

Services with gRPC



Bruno Sutic

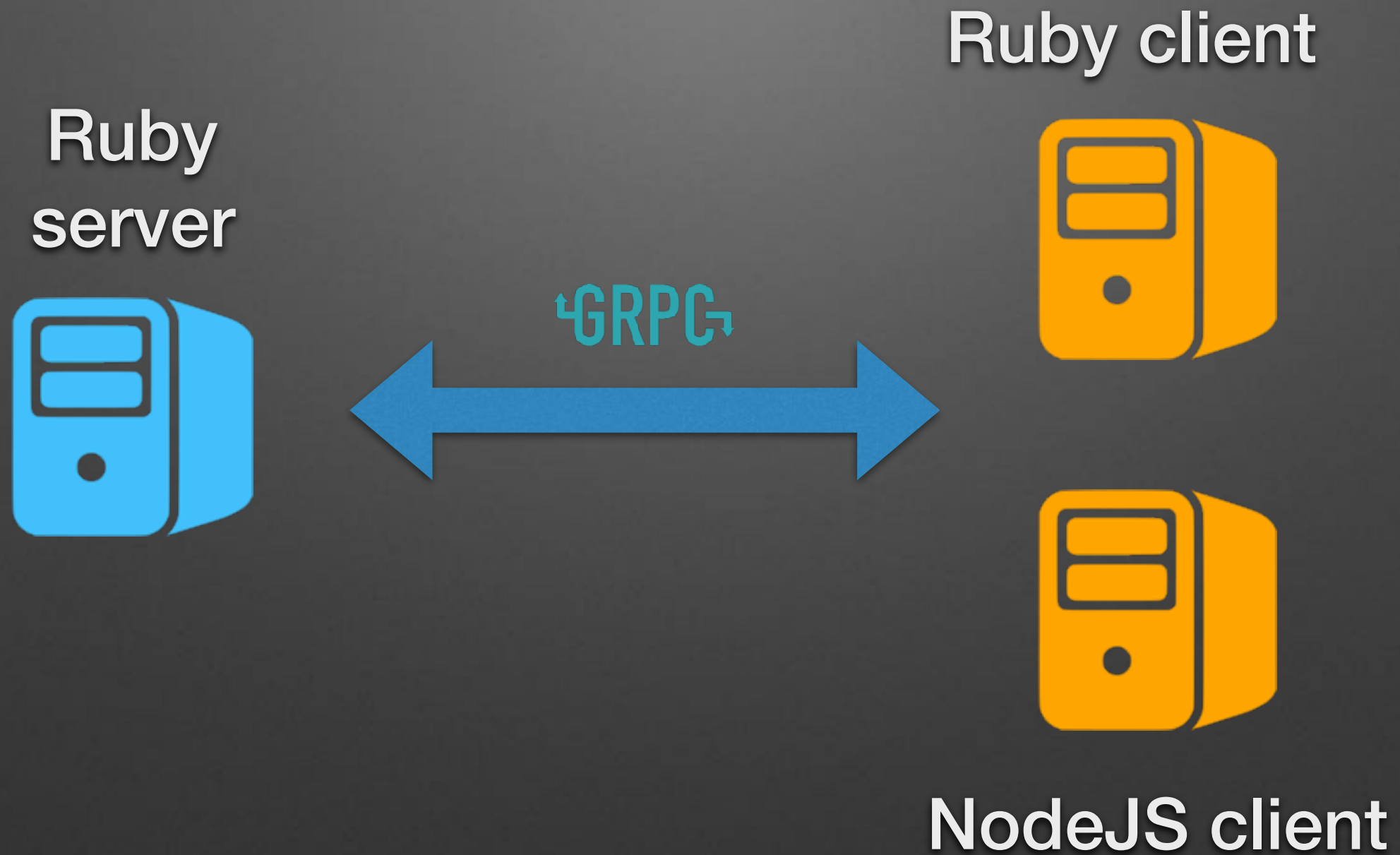
This talk

1. Demo
2. About gRPC
3. How we use gRPC for services

gRPC

- Google Remote Procedure Call
- framework for creating distributed services
- cross language
- designed for high performance and high productivity

Demo: distributed blog app



Demo review

- we were fast & productive!
- cross language: Ruby server, Ruby & JavaScript client
- Idiomatic Ruby, idiomatic JS 👍

About gRPC

- high performance AND high productivity
- used by Google internally for (almost?) everything
- open source, BSD license
- uses 'protocol buffers' as a data interchange format (*.proto file), can use JSON as well
- http2
- cross language: officially supports Python, C++, Java, Ruby, C#, Objective-C, Go, Node + others

About 'Protocol buffers'

- for serializing **structured** data
- think simplified and very optimized XML
- can be used separately from gRPC (eg. to store data)
- cross language
- binary

Ideas

- distributed (micro) services - duh
- server <-> Android / iPhone communication
- public API + client

gRPC vs JSON

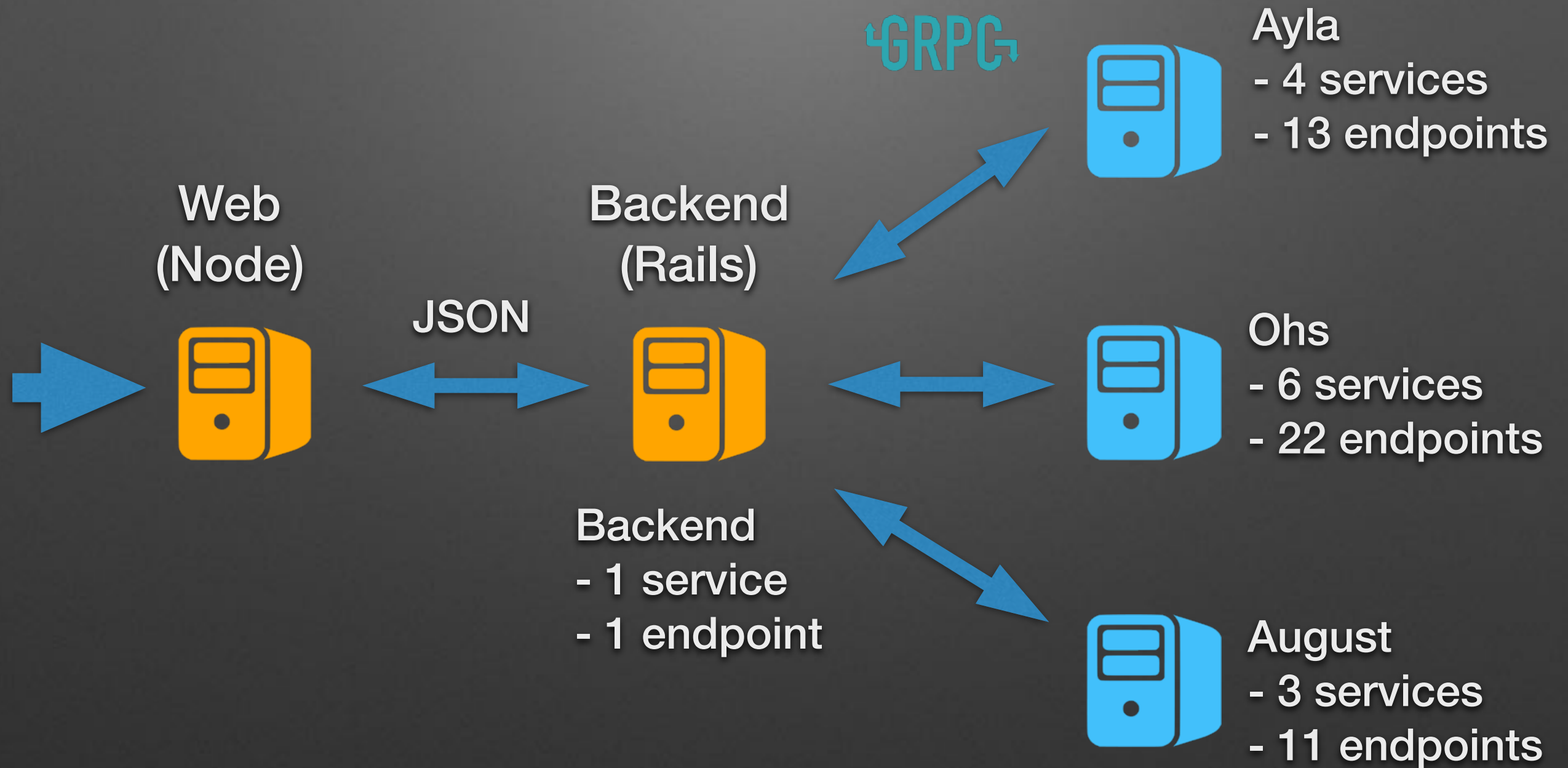
- Pros 👍

- better performance
- uses less network
- strict schema
- more productive

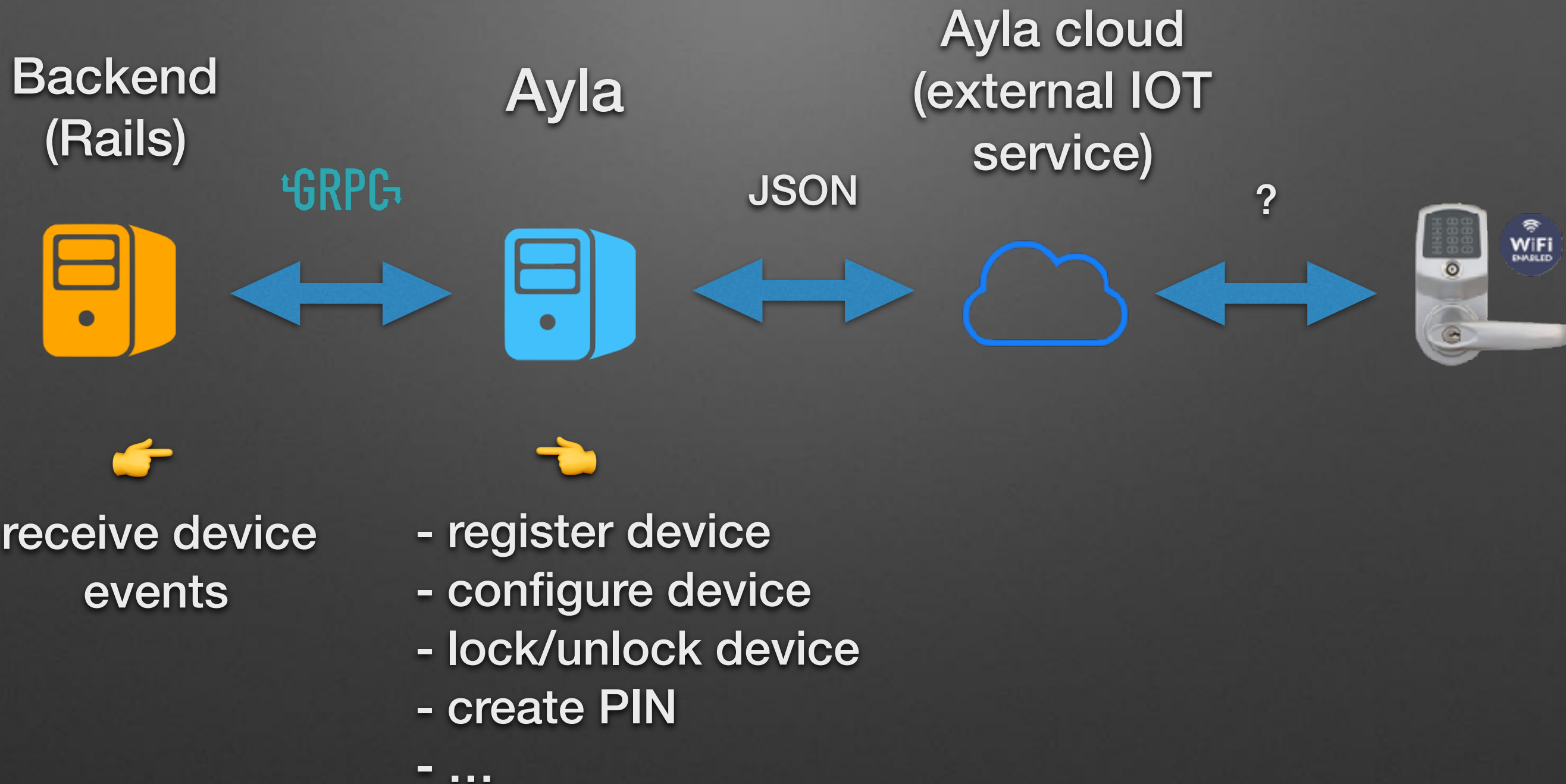
- Cons 👎

- not human readable
- not as popular
- requires learning
- Google 😬

gRPC @ LockState



gRPC @ LockState



gRPC @ LockState

- 5 .proto files, 380 LOC
- Ruby only
- 'rpc-shared' gem for handling gRPC generated files
- we still need caching

Conclusion



Questions?