
       📁 state_1
          🔶 preconditions

          🔶 transitions

      📁 state_2
         🔶 preconditions

         🔶 transitions
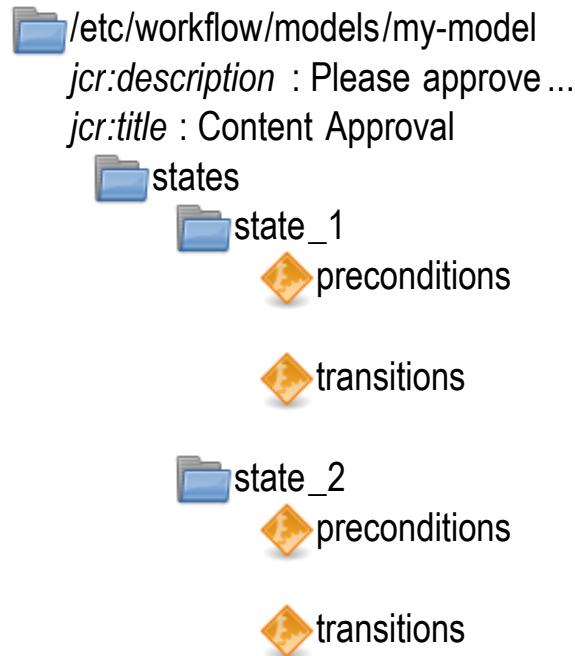
Useful JCR features : versioning, access control, observation, ...

Semi-generic editor is relatively easy to implement.

Workflow instances

📁 /var/workflow/instances
   📁 year2012
      📁 month03
         📁 day24
            🔶 wf_45f23c
               modelPath : /etc/models/approve_content
            🔶 wf_45f23d

         📁 day25
            🔶 wf_6534ef
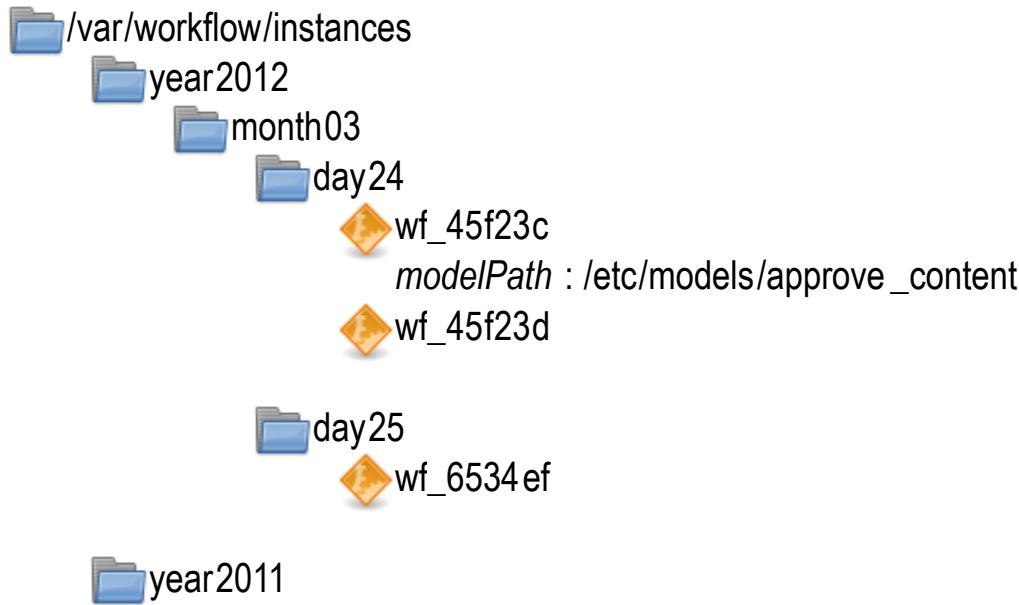
   📁 year2011

Under /var for distinct access control and lifecycle.

Purge old workflows? Delete year2011 folder.

Ingesting assets

📁 /var/public/incoming
    🔶 some_image.jpg

    🔶 invalid.pdf

📁 /var/assets/to-validate
    🔶 image-to-check.jpg

📁 /content/assets
    📁 year2012
        📁 month04
            🔶 validated_image.jpg

📁 /var/readonly/rejected
    🔶 other_invalid.pdf
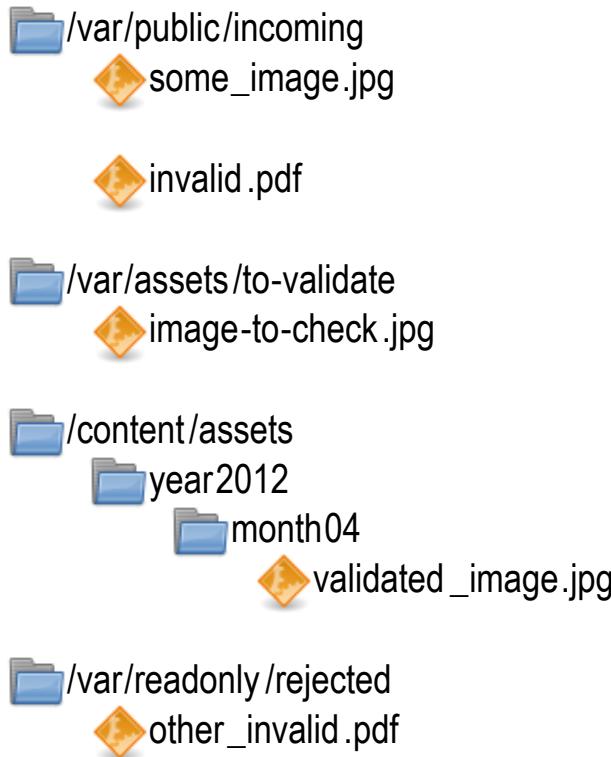
Watch the incoming folder using JCR observation

Validate, accept and route assets based on content, filenames, etc.

Only incoming folder needs to be writable by users.

Slingbucks sample: ordering coffee

📁 /content/slingbucks
   📁 private
      📁 confirmed-orders
         🔶 order_b45627

         🔶 order_f32136

   📁 public
      🔶 order_c325421
   *coffeeType* : espresso
        cupSize : tall
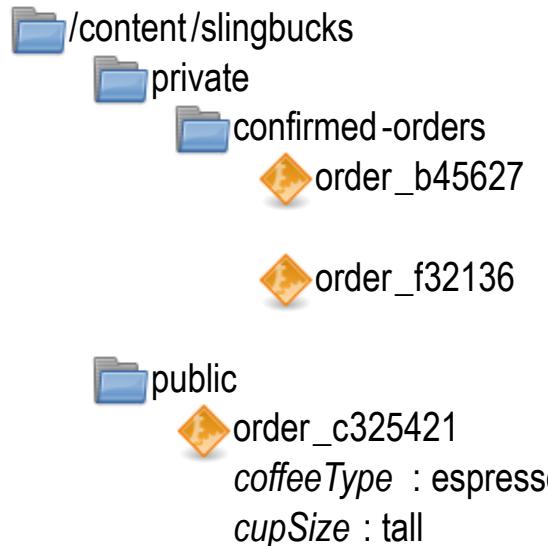
Access control requirements for the private/public folders is obvious.

Orders move from public to private, as if passed over the counter (using JCR observation).

Slingbucks sample: coffee options

📁 /content/slingbucks
   📁 coffee-type
      🔸 capuccino

      🔸 espresso

   📁 cup-type
      🔸 small

      🔸 medium
        price_factor : 1.4
        name_fr : Moyenne
        name_en : Medium

Options are defined in content, updates are dynamically applied.

Access control defines who can modify options.

Thanks for your attention - I'm @bdelacretaz