Building A Customer Journey using Domain Driven Design and GraphQL

Anand Chandrashekar
Customer Success
Dgraph Labs

Agenda

- Resources and Introduction
- Digital Customer Journeys
- Understanding the Ordering Journey
- Modeling the Journey Domain using Domain Driven Design
- Creating the GraphQL Schema
- The Ordering Journey in Action
- Dealing with Changes in the Journey

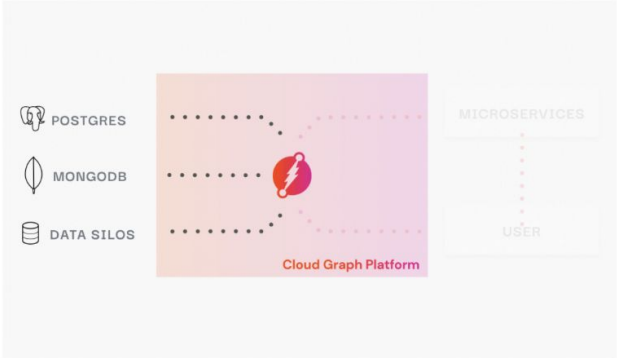
Resources

- Schemas and other artifacts are available at https://dgraph.io/blog/post/ddd-with-graphql/
- Get Started With Dgraph and GraphQL: https://dgraph.io/docs/graphql/quick-start/
- Get Started With Dgraph Cloud: https://dgraph.io/docs/cloud/cloud-quick-start/

A Brief Introduction to Dgraph

What is Dgraph?

Dgraph is a native GraphQL graph database that is built to be distributed. This makes it highly scalable, performant, and blazingly fast – even for complex queries over terabytes of data.



Native GraphQL

Dgraph is the only native GraphQL database on the market. Zero learning curve. Built for modern developers.

Highly scalable

Grow from megabytes to terabytes of data without worrying about database crashes or collapse of functionality.

Blazing fast

Millisecond response speeds on complex, 5+ degree queries at over 15000/s QPS? No problem!

Superior performance

Get fast queries and rapid uploads without needing a mountain of RAM that costs a fortune.

Flexible schema

Save weeks of work when changing your schema on the fly with zero downtime. Get faster iteration and innovation.

Open source

The #1 graph database on GitHub, Dgraph is freely available with Apache 2.0 license. The only open source GraphDB that brings you horizontal scalability and performance.

Digital Customer Journeys

The modern customer expects **nuanced digital** journeys. For example:

- Order via smart devices
- Leverage location for delivery
- Self service options for payments and returns

What does it take to **build** these journeys?

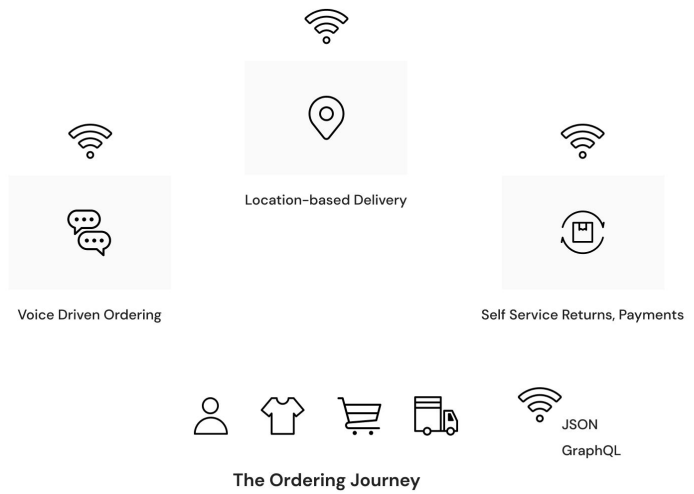
- Iterative, Feedback driven development
- Constant refactoring, especially around data

What are the **challenges** faced by developers?

- Time taken to continuously update data model
- Refactoring data services due to changes in data model
- Manage the bindings between processes and data

These ordering journeys are characterized by a reliance on a mesh of API-driven apps. JSON and GraphQL are popular options for building these APIs and apps.

We will explore **modeling techniques** using GraphQL and Dgraph that support these rapid iteration needs.



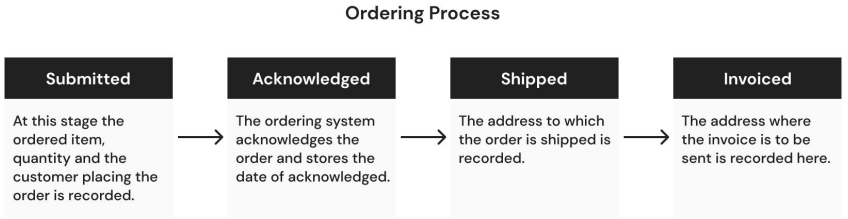
Understanding the Ordering Journey

The Ordering Process has the following 4 steps

- Submit
- Acknowledge
- Shipped
- Invoiced

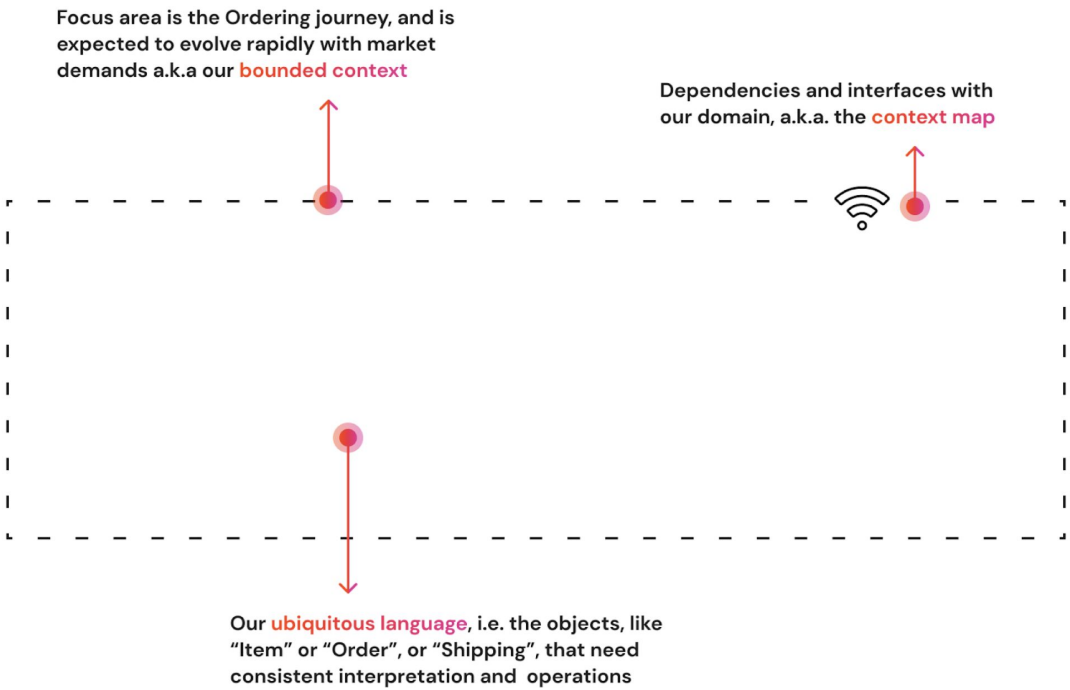
The important entities in the domain are

- Customer
- Product
- The Order and associated Process

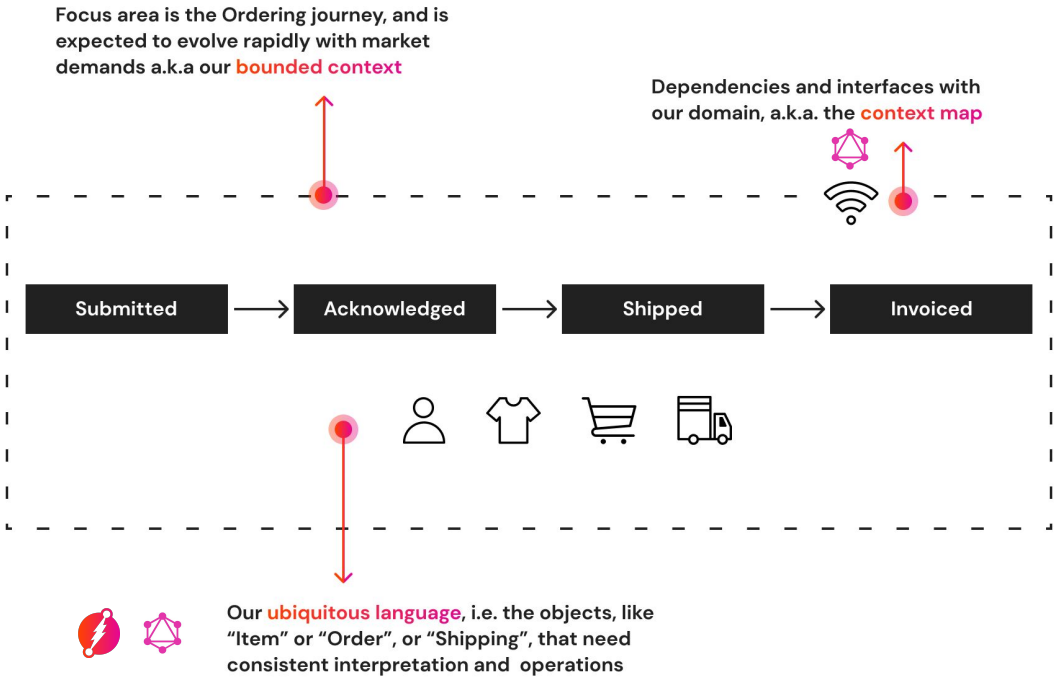


Order Journey Domain

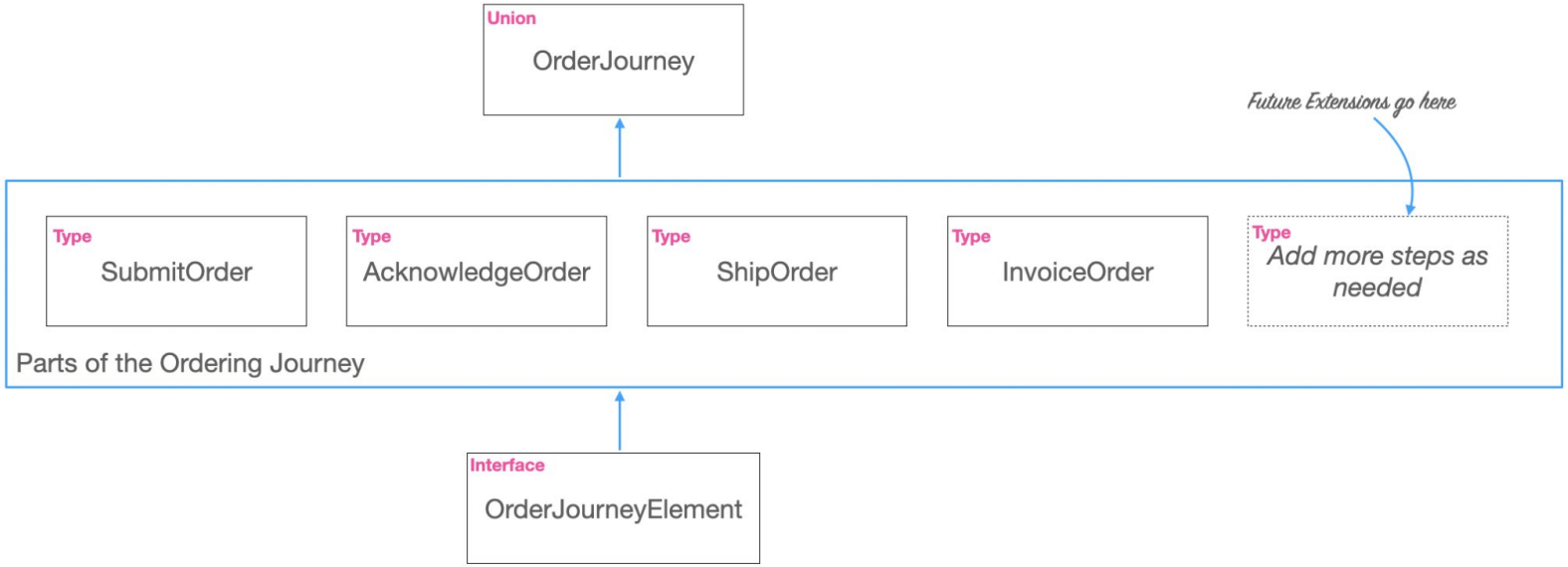
Designing the Customer Journey using DDD



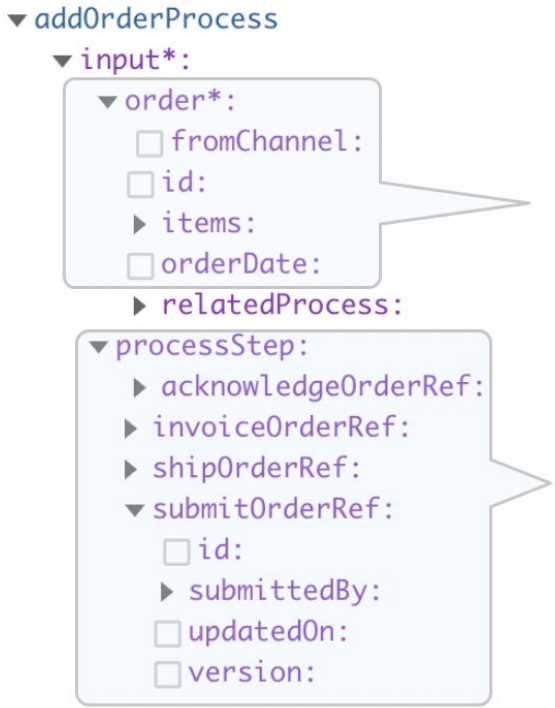
Designing the Customer Journey using DDD



Add the Business Process Context



A peek into the GraphQL API



Order

Journey Steps

```
query MyQuery {
  queryOrderProcess {
    processStep {
      ... on AcknowledgeOrder {
        id
        acknowledgedOn
      }
    }
  }
}
```