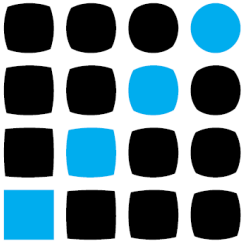


adaptTo()

APACHE SLING & FRIENDS TECH MEETUP

BERLIN, 26-28 SEPTEMBER 2016

Sling Context-Aware Configuration

Stefan Seifert, pro!vision GmbH

About the Speaker

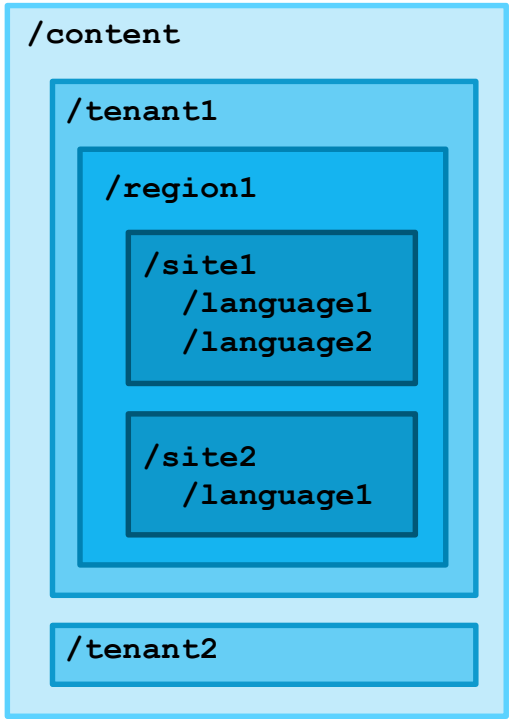
- AEM Developer
- Apache Sling PMC
- CTO of pro!vision GmbH

PRO!VISION
SOFTWARE CRAFTSMANSHIP

https://www.pro-vision.de

What is Context-Aware Configuration?

Configuration example



Tenant-specific configuration

Region-specific configuration

Site-specific configuration

Context-aware = **different configuration for different subtrees in resource hierarchy**

adaptTo()

Context-aware OSGi configuration?

- OSGi configuration is targeted to system-wide configuration
- Difficult to store context-aware configuration
- e.g. AEM Link Externalizer
  → not a good solution

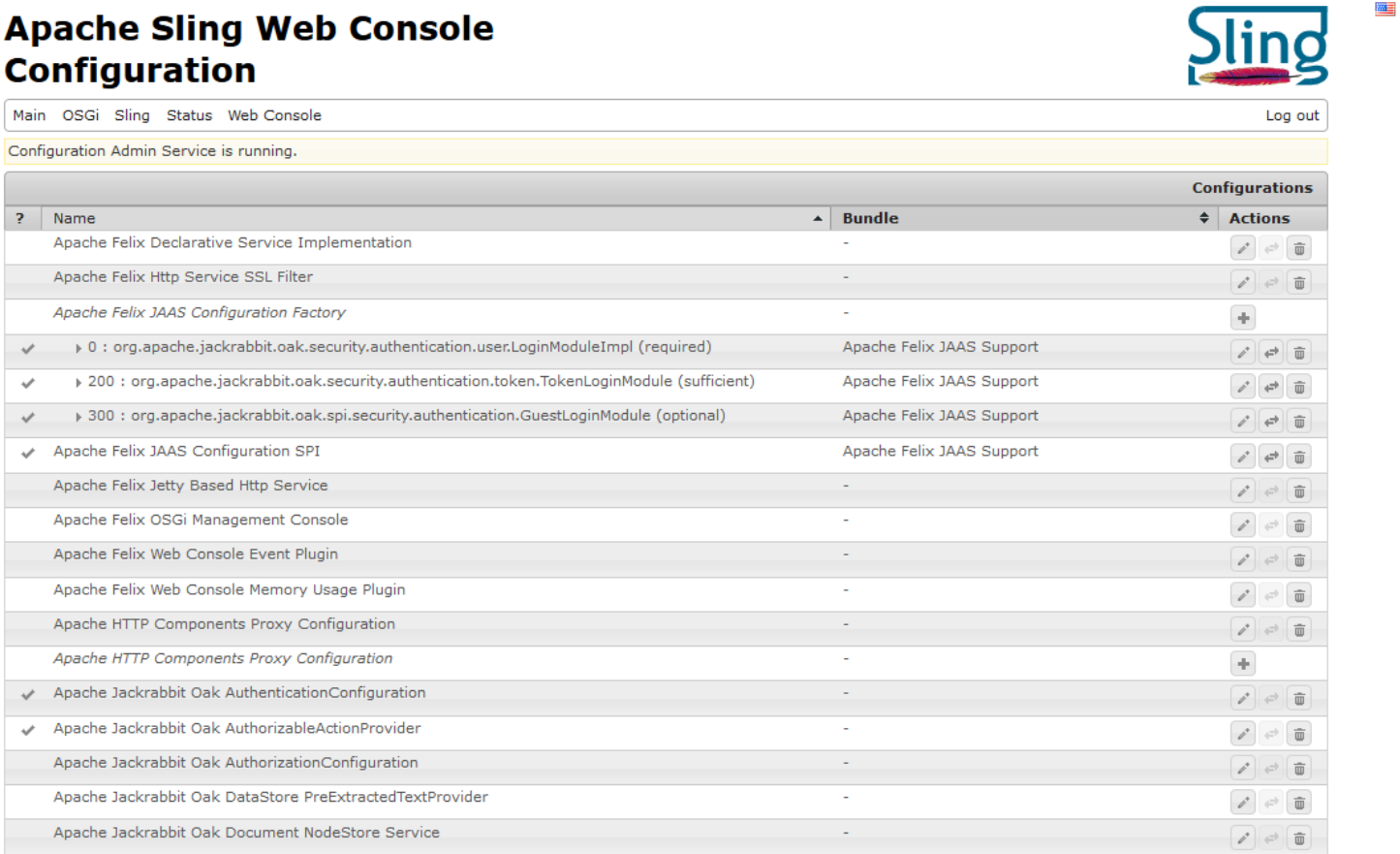
Top-level requirements

- Store configuration per context
- Flexible definition of “context”
- Inheritance and fallback config
- Simple API for accessing configuration
- Configuration GUI Editor

Existing Solutions

FreeImages.com/Robert Driese

OSGi for System-level config



Apache Sling Web Console Configuration

Main OSGi Sling Status Web Console Log out

Configuration Admin Service is running.

		Configurations	
?	Name	Bundle	Actions
	Apache Felix Declarative Service Implementation	-	
	Apache Felix Http Service SSL Filter	-	
	Apache Felix JAAS Configuration Factory	-	
✓	0 : org.apache.jackrabbit.oak.security.authentication.user.LoginModuleImpl (required)	Apache Felix JAAS Support	
✓	200 : org.apache.jackrabbit.oak.security.authentication.token.TokenLoginModule (sufficient)	Apache Felix JAAS Support	
✓	300 : org.apache.jackrabbit.oak.spi.security.authentication.GuestLoginModule (optional)	Apache Felix JAAS Support	
✓	Apache Felix JAAS Configuration SPI	Apache Felix JAAS Support	
	Apache Felix Jetty Based Http Service	-	
	Apache Felix OSGi Management Console	-	
	Apache Felix Web Console Event Plugin	-	
	Apache Felix Web Console Memory Usage Plugin	-	
	Apache HTTP Components Proxy Configuration	-	
	Apache HTTP Components Proxy Configuration	-	
✓	Apache Jackrabbit Oak AuthenticationConfiguration	-	
✓	Apache Jackrabbit Oak AuthorizableActionProvider	-	
	Apache Jackrabbit Oak AuthorizationConfiguration	-	
	Apache Jackrabbit Oak DataStore PreExtractedTextProvider	-	
	Apache Jackrabbit Oak Document NodeStore Service	-	

OSGi for System-level config

- Editor GUI
- Flexible deployment: filesystem, repository, web console, factory configurations
- “Self-describing" with metadata
- Good API support (esp. in OSGi R6)
- Runmode-specific configuration

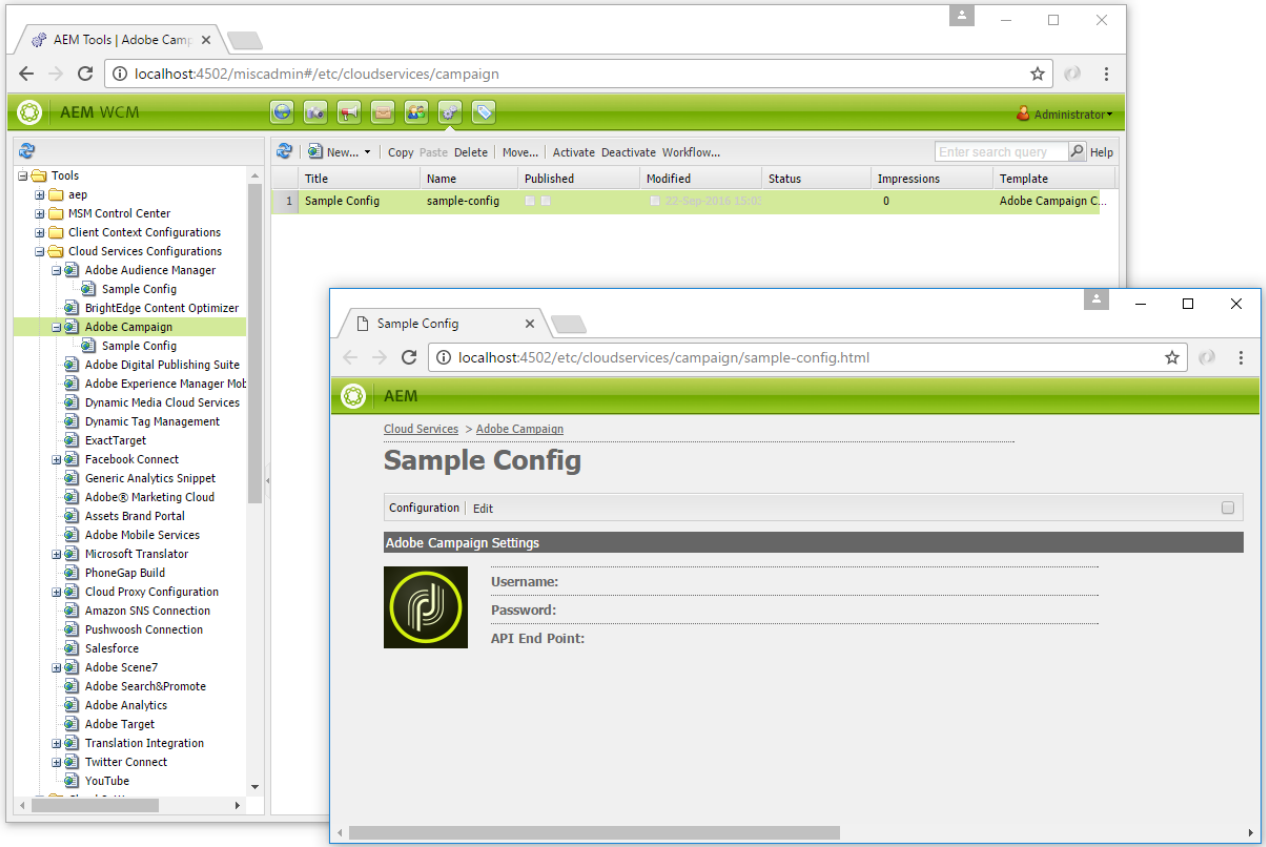
Existing Solutions for Context-Aware Config

- In Sling: none
- In AEM:
  - AEM ConfMgr (since AEM 6.1)
  - Cloud Service Configurations a.k.a. CSC (since CQ 5.5)
  - wcm.io Configuration

AEM ConfMgr

- Simple API
- Flexible inheritance support
- No Editor GUI
- Lacks documentation
- Storage: `/conf`

Cloud Service Configurations (CSC)



Cloud Service Configurations (CSC)

- Edit configuration via AEM templates
- Primary target: Adobe Marketing Cloud integrations
- Custom configurations possible as well
- Storage: `/etc/cloudservices`

wcm.io Config

- API and SPI for managing context-aware configurations
- Pluggable architecture
- Editor GUI

Configuration Hierarchy

Feature	OSGi Config	AEM ConfMgr	AEM CSC	wcm.io Config
Global / Fallback configuration	✓	✓	✗	✓
Hierarchy-based inheritance	✗	✓	✗	✓
Property inheritance merging	✗	✗	✗	✓

Provide Configuration Metadata

Feature	OSGi Config	AEM ConfMgr	AEM CSC	wcm.io Config
Provide Parameters and data types	✓	✗	✓	✓
Additional metadata for Editors	✓	✗	✓	✓
Define Configuration metadata in Java Code	✓	✗	✗	✗

Accessing configuration – API

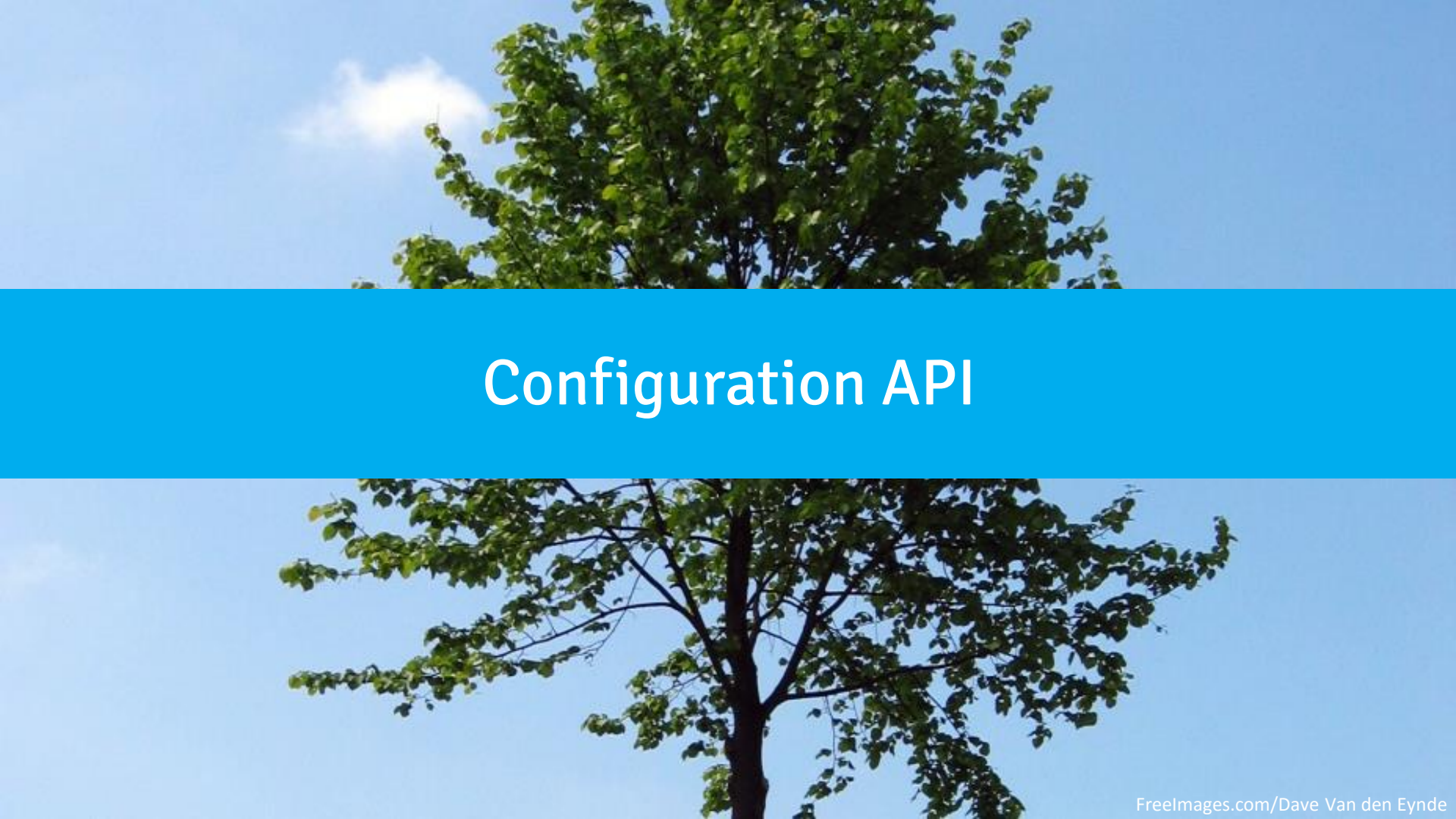
Feature	OSGi Config	AEM ConfMgr	AEM CSC	wcm.io Config
Key/value pairs (ValueMap)	✓	✓	✓	✓
Resource-based access	✗	✓	✓	✗
Map to Java class	✓	✗	✗	✗
Configuration Lists (like OSGi Factory Configs)	✓	✓	✗	✗

Managing configuration

Feature	OSGi Config	AEM ConfMgr	AEM CSC	wcm.io Config
Editor GUI	✓	✗	✓	✓
Web console (debugging)	✓	✓	✗	✗
Override parameters	✗	✗	✗	✓
Lock parameters	✗	✗	✗	✓
Config Change Listener	✓	✗	✗	✗

Existing solutions

- None of the existing solutions fulfills all needs
- Some are closed source
- We need a solution in Apache Sling that can fulfill all requirements


Configuration API
FreeImages.com/Dave Van den Eynde

API Design

- ConfigurationResourceResolver
  - „Low-level“ API for resource-based configuration access
  - Can be used for any resource-like configuration e.g. workflow definitions, policies etc.
  - Normally not used by Applications directly

API Design

- ConfigurationResolver
  - High-level access to configuration data
  - Simple API for reading config
    as ValueMap or mapped to a Java class
  - Define parameter metadata
    via annotation classes

Configuration Builder API

- ## Configuration as Value Map

```
// resource of the current content tree/context
ValueMap props = resource.adaptTo(ConfigurationBuilder.class).asValueMap();
String param1 = props.get("param1", String.class);
```

- ## Configuration mapped to class

```
MyConfig config = resource.adaptTo(ConfigurationBuilder.class).as(MyConfig.class);
String param1 = config.param1();
```

Define parameters via annotation class

```
@Configuration
public @interface MyConfig {

    String param1();

    String paramWithDefault() default "defValue";

    int intParam();

}
```

Additional metadata

```
@Configuration(name="myconfig", label="My Configuration", description="Describe me")
public @interface MyConfig {

    @Property(label="Parameter #1", description="Describe me")
    String param1();

    @Property(label="Parameter with Default value", description="Describe me")
    String paramWithDefault() default "defValue";

    @Property(label="Integer parameter", description="Describe me",
            property={
                    "guiWidget=dropdown",
                    "dropdownValues=1,2,3"
            })
    int intParam();

}
```

Configuration Builder API

- Configuration collections

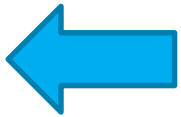
```
Collection<MyConfig> configs = resource
    .adaptTo(ConfigurationBuilder.class)
    .asCollection(MyConfig.class);

for (MyConfig config : configs) {
    // ...
}
```

Content model and inheritance

FreeImages.com/José A. Warletta

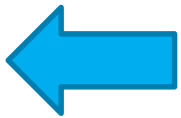
Default content model

```
/content
    /site1
      @sling:config-ref = "/conf/site1"
        /page1
```



Context / Content resources

```
/conf
    /site1
        /sling:settings
            x.y.z.MyConfig
              @param1 = "value1"
```

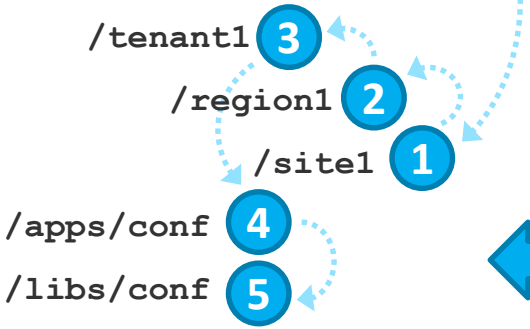


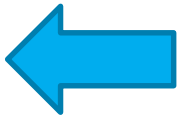
Configuration resources

```
MyConfig config = resource.adaptTo(ConfigurationBuilder.class).as(MyConfig.class);
String param1 = config.param1();
```

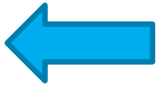
Inheritance and Fallback

```
/content
    /tenant1
      @sling:config-ref = "/conf/tenant1"
        /region1
          @sling:config-ref = "/conf/tenant1/region1"
            /site1
              @sling:config-ref = "/conf/tenant1/region1/site1"

/conf
    /tenant1
        /region1
            /site1
/apps/conf
/libs/conf
```





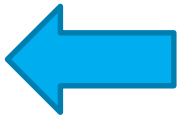
Resource lookup order
for configuration hierarchies



Fallback to `/apps/conf` and `/libs/conf`

Configuration collections

```
/content
    /site1
      @sling:config-ref = "/conf/site1"
        /page1


/conf/site1/sling:settings
    feature
        C
/conf/global/sling:settings
    feature
        A
        B
```



Results in:

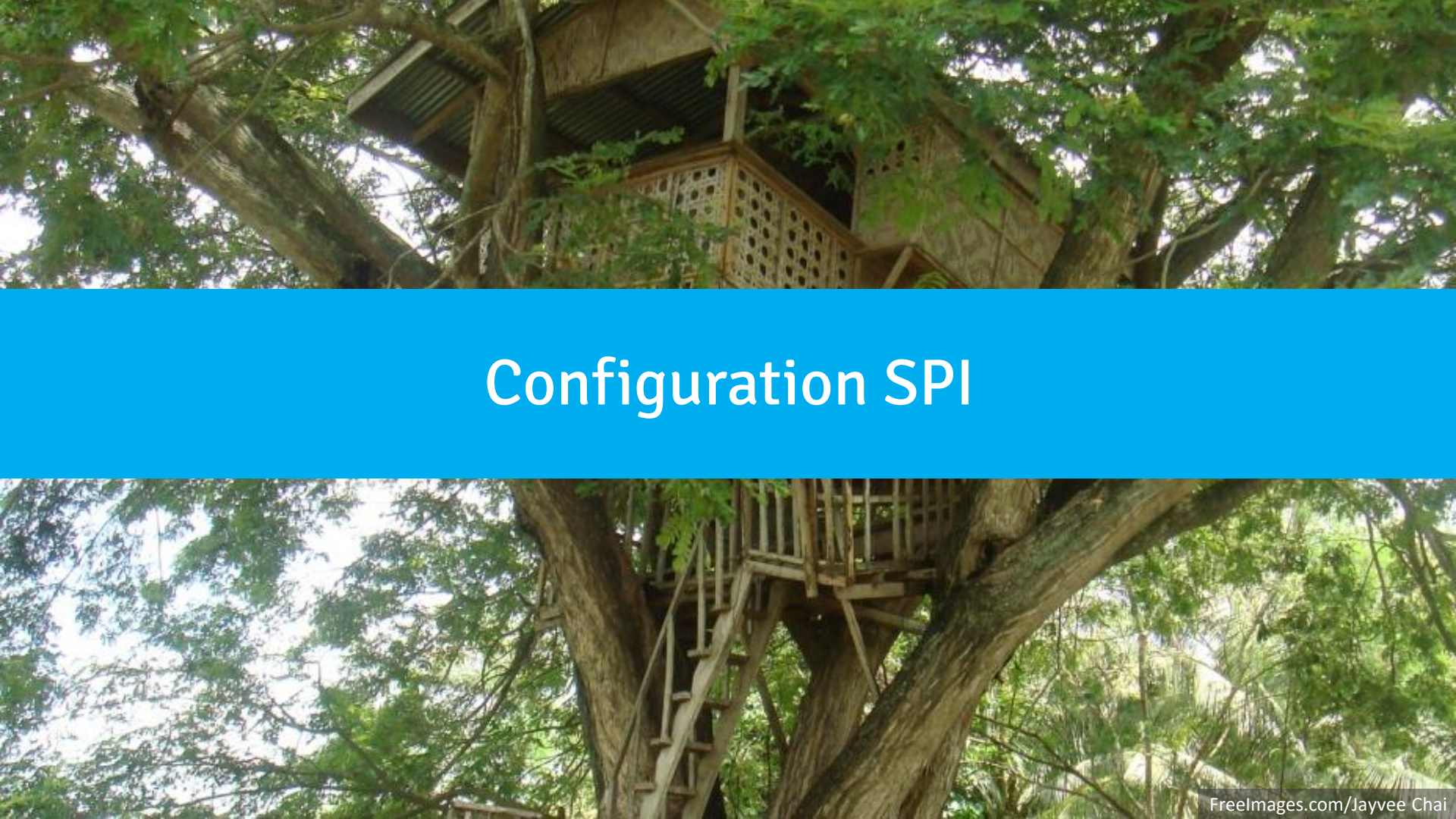
C

or

A, B, C

depending on merging configuration.

```
Collection<ValueMap> configs = resource.adaptTo(ConfigurationBuilder.class)
    .name("feature").asValueMapCollection();
```



Configuration SPI

FreeImages.com/Jayvee Chai

ConfigurationResourceResolver SPI

- ContextPathStrategy
  - Default: Detect context levels by `sling:config-ref` property
  - Allows to add other strategies, e.g.
    - detected by project-specific path conventions
    - AEM Templates

ConfigurationResourceResolver SPI

- ConfigurationResourceResolvingStrategy
  - Default: as described
  - Allows to add other strategies where configuration is stored, e.g. store non-technical “business configuration” together with content

ConfigurationResolver SPI

- ConfigurationMetadataProvider
  - Default: Read metadata from annotation classes deployed in OSGi bundles
  - bnd plugin to auto-generate list of annotation classes at build time
    - Usable with **`maven-bundle-plugin`** and **`bnd-maven-plugin`**

ConfigurationResolver SPI

- ConfigurationPersistenceStrategy
  - Default: store configuration properties directly in resource
  - Allows to add strategies e.g.
    - set specific resource types
    - use cq:Page/jcr:content nodes instead

SPI design strategy

- All SPIs share a common principle
  - Support multiple strategies at the same time
  - No need to switch off or „copy“ the initial strategy
  - Apply additional strategies only for those places where needed (“minimally invasive”)

Current status and outlook

FreeImages.com/Miguel Ugalde

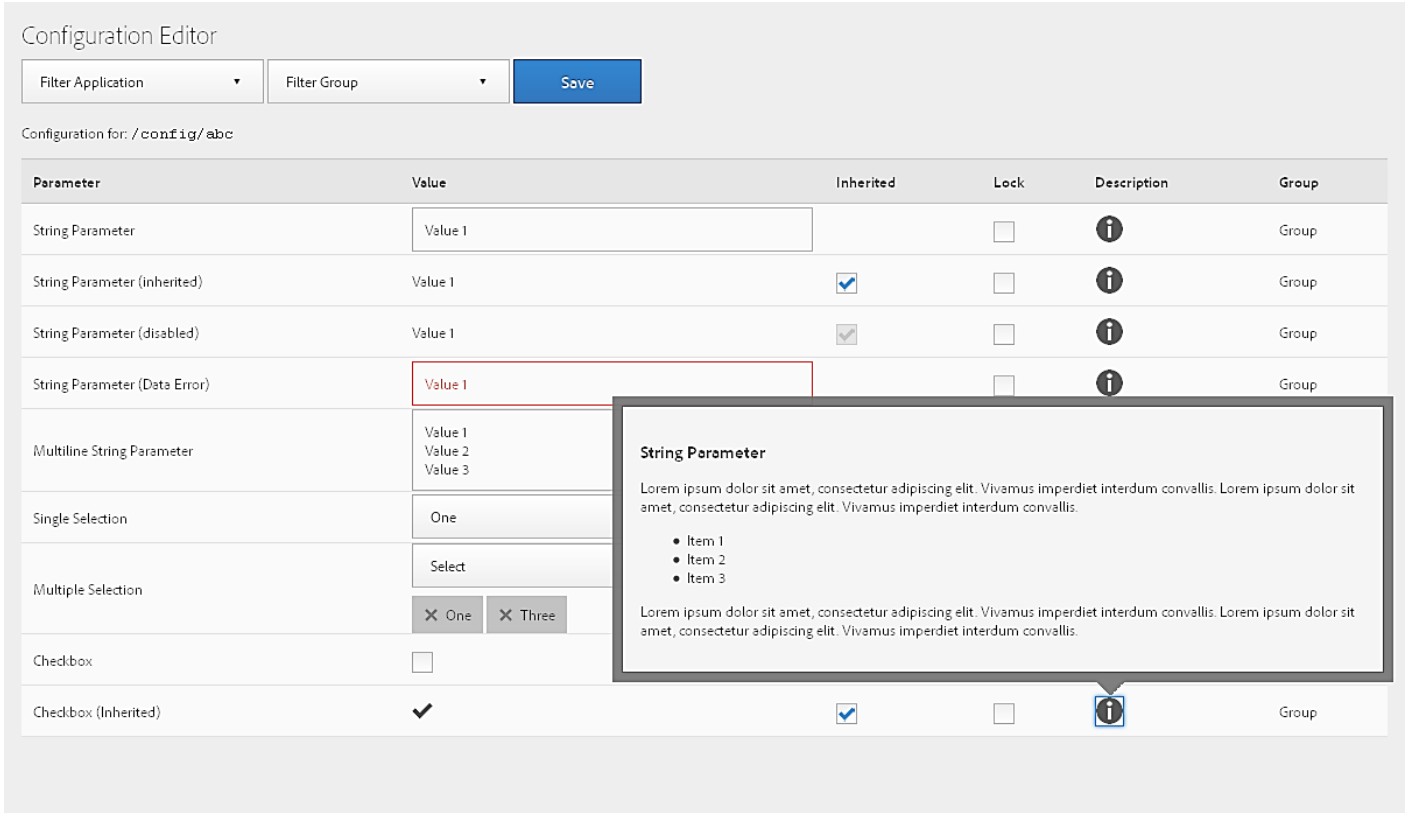
Current status

- Implementation at sling/contrib
- The described features are available with good test coverage
- 1.0 release planned soon

Outlook

- Add more features like
  - Property inheritance/merging
  - Advanced features like override, locking
- wcm.io Config 2.0
  - Migrate to Sling Context-Aware Config
  - Provide AEM-based Editor GUI

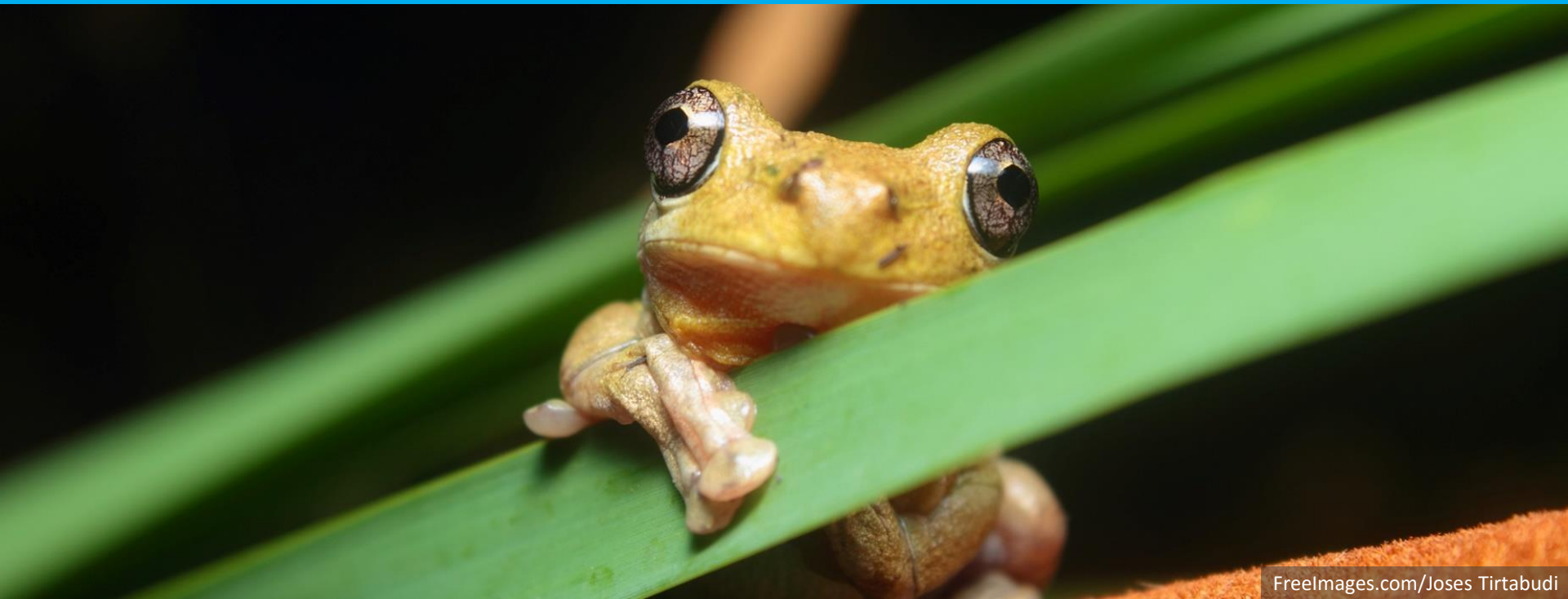
wcm.io Configuration Editor (today)



References

- Sling Context-Aware Configuration
  https://github.com/apache/sling/tree/trunk/contrib/extensions/contextaware-config
- Existing AEM configuration solutions + use case comparison
  https://wcm-io.atlassian.net/wiki/x/BwBLAQ
  https://wcm-io.atlassian.net/wiki/x/BQBLAQ
- AEM ConfMgr
  http://www.nateyolles.com/blog/2016/03/aem-slash-conf-and-confmgr
- AEM Cloud Service Configurations
  https://docs.adobe.com/docs/en/aem/6-2/develop/extending/cloud-service-configurations.html
- wcm.io Configuration
  http://wcm.io/config/

Questions?



FreeImages.com/Joses Tirtabudi