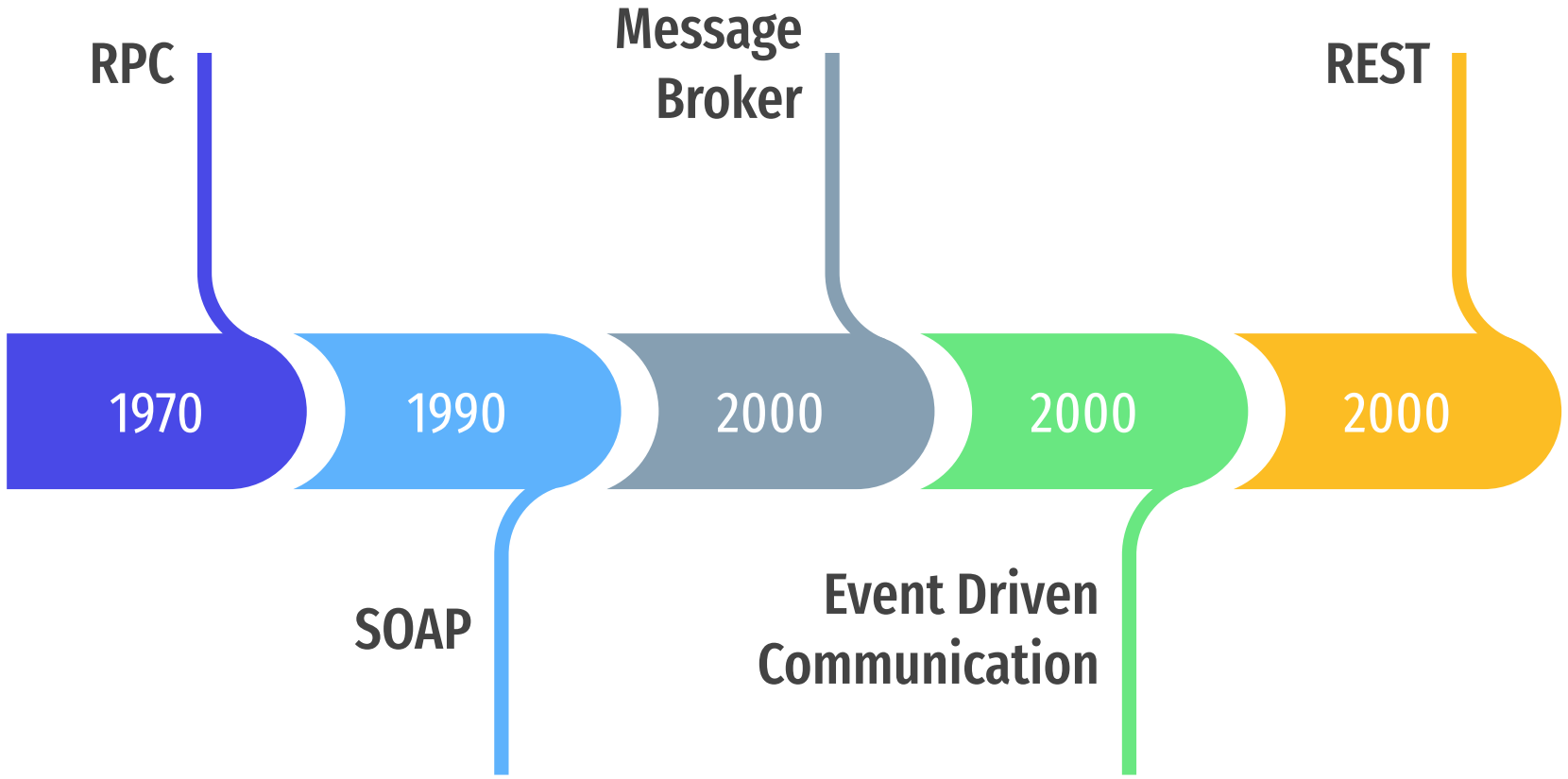
gRPC and more ...

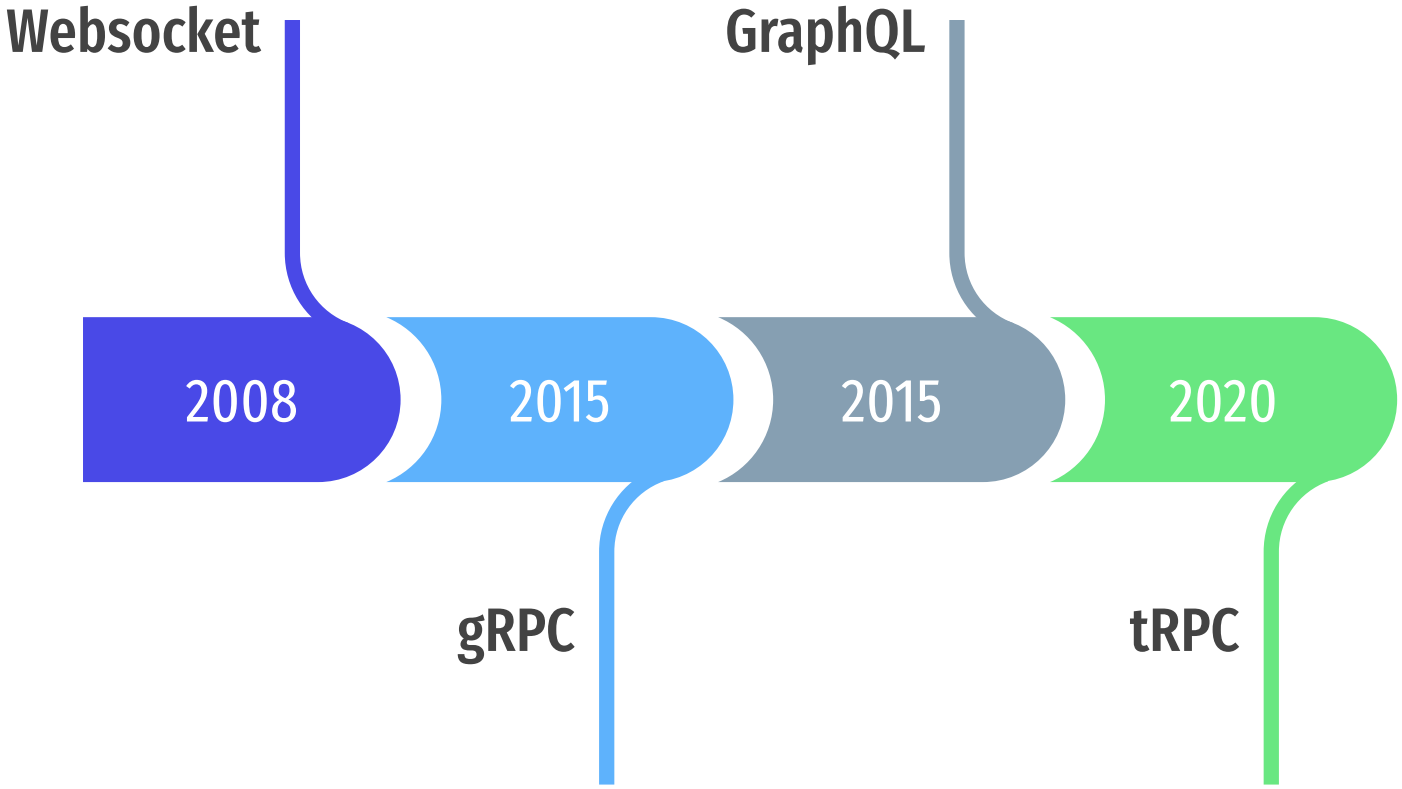
Exploring gRPC it's exclusivity, where it should be used along with implementations

Presented by
Md Sabbir Rahman
Associate Software Engineer
Cefalo Bangladesh Ltd

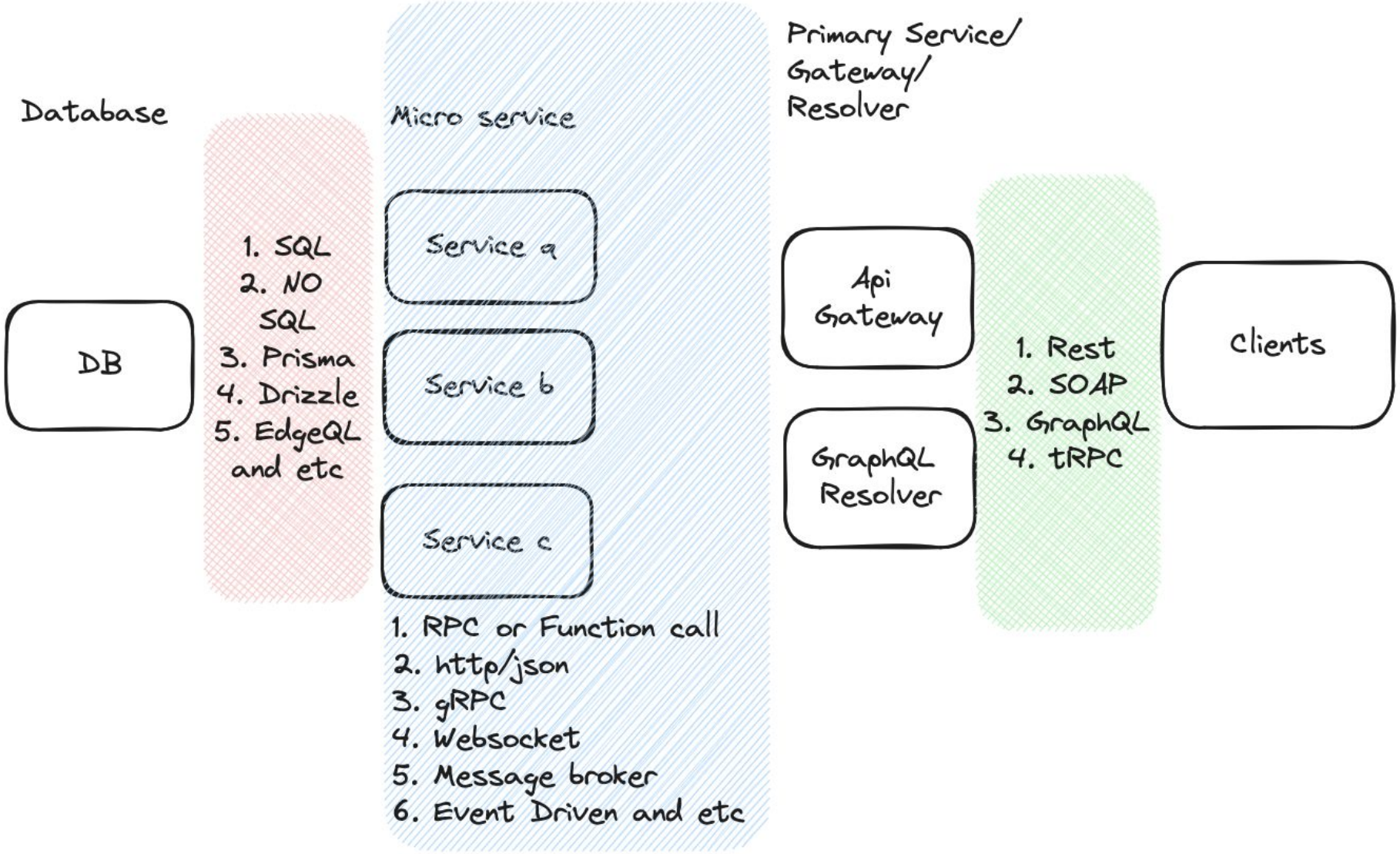
The Timeline



The timeline contd...



The heatmap ...


Database
DB
1. SQL
2. NO SQL
3. Prisma
4. Drizzle
5. EdgeQL
and etc
Micro service
Service a
Service b
Service c
1. RPC or Function call
2. http/json
3. gRPC
4. Websocket
5. Message broker
6. Event Driven and etc
Primary Service/
Gateway/
Resolver
Api Gateway
GraphQL Resolver
1. Rest
2. SOAP
3. GraphQL
4. tRPC
Clients

Some facts about gRPC

- g of gRPC initially means google. But after it become open source it changes with every version (link). Now 1.64 g stands for grateful.
- gRPC use http2 under the hood so by default it use the advantages of http2. There is already strong discussion about using http3
- gRPC is optimized for server-server communication. Where you may need large payload/ high throughput/ real time data / multiplexing/ streamline etc.
- There is grpc-web for browser but it is not optimized . To make support below http2 proxy need to be used.

Some facts about gRPC contd ..

- Protobuf (Protocol Buffer) and binary serialization format makes gRPC stands out from the crowd.
- gRPC as modern tech have some upper hand with some modern facility like load balancing , service discovery
- In gRPC, the skeleton (or server stub) refers to the server-side code that handles incoming RPC calls and executes the corresponding service methods defined in the service contract. The stub (or client stub) is the client-side code that allows applications to call server methods as if they were local functions, managing the communication details and data serialization transparently

What is Protobuf

Protobuf keyfeatures

- You can make your own implementation using this.
- Proto files
- Protoc (Proto compiler)
- Serialisation of Protobuf Objects
- Binary Encoding

Some Facts

- .proto file is independent to both the platform and languages
- Proto file includes generally three thing Field type, Key name, Field number
- The field number serves as unique tag in binary encoding and it's make protobuf smaller in size

Explaining protobuf binary encoding

```
jsonstr = {
    "name": "Sabbir",
    "age": 24
}
```

```
message Person {
    string name = 1;
    int age = 2;
}
```

```
125Sabbir2024
```

jsonString = '{"name": "Jason", "money": 300}'

125Sabbir2024

Explaining protobuf binary encoding contd ..

125Sabbir2024

- 1 represents the field number
- 2 is the wire type (value type)
- 5 refers to length of string (only wire type 2 needs it)
- Sabbir is the data
- 2 represents field number
- 0 is the wire type
- 24 is the data here
- The order of the fields in the `.proto` file is not important. The recipient decodes the data according to the field number.

gRPC vs Other Inter Service

Feature	Message Brokers	Event Driven	gRPC
Type	Async	Async/Sync	Async/Sync
Coupling	Loose	Loose	Tight
Transport	MQTT, AMQP etc	HTTP etc	Http/2
Serialization	Json etc	Json etc	Protobuf
Scalability	High	High	High
Reliability Features	Ack, Retries, persistence	Event sourcing, stream processing	Built in retry, streaming
Complexity	Medium	Medium	High
Performance	Medium	Medium	High

gRPC Demo

tRPC Demo

gRPC Performence test

References

- https://protobuf.dev/programming-guides/encoding/#structure
- https://betterprogramming.pub/what-is-protobuf-and-why-you-should-use-it-14d52646f2a7
- https://aws.plainenglish.io/getting-started-with-grpc-and-protocol-buffers-13b9cf24650a
- https://www.youtube.com/watch?v=ZfccwYUD8H0&t=1s
- https://www.youtube.com/watch?v=Yw4rkaTc0f8&t=813s
- https://www.youtube.com/watch?v=UfUbBWIFdJs&t=250s
- https://grpc.io/docs/

Code

https://github.com/Sabbir-Rahman/grpc-and-more

Thank You!