APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 25-27 SEPTEMBER 2017

Future-proof content architecture designs
Florin Iordache, Adobe Systems

Agenda

- Content structure analysis in AEM

- Content categories

- Content maintenance and management

- Use case: Sling CA configurations
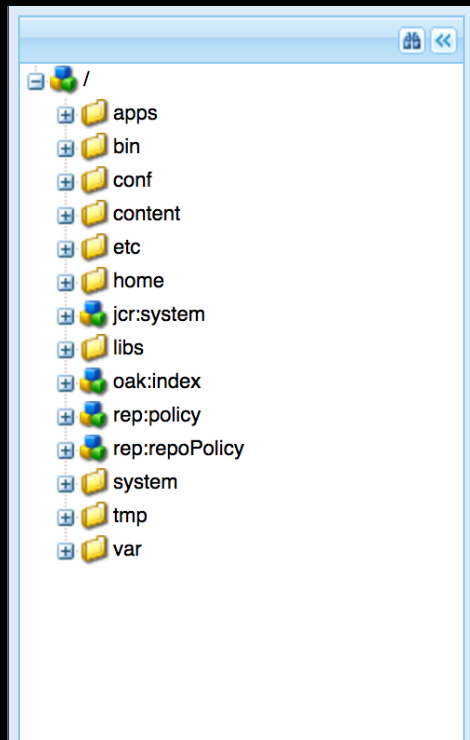
Content structure analysis in AEM

Why?

- Provide clear guidelines for the future
- Prevent clashes with custom content
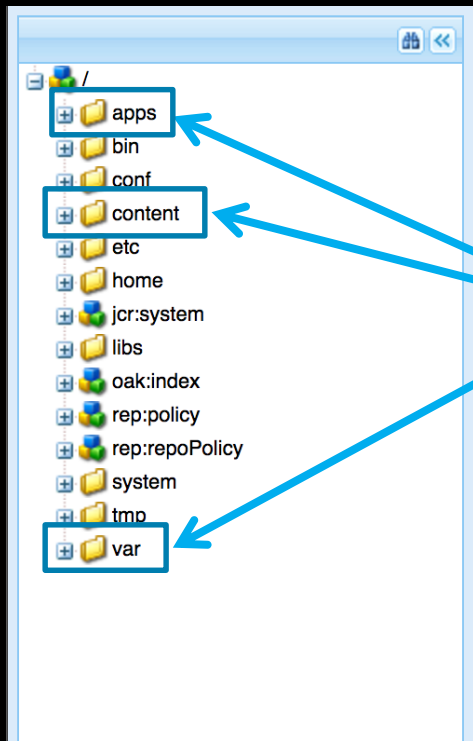- Avoid and solve the /etc/… mess

Content analysis

- Inventoried all AEM content models
- Extracted sets of feature requirements
- Mapped storage location to feature set(s)

Content inventory

What do we have here?

Content Categories

- Who produces this content?

    - Producer category

- Who consumes this content?

    - Consumer category

- Is this a kind of configuration?

    - Configuration category

- **Is this disposable at runtime?**
  - Disposable category
- **Are updates at runtime required?**
  - Updateable at runtime category
- **Should this be shared in a cluster?**
  - Cluster shared category

- AEM Developer

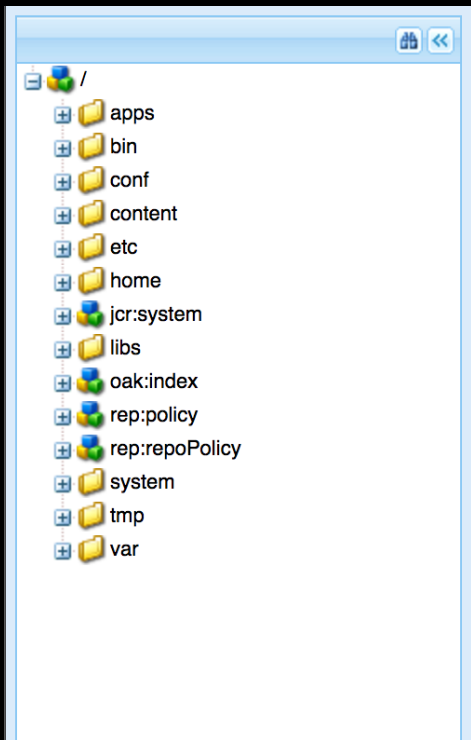- Integrator

- DevOps

- User

- System

- User

- System

Content maintenance and management

- Which ones do we use?

Content architecture – Root locations
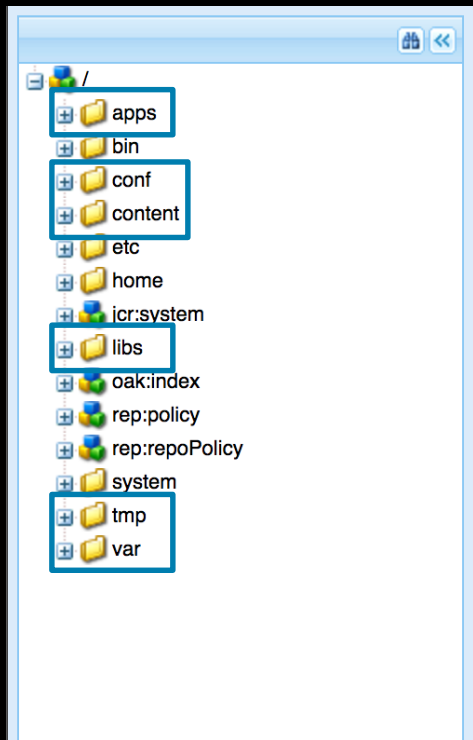
- /libs

- /apps

- /conf

- /content

- /var

- /tmp

Content architecture - Mapping

Producer	Consumer	Config	Disposable	Clusterable	Writable	Destination
AEM Dev	User	No	No	Yes	*	/content
AEM Dev	User	*	No	No	No	/libs
AEM Dev	System	*	No	*	No	/libs

- Easier future maintenance
- Overlay instead of in-place edits
- Avoids overlapping filters
- No generic storage locations

Use Case:Context-Aware configurations in Sling

Sling CA Configuration – Use case

- **Assume an OOTB configuration node in the repository**

```
/etc
    /media
        /player-config
            @speed="1.0"
```

- **We want to update the config model**

```
/etc
    /media
        /player-config
            @speed="1.0"
            @quality="HD"
```

Sling CA Configuration – Use case

- **OOTB configuration**

  ```
  /libs/sling:settings
      /media/player-config
  ```

- **Integrator**

  ```
  /apps/sling:settings
      /media/player-config
  ```

- **Runtime**

  ```
  /conf/*/sling:settings
      /media/player-config
  ```

- **Reading the configuration properties**

```
ConfigurationResolver configResolver = sling.getService(…);
ConfigurationBuilder configBuilder = configResolver.get(resource)
                                 .name("media/player-config");
ValueMap playerConfigProps = configBuilder.asValueMap();
```

Lookup order

- /libs

- /apps

- /conf/global

- /conf/tenant

- /conf/tenant/sub-tenant

- Property inheritance
- Child configuration list merging
- Customizable configuration lookup

Content management ownership

Location	Owner(s)	Remarks
/libs	AEM Dev	Deploy
/apps	Integrator, DevOps	Deploy
/conf	DevOps, User, System	Runtime
/content	All	All
/var	User, System	Runtime
/tmp	User, System	Runtime

References

- **Sling Context-Aware Configurations**

https://sling.apache.org/documentation/bundles/context-aware-configuration/context-aware-configuration.html

Q & A

Appendix

Mapping -1

Producer	Consumer	Config	Disposable	Clusterable	Writable	Destination
AEM Dev	User	No	No	Yes	*	/content
AEM Dev	User	*	No	No	No	/libs
AEM Dev	System	*	No	*	No	/libs
Integrator	User	No	No	Yes	*	/content
Integrator	User	*	No	No	No	/apps
Integrator	System	*	No	No	No	/apps

Mapping - 2

Producer	Consumer	Config	Disposable	Clusterable	Writable	Destination
DevOps	System	Yes	No	Yes	No	/conf
DevOps	System	*	No	No	No	/apps
User	User	Yes	No	Yes	*	/conf
User	User	No	No	Yes	*	/content
User	System	No	Yes	Yes	*	/tmp
User	System	No	No	Yes	*	/var

Mapping - 3

Producer	Consumer	Config	Disposable	Clusterable	Writable	Destination
System	User	No	Yes	Yes	*	/var
System	User	No	No	Yes	*	/content
System	System	Yes	No	Yes	*	/conf
System	System	No	Yes	Yes	*	/tmp
System	System	No	No	Yes	*	/var