e-commerce from **A** to **Z**

Xebia and *The Sting*

Background of the project

- Running legacy Hybris (*old old version*)
- Maintained by integrator which employs **30** people on the project
- Road to production > **30 days**
- **1 store** + copy pasta === **another store**
- Without being able to handle trivial load

Scope of the project

- Multi brand from a single codebase
- Multi language
- Product discovery (core shopping)
- Content management (promotional content)
- Ordering and payment (checkout)
- Integrations with external systems

Team

Consists of +- **8** developers, where half are **frontend** and the other half are **backend** developers

Xebia consultants

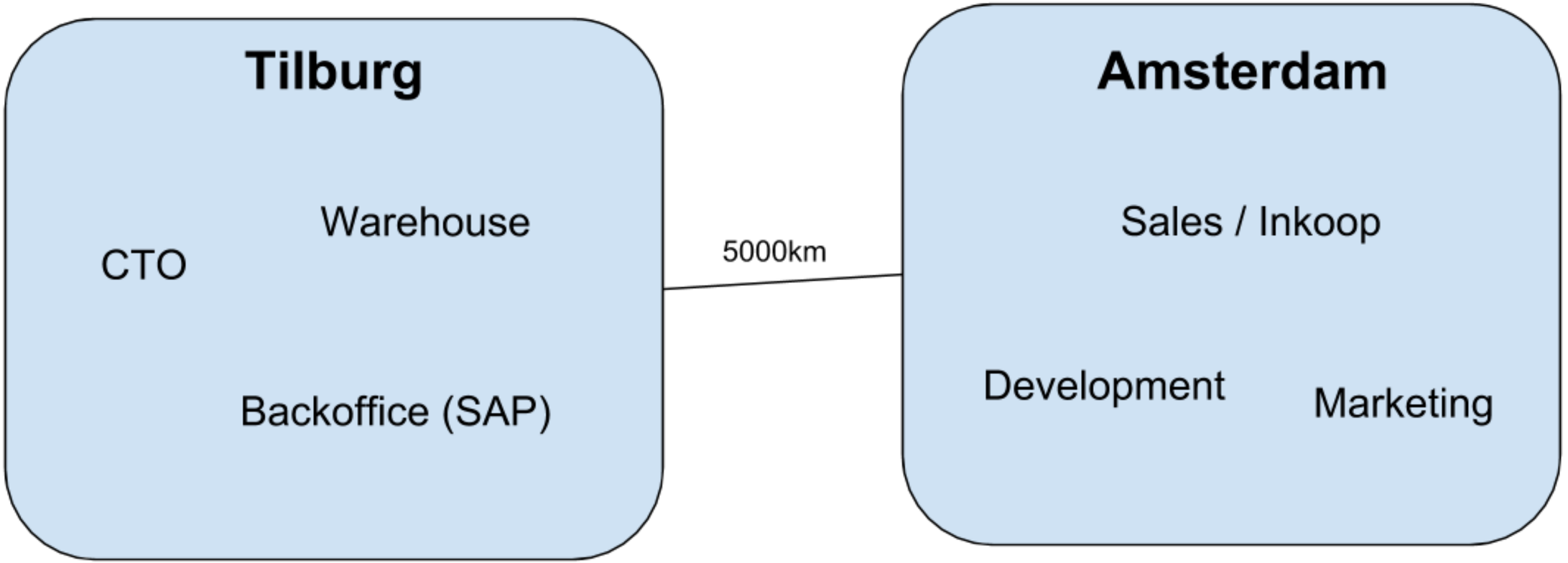
Gideon de Kok (back-end), **Ruben Oostinga** (front-end), **Mike Woudenberg** (front-end?), **Jesse van Bekkum** (back-end), **Norbert de Lange** (front-end), **Anton Lijcklama** (back-end), **Cengiz Ulusoy** (front-end / UX), **Khrystyna Svarok** (front-end), **Michael Meiers** (product owner), **Frank Visser** (back-end), **Joachim Bargsten** (back-end)[1]

[1] missing someone? 😅

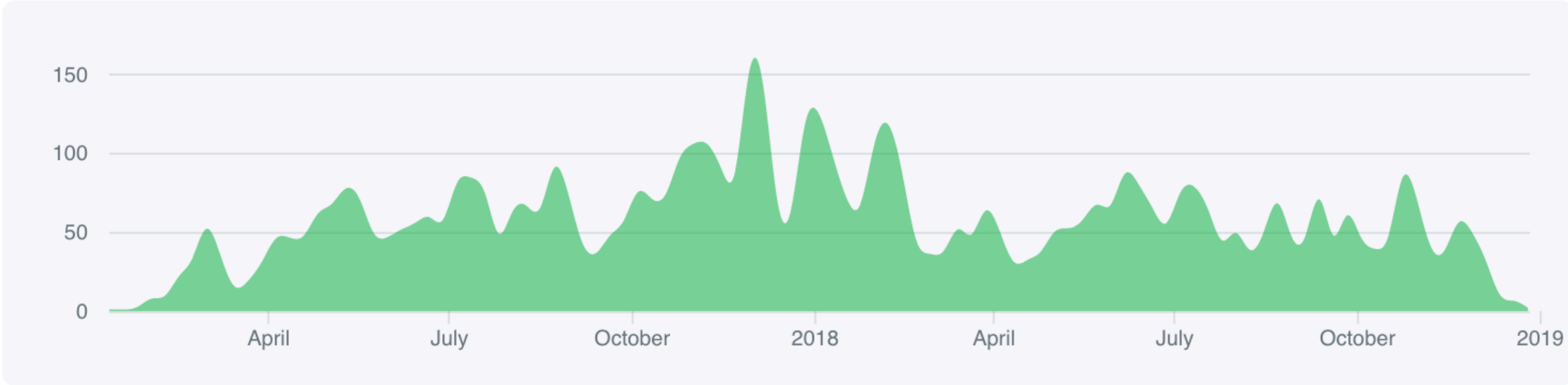
Challenges

- Build an in house e-commerce platform with **tiny** team
- For a company that wants to sell clothes from a **table** not an **iphone**
- While decreasing their **marketing budget**
- Without any (serious) **UX** expertise in-house

Organization



Timeline



Project Timeline

Core Shopping Costes - march 2017 - december 2017

Core Shopping The Sting - january 2018 - may 2018

Checkout Costes - january 2018 - october 2018

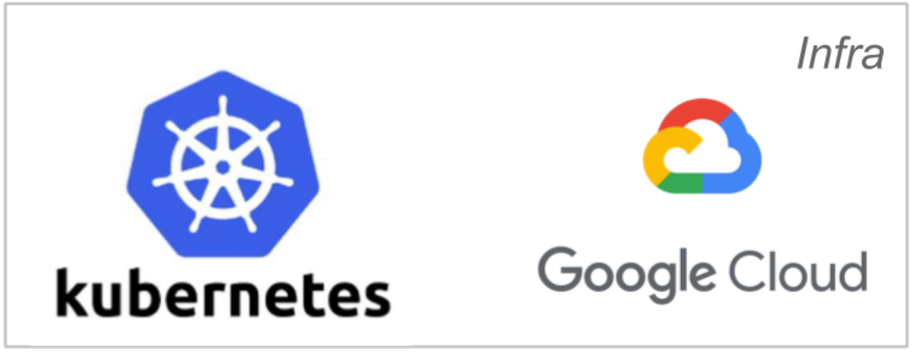
Checkout The Sting - october 2018 - january 2019

🍾 Going live 🍾

As of today both **core shopping** and **checkout** have been delivered for all **Sting brands**

Technology Stack


Application
lagom
akka
play


Infra
kubernetes
Google Cloud


Persistence
elastic
APACHE
kafka
cassandra

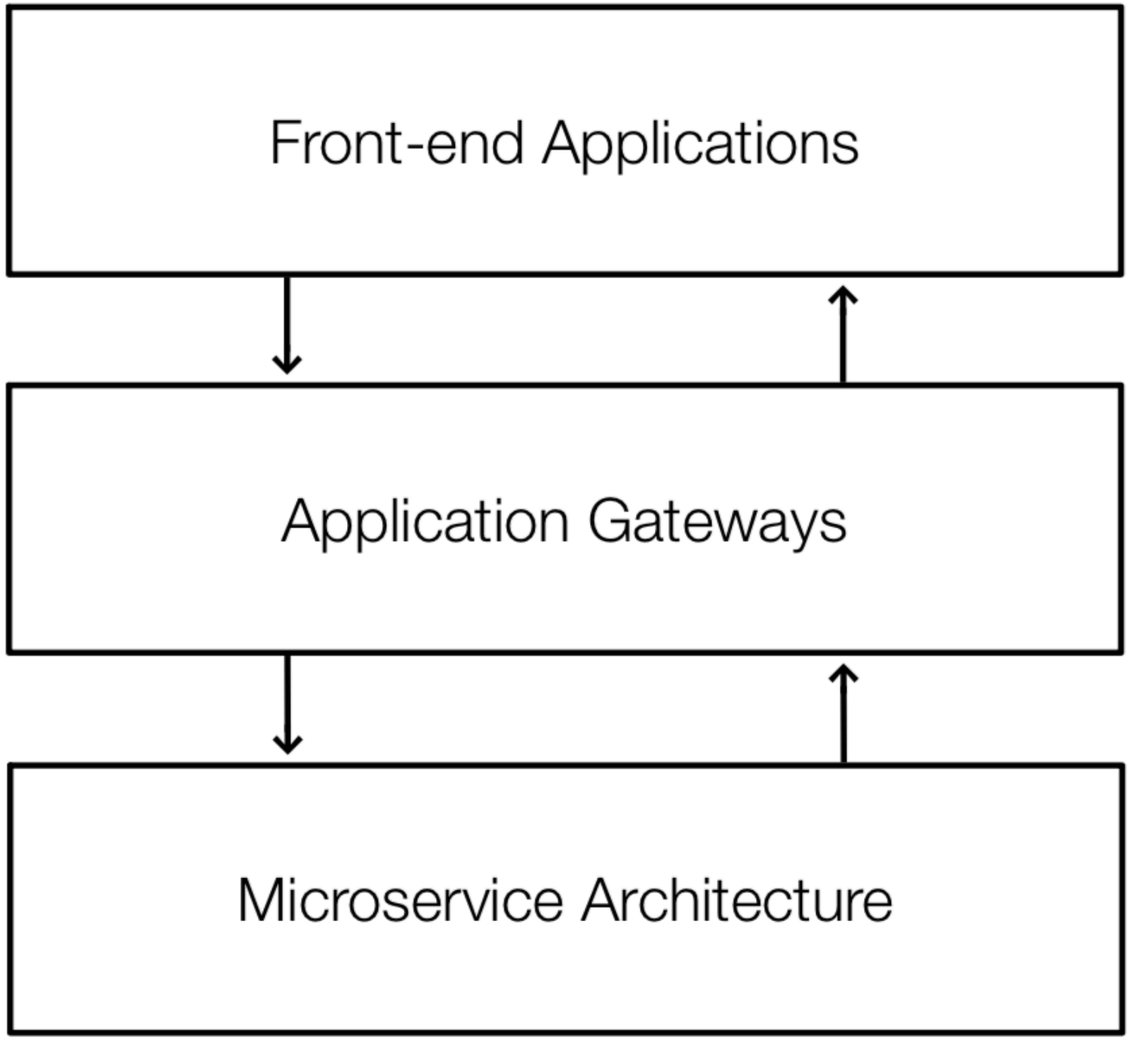
Deliverables

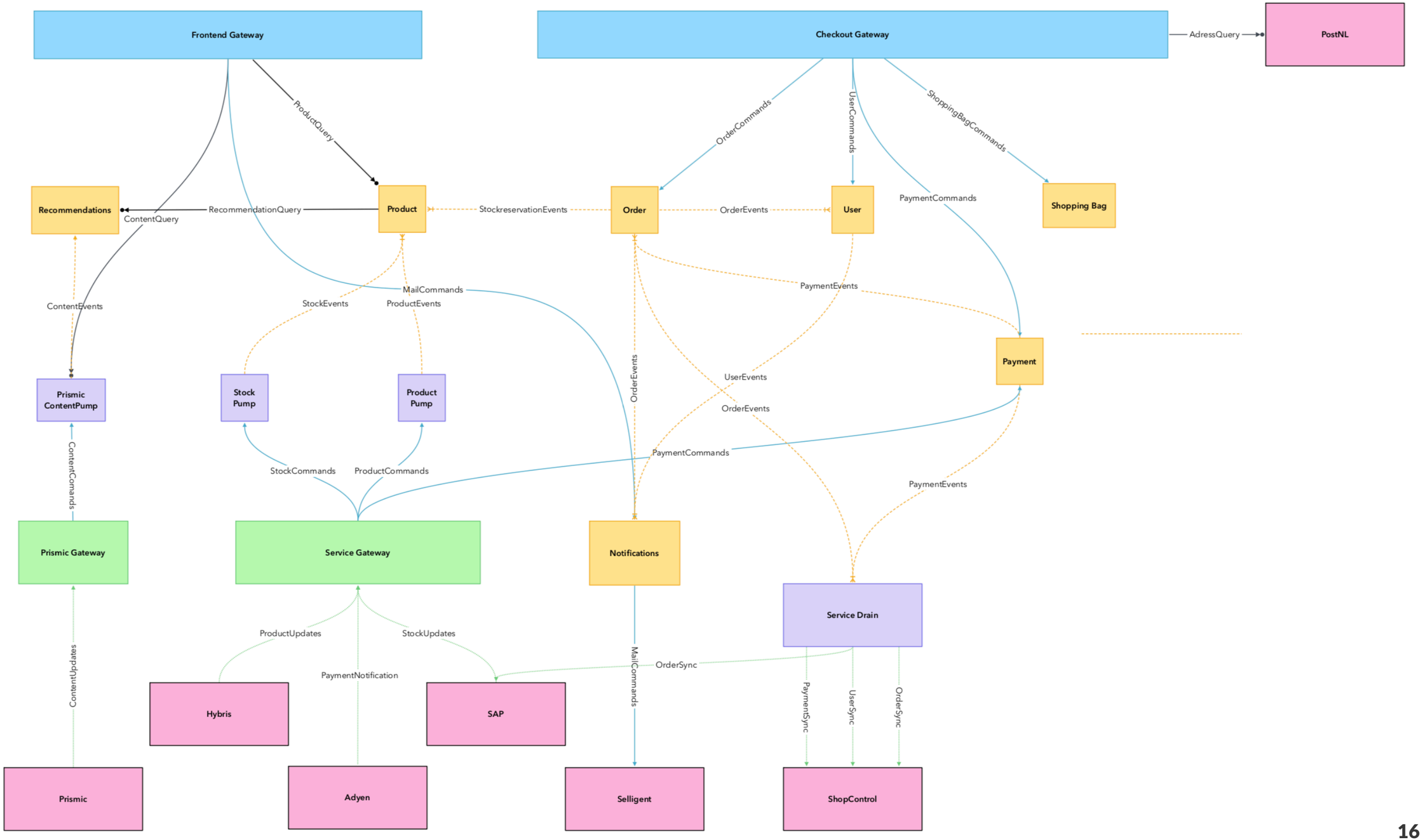
- *10.000+* **commits**
- *2000+* **pull requests**

Language	files	blank	comment	code
Java	389	7867	629	39355
JSX	536	4101	395	35985
JavaScript	321	2936	398	29870
YAML	77	205	171	5531

Architecture

- Layered architecture
- Event sourced persistence
- Read optimized storage through CQRS


Front-end Applications
Application Gateways
Microservice Architecture


Frontend Gateway
Checkout Gateway
AdressQuery
PostNL
ProductQuery
OrderCommands
UserCommands
ShoppingBagCommands
Recommendations
RecommendationQuery
ContentQuery
Product
StockreservationEvents
Order
OrderEvents
User
PaymentCommands
Shopping Bag
ContentEvents
StockEvents
ProductEvents
MailCommands
PaymentEvents
Payment
Prismic ContentPump
Stock Pump
Product Pump
OrderEvents
UserEvents
OrderEvents
PaymentCommands
ContentCommands
StockCommands
ProductCommands
PaymentEvents
Prismic Gateway
Service Gateway
Notifications
Service Drain
ProductUpdates
StockUpdates
MailCommands
OrderSync
PaymentNotification
ContentUpdates
Hybris
SAP
PaymentSync
UserSync
OrderSync
Prismic
Adyen
Selligent
ShopControl

Architecture

Where it shines

- Nearly no changes to *out-of-scope* services
- Events can easily be propagated to *external services*
- Free *audit trails*

Architecture

Where it hurts

- **event sourcing** is hard to debug
- Consistency in an **in-consistent** world
- Mapping, mapping and mapping...
- Maintainability over development speed

Classic Issues

- CI on a monorepo 😭
- Running the micro services landscape locally ♨️
- Clusters are hard 🧐
- Choosing service boundaries 👉|👈
- Building the distributed monolith 🎭
- Keeping dependencies up-to-date 🌱

Results

- 😁 Happy Customer!
- Reduced **running costs**, a lot 💰
- Platform that can handle traffic

Retrospect

- Adyen will **screw you** at one point 💸
- Don't build integrations without documentation 📄
- **Small bang** can be smaller 🤯

The Future

- Autonomy
- Continuity
- Vision