



C++ ZeroCost Conf ^{*/}



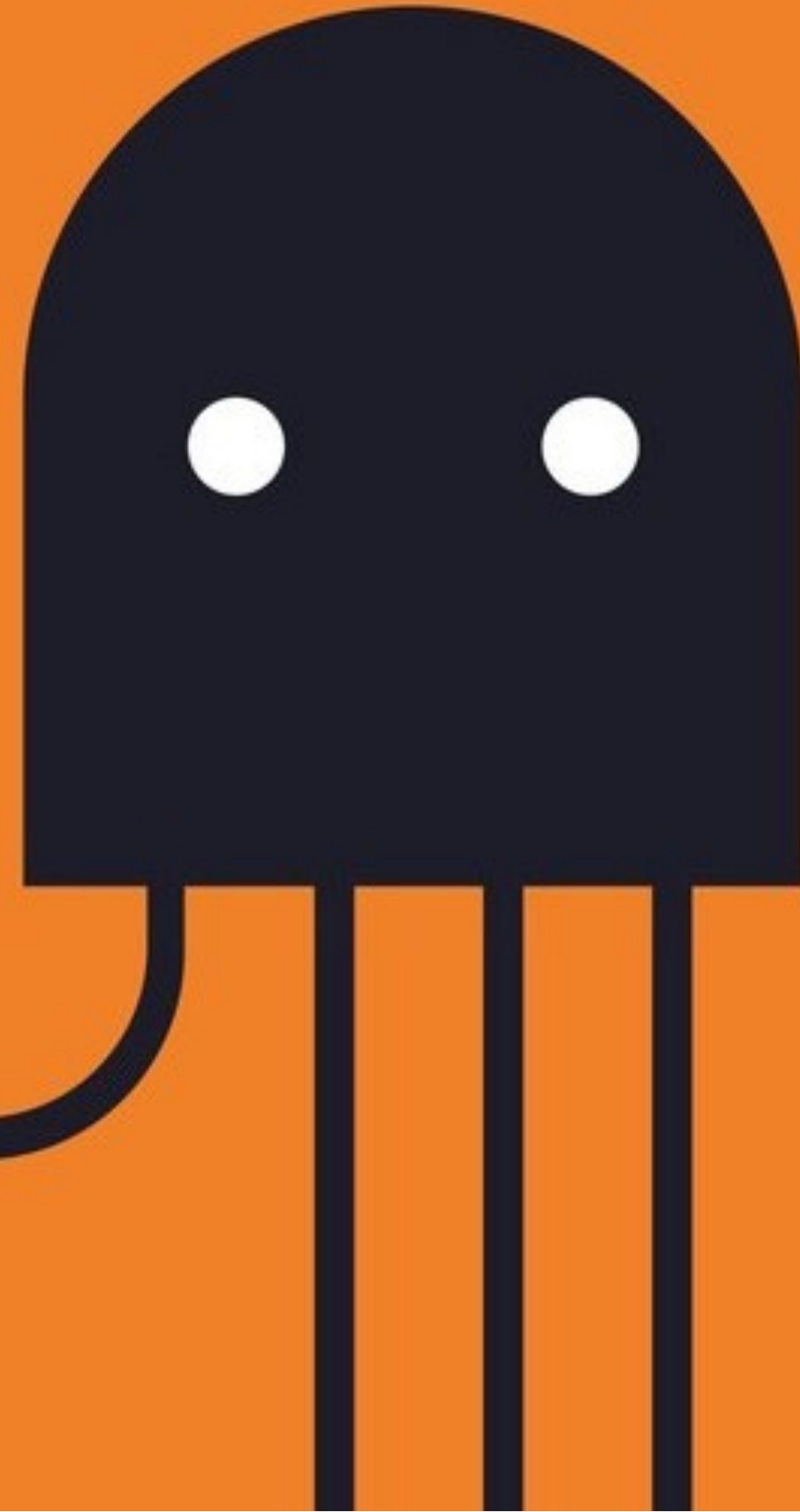
Секретная новость

Полухин Антон
Эксперт разработчик C++



Yandex for **developers** `*//>`

<https://userver.tech/>



Простота использования,
эффективность,
изкоробочность

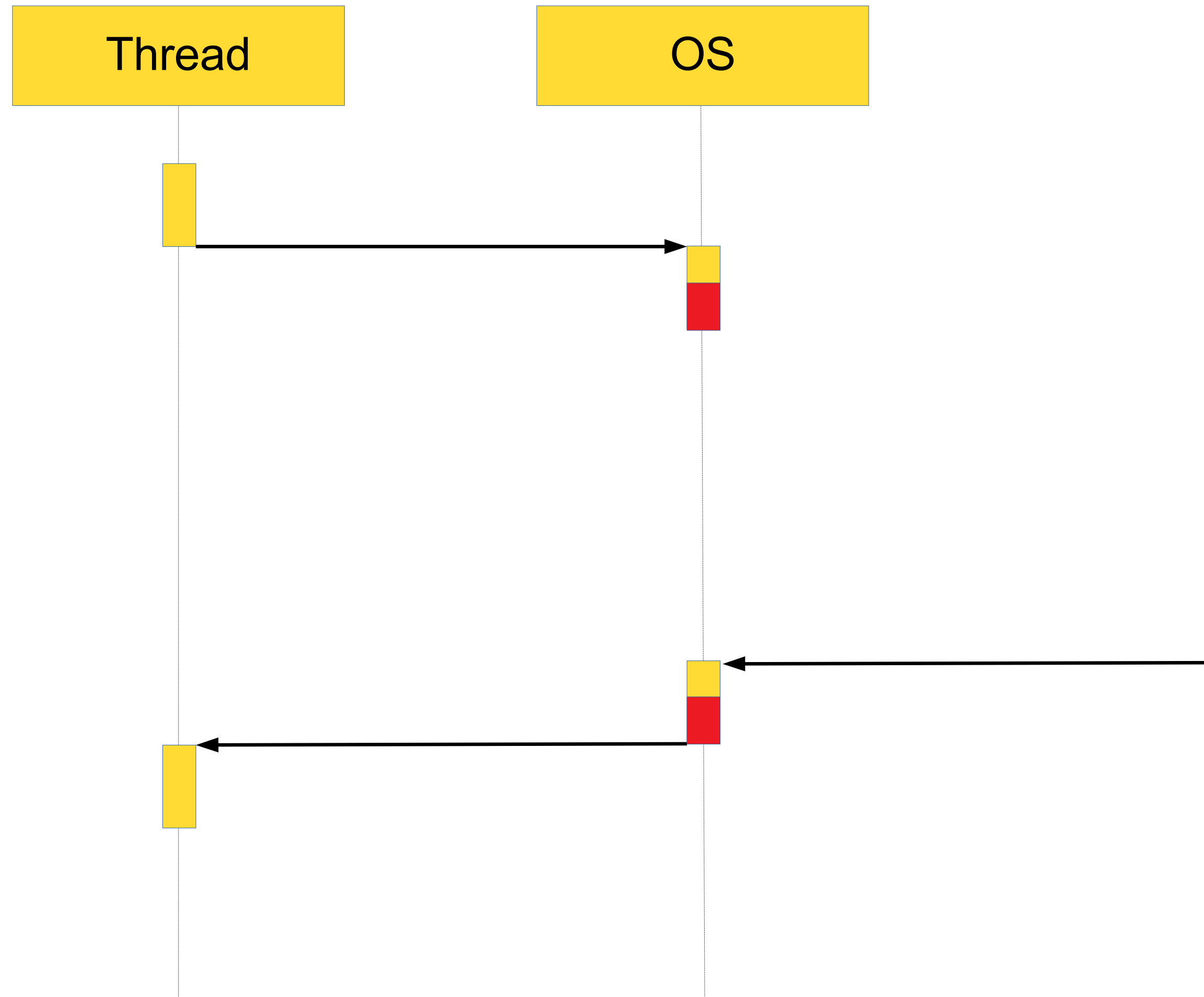
userver

```
ResponseView::Handle(Request&& request, const Dependencies&dependencies) {  
    auto cluster = dependencies.pg->GetCluster(); // ⚡  
    auto trx = cluster->Begin(storages::postgres::ClusterHostType::kMaster); // ⚡  
  
    constchar* statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";  
    auto row = psql::Execute(trx, statement, request.id)[0]; // ⚡  
    if (!row["ok"].As<bool>()) {  
        LOG_DEBUG() << request.id << " is not OK of "  
            << GetSomeInfoFromDb(); // ⚡  
        return Response400();  
    }  
  
    psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar); // ⚡  
    trx.Commit(); // ⚡  
  
    return Response200{row["baz"].As<std::string>()};  
}
```

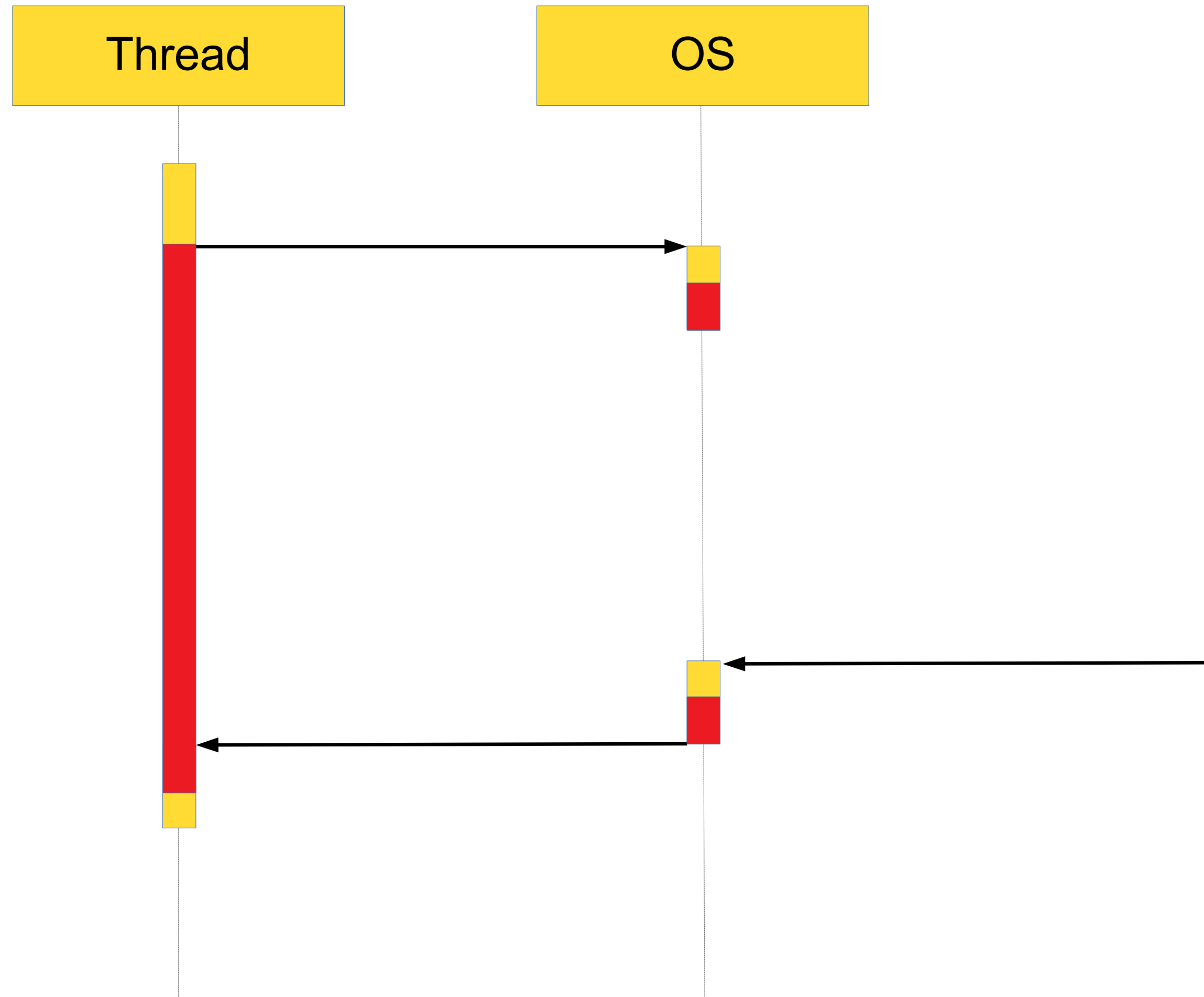
userver

```
ResponseView::Handle(Request&& request, const Dependencies&dependencies) {  
    auto cluster = dependencies.pg->GetCluster(); // ⚡  
    auto trx = cluster->Begin(storages::postgres::ClusterHostType::kMaster); // ⚡  
  
    constchar* statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";  
    auto row = psql::Execute(trx, statement, request.id)[0]; // ⚡  
    if (!row["ok"].As<bool>()) {  
        LOG_DEBUG() << request.id << " is not OK of "  
            << GetSomeInfoFromDb(); // ⚡  
        return Response400();  
    }  
  
    psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar); // ⚡  
    trx.Commit(); // ⚡  
  
    return Response200{row["baz"].As<std::string>()};  
}
```

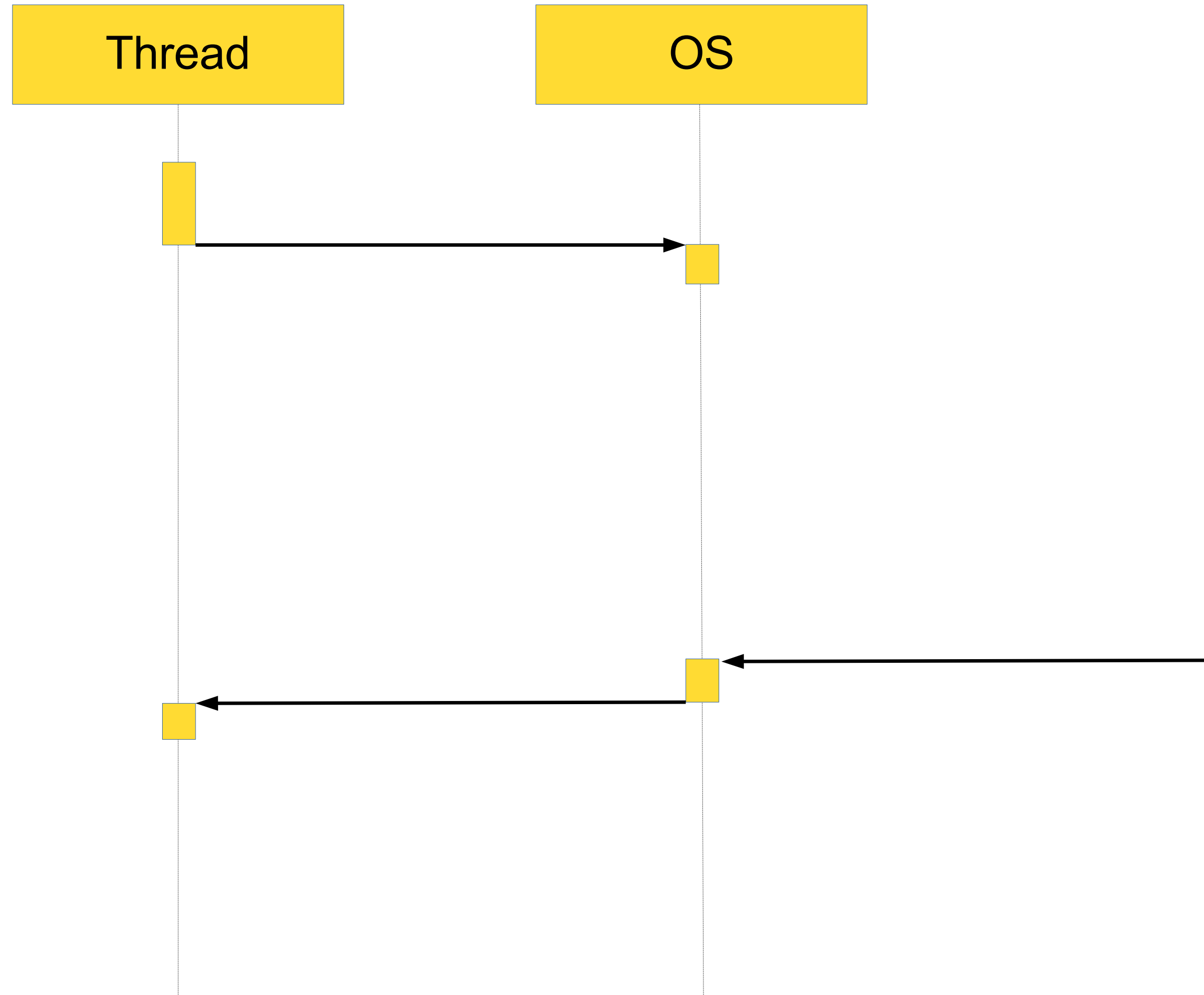
Синхронный подход (не usegver!)



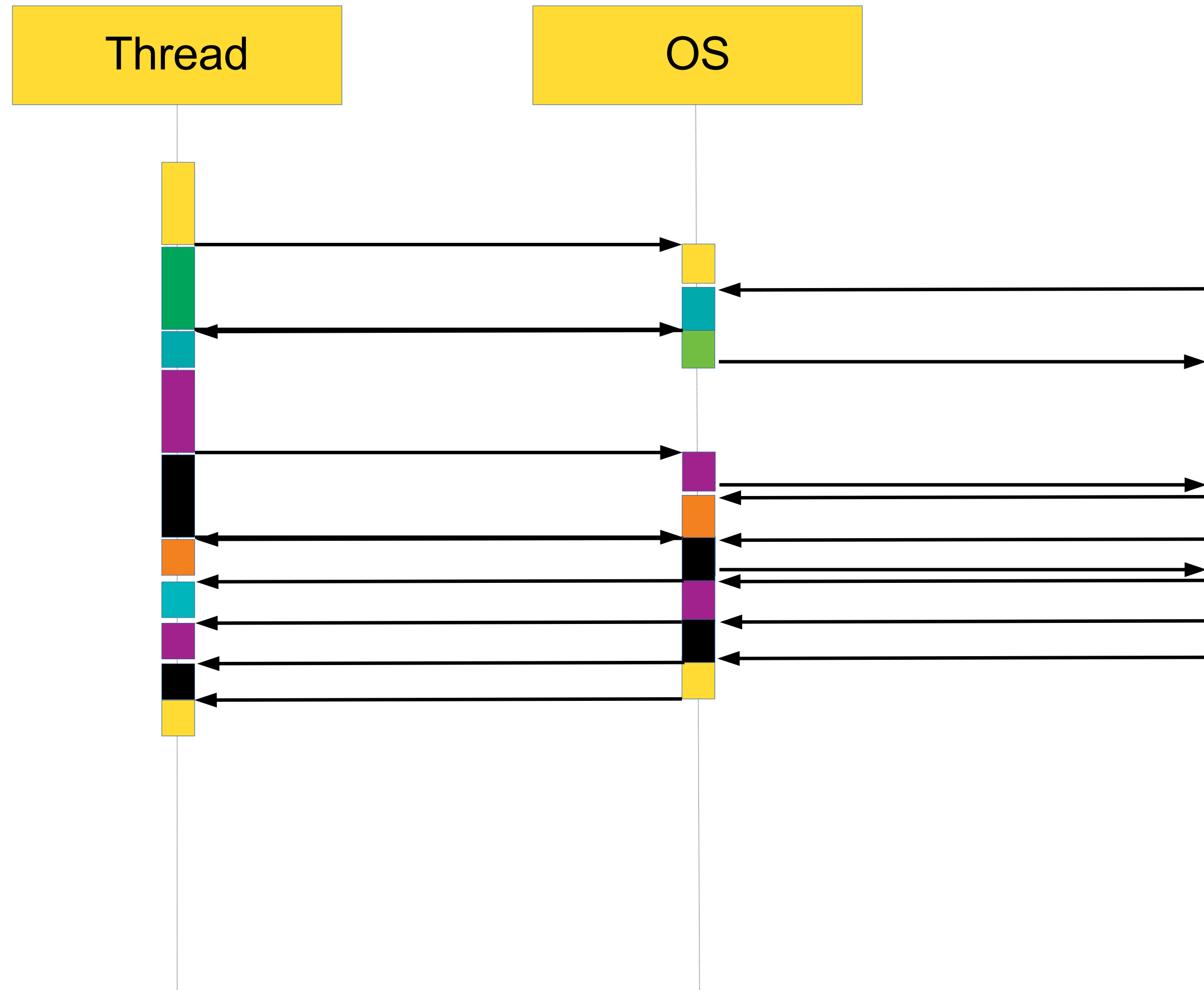
Синхронный подход (не usegver!)



userver



userver



userver

```
ResponseView::Handle(Request&& request, const Dependencies&dependencies) {
    auto cluster = dependencies.pg->GetCluster();
    auto trx = cluster->Begin(storages::postgres::ClusterHostType::kMaster);

    const char* statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";
    auto row = psql::Execute(trx, statement, request.id)[0];
    if (!row["ok"].As<bool>()) {
        LOG_DEBUG() << request.id << " is not OK of "
            << GetSomeInfoFromDb();
        return Response400();
    }

    psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar);
    trx.Commit();

    return Response200{row["baz"].As<std::string>()};
}
```

Фичи

Фичи

- PostgreSQL

Фичи

- PostgreSQL
- Mongo

Фичи

- PostgreSQL
- Mongo
- Clickhouse

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics
- Subprocesses

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics
- Subprocesses
- Rcu, Queue, Subscriptions, MutexSet

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics
- Subprocesses
- Rcu, Queue, Subscriptions, MutexSet
- Deadlines / Timeouts

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics
- Subprocesses
- Rcu, Queue, Subscriptions, MutexSet
- Deadlines / Timeouts
- Caches and cache dumps

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics
- Subprocesses
- Rcu, Queue, Subscriptions, MutexSet
- Deadlines / Timeouts
- Caches and cache dumps
- Decimal, FastPimpl, Containers

Фичи

- PostgreSQL
- Mongo
- Clickhouse
- Redis
- Socket, TLS
- Mutex/ConditionVariable/Semaphore...
- Logs
- Dynamic Configs
- HTTP server/client
- gRPC
- Periodic Tasks
- DistLocks
- Unit testing
- Functional Testing (Testuite)
- Formats (JSON, YAML, BSON, ...)
- Tracing, Metrics
- Subprocesses
- Rcu, Queue, Subscriptions, MutexSet
- Deadlines / Timeouts
- Caches and cache dumps
- Decimal, FastPimpl, Containers
- Stacktraces, PFR

Спасибо

Полухин Антон

Эксперт-разработчик C++



antoshkka@gmail.com



antoshkka@yandex-team.ru

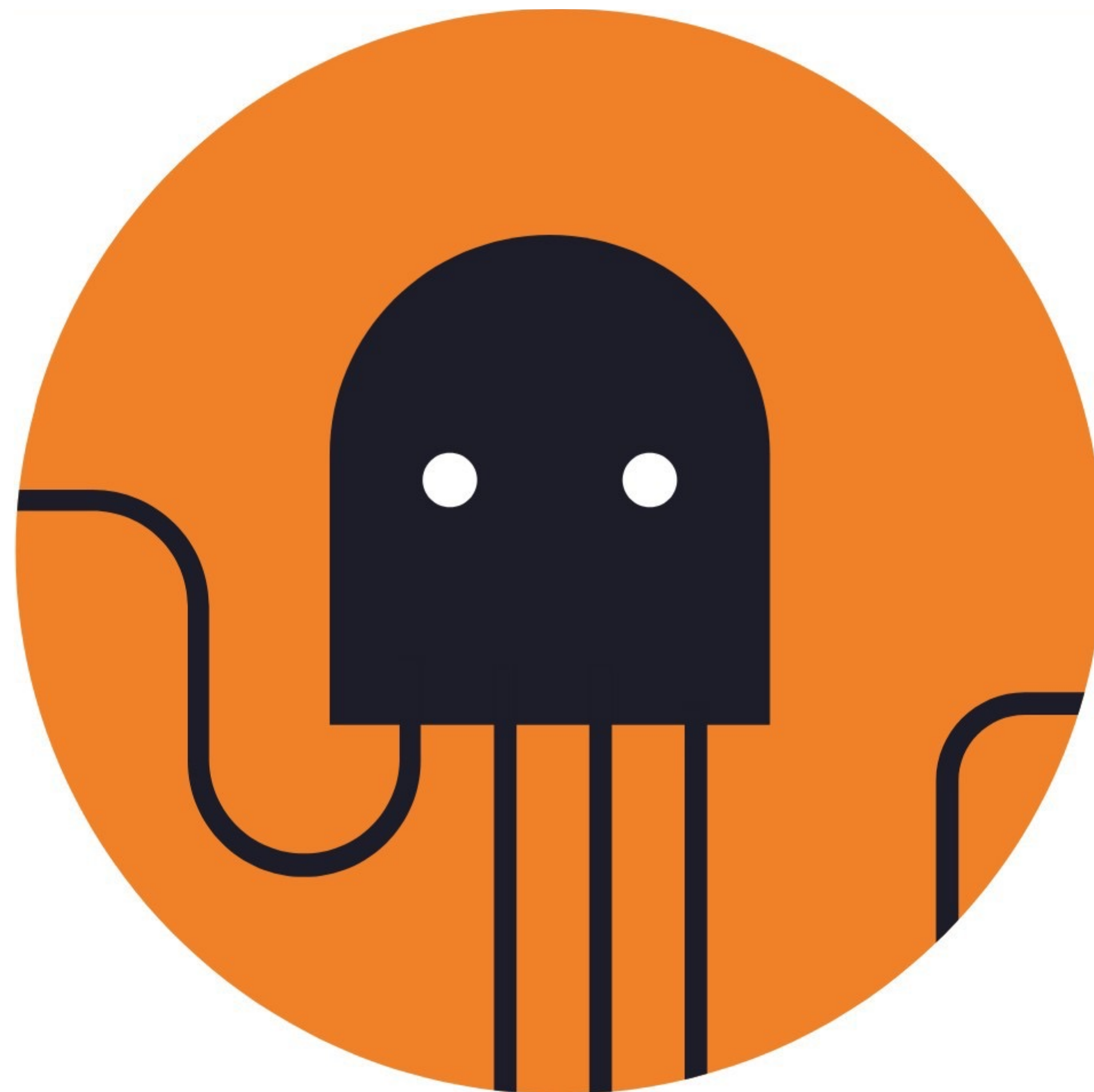


<https://github.com/apolukhin>



РГ21 C++ РОССИЯ

<https://stdcpp.ru/>



<https://github.com/userver-framework>