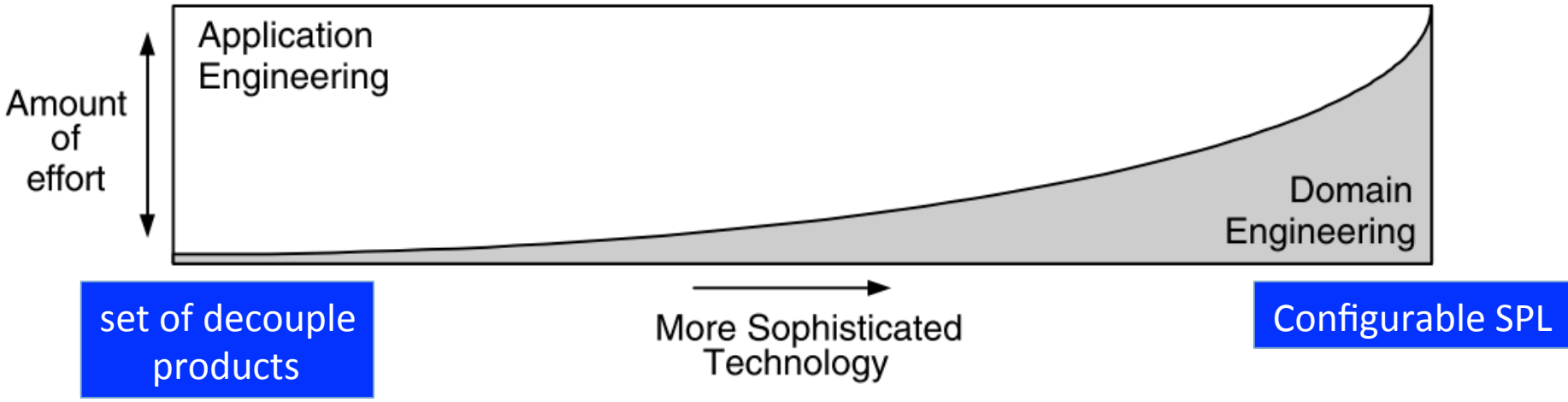
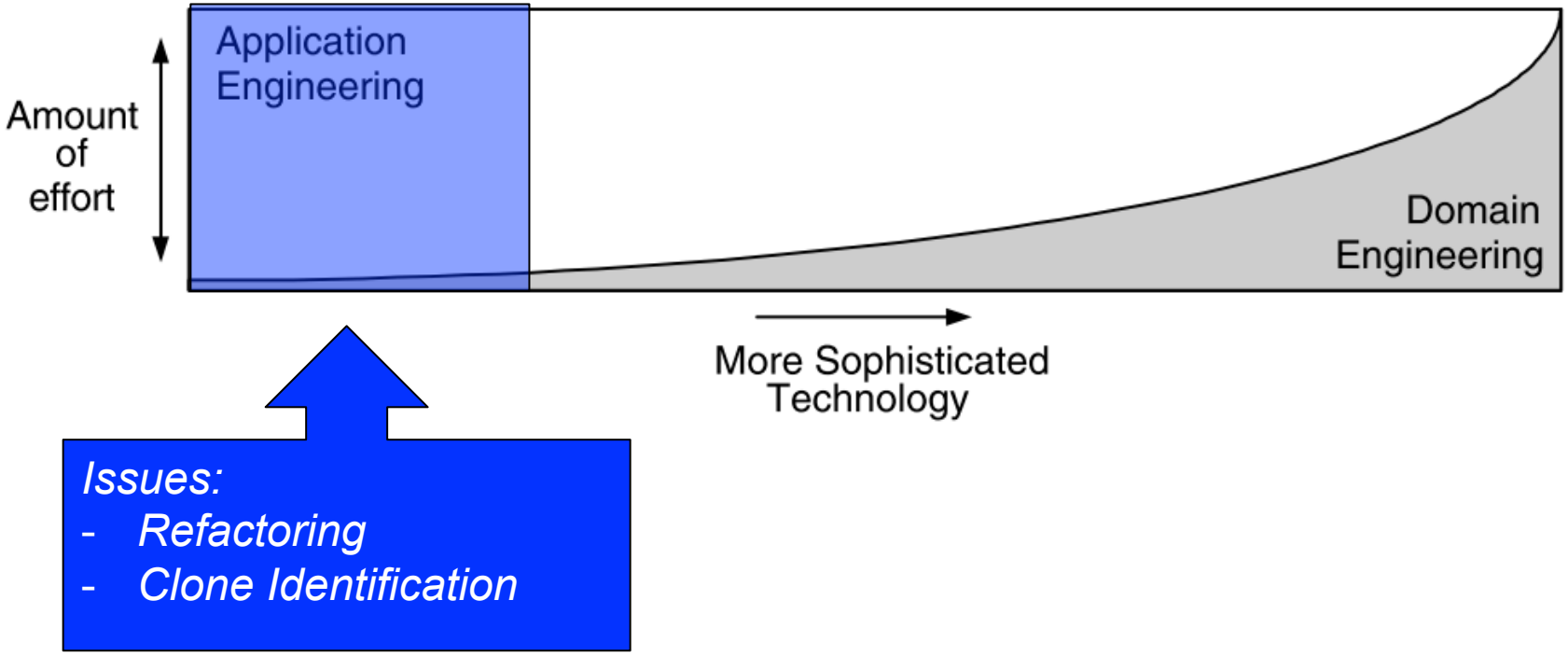

The long journey
towards
configurable SPL

The long journey towards configurable SPL

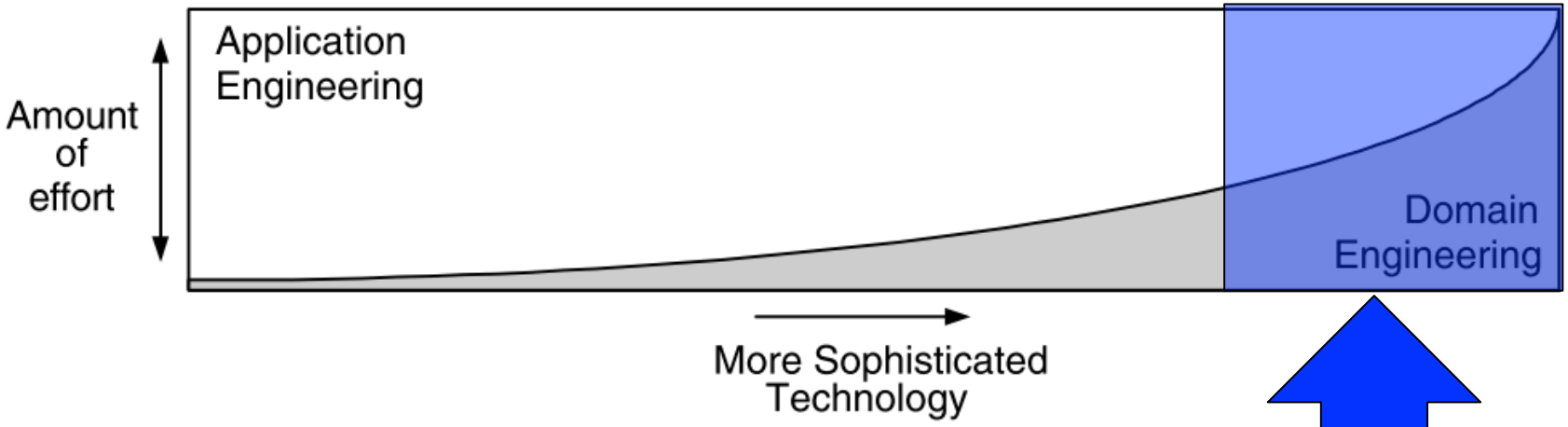


Sybren Deelstra, Marco Sinnema, Jan Bosch: Product derivation in software product families: a case study. Journal of Systems and Software. (2005)

The long journey towards configurable SPL

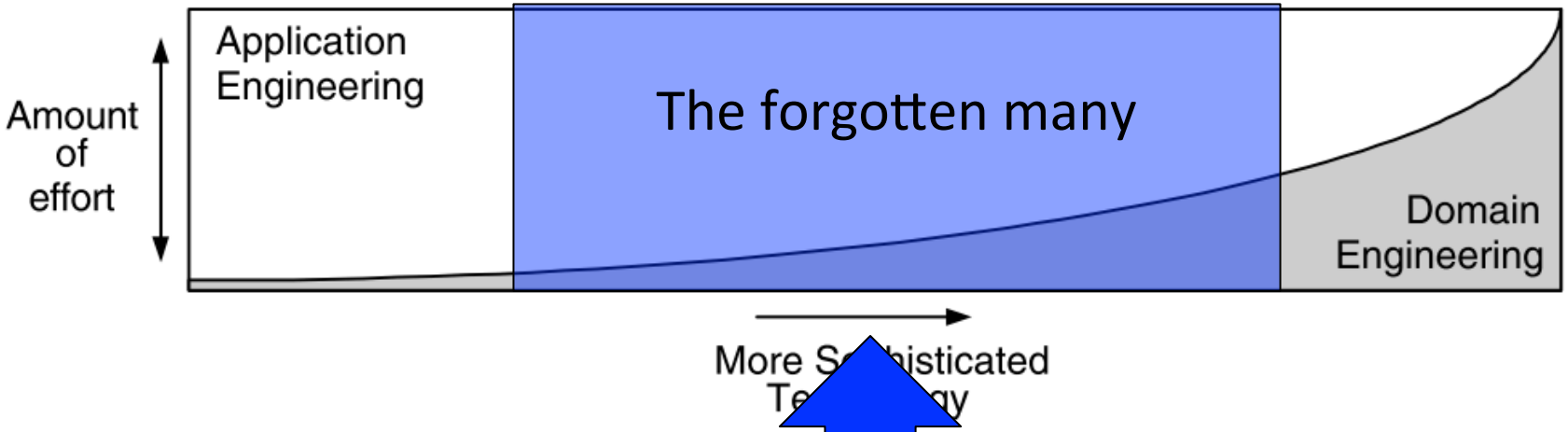


The long journey towards configurable SPL



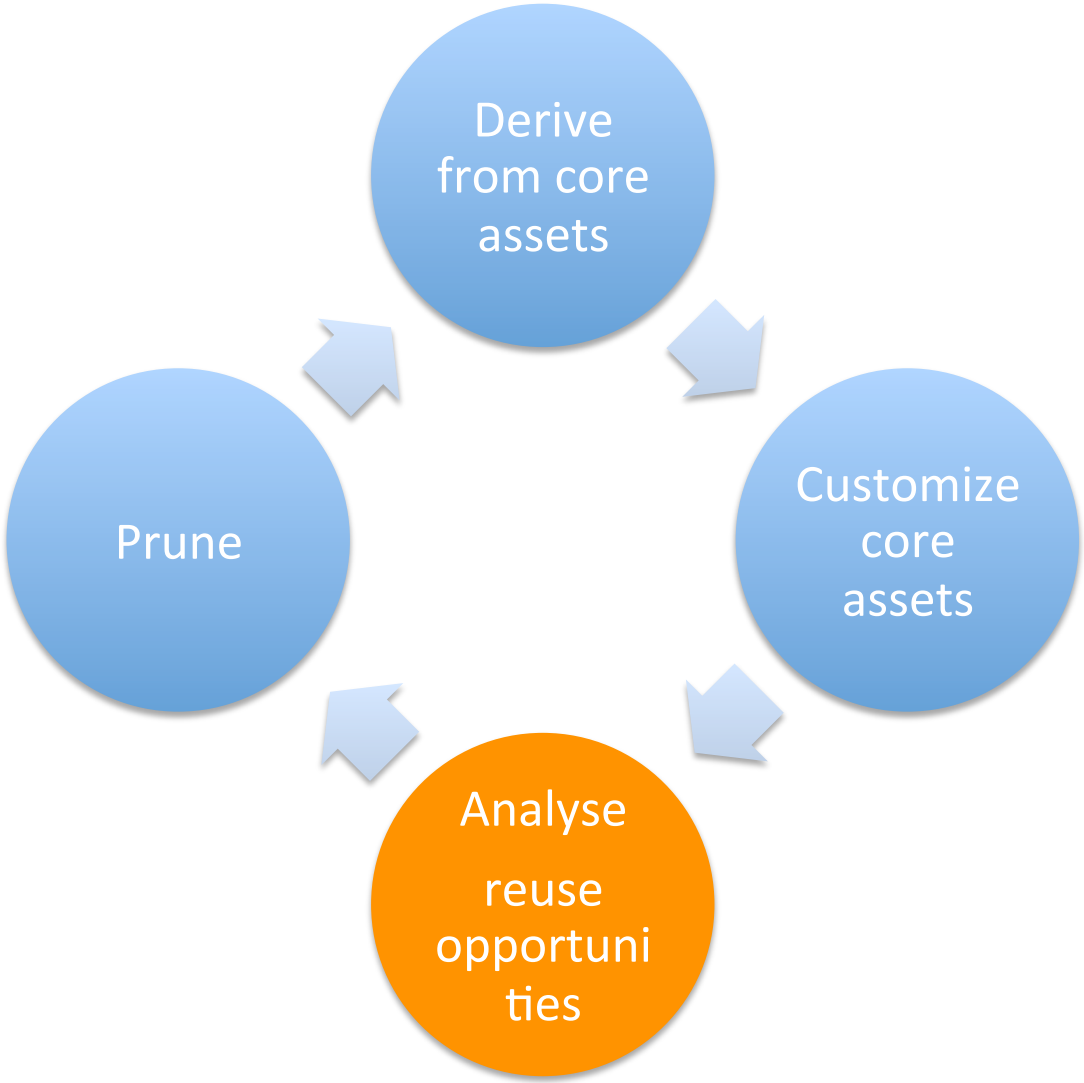
Issues:
- *Large feature models*
- *Configuration*

The long journey towards configurable SPL



Issues:
- *Evolution / incremental SPL development*
- Grow-and-prune process **[8, 18]**

Grow-and-prune process in product handling



SPL engineers might need to track

- Core assets
  - if they are being used as-is or rather, they had been customized to account for product specifics
- Product customizations
  - if ad-hoc product specifics might be amenable to reuse, and hence, be included into the next core-asset baseline release.

Visualizing product customization efforts for spotting SPL reuse opportunities

Leticia Montalvillo
University of the Basque Country
(UPV/EHU)
ONEKIN Research Group -
San Sebastián, Spain
leticia.montalvillo@ehu.eus

Oscar Díaz
University of the Basque Country
(UPV/EHU)
ONEKIN Research Group -
San Sebastián, Spain
oscar.diaz@ehu.eus

Maider Azanza
University of the Basque Country
(UPV/EHU)
ONEKIN Research Group -
San Sebastián, Spain
maider.azanza@ehu.eus

The diff utility

diff(core-asset, product -asset)

9 input/js/sensors.js

```
@@ -117,7 +117,14 @@ function setWarnings() {
117 117                 warningText += windWarning;
118 118                 if (warningText && pointer) {
119 119                         tempMeasure = warningText.value;
120     -                   var intValue = checkMeasure(min, max, tempMeasure);
    120 +                   if (measureText && pointer) {
    121 +                           tempMeasure = measureText.value;
    122 +                           var intValue = checkMeasure(min, max, tempMeasure);
    123 +                           if (isNaN(intValue)) return false;
    124 +                           intValue = (intValue - min)*(pxRange / (max - min));
    125 +
    126 +                           pointer.style.height = (177 - intValue) + "px";
    127 +                   }
121 128                   if (isNaN(intValue)) return false;
122 129                   intValue = (intValue - min)*(pxRange / (max - min));
123 130
```

Issue 1: the context

9 input/js/sensors.js

```
@@ -117,7 +117,14 @@ function setWarnings() {
117 117                 warningText += windWarning;
118 118                 if (warningText && pointer) {
119 119                         tempMeasure = warningText.value;
120     -                       var intValue = checkMeasure(min, max, tempMeasure);
    120 +                       if (measureText && pointer) {
    121 +                               tempMeasure = measureText.value;
    122 +                               var intValue = checkMeasure(min, max, tempMeasure);
    123 +                               if (isNaN(intValue)) return false;
    124 +                               intValue = (intValue - min)*(pxRange / (max - min));
    125 +
    126 +                               pointer.style.height = (177 - intValue) + "px";
    127 +                       }
121 128                         if (isNaN(intValue)) return false;
122 129                         intValue = (intValue - min)*(pxRange / (max - min));
123 130
```

Issue 2: the abstraction level

9 input/js/sensors.js

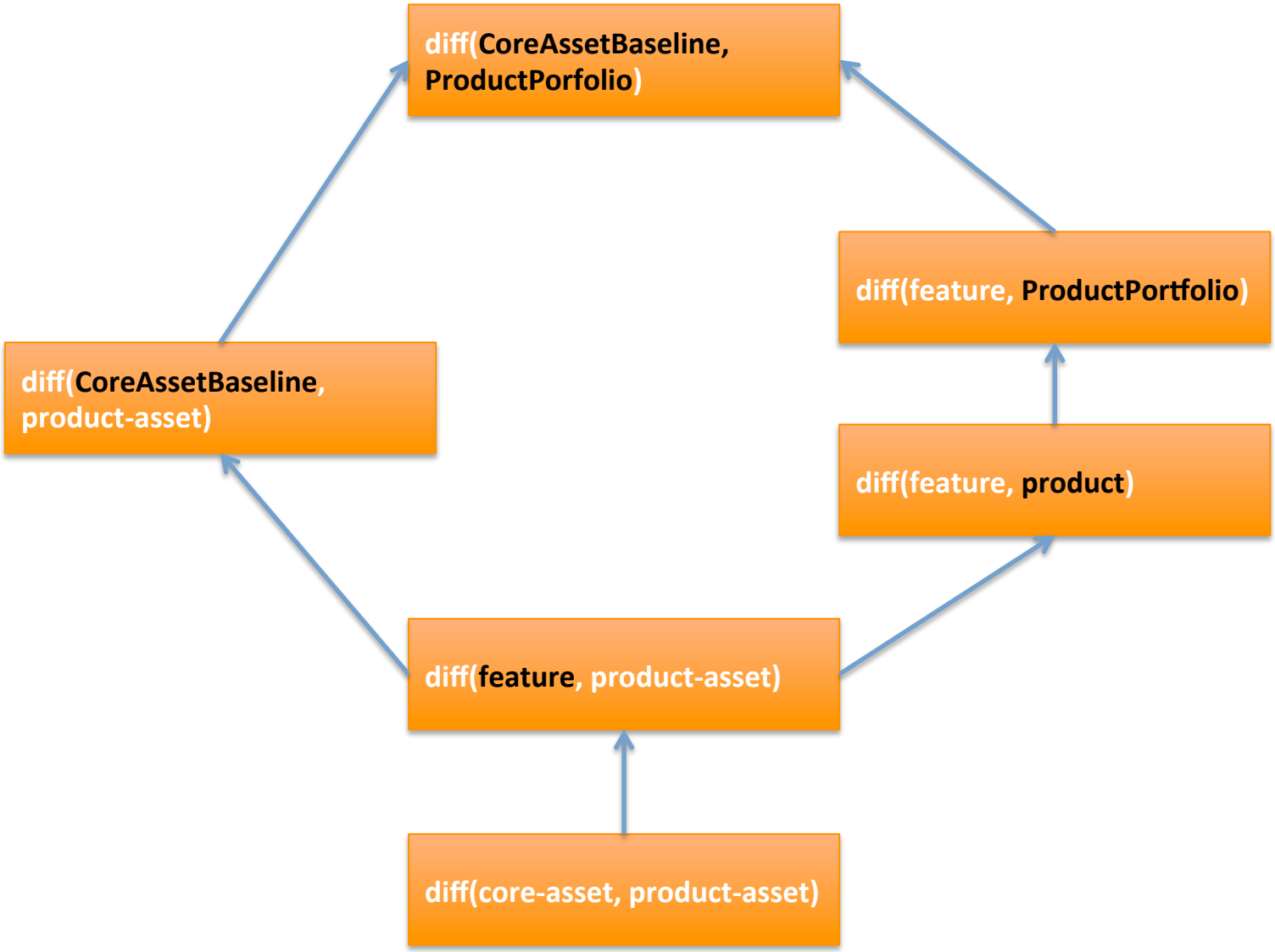
```
@@ -117,7 +117,14 @@ function setWarnings() {
117 117                 warningText += windWarning;
118 118                 if (warningText && pointer) {
119 119                         tempMeasure = warningText.value;
120     -                       var intValue = checkMeasure(min, max, tempMeasure);
    120 +                       if (measureText && pointer) {
    121 +                               tempMeasure = measureText.value;
    122 +                               var intValue = checkMeasure(min, max, tempMeasure);
    123 +                               if (isNaN(intValue)) return false;
    124 +                               intValue = (intValue - min)*(pxRange / (max - min));
    125 +
    126 +                               pointer.style.height = (177 - intValue) + "px";
    127 +                       }
121 128                         if (isNaN(intValue)) return false;
122 129                         intValue = (intValue - min)*(pxRange / (max - min));
123 130
```

Issue 2: the abstraction level

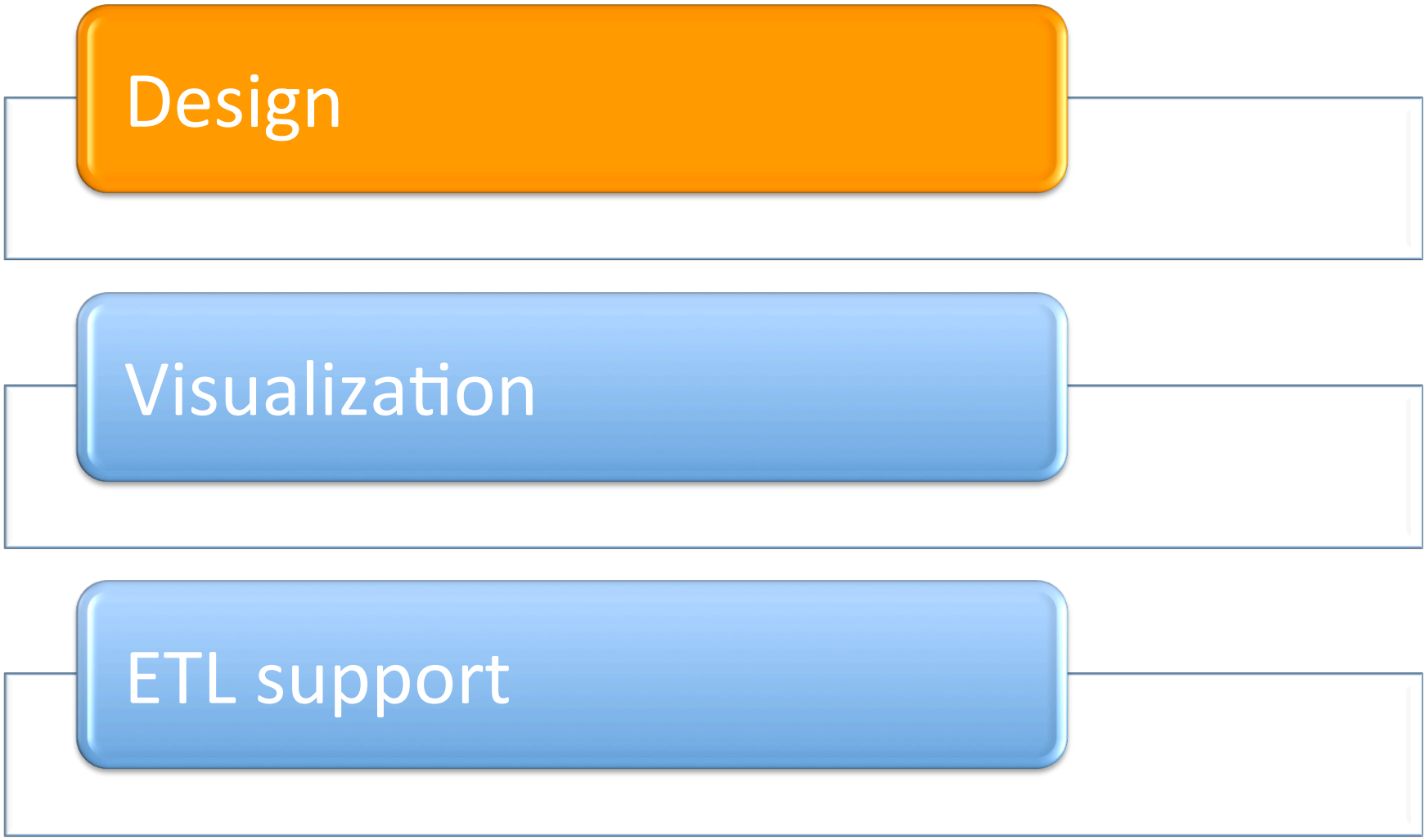
9 input/js/sensors.js

```
@@ -117,7 +117,14 @@ function setWarnings() {
117 117                 warningText += windWarning;
118 118                 if (warningText && pointer) {
119 119                         tempMeasure = warningText.value;
120     -                   var intValue = checkMeasure(min, max, tempMeasure);
    120 +                   if (measureText && pointer) {
    121 +                           tempMeasure = measureText.value;
    122 +                           var intValue = checkMeasure(min, max, tempMeasure);
    123 +                           if (isNaN(intValue)) return false;
    124 +                           intValue = (intValue - min)*(pxRange / (max - min));
    125 +
    126 +                           pointer.style.height = (177 - intValue) + "px";
    127 +                   }
121 128                         if (isNaN(intValue)) return false;
122 129                         intValue = (intValue - min)*(pxRange / (max - min));
123 130
```

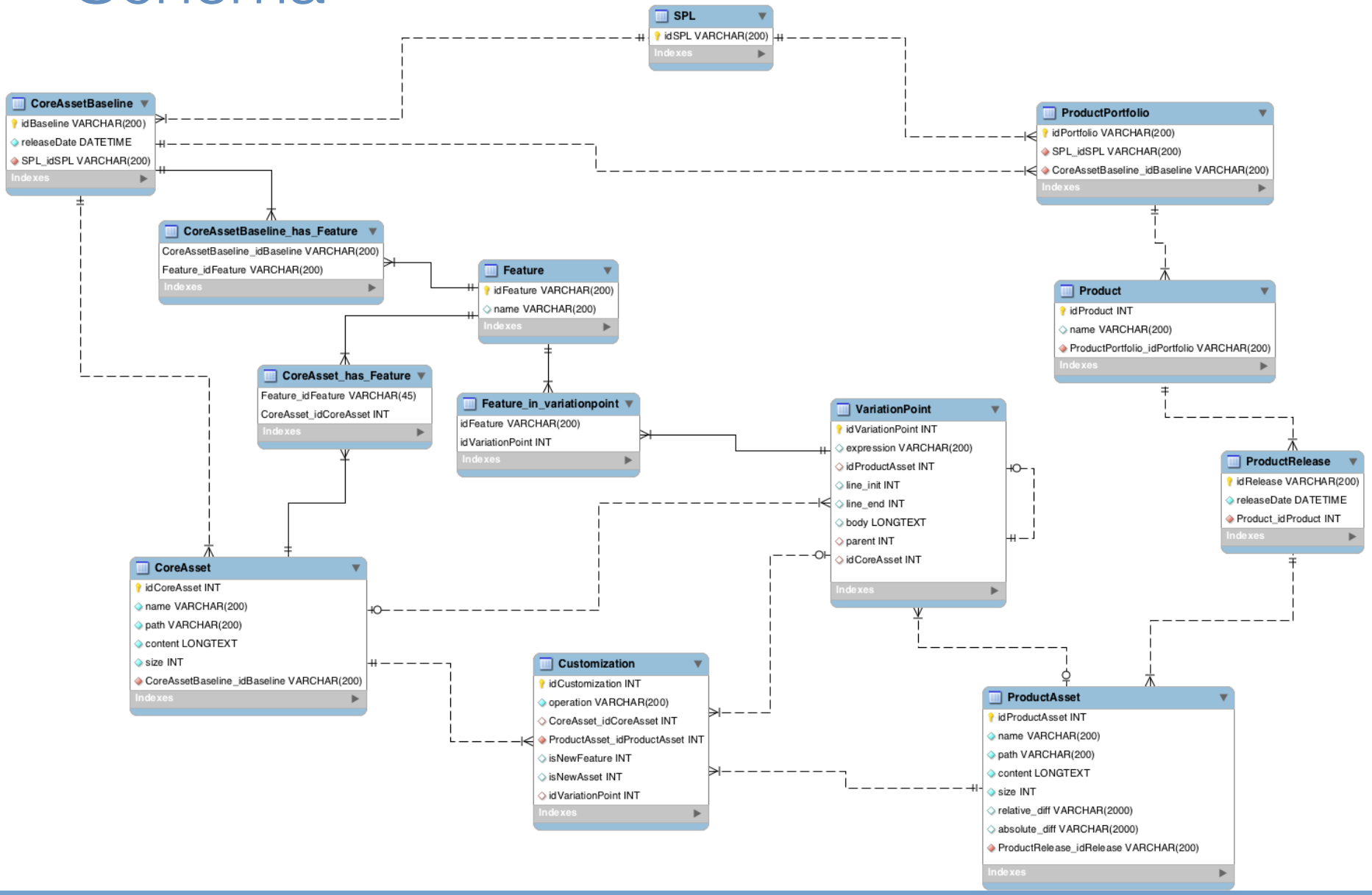
From diff(core-asset, product-asset)…




Design
Visualization
ETL support


Design
Visualization
ETL support

Schema


SPL
idSPL VARCHAR(200)
Indexes
CoreAssetBaseline
idBaseline VARCHAR(200)
releaseDate DATETIME
SPL_idSPL VARCHAR(200)
Indexes
ProductPortfolio
idPortfolio VARCHAR(200)
SPL_idSPL VARCHAR(200)
CoreAssetBaseline_idBaseline VARCHAR(200)
Indexes
CoreAssetBaseline_has_Feature
CoreAssetBaseline_idBaseline VARCHAR(200)
Feature_idFeature VARCHAR(200)
Indexes
Feature
idFeature VARCHAR(200)
name VARCHAR(200)
Indexes
Product
idProduct INT
name VARCHAR(200)
ProductPortfolio_idPortfolio VARCHAR(200)
Indexes
CoreAsset_has_Feature
Feature_idFeature VARCHAR(45)
CoreAsset_idCoreAsset INT
Indexes
Feature_in_variationpoint
idFeature VARCHAR(200)
idVariationPoint INT
Indexes
VariationPoint
idVariationPoint INT
expression VARCHAR(200)
idProductAsset INT
line_init INT
line_end INT
body LONGTEXT
parent INT
idCoreAsset INT
Indexes
ProductRelease
idRelease VARCHAR(200)
releaseDate DATETIME
Product_idProduct INT
Indexes
CoreAsset
idCoreAsset INT
name VARCHAR(200)
path VARCHAR(200)
content LONGTEXT
size INT
CoreAssetBaseline_idBaseline VARCHAR(200)
Indexes
Customization
idCustomization INT
operation VARCHAR(200)
CoreAsset_idCoreAsset INT
ProductAsset_idProductAsset INT
isNewFeature INT
isNewAsset INT
idVariationPoint INT
Indexes
ProductAsset
idProductAsset INT
name VARCHAR(200)
path VARCHAR(200)
content LONGTEXT
size INT
relative_diff VARCHAR(2000)
absolute_diff VARCHAR(2000)
ProductRelease_idRelease VARCHAR(200)
Indexes

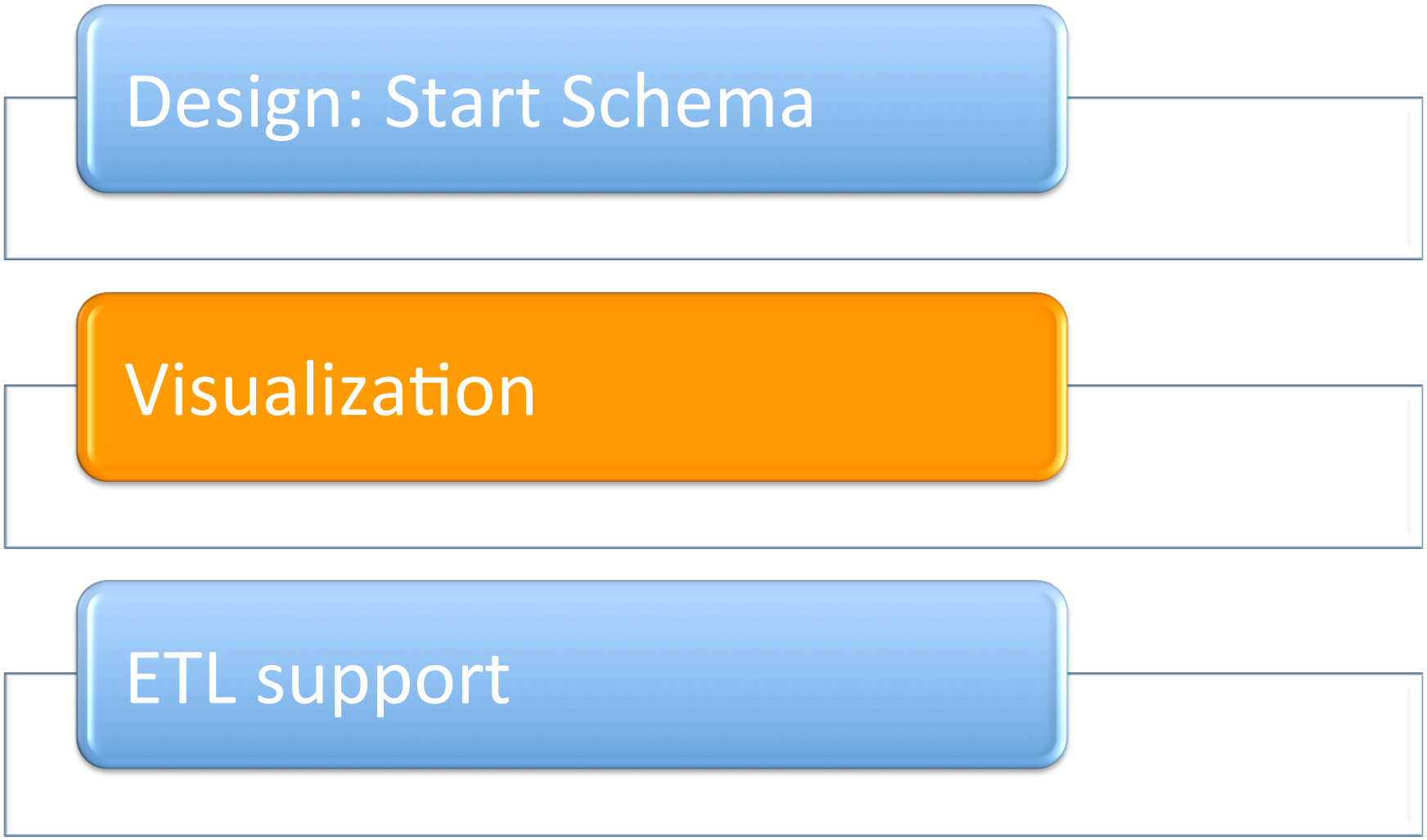
Demo: Customization analysis for pure::variants WeatherStation SPL

Customization

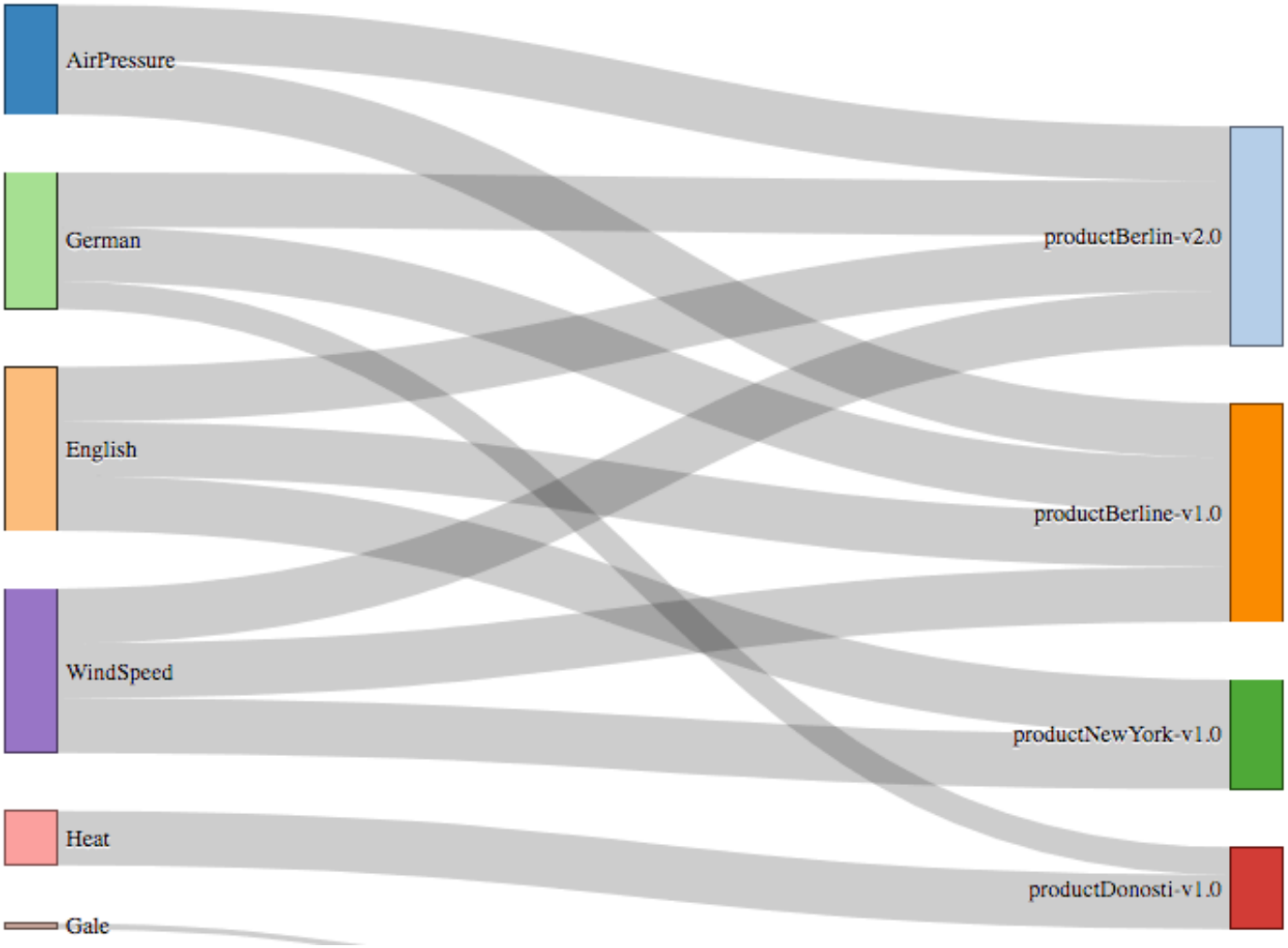
53 input/js/sensors.js

```
@@ -14,7 +14,10 @@ function applyWindSpeed() {
14 14         var measureText = document.getElementById("w_measure");
15 15         windMeasure = measureText.value;
16 16         var pointer = document.getElementById("w_point");
   17 +       var measureText = document.getElementById("p_measure");
   18 +       var pointer = document.getElementById("p_point");
17 19
   20 +       applyTachoValue(minPres, maxPres, measureText, pointer);
18 21         applyTachoValue(minWind, maxWind, measureText, pointer);
19 22         setWarnings();
20 23         return false;
@@ -25,12 +28,25 @@ function applyWindSpeed() {
25 28  function applyTachoValue(min, max, measureText, pointer) {
26 29         var divisor = Math.round((max - min)/13);
27 30         var c = Math.round(divisor/2);
28    -
   31 +       var pointer = document.getElementById("w_point");
   32 +       var measureText = document.getElementById("p_measure");
   33 +       var pointer = document.getElementById("p_point");
29 34         if (measureText && pointer) {
30 35                 var measure = measureText.value;
31 36                 var intValue = checkMeasure(min, max, measure);
32 37                 if (isNaN(intValue)) return false;
```

Dimension Analysis Highlights

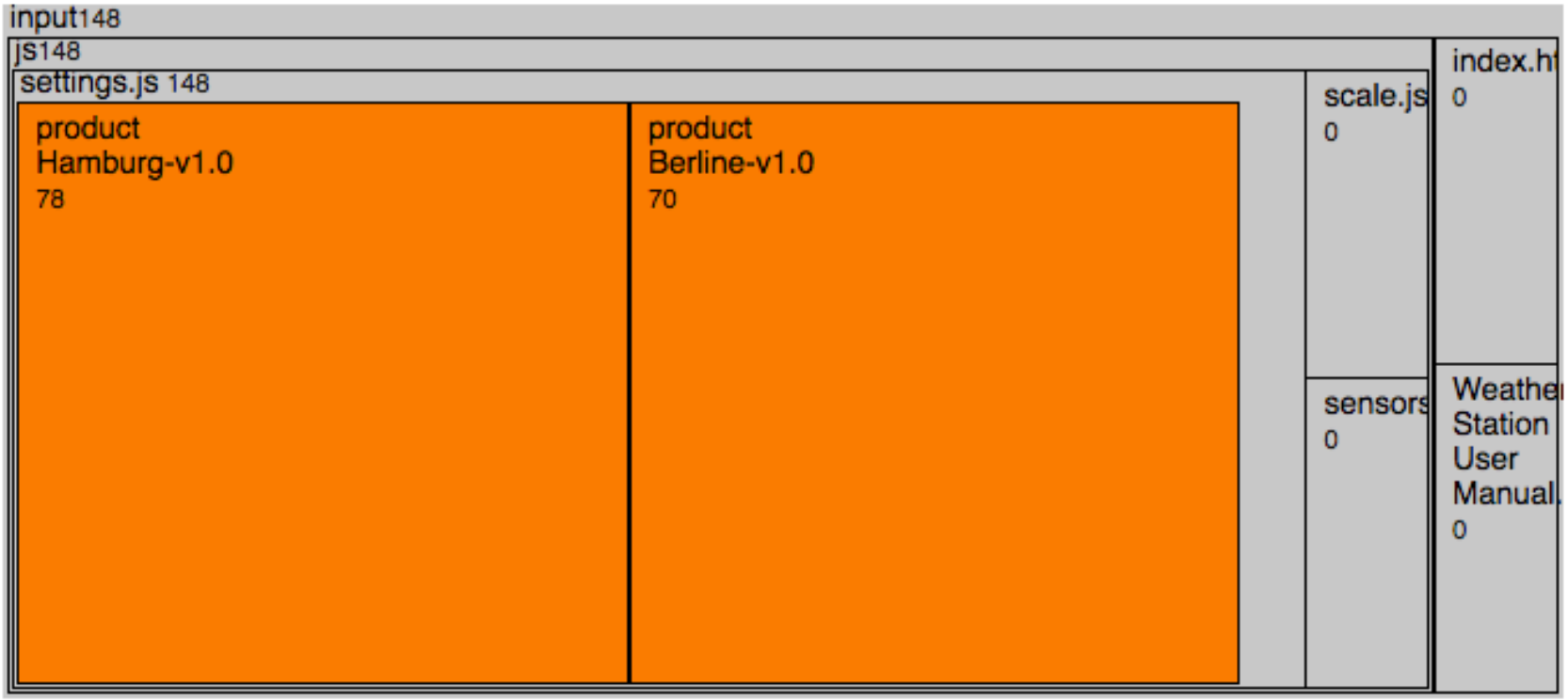


diff(CoreAssetBaseline, ProductPorfolio)

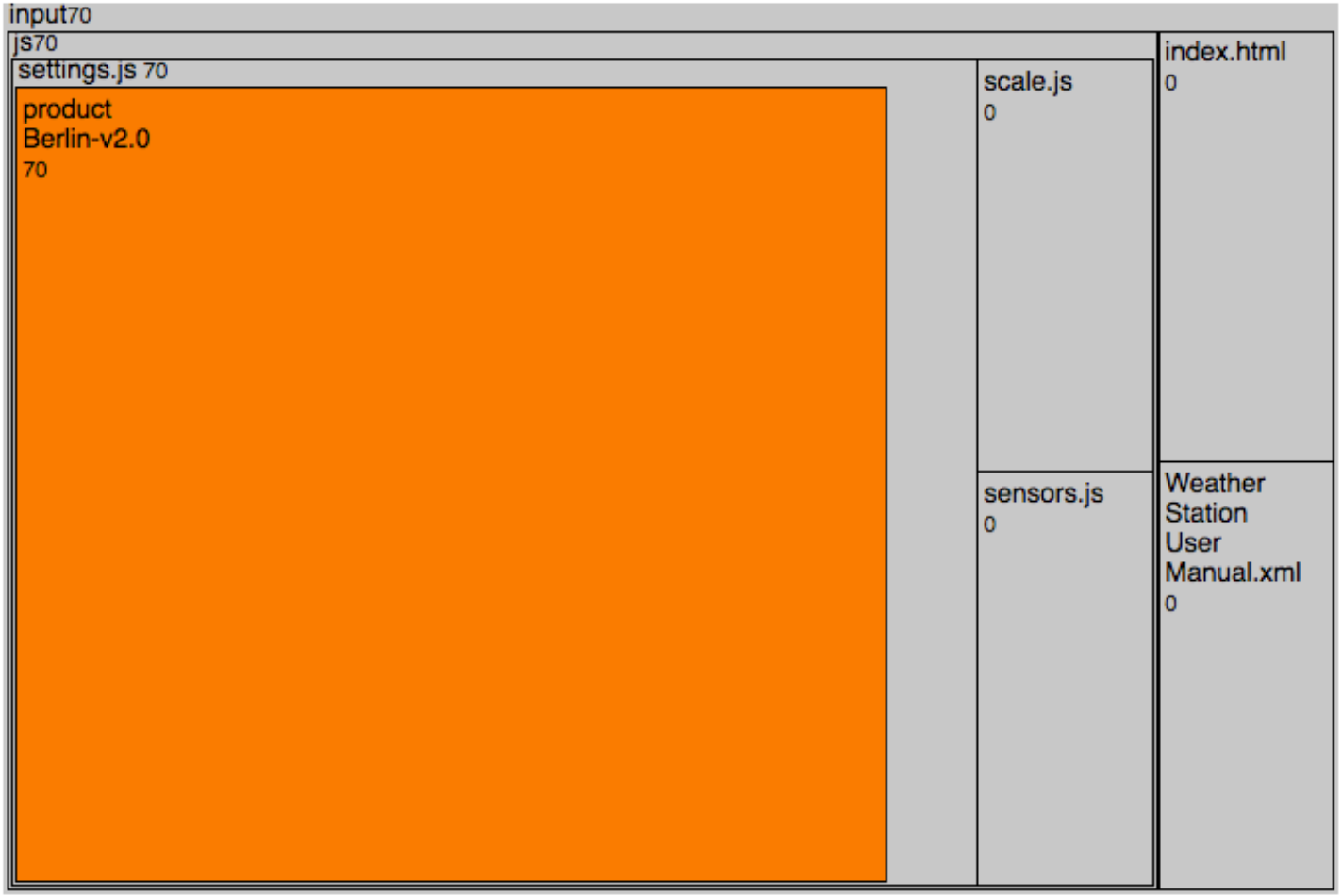


diff(feature, ProductPortfolio)

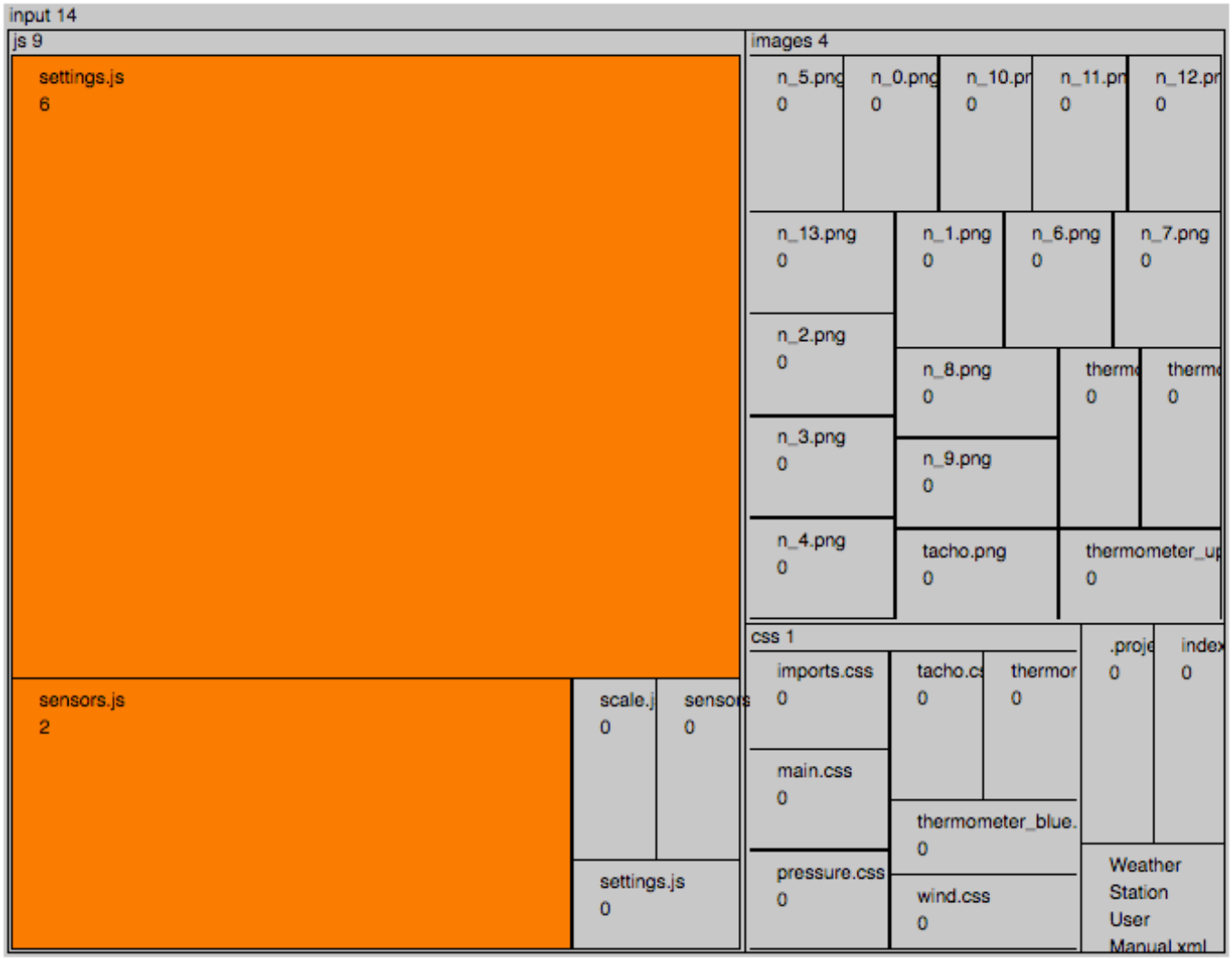
Note: Treemap area denotes customization change-size.



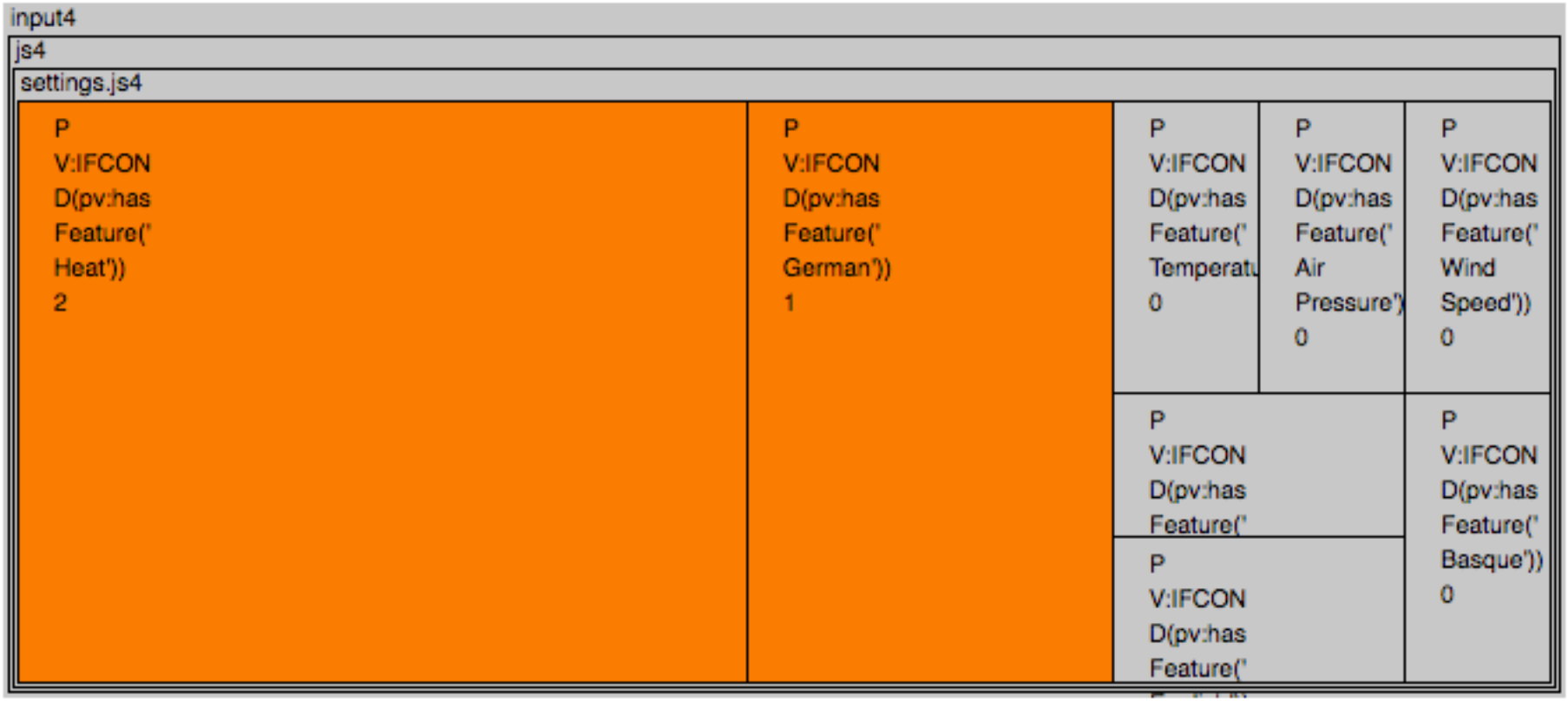
diff(feature, Product)

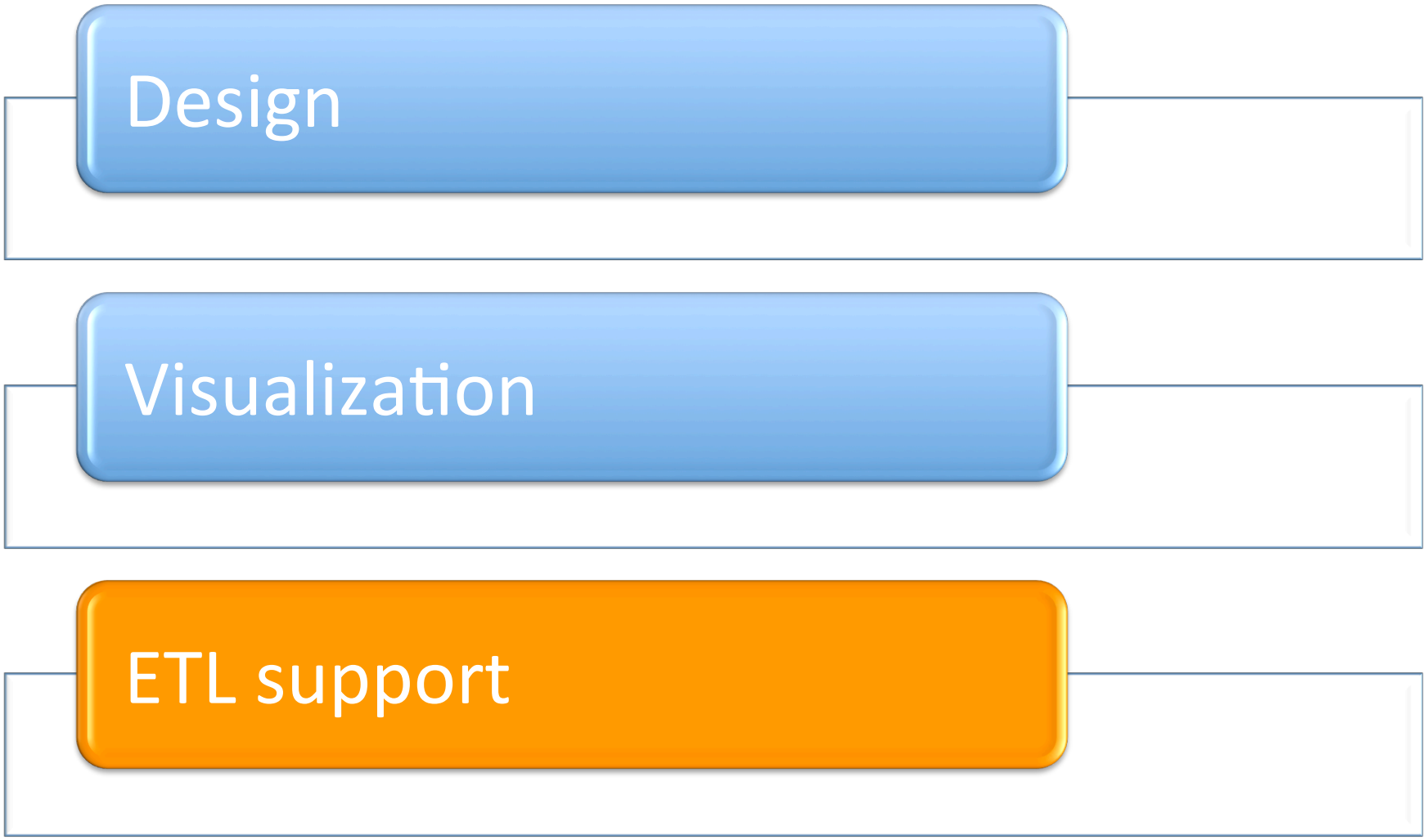


diff(CoreAssetBaseline, Product)

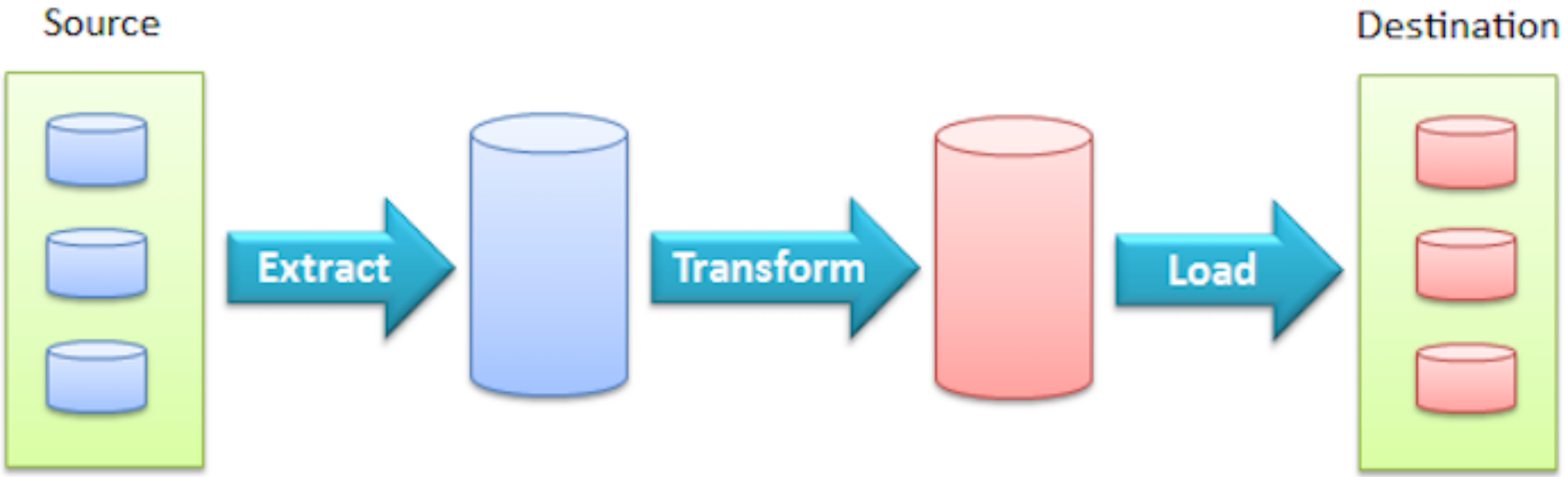


diff(feature, product asset)




Design
Visualization
ETL support

ETL Process



Demo: Customization analysis for pure::variants WeatherStation SPL

Thanks for your attention!

@ {oscar.diaz,leticia.montalvillo}
@ehu.eus