

**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 23-25 SEPTEMBER 2013

## Lightning Talk: Superimposing Content

Julian Sedding (Code Distillery GmbH)  
Stefan Seifert (pro!vision GmbH)

# Use case 1: Multi Language Scenario

- Website with multiple languages
- Allow for partial/gradual translation
- Fallback to a master version, e.g. to the English version

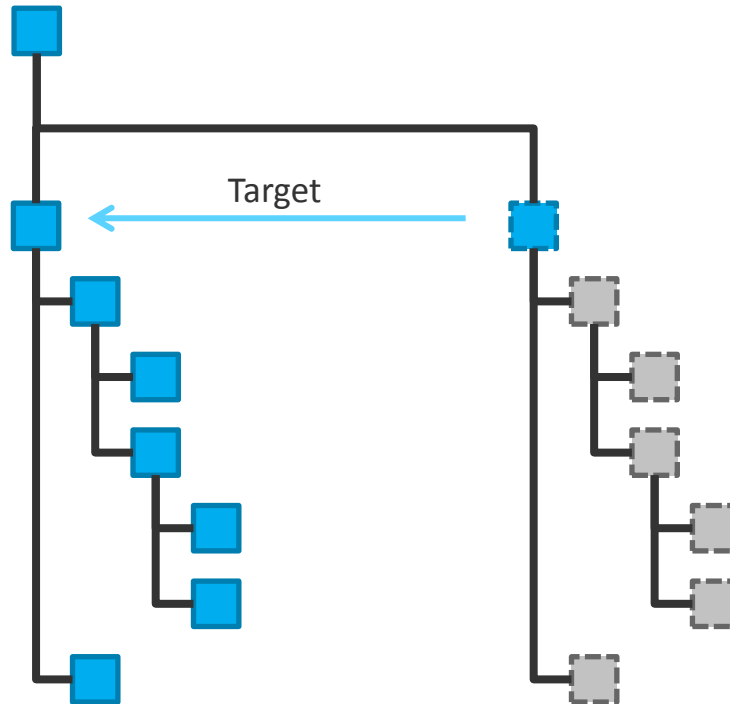
## Use case 2: Heavy Multi Site Scenario

- 1 Master site with ~500 pages having each ~15 nodes = 7,500 Nodes
- Rollout this site to 1,000 tenants
- Content is 95% identical for each tenant, but rendered differently (e.g. company name and logo)
- You do not want to create 7.5mio unnecessary nodes in your repository

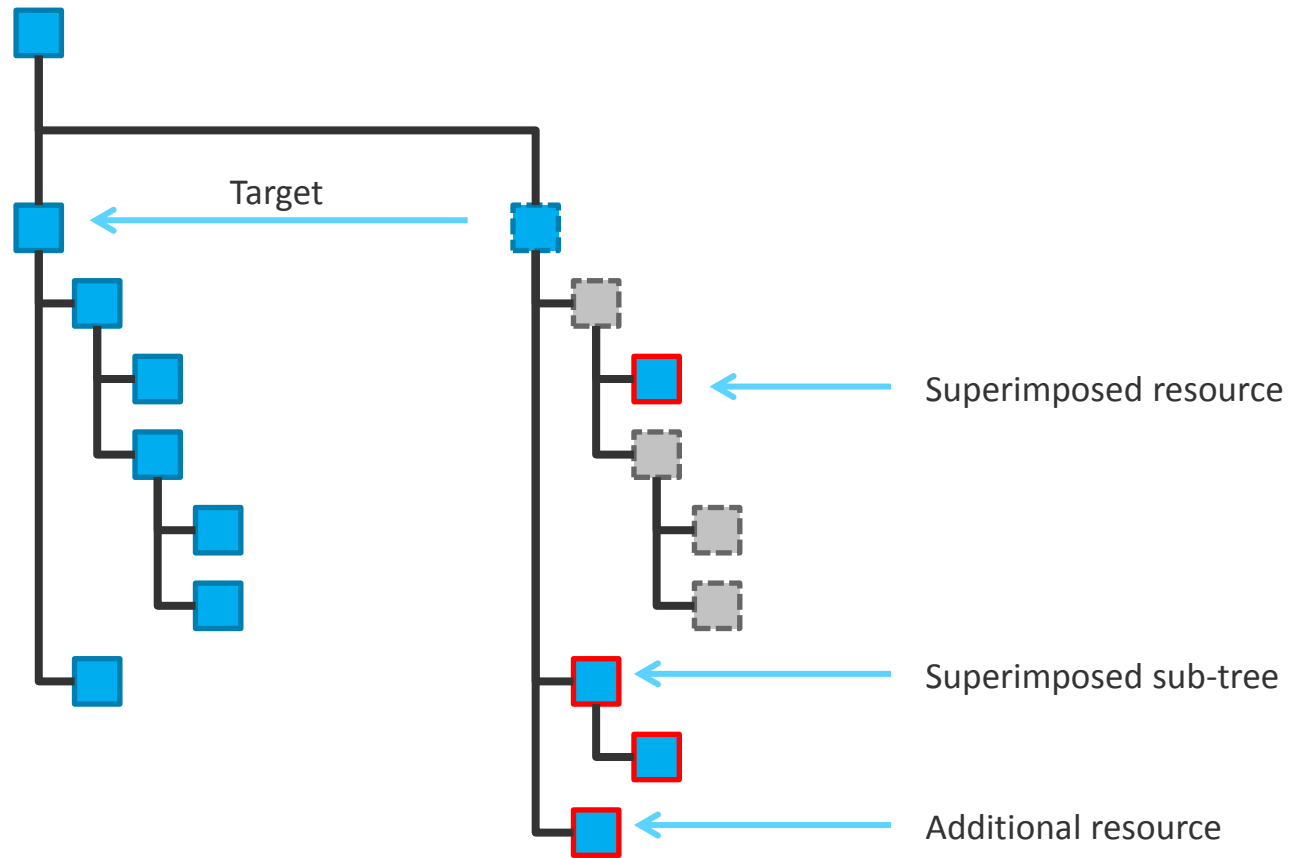
# Requirements for the solution

- **Mirroring resource trees**
  - Reflect changes from master tree
  - Avoid unnecessary copies
  
- **Superimposing resources**
  - Add
  - Remove
  - Overlay

# Mirroring resource trees



# Superimposing resources



# ResourceProvider API

```
package org.apache.sling.api.resource;
```

```
public interface ResourceProvider {
```

```
    /**  
     * Returns a resource from this resource provider or null if  
     * the resource provider cannot find it. The path should have one of the #ROOTS  
     * strings as its prefix.  
     */
```

```
    Resource getResource(ResourceResolver resourceResolver, String path);
```

```
    /**  
     * Returns an Iterator of Resource objects loaded from  
     * the children of the given Resource. The returned Resource instances  
     * are attached to the same  
     * ResourceResolver as the given parent resource.  
     */
```

```
    Iterator<Resource> listChildren(Resource parent);
```

```
}
```

# DEMO



- Sling ticket with patch  
<https://issues.apache.org/jira/browse/SLING-1778>
- Proposed name:  
SuperimposingResourceProvider

What do you think?

Thank you for listening  
and enjoy adaptTo() 2013.

# Why not JCR Shareable Nodes

- JCR Shareable Nodes allowing multiple parent nodes, thus turning the tree into a graph
- There is no “master” parent node
- Only identical content can be reused, no superimposed modifications possible

# Ideas for future enhancements

- Get rid of JCR dependency
- Automatically create structure nodes when superimposing deep nodes
- Ordering of superimposed nodes
- Support editing UIs – mirrored content is read-only