

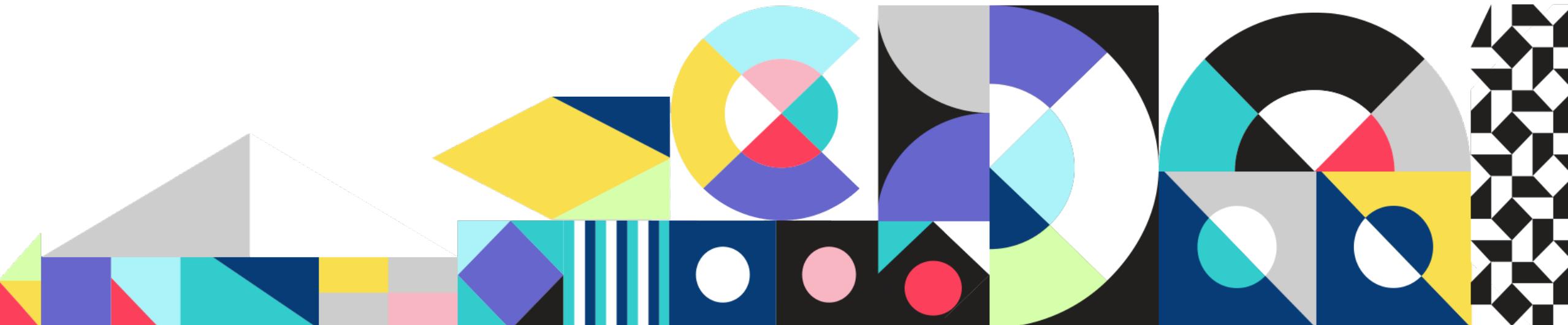
Deconstructing Wireframes into AEM Templates

Dennis Selfridge | Technical Instructor



Our goals for the next 90 minutes are...

- 1 | Get to know some tools for reuse and governance AEM provides.
- 2 | Tune our vision to see AEM patterns and components.
- 3 | Identify audiences and use Editable Templates to separate concerns.
- 4 | Devise a long term plan to empower authors!



Deconstructing Wireframes – Class Materials

- AEM 6.4 running Author instance
- L732 Workbook
- Activities Folder

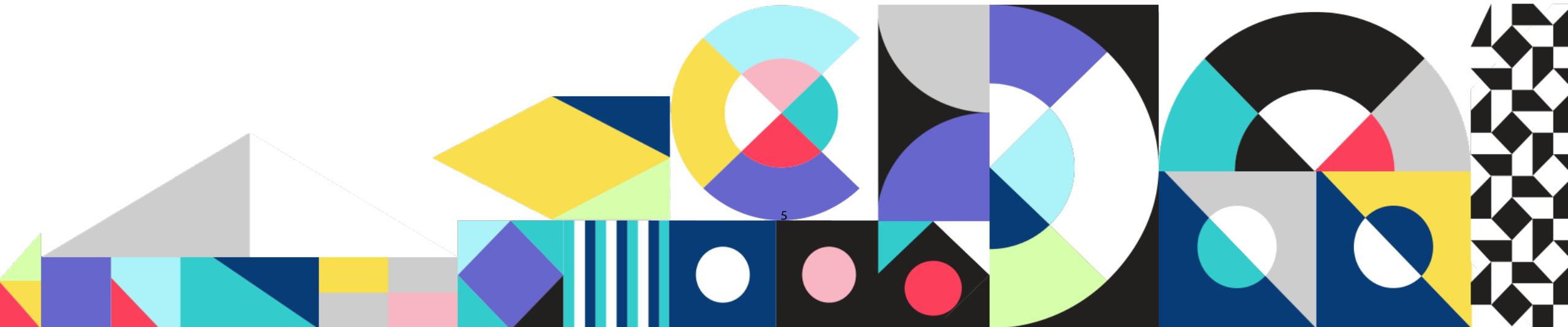
Pre Class set up:

- Start AEM
- Locate and Unzip Activities Folder (ONLY the Activities ROOT Folder!)

Summit Getaways- Static Content and Requirements

- This is a faux vacation rental site to demonstrate both the deconstructing wireframe workshop and the context hub personalization data store workshop.
- The creative delivered has two different looks for header and footer for registered users and the general public.
- Our aim is to divide the creative into logical chunks of content we can map to components.
- While we will do that in this workshop, actual creation of a Brand framework, audiences and activities will be saved for L733.
- In this Lab we will create a Template Folder, Template Type and use those to create Templates specifically configured for each of the audiences.
- Logged in Users will have separate content policies (the old Design Mode) to distinguish content for Renters and Owners.

Let's get started!



Deconstructing Wireframes

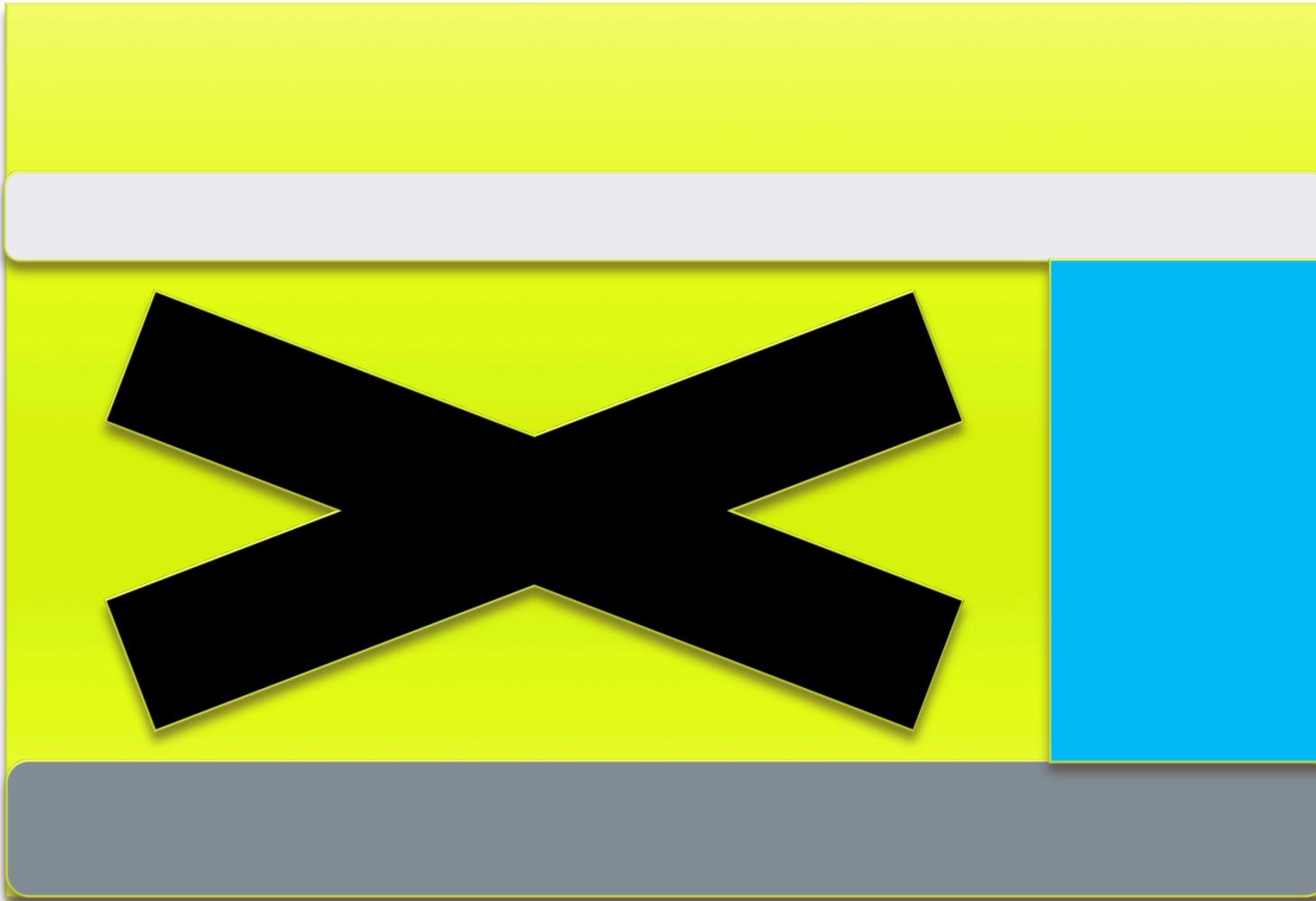
- Implementation can be done in phases to gradually transition content from IT controlled to SME controlled.
- How the content variations are applied need not be a concern at the design phase, leave that up to the developers.
- What the design wireframes do is deliver the genesis.
- Look at the creative and start to analyze for similarities and differences.

Activity 1. Examine the HTML Creative Delivered by the Designer

Creative Breakdown

- What patterns have you identified?
- What are the advantages to being more "modular"?
- Did you know that we can start with one core template type and create editable templates that allow authors to implement both content and design policies?
- A good practice is to break up the HTML as static pieces to visualize components.
- Let's talk about the "Big X".

Big X = Page level, searchable, SME controlled content



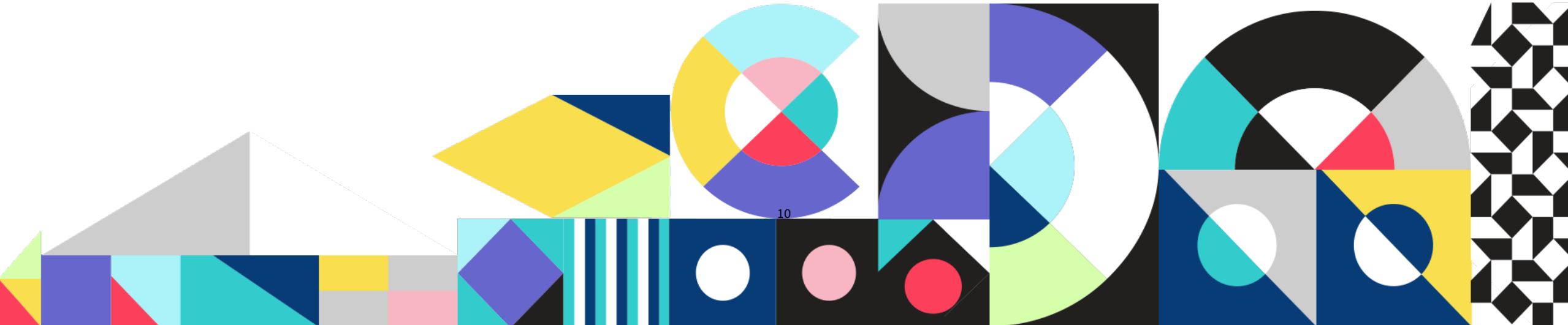
Essential Tools for the rest of the session

- The Site Admin Console
 - URL <http://localhost:4502> and click on "Sites"
- CRXDE Lite browser based Development Environment
 - URL <http://localhost:4502/crx/de>
 - Click Logo, Hammer, CRXDE Lite
- The CRX Package Manager Tool
 - URL <http://localhost:4502/crx/packmgr>
 - Conveniently next door to CRXDE Lite

Quick Tour with help of Teaching Assistants

Template Types in AEM

What do I have?



1

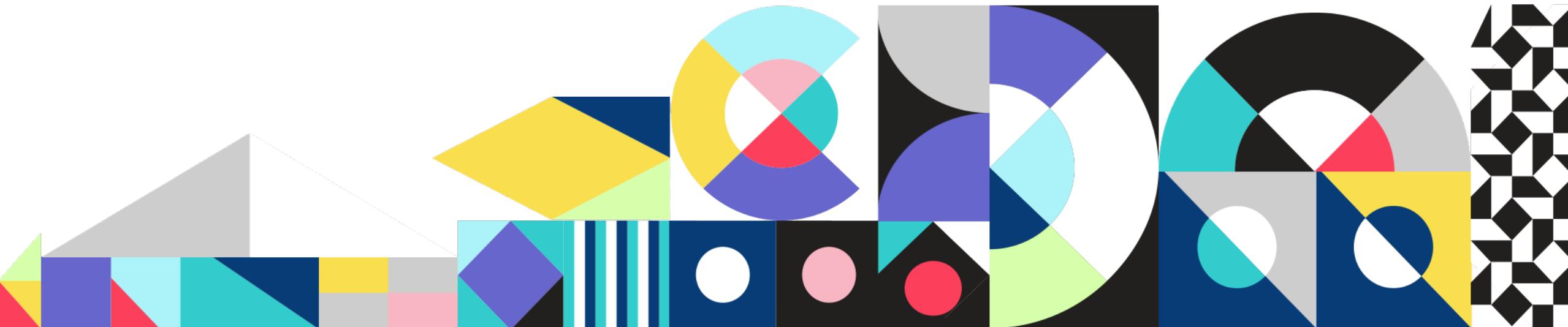
Sub Structure:
Developer Controlled
Invisible to Authors

2

Structure:
Developer Controlled
Selected by Authors

3

Editable:
SME Controlled
Selected by Authors



Let's do the whole site with Editable Templates

- Introduced in AEM 6.2.
- UI enhancements continue to allow more front end creation and configuration.
- Stored in the "conf" folder where more functionality continues to shift.
- Configuration Browser now allows us to start this from the UI.
- These will allow specialized authors to create and edit dynamic templates.
- Predefine Structure, Initial Content and Layout (using modes).

Activity 2 Set up: Let's create a conf folder, a template and convert to a template type.

- This is detailed in the AEM Documentation but let's walk through it...

<https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/page-templates-editable.html>

Activity 2: Creating a Site Level "conf" directory as a tenant base

- A Simple Structure under the conf directory is required.
- Create a folder using Configuration Browser.
- Adjust the Title using properties.
- This is a recommended best practice.

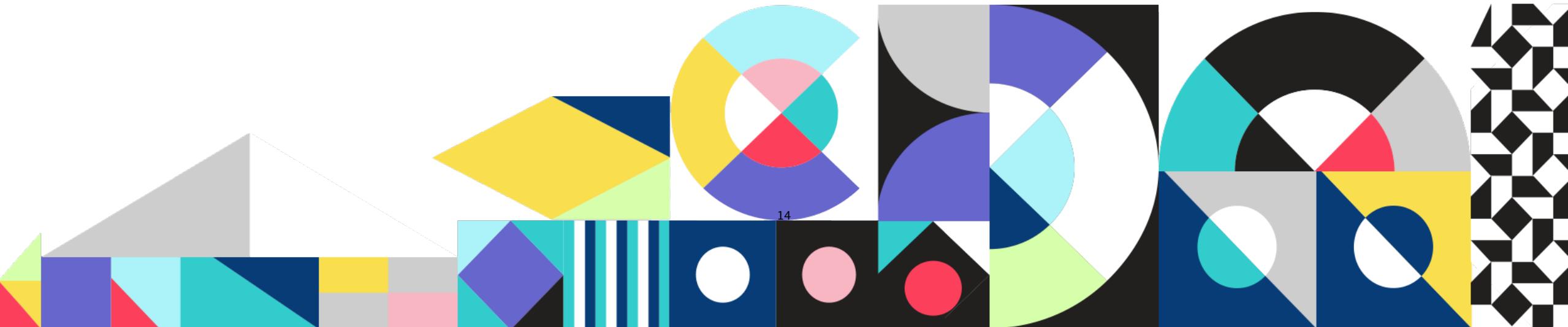
Create a Template Type

- This is a developer task in CRXDE-Lite.
- Create an Empty Page Template using a foundation Template Type.
- Use this as a base to create all of the Templates for our site.

Activity 2: Create a Directory and a Base Template Type for Getaways

Component Creation for Editable Templates

Prep for Activity 3



Editable Templates- Exposing Structure and Content Components

- Static Templates: Structure Components were created and implemented entirely by Developers and completely shielded from the UI.
- Any structure change requires a backend task.
- Content Components are assigned using Design Mode in the UI.
- Design configuration stores values in Current Design Object.

- Editable Templates: Structure Components become drag and drop from the Template Editor UI and configurable using content policies.
- Dialogs and Components Groups are used to group and expose for Template Authors.
- Content Components are assigned using Template Policies.
- Design configuration stores values in Content Policies.

So What?

What are the advantages to using Editable Templates?

- Transfer the capability from IT to SME (no requests, tickets or IT red tape).
- Template Authors can create new and edit existing Templates in 3 modes.
- Policies are really easy to create, reuse and share (Let's talk about that later).

Are there any downsides?

- The only real downside for clients who've used Static and Design Mode is that Design Mode is not exposed in the Editable Template Interface but.....
- The switch to Editable is creating policies for storage: Your existing design_dialogs are immediately recognized and the code in the scripts remains unchanged. “**currentStyle.get()**”.

There is some additional one time prep work up front...

Dialogs

- A dialog (even an empty one) is a "trigger" in AEM Components. A dialog makes that component available in Design Mode for selection by a power user.
- Sooooo Design Mode didn't really go away, it just moved to the "conf" directory.
- We'll look at that a little later.

The Component Group Property

- This value can be added via wizard or later in CRXDE-Lite (or other IDE).
- A Property that "Groups" Components into categories/buckets to be assigned via policy.

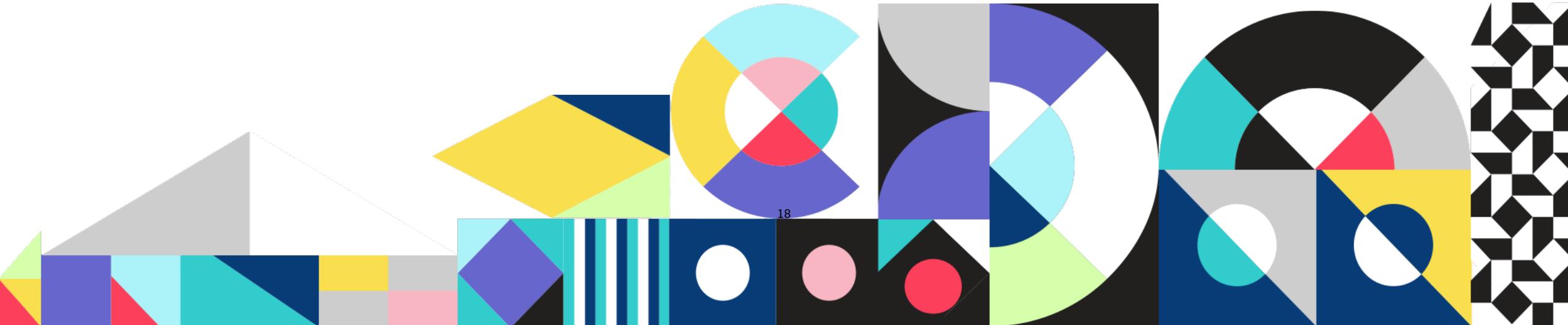
Activity 3. Upload the "Components in Progress" CRX Package and discuss

sling:resourceType property

Guide my templates to my page

sling:resourceSuperType property

Use the foundation core page component!



Now that my Developer created my page component..

Re Assign the resourceType

- When we created our Initial Template Type, it points to /wcm/foundation.
- Change that property to our Getaways Core page so we transfer control.

Observe the resourceSuperType in Gateways Core page

- I extend the Core Page Component so I am “riding the foundation”.
- This is an Adobe Best Practice: Don’t reinvent the wheel!
- (Unless you really want to – you get plenty of rope but THINK!).
 - Technological Debt
 - More to do on upgrade

Site Styling and Core Components

- Improved Editable Template Functionality coming in AEM 6.4
- Core Components as a best practice
 - Version-able so they don't break upgrades
 - Configurable from the Policy level (replaces design mode configurations)
- Though not in scope for this beginner lab, here are a few "sneak peek" slides
- This knowledge should be communicated between business and developers so we avoid the need to refactor too much of what we've rolled out
- Those decisions will come from your content modeling. Don't expect the developers to magically figure that out! They need your guidance!

Core Components (Sneak Peek of 6.4)

- 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 (Sneak Peek of 6.4)

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.

HTML Client Library Manager in AEM

- Helps with organizing and optimizing client side code.
- Combine designer supplied css with client side site functionality.
- Added to page templates using headlibs.html.

Best practices:

- Store libraries in /etc/clientlibs.
- Use site design stored in /etc/designs to drive css and js.
- These are still used in Editable Templates but no longer need to store configuration.
- The advantage to still using this? REUSE and EFFICIENCY (AE-RE-M).

Activity 4. Upload and Install getaways-design.zip Package

Design Content Policies- The new approach

AEM is migrating from Design Mode to Content Policies

- Legacy code still works but has to be re-configured.
- Designs are still portable but now should live in "conf" instead of "etc".
- Policies are better paths to re-use and multi-tenancy.
- I haven't lost "global" styles but have a new layer of abstraction.

Let's see how by adding Logged In Header and Logged In Footer Components.

Activity 5: Upload the Logged In Header and Footer and create a Logged In Template

A Few More Things to Talk About Before We Dive In!

The Framework is Essentially Created

- Navigation and other technical details are not really in scope at this juncture.
- We can now explore possibilities for the remainder of our time.

Tips for creating pages and Templates

- The content components represent "chunks" of content extracted from the creative.
- What are the content velocity requirements of the creative?
- Identify the "Big X". What is the area of a web page that needs free form creativity?
- Which sections can be fixed and simply updated on a regular basis?
- What truly needs to be customized and what can leverage foundation components?
- IT/Development should simply deliver what the business dictates.
- Who really "owns" the content?

Time to Build, Play and Emulate the Creative

Open the design creative in another browser and try to recreate the pages.

Try updating Templates in the editor and explore the resulting changes.

Additional Packages to Install

- getaways-completed-templates.zip
- getaways-components-complete.zip
- getaways-content.zip
- getaways-core-content-examples.zip

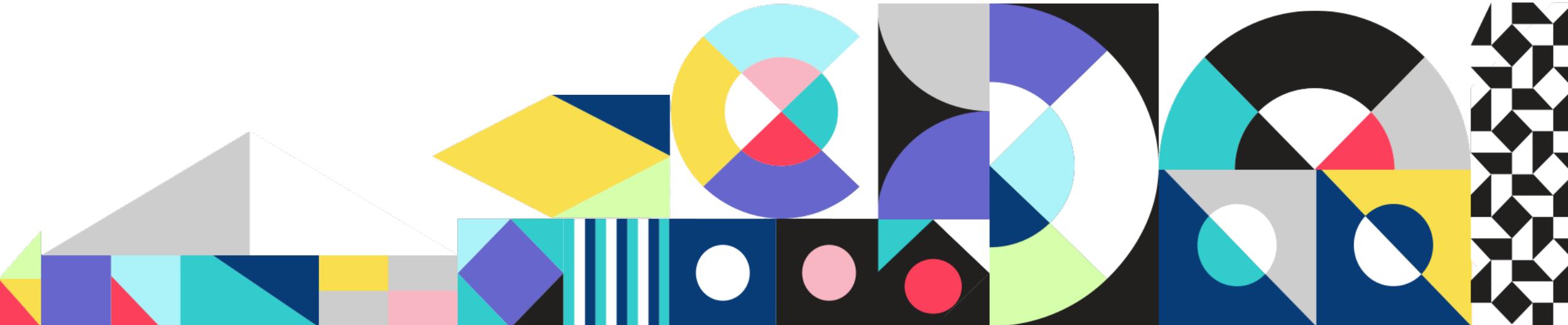
These Packages will ultimately create a "completed" site. Once these are installed, try updating the component content. We will compare "core" components and the steps to extend.

Q & A

Dennis Selfridge

selfridg@adobe.com

<https://www.linkedin.com/in/dennis-selfridge-6973081/>



Take the Survey to win!

(Survey section of the mobile app)

SESSION PRIZE

one per session

STARBUCKSCARD



\$10 Starbucks
Card

DAY 1



Bash
Experience

DAY 2



Signed football
helmet

DAY 3



Amazon Echo Show
& Cloud Cam
Bundle



Adobe Experience Manager Community

LAB Thread: <http://bit.ly/lab0732>

Join Community : <https://adobe.com/go/AEMCommunity>



Ask the
Community



Share
Resources



Interact with Adobe
Product Experts



Attend Live
Webinars



Be an
Ambassador



Get
Recognized



Share
Feedback

SUMMIT

ADOBE SUMMIT | MARCH 25-29, LAS VEGAS



Adobe

MAKE IT AN EXPERIENCE