



adaptTo()

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

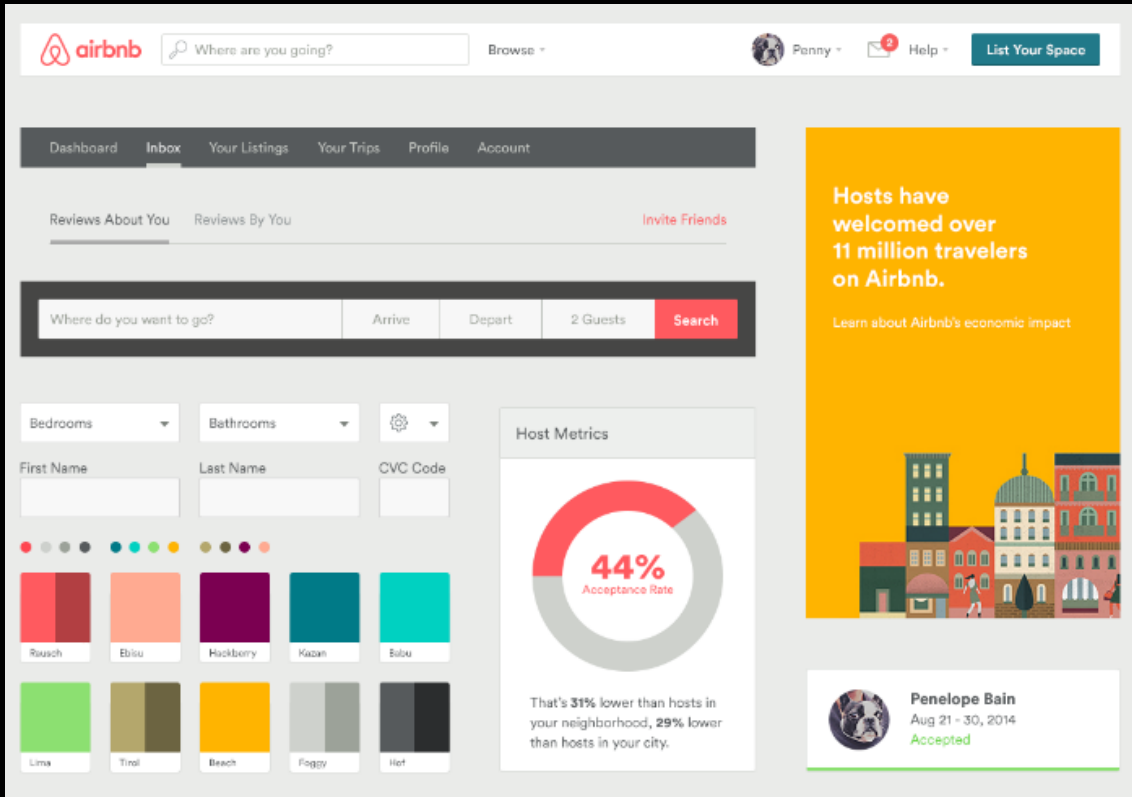
Pattern Lab for Sling – Atomic Design approach for building Sling based websites

Kamil Ciecierski, Freelance AEM Consultant

1. Importance of website style guides.
2. Atomic Design methodology.
3. Pattern Lab for Sling.

Importance of website style guides

Why do we need website style guides?



The screenshot displays the Airbnb website interface, illustrating a consistent design style. Key elements include:

- Header:** The Airbnb logo, a search bar with the placeholder text "Where are you going?", and navigation links for "Browse", "Penny", "Help", and "List Your Space".
- Navigation:** A dark grey bar containing "Dashboard", "Inbox", "Your Listings", "Your Trips", "Profile", and "Account".
- Search Bar:** A prominent search bar with fields for "Where do you want to go?", "Arrive", "Depart", "2 Guests", and a red "Search" button.
- Filters:** Dropdown menus for "Bedrooms" and "Bathrooms", and input fields for "First Name", "Last Name", and "CVC Code".
- Color Palette:** A grid of color swatches with names: Rousch, Ebiu, Hackberry, Kazan, Bibu, Lima, Tiral, Beach, Foggy, and Hat.
- Host Metrics:** A circular progress indicator showing a "44% Acceptance Rate". Below it, text states: "That's 31% lower than hosts in your neighborhood, 29% lower than hosts in your city."
- Host Card:** A card for "Penelope Bain" with a profile picture, dates "Aug 21 - 30, 2014", and the status "Accepted".
- Hero Section:** A large yellow banner with the text "Hosts have welcomed over 11 million travelers on Airbnb." and a link to "Learn about Airbnb's economic impact".

Why do we need website style guides?

„Education is as important as documentation. A style guide can show clients that websites are systems rather than collections of pages.“

Anna Debenham, „Front-End Style Guides“

Why do we need website style guides?

- keep information about design language
- promote consistency and cohesion across a user interface
- allow fast prototyping and validation of designs
- help to keep neatness in FE code structure

Style guides in website integration process

Website integration process

Designers

Site designs

FE Developers

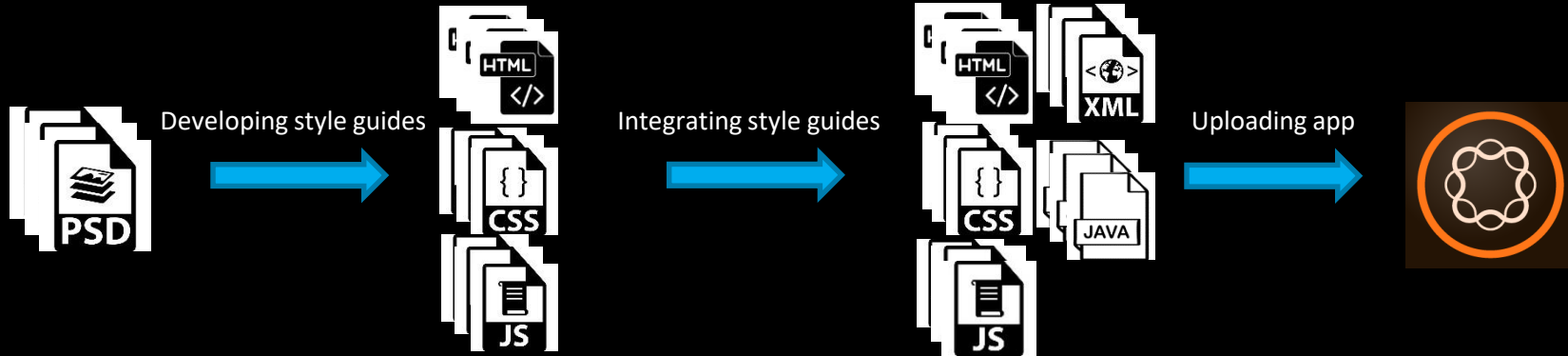
Style guides

AEM Developers

AEM package

QA, Designers, PO

AEM integration



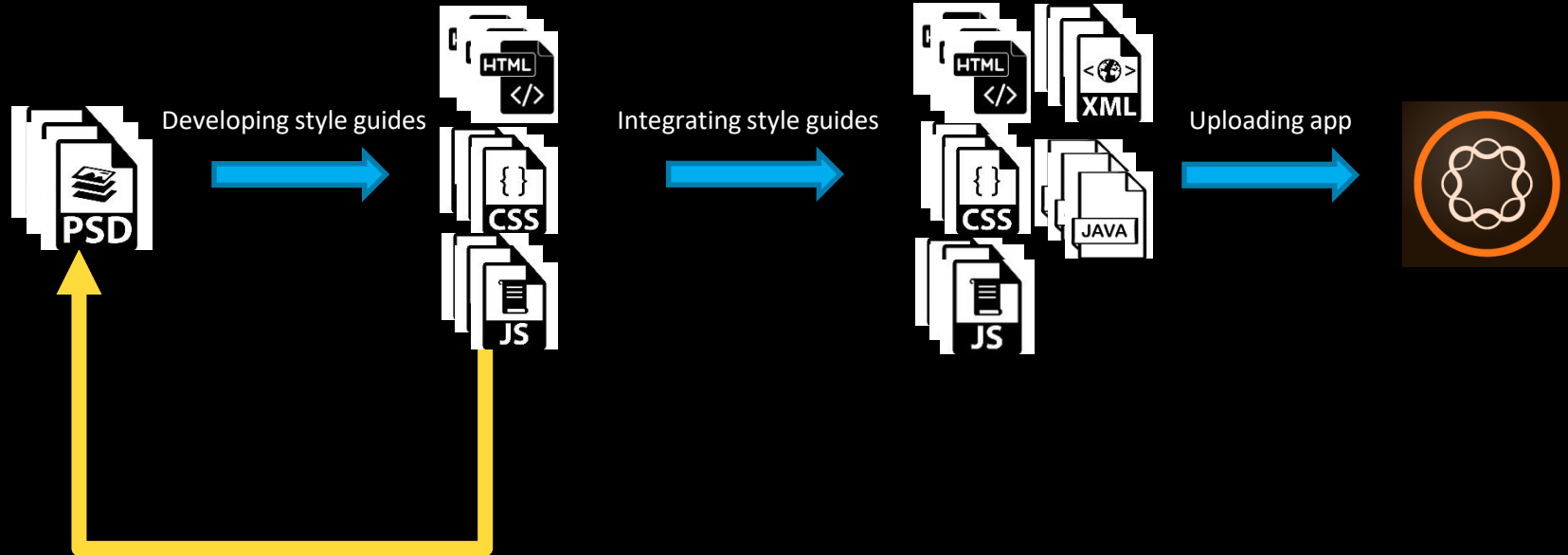
Website integration process – feedback loop

Designers
Site designs

FE Developers
Style guides

AEM Developers
AEM package

QA, Designers, PO
AEM integration



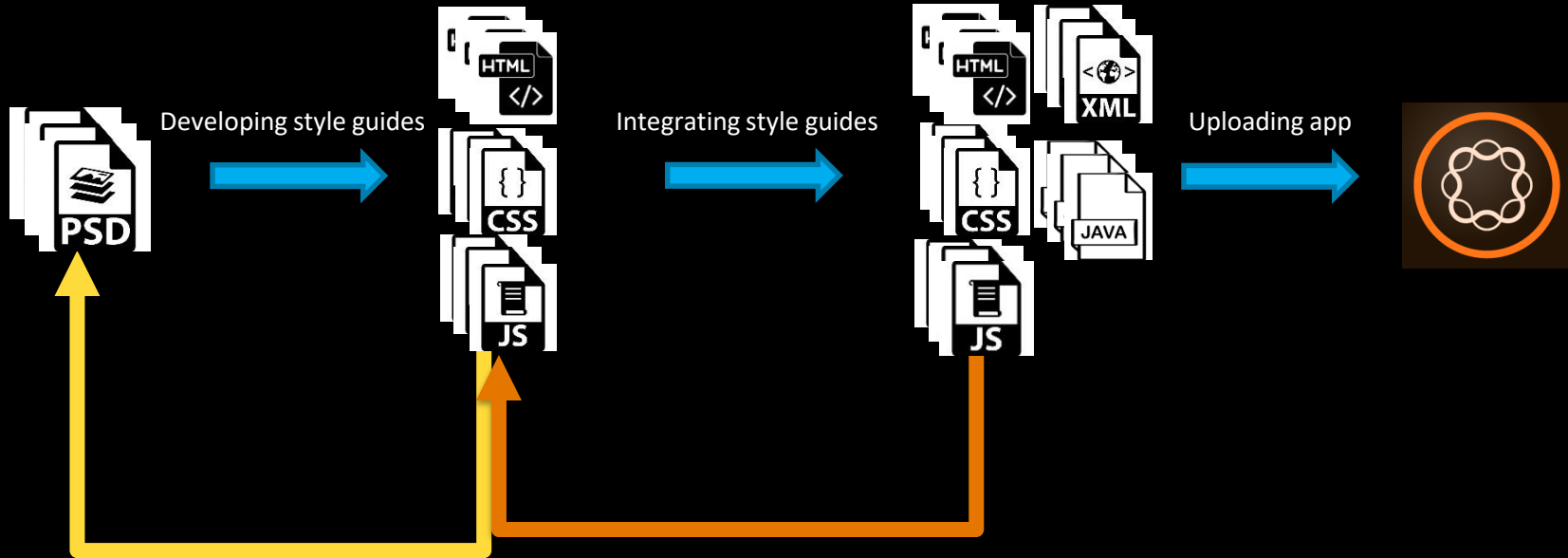
Website integration process – feedback loop

Designers
Site designs

FE Developers
Style guides

AEM Developers
AEM package

QA, Designers, PO
AEM integration



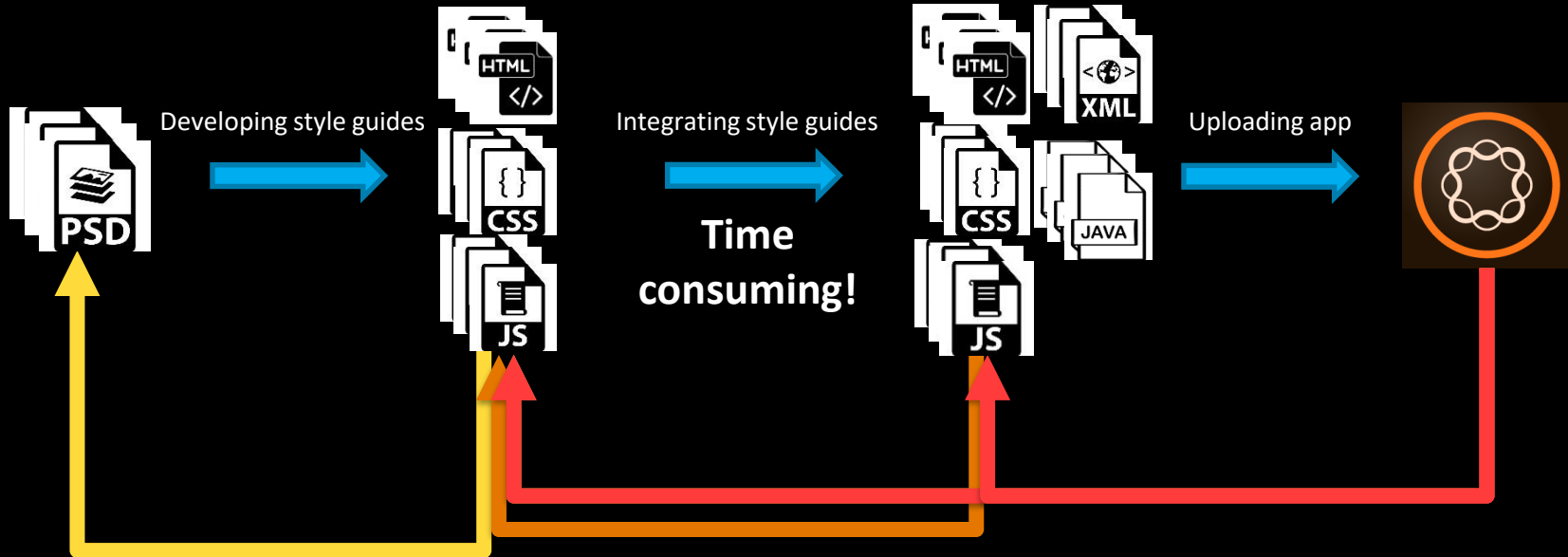
Website integration process – feedback loop

Designers
Site designs

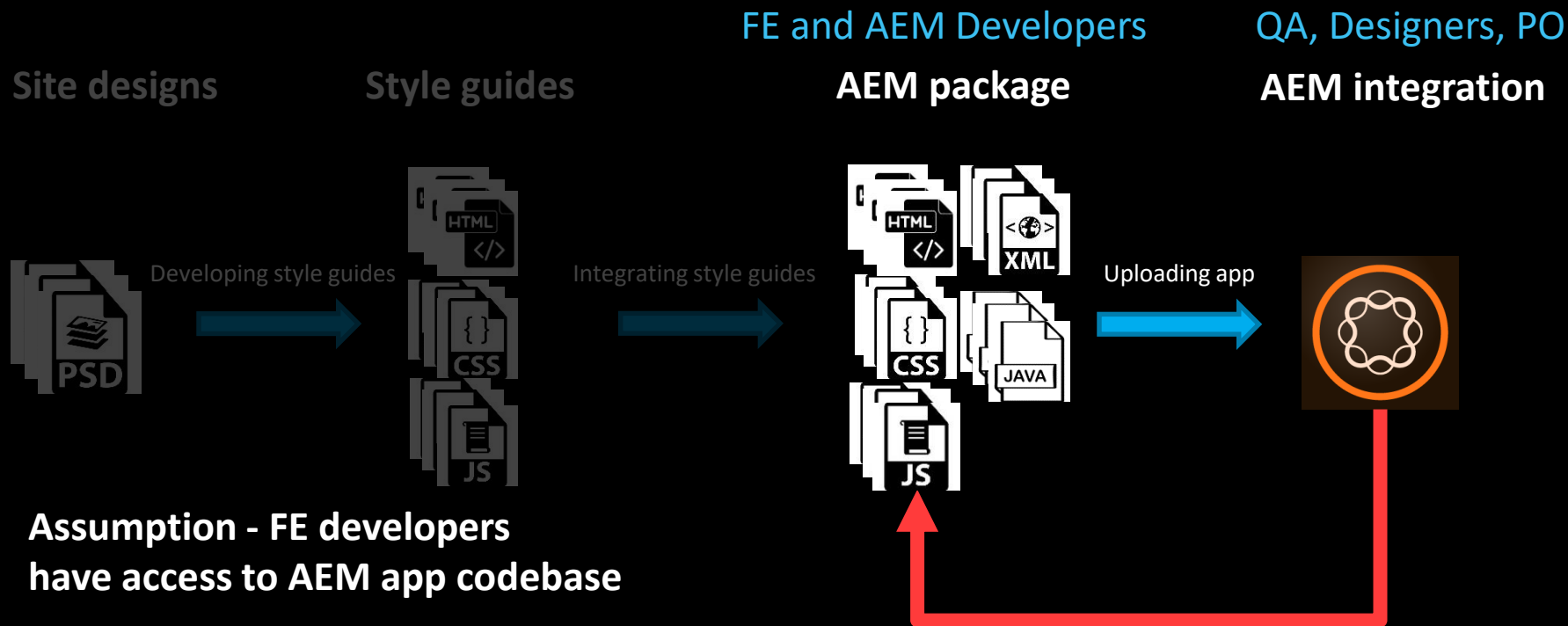
FE Developers
Style guides

AEM Developers
AEM package

QA, Designers, PO
AEM integration



Abandoning style guides



What about new website features?

Usual AEM site integration process

Designers

Site designs



Developing style guides



FE Developers

Style guides

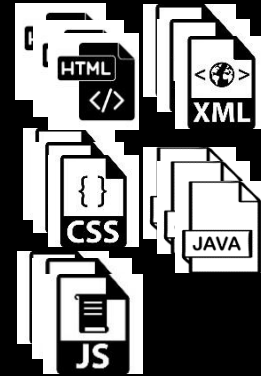


Integrating style guides



AEM Developers

AEM package



Uploading app



QA, Designers, PO

AEM integration



Style guides regression

Designers

Site designs

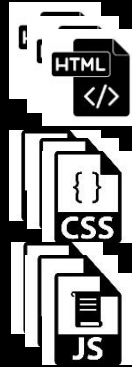


Developing style guides



FE Developers

Style guides

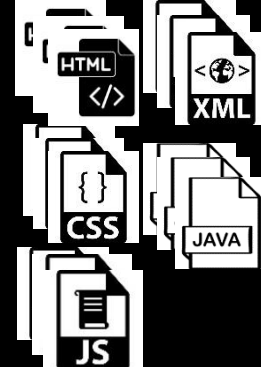


Integrating style guides



AEM Developers

AEM package



Uploading app



QA, Designers, PO

AEM integration



← Keeping style guides in sync with codebase

Style guides integration challenges

- **present only few content scenarios**
 - no readiness for dynamic content
 - features for design authoring not taken into account
- **lack of components modularity**
 - dependencies between components
- **aligning with authoring features**
- **manual rewriting of HTML examples into HTL scripts**
- **considered as auxiliary project rather than designs foundation**
- **lack of clear methodology**



Website style guides integration improvements

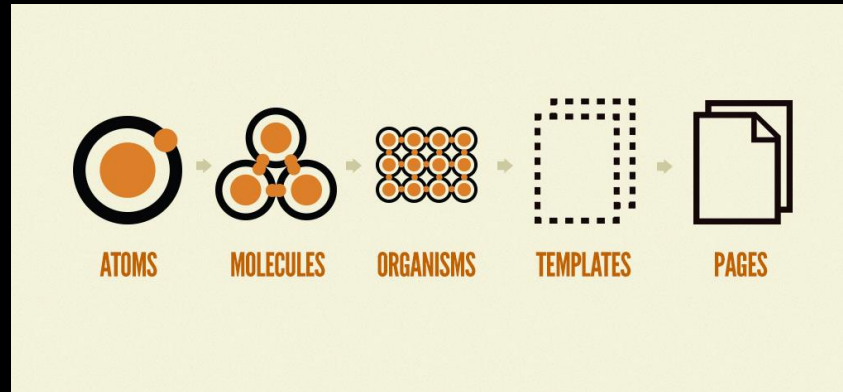
- designs sessions to produce content more examples
- sharing information about authoring features
- testing style guides separately
- style guides integration automation – requires additional assumptions:
 - AEM application and style guides integrated with templating language
 - dependency on custom solutions
 - rendering website client-side
- introducing modularity to style guides components
 - building **pattern libraries**

Atomic Design

methodology for building living style guides

Introduction to Atomic Design

„Atomic design is a methodology composed of five distinct stages working together to create interface design systems in a more deliberate and hierarchical manner.“



Brad Frost

Atomic Design patterns - atoms

SEARCH THE SITE

LABEL

ENTER KEYWORD

INPUT

SEARCH

BUTTON

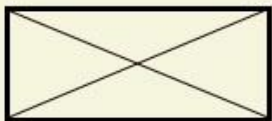
Atomic Design patterns - molecules

SEARCH THE SITE

ENTER KEYWORD

SEARCH

Atomic Design patterns - organisms



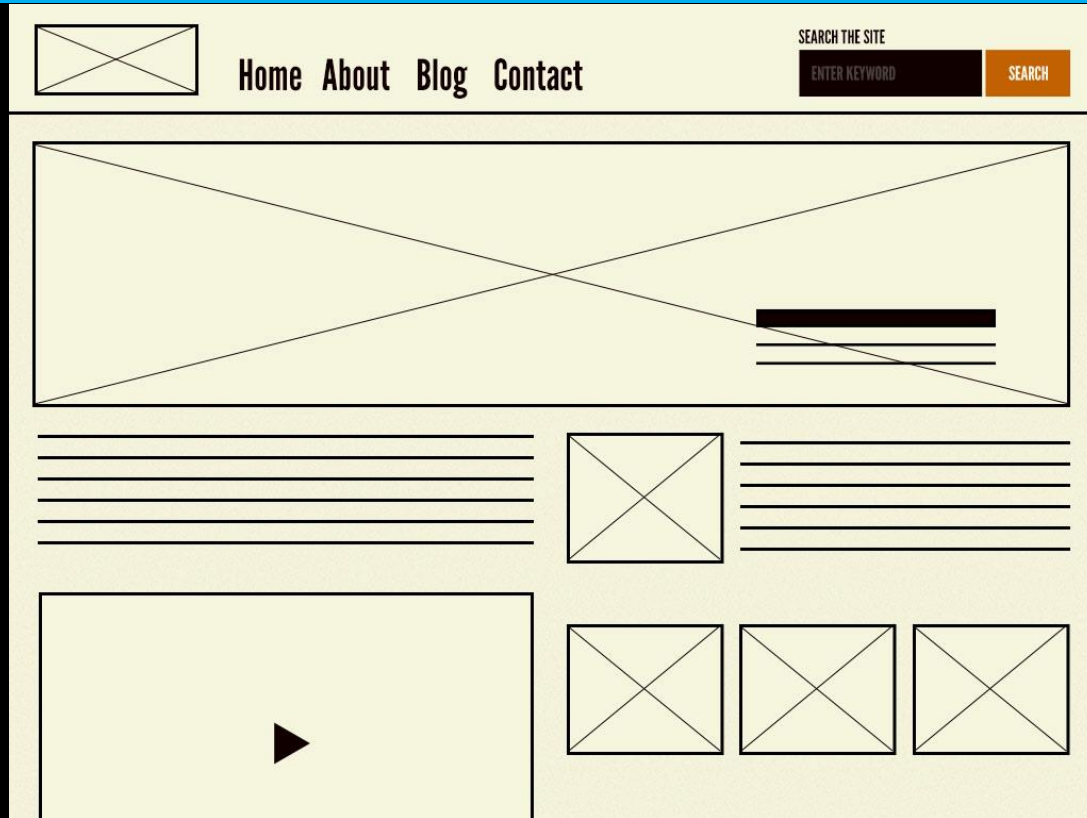
[Home](#) [About](#) [Blog](#) [Contact](#)

SEARCH THE SITE

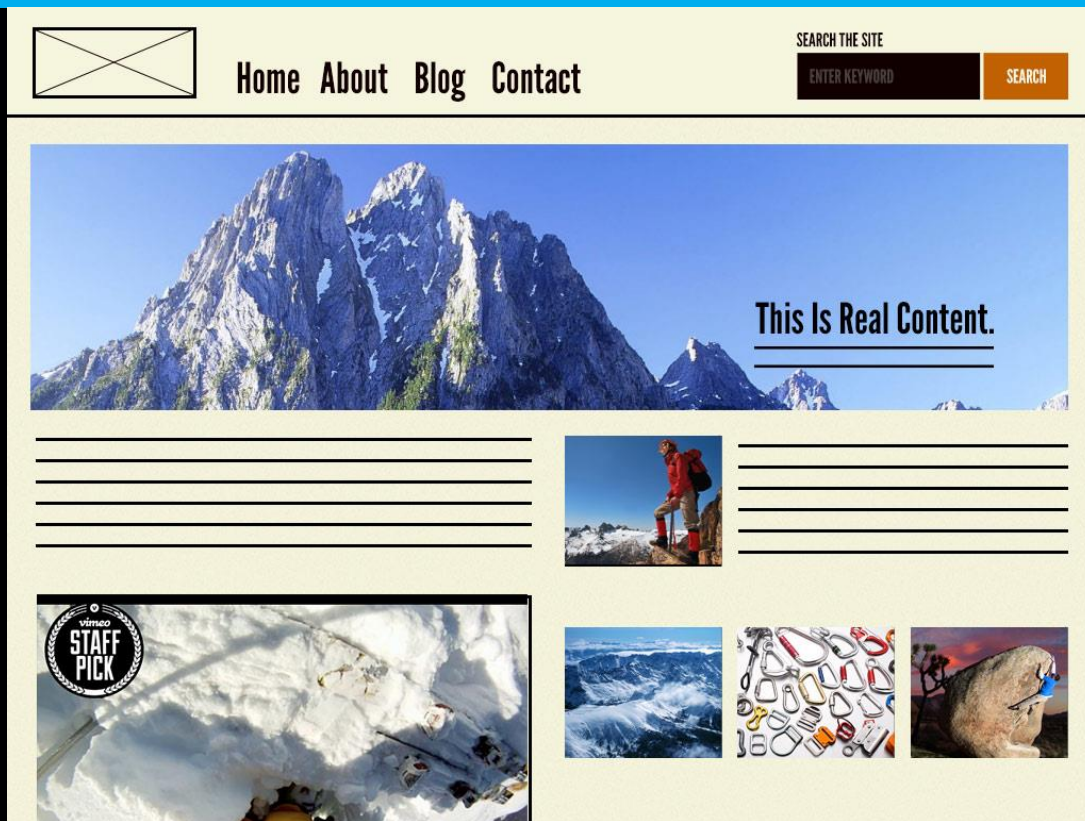
ENTER KEYWORD

SEARCH

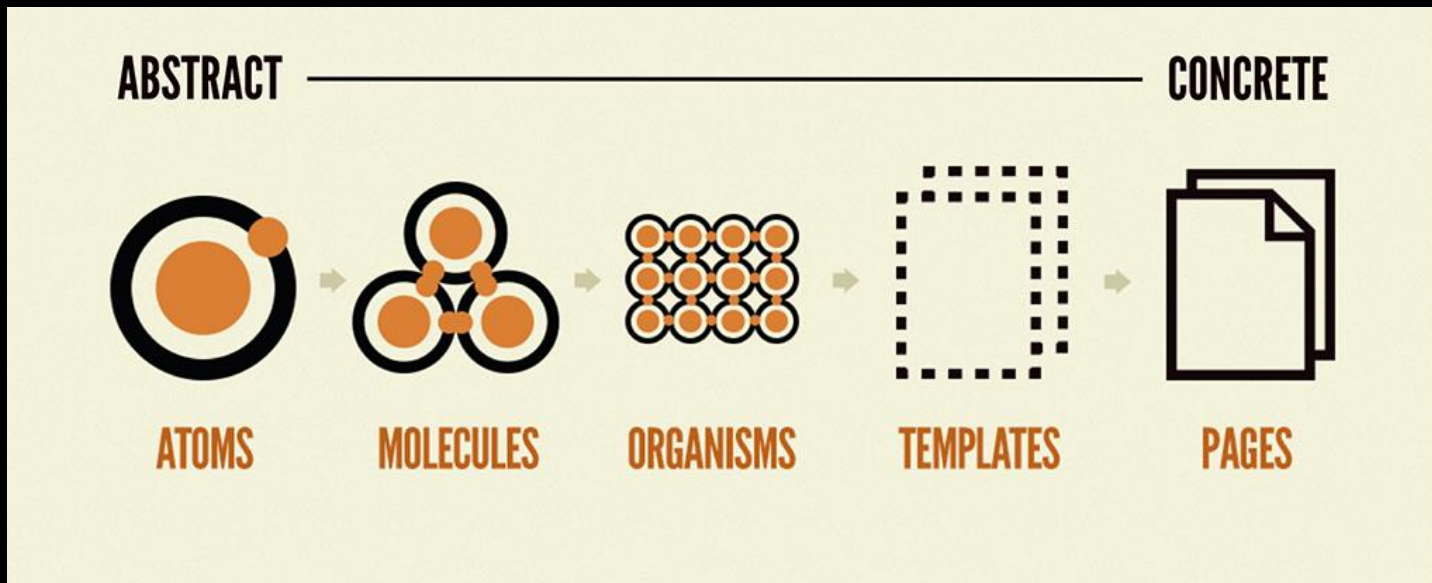
Atomic Design patterns - templates



Atomic Design patterns - pages



Atomic Design patterns = AEM components?



Pattern Lab - tool for building atomic design



adaptTo()systems

ATOMS • MOLECULES • ORGANISMS • TEMPLATES • PAGES • ALL

SIZE 1906PX / 119.1EM S M L FULL RAND DISCO ⚙

Blocks

Hero

The hero is a large block that typically appears at the top of the page and contains a large full-bleed image and a title.

The Hero pattern is included in the following patterns: • [templates-homepage](#)

MUSTACHE

HTML

```
<a href="{{ ur1 }}" class="c-block-hero">
  {{> atoms-hero:c-block-hero_img }}

  <h2 class="c-block-hero_headline">{{ headline.medium }}</h2>
</a><!-- end c-block-hero-->
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit iopa. (76 characters)

1280x720
16:9

Media Block • inreview

The media block is based on Nicole Sullivan's popular media object pattern. It contains a media element

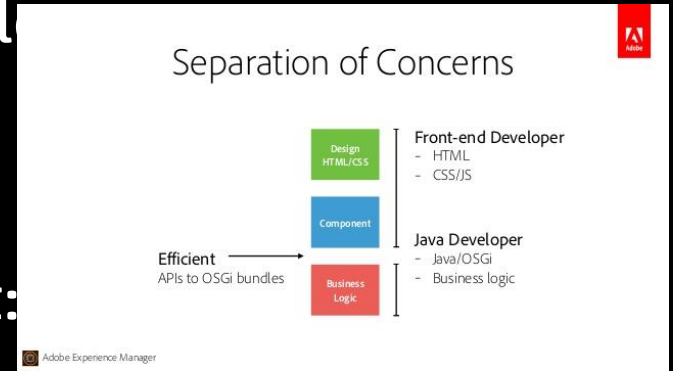
Atomic Design for WCM project development

- patterns are suitable for WCM:
 - modular
 - dynamic content ready
 - adaptable to different (also future) channels
 - reusable in different contexts
- nesting of patterns increases code reusability and reduces time needed to apply design changes
- Pattern Lab allows to build easily and quickly modular patterns
- but still, patterns require FE integration with AEM application

Developing FE in AEM vs developing style guides

Developing FE straight in AEM

- thanks to HTL and JS Use API, possible to build view without authoring logic
- easier with tools supporting content repository sync and HTL development:
 - Plugins for IDEs: Eclipse, IntelliJ, Brackets
 - Slang - Gulp module
- integration of FE build systems with AEM application
 - front-end-maven plugin
 - AEM Sass compiler



Developing FE straight in AEM - challenges

- **more time consuming than building style guides**
 - dependency on components, templates and content creation
- **difficult to validate HTL scripts with different data examples**
 - dependency on authoring development or script rework required
- **less visibility of design dependencies between components**
 - might enforce developing new components styles and scripts separately, to reduce impact on the rest of them
- **AEM component libraries might not keep enough data about design language to build new features**

~~Developing FE in AEM vs developing style guides~~

Style guides within Sling/AEM!

ORGANISMS > GLOBAL

organisms-global-footer-footer

footer.html > footer > footer.json

Contains the following patterns: [molecules-navigation/footer-nav/footernav](#), [molecules-media/logo/footer-link/logofooterlink](#)

Included in the following patterns: [pages-blog-index/blogindex](#), [pages-dashboard-dashboard-default](#), [pages-dashboard-dashboard-backed](#), [pages-dashboard-dashboard-new-user](#), [pages-detail-detail-about](#), [pages-detail-detail-blog](#), [pages-detail-detail-contact](#), [pages-homepage-homepage](#)

HTL DATA HTML

```
<template data-sly-template.footer="${@ logoFooterLink, footerNav, info, year, company}">
  <footer class="c-footer" role="contentinfo">
    <div class="l">
      <sly data-sly-use.pattern="patternlab-demo/molecules/media/logo/footer-link.html"
        data-sly-call="{pattern.logoFooterLink @ link=logoFooterLink.link, logo=logoFooterLink.logo, style=logoFooterLink.style}>
      <sly data-sly-use.pattern="patternlab-demo/molecules/navigation/footer-nav.html"
        data-sly-call="{pattern.footerNav @ items=footerNav.items}"></sly>
      <p class="c-footer__info">${info @ context='html'}</p>
      <p class="copyright">&copy; ${year} ${company.name}. All rights reserved.</p>
    </div><!-- end l -->
  </footer><!-- end c-footer -->
</template>
```



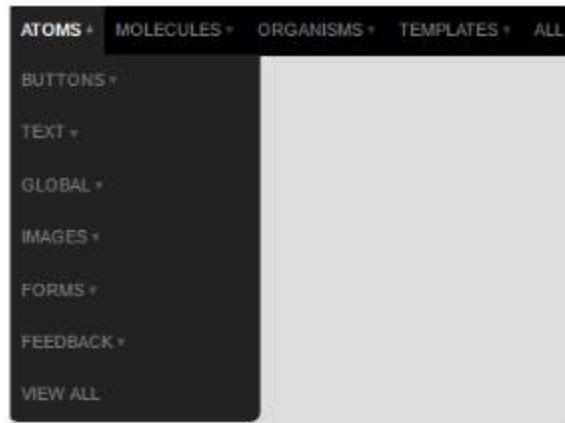
[About](#) [Blog](#) [Contact](#) [Login](#)

This is a demonstration of what's possible with [Pattern Lab](#). [Download Pattern Lab](#), check out the [documentation](#), or learn more about [atomic design](#).

© 2017 HikeTracker. All rights reserved.

Pattern Lab for Sling

- apps.patternlab-demo
 - atoms
 - buttons
 - feedback
 - forms
 - global
 - images
 - text
 - molecules
 - organisms
 - templates



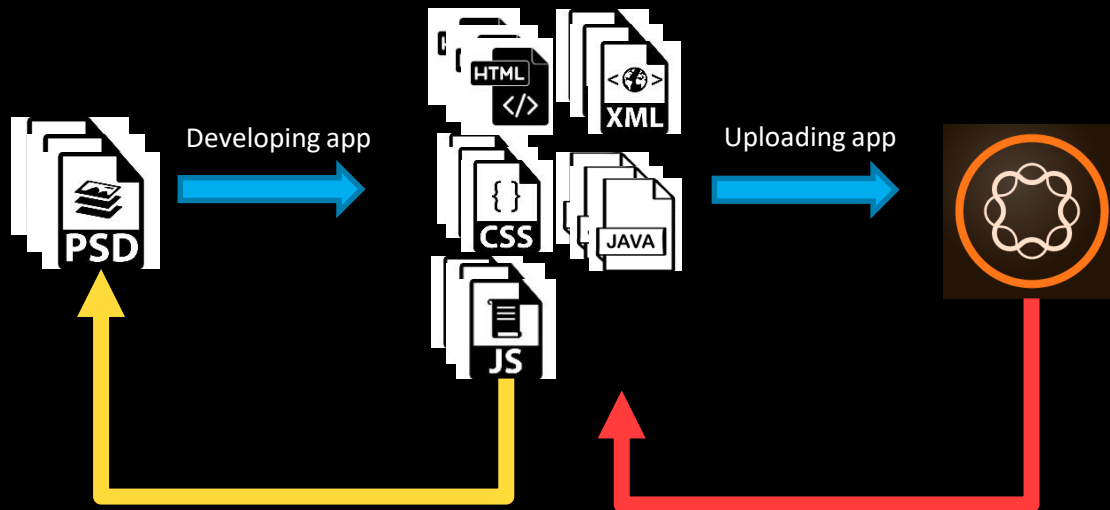
Demo

Simplified workflow with Pattern Lab for Sling

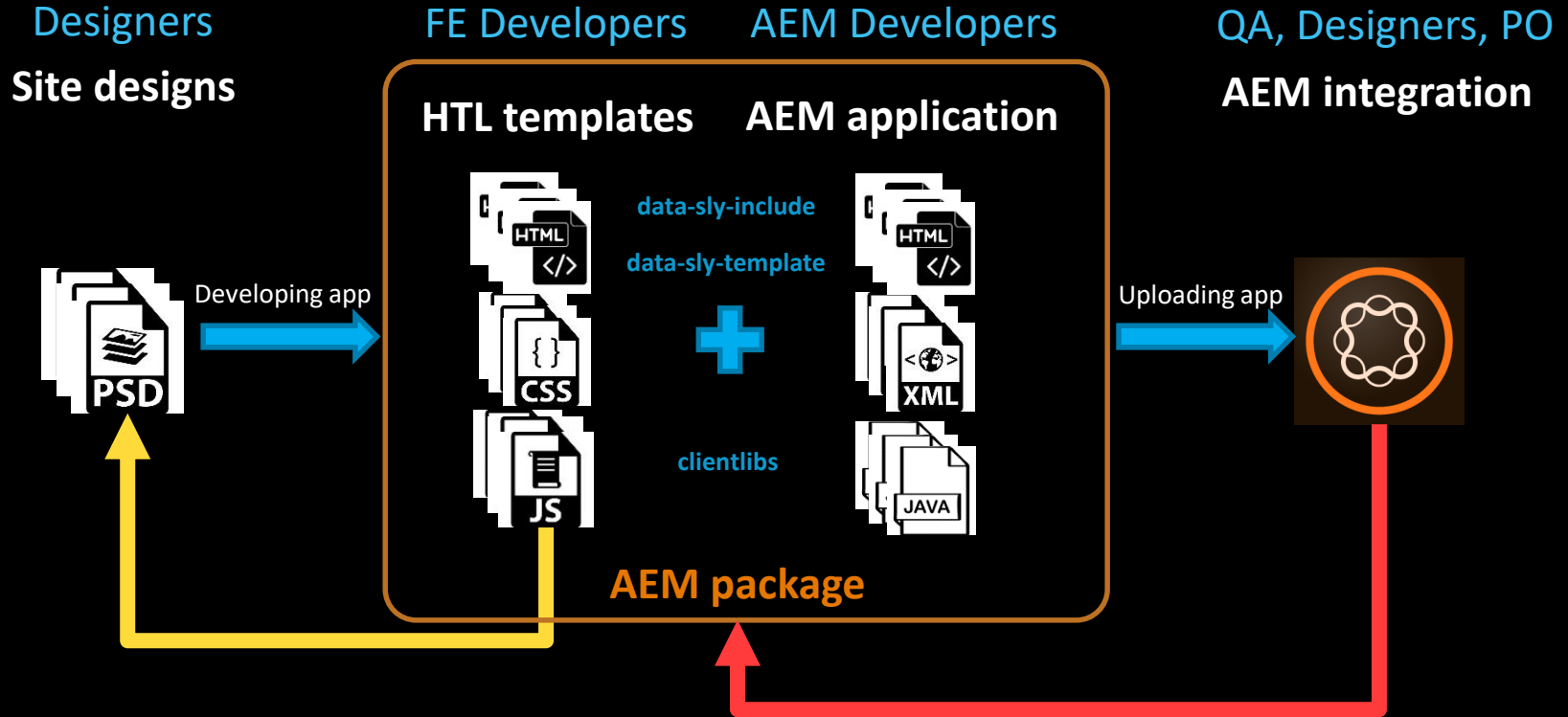
Designers
Site designs

FE and AEM Developers
AEM package
with Pattern Lab

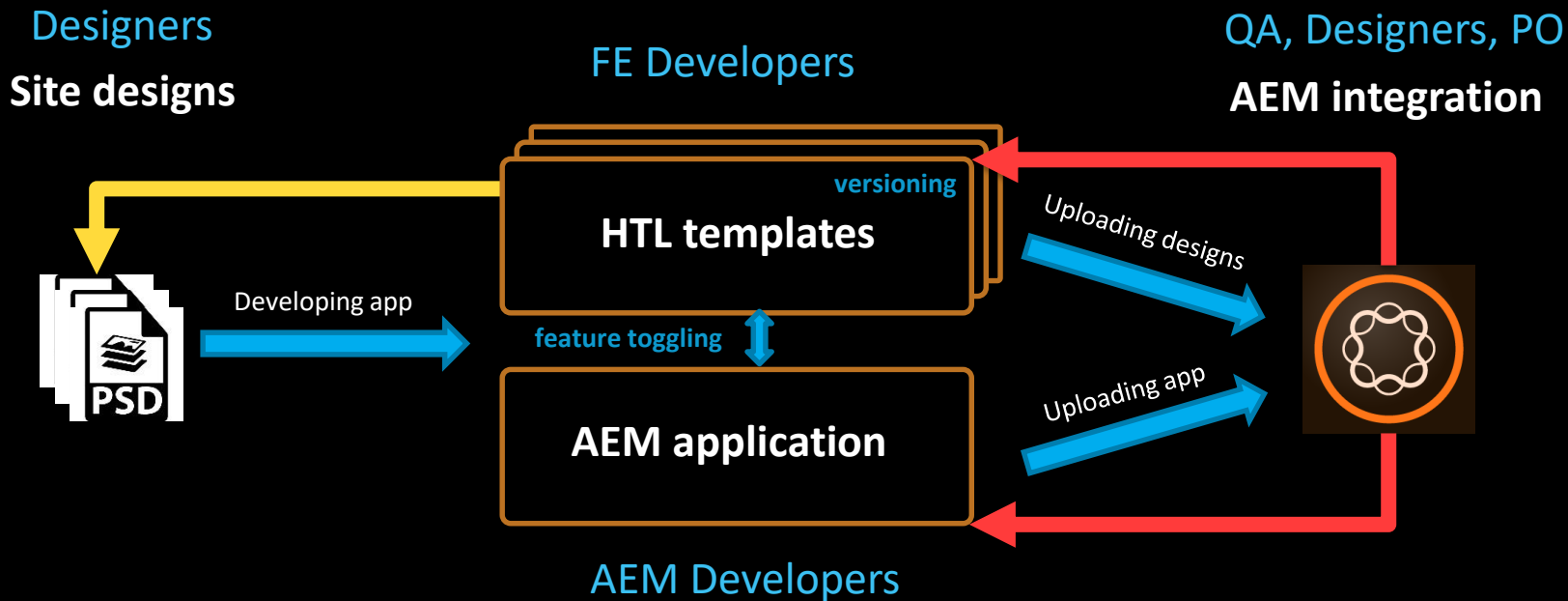
QA, Designers, PO
AEM integration



Simplified workflow – segregation of duties



Simplified workflow – view and logic separation



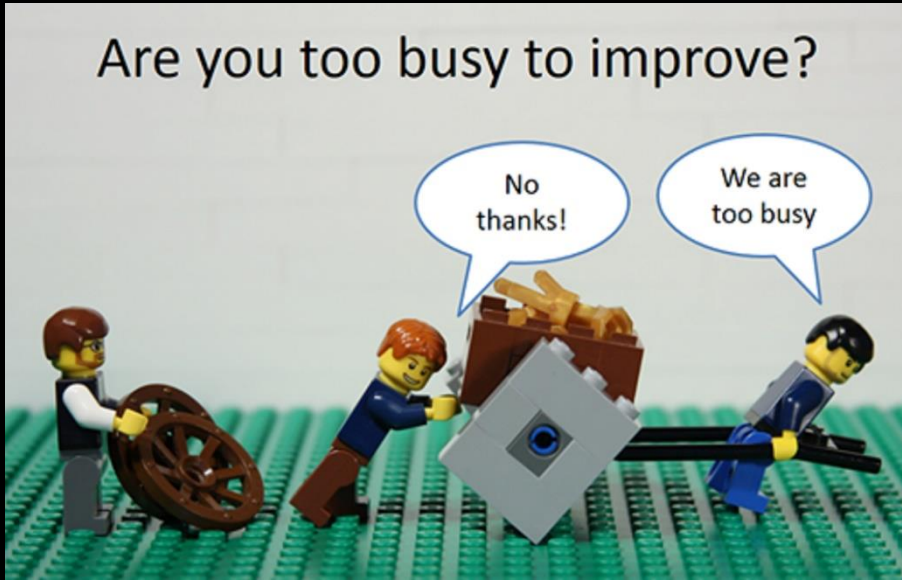
Pattern Lab for Sling – benefits

- no more style guides integration
- style guides are not only auxiliary projects anymore
 - design language embedded in codebase
- true separation of view and business logic
 - testing view without logic using *raw* patterns
- segregation of duties
 - FE and AEM developers can do their best in what they love
 - flexibility on organizing developers workflow
- fast designs prototyping and validation straight on Sling/AEM instance
- ~~no dependency on custom solutions or client side rendering~~

Future work

- **Integration with AEM**
 - Pattern Lab Template
 - Pattern component
 - AEM Annotations
 - AEM projects - tasks
- **Patterns presenting different Sling context**
 - mocking Sling bindings like resource, request

Are you ready for your next project? 😊



*„The hard part
is building the machine
that builds the product.”*

Dennis Crowley, Foursquare CEO

Appendix

References

- Atomic Design: <http://atomicdesign.bradfrost.com/>
- Pattern Lab: <http://patternlab.io/>
- Pattern Lab for Sling: <https://github.com/deepthinkit/patternlab-for-sling>
- Pattern Lab for Sling – demo content: <https://github.com/deepthinkit/patternlab-for-sling-demo>
- Front end style guides: <http://maban.co.uk/projects/front-end-style-guides/>

Useful tools for building FE in AEM

- Gulp Slang: <https://www.npmjs.com/package/gulp-slang>
- AEM Developer Tools: <https://eclipse.adobe.com/aem/dev-tools/>
- AEM Brackets Extension: <https://docs.adobe.com/docs/en/aem/6-3/develop/dev-tools/aem-brackets.html>
- Frontend-maven-plugin: <https://github.com/eirslett/frontend-maven-plugin>
- AEM Sass Compiler: <https://github.com/mickleroy/aem-sass-compiler>

Images used in presentation

- Slide 3 - <https://1.bp.blogspot.com/-XeE07u3bQRU/WBvCieou3DI/AAAAAAAAAC2I/WPdjK6hRwHUR2WNxbphb3mG453ft8YM8wCK4B/w1200-h630-p-k-no-nu/airbnb.png>
- Slide 18-25 - <http://bradfrost.com/blog/post/atomic-web-design/>
- Slide 28 - <https://image.slidesharecdn.com/aem-best-practices-150904054458-lva1-app6891/95/aem-best-practices-for-component-development-6-638.jpg?cb=1441617960>
- Slide 38 - <https://fernandofranzini.files.wordpress.com/2017/01/too-busy-to-improve.png>