Server Driven UI @ Zalando

14/03/2024 - David Jones

Tech at Zalando

The tech community at Zalando is made up of almost 2000 people with the web frontend receiving ~5000 page loads per second.

The engineering community working on the Zalando Fashion Store has spent the last couple of years heavily focused on the building and migration of a new architecture called Interface Framework which is now considered stable.

How is the Zalando Web Store built?

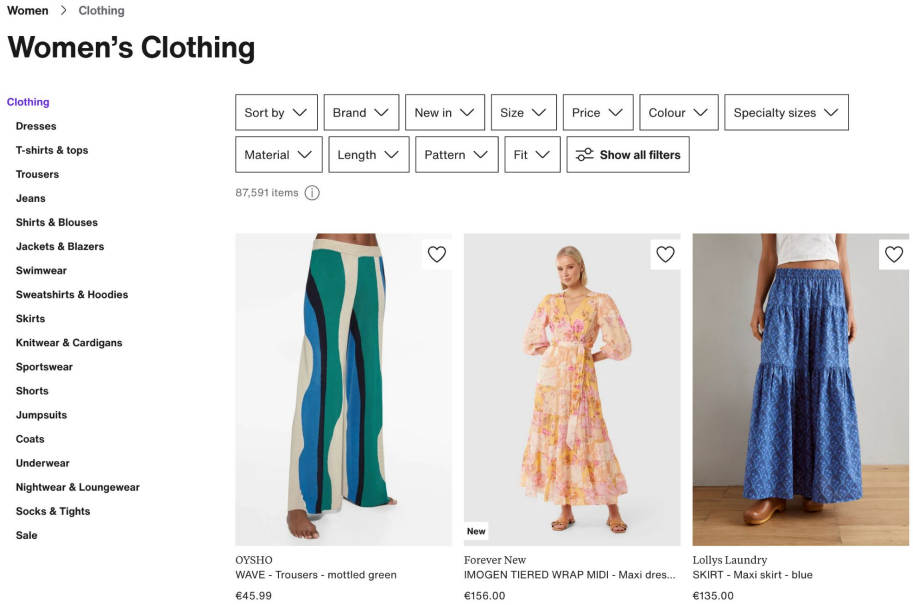
Using the Interface Framework enables the creation of customer experiences for both web and app and uses an entity-based data model.

Skipper: Edge-routing system used to send traffic to the correct systems

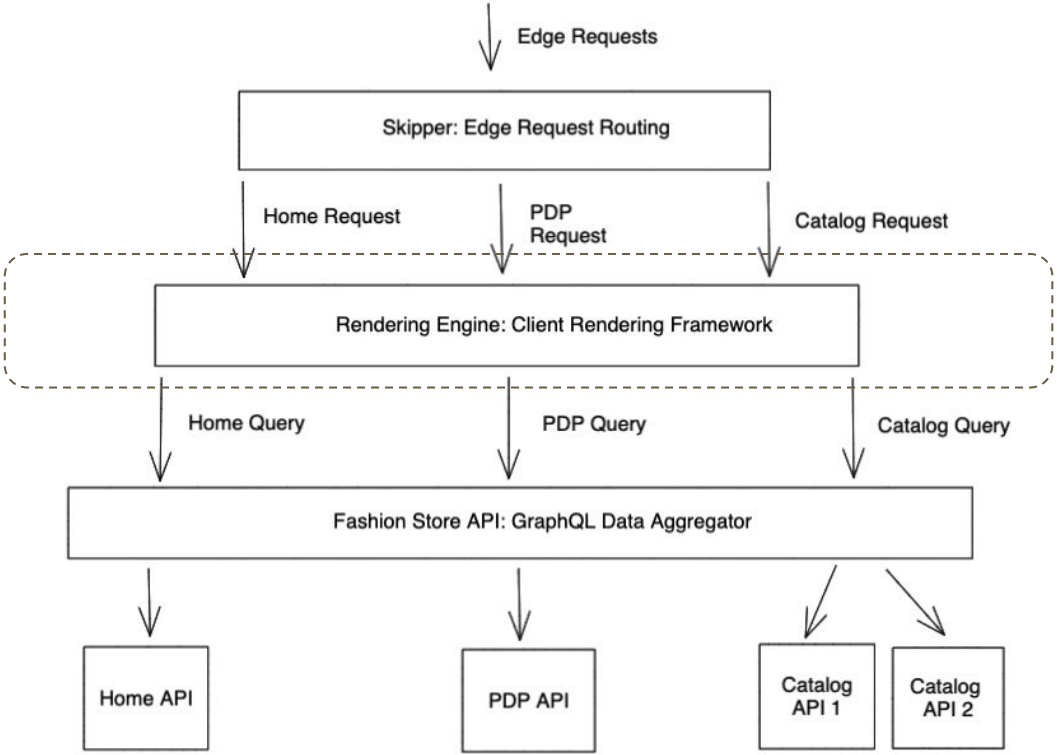
Rendering Engine & Apps: Systems used to render content and fetch the necessary entity data

Fashion Store API: API data aggregation using GraphQL

Relevant Entities System: API which provides clients with additional content based on the request context.



Requesting data for a page



Display logic implemented here

Motivations for Server-Driven UI

Reduced need for App Releases: By enabling the changing of layouts from the server avoids the need to make changes in clients. This is especially useful for Apps which require a full release process and submission to the app store.

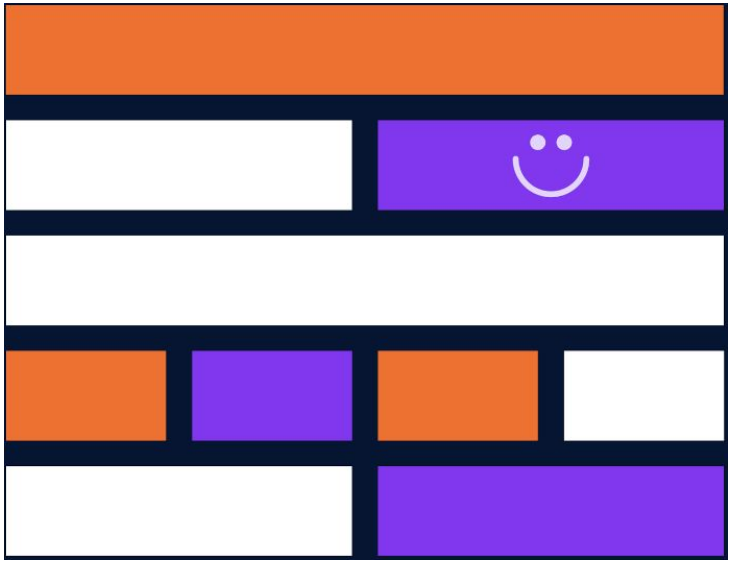
Faster Experimentation: Stakeholders in the organisation are enabled to test different customer flows without the need for engineering effort.

Personalisation: Customers may get different layouts depending on their browsing habits or preferences they have defined in their user settings.

What are Entities?

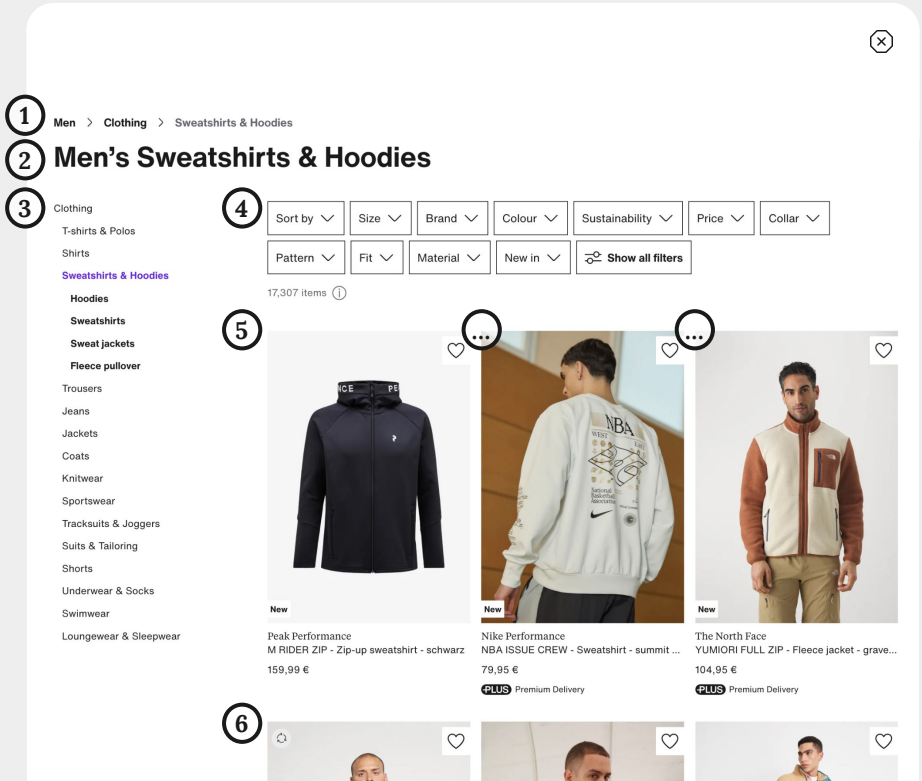
Entities are the basic units of data for building the user experience. They can represent content shown to customers such as products or entity groups based on defined relationships such as collections.

Entities provide both a common language in the organisation for discussing data, as well as anchor points for the different content types on the platform.



A page as a layout tree

- Vertical
  - 1 Breadcrumbs
  - 2 Headline
  - Horizontal
    - 3 Category Tree
    - Horizontal with wrapping
      - 4 Filters
      - 5 Product Card
      - ...
      - 6 Product Card



An example Layout

We currently build our layouts using a generator which takes a JSX layout definition and converts it to JSON for usage by APIs and all clients (both web and apps).

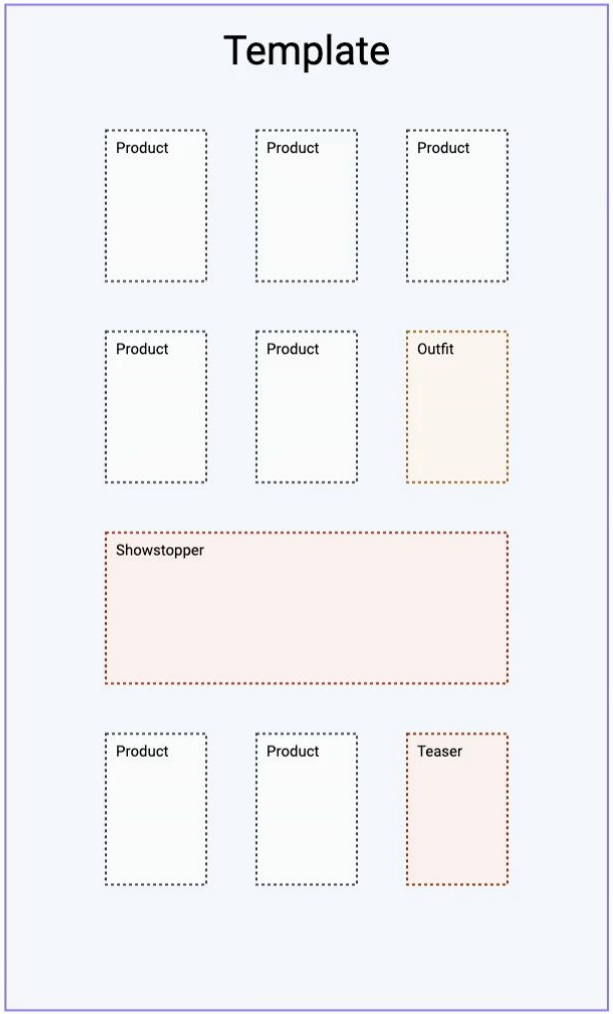

```
{
  "templateId": "342be8c8-aa80-41ed-b608-5137d88f0a9f",
  "root": {
    "_type": "StaticPlaceholder",
    "id": "ern:slottedlayout::horizontal-layout",
    "additionalProperties": {
      "__typename": "SlottedLayout",
      "module": "LAYOUT_HORIZONTAL"
    },
    "slots": [
      {
        "additionalProperties": {
          "__typename": "HorizontalLayoutSlot",
          "span": {
            "0": 6,
            "480": 4
          }
        },
        "fallbackStrategy": {
          "_type": "InterruptAllocation"
        },
        "entity": {
          "_type": "ProductPlaceholder",
          "additionalProperties": {
            "module": "PRODUCT_CARD",
            "visuals": {
              "hints": [
                "WITH_HOVER"
              ]
            }
          }
        }
      },
      ……and so on
```

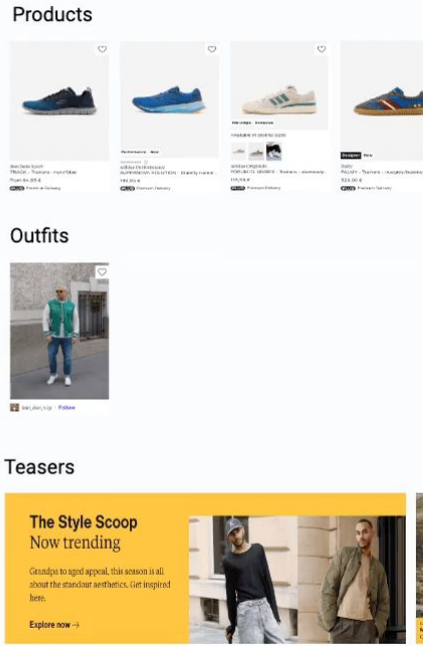
Template Definition

A template contains slots for different types of content. This content is selected by the Relevant Entities System and inserted into the layout.

The template language has been designed to capture business requirements in a premise agnostic way.

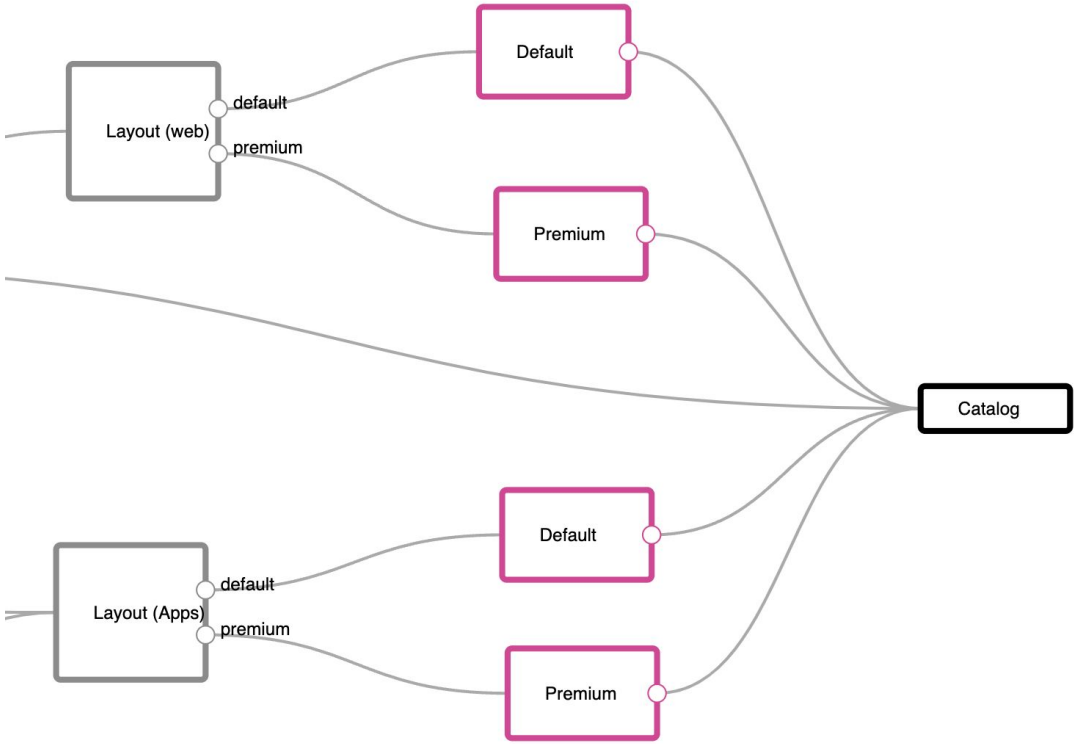
It is also planned to provide templates using ML models in the future.





Layout Selection

- Page Context: What type of page is the customer viewing and how have they gotten there?
- Customer: What type of customer are they and what are they interested in?
- A/B tests: What test variant bucket has the customer been selected for?



Challenges

- Improving monitoring to include layout information
- Adding the necessary guardrails to prevent the serving of broken layouts
- Unification of Client Platform Capabilities

Demo Time

Changing the page layout

Questions?

Thank You!

Want to help us build this? We are hiring!

Head of Engineering (hybrid) Software Engineering - Leadership	Dublin	**View job** →
Scala/Java Engineer- Customer Platform (hybrid) Software Engineering - Backend	Dublin	**View job** →
Senior Frontend Engineer (hybrid) - Search and Browse Experience Software Engineering - Frontend	Dublin	**View job** →
Senior Scala/Java Engineer (hybrid) - Global Catalog Software Engineering - Backend	Dublin	**View job** →

Apply @ jobs.zalando.com