



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
2 - 4 SEPTEMBER 2019

# AEM Template Editor in practice

Stefan Seifert, pro!vision GmbH

# About the Speaker

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



**PRO!VISION**  
SOFTWARE CRAFTSMANSHIP

<https://www.pro-vision.de>



Stefan Seifert

# Quick Stats

- 👤 Who is using **editable templates**
- 👤 Who prefers classic **“static” templates**
- 👤 Who is using the **AEM Responsive Grid** together with the **AEM Layout mode**

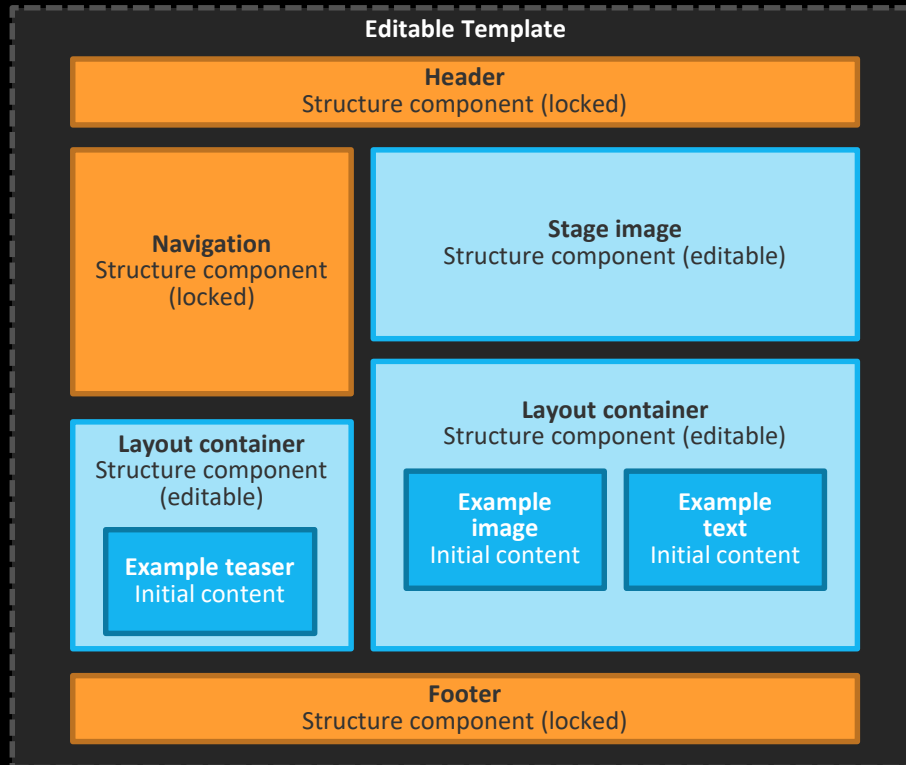


# Introduction / Recap

# What are Editable Templates

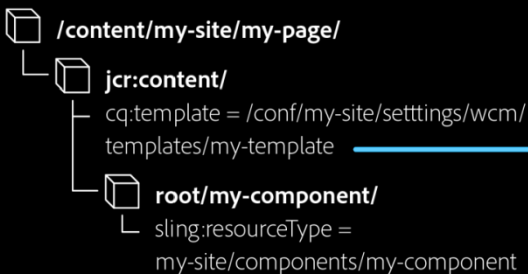
- Replaces static templates
- Structure, allowed components and policies are stored as **configuration, not hard-coded**
  - Stored in /conf instead of /apps
- Separates roles
  - Developer, Page Author, **Template Author**

# Editable Template Example

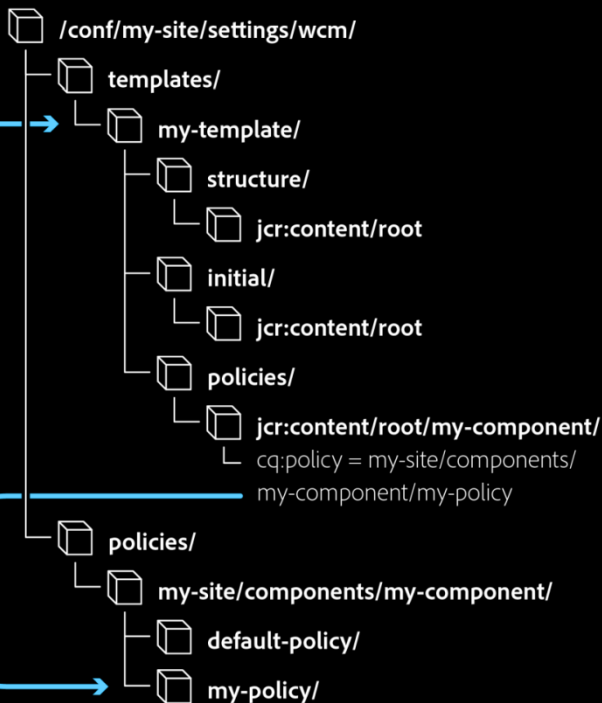


# Editable Template Content Structure

## CONTROLLER CONTENT STRUCTURE



## CONFIGURATION TEMPLATES & CONTENT POLICIES

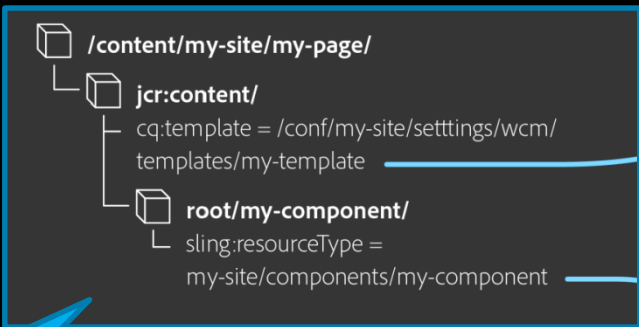


## VIEW COMPONENTS

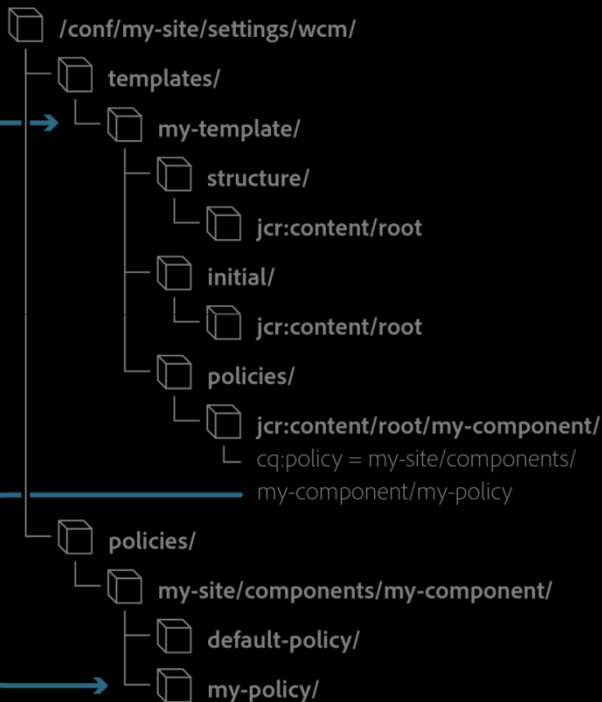


# Editable Template Content Structure

## CONTROLLER CONTENT STRUCTURE



## CONFIGURATION TEMPLATES & CONTENT POLICIES



Content Page

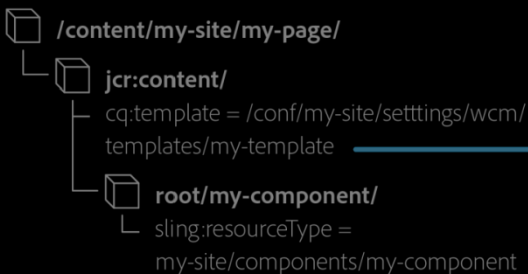
## COMPONENTS



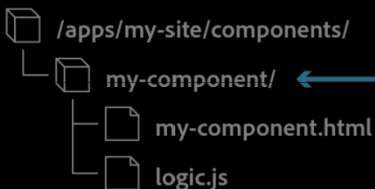


# Editable Template Content Structure

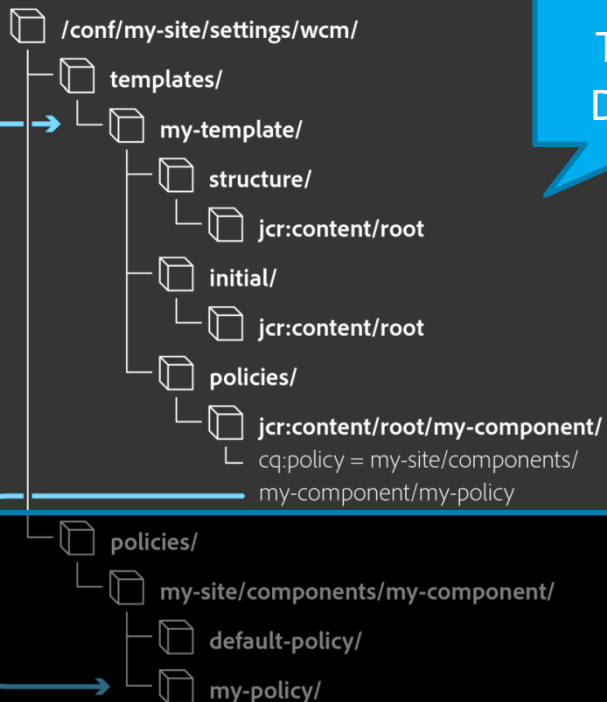
## CONTROLLER CONTENT STRUCTURE



## VIEW COMPONENTS



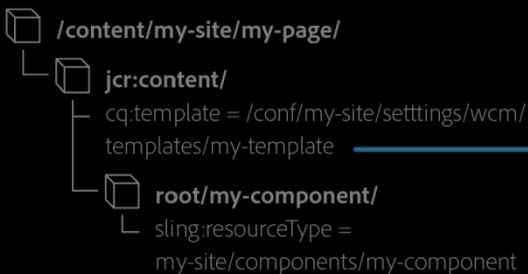
## CONFIGURATION TEMPLATES & CONTENT POLICIES



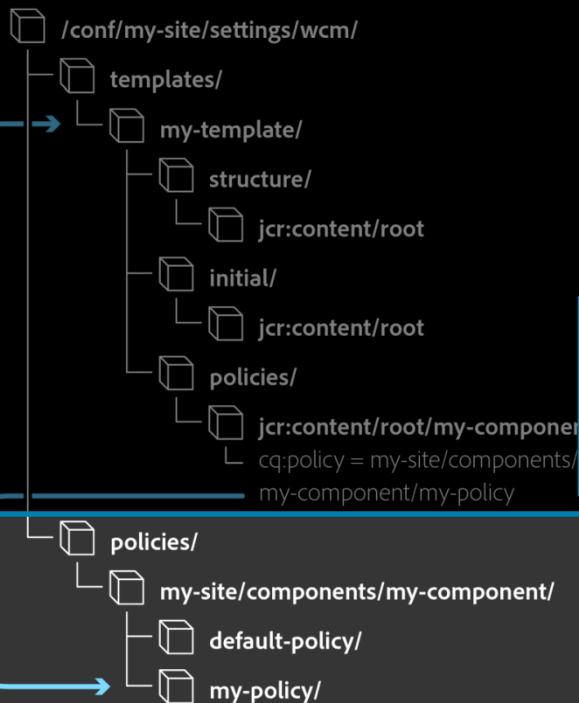
Editable  
Template  
Definition

# Editable Template Content Structure

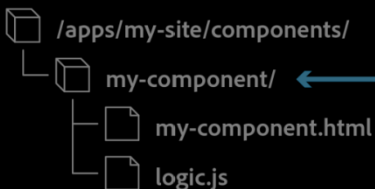
## CONTROLLER CONTENT STRUCTURE



## CONFIGURATION TEMPLATES & CONTENT POLICIES



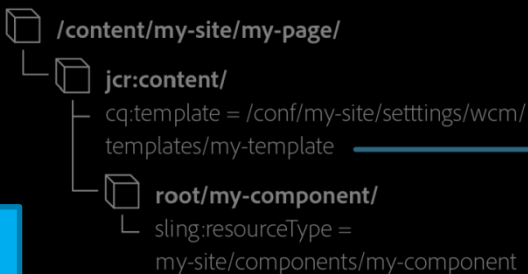
## VIEW COMPONENTS



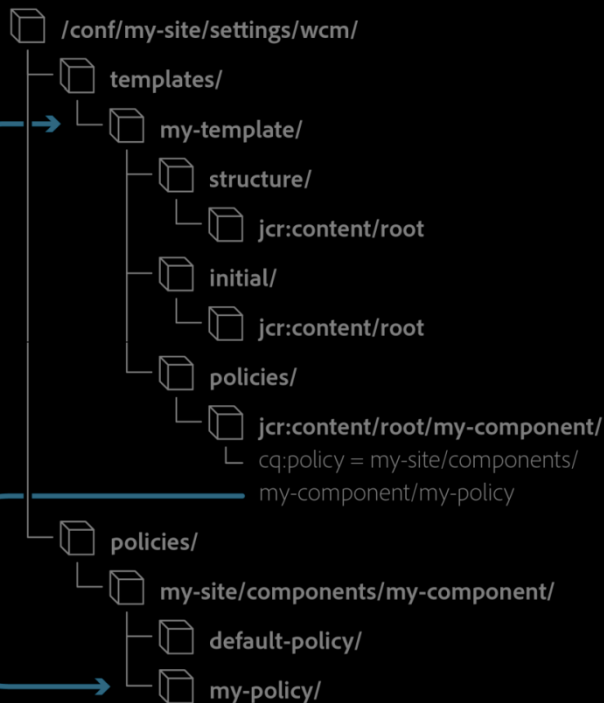
Shared set of policies

# Editable Template Content Structure

## CONTROLLER CONTENT STRUCTURE



## CONFIGURATION TEMPLATES & CONTENT POLICIES



**AEM  
Component**

## VIEW COMPONENTS



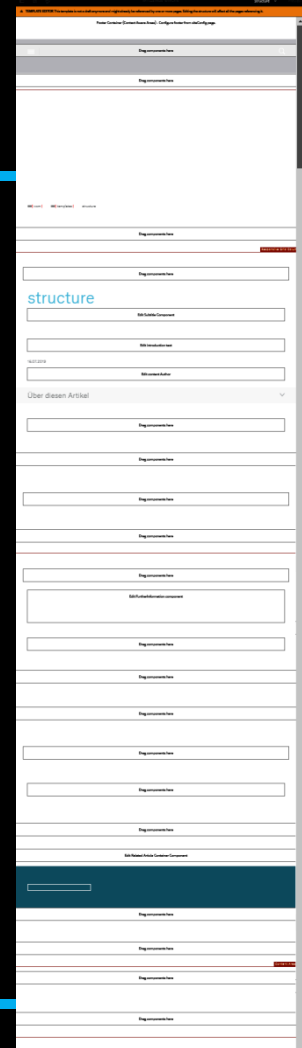
# Challenges & Solutions

(or Workarounds)

# #1 Complexity / Manageability

## Challenge:

- Template Definition can get very complex / deeply nested
- Manageability suffers



# #1 Complexity / Manageability

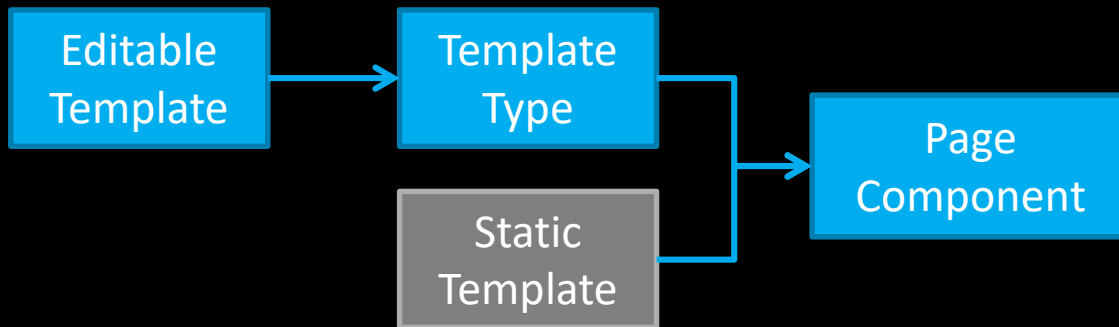
## Solution:

- Move static parts common to all template to the page component
  - e.g. header, footer, navigation
- You can mix static and editable parts of a template to any degree

# #1 Complexity / Manageability

## Solution:

- Keep in mind: Every editable template is based on a page component – similar to static templates

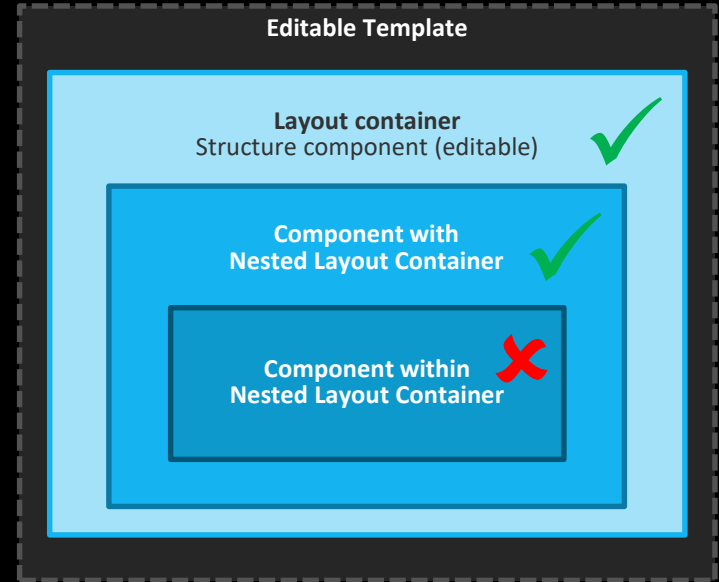


DEMO

## #2 Policies for nested Parsys

### Challenge:

- Policies for components in nested parsys not supported
- E.g. unable to define allowed components for nested parsys





## #2 Policies for nested Parsys

### Solution:

- Define policy for nested component on a higher-level parsys
- Without adding it to the allowed components
- Use add/remove trick to define it in the UI
- (Not possible to use different policies)
- Alternative: Use <https://wcm.io/wcm/parsys/> (without policies)

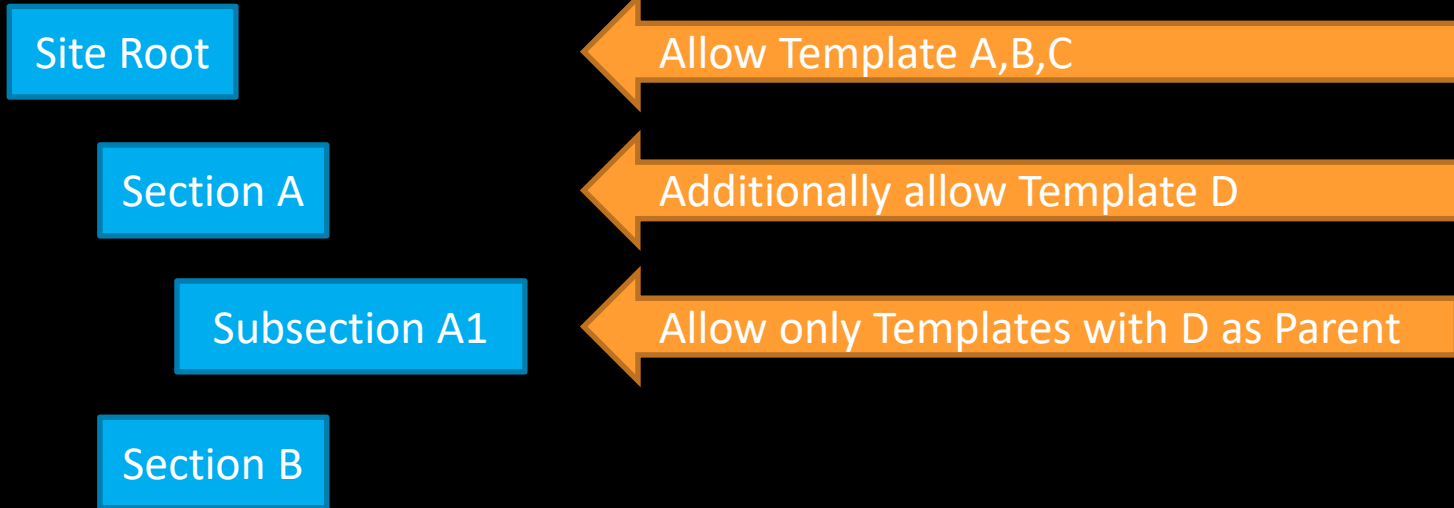
## #3 Template Allowed Paths

### Challenge:

- Allowed templates are defined **in content** via `cq:allowedTemplates` property
- Cumbersome if you have lots of sites
- Fine-grained allow rules for paths difficult

# #3 Template Allowed Paths

## Challenge:



## #3 Template Allowed Paths

### Solution:

- Use properties `allowedPaths`, `allowedParents`, `allowedChildren` (known from static templates)
- Unsupported in the UI
- Set them directly in the repository

## #4 Semantic Naming

### Challenge:

- Adding components in structure view leads to node names like  
`responsivegrid_676359182`
- Policies are named like  
`policy_13967744492500`

## #4 Semantic Naming

### **Solution:**

- **Rename auto-created nodes manually before using the template in production**
  - Created content has clear node structure
  - Benefit when accessing content from code
  - Benefit when merging policies

# #5 Policy Assignment Redundancy

## Challenge:

- No support for reusing parts of templates in other templates
- Policies are shared, but policy assignments have to be done anew for each template
  - Problematic if you have lots of sites, and some site/tenant-specific settings in the policies

# #5 Policy Assignment Redundancy

## Solution:

- No OOTB Solution available
- For nested parsys with a fixed list of child components use <https://wcm.io/wcm/parsys/>



# #6 Template Author Role Conflicts

## Challenge:



Dev Team

Initial versions &  
major updates



Templates  
& Policies

Variations &  
slight modifications



Power User

# #6 Template Author Role Conflicts


## Challenge:



Dev Team



Deploy with App



User

**Clash**  
Changes on PROD get overwritten

No easy merging  
No easy way back

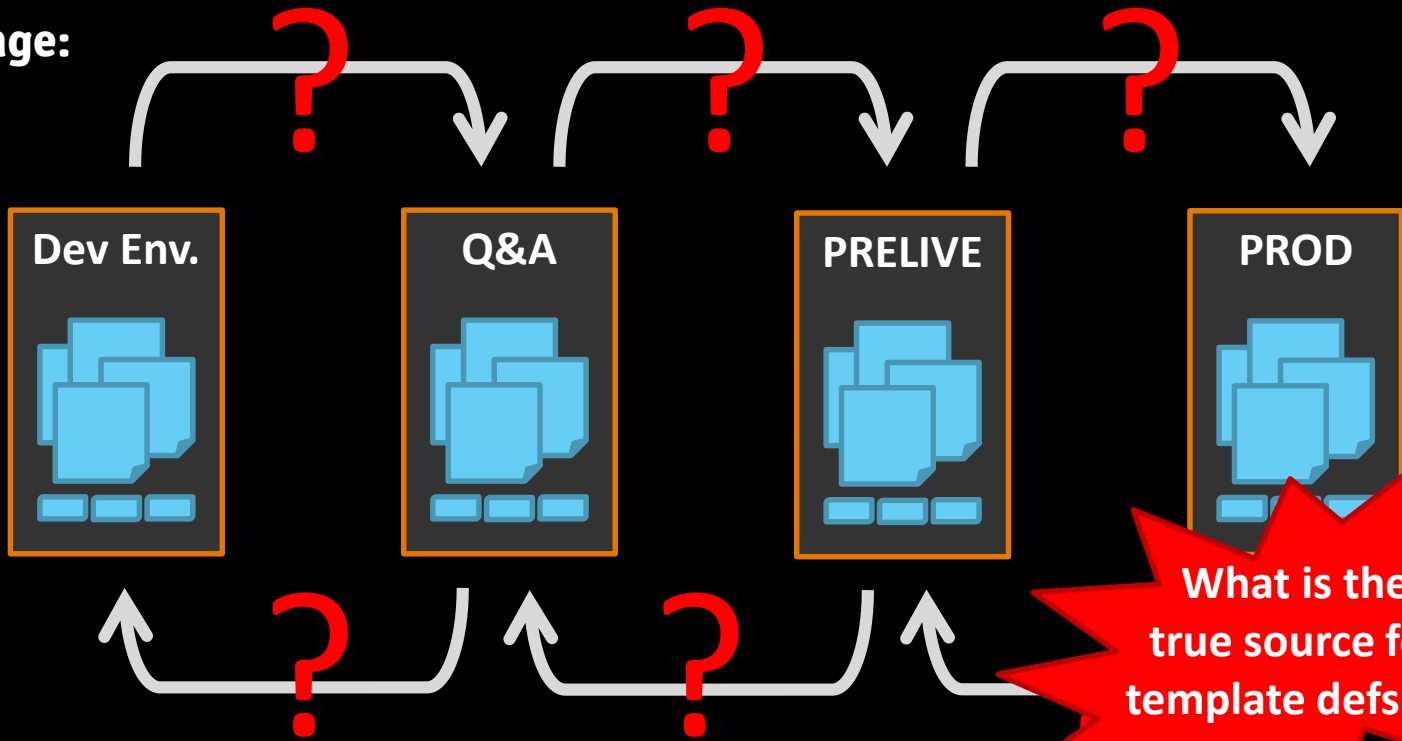
## #6 Template Author Role Conflicts

### Solution:

- Easy (but unsatisfying) solution:
  - Do not allow power users to change anything
  - Only dev team is allowed to edit templates
- Export back updated template Defs to git repo after each change on PROD

# #7 Template Definition Distribution

Challenge:



What is the true source for template defs?

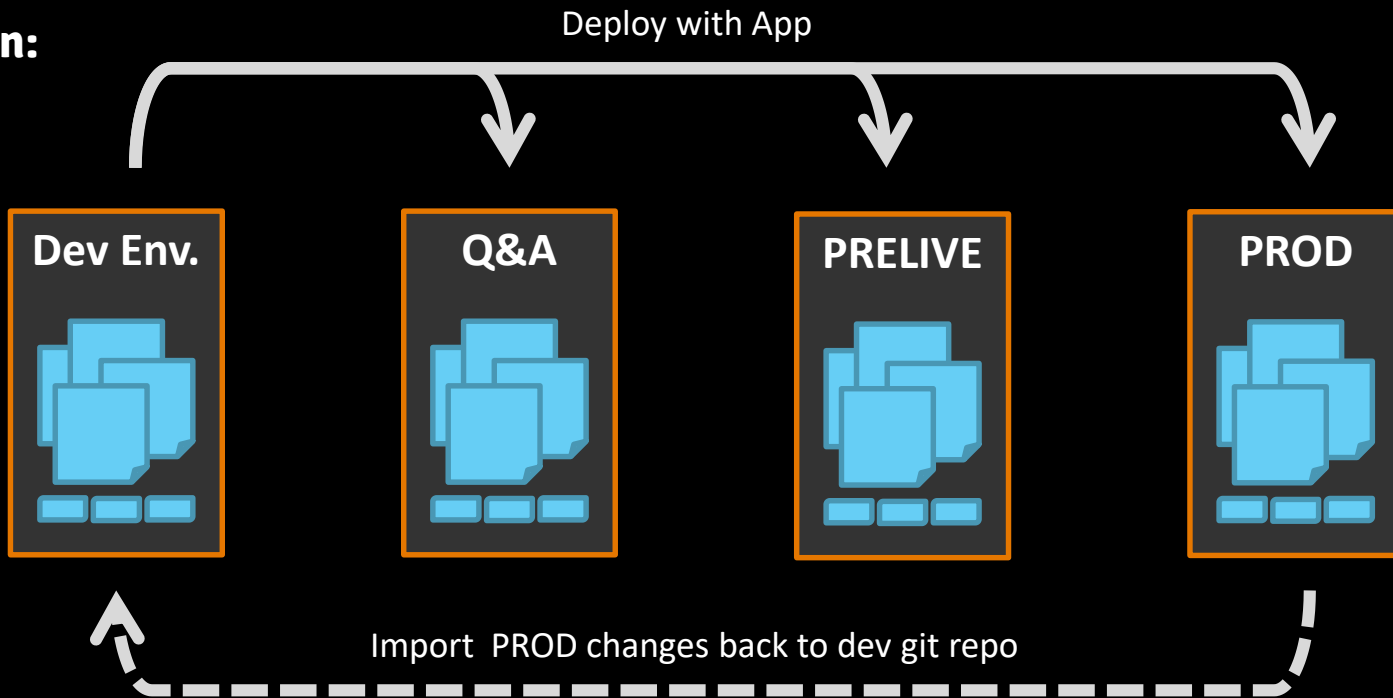
# #7 Template Definition Distribution

## Solution:

- Easy solution (again):  
Templates only by dev team, not power users
- True source should be the **git repo**
- Implement a “backchannel” from PROD

# #7 Template Definition Distribution

**Solution:**



# Summary

# Summary

- **Template Editor and Policies are powerful**
- **Avoid the pitfalls**
- **Use deployment strategy with care**



- **Demo Project**

<https://github.com/adaptto/2019-aem-template-editor>

- **GEM Session for Template Editor**

<https://helpx.adobe.com/experience-manager/kt/eseminars/gems/aem-managing-content-with-template-editor.html>

- **AEM Documentation**

<https://helpx.adobe.com/experience-manager/6-5/sites/authoring/using/templates.html>

Bootstrap Template Used for the Demo: <https://github.com/BlackrockDigital/startbootstrap-agency>