EP32: REST vs. GraphQL

ALEX XU
NOV 12, 2022

This week's system design refresher:

- REST vs. GraphQL (Youtube video)
- How does gRPC work?
- Domain-Driven Design (DDD)
- Data migration with Avro
- Join ByteByteGo Talent Collective

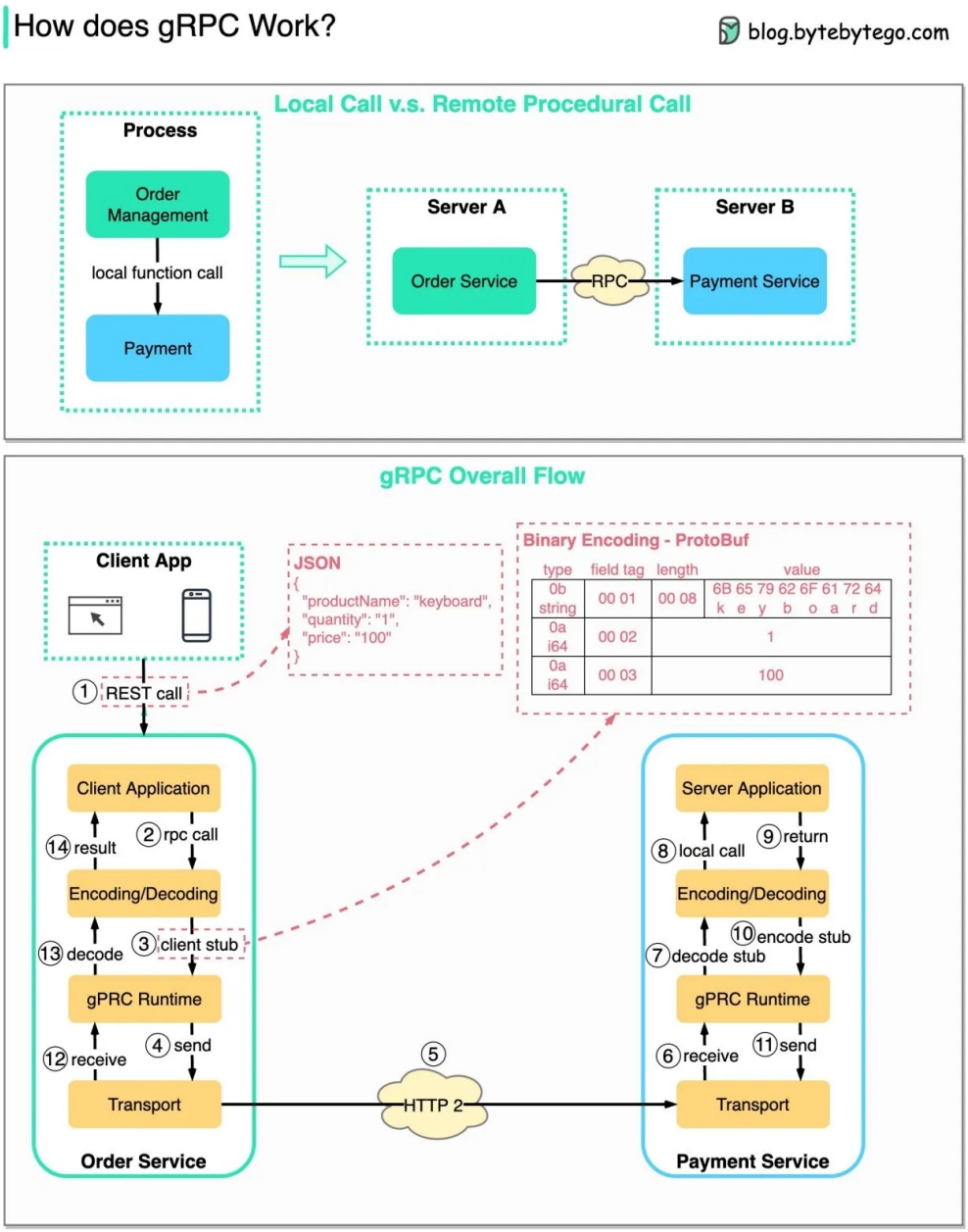
What Is GraphQL? REST vs. GraphQL

GraphQL is a query language for API developed by Meta. It provides a schema of the data in the API and gives clients the power to ask for exactly what they need.

How does gRPC work?

RPC (Remote Procedure Call) is called "**remote**" because it enables communications between remote services when services are deployed to different servers under microservice architecture. From the user's point of view, it acts like a local function call.

The diagram below illustrates the overall data flow for **gRPC**.



Step 1: A REST call is made from the client. The request body is usually in JSON format.

Steps 2 - 4: The order service (gRPC client) receives the REST call, transforms it, and makes an RPC call to the payment service. gPRC encodes the **client stub** into a binary format and sends it to the low-level transport layer.

Step 5: gRPC sends the packets over the network via HTTP2. Because of binary encoding and network optimizations, gRPC is said to be 5X faster than JSON.

Steps 6 - 8: The payment service (gRPC server) receives the packets from the network, decodes them, and invokes the server application.

Steps 9 - 11: The result is returned from the server application, and gets encoded and sent to the transport layer.

Steps 12 - 14: The order service receives the packets, decodes them, and sends the result to the client application.

Domain-Driven Design (DDD)

DDD was introduced in Eric Evans' classic book "Domain-Driven Design: Tackling Complexity in the Heart of Software". It explained a methodology to model a complex business. There is a lot of content in this book, so I'll summarize the basics.

The composition of domain objects:

- Entity: a domain object that has ID and life cycle.
- Value Object: a domain object without ID. It is used to describe the property of Entity.
- Aggregate: a collection of Entities that are bounded together by Aggregate Root (which is also an entity). It is the unit of storage.

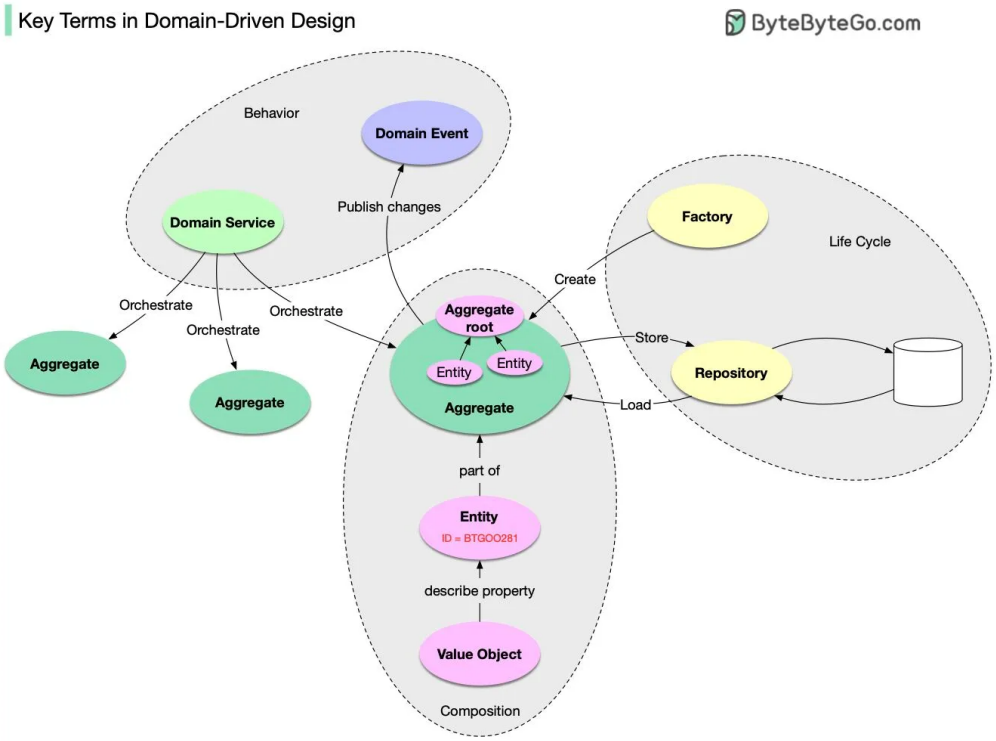
The life cycle of domain objects:

- Repository: storing and loading the Aggregate.
- Factory: handling the creation of the Aggregate.

Behavior of domain objects:

- Domain Service: orchestrate multiple Aggregate.
- Domain Event: a description of what has happened to the Aggregate. The publication is made public so others can consume and reconstruct it.

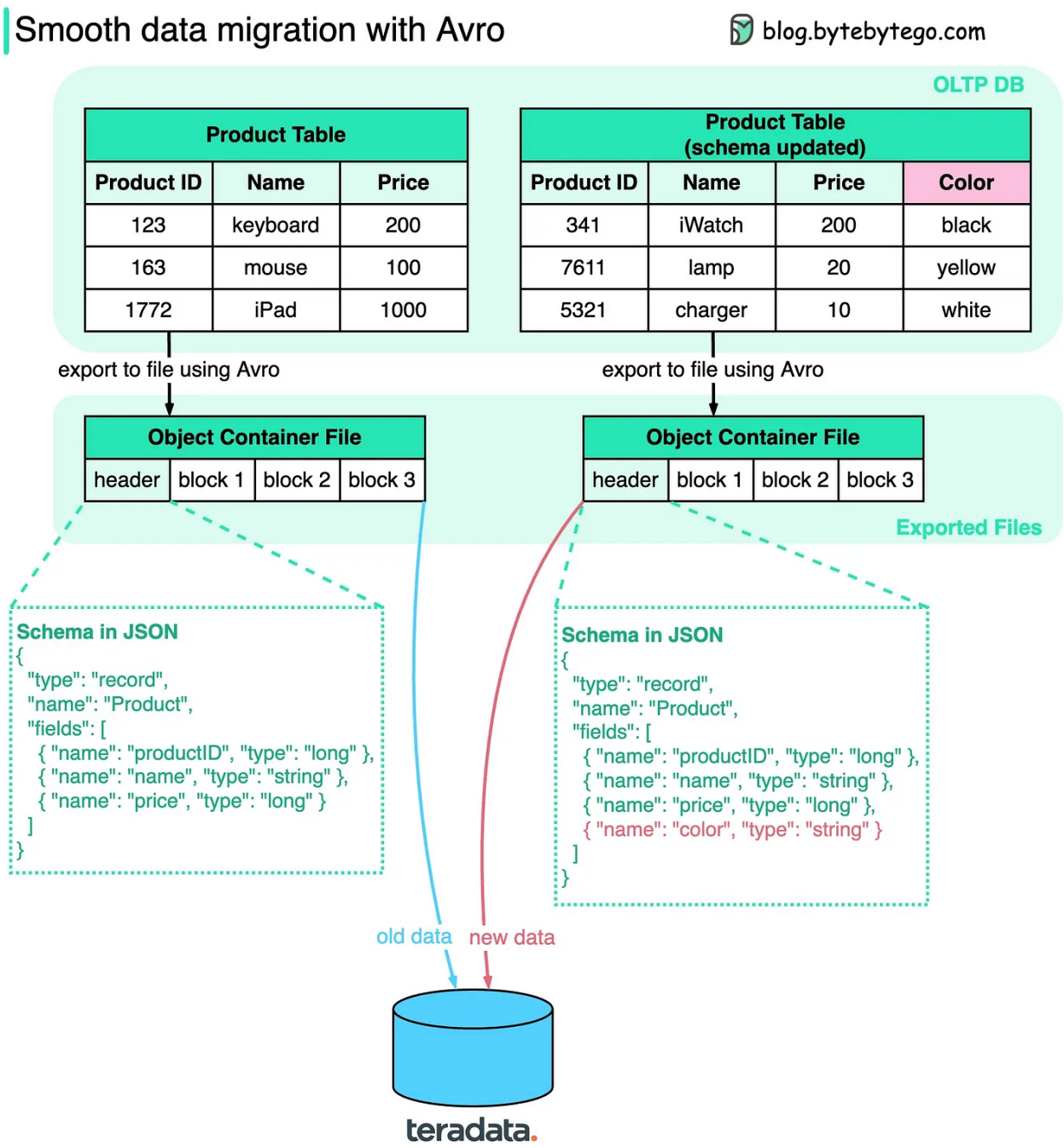
Congratulations on getting this far. Now you know the basics of DDD. If you want to learn more, I highly recommend the book. It might help to simplify the complexity of software modeling.



Data migration with Avro

How do we ensure when performing data migration? The diagram below shows how Apache Avro manages the schema evolution during data migration.

Avro was started in 2009, initially as a subproject of Apache Hadoop to address Thrift's limitation in Hadoop use cases. Avro is mainly used for two things: Data serialization and RPC.



Key points in the diagram:

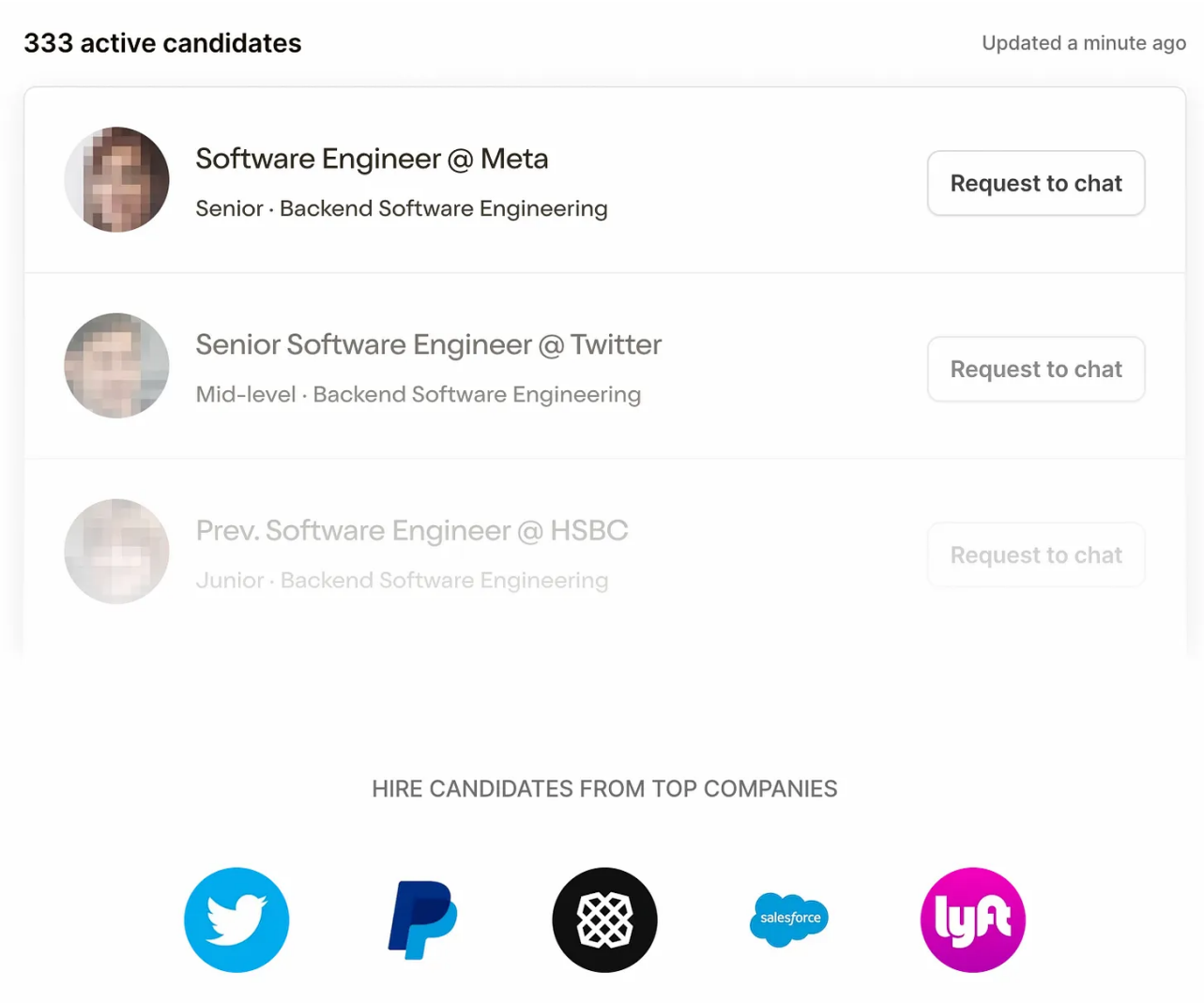
- We can export the data to **object container files**, where schema sits together with the data blocks. Avro **dynamically** generates the schemas based on the columns, so if the schema is changed, a new schema is generated and stored with new data.
- When the exported files are loaded into another data storage (for example, teradata), anyone can read the schema and know how to read the data. The old data and new data can be successfully migrated to the new database.

> Unlike gRPC or Thrift, which statically generate schemas, Avro makes the data migration process easier.

Over to you: There are so many file formats for big data. Avro vs Parquet vs JSON vs XML vs Protobuf vs ORC. Do you know the differences?

Join ByteByteGo Talent Collective

If you're hiring, join ByteByteGo Collective to start getting weekly drops of world-class hand-curated engineers who are open to new opportunities.



If you're looking for a new job, join to get personalized opportunities from hand-selected companies. You can join publicly or anonymously, and leave anytime.

Featured job openings

Openedges: Chief Architect (San Jose, Austin, Remote)

Thanks for reading ByteByteGo Newsletter!
Subscribe for free to receive new posts and
support my work.

172 Likes

2 Comments

Write a comment...

christian picon Nov 21, 2022

I was googling about "Thrift's limitation in Hadoop use cases" without success. May I ask you to point me out to a resource or definition on this regard?

LIKE (1) REPLY SHARE

Chetan Krishna Sangoram Nov 14, 2022

Thanks Alex for small bytes big impact simple technology explanations; interesting to compare gRPC with traditional RPC. Looking at merely the explanation one can fathom, isn't the gRPC so complicated, I mean 14-16 steps, things can go wrong any handoff, why are technologies making the things so complicated? Is it for making a living :)

LIKE REPLY SHARE