Domain Driven Design for decoupling monoliths

Berlin DevFest 2019

Magomed Chatuev
Software Engineer
twitter: @magomed_chatuev
telegram: @mchatuev

Microservices

*Applications built from microservices aim to be as **decoupled** and as **cohesive** as possible – they own their own **domain logic** [that applies to their part of the business problem], and act more as filters in the classical Unix sense – receiving a request, applying logic as appropriate and producing a response.*

Martin Fowler

Domain Driven Design

DDD is primarily about modeling a

Ubiquitous Language

in an explicitly defined

Bounded Context

DDD strategic patterns

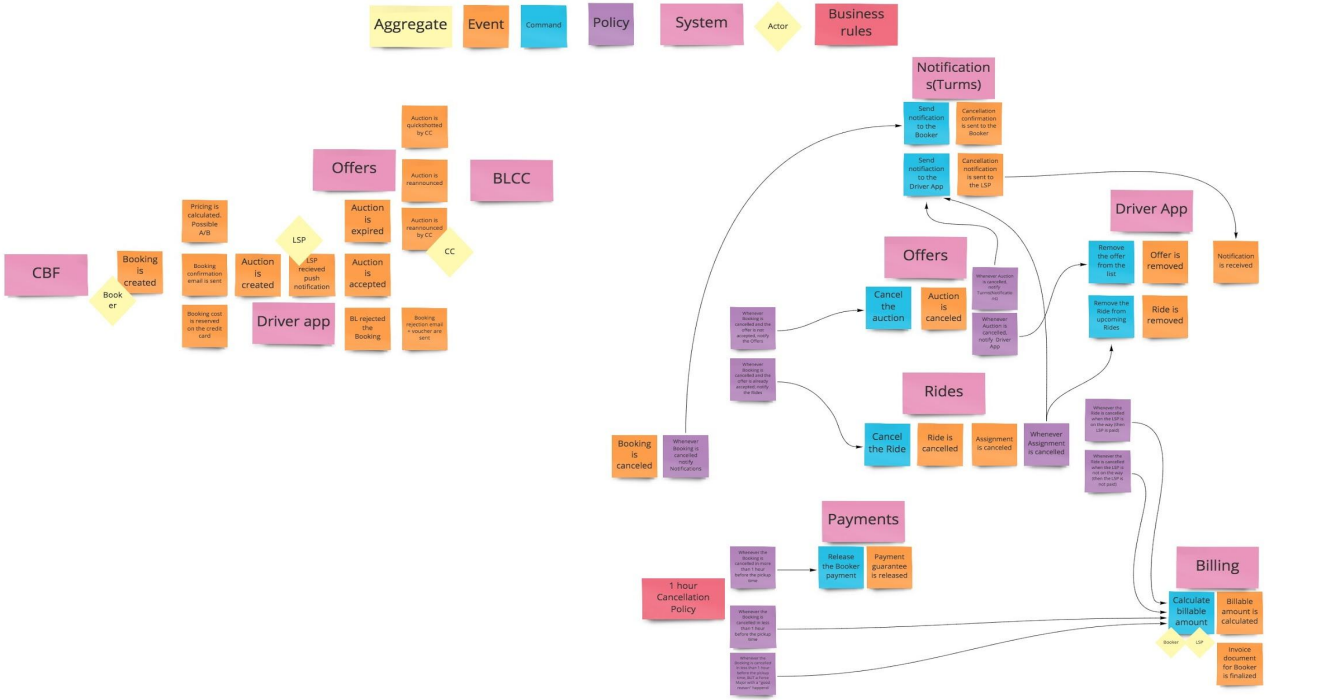
- **Bounded Context** is a semantic contextual boundary. Within the boundary each component of the software model has a specific meaning and does specific things. The components inside a Bounded Context are context specific.

- The software model inside the context boundary reflects a language that is developed by the team working in the Bounded Context and is spoken by every member of the team. The language is called the **Ubiquitous Language**

Event Storming

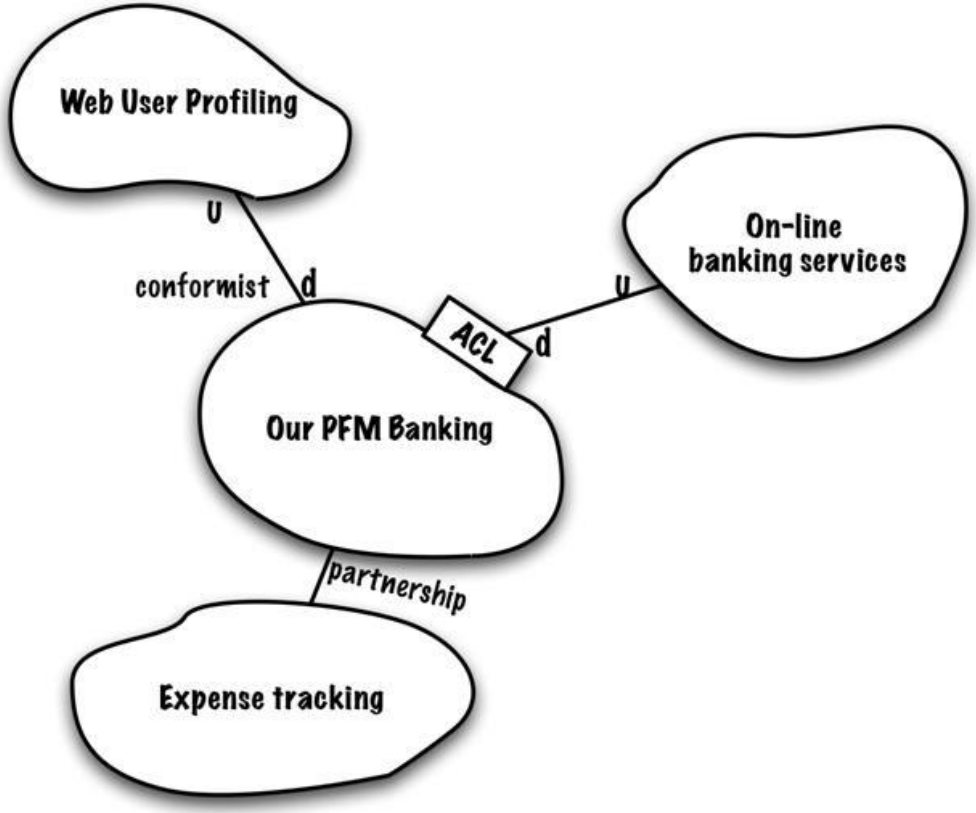
- to discover Business Domain and its most effective areas for improvements;
- to bring people with the questions and knowledge holders who know the answers in the same room and build a model together.
- to design the model, which is perfectly aligned with a **Domain-Driven Design** implementation style and allows for a quick determination of **Context** and **Aggregate** boundaries.

Event Storming example: Bookings subdomain



Context mapping

Context mapping is a topology of communication of subdomains in their defined contexts.



DDD tactical patterns. Aggregate

Aggregate *is a cluster of domain objects that can be treated as a single unit.*

An example may be an order and its line-items, these will be separate objects, but it's useful to treat the order (together with its line items) as a single aggregate.

Martin Fowler

DDD tactical patterns. Entities and Value Objects

- ***Entity:*** *Objects that have a distinct identity that runs through time and different representations.*

  E.g. Booking, Auction/Offer, Money banknote

- ***Value Object:*** *Objects that matter only as the combination of their attributes. Two value objects with the same values for all their attributes are considered equal.*

  E.g. Date, Money value

Microservice boundaries

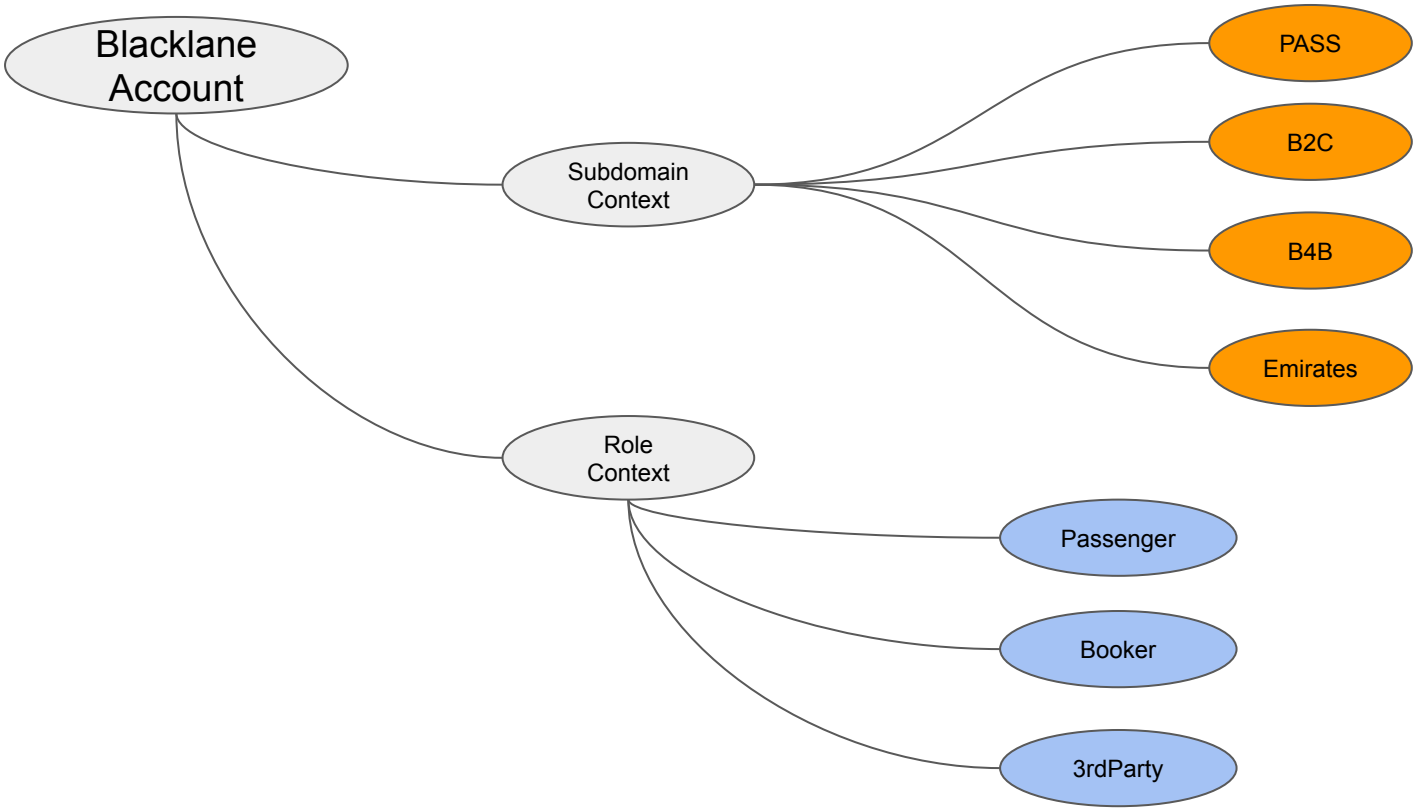
Depending on the size,
a **microservice** could be build as a
Subdomain or just an **Aggregate**
in a single **bounded context**

Real-life example: User Account

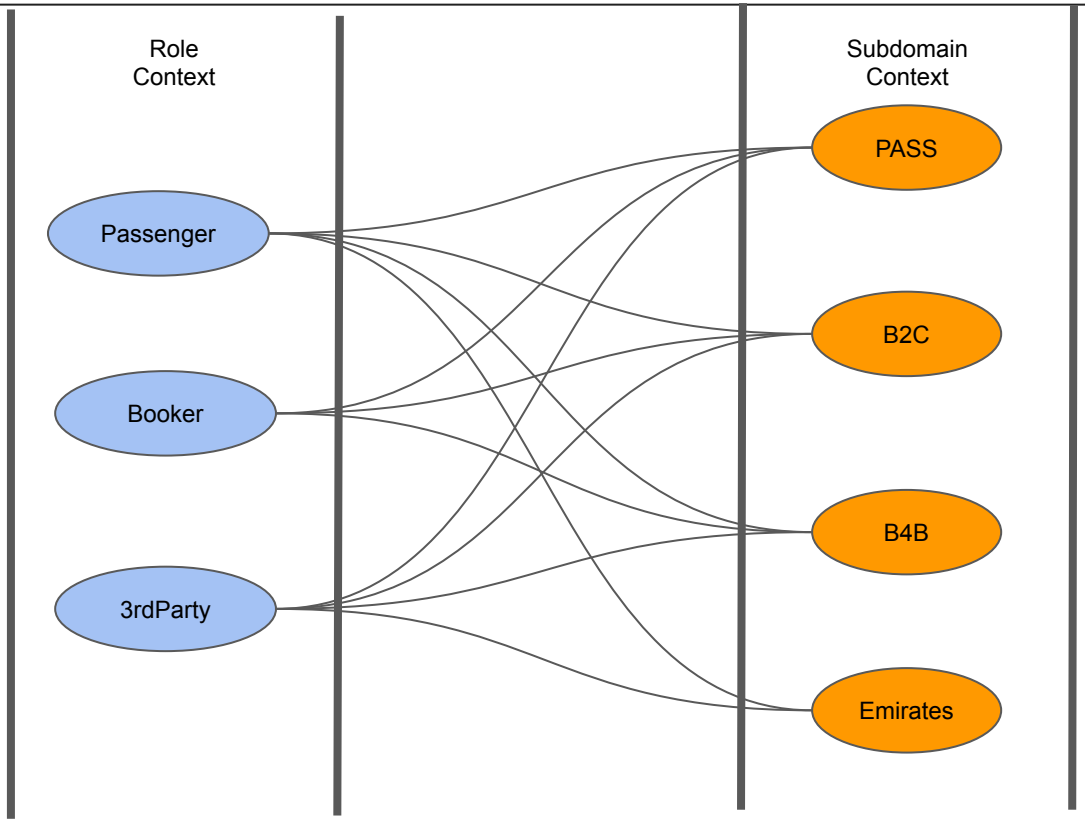
Ubiquitous Language

1. **Go Show** - immediate/short term Booking Request done by business or 1st class flight travellers in an airport via ASM
2. **ASM(Airport Station Manager)** - a manager at a standesc in an airport
3. **Third-party Booking -** a booking done by a registered User to an unregistered third-party
4. **PNR(Personal Name Record)** - an ID, created by flight-operator, and used by BL match Passenger to a Booking

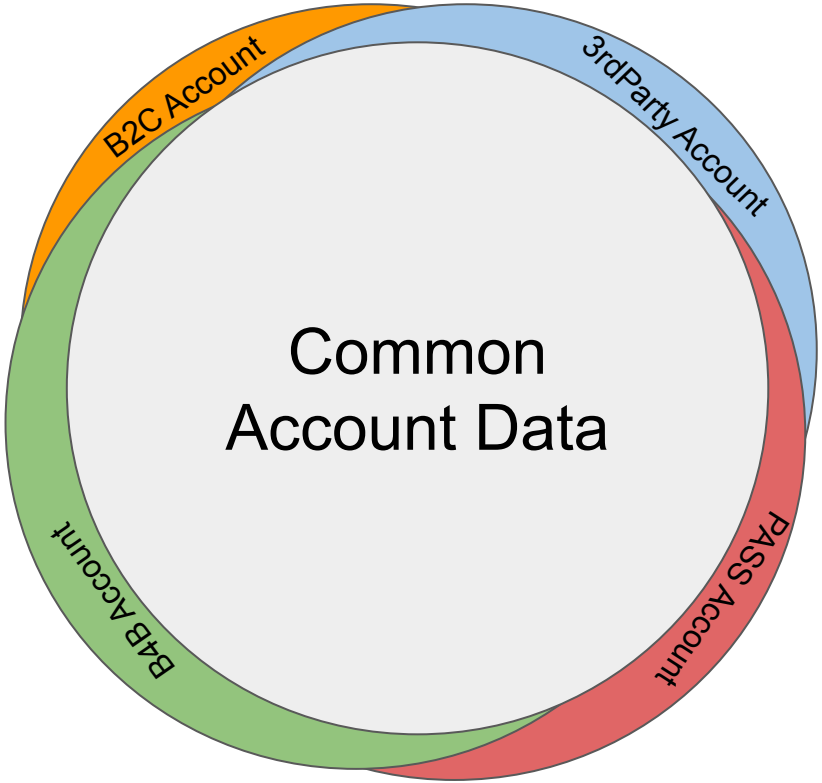
Subdomains and Roles



Roles and Contexts Intersections



User Account



Thank you!
&
Let’s keep in touch

twitter: @magomed_chatuev
telegram: @mchatuev