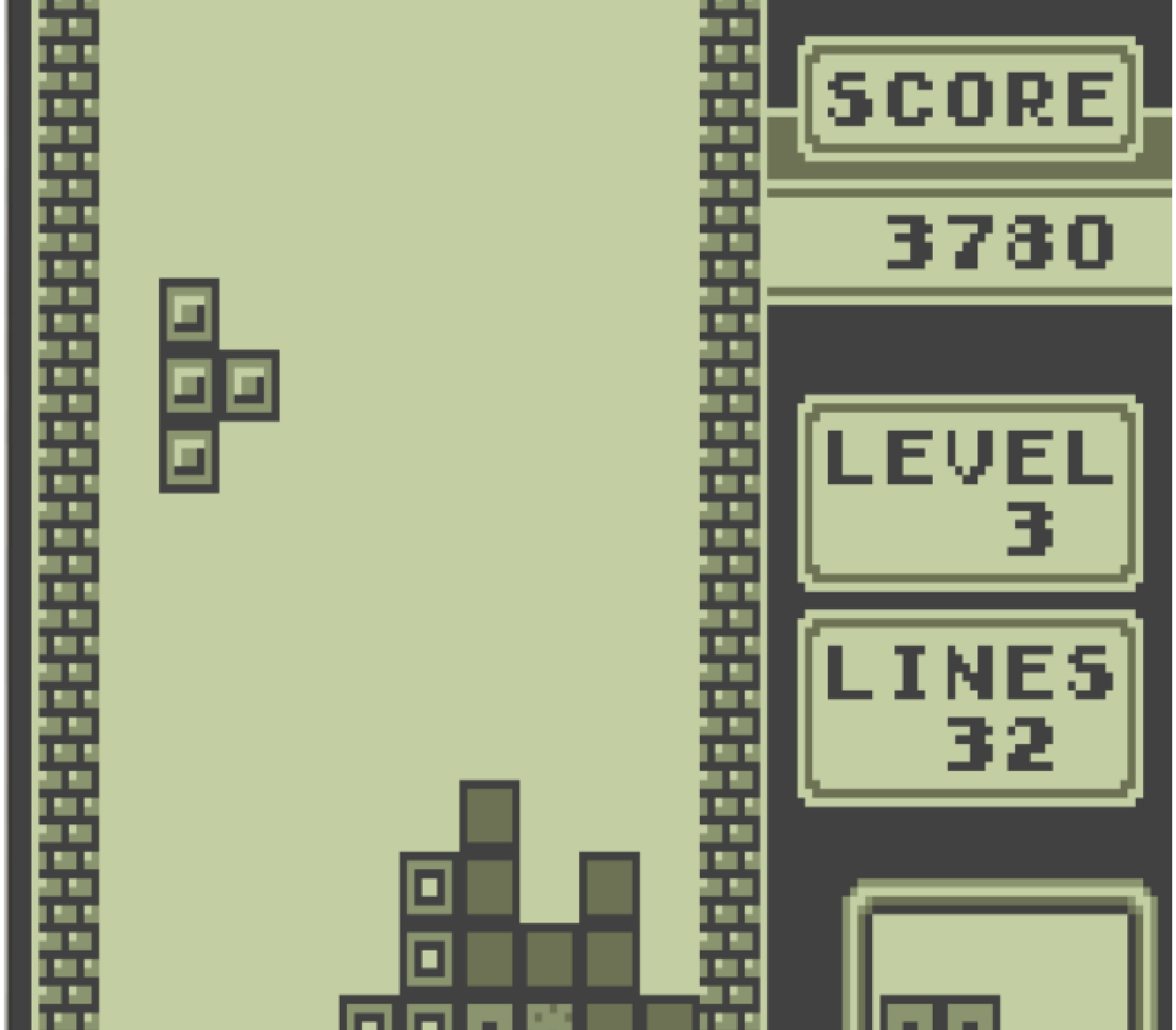
Microservices, Protobuf, gRPC, CERNBox and EOS

Hugo González Labrador

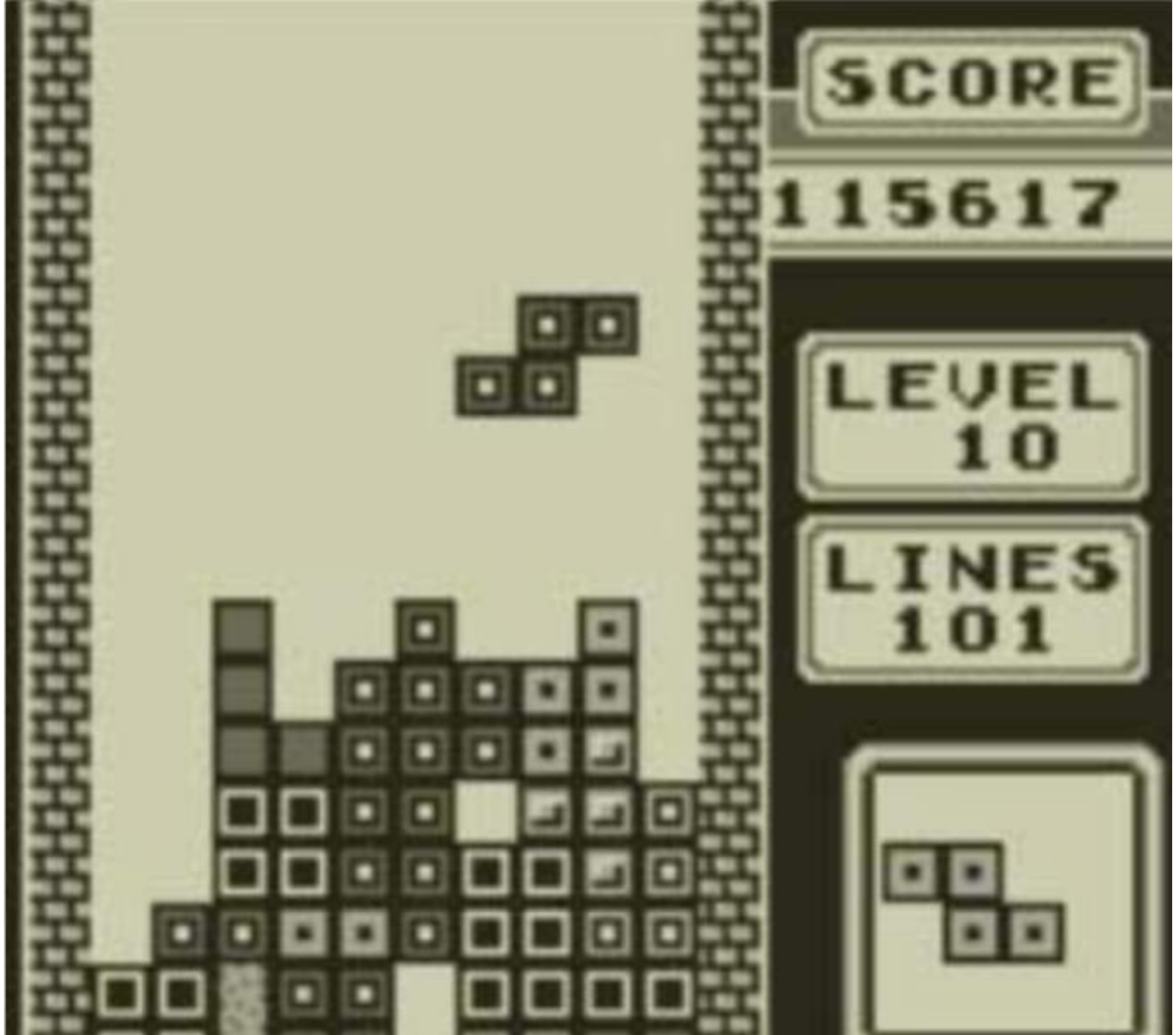
EOS Workshop 2019

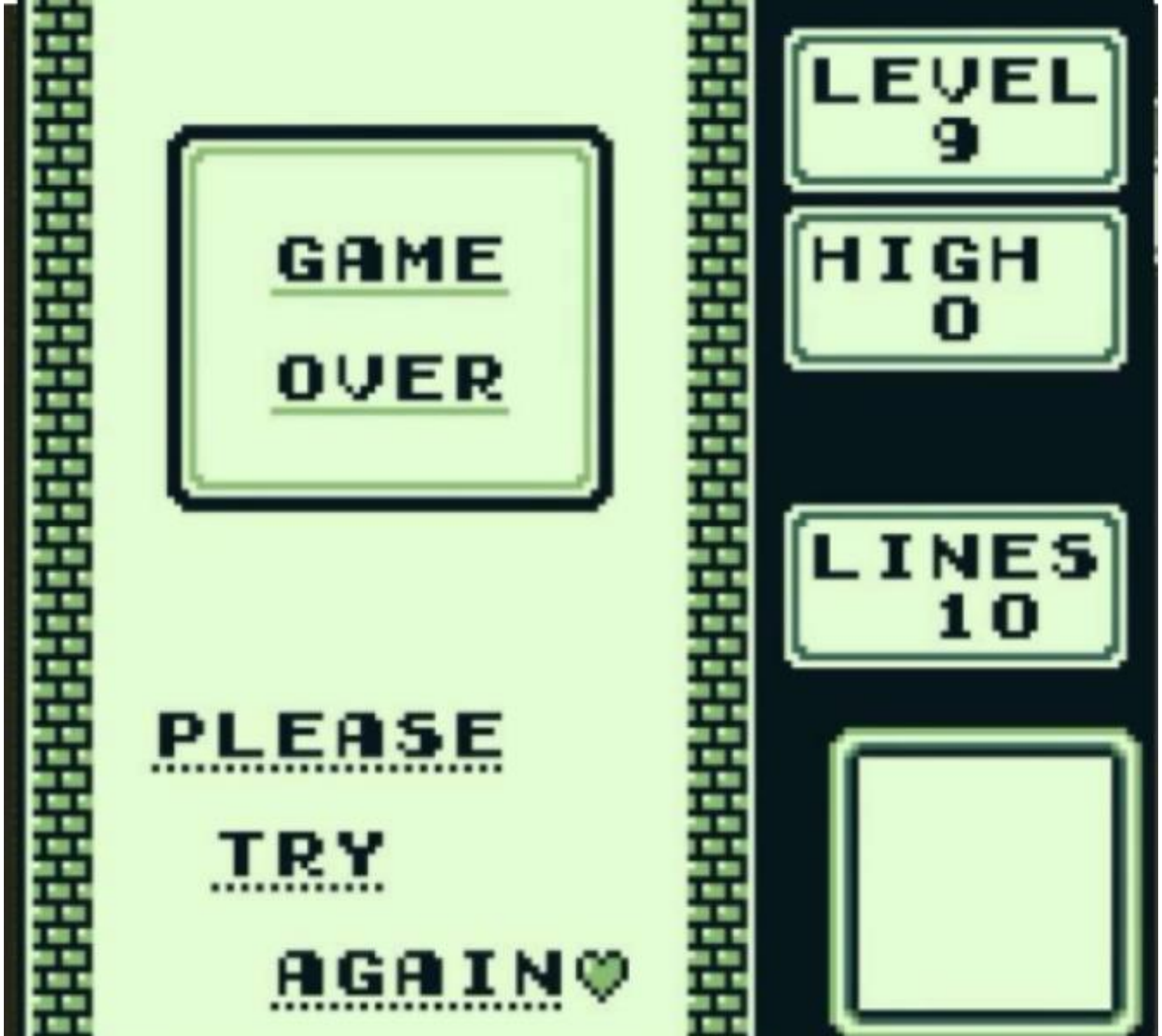


SCORE
3780

LEVEL
3

LINES
32


SCORE
115617
LEVEL
10
LINES
101



GAME

OVER

LEVEL
9

HIGH
0

LINES
10

PLEASE

TRY

AGAIN♥

Challenges of the monolith

Lack of agility

- Difficult to scale
- Architecture gets harder to evolve

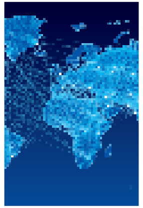
Lack of innovation

- New versions can take months/years
- Long build/test/release cycles

Frustrated customers

- Operations/Support gets complex
- Long time to add features

What are Micro-services



Service-oriented architecture

- Services communicate to each other over networks

Composed of loosely coupled elements

- Update services independently

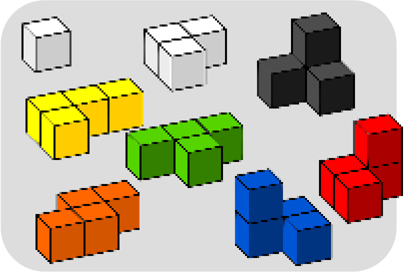
With bounded contexts

- Update of one services without knowing the others

Anatomy of a micro-service

DataStore

- Redis
- Filesystem
- MySQL

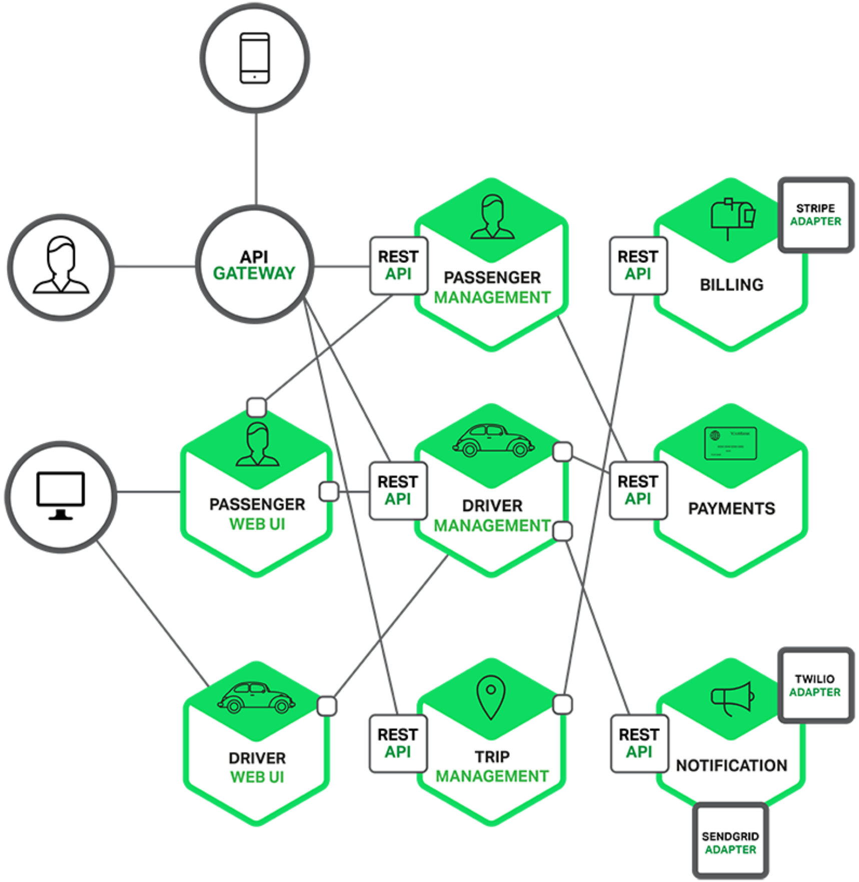


Application

- Libraries
- Tools
- Code

Public API (transport + payload)

- HTTP + JSON
- **gRPC + Protobuf**
- ZeroMQ + JSON
- XROOTD + Protobuf


API
GATEWAY
REST
API
PASSENGER
MANAGEMENT
REST
API
BILLING
STRIPE
ADAPTER
PASSENGER
WEB UI
REST
API
DRIVER
MANAGEMENT
REST
API
PAYMENTS
DRIVER
WEB UI
REST
API
TRIP
MANAGEMENT
REST
API
NOTIFICATION
TWILIO
ADAPTER
SENDGRID
ADAPTER

Benefits of Micro-services

Increased Agility

- Easier to scale each service individually
- Easier to maintain and evolve

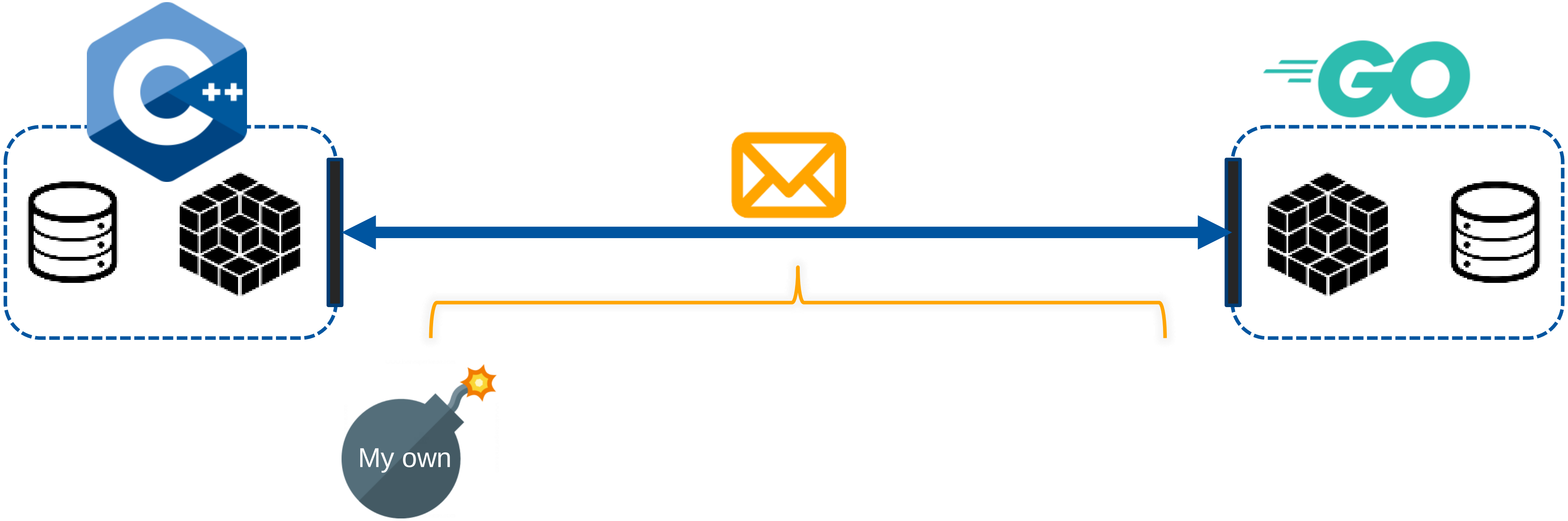
Faster innovation

- Quick development cycle
- New releases arrive faster to customers
- Adopt new technology at fairly low cost

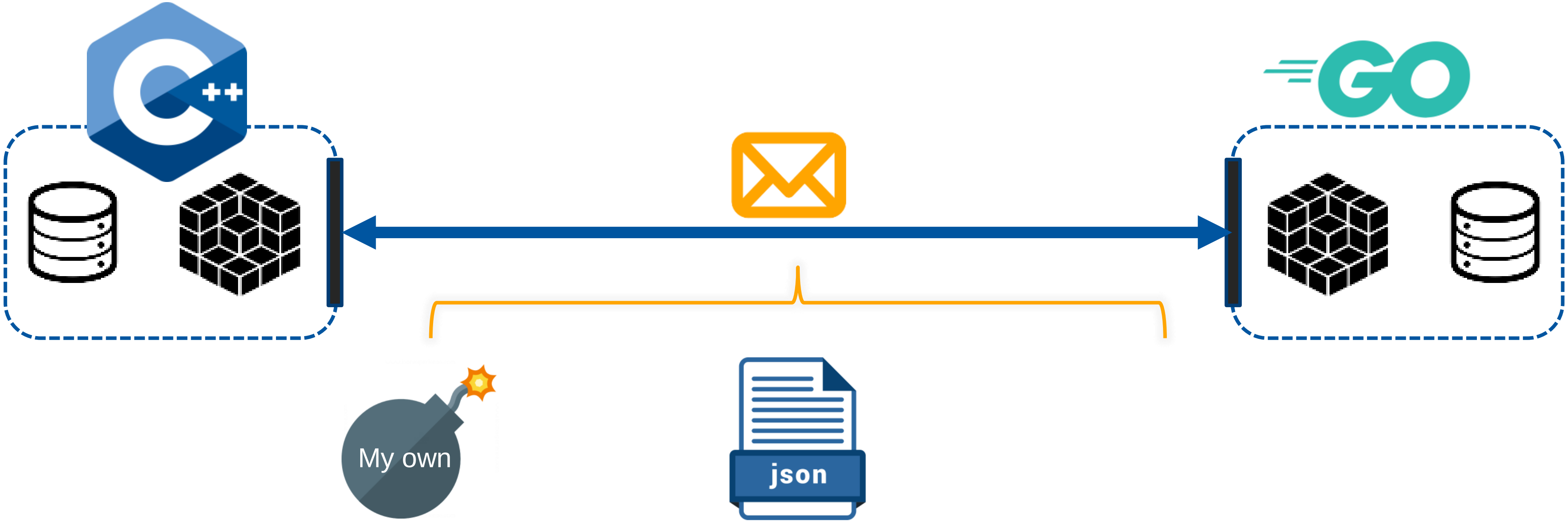
Better consumer experience

- Clear ownership and accountability
- Boost innovation
- Accelerate time to market

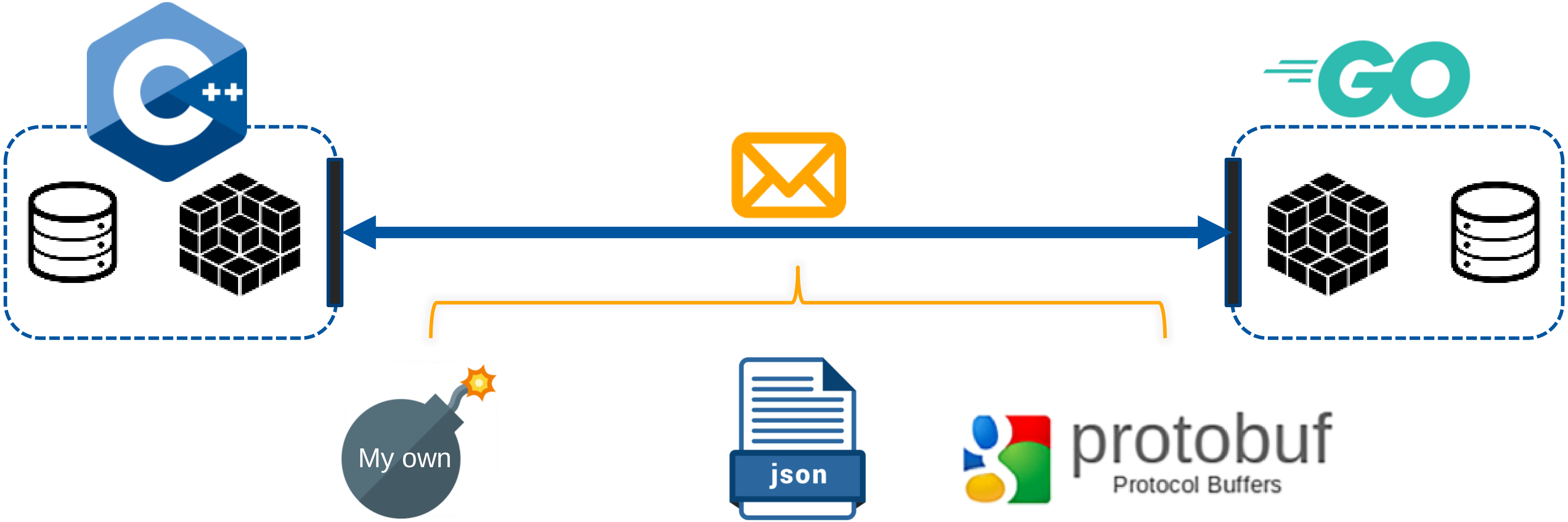
The API: **Payload** and Transport



The API: **Payload** and Transport



The API: **Payload** and Transport

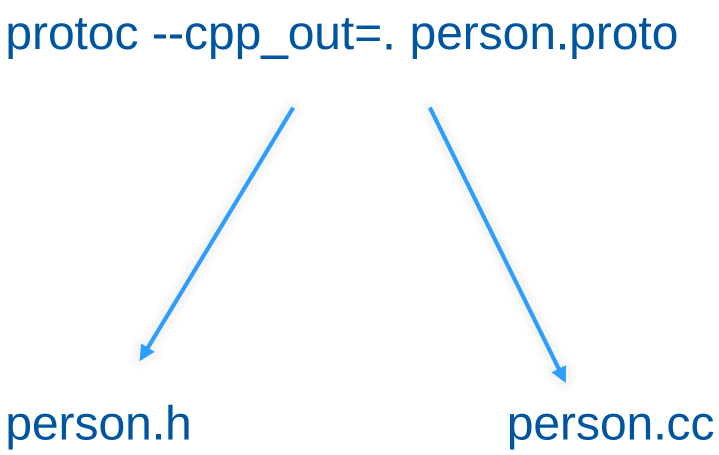


Why Protobuf?

Language-neutral, platform-neutral binary serialization structured data format

Like JSON, but smaller, faster, and simpler

```
message Person {
        int32 id     = 1;
        string name  = 2;
        string email = 3;
}
```

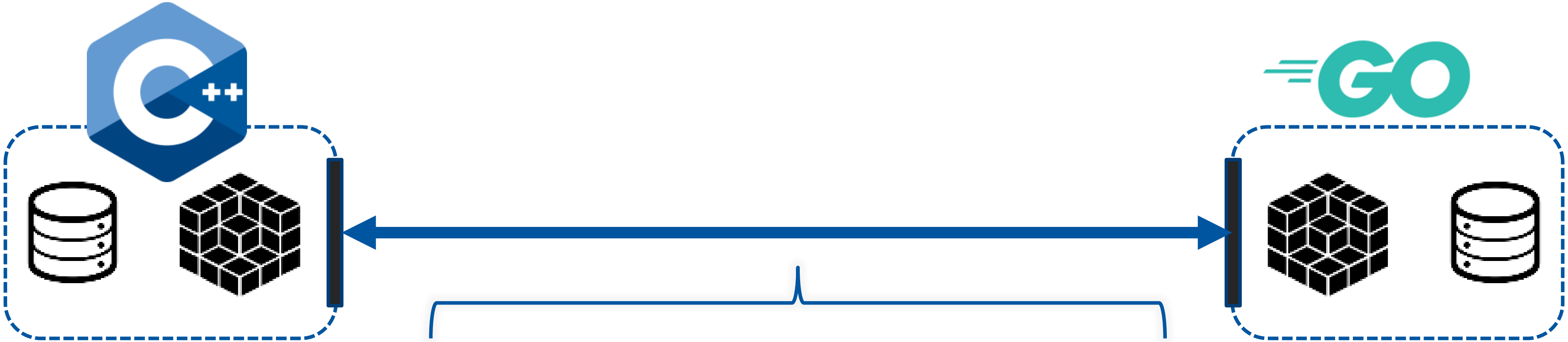


protoc --cpp_out=. person.proto

person.h

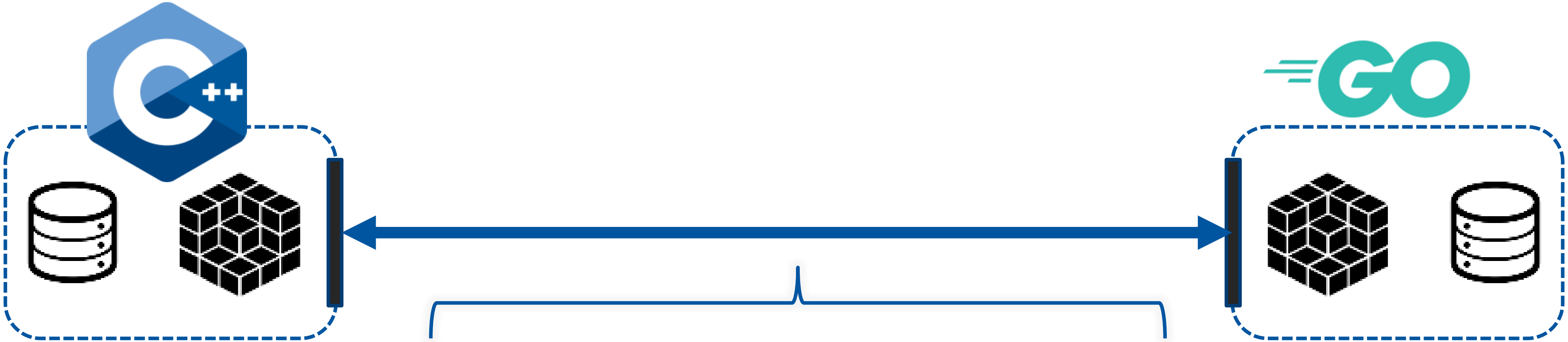
person.cc

The API: Payload and **Transport**



XROOTD

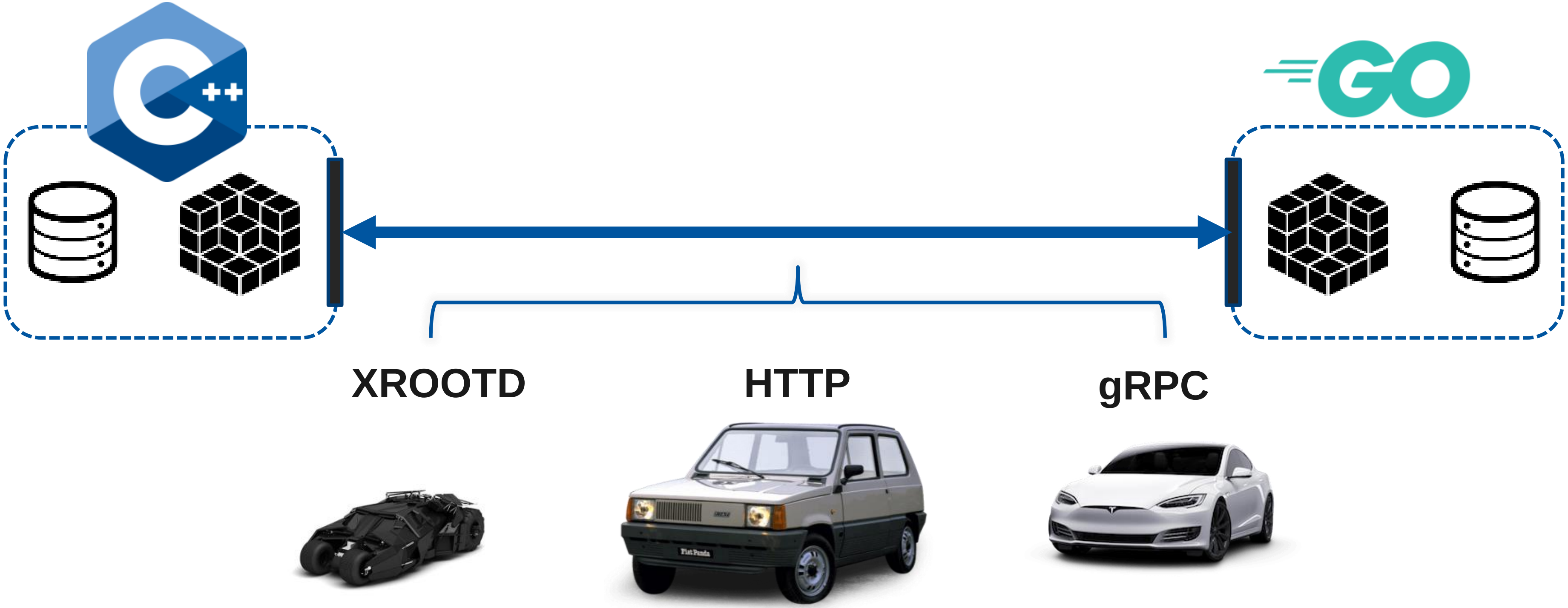
The API: Payload and **Transport**



XROOTD

HTTP

The API: Payload and **Transport**



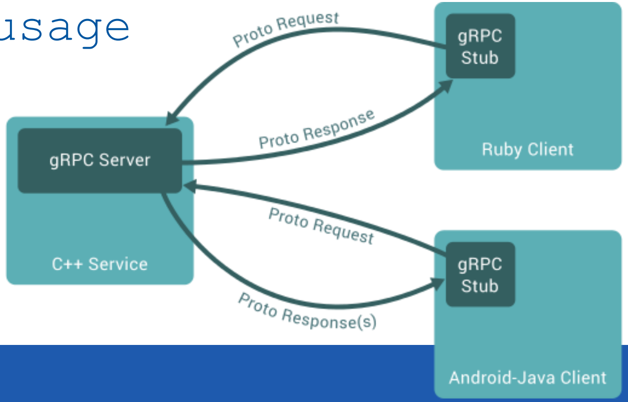
XROOTD

HTTP

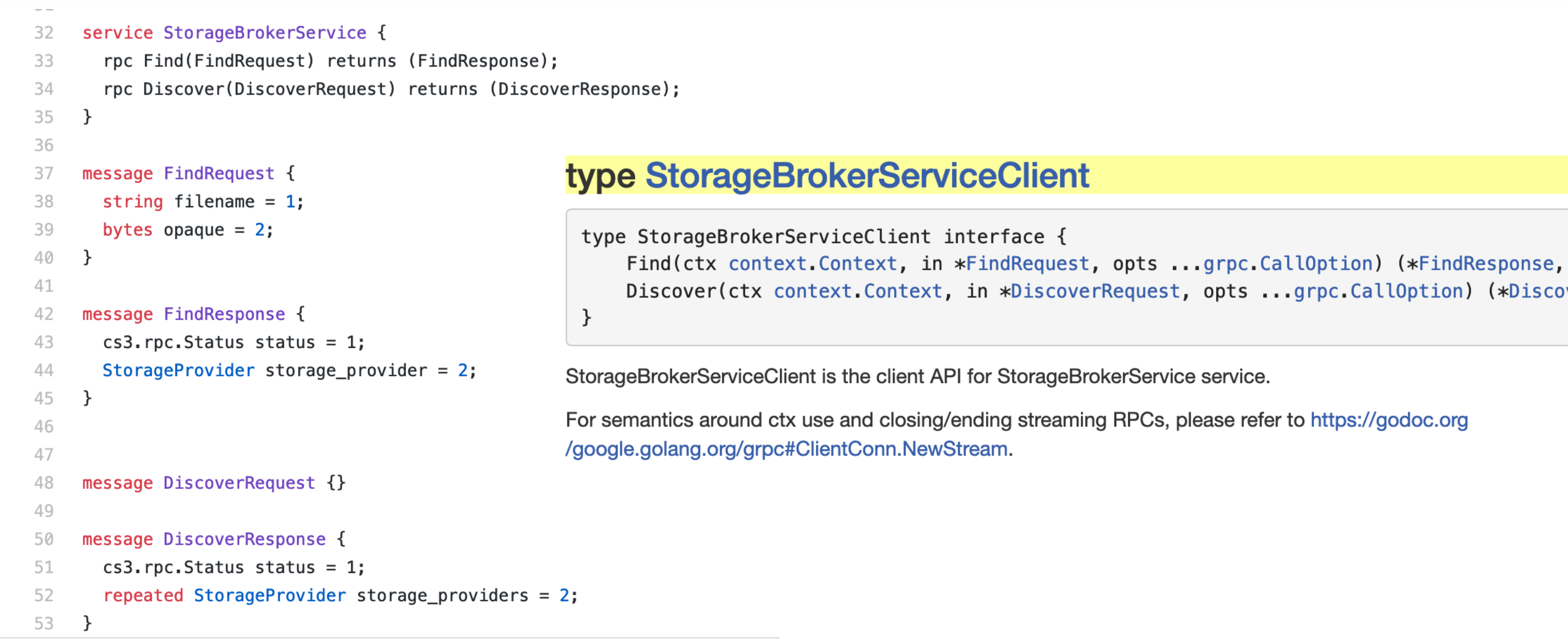
gRPC

Why gRPC?

- Based on http2
- Compressing headers
- Persistent single TCP connections
- Cancellation and timeout contracts between client and server
- Unary and streaming requests
- Requests are multiplexed over a single TCP connection, allowing multiple concurrent messages to be in flight without compromising network resource usage



gRPC + Protobuf = ❤



```
service StorageBrokerService {
  rpc Find(FindRequest) returns (FindResponse);
  rpc Discover(DiscoverRequest) returns (DiscoverResponse);
}

message FindRequest {
  string filename = 1;
  bytes opaque = 2;
}

message FindResponse {
  cs3.rpc.Status status = 1;
  StorageProvider storage_provider = 2;
}


message DiscoverRequest {}

message DiscoverResponse {
  cs3.rpc.Status status = 1;
  repeated StorageProvider storage_providers = 2;
}
```

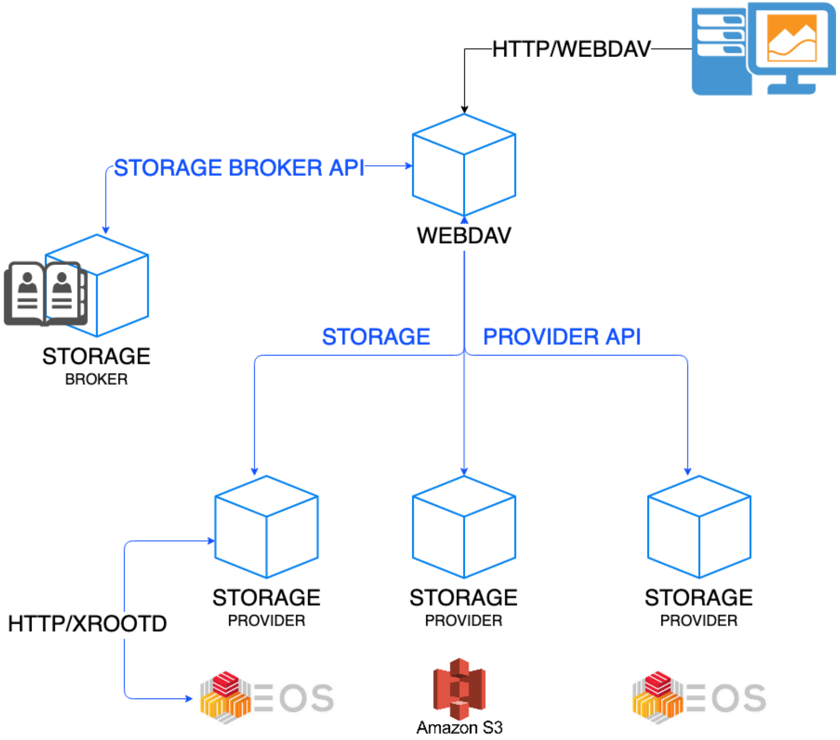
type StorageBrokerServiceClient

```
type StorageBrokerServiceClient interface {
    Find(ctx context.Context, in *FindRequest, opts ...grpc.CallOption) (*FindResponse,
    Discover(ctx context.Context, in *DiscoverRequest, opts ...grpc.CallOption) (*Disco
}
```

StorageBrokerServiceClient is the client API for StorageBrokerService service.

For semantics around ctx use and closing/ending streaming RPCs, please refer to https://godoc.org/google.golang.org/grpc#ClientConn.NewStream.

CERNBox ecosystem



CS3 APIS

Specs on gRPC + Protobuf
Last mile distributed computing
Neutral-vendor APIS
Regain user freedom
Driven by the community

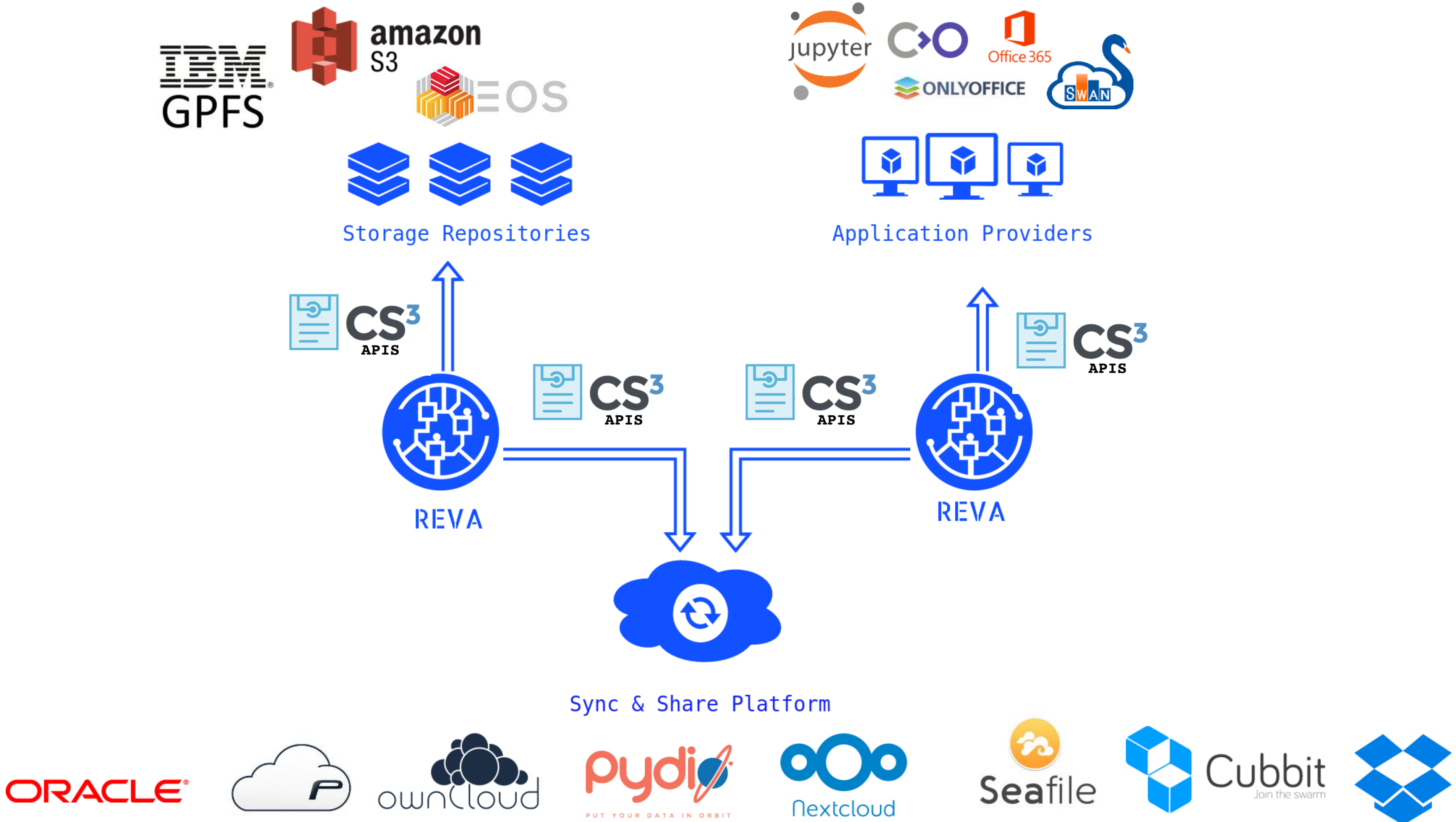
https://github.com/cernbox/cs3apis

REVA

Reference implementation of the CS3 APIs

Modular platform to build on top

https://github.com/cernbox/reva



Storage Repositories

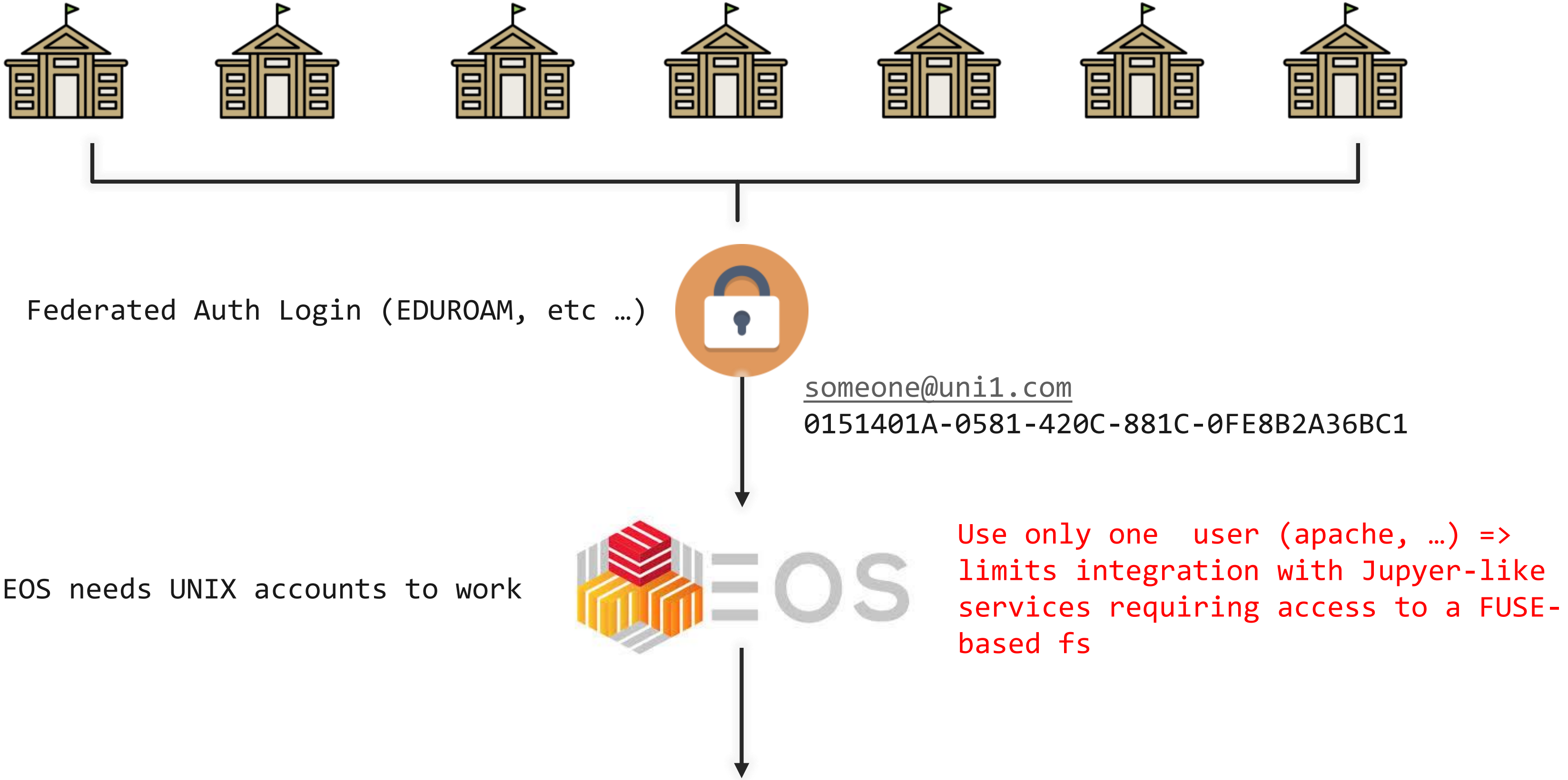
Application Providers

REVA

REVA

Sync & Share Platform

My TODO item for EOS



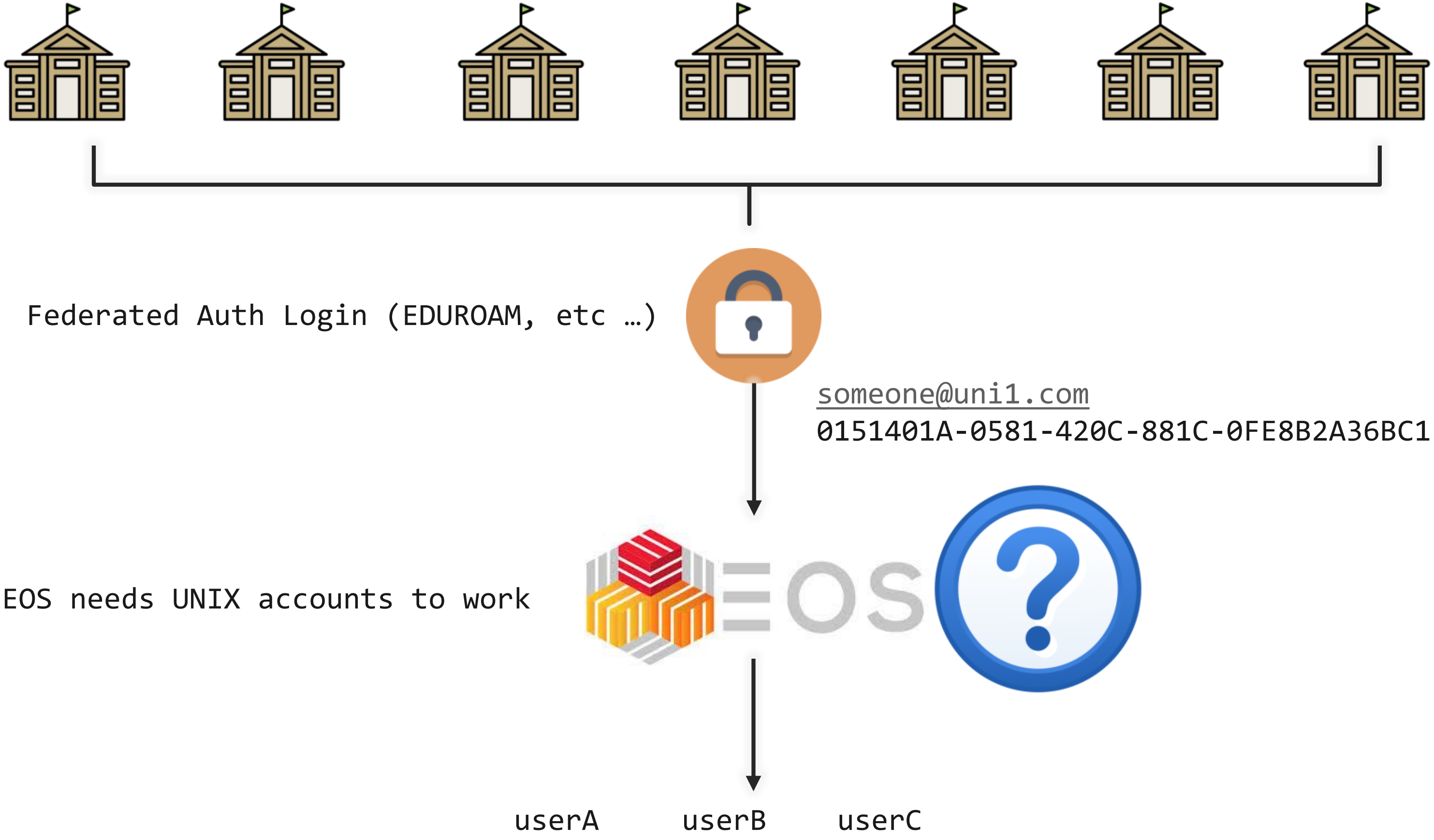
Federated Auth Login (EDUROAM, etc …)

someone@uni1.com
0151401A-0581-420C-881C-0FE8B2A36BC1

EOS needs UNIX accounts to work

Use only one user (apache, …) => limits integration with Jupyer-like services requiring access to a FUSE-based fs

My TODO item for EOS



Federated Auth Login (EDUROAM, etc …)

someone@uni1.com
0151401A-0581-420C-881C-0FE8B2A36BC1

EOS needs UNIX accounts to work

userA userB userC

Merci!