



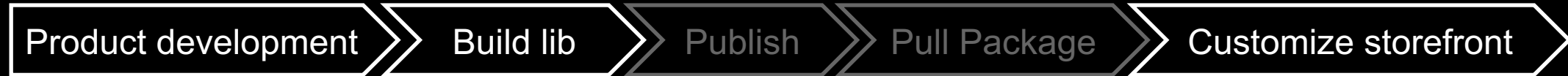
CSS **Extensible Architecture** for the JavaScript Storefront

Tobias Ouwejan, Front-end Architect, SAP Customer Experience

Goals

- Deliver configurable styles
- Deliver extensible styles
- Deliver component lib that allow for style customisation without replacing components
- Deliver optimized styles (no duplication)

Problem #1 SASS is not processed as we ship compiled components



Component Library



Component Package

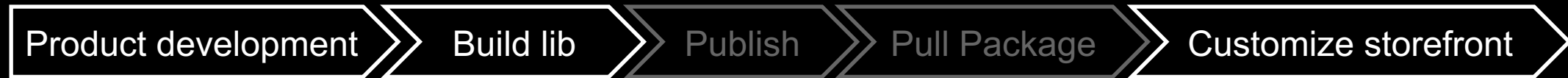


Application



```
$cx-product-images-pdp-items-trbl-margin: 20px;
```

Problem #2 Selectors will never get the highest specificity



Component Library



Component Package



Application



```
.y-product-images {  
  &_items {  
    margin: 20px;  
  }  
}
```

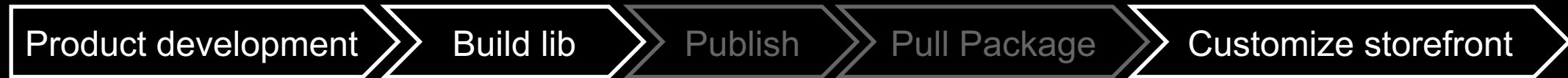
SOLUTION

Problem #1 SASS is not processed as we ship compiled components

Problem #2 Selectors will never get the highest specificity

- Use CSS variables for CSS configuration
- Move scss to style library (so SASS is compiled)

Problem #3 Spartacus styles will not get replaced in the compiled bundles



Component Library



Component Package



Application



```
.selector-x {  
  rule: value;  
}
```



```
.selector-x {  
  rule: custom-value;  
}
```

SOLUTION

Problem #3 Spartacus styles will not get replaced in the compiled bundles

- Move CSS to library
- Use placeholder selectors
- Use sass feature toggles

BONUS

- Use flags in themes to demonstrate styles

storefrontstyles/scss/[...]/_pictures.scss

```
@if $overlayWhileLoading == true {  
  &.initialized:before {  
    @extend %overlay !optional;  
  }  
}
```

Problem #4 Mark-up is too opinionated and doesn't provide enough flexibility

```
<div class="y-product-reviews__tab-review--container">
  <div class="y-product-reviews__tab-review--details">
    <h5>{{review.headline}}</h5>
    <div class="row">
      <div class="col-auto mr-auto">
        <y-star-rating [rating]="review.rating" [disabled]="true"></y-star-rating>
      </div>
      <div class="col-auto">
        <div class="y-product-reviews__tab-review--date">{{review.date | date}}</div>
      </div>
    </div>
  </div>
</div>
<p class="y-product-reviews__tab-review--text"> {{review.comment}} </p>
  <span class="y-product-reviews__tab-review--name">
    {{review.alias? review.alias : review.principal?.name}}
  </span>
</div>
```


SOLUTION

Problem #4 Mark-up is too opiniated and doesn't provide enough flexibility

```
<div class="review">
  <div class="title">{{review.headline}}</div>
  <cx-star-rating [rating]="review.rating" [disabled]="true"></cx-star-rating>
  <div class="name">{{review.alias? review.alias : review.principal?.name}}</div>
  <div class="date">{{review.date | date}}</div>
  <div class="text">{{review.comment}}</div>
</div>
```

BONUS

- Smaller DOM
- Simpler selectors

Problem #5 DOM is polluted with unnecessary mark-up

```
<div class="y-product-details__item-images">
```

```
  <cx-product-images>
```

```
    <div class="y-product-images">
```

```
      [...]
```

```
    </div>
```

```
  </cx-product-images>
```

```
</div>
```

SOLUTION

Problem #5 DOM is polluted with unnecessary mark-up

```
<cx-product-images>  
  [...]  
</cx-product-images>
```

- Avoid outer DIV
- Use **:host** selector for component styles
- Avoid BEM (see next)

BONUS

- Smaller DOM (performance)
- Simpler selectors

Problem #6 Selectors are over engineered

- BEM was introduced to provide unique selectors cross a single DOM
- Today's JS frameworks provides component encapsulation
- There's still the need for a clear convention to selectors

SOLUTION

Problem #6 Selectors are over engineered

- Avoid BEM
- Use simple selectors
- Use clean mark-up, no nesting if possible
- Use HTML semantics where possible

Problem #7 Some components are duplicated in the DOM

```
<ng-container *cxOutlet="outlets.TAB">
  <div class="y-product-details__tabset" #tabSetWrapper>
    [...]
    <y-product-attributes [product]="product"></y-product-attributes>
    [...]
  </div>

  <!-- Mobile accordion -->
  <div class="y-product-details__tabset-mobile">
    <ngb-accordion #acc="ngbAccordion">
      [...]
      <y-product-attributes [product]="product"></y-product-attributes>
      [...]
    </ngb-accordion>
  </div>
</ng-container>
```

SOLUTION

Problem #7 Some components are duplicated in the DOM

- Use CSS to control responsive layouts