

---

# Scalable Frontends @ Zalando

— 20/10/2022 - David Jones —

---

# Tech at Zalando

The tech community at Zalando is made up of almost 3000 people with the web frontend receiving almost 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.

# What is Interface Framework?

The Interface Framework is a system that enables the creation of customer experiences for both web and app. It is based around an entity-focused structure and is made up of the below components.

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

**Rendering Engine:** System used to render frontend content and fetch the necessary entity data

**Fashion Store API:** API data aggregation using GraphQL

**Zalando Design System:** UI Component Library

# Why Interface Framework?

**Consistent Entity Data:** Ensure all platforms use data (entities) in a consistent way.

**Relevant Content Selection:** Provide personalisation and steering as part of a unified response.

**Consistent UX:** A global design language which consists of a library of shared components.

**Community Focus:** Enables community driven work, shared best practices and knowledge sharing.

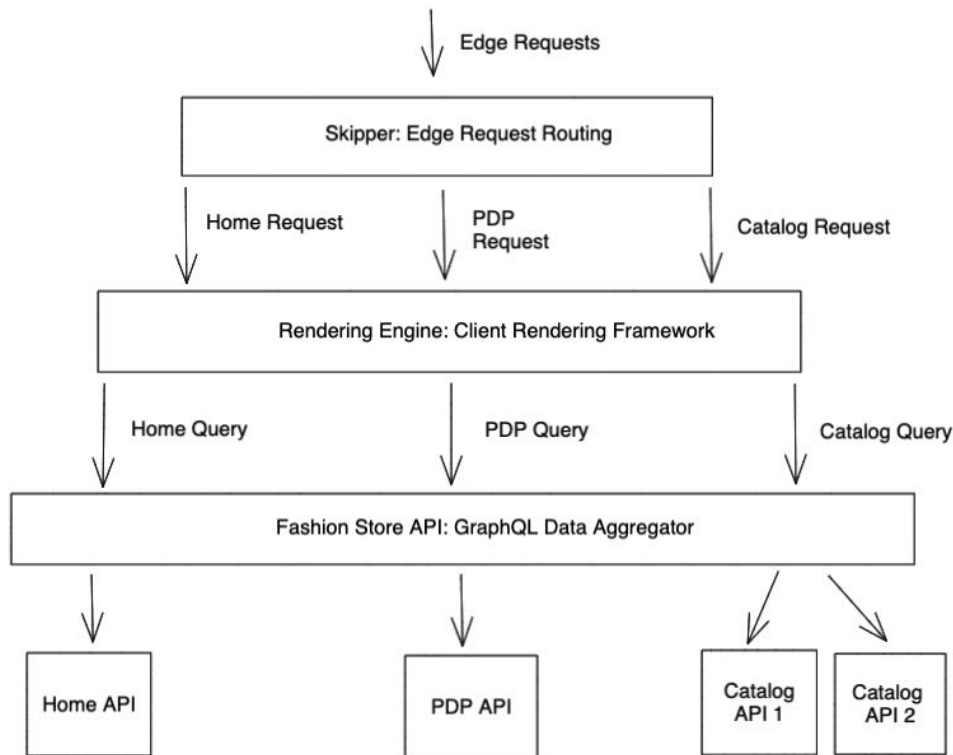
# What are Entities?

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 (the nouns) in the organisation for discussing data, as well as anchor points for the different content types on the platform.

# Request Flow

The request flow of a user accessing the Zalando Fashion Store.



# Rendering Engine

On web, the Rendering Engine hosts self-contained units of code called **Renderers**.

Renderers are used to provide a visual representation of entities. Each renderer is built using Zalando Design System components and have their own queries for fetching data based on their own entity type and ID.

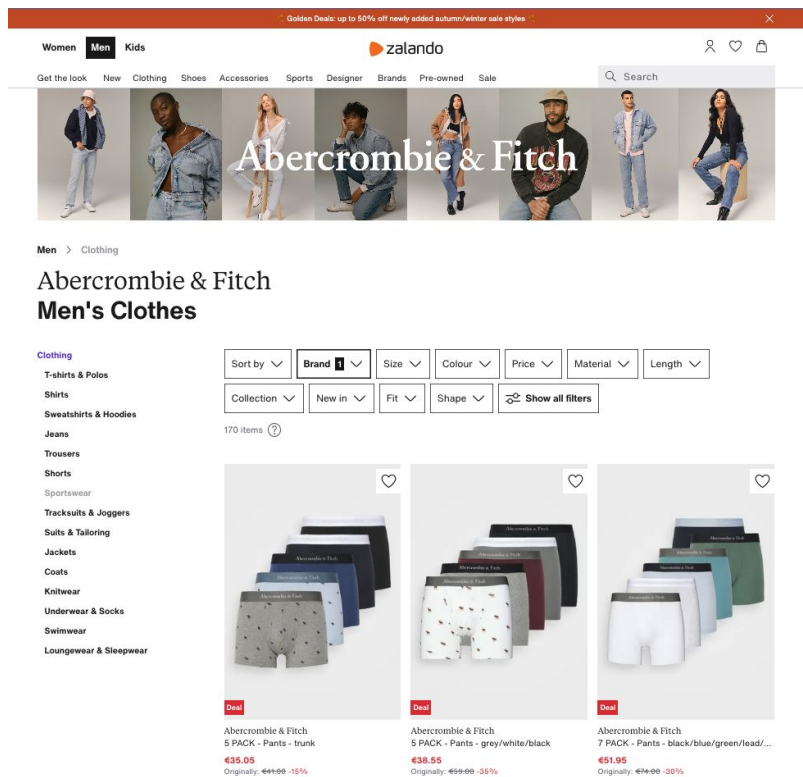
Product Renderer



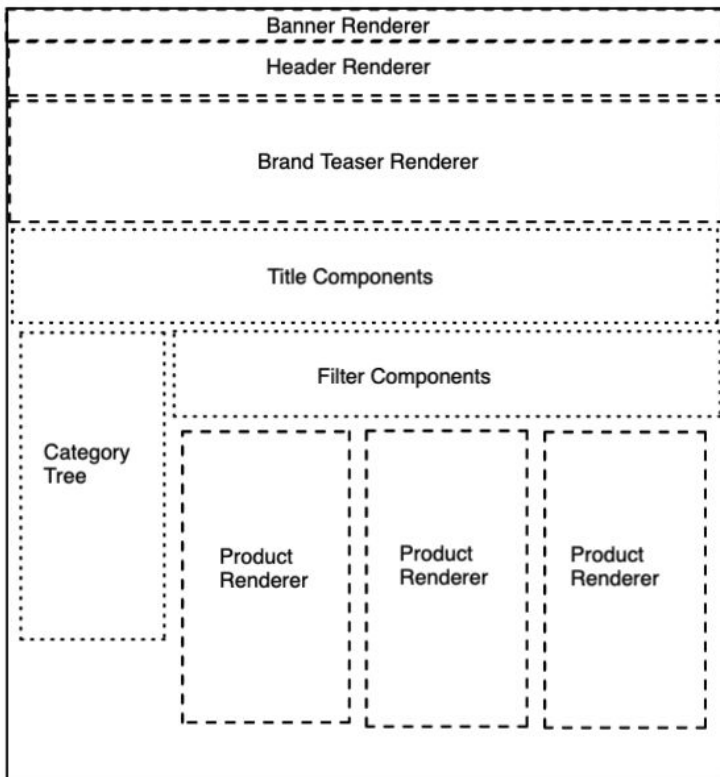
Consists of:

- ZDS Components
- GraphQL Queries

# Page built with renderers



## Catalog Renderer





# Fashion Store API - GraphQL

The Fashion Store API is the API aggregation layer which provides a consistent schema and allows access to entity information for all pages.

```
{
  product(id: "ern:product::THC21U008-C11") {
    brand {
      name
    }
    color {
      label
    }
    displayPrice {
      original {
        formatted
      }
    }
  }
  name
  primaryImage {
    ... on Image {
      uri
    }
  }
}
}
```

# Zalando Design System

The Zalando Design System provides the building blocks of the Zalando user experience. It includes everything from CSS classes and buttons to more complex components such as carousels and product cards.

## Code examples

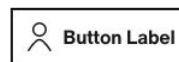
[Css](#)

### Background colors

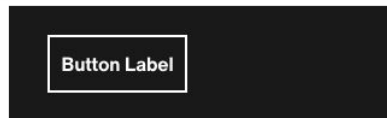
Hello world



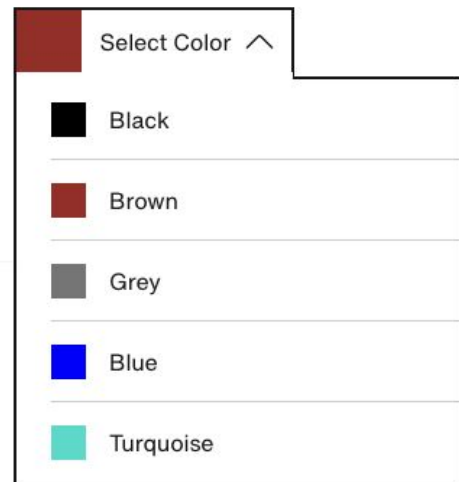
### Button secondary



### Button secondary inverted



## Color list



# What makes it scalable?

- The codebase accommodates an inner-sourcing model, allowing teams to contribute to all domains and support each other with best practices.
- Premise teams focus on contributing new user experiences for our customers while platform teams maintain the systems and operate them.
- Optimising flows in the architecture and providing additional functionality allows any of the contributing teams to make use of it.

# Challenges

- Trade-off between quality and fast deployment to production
- Ownership of shared renderers is sometimes blurry
  - Changes in shared renderers can be hard to track
- Maintaining a great developer experience used by dozens of teams

# Demo Time

Displaying entities

**Questions?**

**Thank You!**