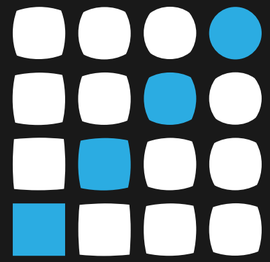


EUROPE'S LEADING AEM DEVELOPER CONFERENCE

28th - 30th SEPTEMBER 2020

Escape the defaults

Configuring Sling to behave like AEM as a Cloud Service

Robert Munteanu, Adobe

Slides revision: 20200827-b88ebd4

adaptTo()

About me

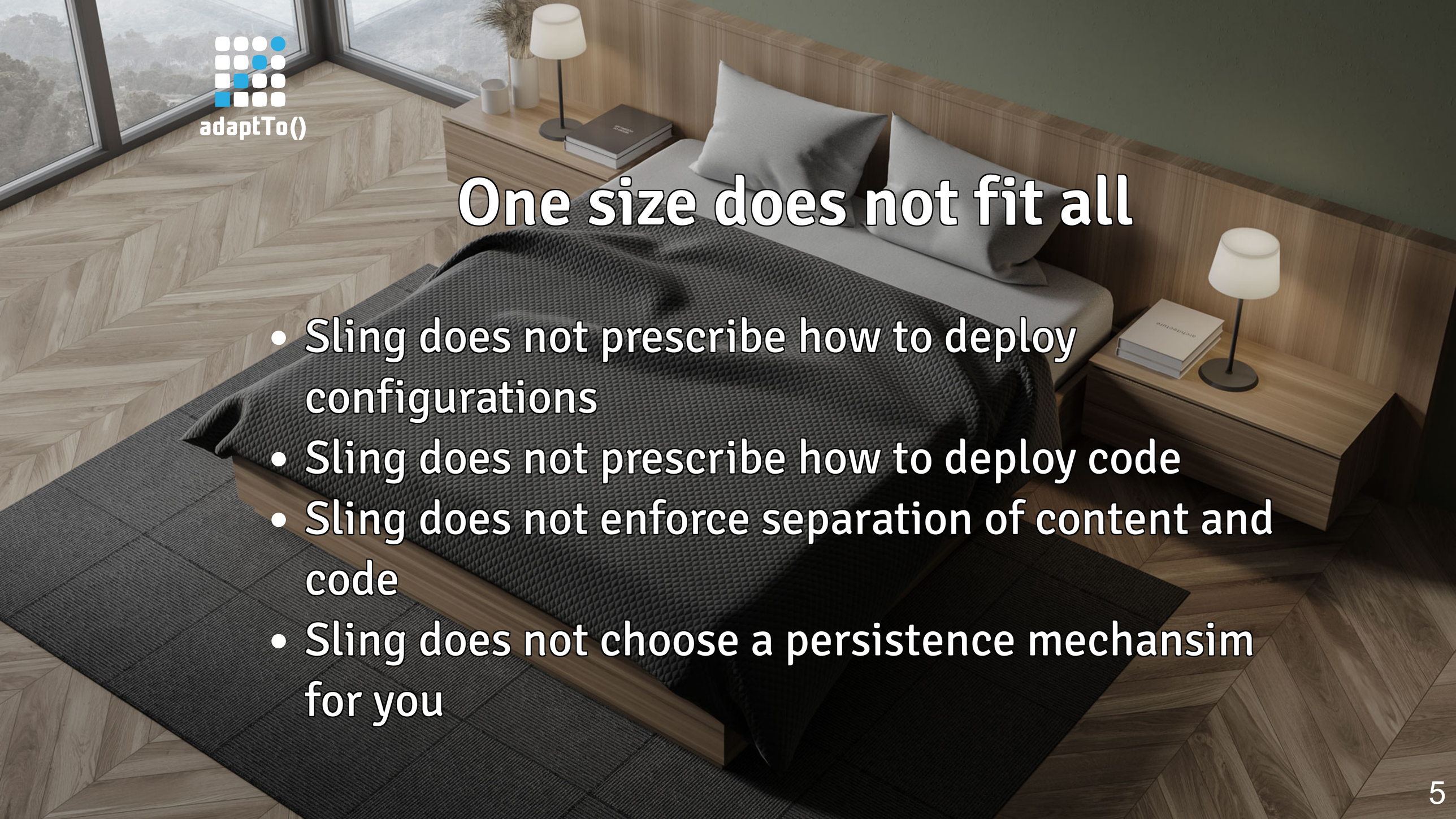
Sling™

@rombert

Outline

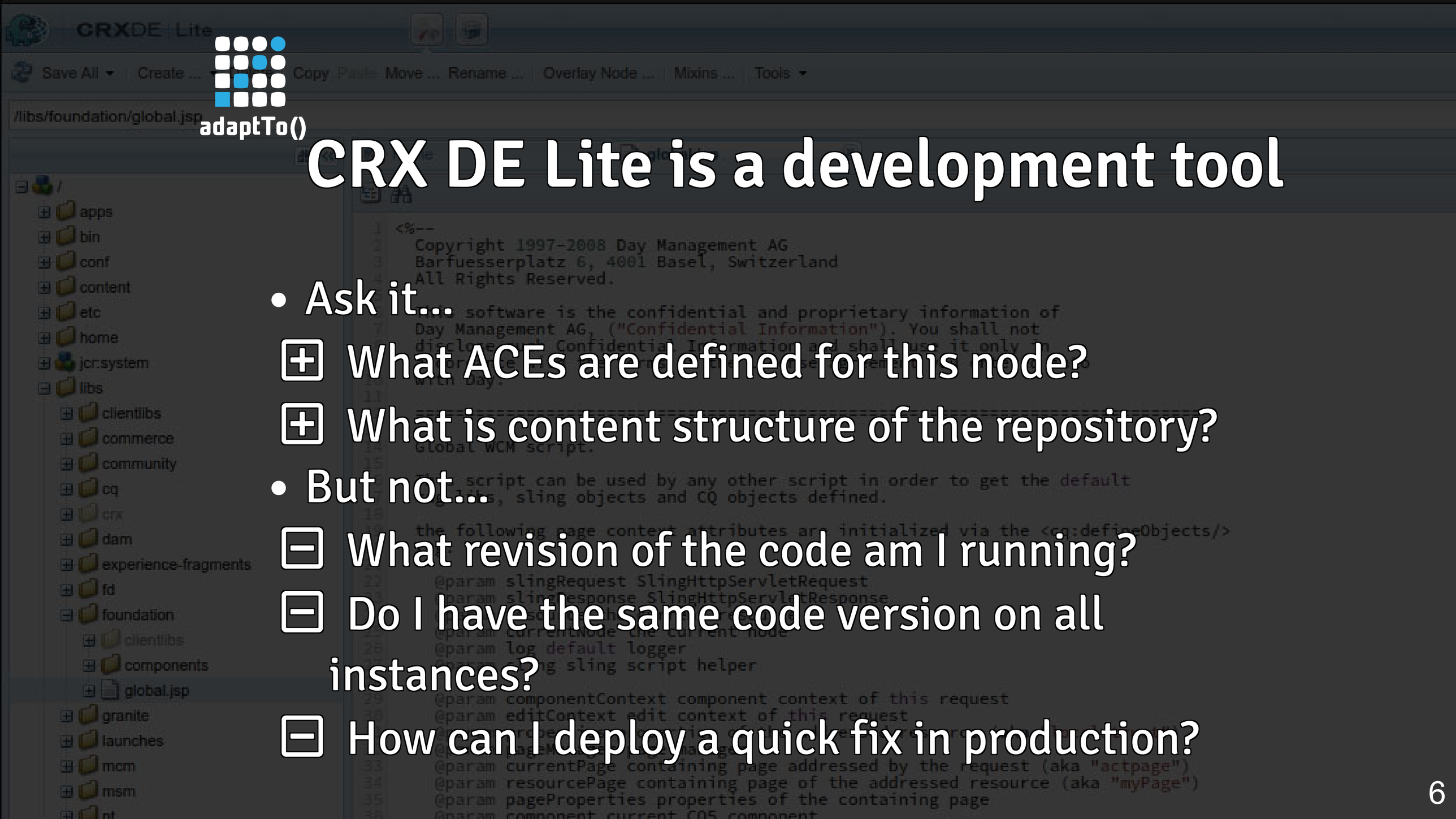
- Why change the defaults?
- Separation of content and apps
- Principal-based authentication
- Pre-authenticating system users
- Removing the OSGi installer
- Demo

Why change the defaults?



One size does not fit all

- Sling does not prescribe how to deploy configurations
- Sling does not prescribe how to deploy code
- Sling does not enforce separation of content and code
- Sling does not choose a persistence mechansim for you



CRX DE Lite is a development tool

- Ask it…
  - ⊞ What ACEs are defined for this node?
  - ⊞ What is content structure of the repository?
- But not…
  - ⊟ What revision of the code am I running?
  - ⊟ Do I have the same code version on all instances?
  - ⊟ How can I deploy a quick fix in production?

Better ways of asking ...

```
$ kubectl describe deployment sling-starter
Containers:
  main:
    Image:        apache/sling-starter:1.0-cafebafe

$ kubectl apply -f deployments/sling-starter.yaml
```

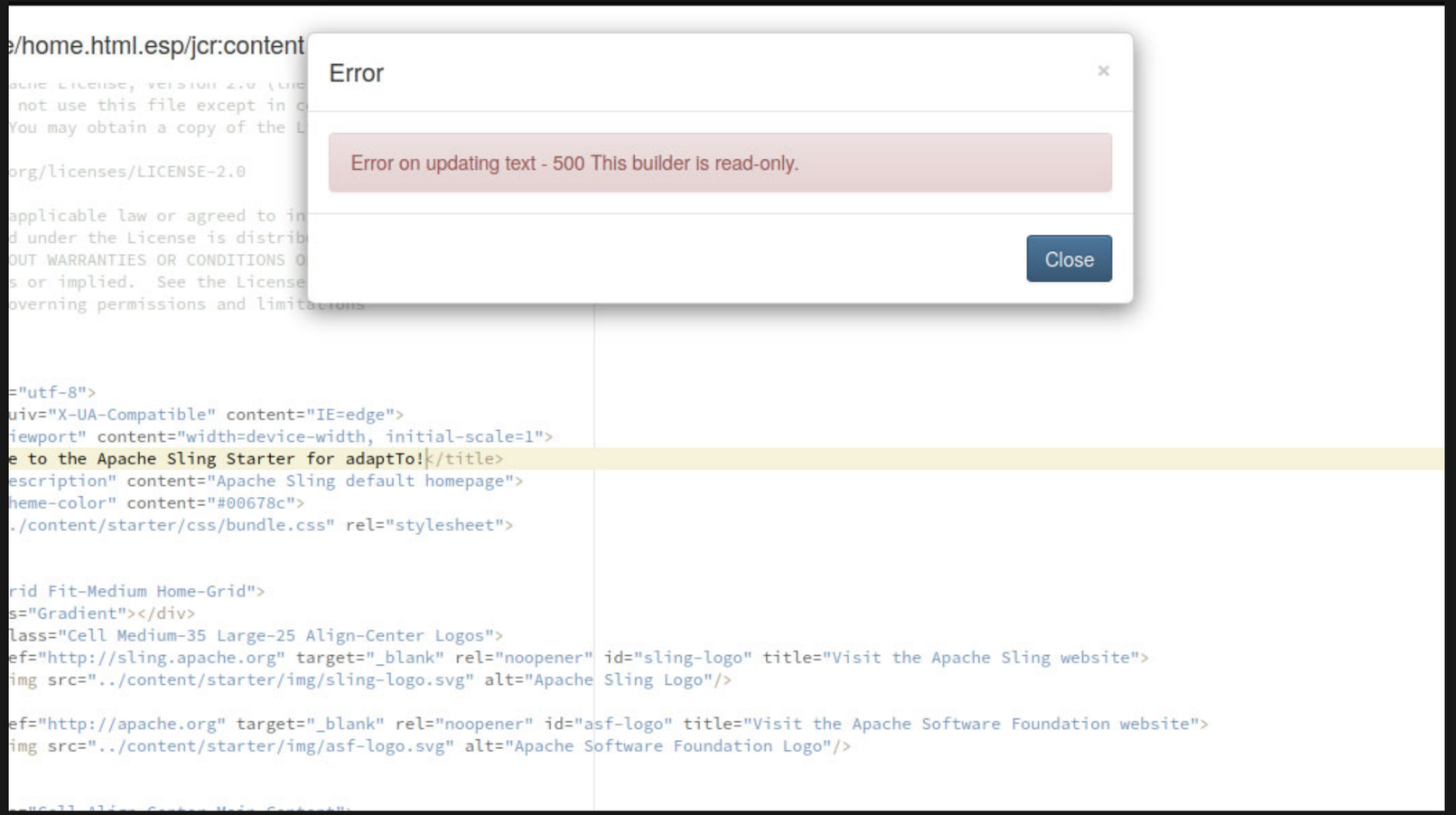
```
$ rpm -q sling-starter
sling-starter-1.0_cafebabe-1.noarch

$ rpm -Uvh sling-start-1.1_sodadude-1.noarch
```

```
$ curl http://sling-starter.example.org/version
1.0-cafebabe
```

adaptTo()

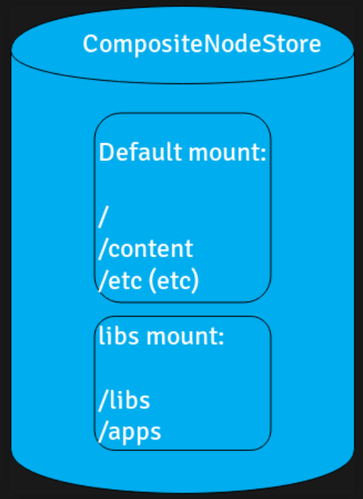
Prevent mistakes



Separation of content and apps

adaptTo()

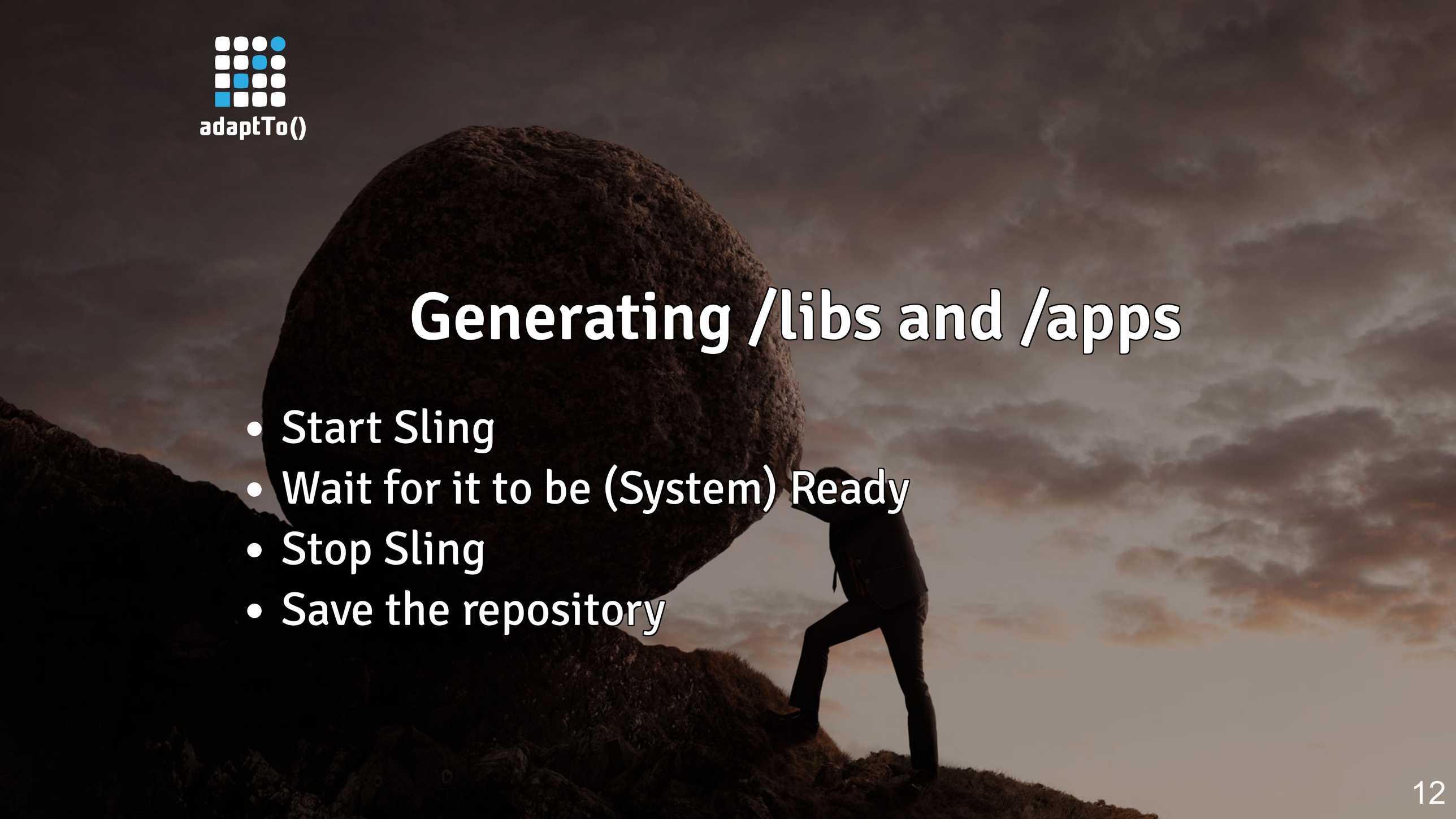
The composite node store



- Clear separation of code from content
- Enforces read-only status of code and (mostly) configuration
- Allows swapping in a separate node store (with restart)

Generating /libs and /apps

- Must be a NodeStore
- Must reflect the state of the deployment:
  - Nodes
  - Principals
  - Indexes
  - Access Control entries
- Coming from:
  - Repoinit
  - Content Packages
  - Sling Content Loader
  - Java code - Bundle Activator, Install Hooks, ...

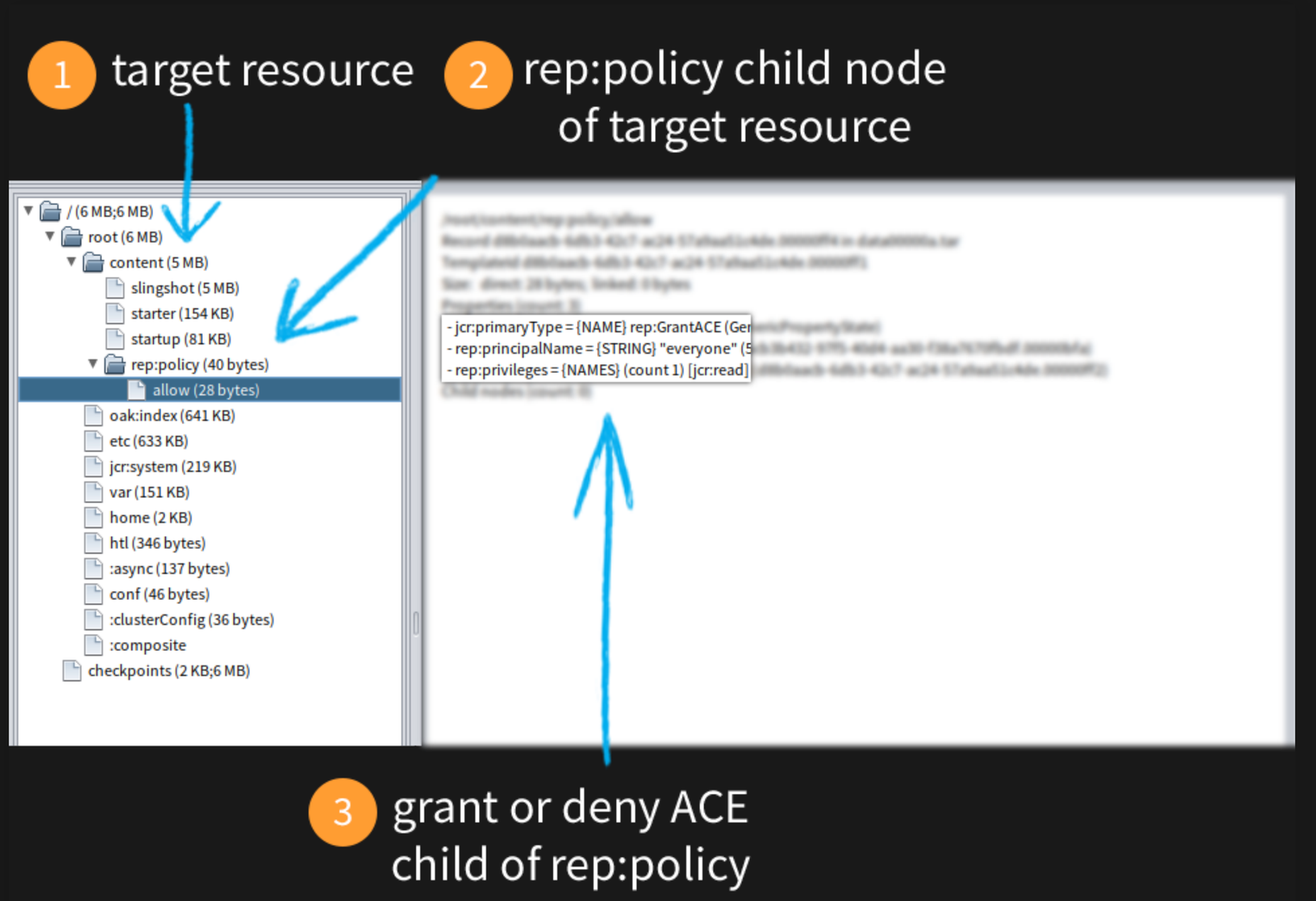


Generating /libs and /apps

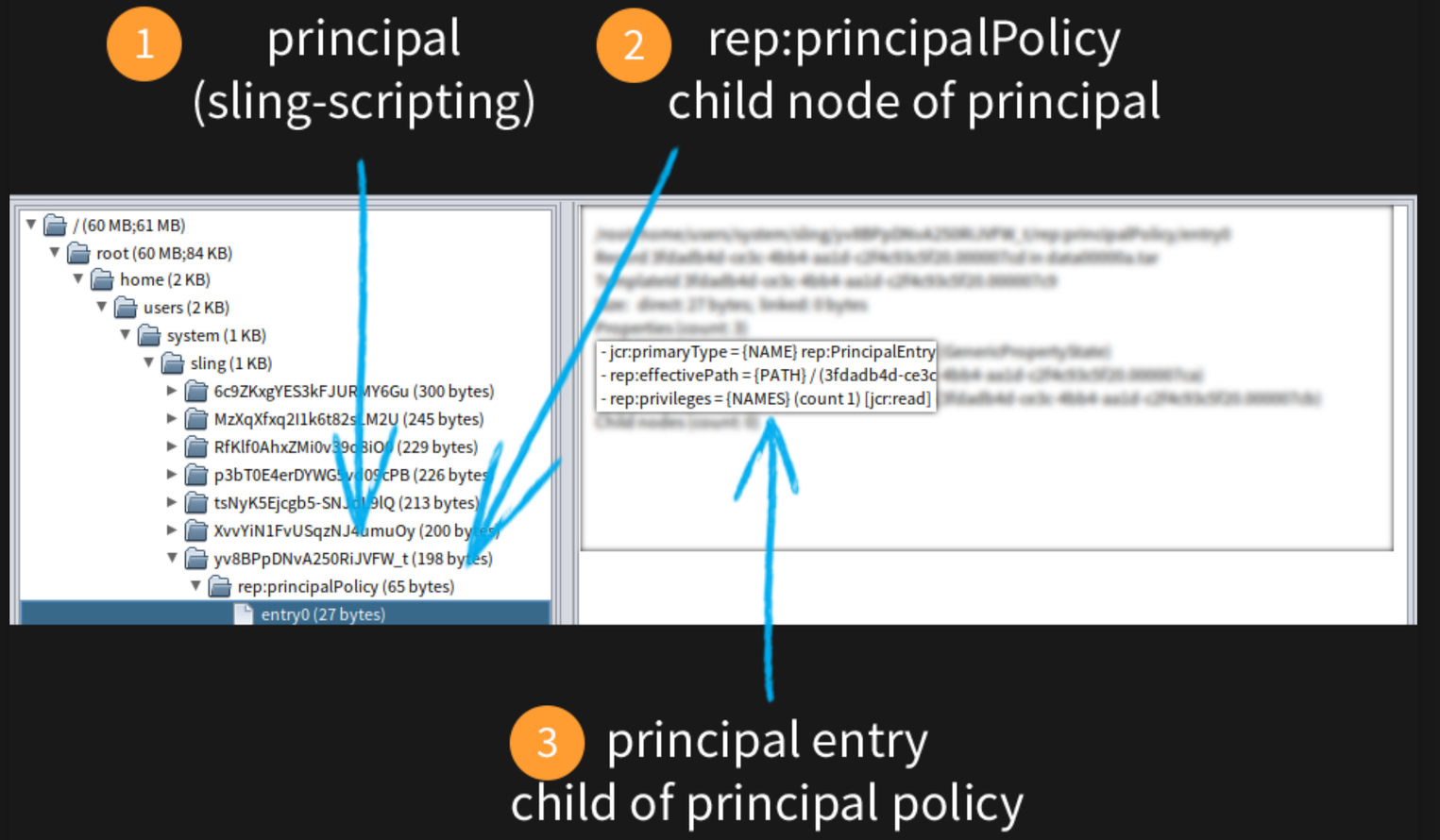
- Start Sling
- Wait for it to be (System) Ready
- Stop Sling
- Save the repository

Principal-based authentication

Resource-based authentication



Principal-based authentication



Benefits

- Easy to inspect access control entries for a given principal
- Access control entries independent of the existence of their target
- Much simpler packaging story in content packages

Usage

FileVault

- `rep:PrincipalBasedMixin`
- `rep:PrincipalPolicy`
- `rep:PrincipalEntry`

Repoinit

```
create service user sling-readall with path system/sling

set principal ACL for sling-readall
    allow jcr:read on /
end
```

Pre-authenticating system users

Reminder: loginAdministrative

loginAdministrative

```
@Deprecated
javax.jcr.Session loginAdministrative(String workspace)
                                      throws javax.jcr.LoginException,
                                             javax.jcr.RepositoryException
```

Deprecated. *as of 2.2 (bundle version 2.2.0) because of inherent security issues. Services requiring specific permissions should use the `loginService(String, String)` instead.*

Returns a session to the given workspace which has administrative powers.

NOTE: This method is intended for use by infrastructure bundles to access the repository and provide general services. This method MUST not be used to handle client requests of whatever kinds. To handle client requests a regular authenticated session retrieved through `Repository.login(javax.jcr.Credentials, String)` or `Session.impersonate(javax.jcr.Credentials)` must be used.

This method is deprecated. Services running in the Sling system should use the `loginService(String serviceInfo, String workspace)` method instead. Implementations of this method must throw `javax.jcr.LoginException` if they don't support it.

Parameters:

`workspace` - The name of the workspace to which to get an administrative session. If null the `getDefaultWorkspace()` default workspace is assumed.

Returns:

The administrative Session

Throws:

`javax.jcr.LoginException` - If this method is not supported or is disabled by the implementation.

`javax.jcr.RepositoryException` - If an error occurrs creating the administrative session

Reminder: service user mappings

```
// uses bundle name, no subservice, default workspace
slingRepository.loginService(null, null);

// uses bundle name, 'scripts' subservice, default workspace
slingRepository.loginService("scripts", null);
```

```
"o.a.s.....ServiceUserMapperImpl.amended~i18n":{
  "user.mapping":[
    "org.apache.sling.i18n=sling-i18n"
  ]
},
"o.a.s.....ServiceUserMapperImpl.amended~servletsresolver":{
  "user.mapping":[
     "o.a.s.servlets.resolver:console=sling-readall",
     "o.a.s.servlets.resolver:scripts=sling-scripting"
  ]
}
```

Duplicated system user privileges

```
set ACL for sling-mapping
    allow jcr:read on /
end
```

```
set ACL for sling-i18n
    allow jcr:read on /
end
```

```
set ACL for sling-jcr-install
    allow jcr:read on /
    allow rep:write on /apps/sling/install
end
```

Pre-authenticated login

- Supply all principals at login in time
- Faster by skipping authentication
- Allows mapping a service to multiple users

Pre-authenticated system users

```
create service user sling-readall
set ACL for sling-readall
    allow jcr:read on /
end
create service user sling-jcr-install
set ACL for sling-jcr-install
    allow rep:write on /apps/sling/install
end
```

```
"o.a.s.....ServiceUserMapperImpl.amended~i18n":{
    "user.mapping":[
        "org.apache.sling.i18n=[sling-readall]"
    ]
}, "o.a.s.....ServiceUserMapperImpl.amended~~jcr-install":{
    "user.mapping":[
        "o.a.s.installer.provider.jcr=[sling-jcr-install,⏎
            sling-readall]"
    ]
}
```

Removing the OSGi installer

The OSGi installer...

- installs
  - bundles
  - configurations
  - content packages
- from
  - filesystem
  - launchpad
  - JCR repository

The OSGi installer...

- is a dynamic component
- reconciles application state from multiple sources
- is not needed in a statically defined/immutable application
  - ... we are buiding immutable applications

FM: Installing bundles

```
{
  "bundles":[
    {
      "id":"org.apache.aries:org.apache.aries.util:1.1.3",
      "start-order":"1"
    }
  ]
}
```

FM: Installing configurations

```
{
  "configurations": {
    "o.a.s.jcr.davex.impl.servlets.SlingDavExServlet":{
      "alias":"/server"
    }
  }
}
```

FM: Installing content packages

```
{
  "content-packages:ARTIFACTS|true": [
    "o.a.s:sling-slingshot-apps-pkg:zip:1.0-SNAPSHOT",
    "o.a.s:sling-slingshot-content-pkg:zip:1.0-SNAPSHOT"
  ]
}
```

⚠ content packages are by default forwarded to the OSGi installer

```
<workspaceFilter version="1.0">
    <filter root="/libs/slingshot/config"/>
    <filter root="/apps/bundles/install/bundle.jar"/>
</workspaceFilter>
```

Content Package to Feature Model

- Handles nested artifacts inside content packages
  - OSGi bundles
  - OSGi configurations
  - Other content packages
- Converts access control configurations
  - System Users
  - Access control entries (resource-based)
  - No support for principal-based access control entries

Conversion result

```
{
  "id":"o.a.s:sling-slingshot-apps-pkg:slingosgifeature:1.0",
  "bundles":[
    "o.a.s:o.a.s.sample.slingshot:0.9.1"
   ], "configurations":  {
    "o.a.s....ServiceUserMapperImpl.amended~slingshot": {
      "o.a.s.sample.slingshot=[slingshot-service]"
    }
  }, "content-packages:ARTIFACTS|true":[
    "o.a.s:sling-slingshot-apps-pkg:zip:cp2fm-converted:1.0"
  ], "repoinit:TEXT|true":[
    "create service user slingshot-service"
  ]
}
```

Alternative Package Registry

`o.a.j.vault....FSPackageRegistry`

- Stores packages in the filesystem
- Can be assembled without a JCR repository
- Relies on an existing execution plan
  - Plan prepared by the Sling Content Deployment Extension

Demo

Demo recap

- Sling Docker image using
  - the feature model
  - the composite node store
- Docker updates
  - done with a simple restart
  - preserving the 'content' part of the repository
- Removing the OSGi installer using the content-package converter

Resources

- (Sling) Content Package to Feature Model Converter
- (Sling) Content Deployment Extension
- (Oak) Composite Node Store
- (Oak) Principal-Based Authentication
- (Oak) Pre-Authenticated Login