



Adobe



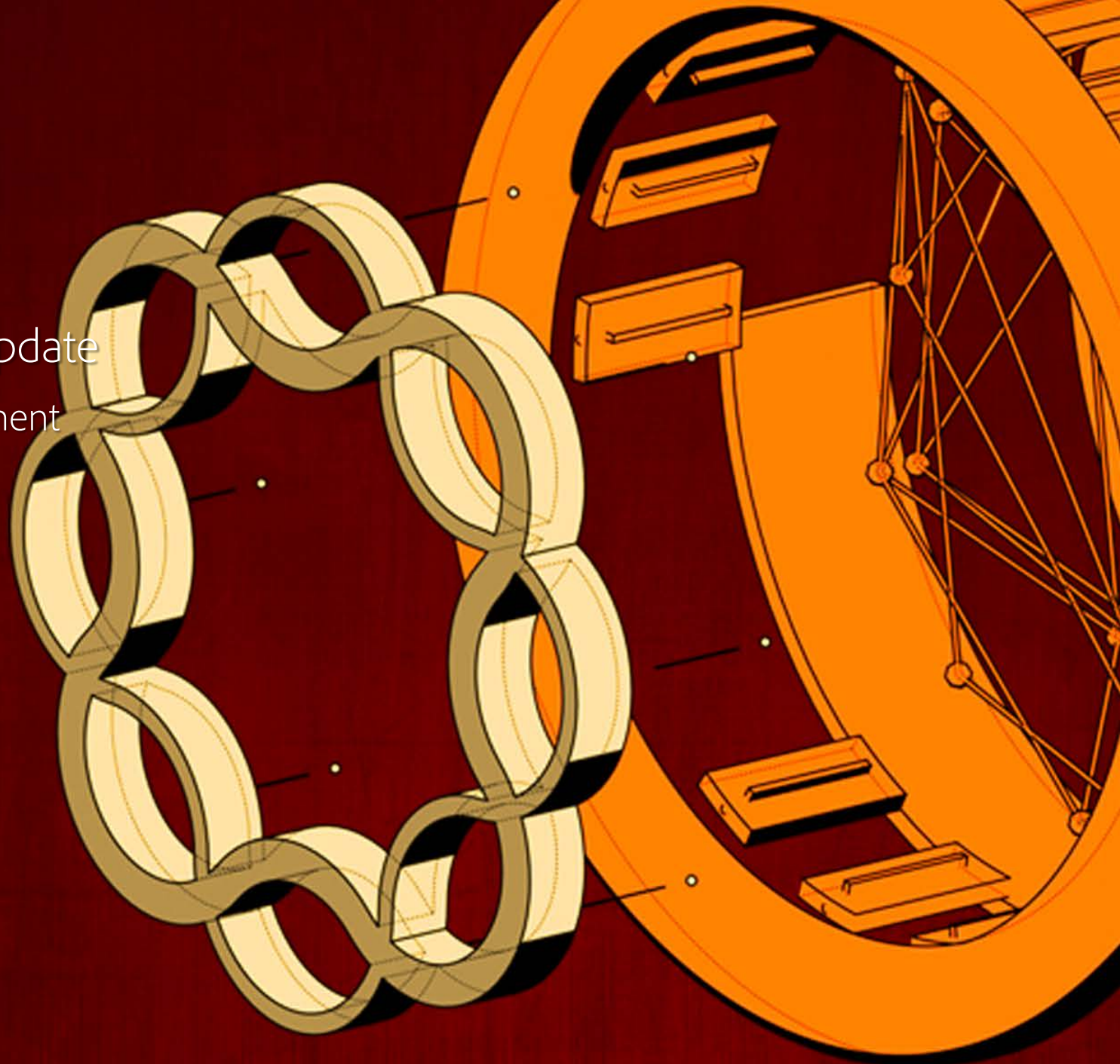
ADOBE EXPERIENCE MANAGER

# AEM GEMS

March 2018 Developer Community Update

Cedric Huesler | Director of Product Management

Gabriel Walt | Product Management



# Agenda

1. AEM 6.4 Overview
2. Building Sites
3. Style System
4. Repository Cleanup
5. Classic UI Status
6. Supported Environments
7. Content Services
8. Single Page Applications

# AEM 6.4 Overview

# AEM in Numbers and Dates

- ~55% of hostnames running AEM are on 6.3 (~30%) and 6.2 (~25%)
- We estimate around ~32'000 servers running AEM
- We estimate around ~120'000 domain names running on AEM servers
  
- AEM 6.4 was announced on February 8th
- AEM 6.4 will be generally available in April 2018

# What's New in AEM 6.4

## Sites

- Style System for Components
- Experience Fragments incl. integration with Adobe Target
- Custom Content Fragments models
- Single Page App Framework powered sites
- Extended publish for AEM Screens
- Enhanced Translation integrations

## Assets

- CC In App Panels
- Enhanced Smart Tags
- Smart Translation Search
- Smart Crop & Smart Imaging
- Metadata Enhancements (Cascade/Contextual, Bulk Import & Export)
- Usability Improvements (Browse, Search)
- Reporting Enhancements
- Brand Portal Improvements

## Forms

- Multi-channel authoring of customer communications
- Forms Conversion at scale using AI
- Enhanced Data integrations
- Simplified authoring, Enhanced workflow and configurations

## Managed Services

- AWS & Azure support
- Continuous Integration / Continuous Delivery
  - Infrastructure & tools
  - CI/CD pipeline setup
  - Code inspection
  - Functional testing
  - Performance/Load testing
  - Security testing
  - Graphical User Interface

## Foundation

Minor touch upgrade (6.3), Online Tar Revision Clean-up performance, Indexing resilience and lower impact, Maintenance & Monitoring Improvements , Repository restructuring



# Building Sites – Getting Started

- WKND Tutorial  
<https://helpx.adobe.com/experience-manager/kt/sites/using/getting-started-wknd-tutorial-develop.html>
- We.Retail 3.0  
<https://github.com/Adobe-Marketing-Cloud/aem-sample-we-retail>
- Project Templates
  - Archetype  
<https://helpx.adobe.com/experience-manager/using/first-arch10.html>  
<https://gist.github.com/gabrielwalt/2c645dbcf974f25f9add192f44bb387>
  - Lazybones  
[https://helpx.adobe.com/experience-manager/using/aem\\_lazybones.html](https://helpx.adobe.com/experience-manager/using/aem_lazybones.html)

# Core Components

- Production-ready components
  - **Clean Markup:** HTL templates, Semantic HTML, BEM class name notation, Data attributes for JS binding.
  - **Configurable:** The component capabilities that the author can use are defined in the template editor.
  - **Extensible:** The HTL template and SlingModels implementation can be extended or customized.
  - **Consumable:** The SlingModels can be consumed as pure JSON form to be rendered elsewhere.
  - **Versioned:** The components can evolve without breaking existing implementations.

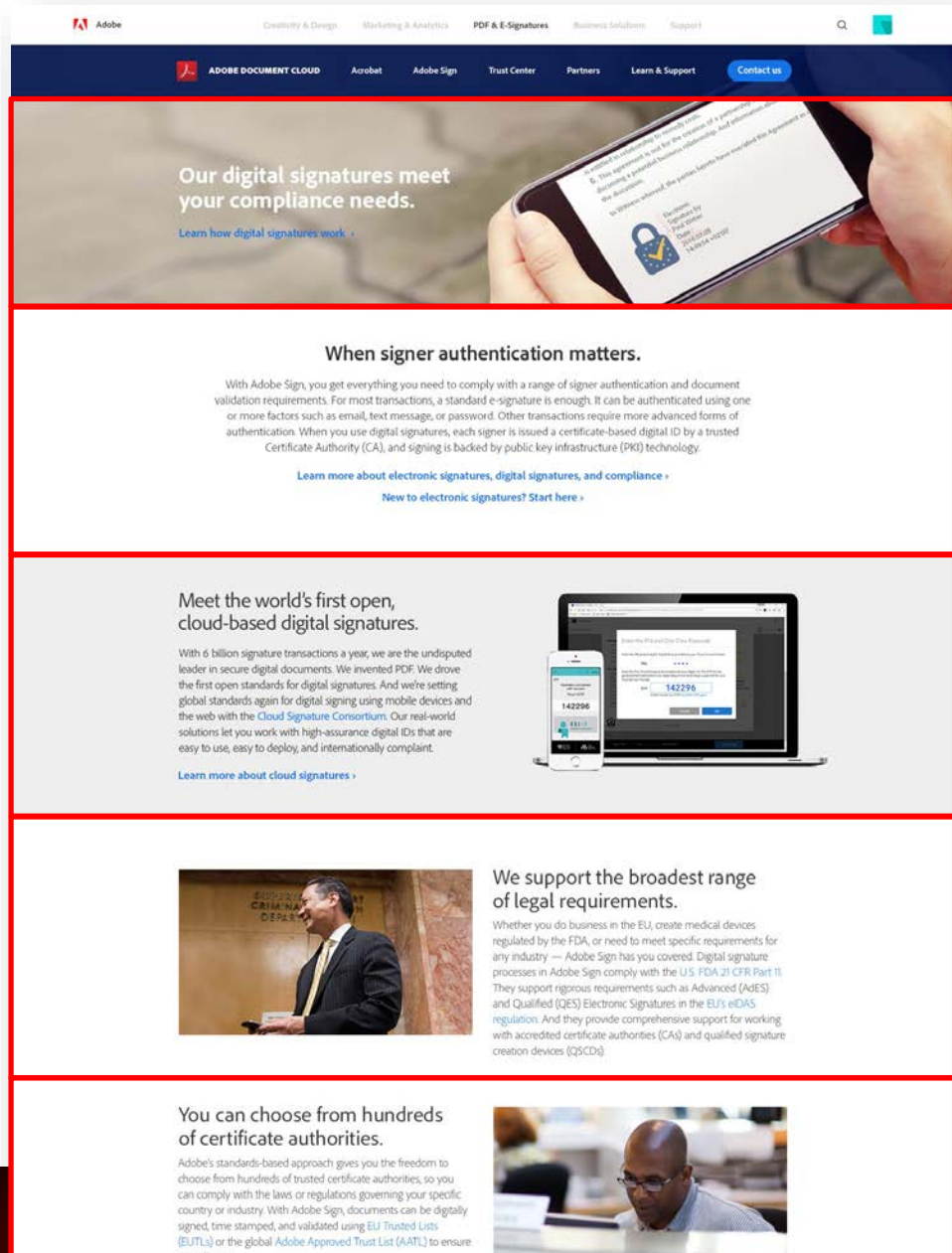


# Core Components

1. **Page:** Responsive page that works well with the Template Editor.
2. **Title:** Headings configurable to allow levels 1 to 6.
3. **Text:** Plain or rich text with configurable capabilities.
4. **Image:** Smart image display with configurable capabilities.
5. **List:** Lists pages that match the configured criteria.
6. **Breadcrumb:** Lists the hierarchy of parent pages.
7. **Form Container:** Form paragraph system.
8. **Form Button:** Submit or scriptable button.
9. **Form Text:** Text input field (text, text area, email, phone, date, number).
10. **Form Options:** Multi-options input field (checkboxes, radios, drop-down, multi-select).
11. **Form Hidden:** Invisible input field, used for sending information along a form.
12. **Navigation:** Site navigation that handles language structures, live copies and redirections.
13. **Language Navigation:** Displays the language structure of a site.
14. **Quick Search:** Incremental search field.
15. **Sharing:** Facebook and Pinterest widgets.



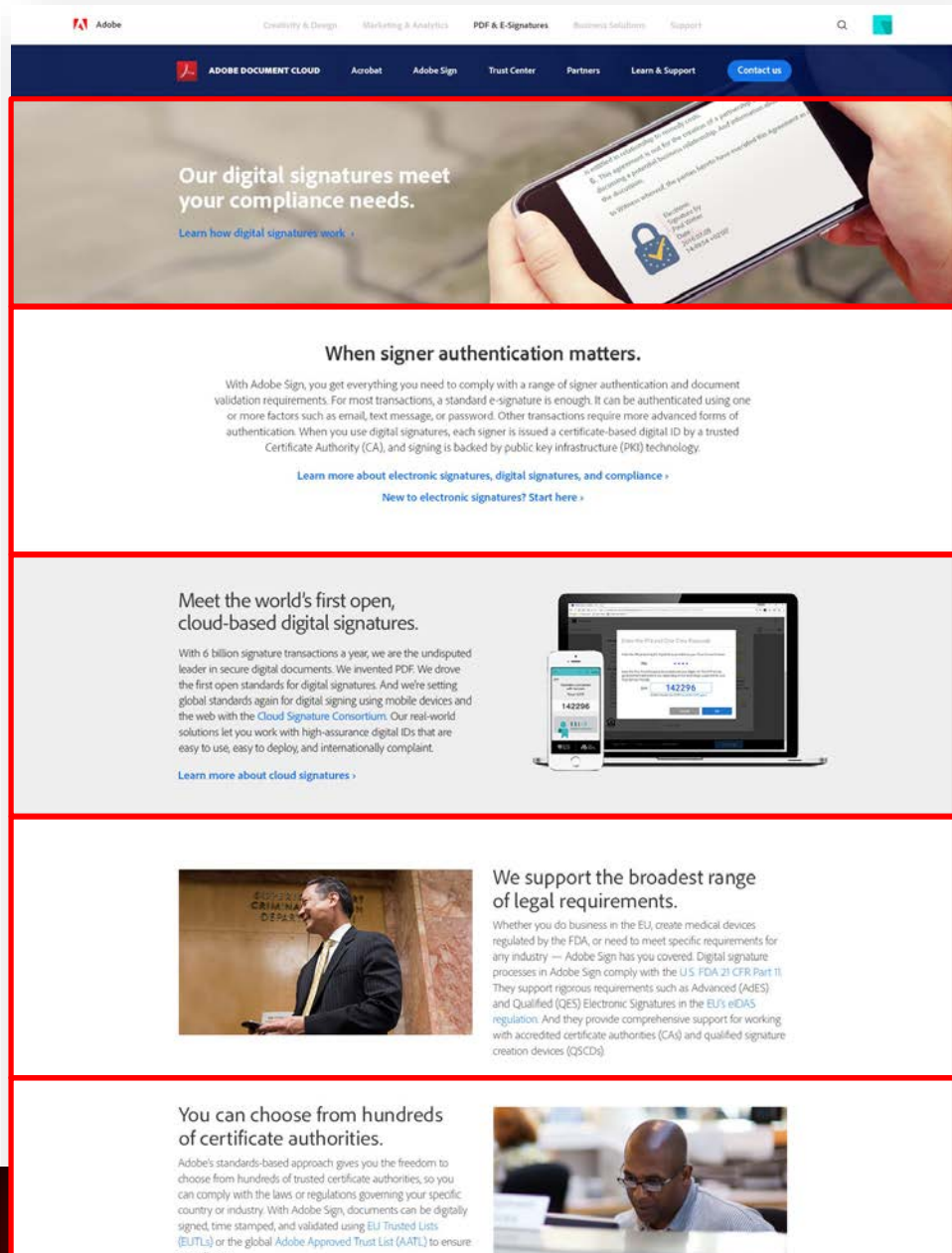
# Style System – An Example



Components on the page:

- Hero banner
- Rich text
- Highlighted banner
- Image and text
- Text and image

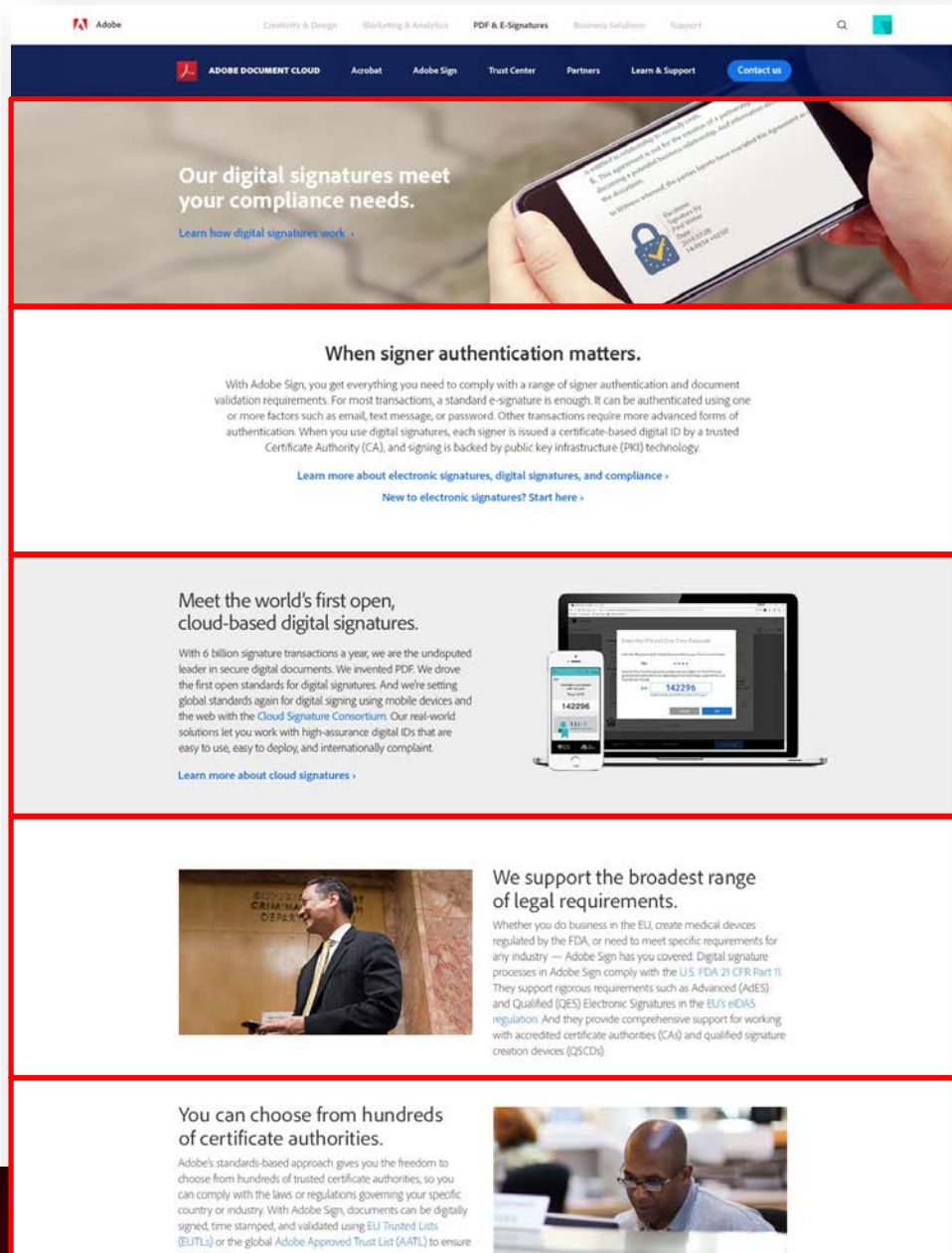
# Style System – An Example



Components on the page:

- Title, ~~text~~, link(s), image (background)
- Title, text, link(s), ~~image~~ (centered)
- Title, text, link(s), image (highlighted)
- Title, text, ~~link(s)~~, image (image left)
- Title, text, ~~link(s)~~, image

# Style System – An Example



Single component instead:

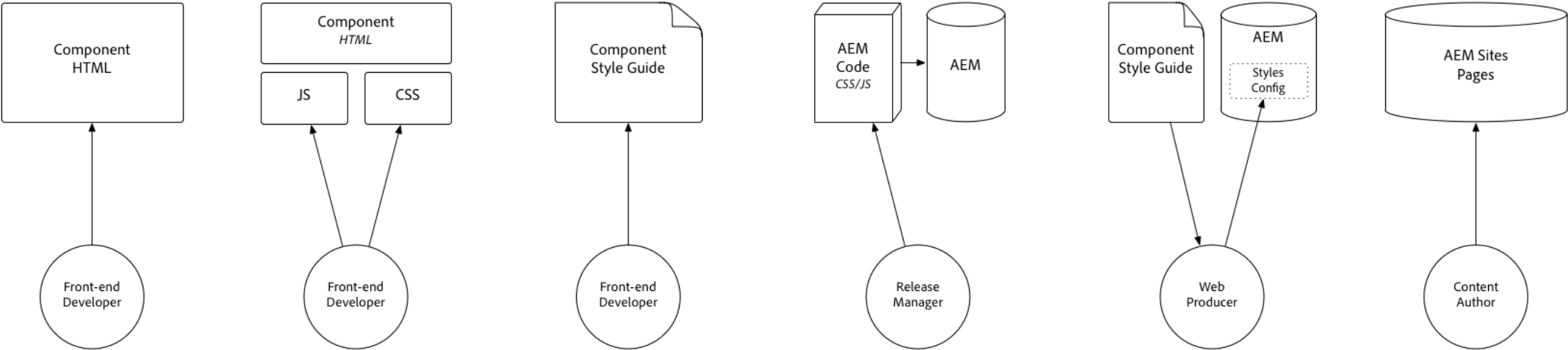
- Title
- Rich text – optional
- List of link(s) – optional
- Image – optional

With component styles:

- Image as background
- Center text
- Highlighted
- Image on the left

# Front-end Designer & Developer Workflow

Starting 2018 with AEM 6.3 SP2 and 6.4 with introduction of the Style System:



The Front end developer reviews and understands the HTML generated by the Component.

The HTML definition of the Component is an API. The CSS and JS will be "consumers" of the this HTML.

The Front end developer builds CSS and JavaScript that targets the existing HTML of the Component.

The FED can introduce variations by assuming the component's wrapping tag can:

- 1) Change the component's wrapping HTML element (defaults to div)
- 2) Set any new CSS Classes to the component's wrapping HTML element

The Component Style Guide is updated with the new Component elements, CSS selectors and any valid combinations therein.

The new CSS and JS are deployed to AEM.

The Web Producer uses the Component Style guide to update Component Policies with the new Style -> Selector mappings.

The Content Author can apply the Styles defined by the Web Producer on individual components on Web Pages.

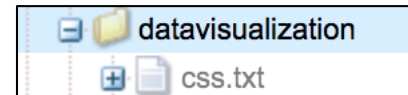
# Repository Cleanup

# Repository Classification

**Public** (granite:PublicArea) – Defines a node as public so that it can be overlaid, inherited(sling:resourceSuperType) or used directly(sling:resourceType). Nodes beneath /libs marked as Public will be upgrade safe with the addition of a Compat Package. In general customer apps should only leverage nodes marked as Public.

Properties				Access Control	Replication	Console
	Name ▲	Type	Value			
1	jcr:created	Date	2018-02-01T00:14:40.015-08:00			
2	jcr:createdBy	String	admin			
3	jcr:mixinTypes	Name[]	granite:PublicArea			
4	jcr:primaryType	Name	nt:folder			

**Final** (granite:FinalArea) – Defines a node as final. Nodes classified as final cannot be overlaid or inherited. Final nodes can be used directly via sling:resourceType. Subnodes under final node are considered internal by default.

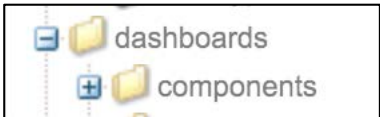


Properties				Access Control	Replication	Console
	Name ▲	Type	Value			
2	categories	String[]	granite.datavisualization			
3	dependencies	String[]	underscore, jquery			
4	embed	String[]	granite.d3.embedded, gra			
5	jcr:created	Date	2018-02-01T00:14:12.123			
6	jcr:createdBy	String	admin			
7	jcr:mixinTypes	Name[]	granite:FinalArea			



# Repository Classification

**Internal** (granite:InternalArea) – Defines a node as internal. Nodes classified as internal cannot be overlaid, inherited, or used directly. These nodes are meant only for internal functionality of AEM.

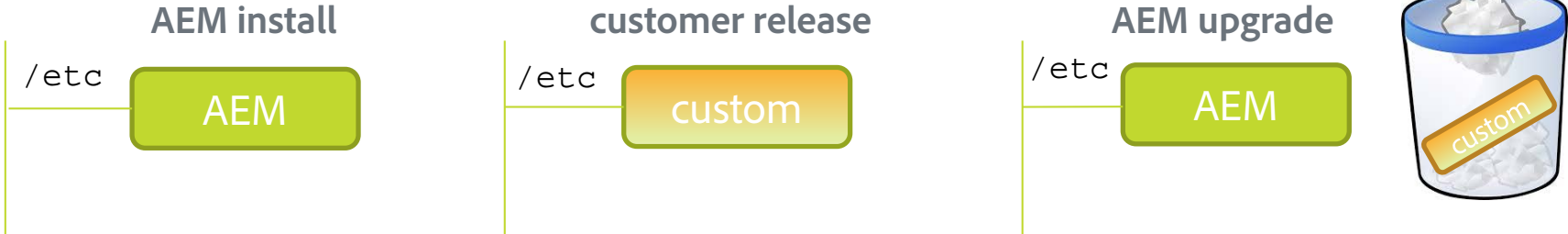


Properties					
Access Control		Replication		Console	
	Name ▲	Type	Value		
1	jcr:created	Date	2018-02-01T00:14:51.546-08:00		
2	jcr:createdBy	String	admin		
3	jcr:mixinTypes	Name[]	granite:InternalArea		
4	jcr:primaryType	Name	slings:Folder		

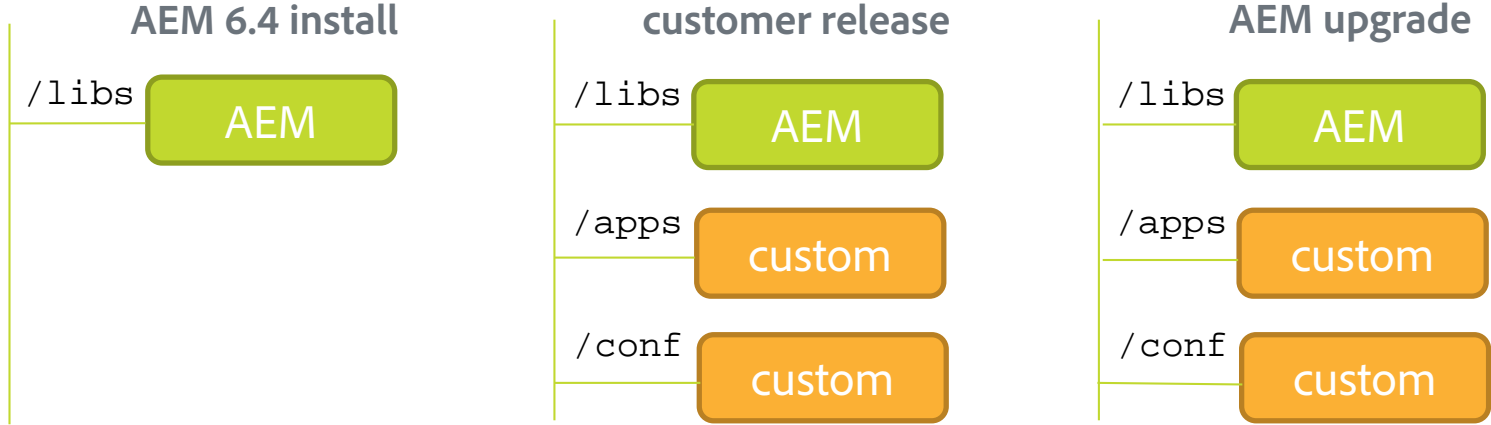
**No Annotation** – Nodes inherit classification based on the tree hierarchy. The /root is by default public. Nodes with a parent classified as Internal or Final are also to be treated as Internal.

# Repository restructuring for easier upgrades

- 6.3 and prior, ambiguity over ownership of /etc/\* caused this:



- 6.4 moves content out of /etc, with guidelines on who owns what and how to extend that will help in future releases

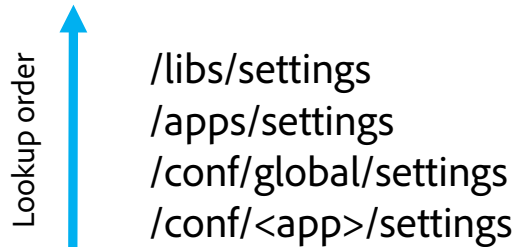


# Repository restructuring for easier upgrades

Extensibility patterns – public documentation will map /etc content to its appropriate pattern:

## 1) Context-aware configuration

- A multi-layered configuration approach
- For content moved from /etc to /libs/settings
- Example: /etc/workflow/notifications → /libs/settings/workflow/notifications
- Layering follows these rules:



- Cloud services is one example of a feature which has benefitted from this change

## 2) Overlay in /apps

- The more traditional layered approach
- For content moved from /etc to /libs, but **not** under settings
- Check documentation on a per-feature basis to ensure content is extensible in this way
- Example: /etc/clientlibs/fd/fmaddon → /libs/fd/fmaddon

## 3) Clientlibs

- Extensibility strategy that appends multiple JS files
- Content with the same “categories” property will be appended
- Example: /etc/clientlibs/ckeditor → /libs/clientlibs/ckeditor

# Classic UI Status

# Classic UI Deprecation and Support (as of March 2018)

- Classic UI is default part of AEM in version 6.1, 6.2, 6.3 and 6.4
- Classic UI is supported (as in you can call support or raise a support ticket) for versions 6.1\*, 6.2, 6.3 and 6.4.
- Classic UI is deprecated starting 6.4, that means, we are planning to ship the 2019 version of AEM without the Classic UI per default. The Classic UI will be installable via package.
- Recommendation for customers is to upgrade to newest available AEM version 1st, and then make the switch away from the Classic UI.

\* ) AEM 6.1 is end of core support on May 31<sup>st</sup> 2018.

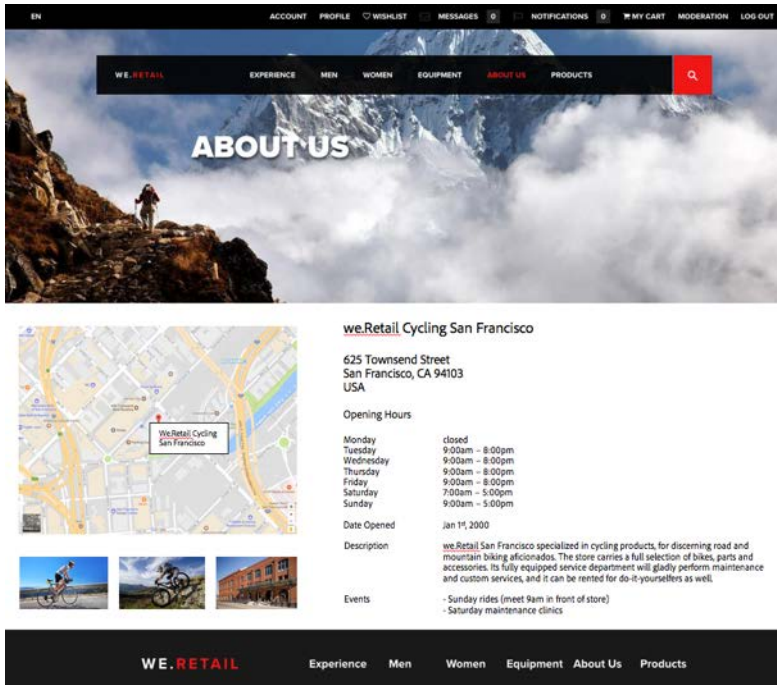
# Oracle Java version support with AEM (as of March 2018)

- **Java 8** is the recommended version for AEM 6.4, 6.3, 6.2 and 6.1
- At this point – Oracle says they provide public updates until January 2019
- **Java 9 and 10** are not supported, because they are non-LTS releases by Oracle
- **Java 11** – the next LTS release - is planned for September 2018 and Adobe will communicate support plans closer to the release date.



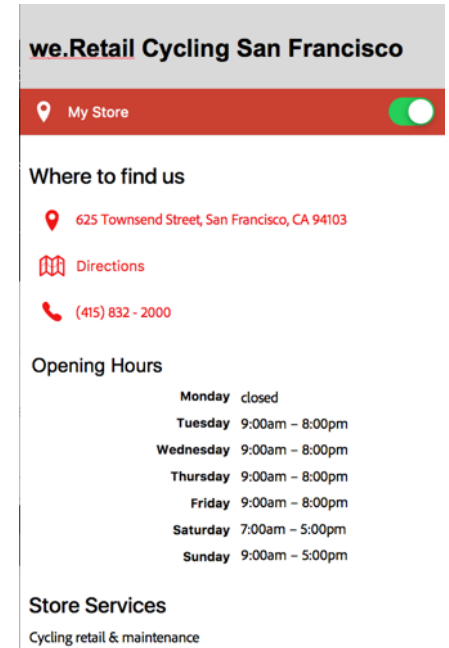
# AEM Content Fragments & Services – Use Cases

Create, manage and deliver structured content to any channel



```
JSON Raw Data Headers
Save Copy
description: "global/models/store-info"
title: "Store-01"
type: "core/wknd/boxes/components/contentfragment/v1/contentfragment"
items:
  - zipcode: "94133"
    value: "ZIP code"
    type: "string"
  - country: "United States"
    value: "Country"
    type: "string"
  - images:
    - title: "Store Images"
      type: "string"
  - address:
    - value: "625 Townsend Street"
      title: "Address"
      type: "string"
  - description-events:
    - value: "op=description/box/zipcode, Retail San Francisco specialized in cycling products, for discerning road and mountain biking aficionados. The store carries a full selection of bikes, parts and accessories. Its fully equipped service department will gladly perform maintenance and custom services, and it can be rented for do-it-yourselfers as well. op=events/box/zipcode"
      title: "description-events"
      type: "string"
  - city:
    - value: "San Francisco"
      title: "City"
      type: "string"
  - openinghours:
    - value: "op=openinghours/box/zipcode, table cellpadding="2" cellspacing="2" border="1" style="width: 100%; text-align: center;">|  |  |
| --- | --- |
| Monday | closed |
| Tuesday | 9:00am – 8:00pm |
| Wednesday | 9:00am – 8:00pm |
| Thursday | 9:00am – 8:00pm |
| Friday | 9:00am – 8:00pm |
| Saturday | 7:00am – 5:00pm |
| Sunday | 9:00am – 5:00pm |

```



For example

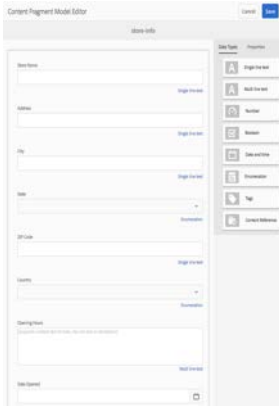
- Store locator
- User/employee /contact information
- Header/footer content





# AEM Content Fragments & Services - Personas

## 1. Information Architect



Content Model

## 2. Content Author



Content Fragment

## 3. AEM Developer



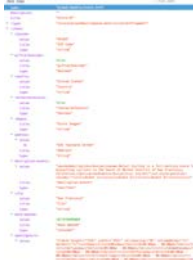
Component



{API}

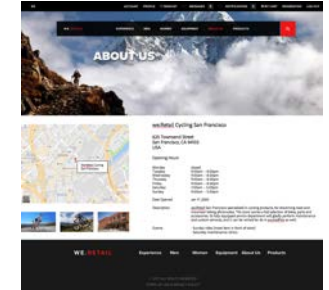


HTML



JSON

## 4. AEM Web Producer



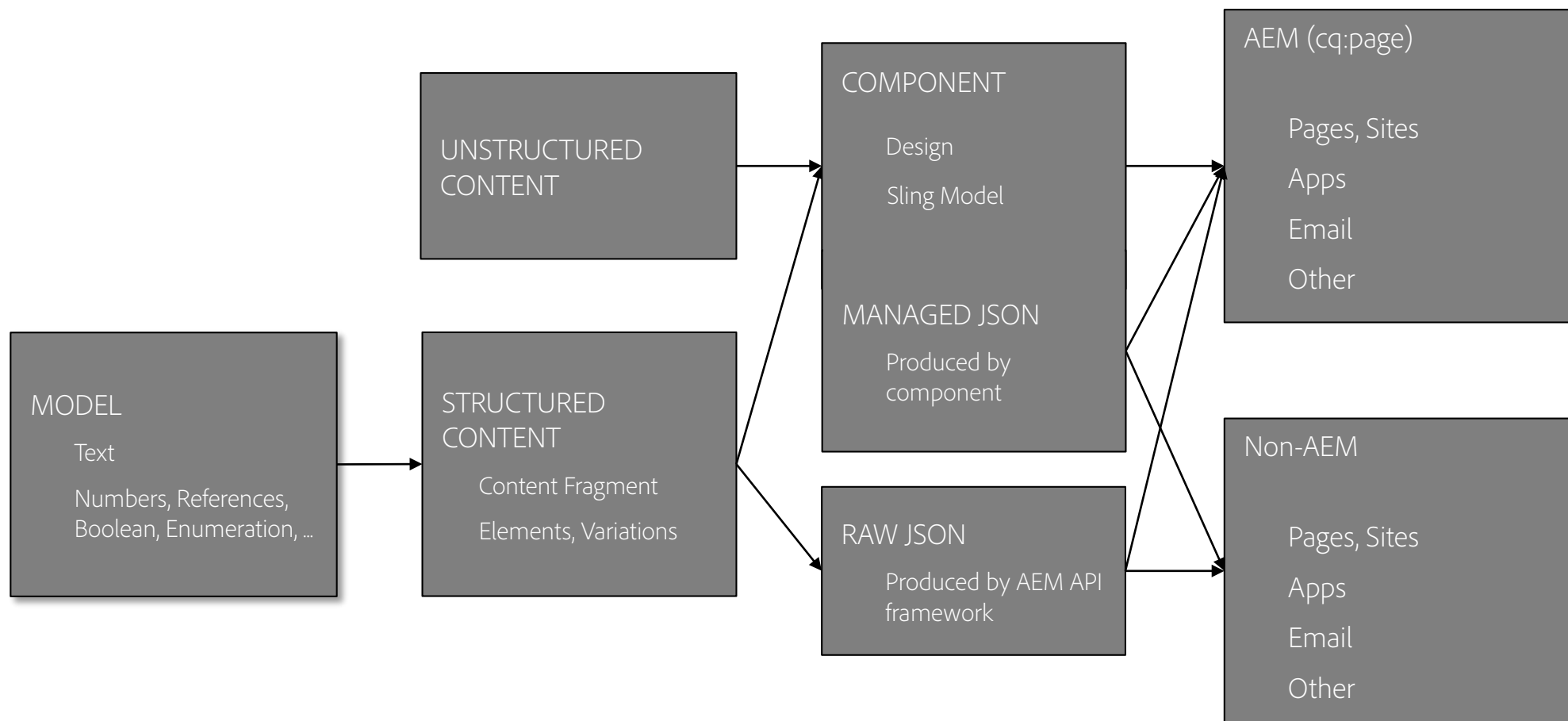
Web Page

## 5. Web Developer



Mobile App  
Single Page App  
...

# AEM Content Fragments & Services - Technology



# Single Page Applications

# Thank You!

Please help us build a product that better fits your needs by taking two minutes to give us some insight. We appreciate your perspective.

→ <http://bit.ly/aem-dev-2018>



**Adobe**