Микросервисы или нет

+ немного про кеши, балансеры и C++

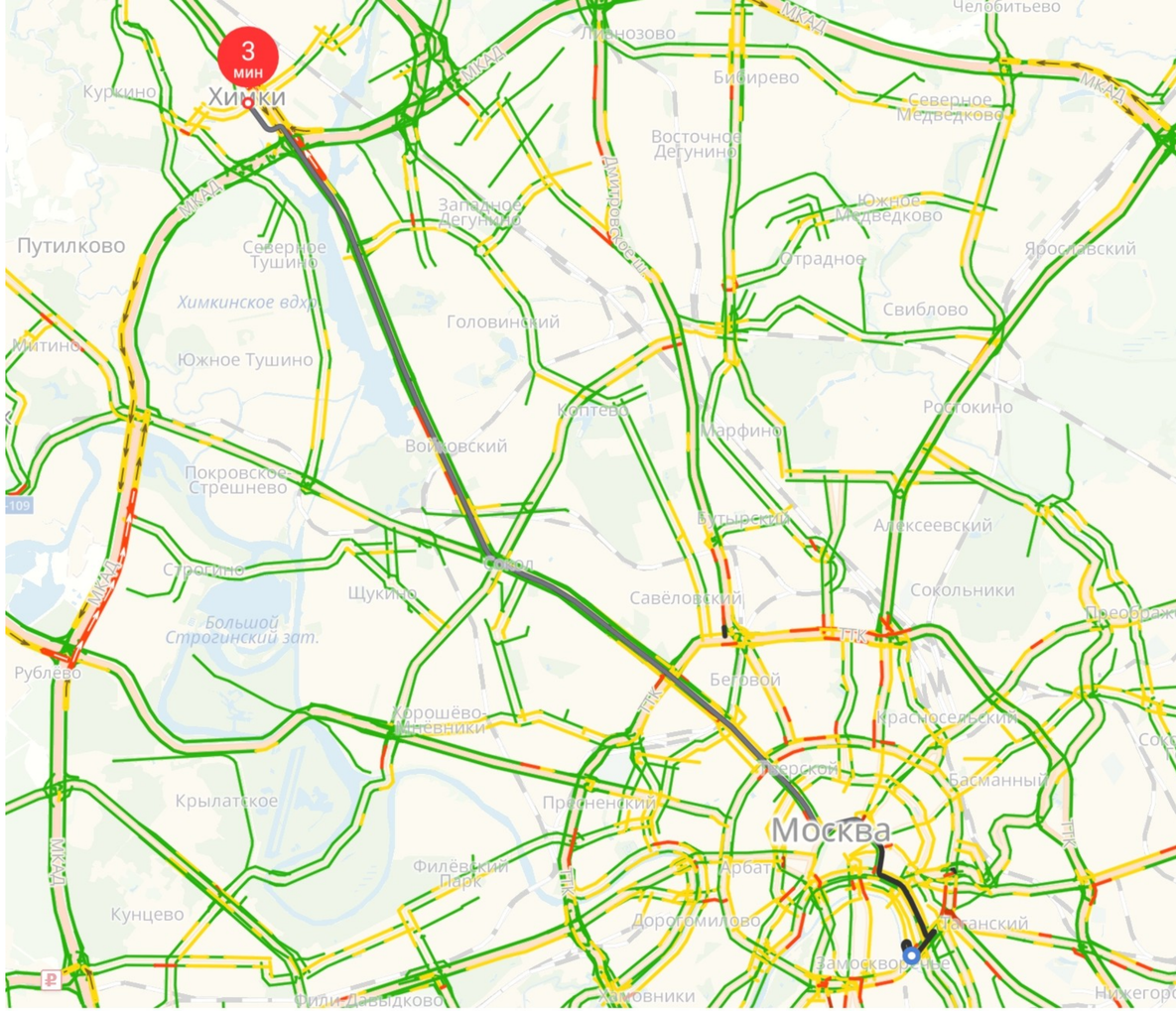
Полухин Антон

Antony Polukhin

Яндекс Go

Содержание

- Архитектуры
  - Лучшая архитектура!
  - Монолит
  - Правильный микросервис
- Балансеры
  - Классика
  - Service Mesh
- Кеши
- IO-bound и C++



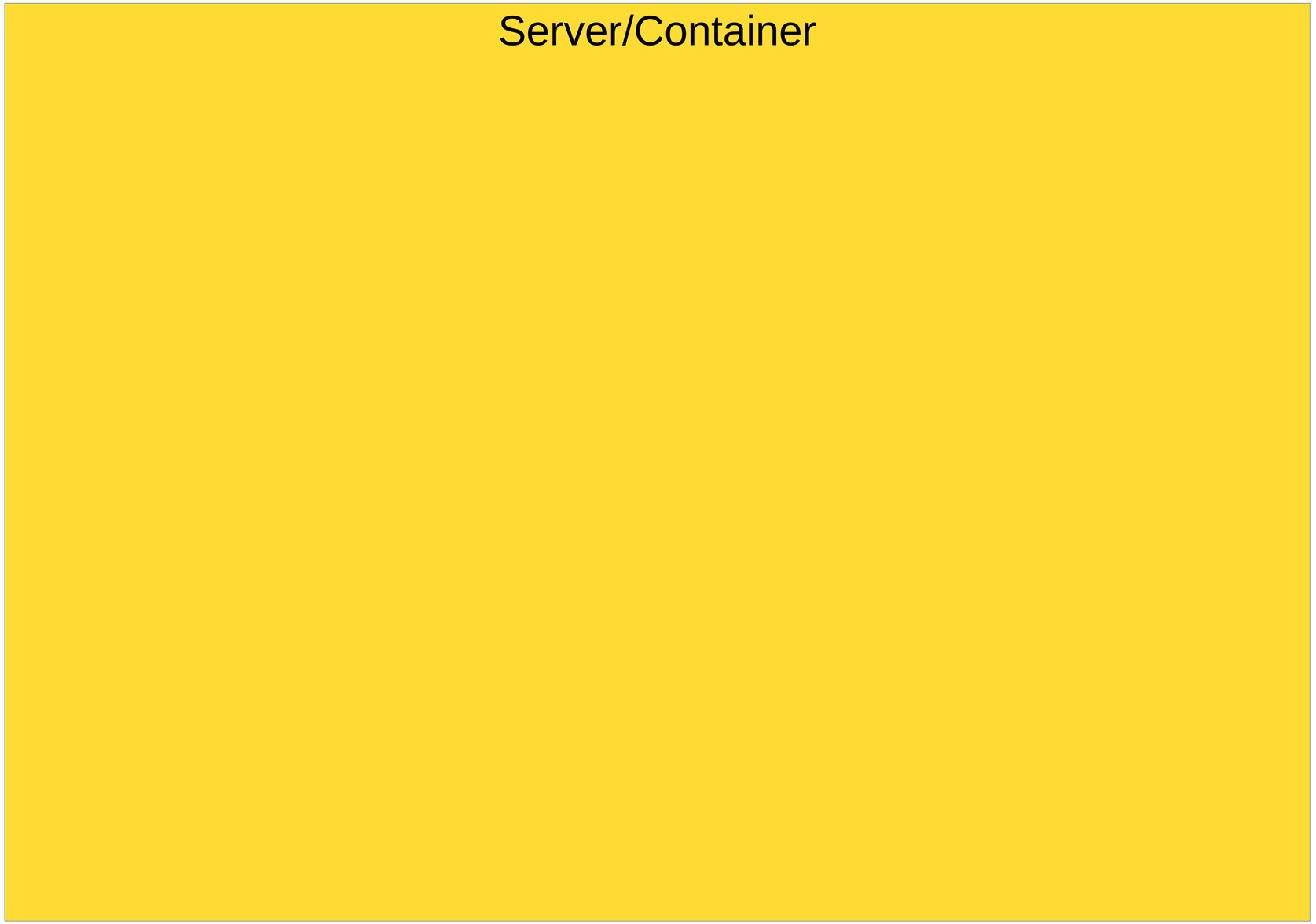
Микросервисы или нет

Самая лучшая архитектура это...

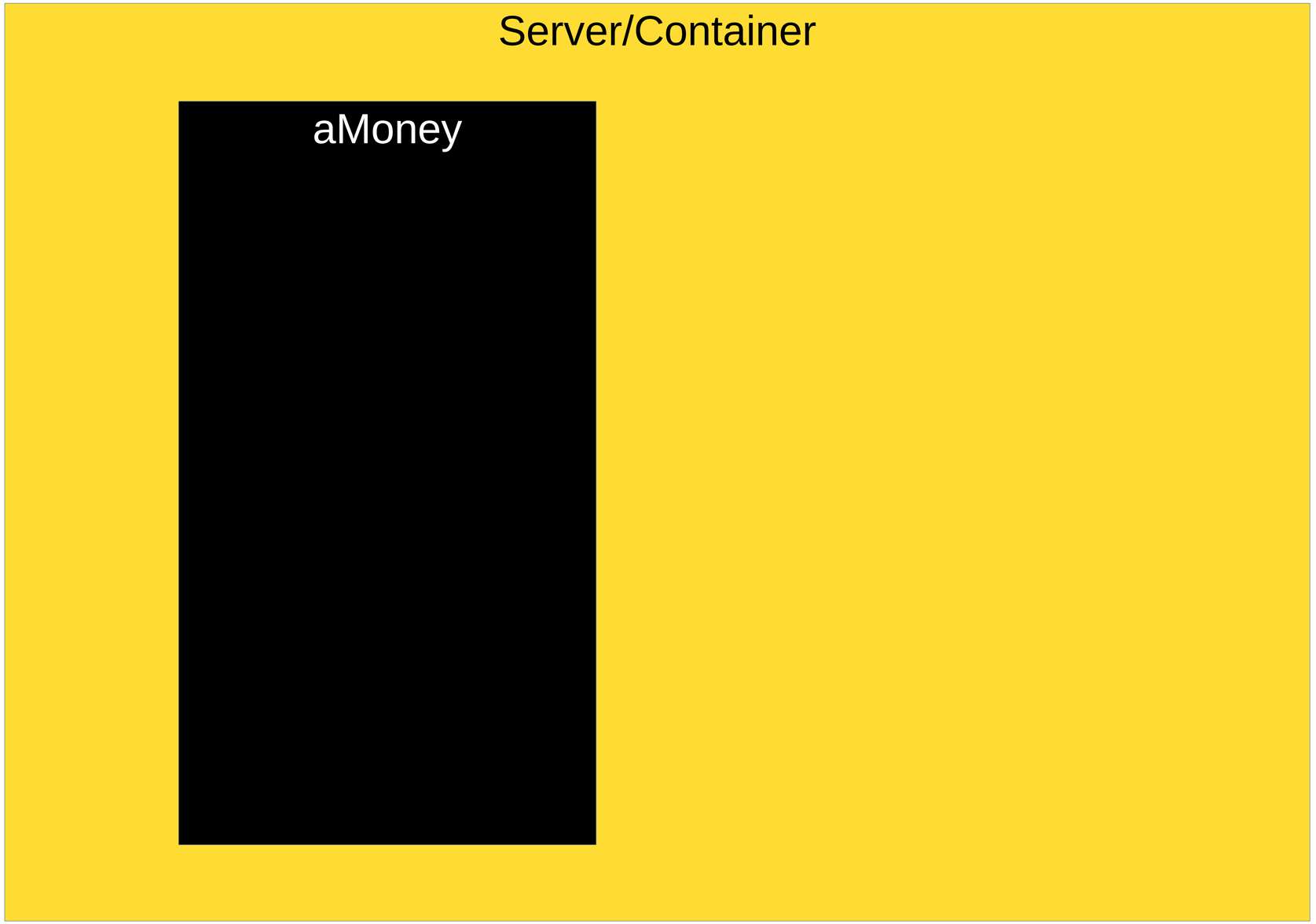
...та, которая ВАМ удобна!

Монолит

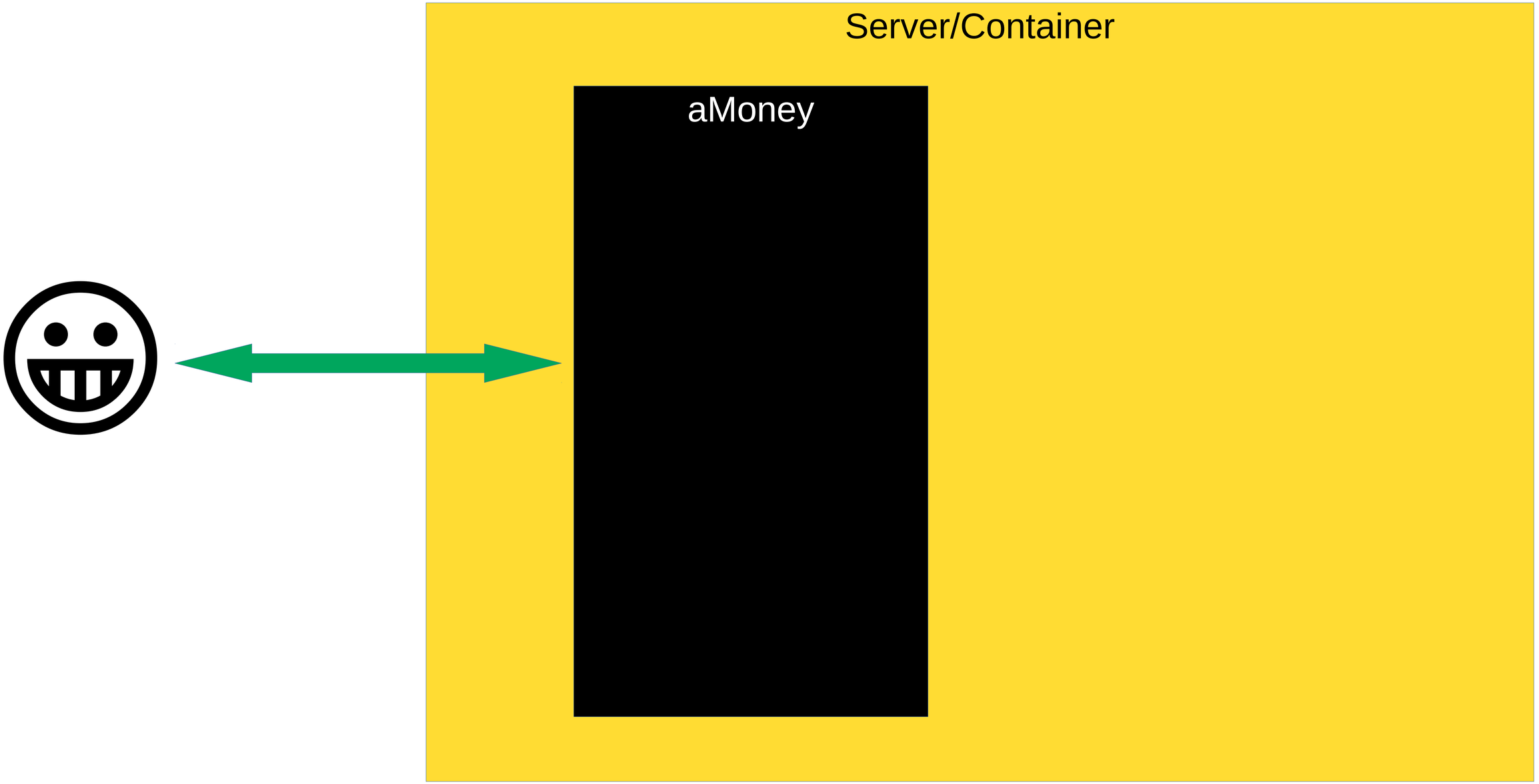
Монолит



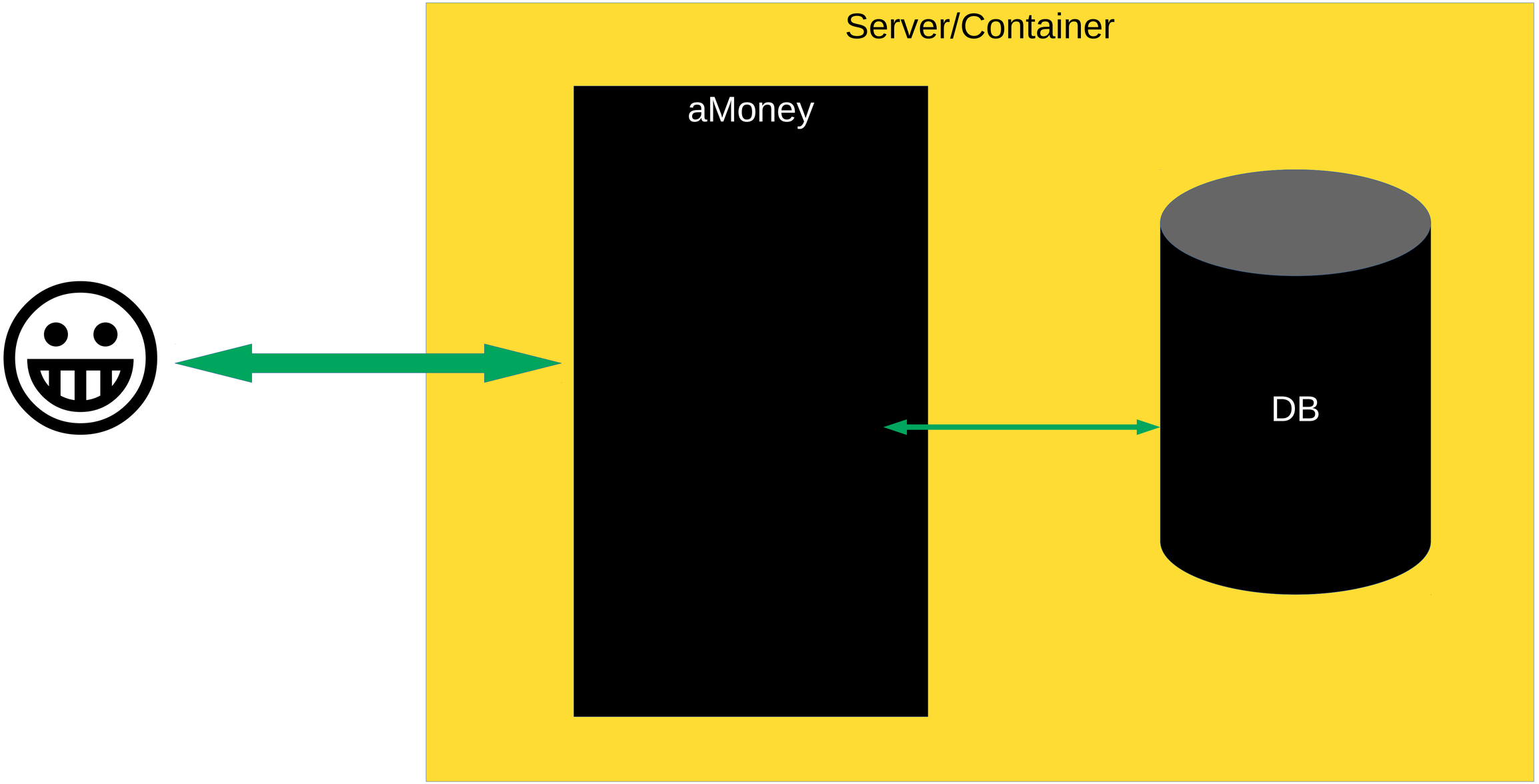
Монолит



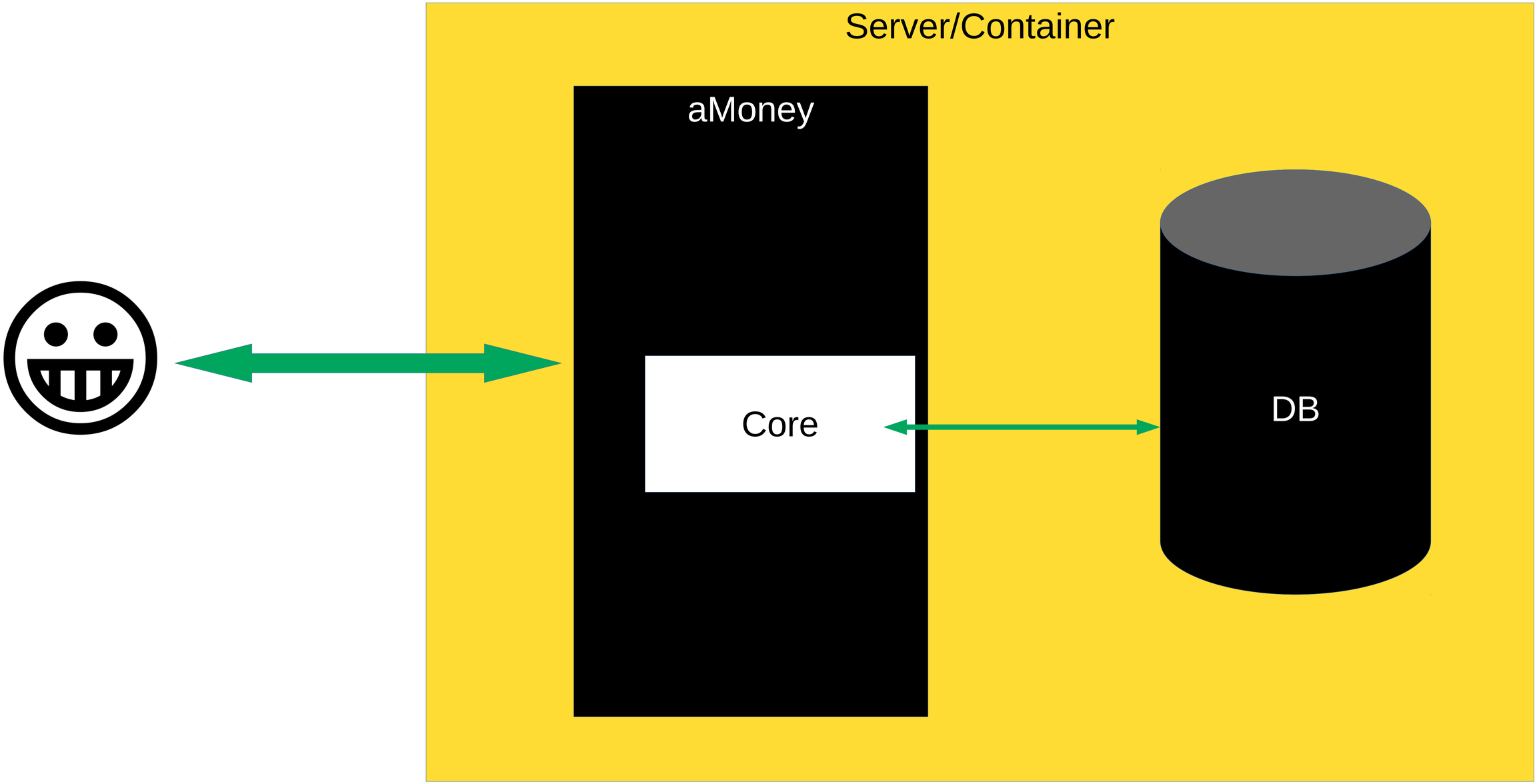
Монолит



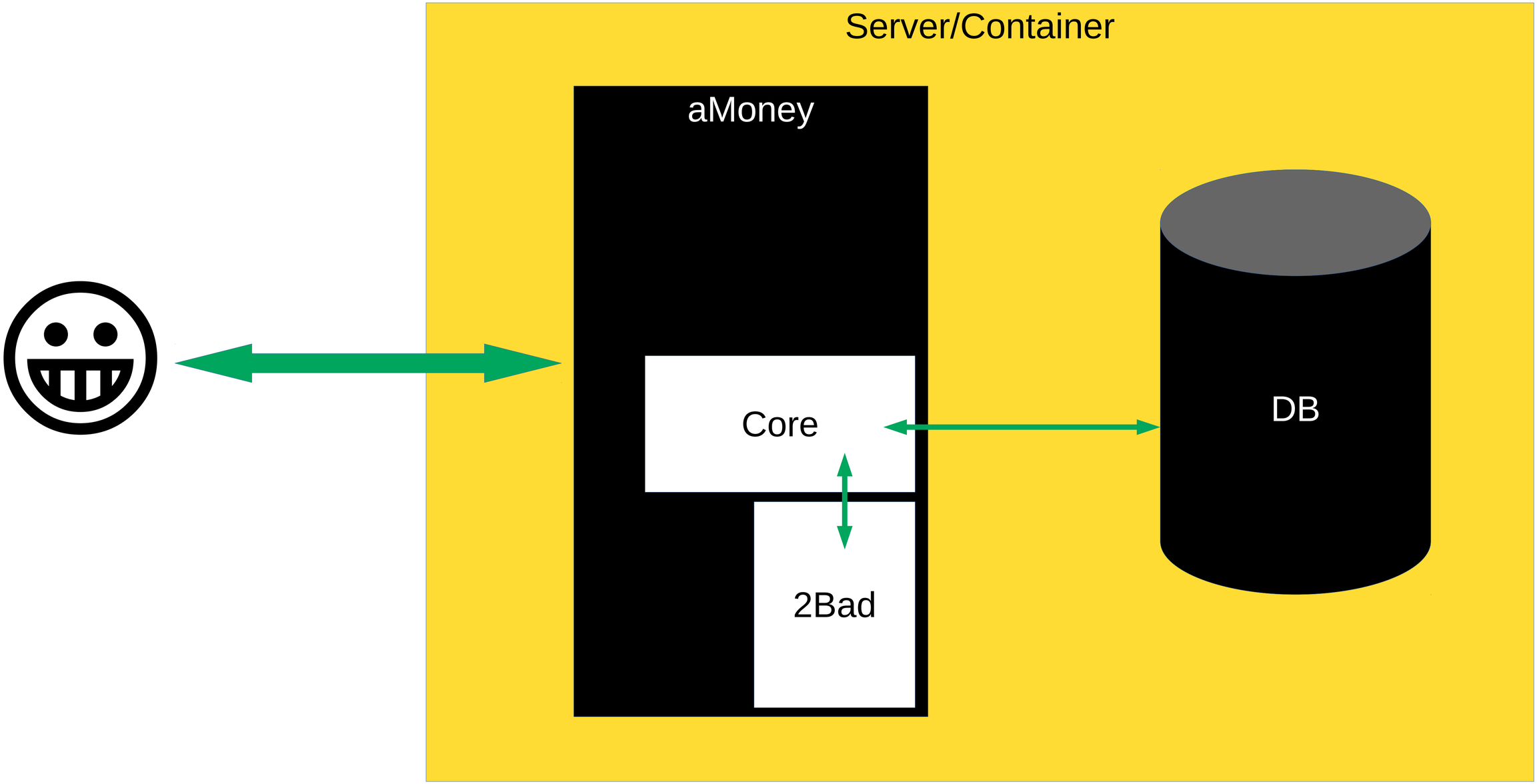
Монолит



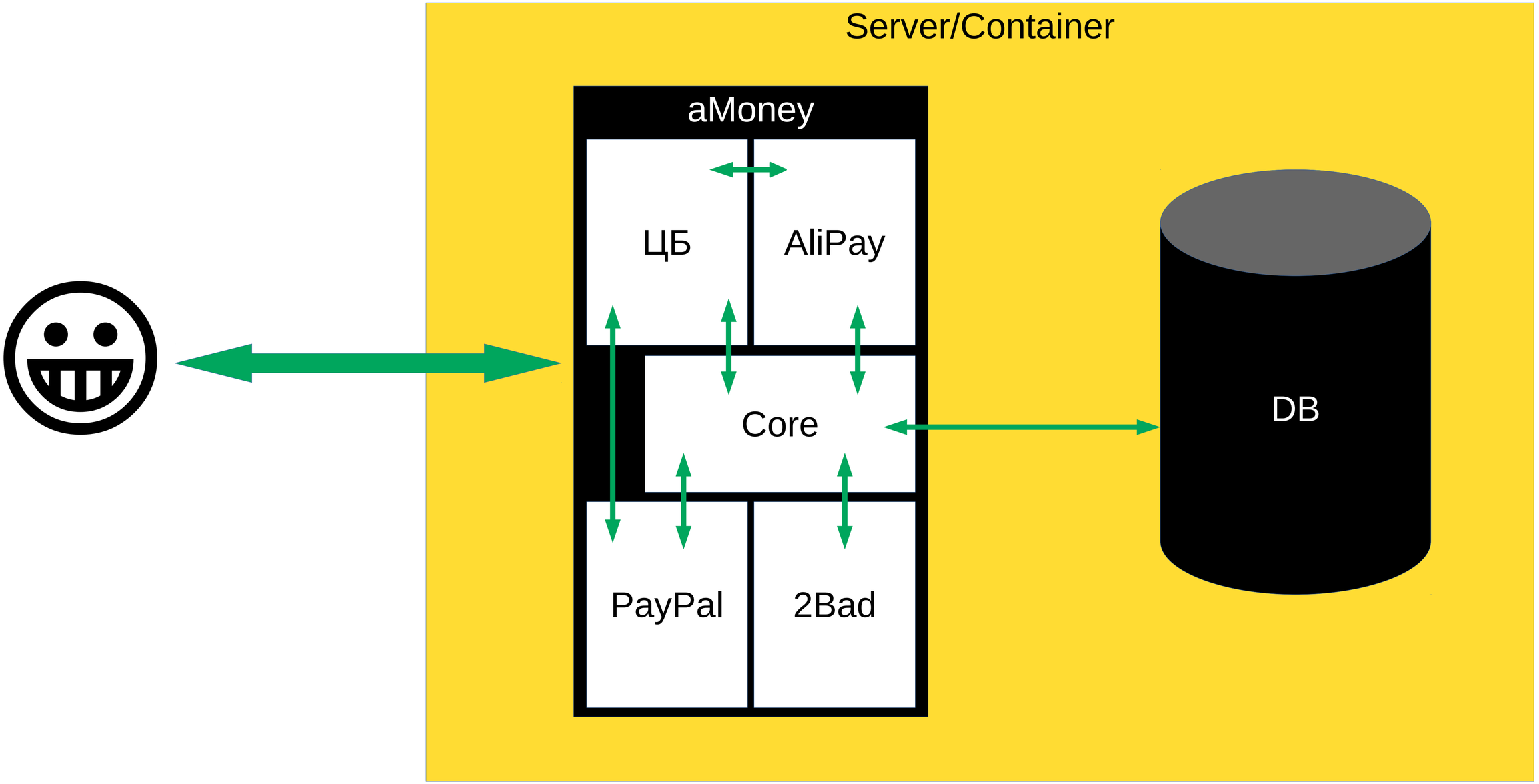
Монолит



Монолит



Монолит



Плюсы/минусы для небольшой команды

Плюсы/минусы для небольшой команды

Плюсы:

Плюсы/минусы для небольшой команды

Плюсы:

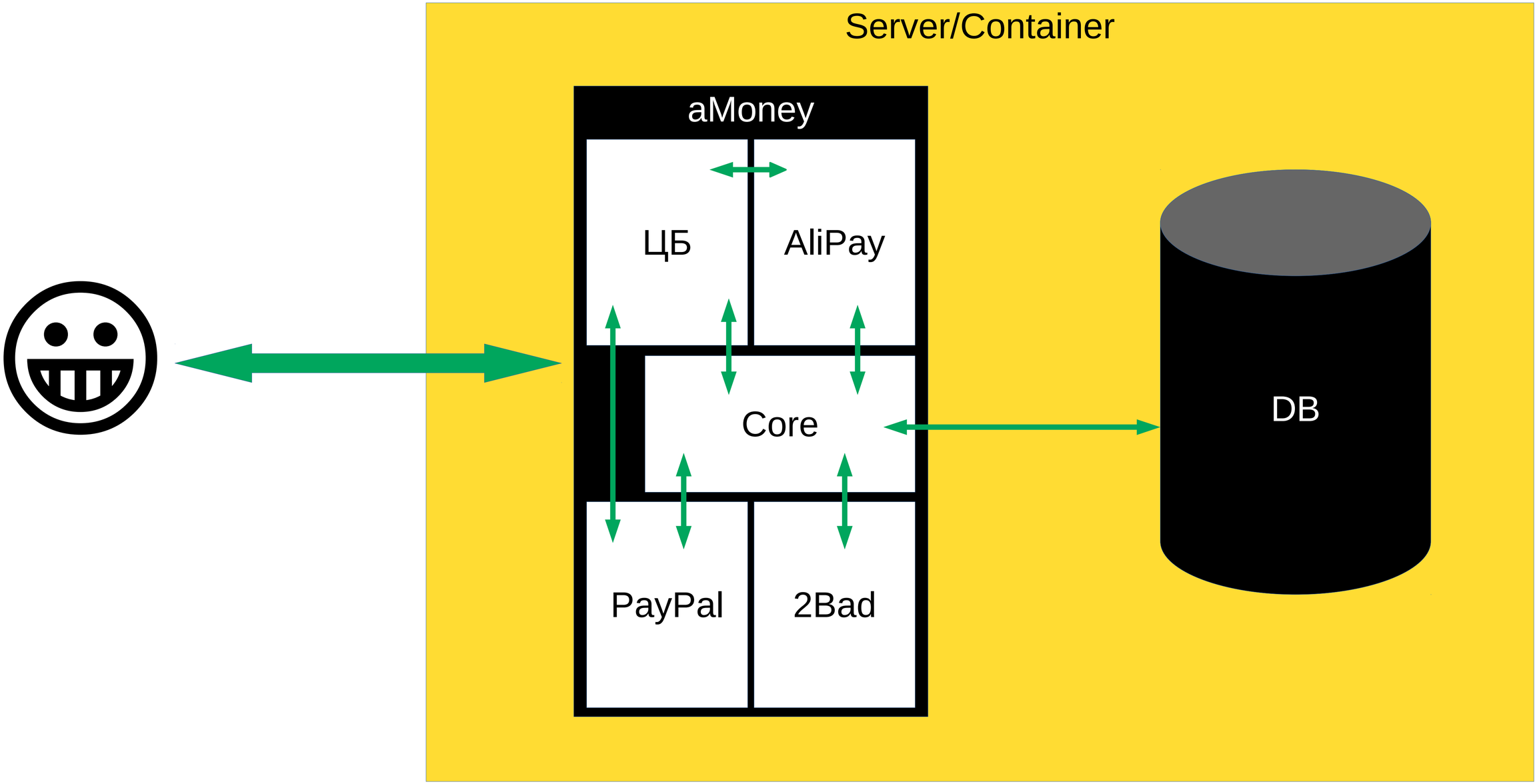
- Простой деплой

Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями

Монолит



Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями

Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями
- Тесное общение между разработчиками

Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями
- Тесное общение между разработчиками

Минусы:

Плюсы/минусы для небольшой команды

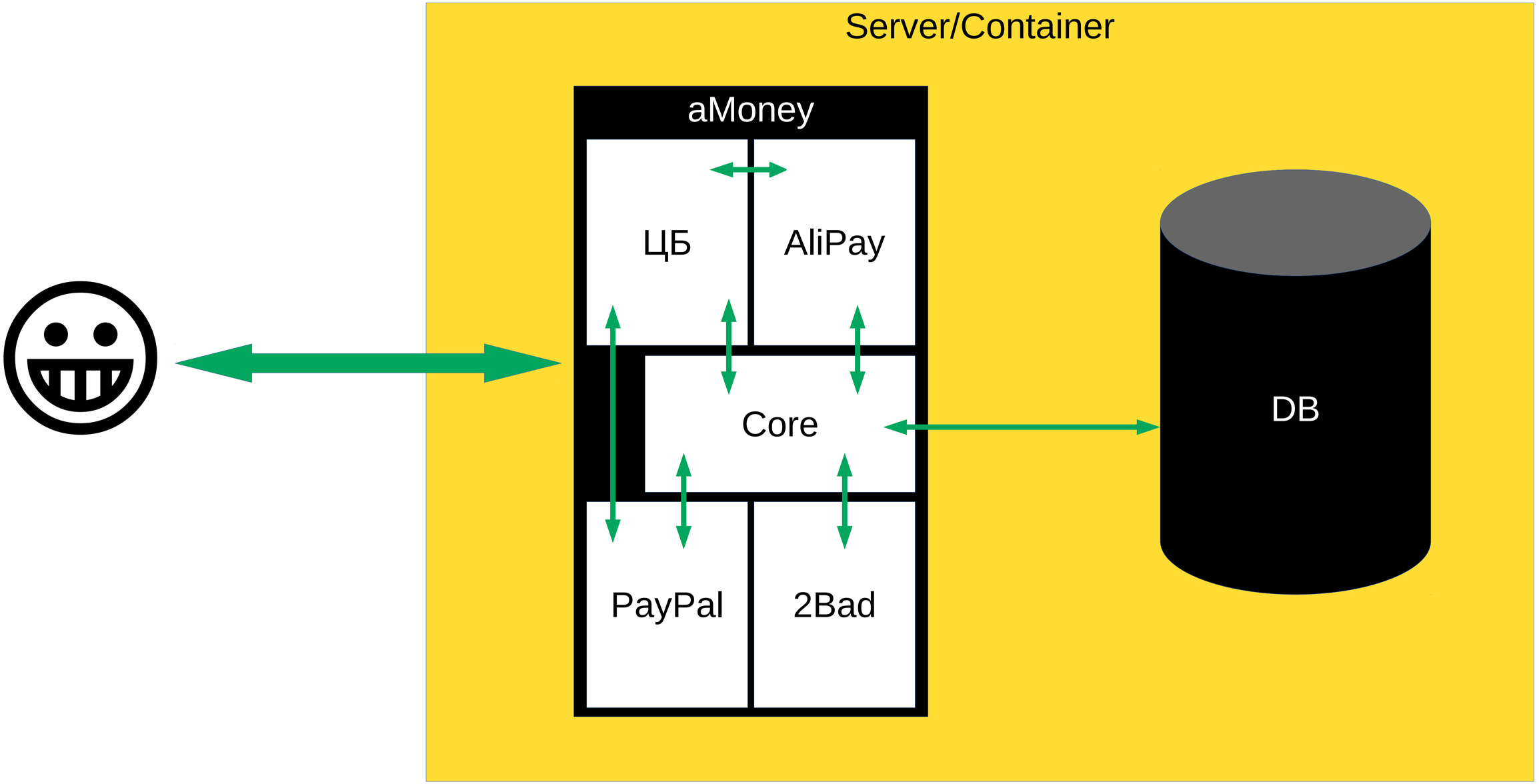
Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями
- Тесное общение между разработчиками

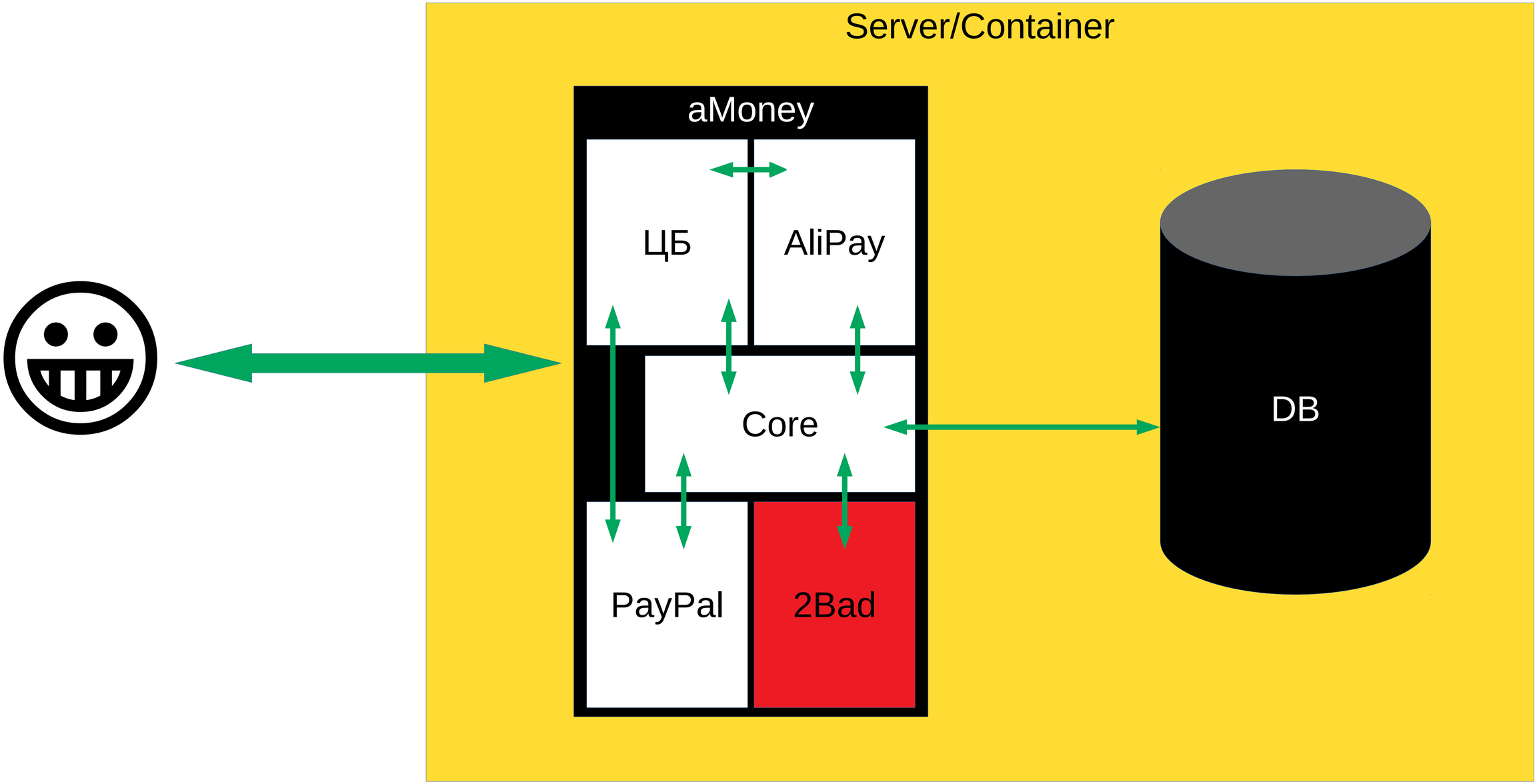
Минусы:

- Не идеальная надёжность

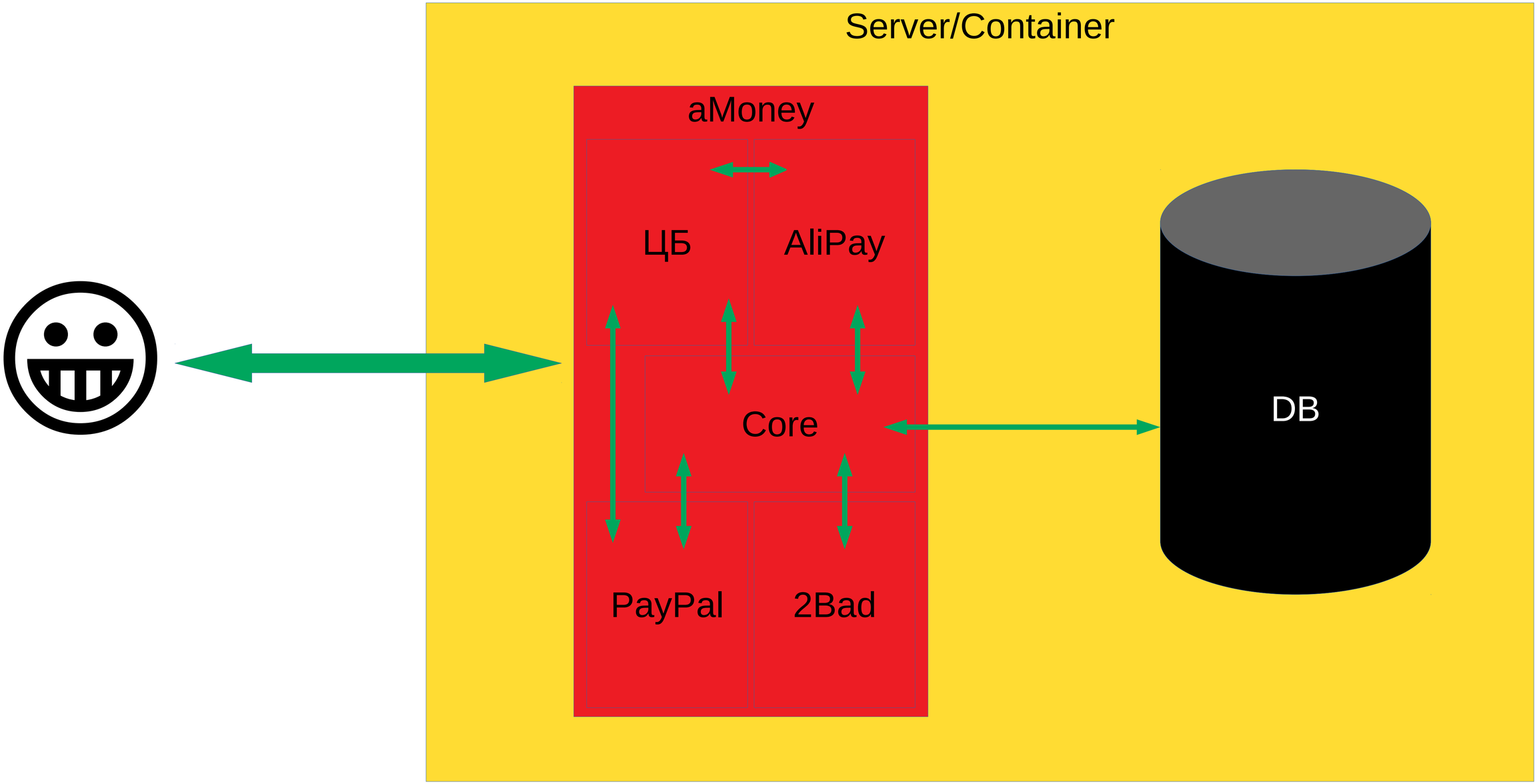
Монолит



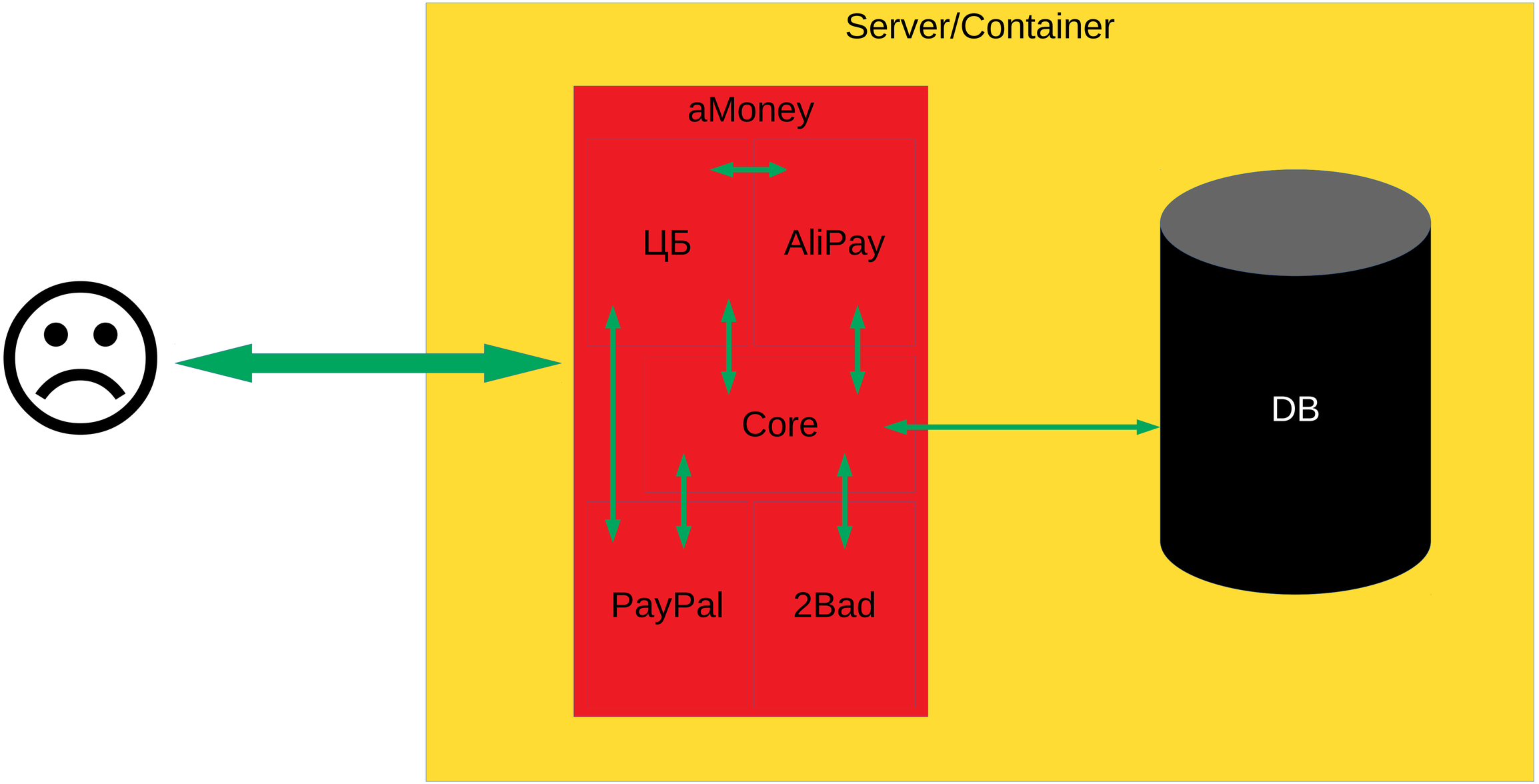
Монолит



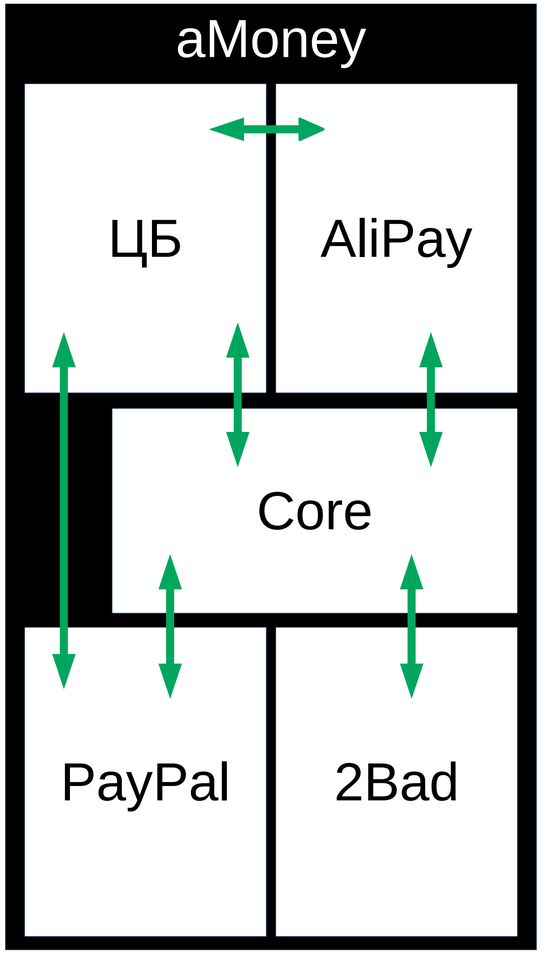
Монолит

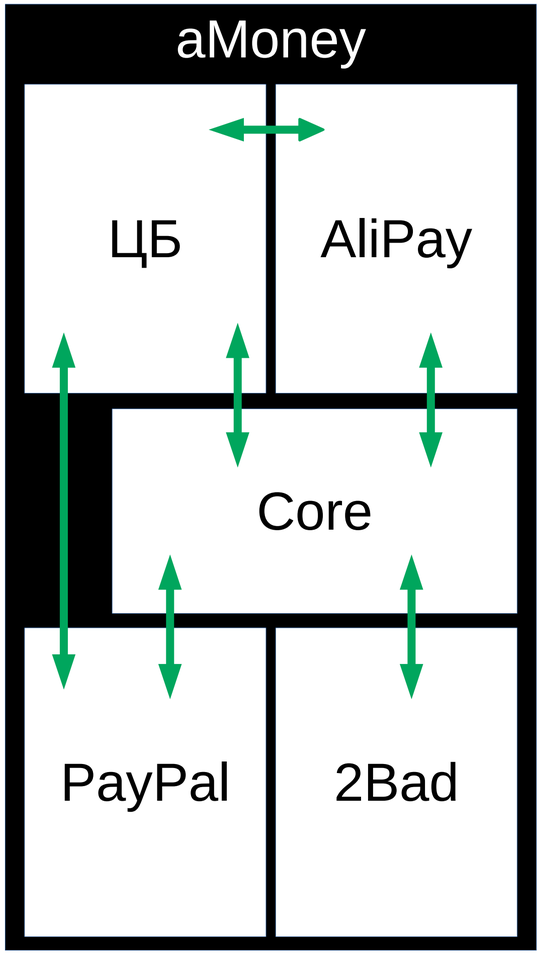


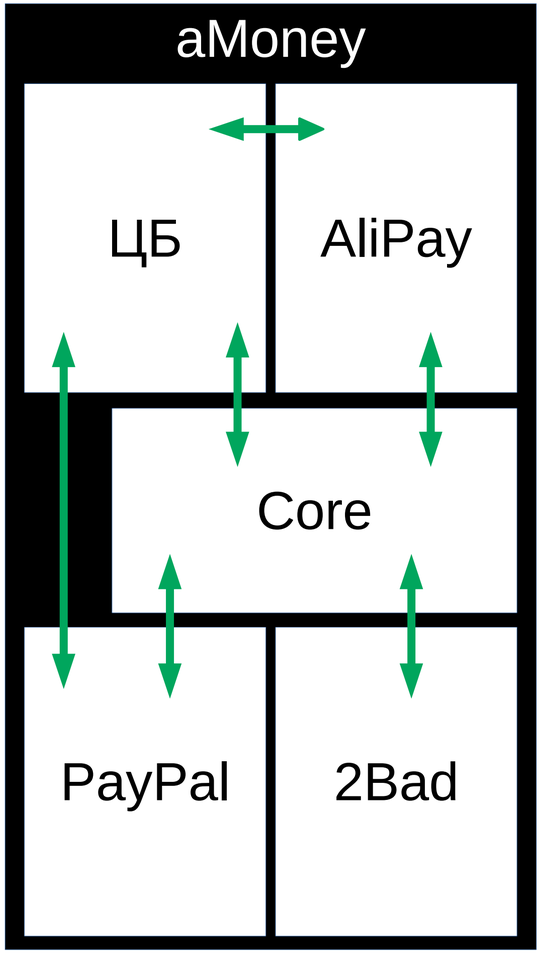
Монолит



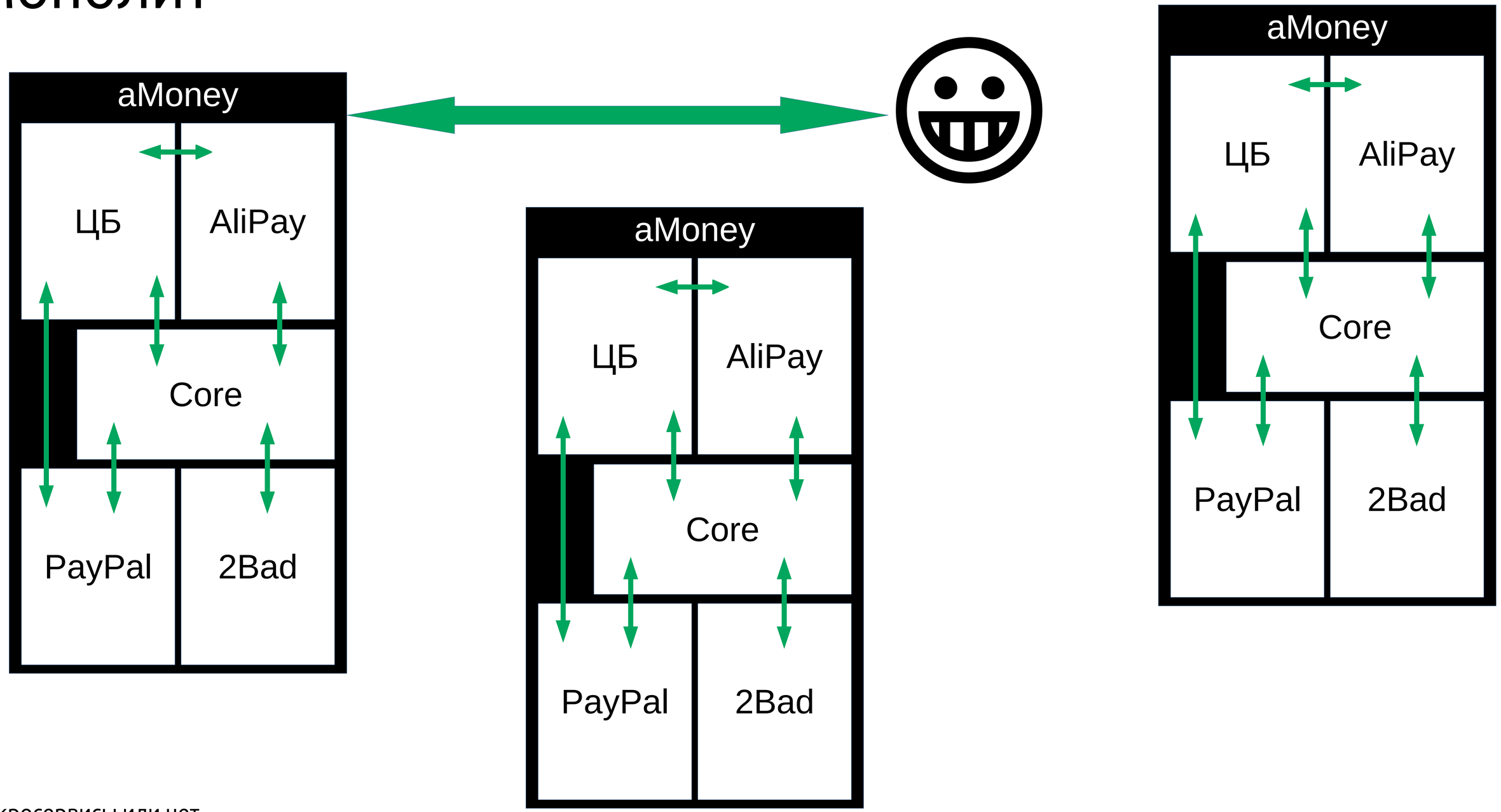
Монолит


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

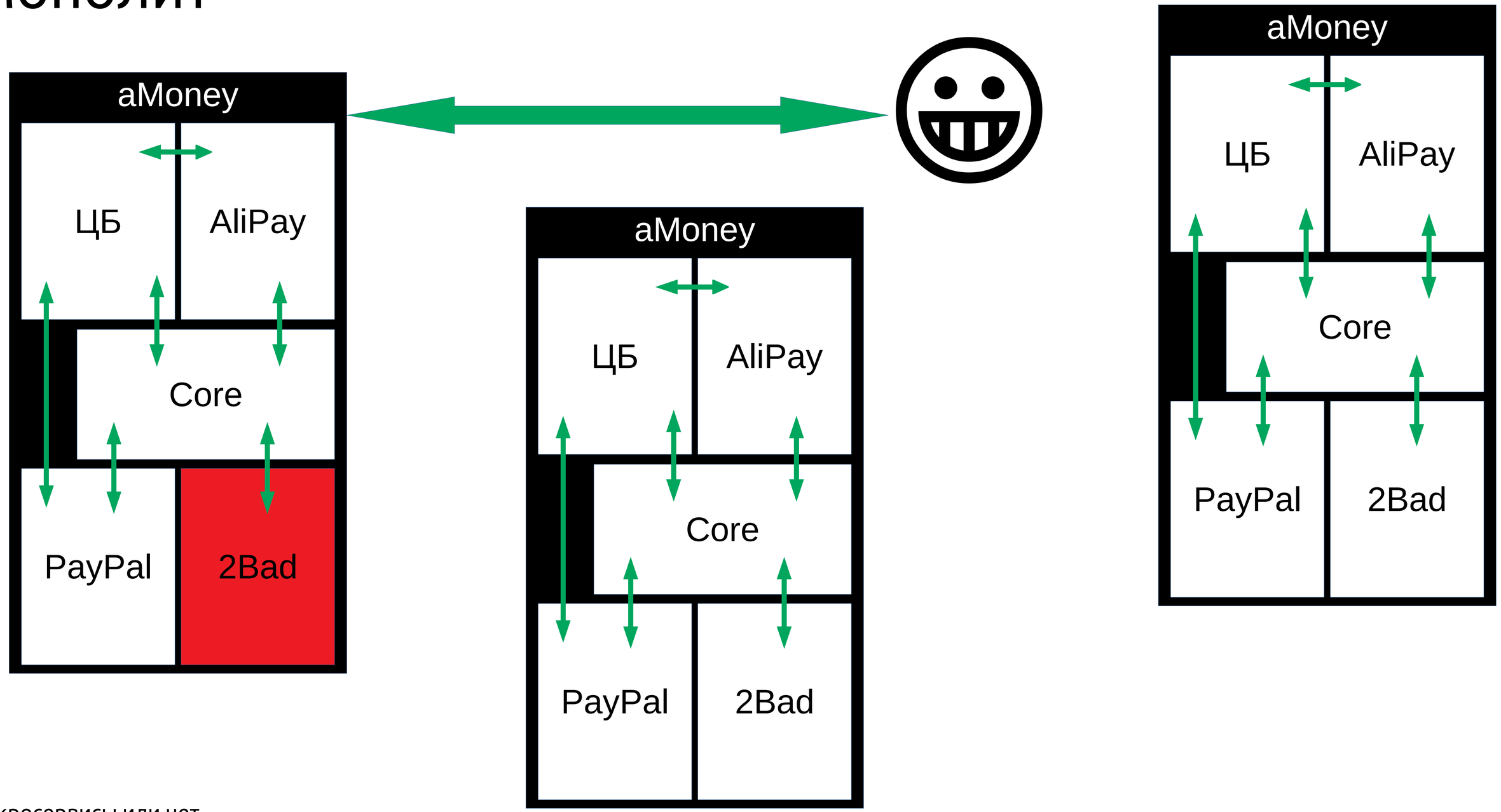

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

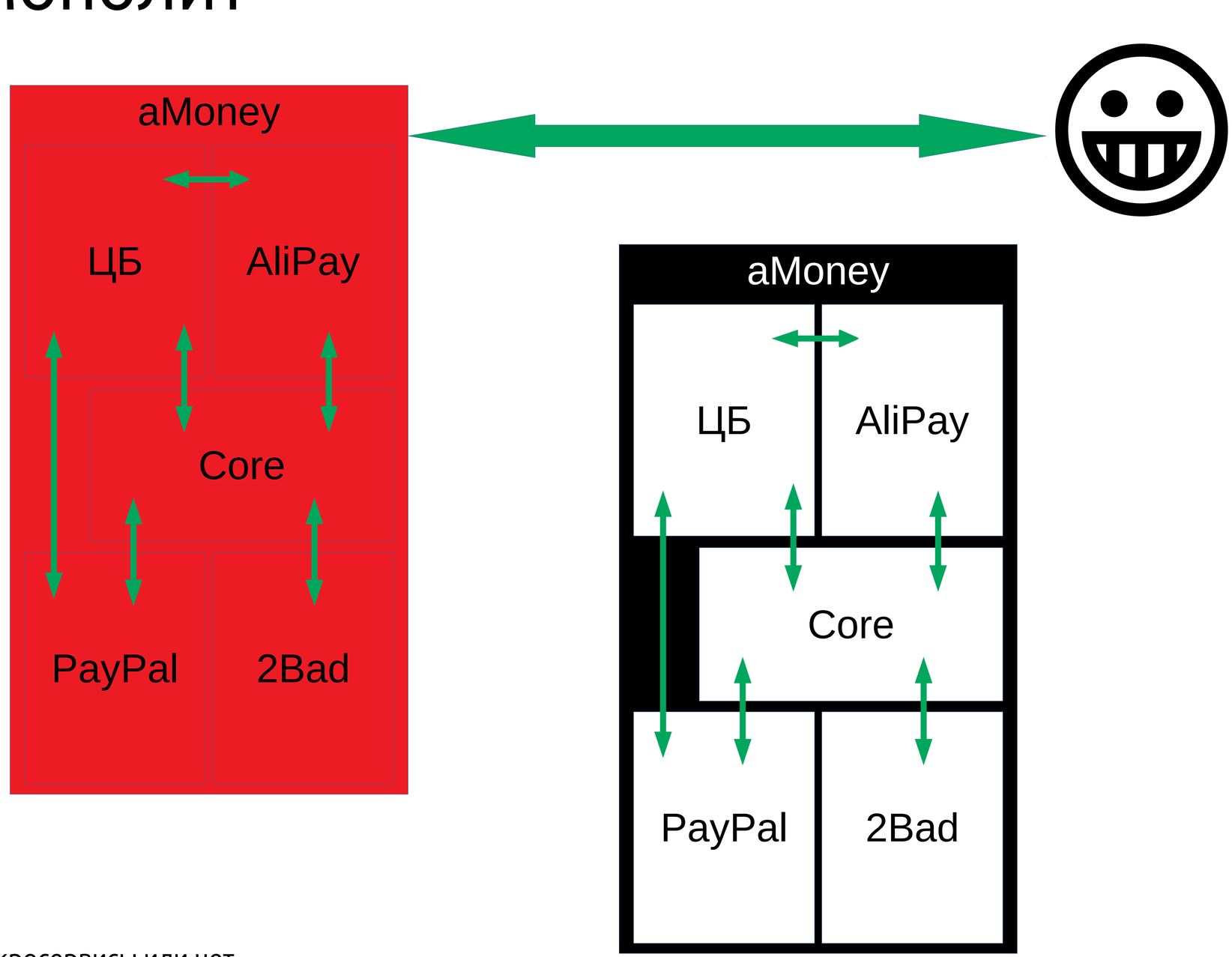
Монолит


aMoney
ЦБ
AliPay
Core
PayPal
2Bad
aMoney
ЦБ
AliPay
Core
PayPal
2Bad
aMoney
ЦБ
AliPay
Core
PayPal
2Bad

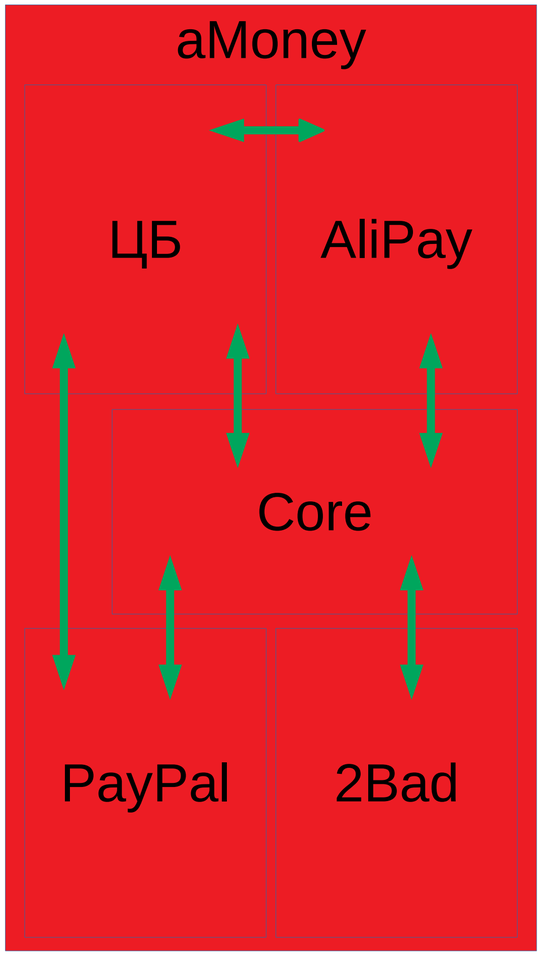
Монолит

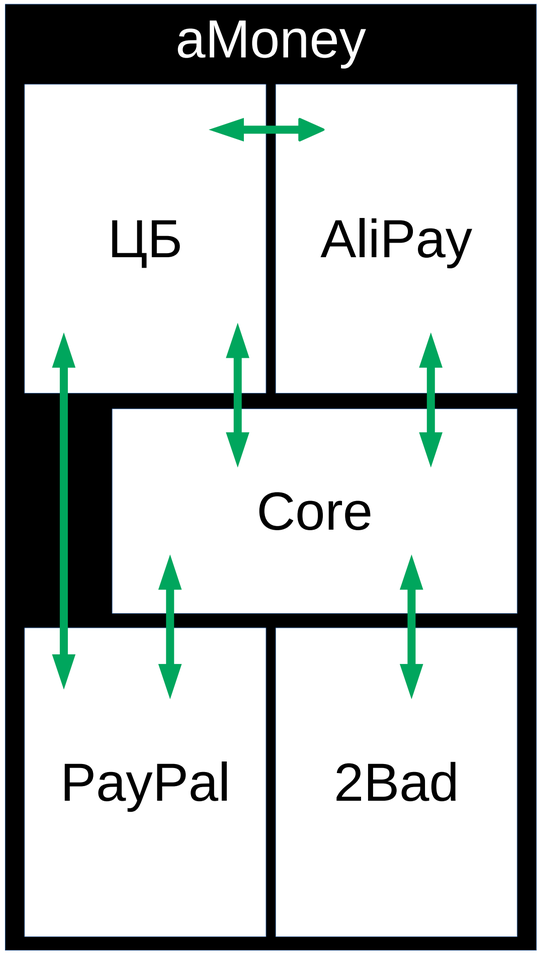

aMoney
ЦБ
AliPay
Core
PayPal
2Bad
aMoney
ЦБ
AliPay
Core
PayPal
2Bad
aMoney
ЦБ
AliPay
Core
PayPal
2Bad

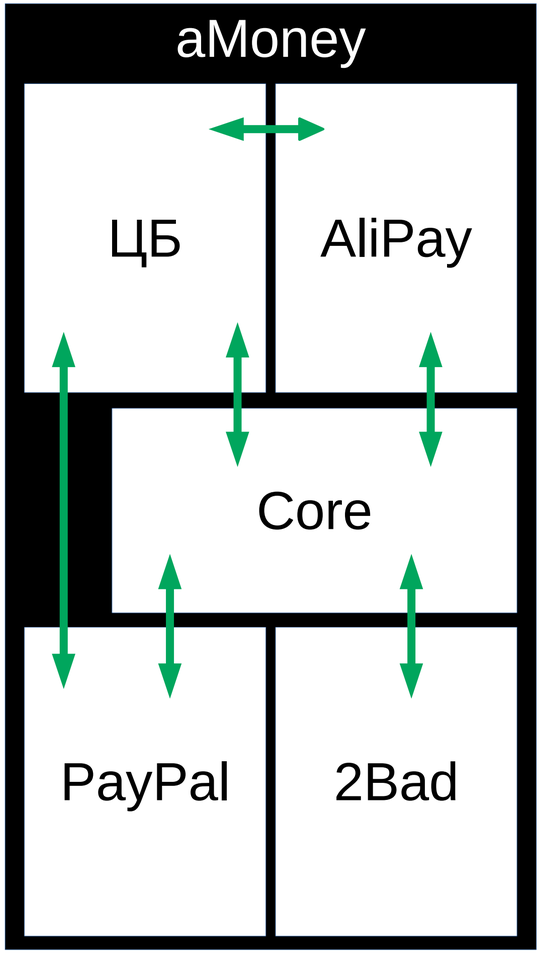
Монолит


aMoney
ЦБ
AliPay
Core
PayPal
2Bad
aMoney
ЦБ
AliPay
Core
PayPal
2Bad
aMoney
ЦБ
AliPay
Core
PayPal
2Bad

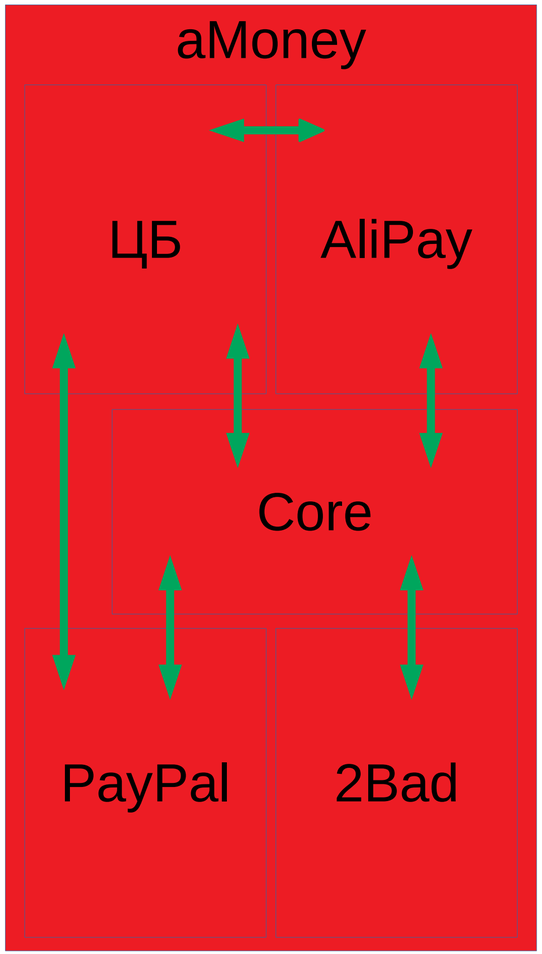
Монолит

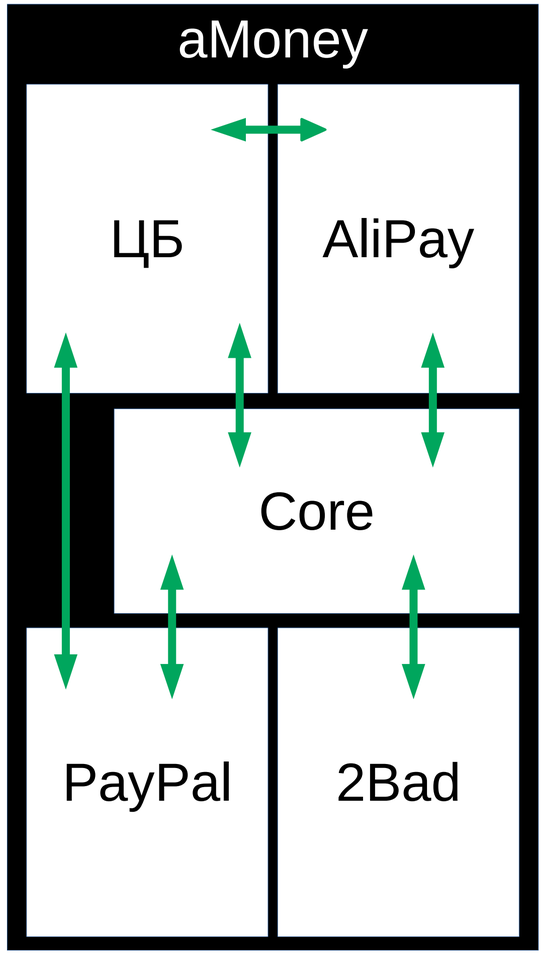

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

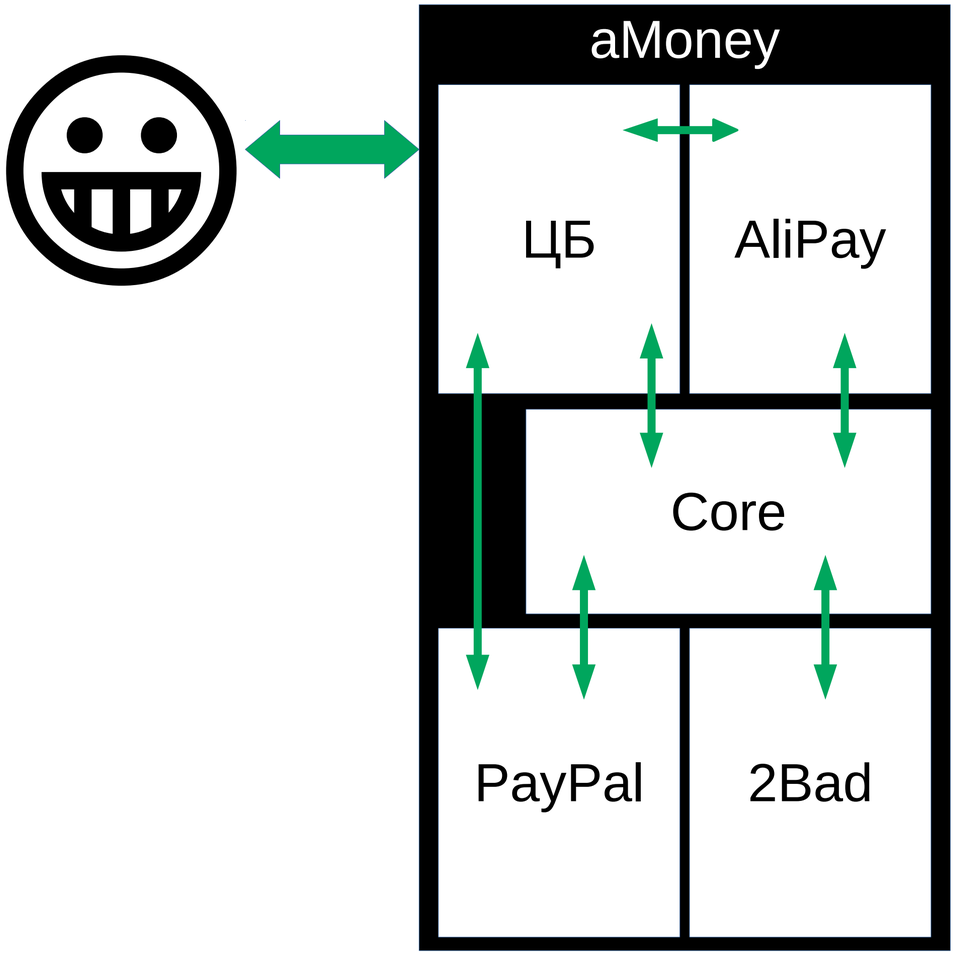

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

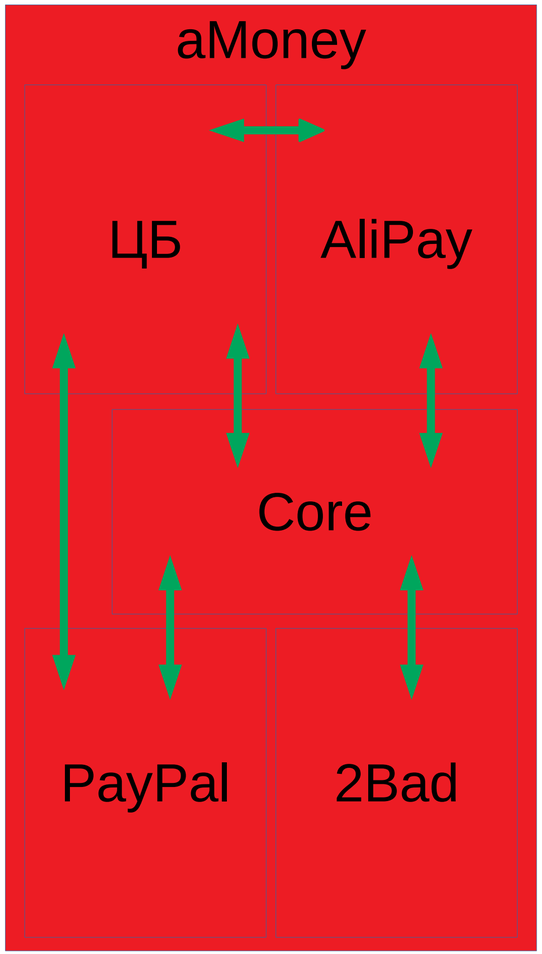
Монолит

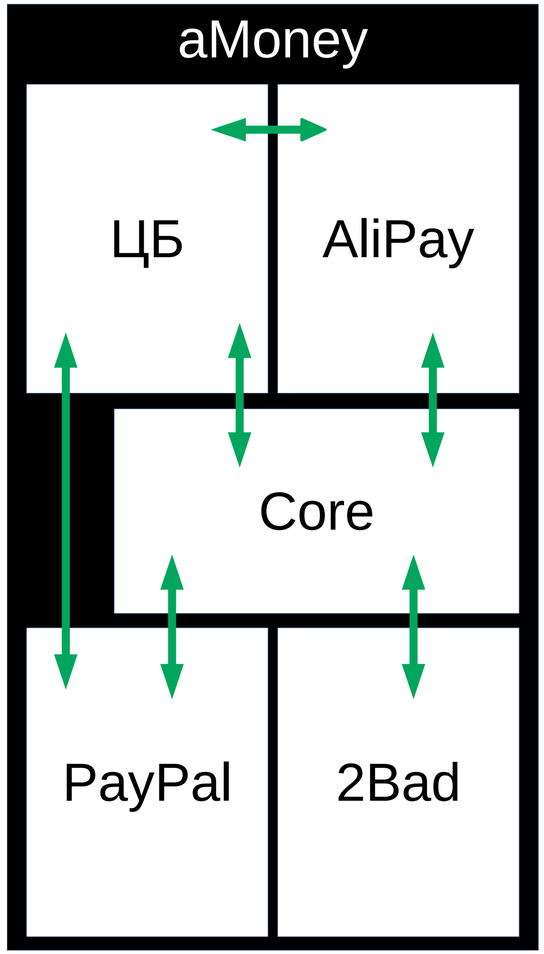

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

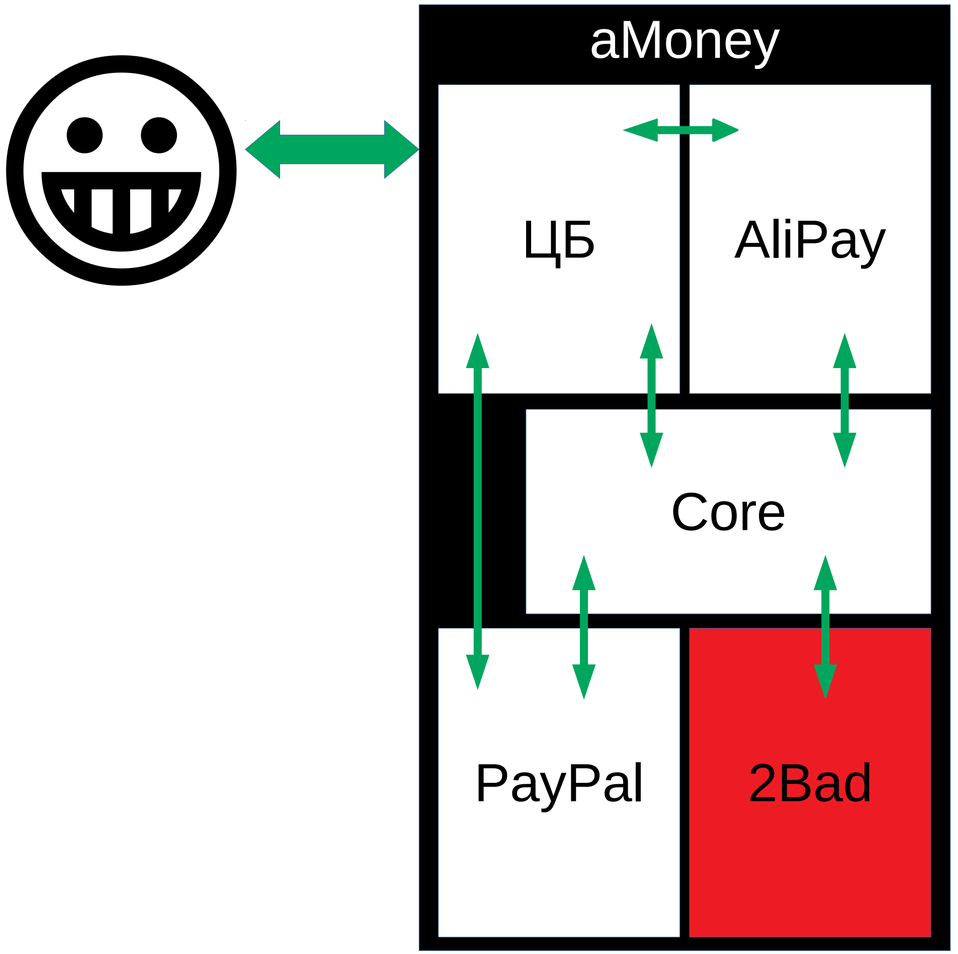

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

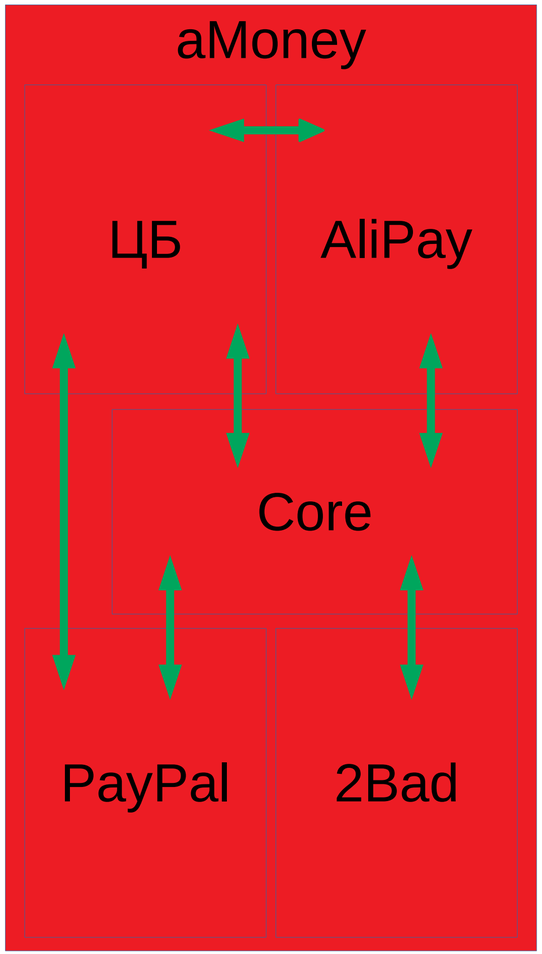
Монолит

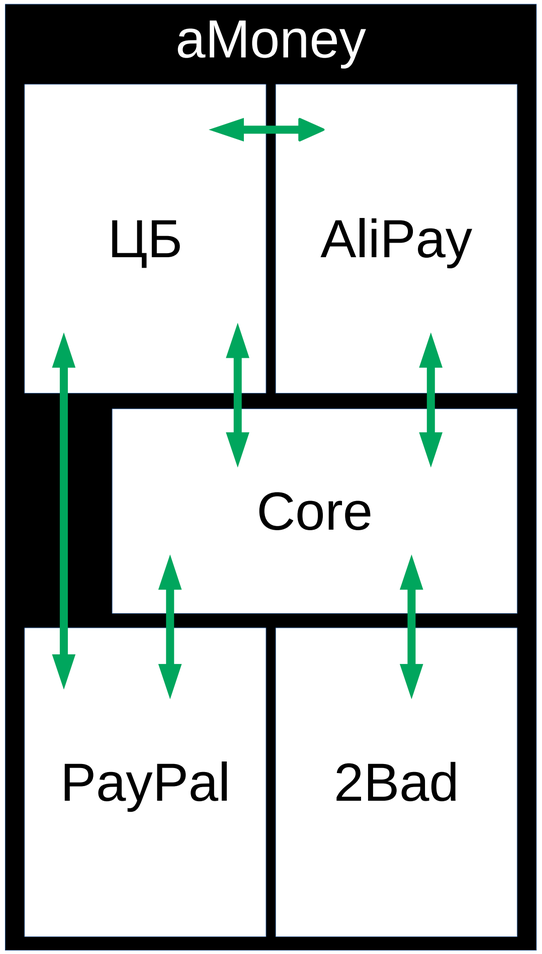

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

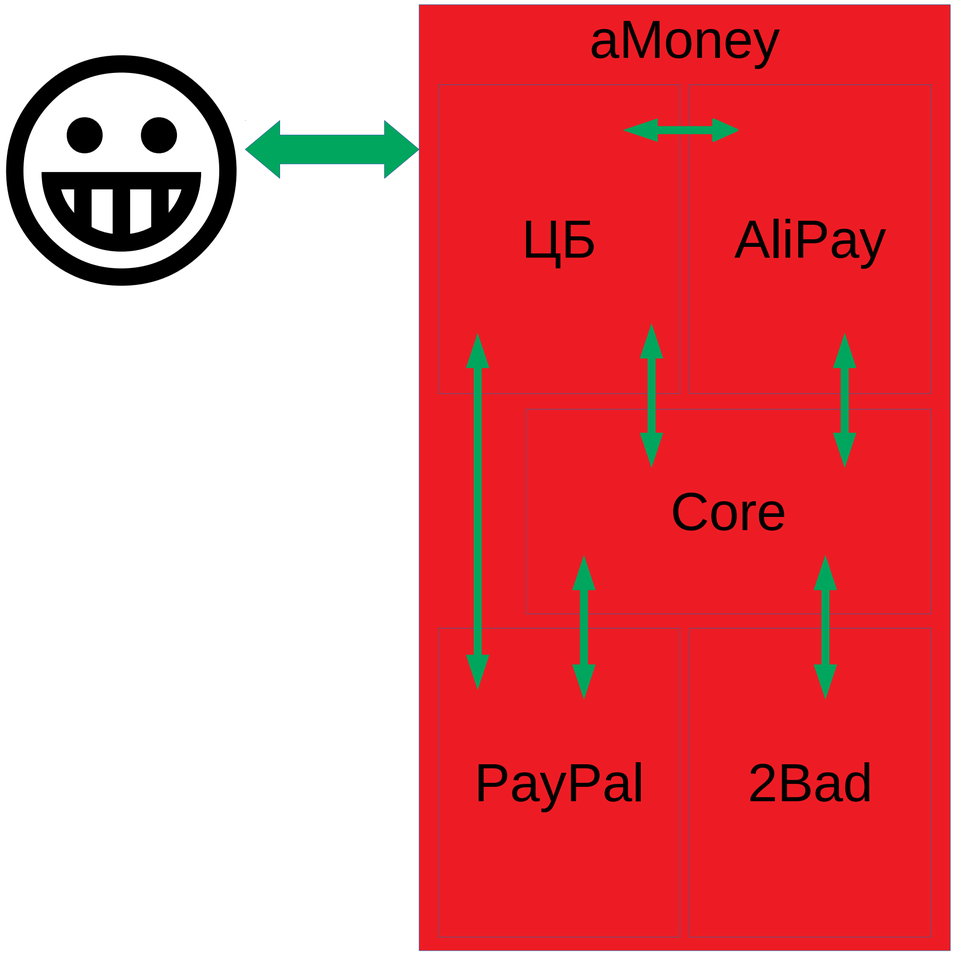

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

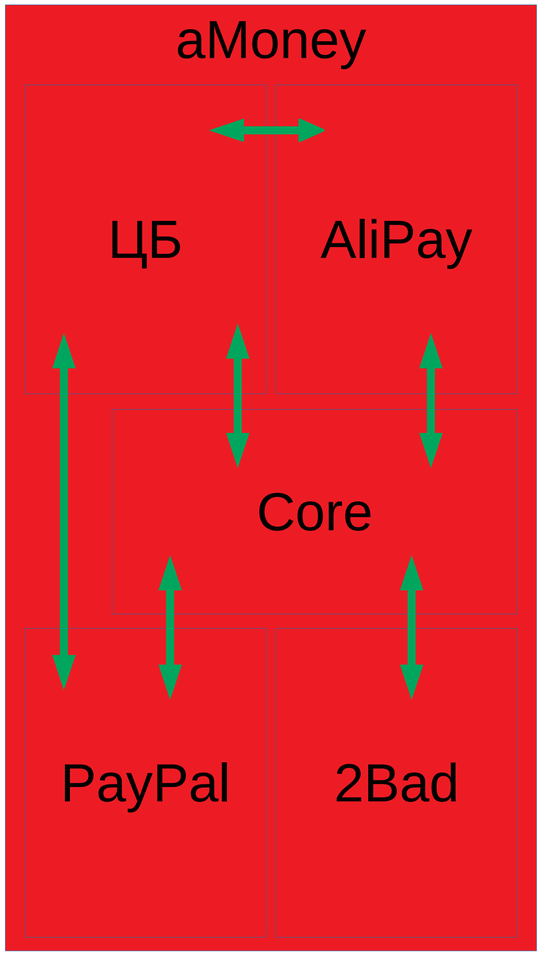
Монолит

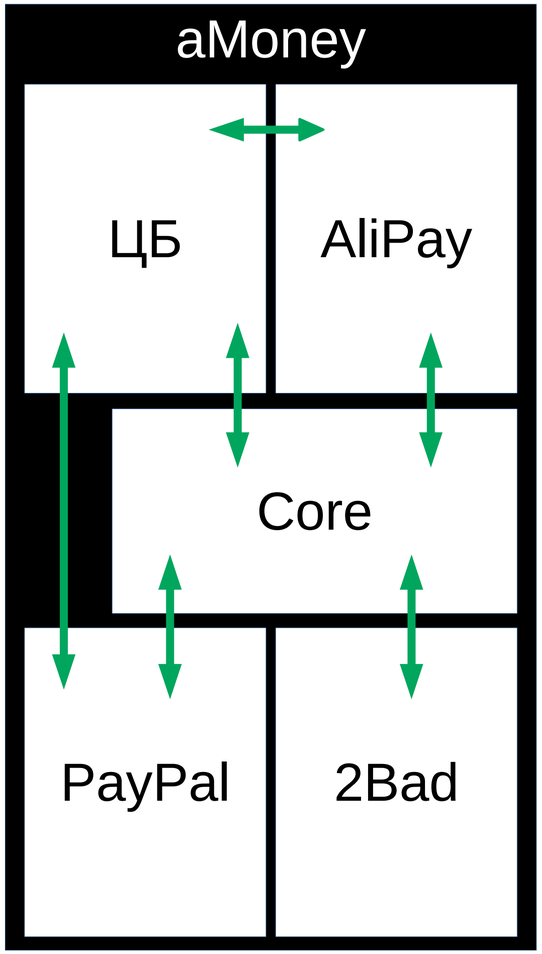

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

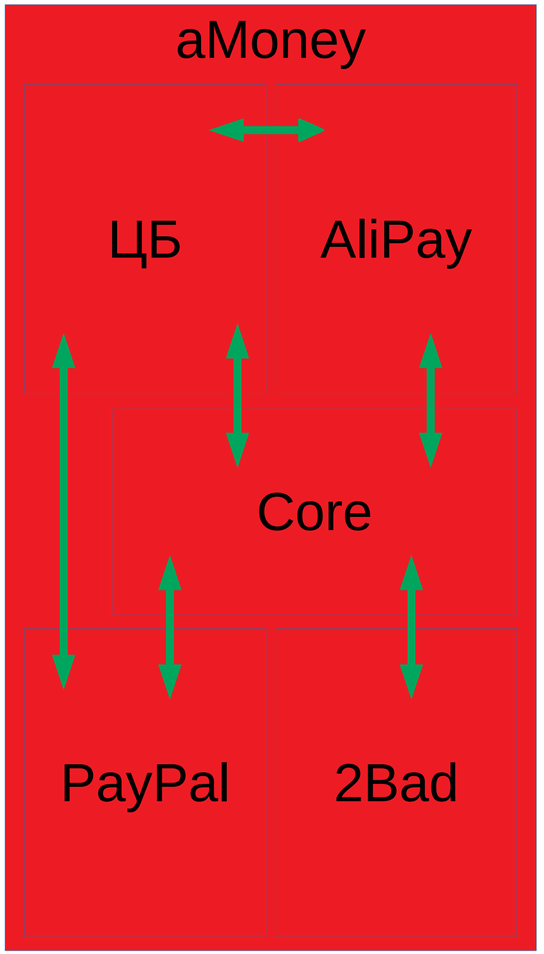

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

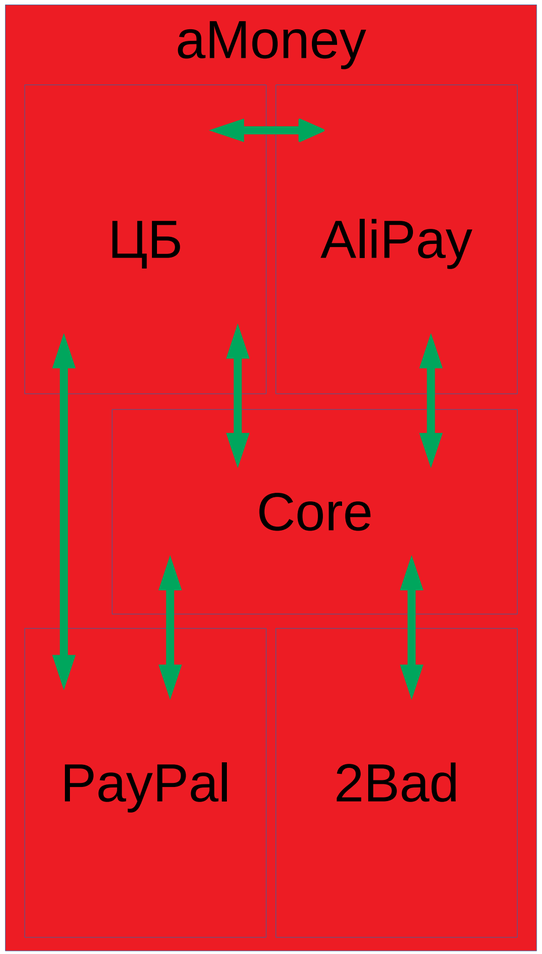
Монолит

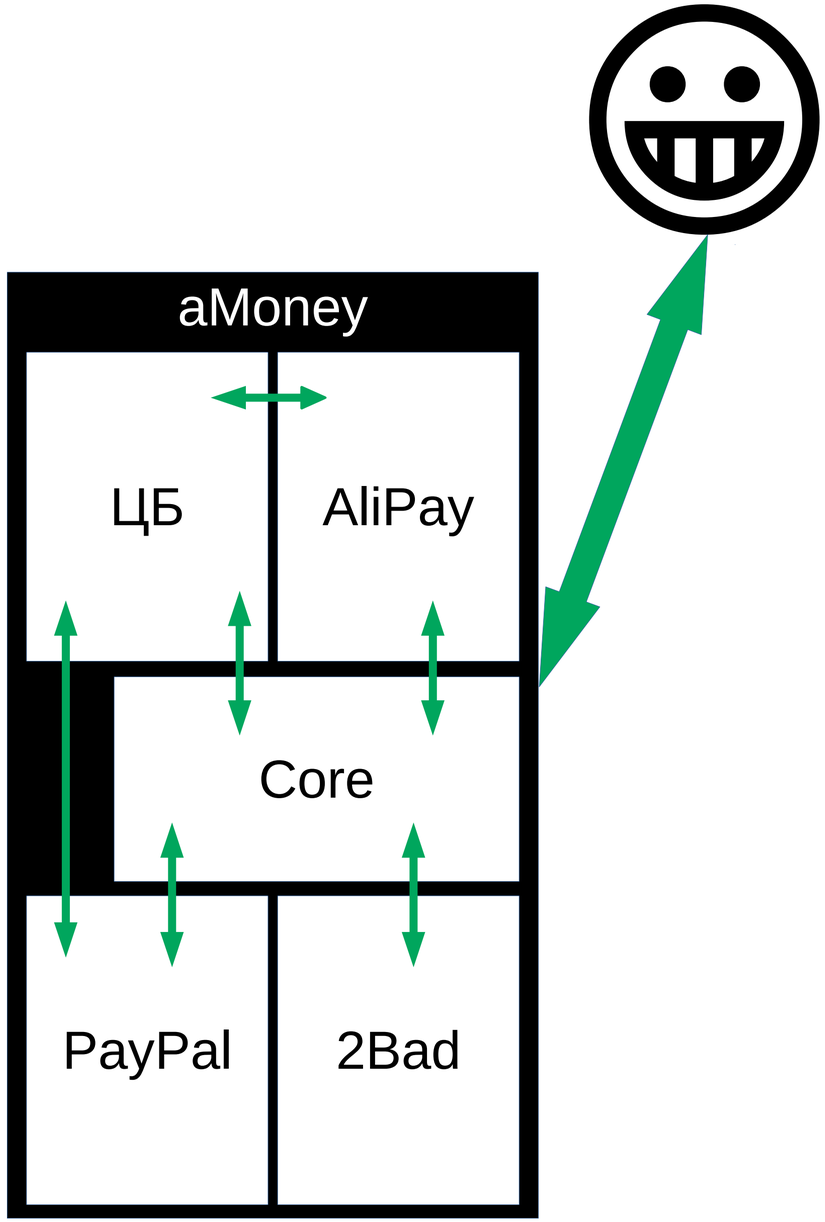

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

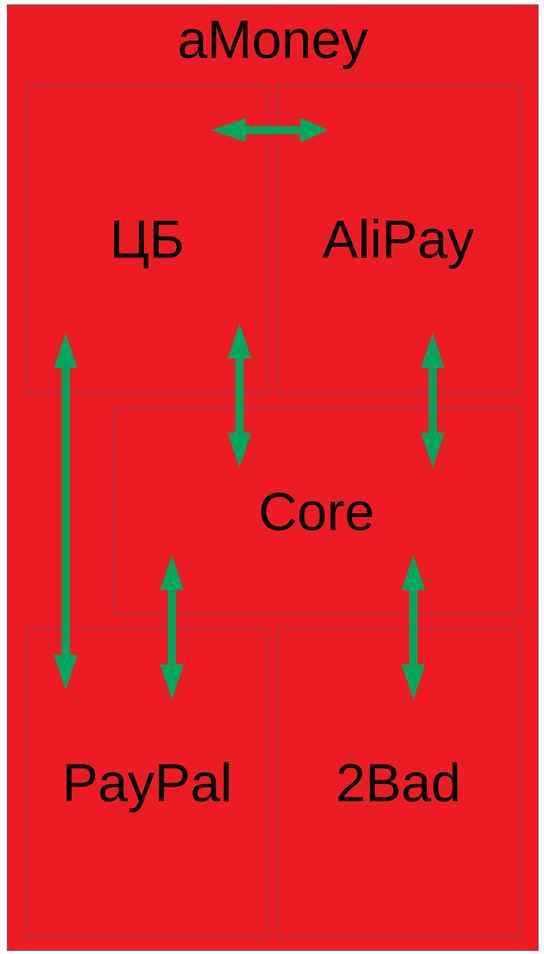

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

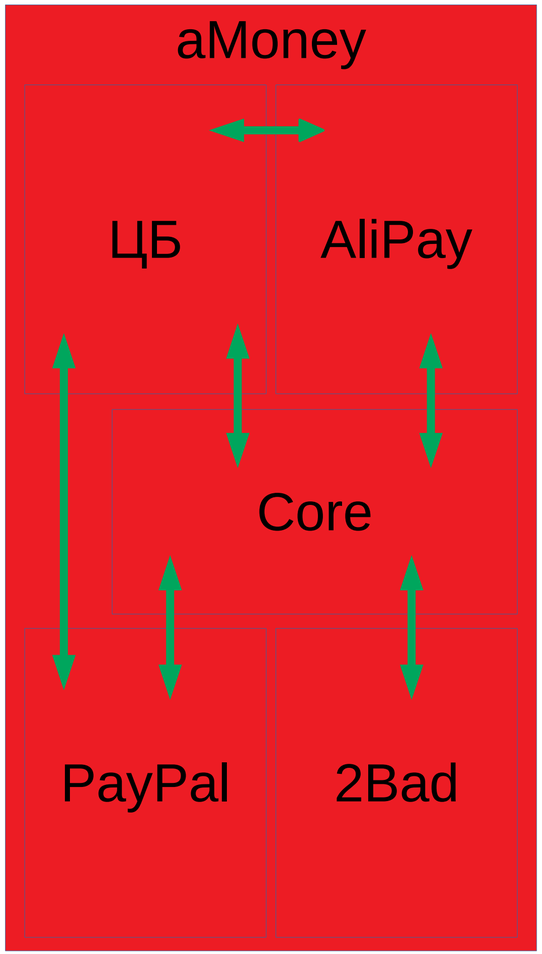
Монолит

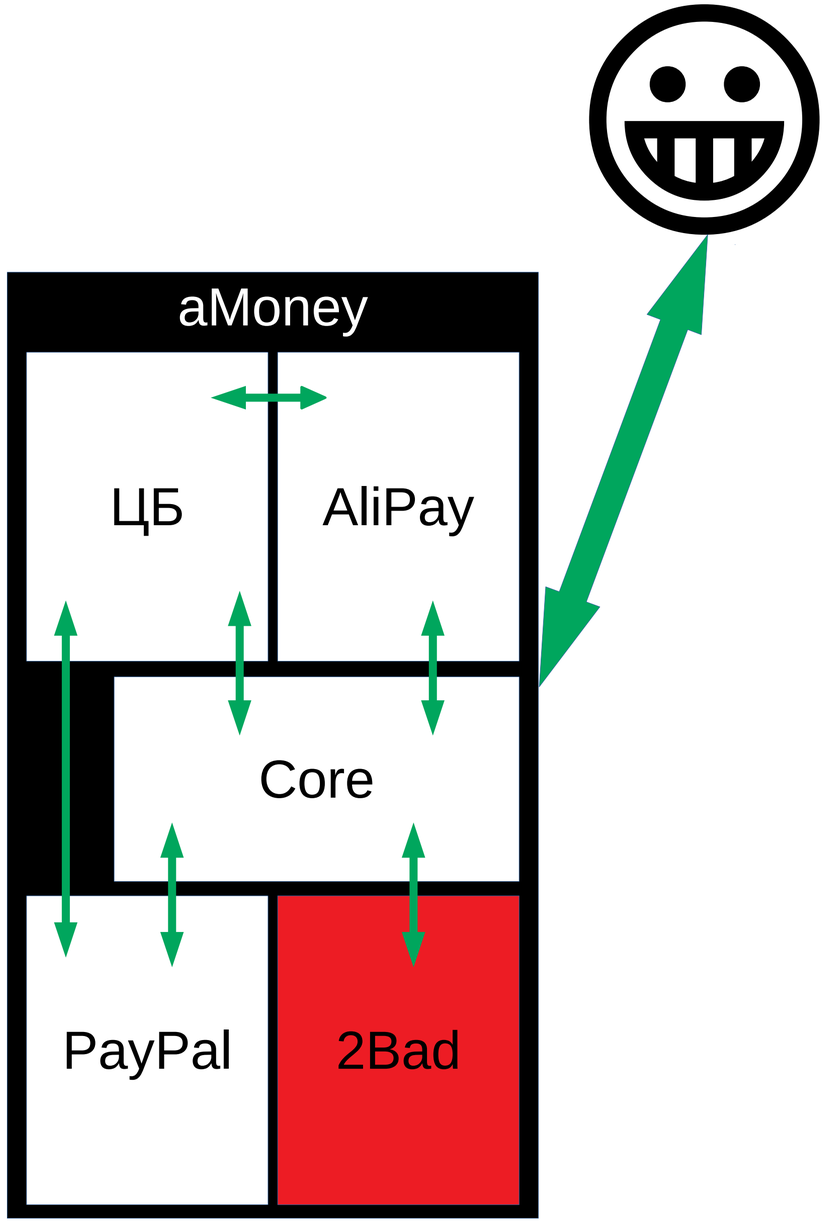

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

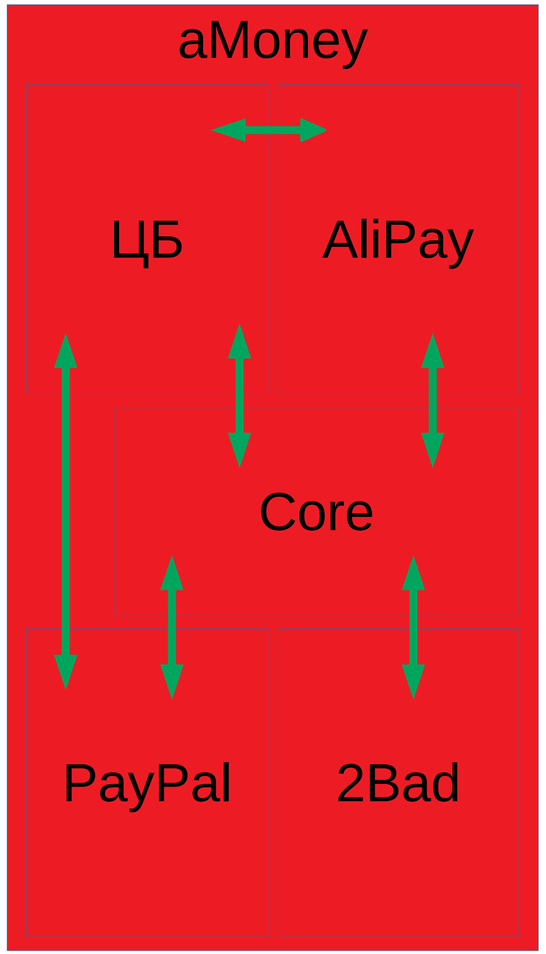

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

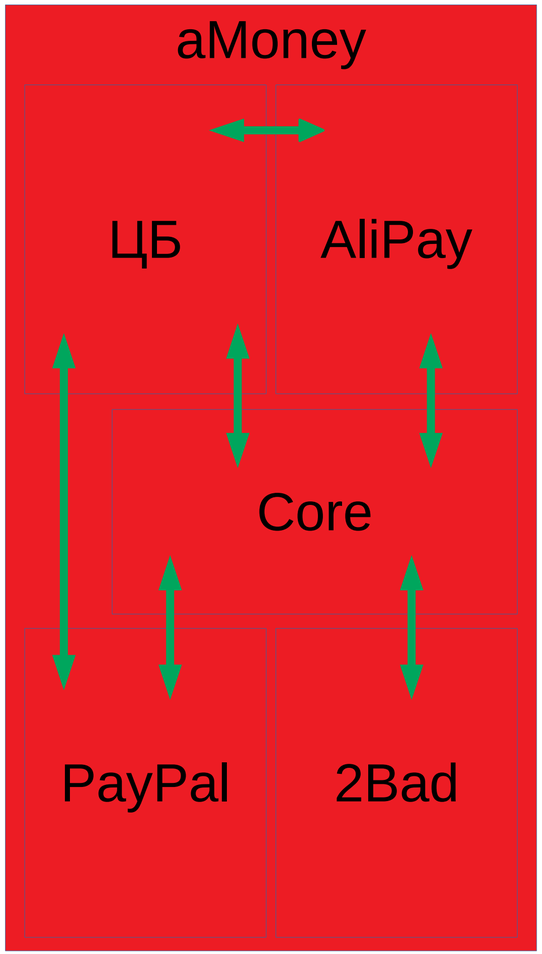
Монолит

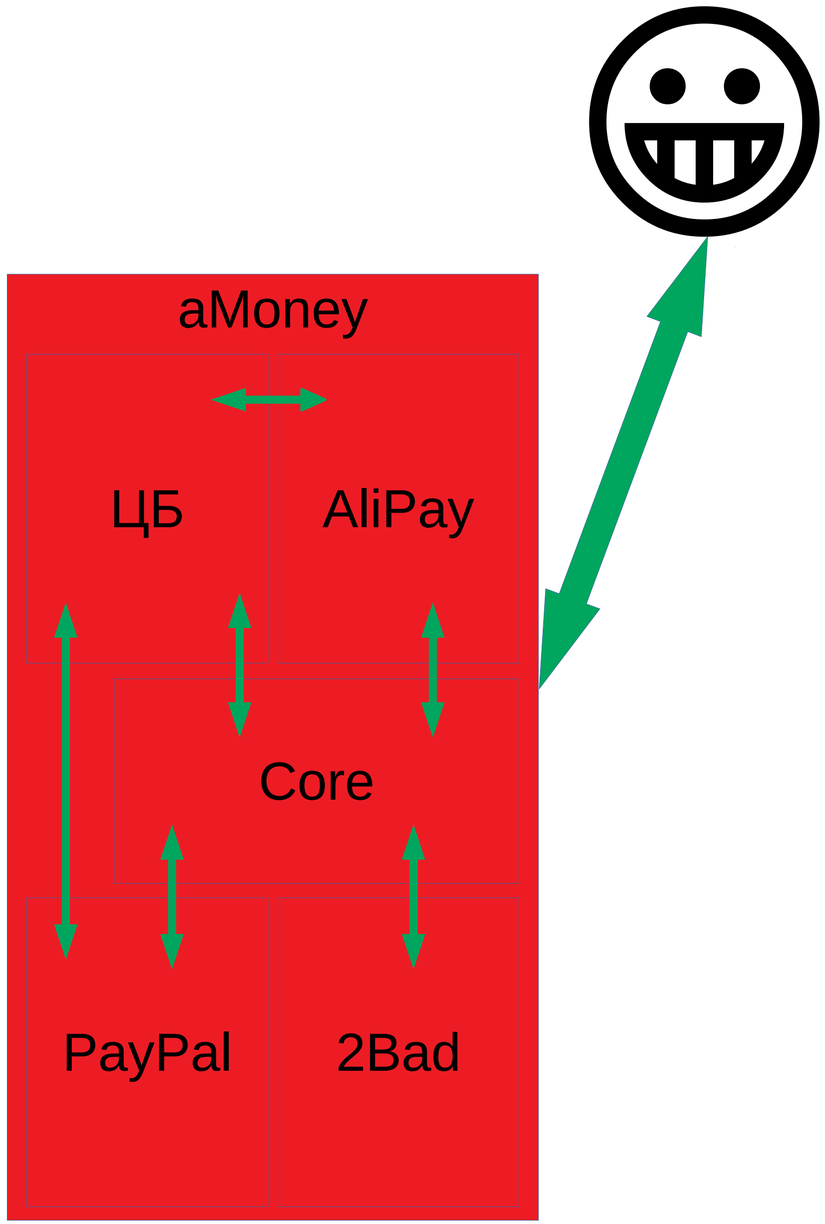

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

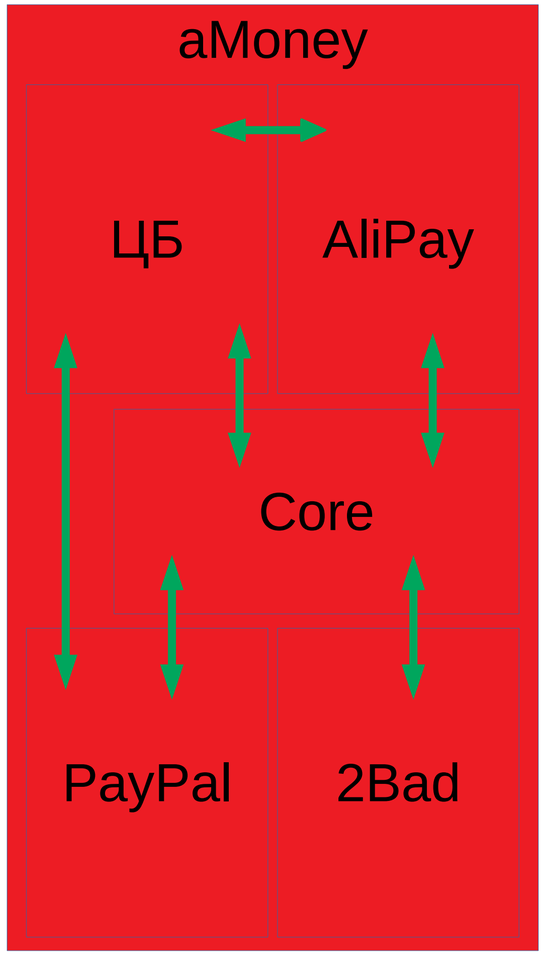

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

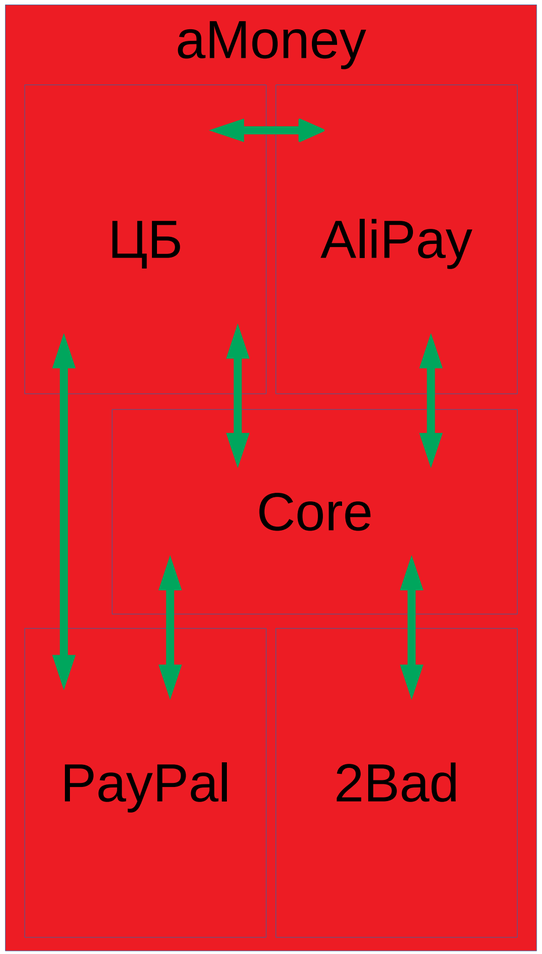
Монолит

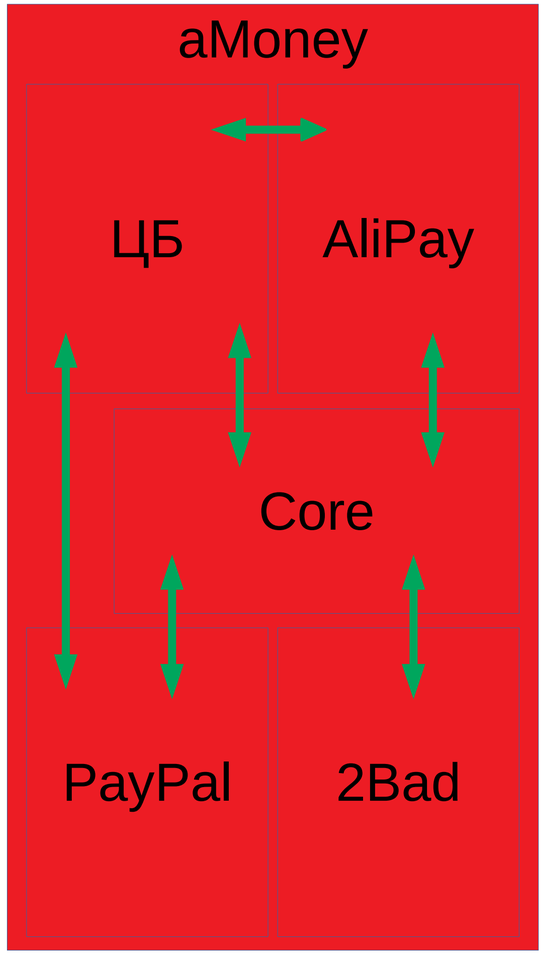

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

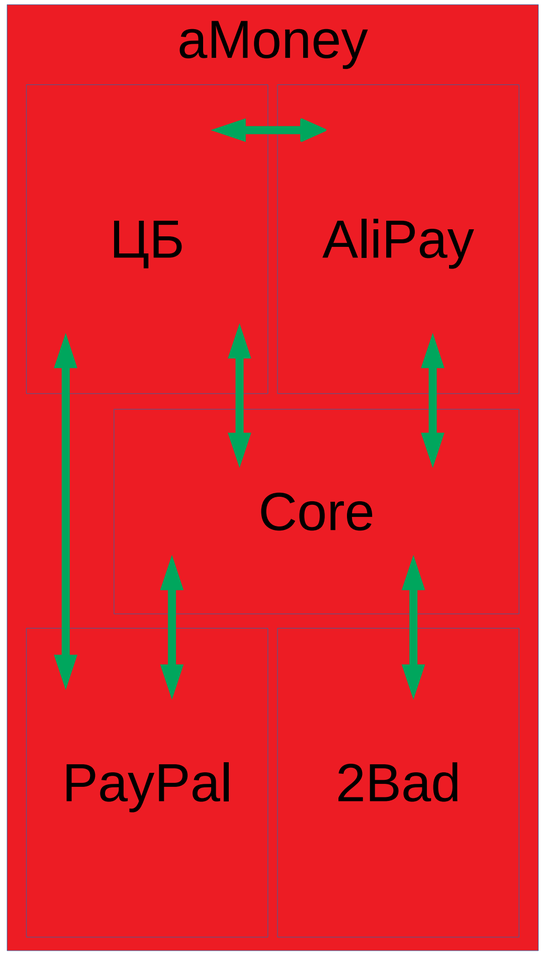

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

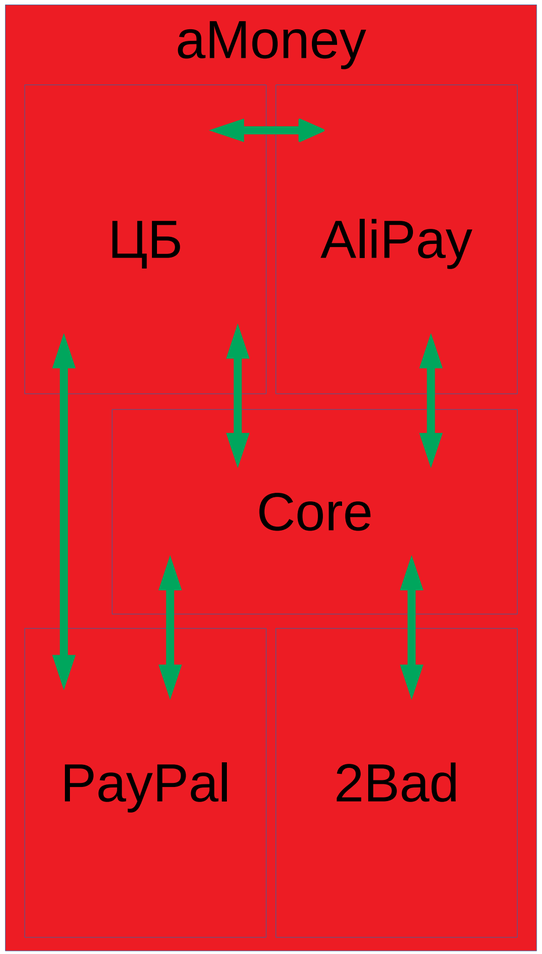
Монолит

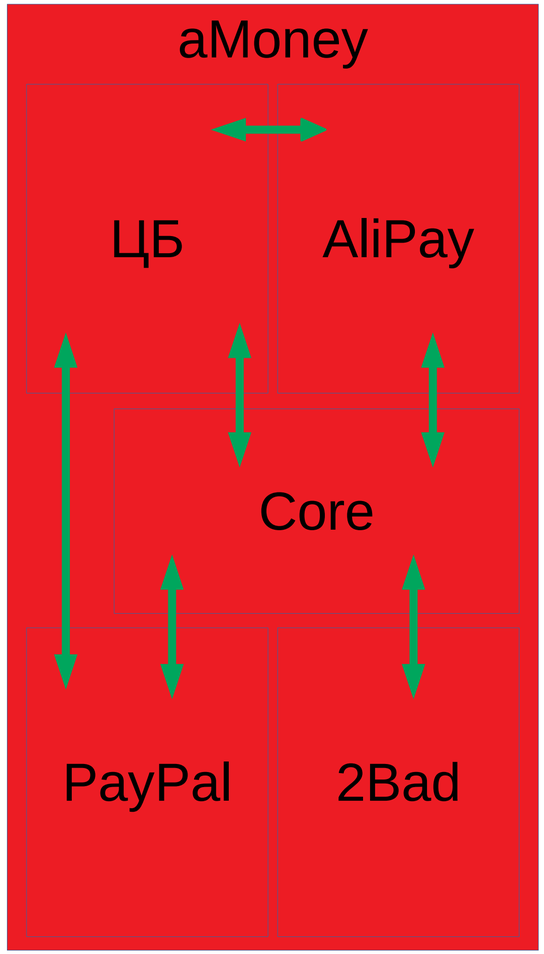

aMoney
ЦБ
AliPay
Core
PayPal
2Bad

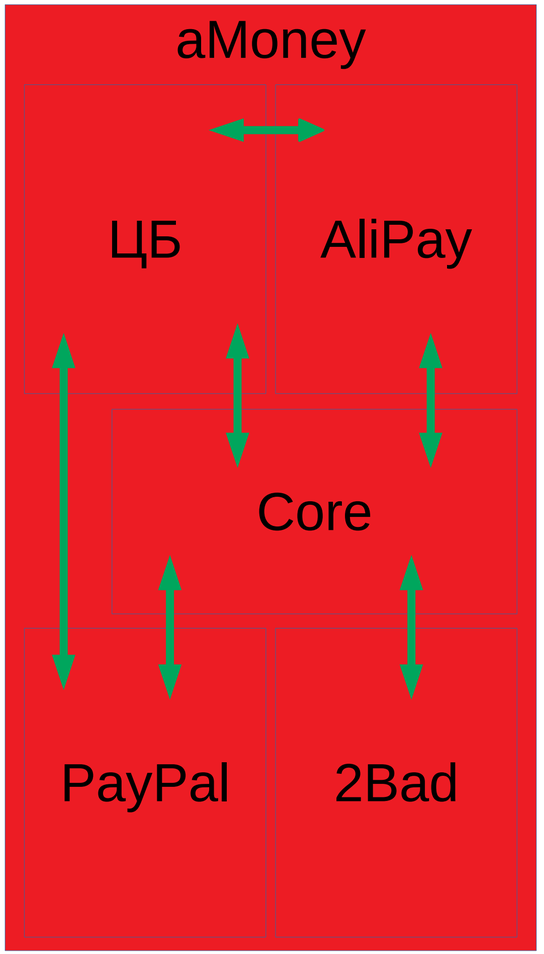

aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

Монолит


aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad


aMoney
ЦБ
AliPay
Core
PayPal
2Bad

Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями
- Тесное общение между разработчиками

Минусы:

- Не идеальная надёжность

Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями
- Тесное общение между разработчиками

Минусы:

- Не идеальная надёжность
  - и при этом множество кластеров не спасает

Плюсы/минусы для небольшой команды

Плюсы:

- Простой деплой
- Дешёвая передача данных между модулями
- Тесное общение между разработчиками

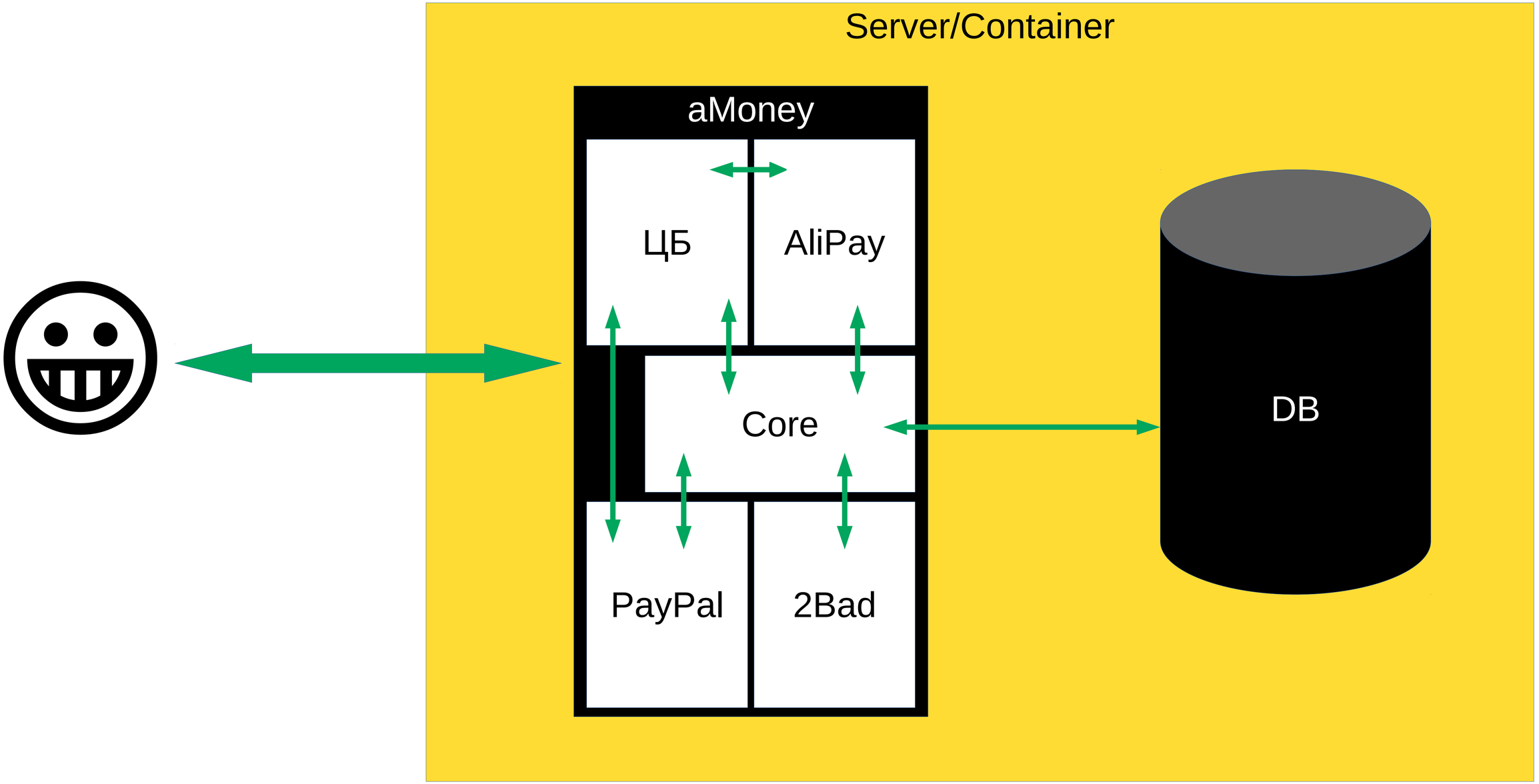
Минусы:

- Не идеальная надёжность
  - и при этом множество кластеров не спасает

Итог: неплохое решение для малых команд и малых кодовых баз

Команда стала больше, больше кода

Монолит



Плюсы/минусы для большой команды

Плюсы/минусы для большой команды

Плюсы:

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

Минусы:

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

Минусы:

- Плохая надёжность

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

Минусы:

- Плохая надёжность
- Тесное общение между разработчиками

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

Минусы:

- Плохая надёжность
- Тесное общение между разработчиками
- Долгая сборка

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

Минусы:

- Плохая надёжность
- Тесное общение между разработчиками
- Долгая сборка
  - Долгий деплой

Плюсы/минусы для большой команды

Плюсы:

- Дешёвая передача данных между модулями

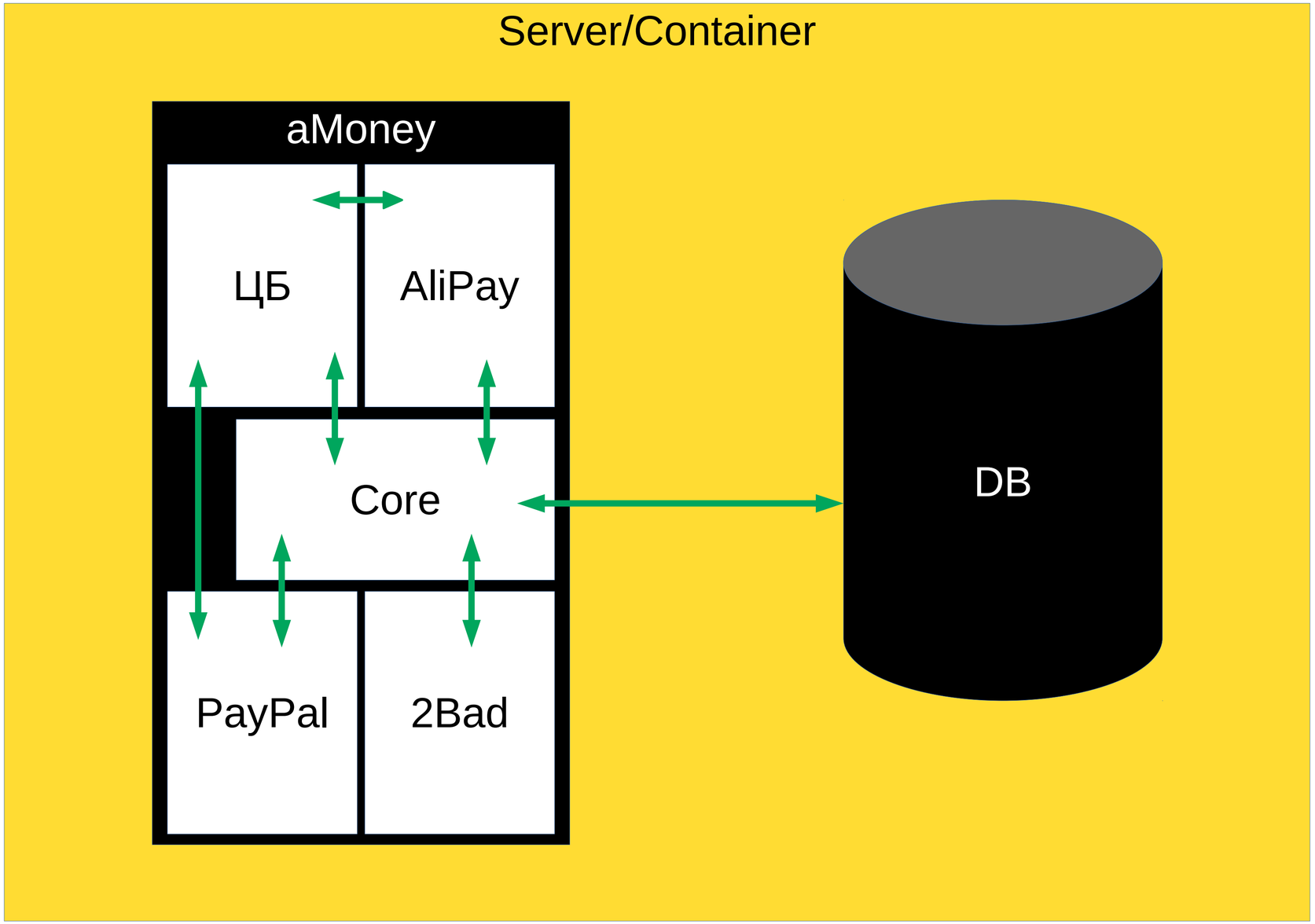
Минусы:

- Плохая надёжность
- Тесное общение между разработчиками
- Долгая сборка
  - Долгий деплой

Итог: жить ещё можно

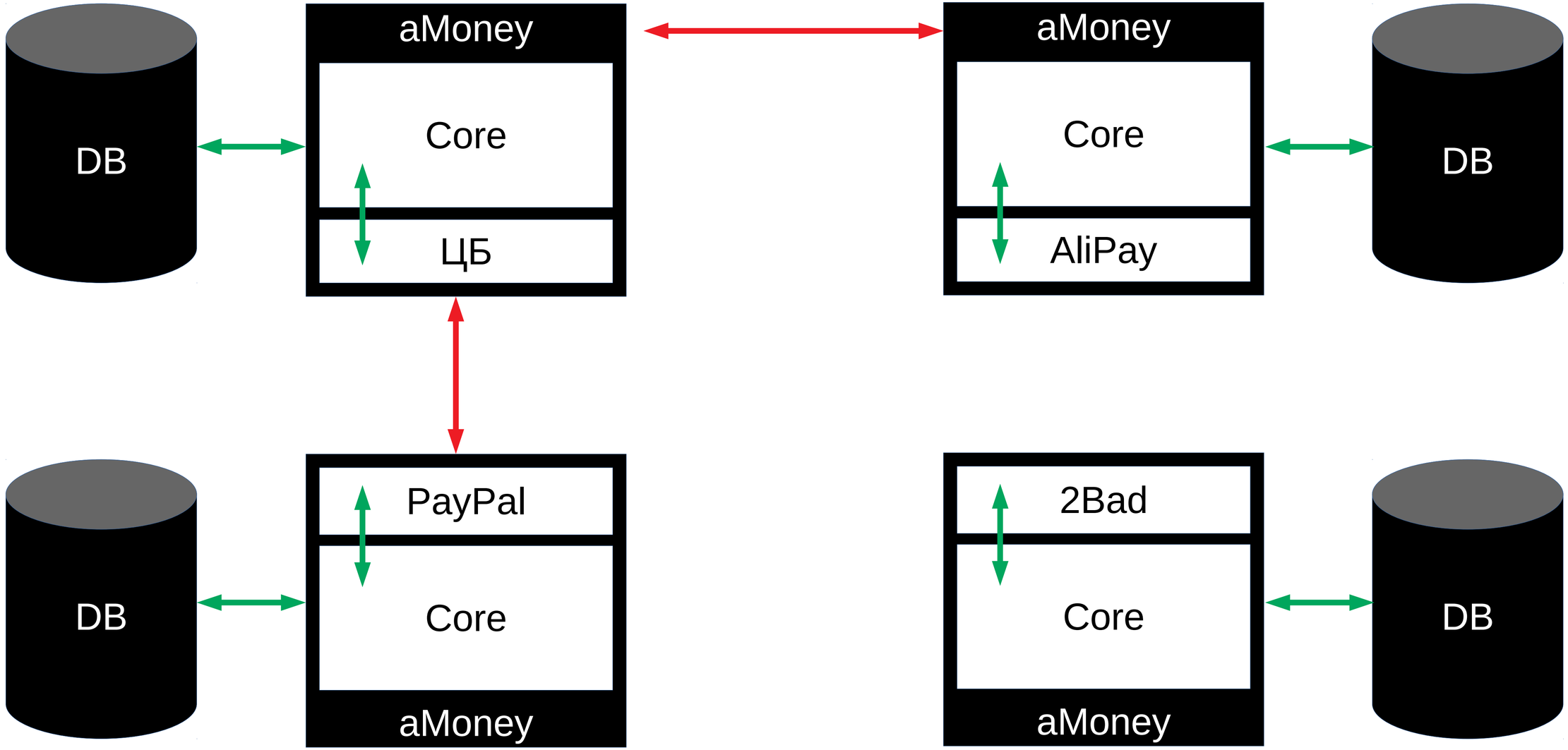
Команда стала ещё больше

Монолит



Правильные микросервисы

Микросервисы



Плюсы/минусы микросервисов для большой команды

Плюсы/минусы микросервисов для большой команды

Плюсы:

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

Минусы:

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

Минусы:

- Дорогая передача данных между модулями

Плюсы/минусы микросервисов для большой команды

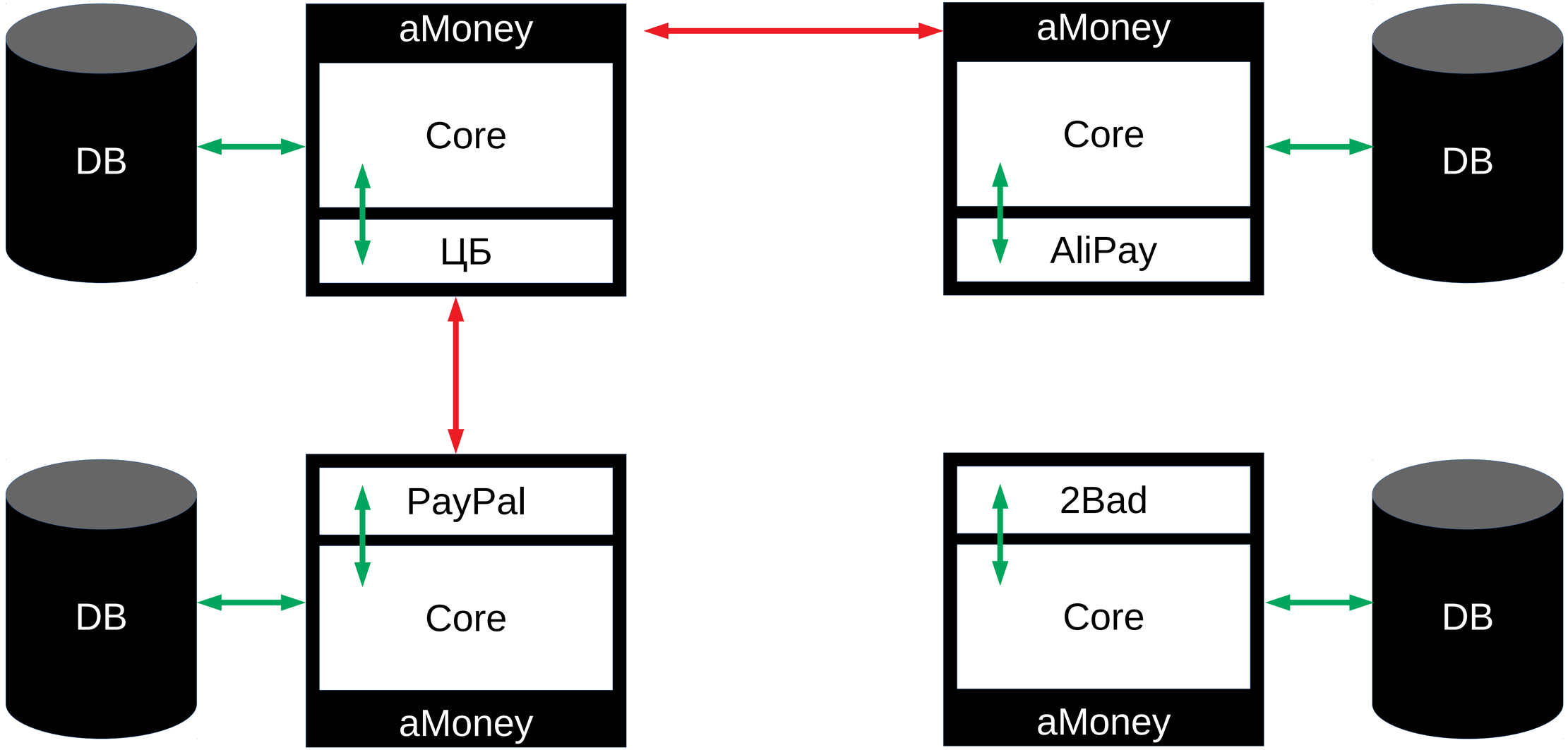
Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

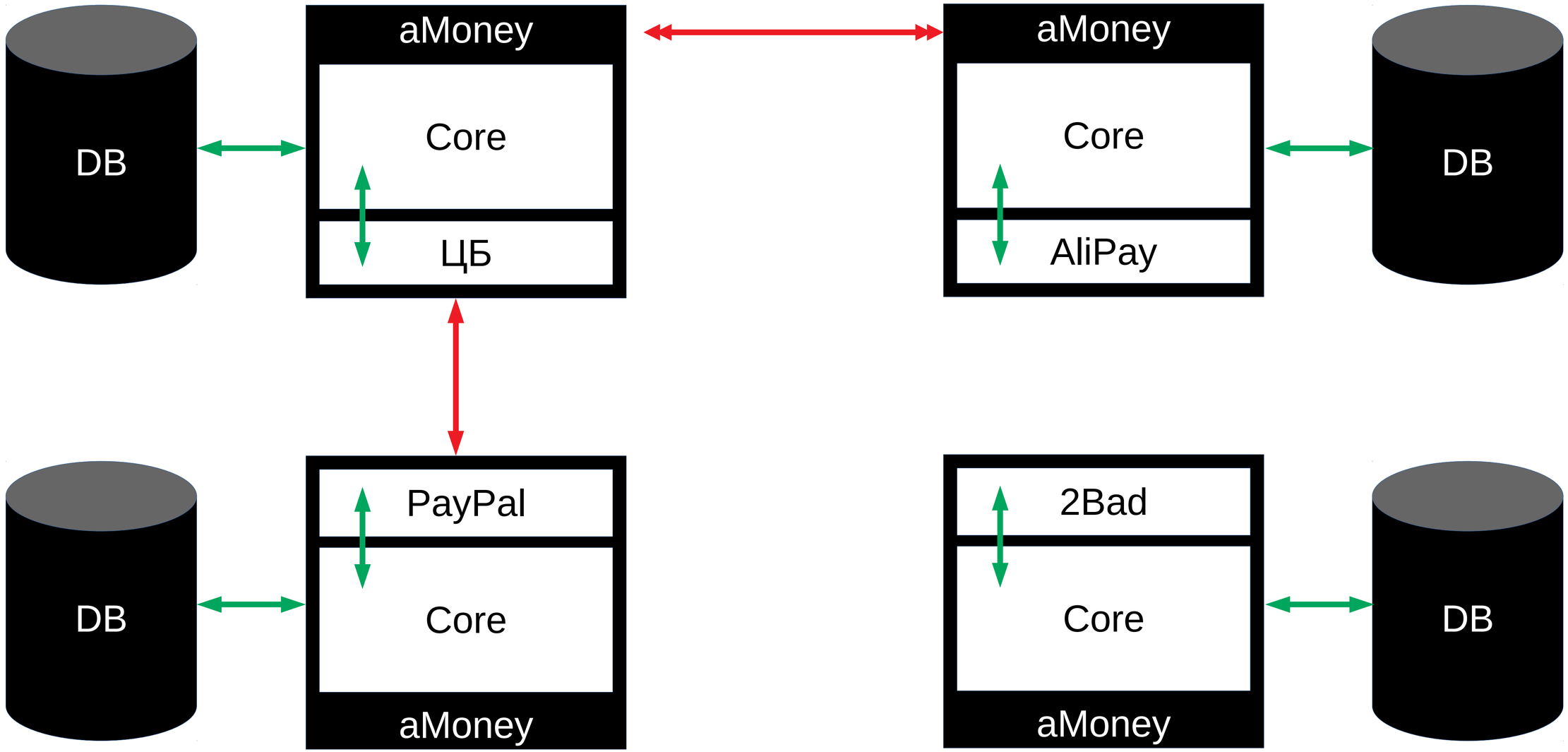
Минусы:

- Дорогая передача данных между модулями
- Обязательное версионирование и поддержка старых версий

Микросервисы



Микросервисы



Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

Минусы:

- Дорогая передача данных между модулями
- Обязательное версионирование и поддержка старых версий

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

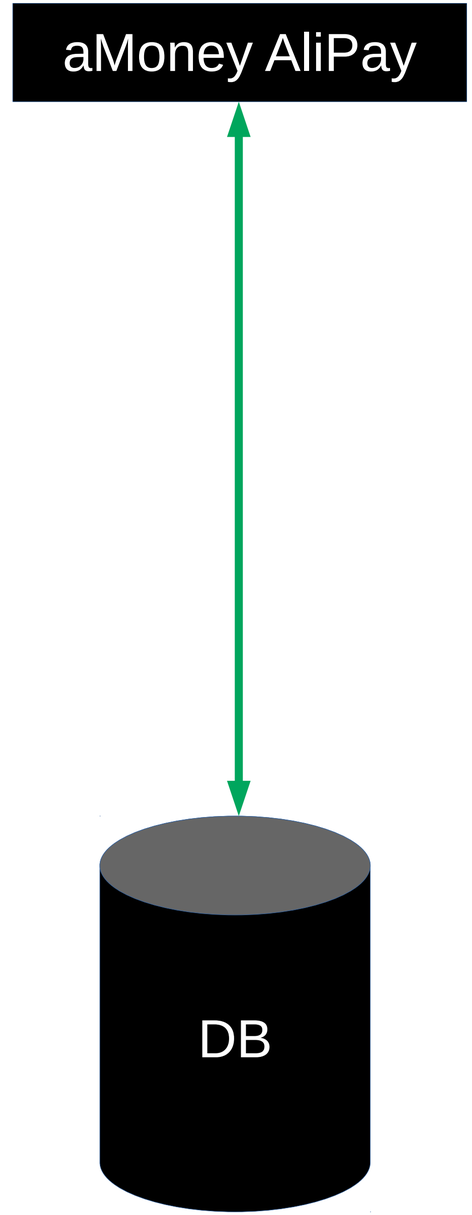
Минусы:

- Дорогая передача данных между модулями
- Обязательное версионирование и поддержка старых версий
- Большие траты на железо

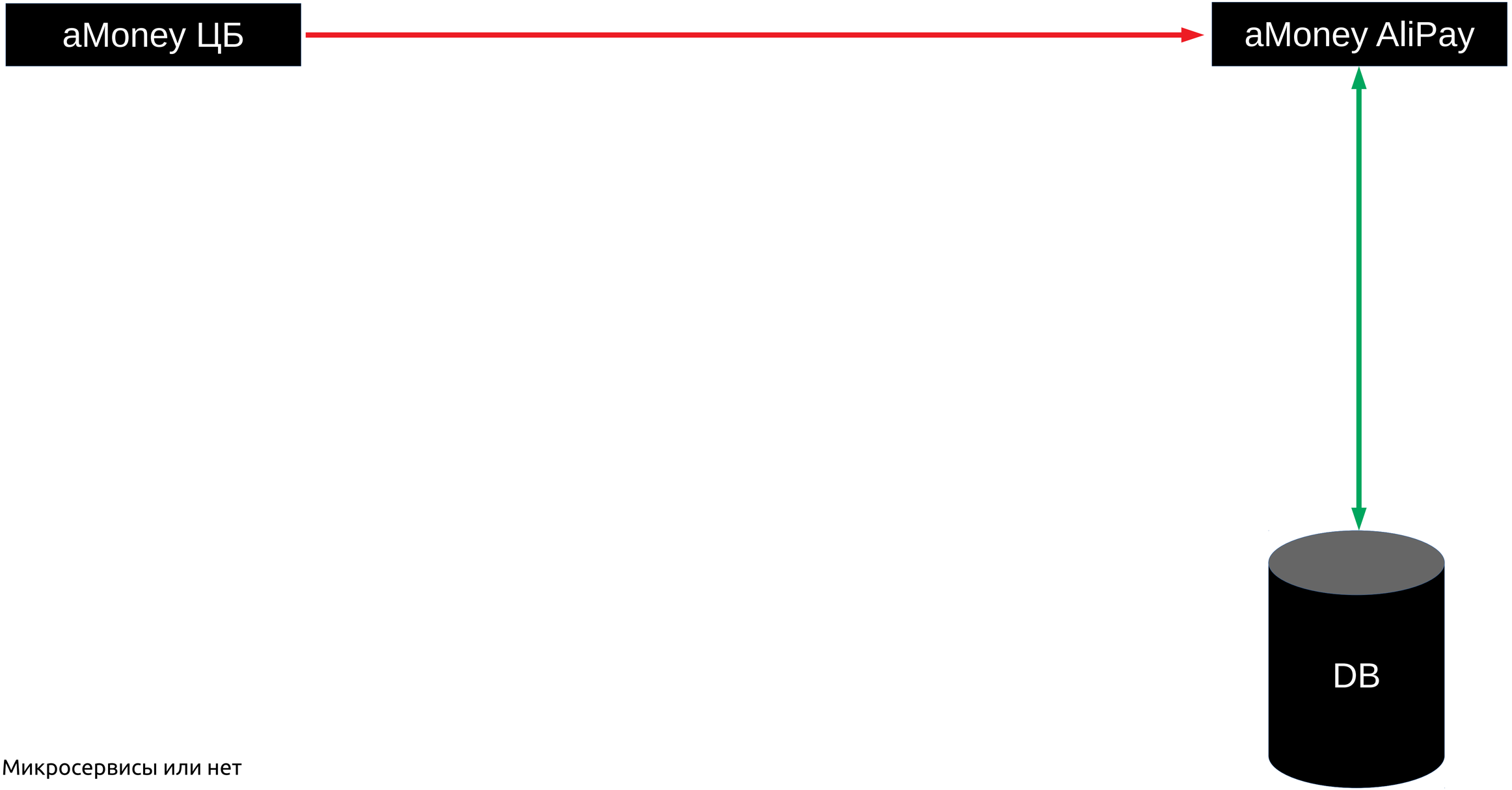
А при чём тут балансеры?

Микросервисы

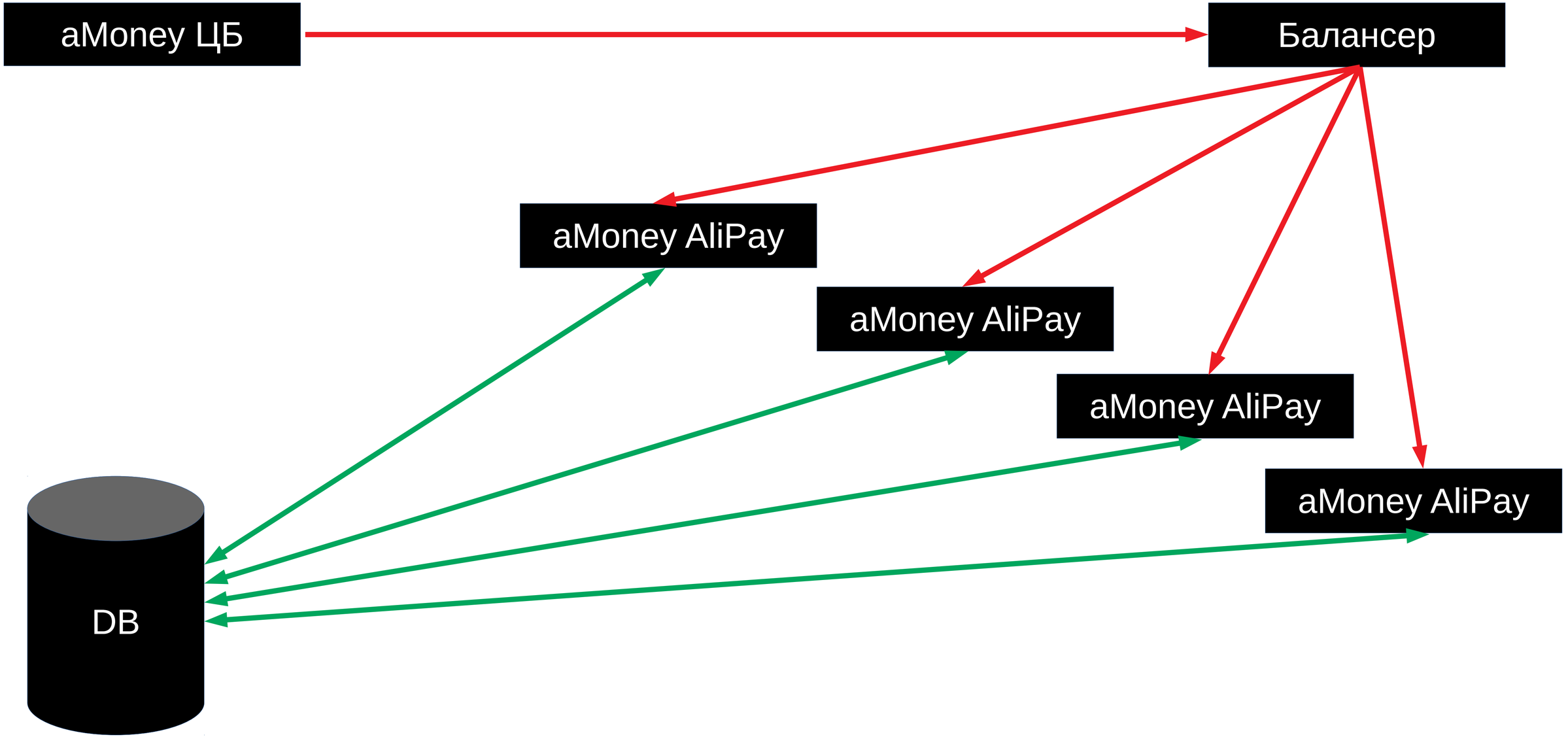
aMoney ЦБ



Микросервисы

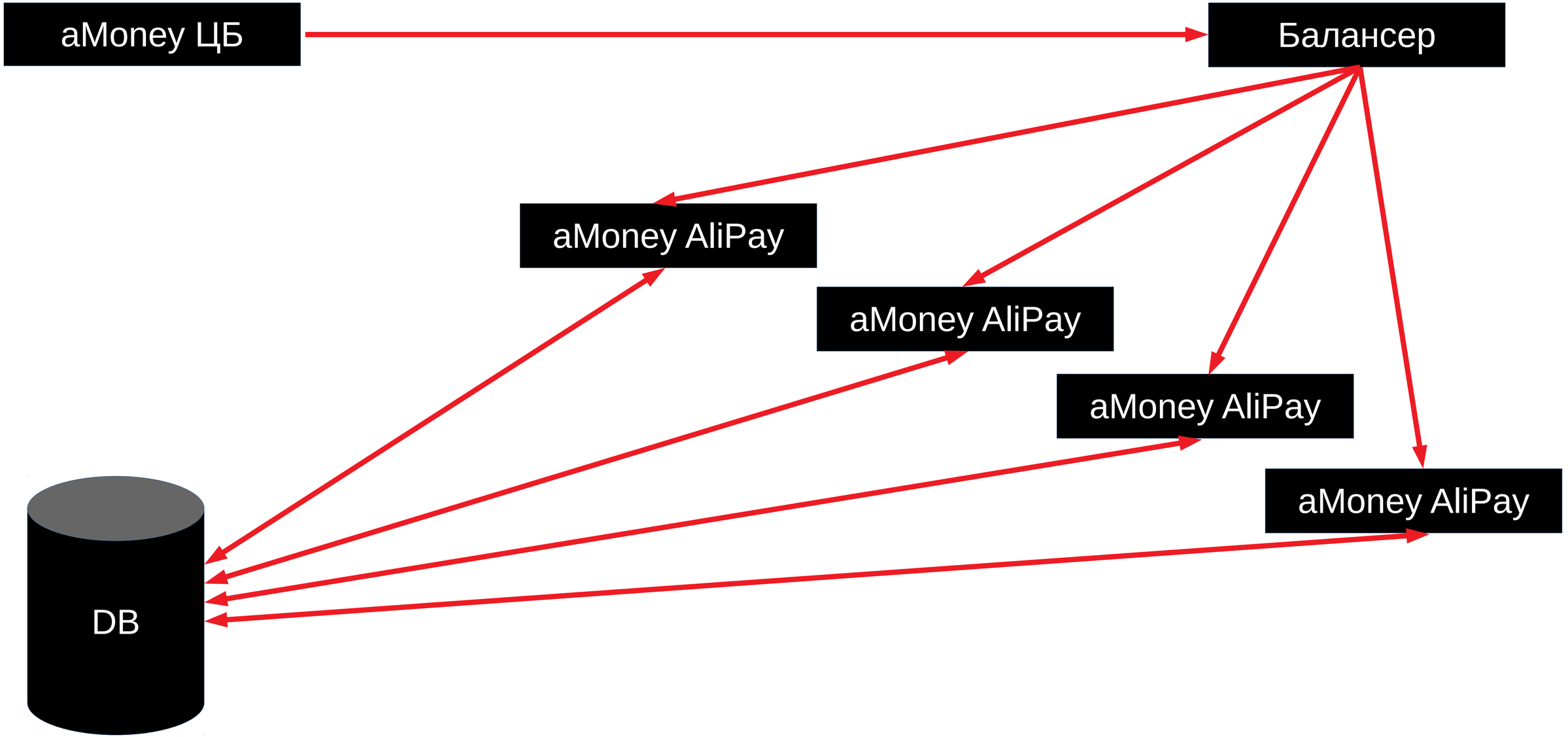


Микросервисы



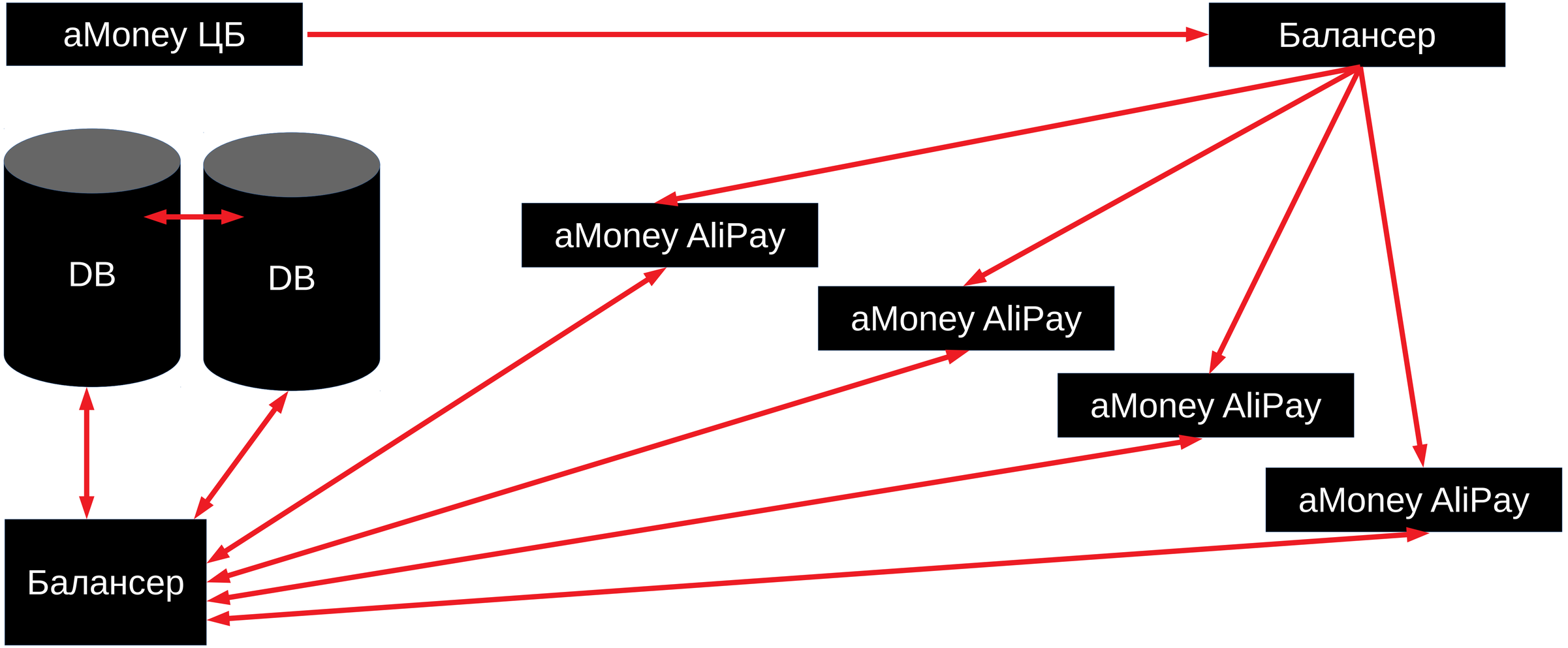
Микросервисы или нет

Микросервисы



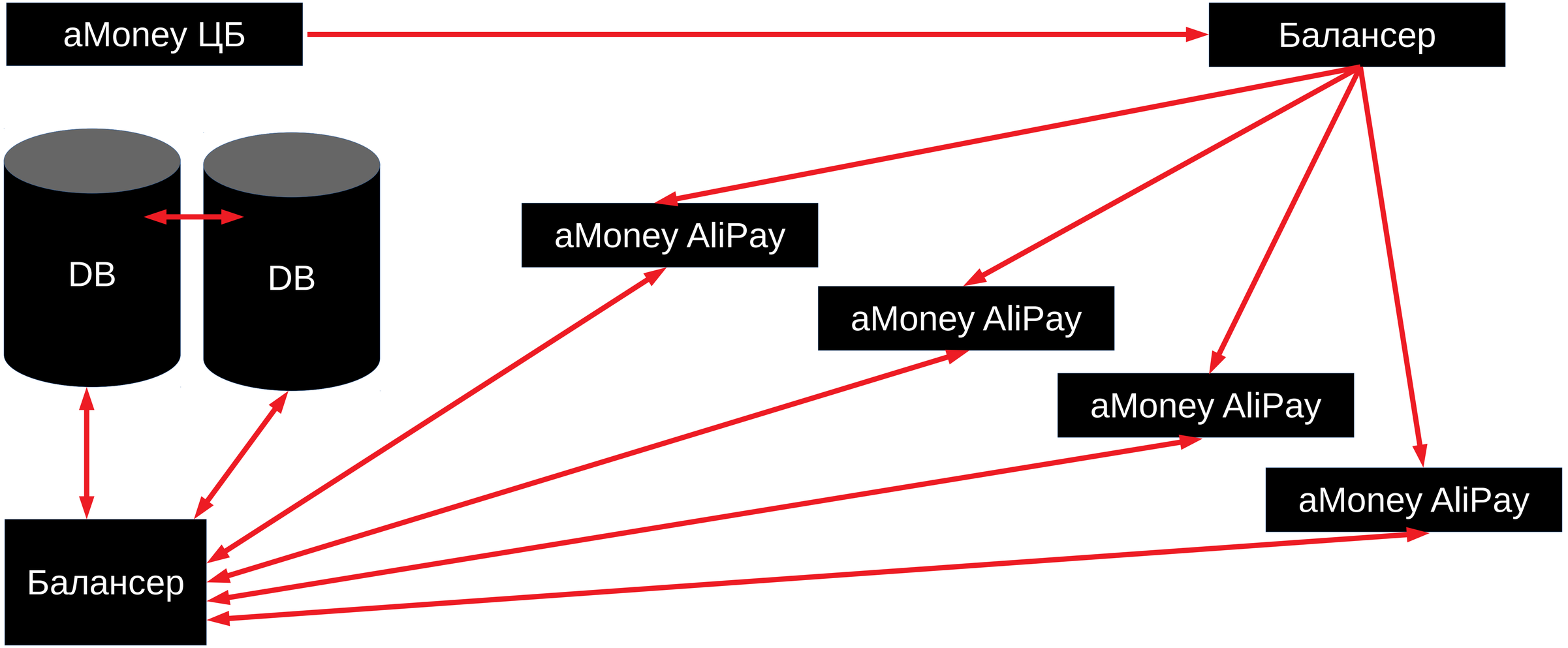
Микросервисы или нет

Микросервисы

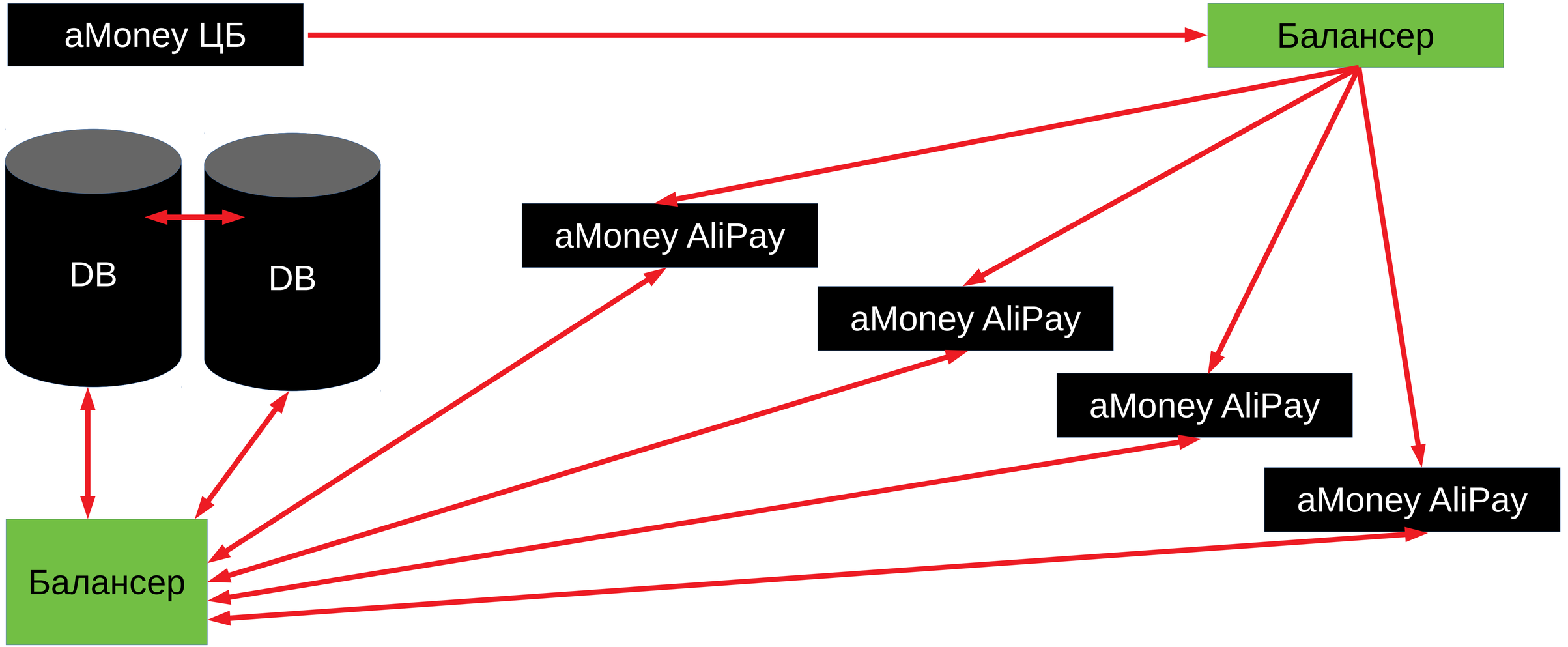


Как ускорить?

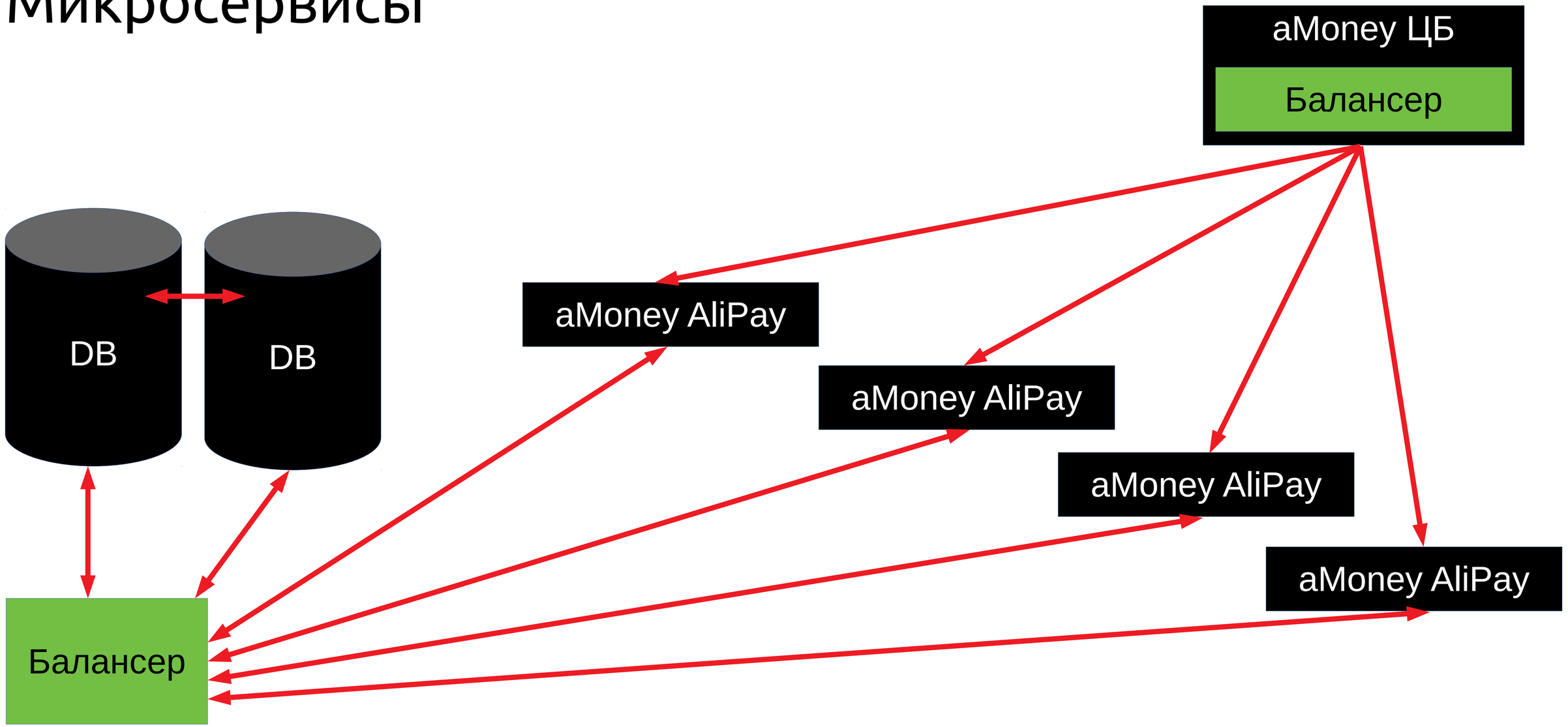
Микросервисы



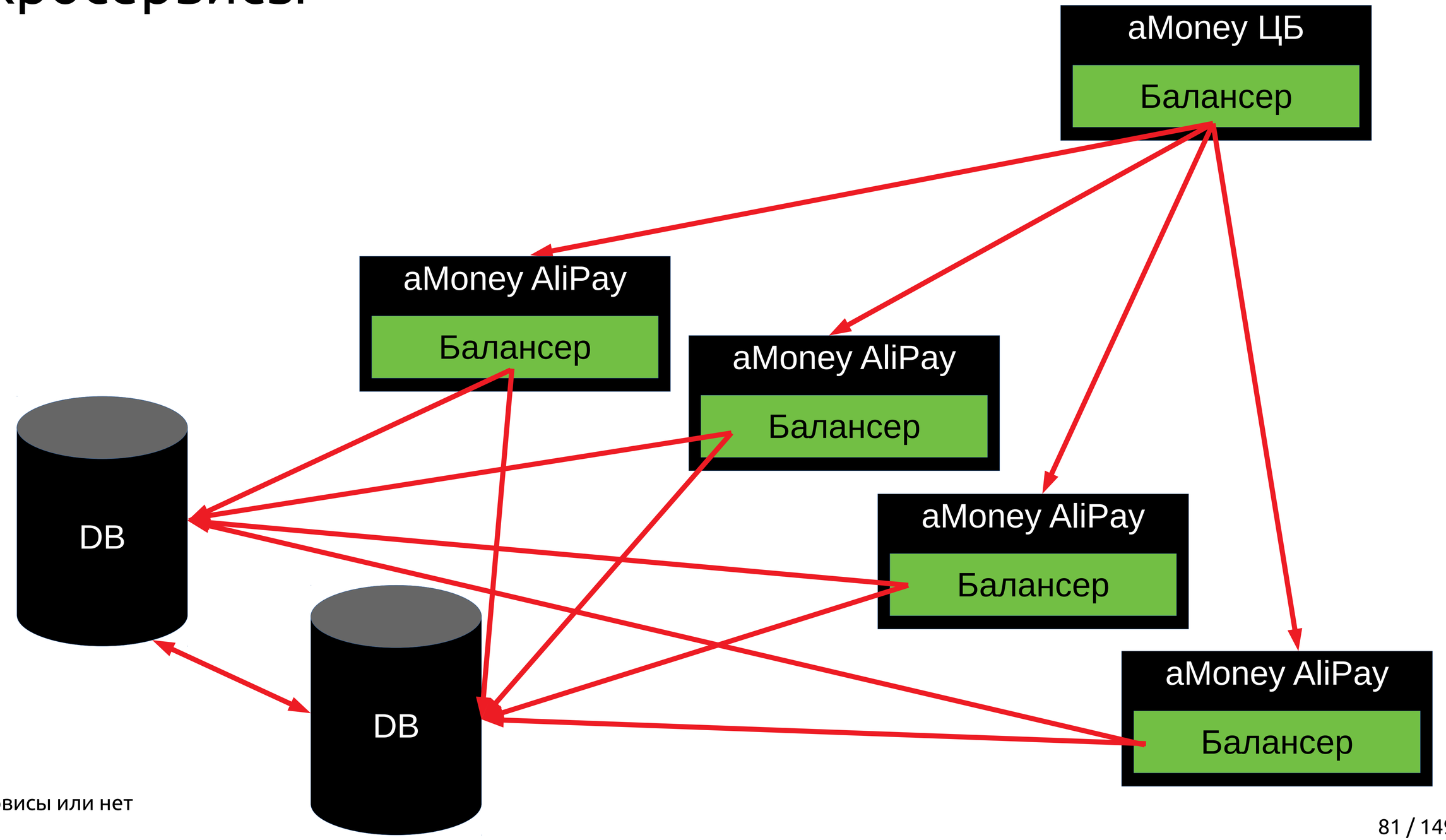
Микросервисы



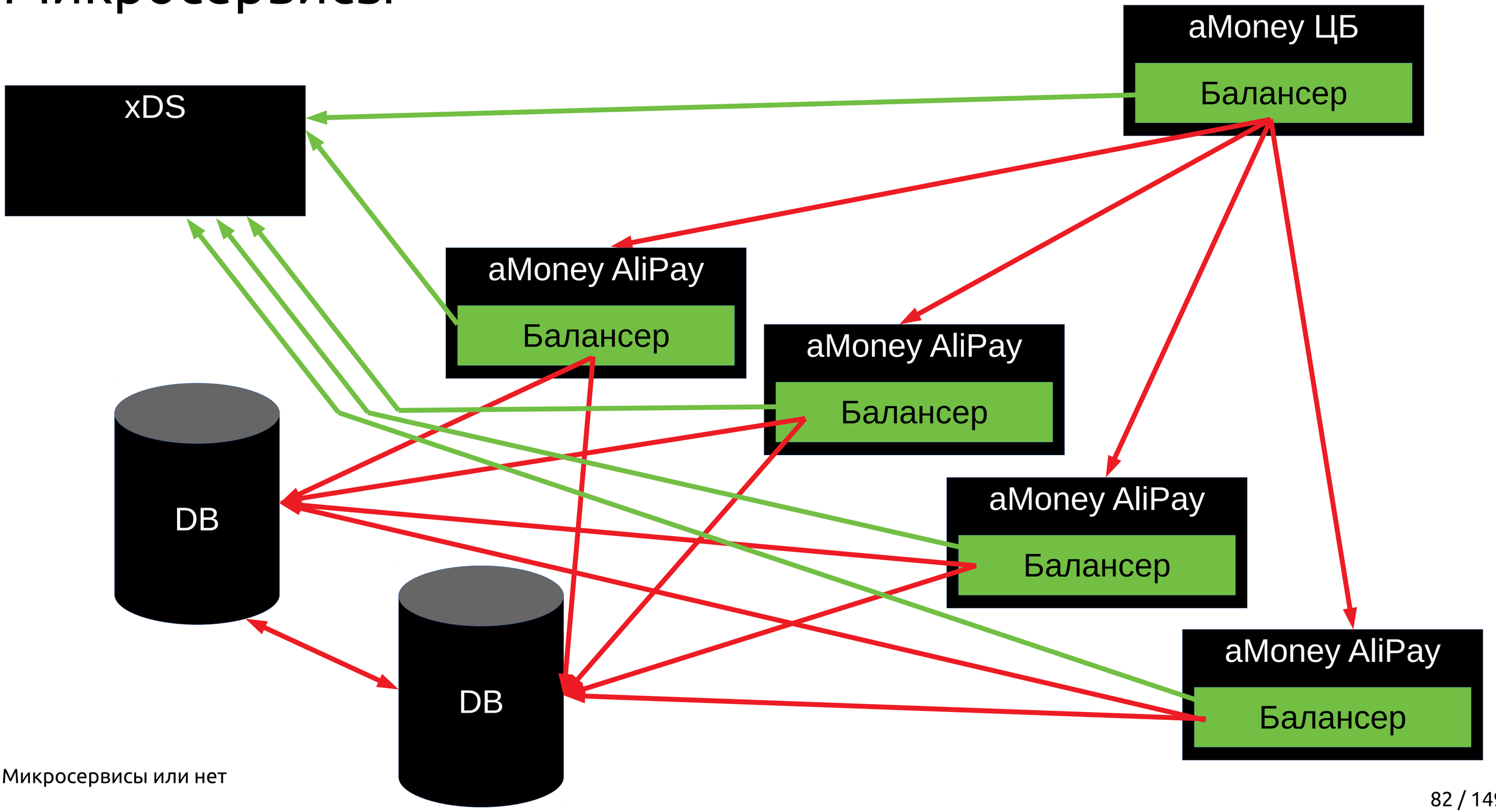
Микросервисы



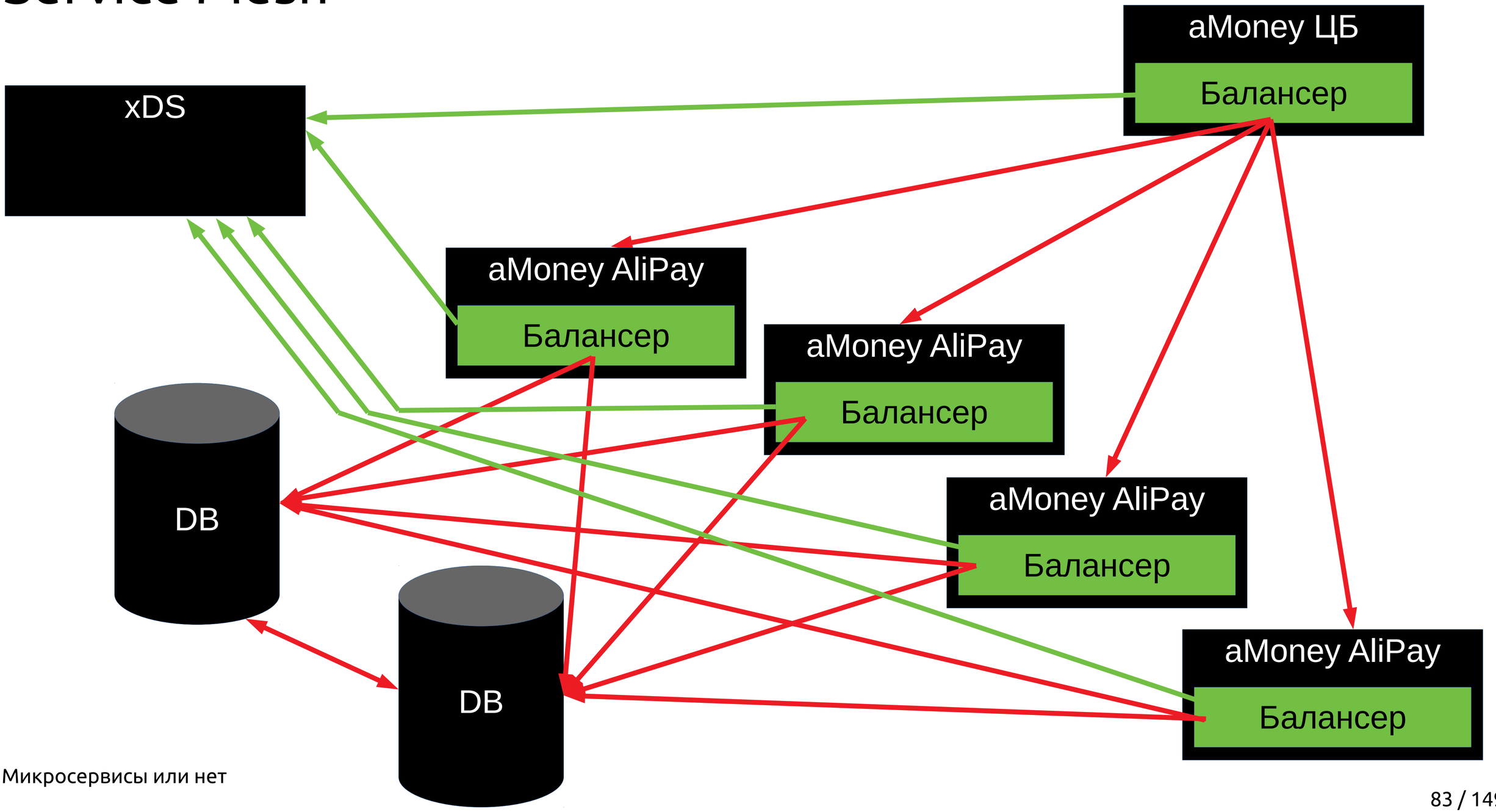
Микросервисы



Микросервисы



Service Mesh

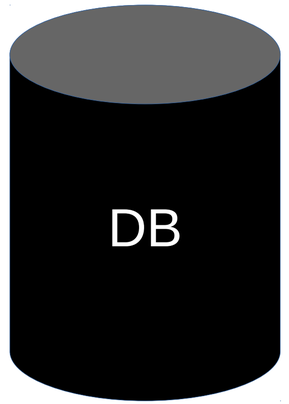


Как ещё сильнее ускорить?

Запрос

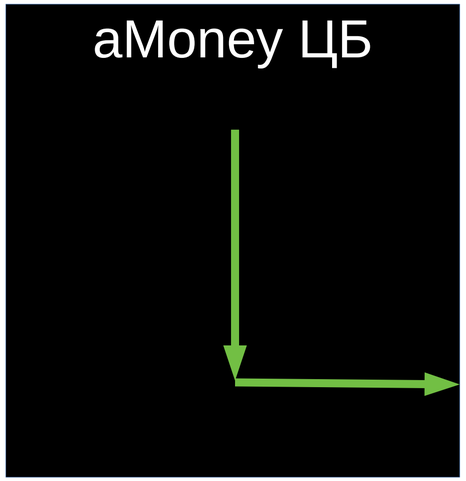
aMoney ЦБ

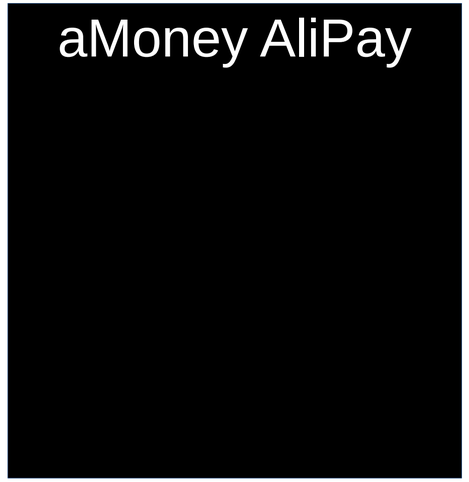
aMoney AliPay

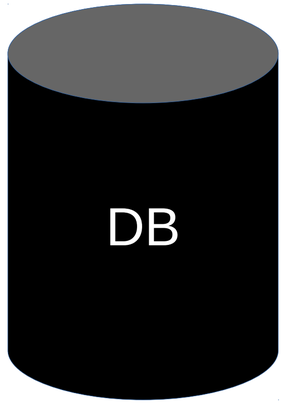


Запрос

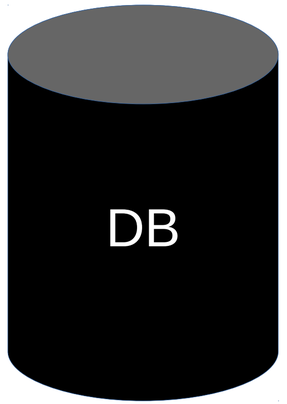
Запрос



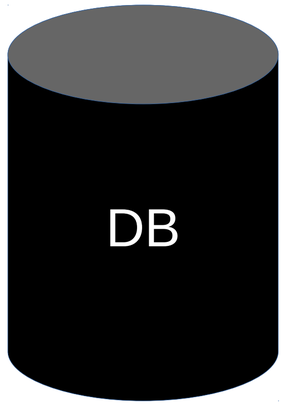




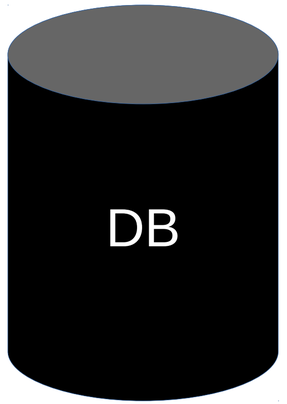
Запрос



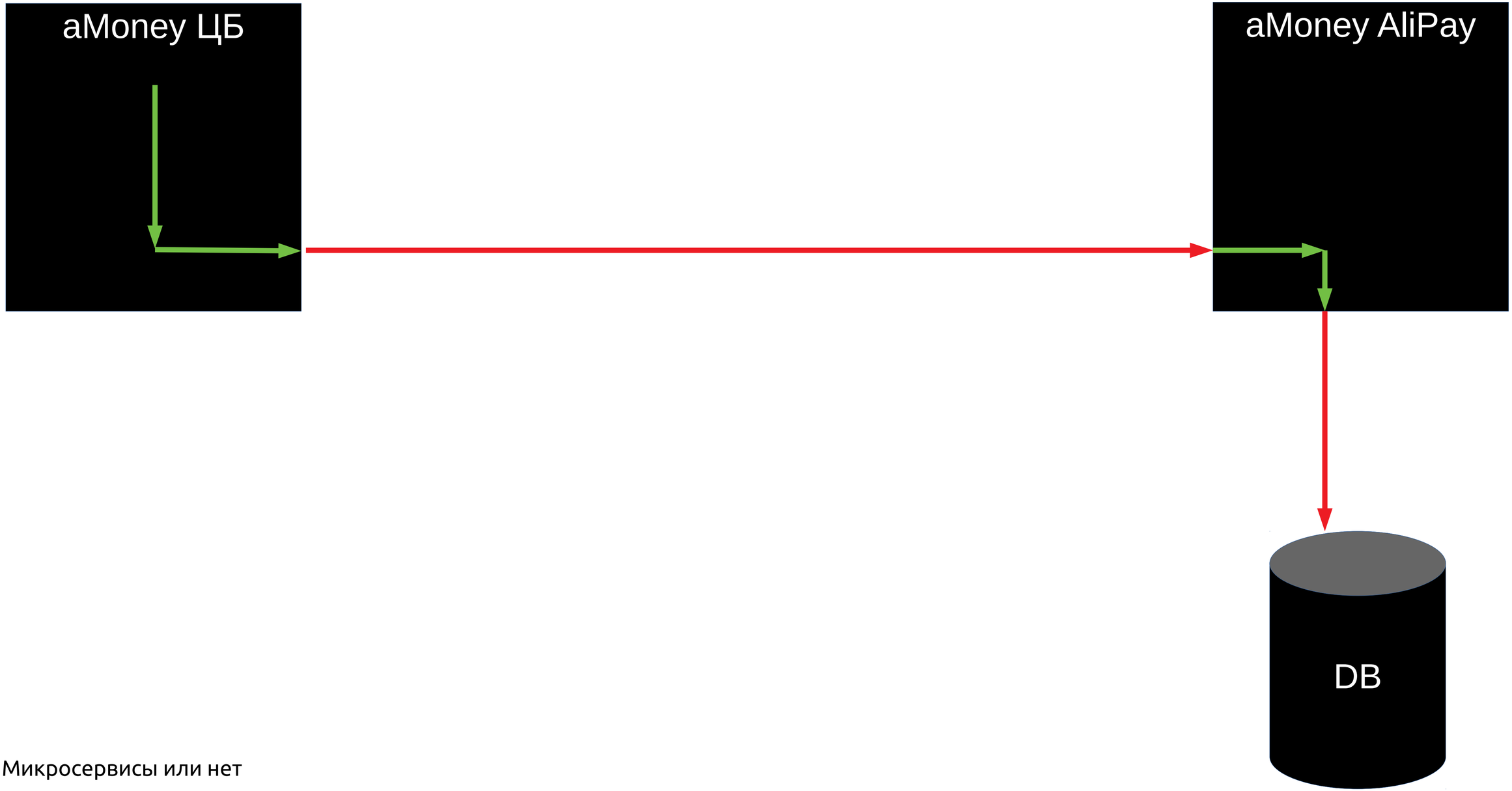
Запрос



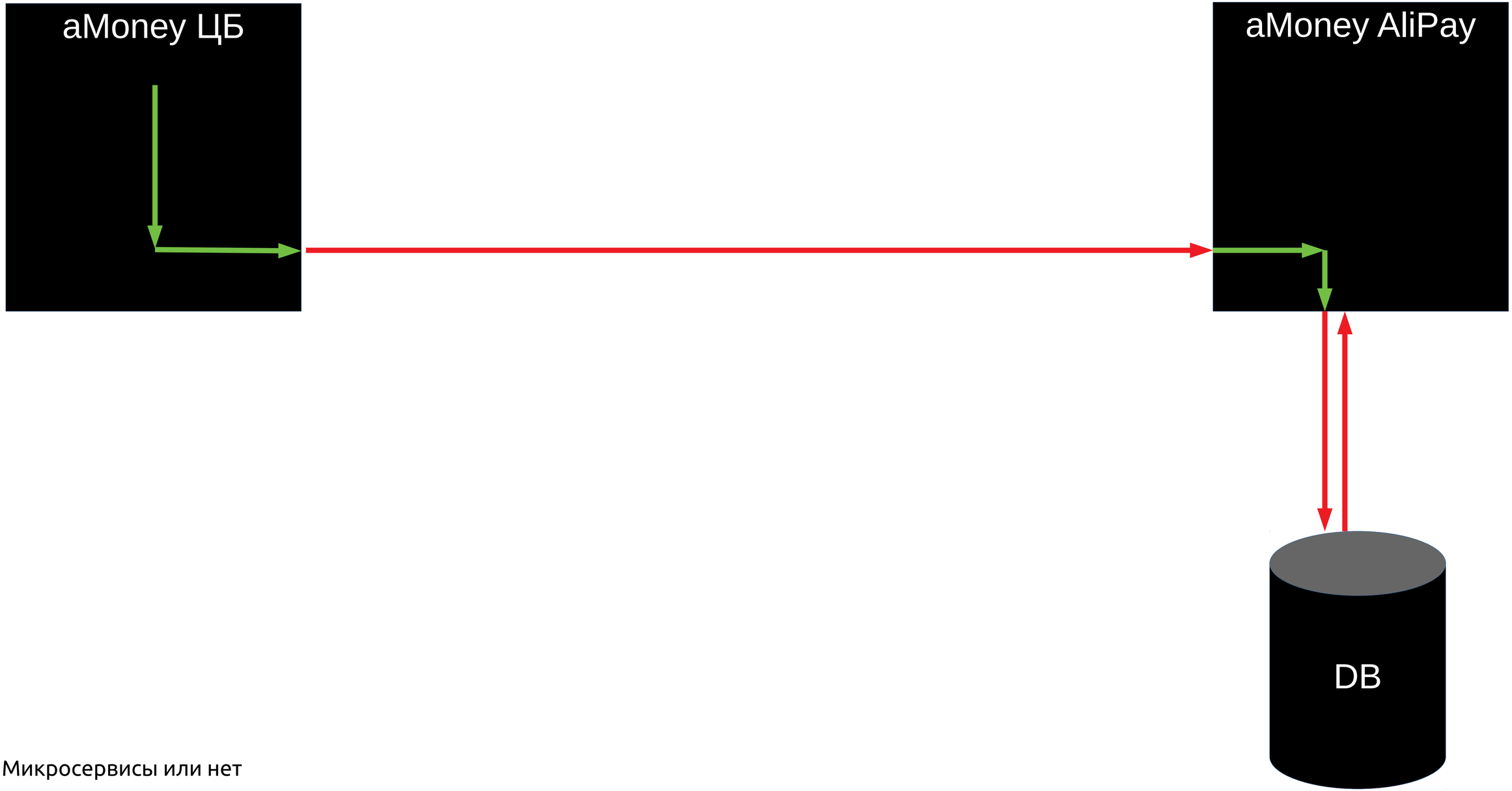
Запрос



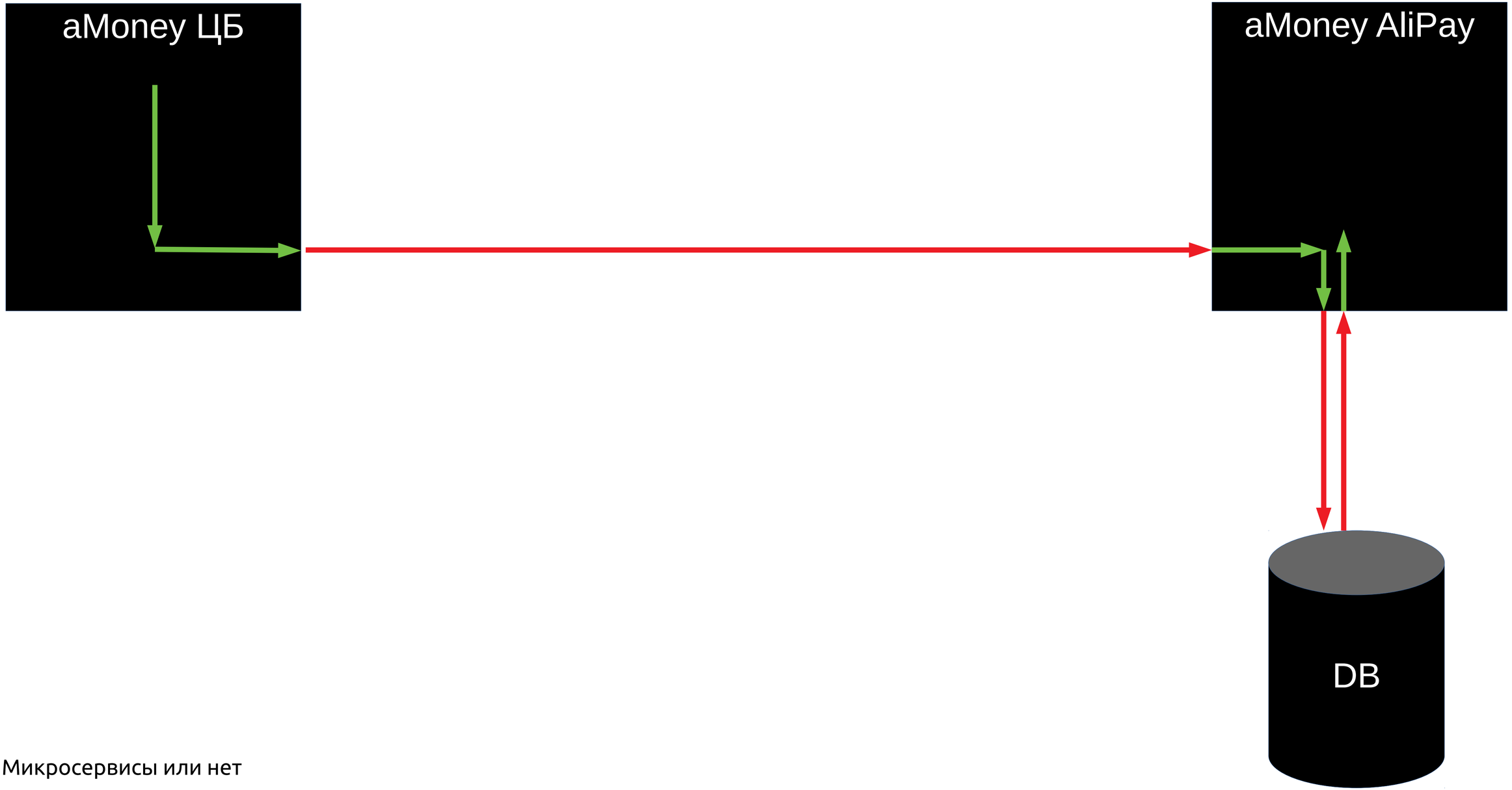
Запрос



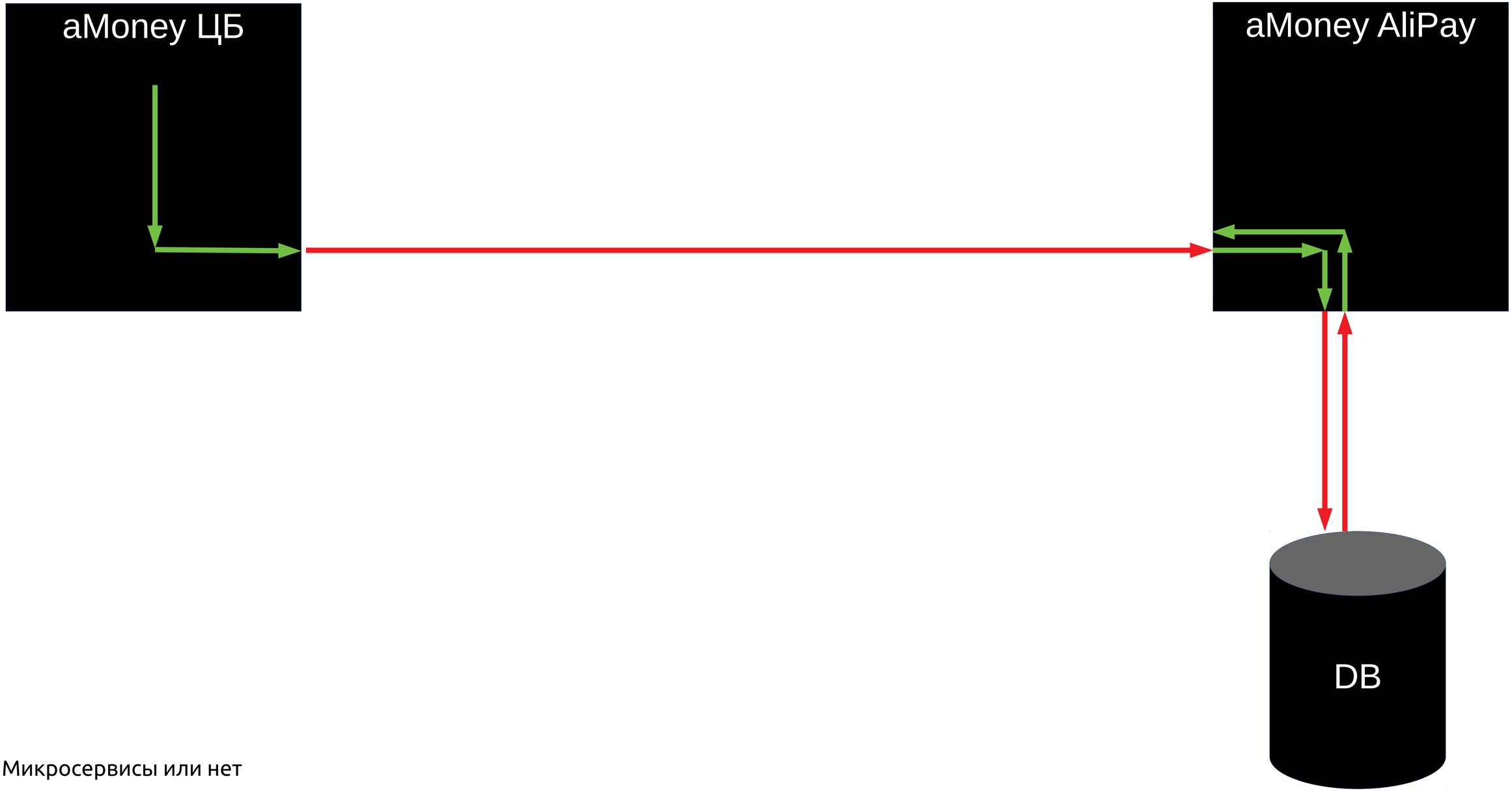
Запрос



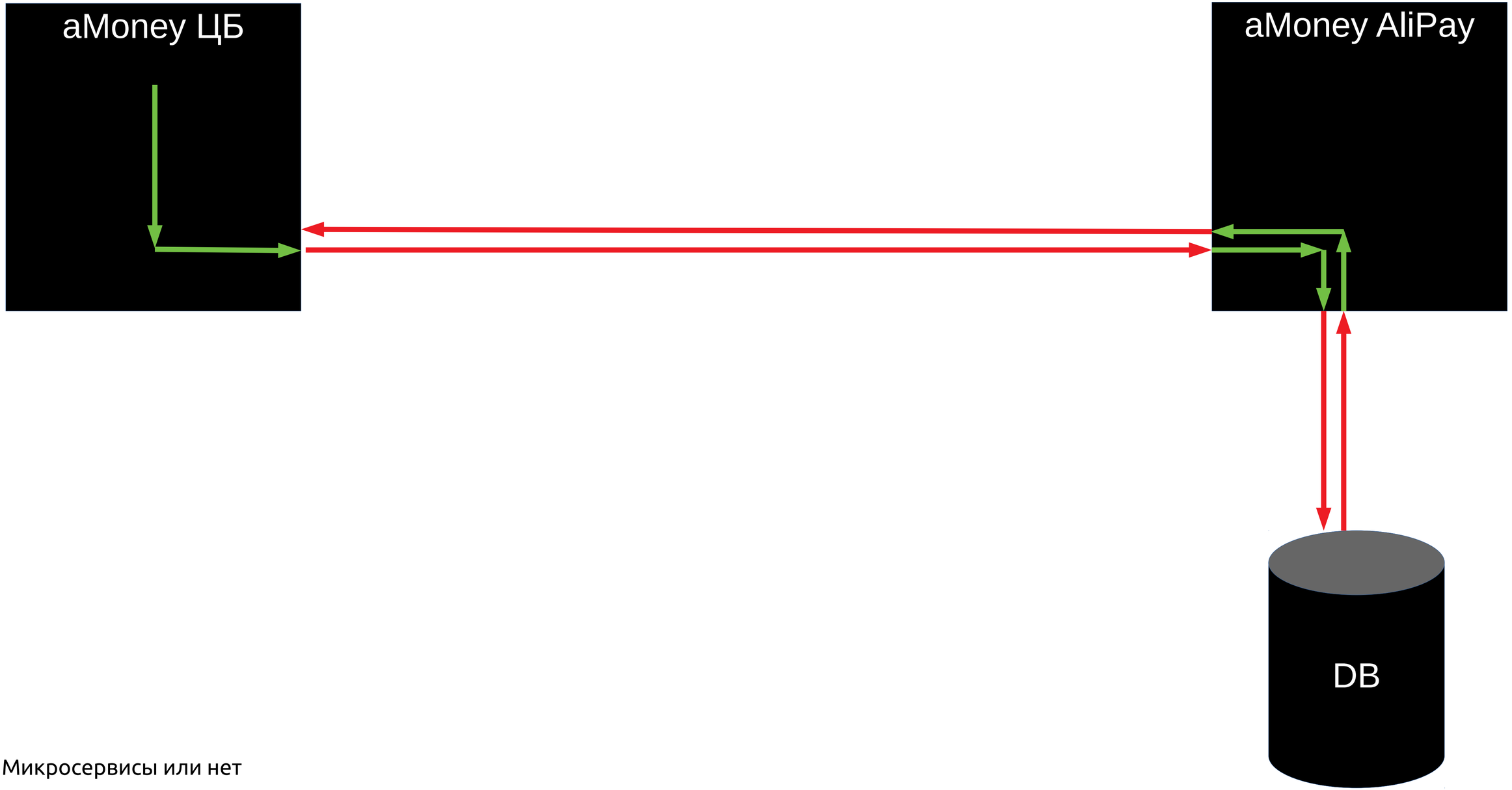
Запрос



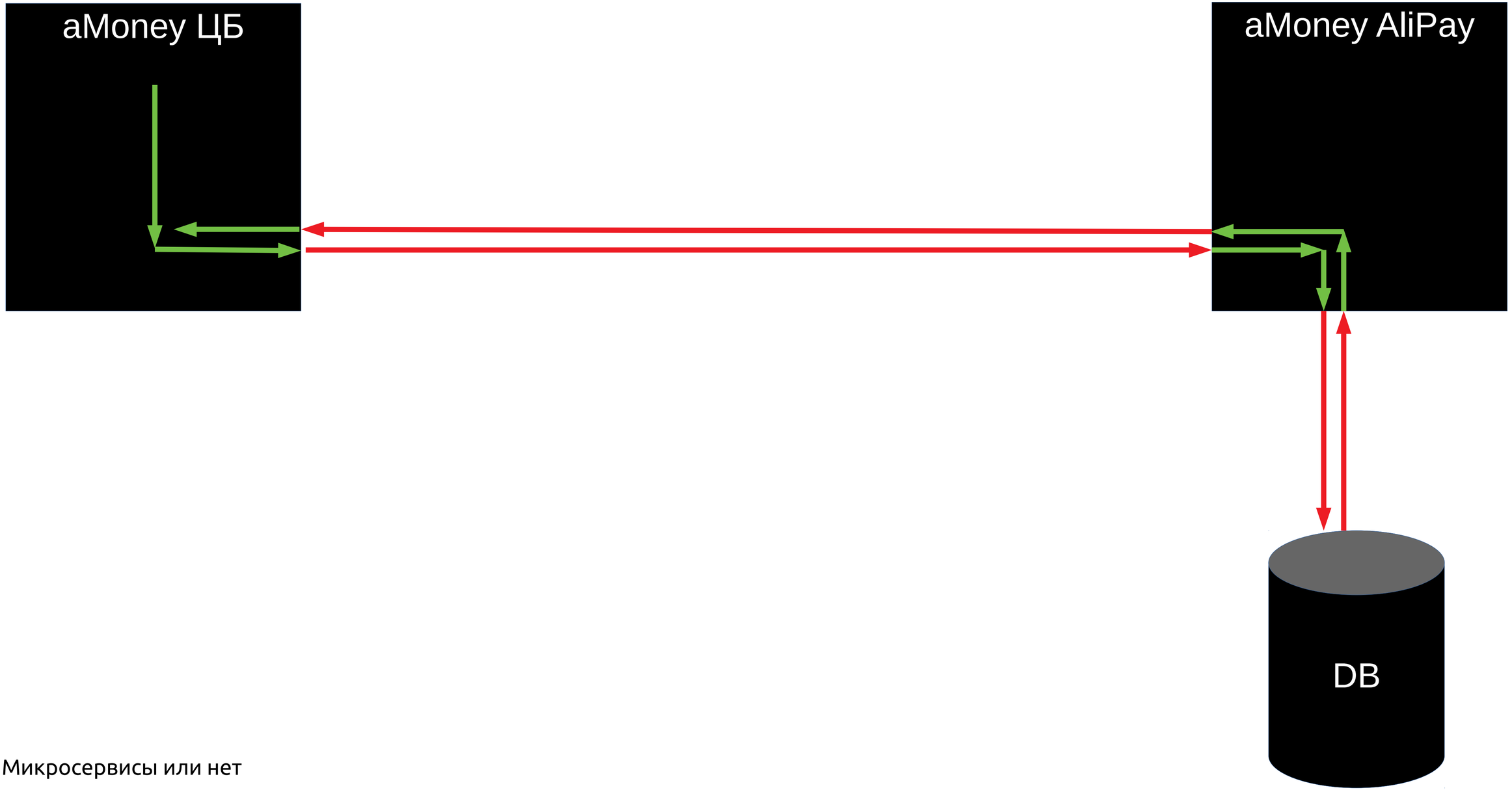
Запрос



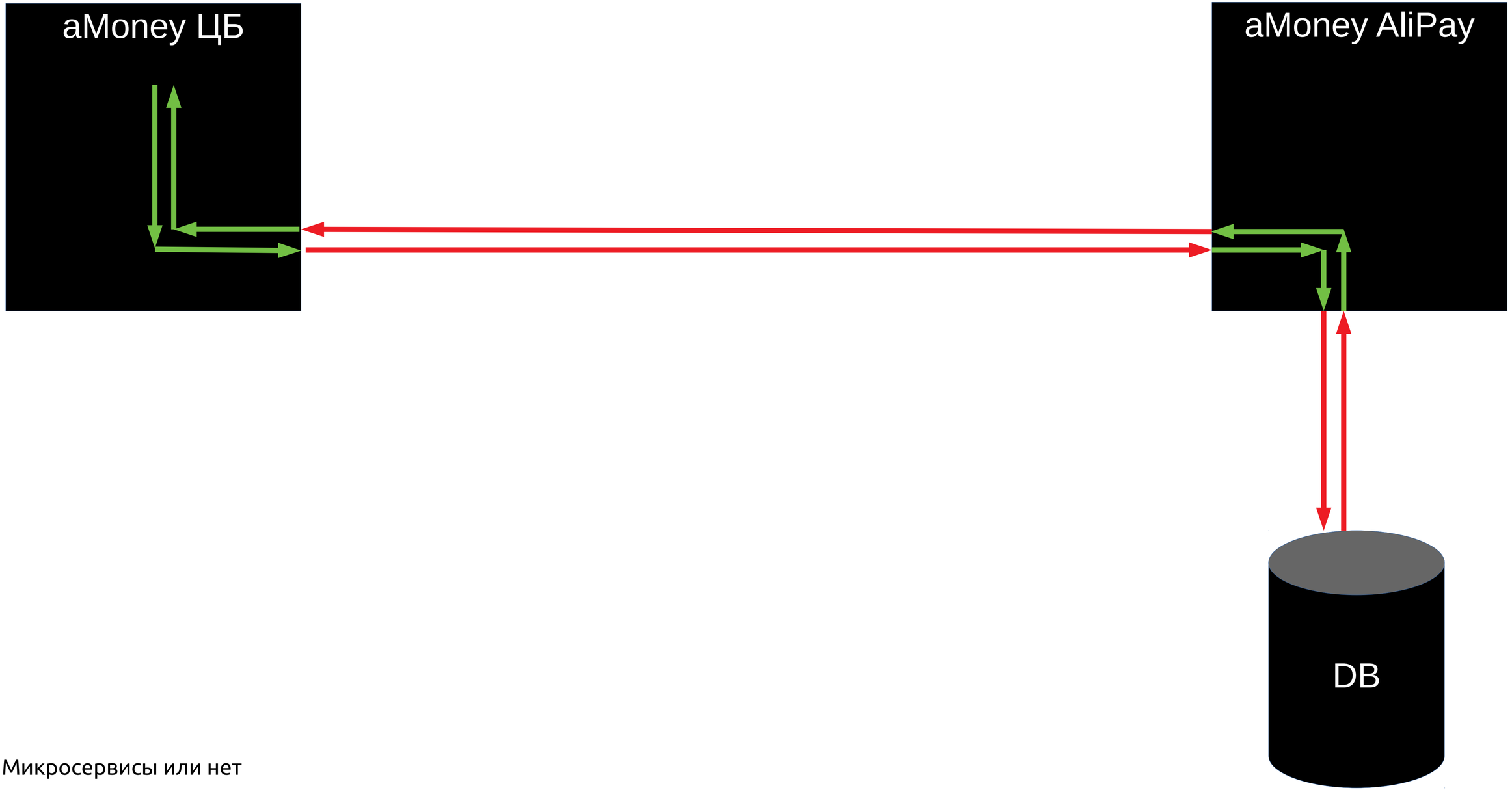
Запрос



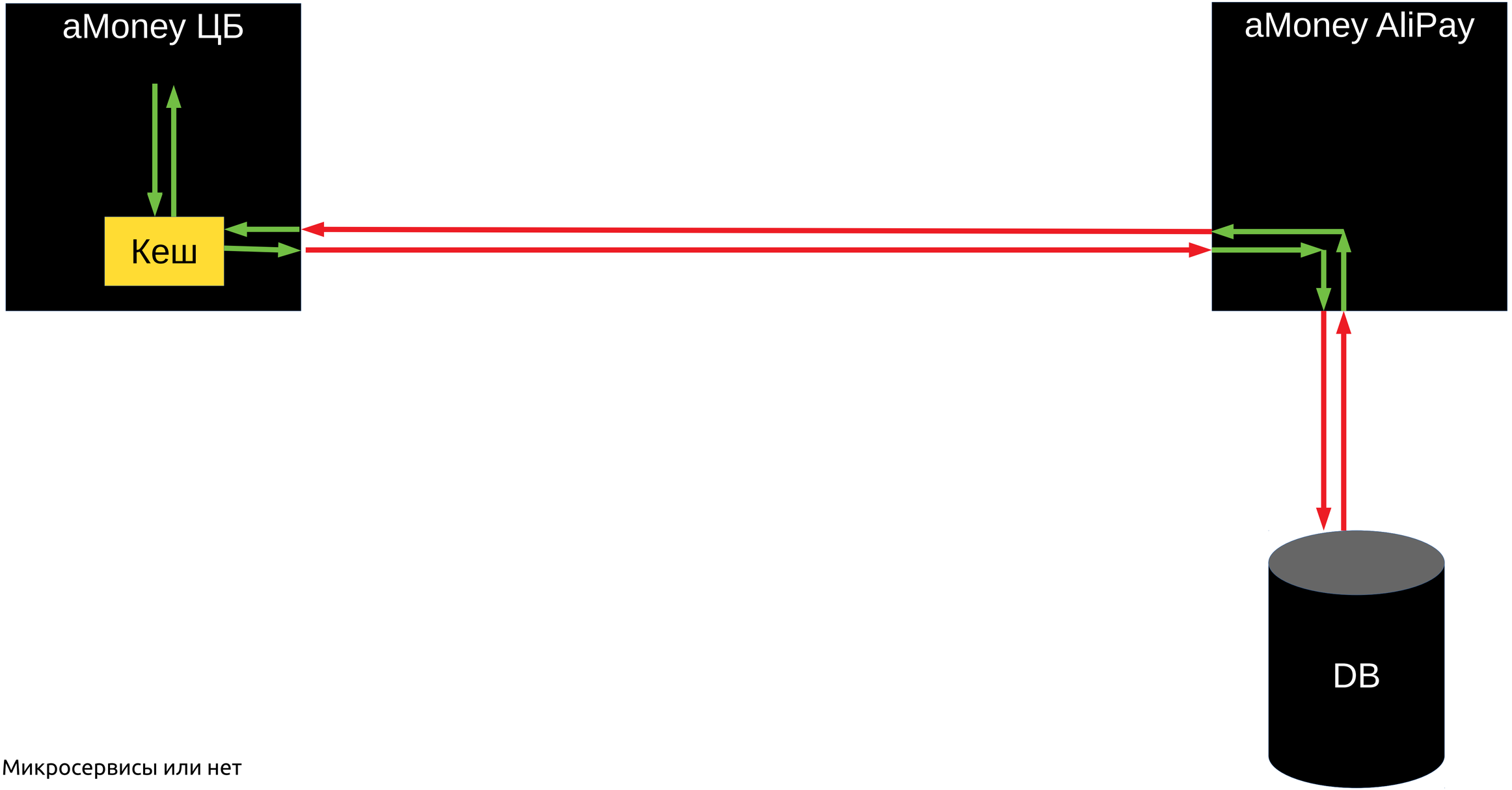
Запрос



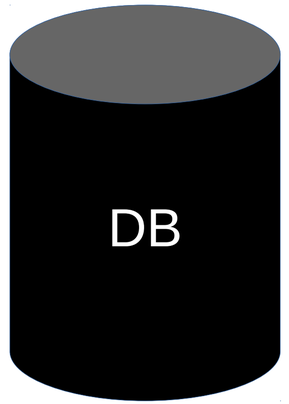
Запрос



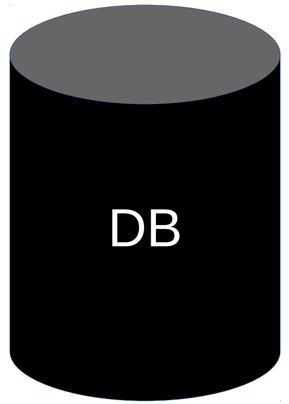
Кеши



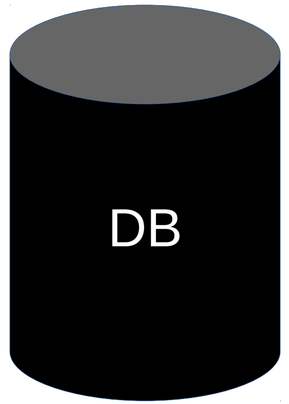
Кеши



Кеши



Кеши



Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

Минусы:

- Дорогая передача данных между модулями
- Обязательное версионирование и поддержка старых версий
- Большие траты на железо

Плюсы/минусы микросервисов для большой команды

Плюсы:

- Надёжность
- Быстрый деплой
- Быстрая сборка
- Тесное общение разработчиков модуля

Минусы:

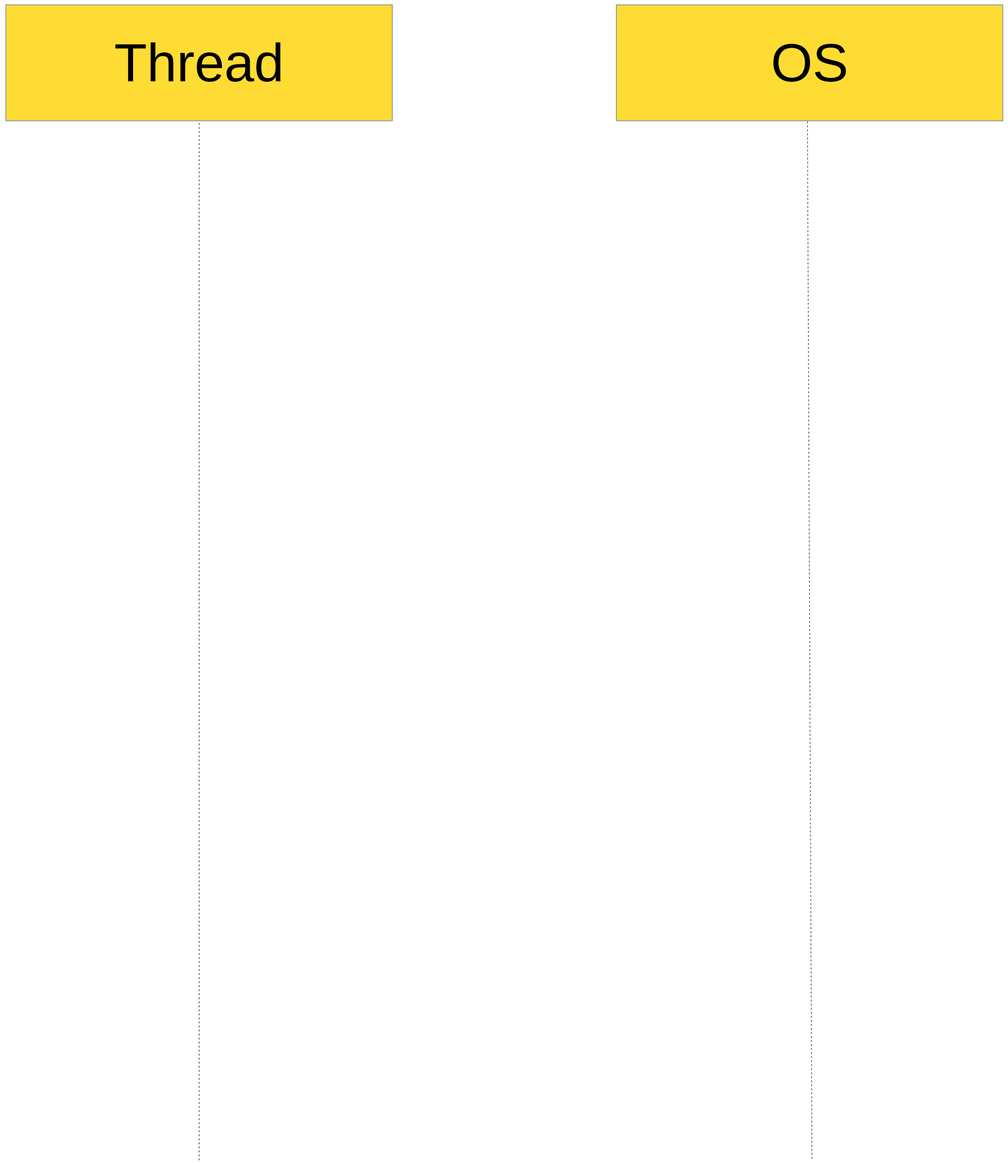
- ~~Дорогая передача данных между модулями~~
- Обязательное версионирование и поддержка старых версий
- Большие траты на железо

IO-bound и C++

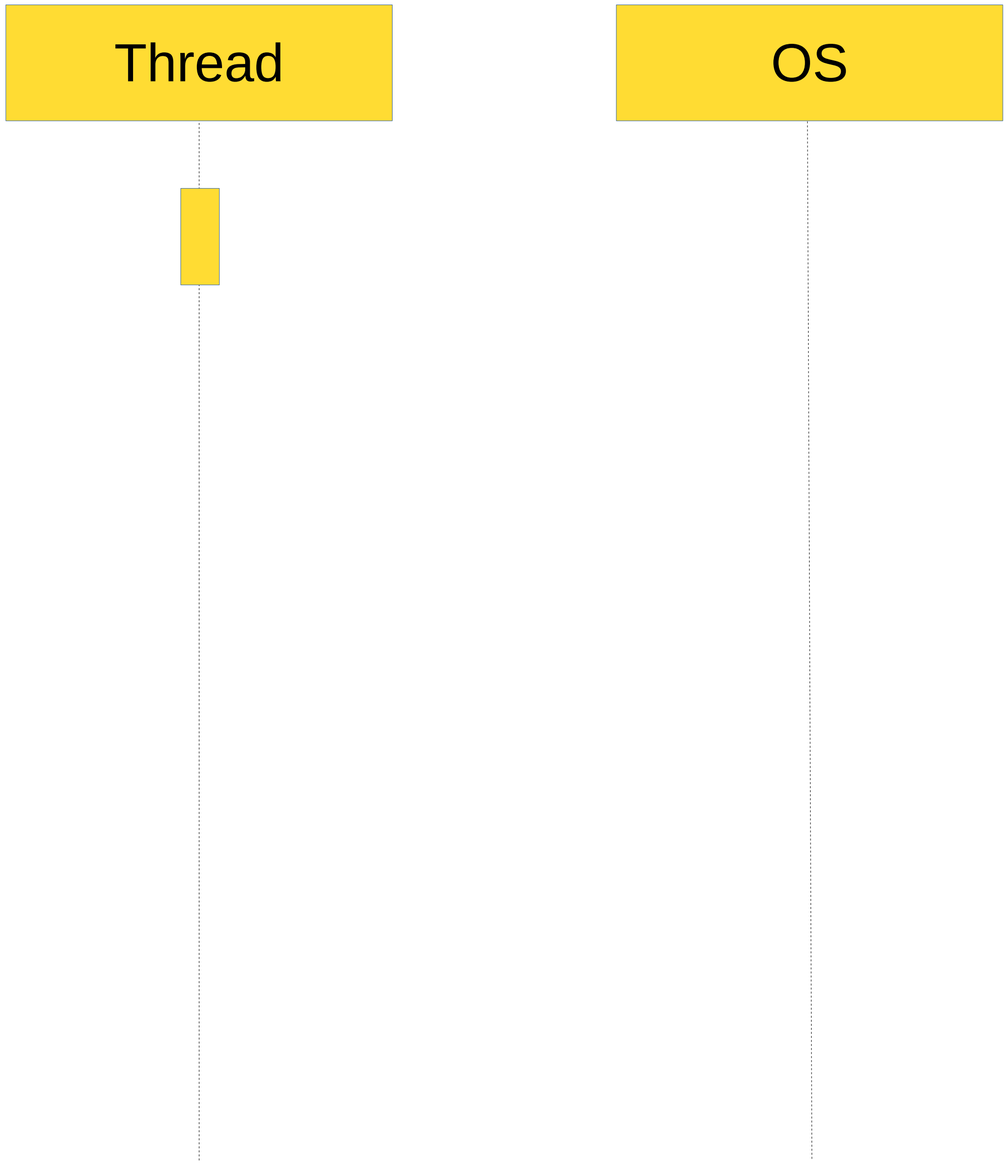
C++

```
pg_cluster_->Execute(storages::postgres::ClusterHostType::kMaster,
                     "DELETE FROM key_value_table WHERE key=$1", key);
```

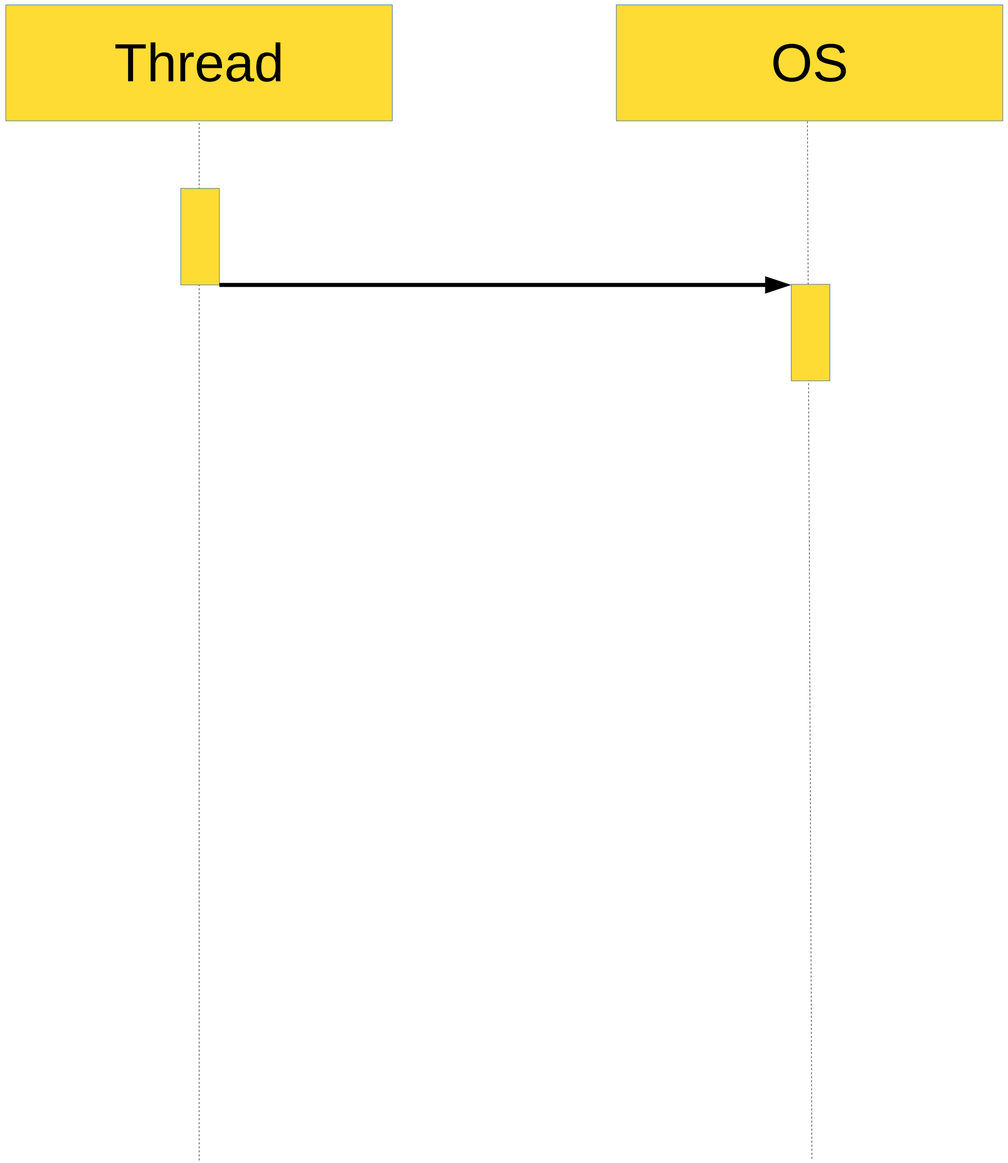
C++



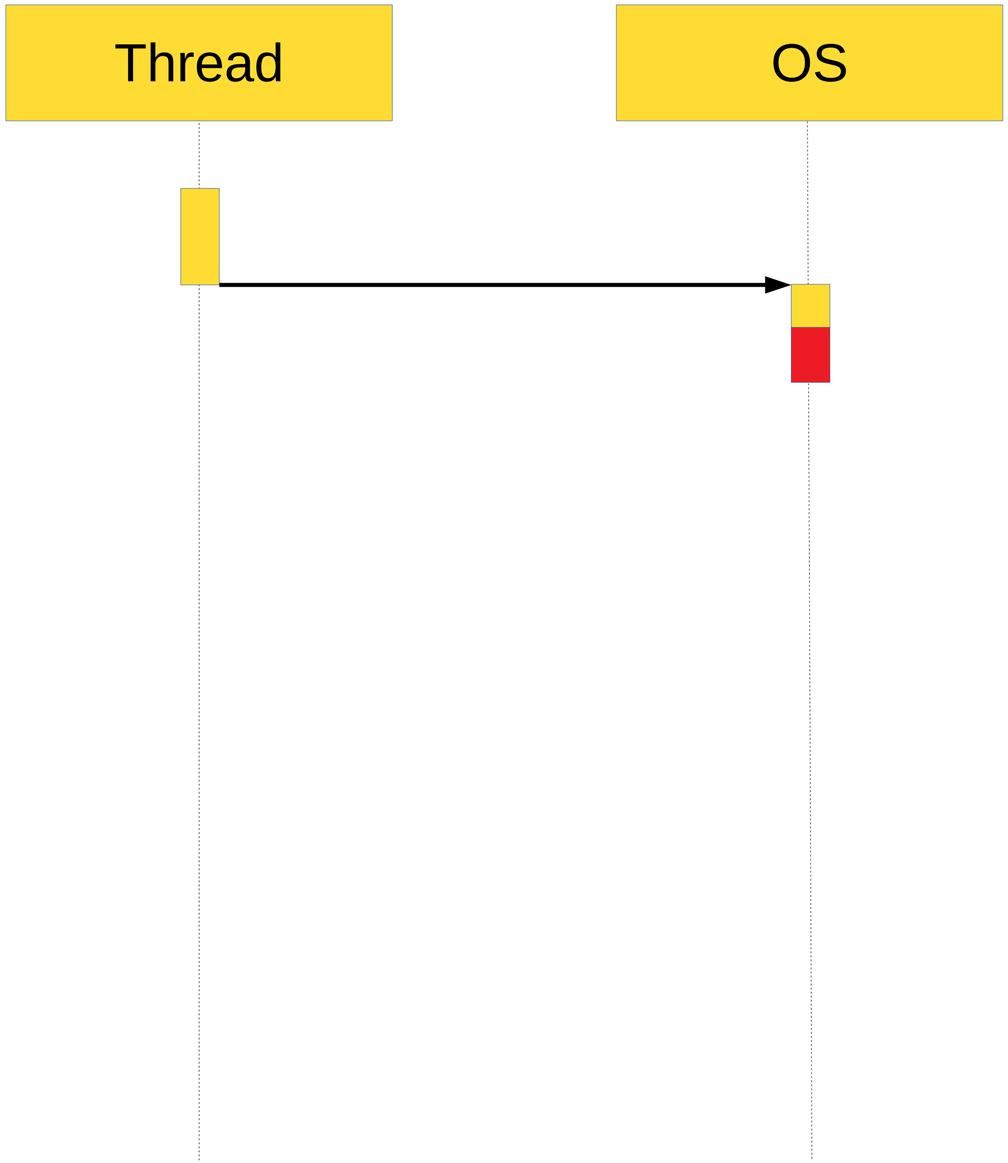
C++



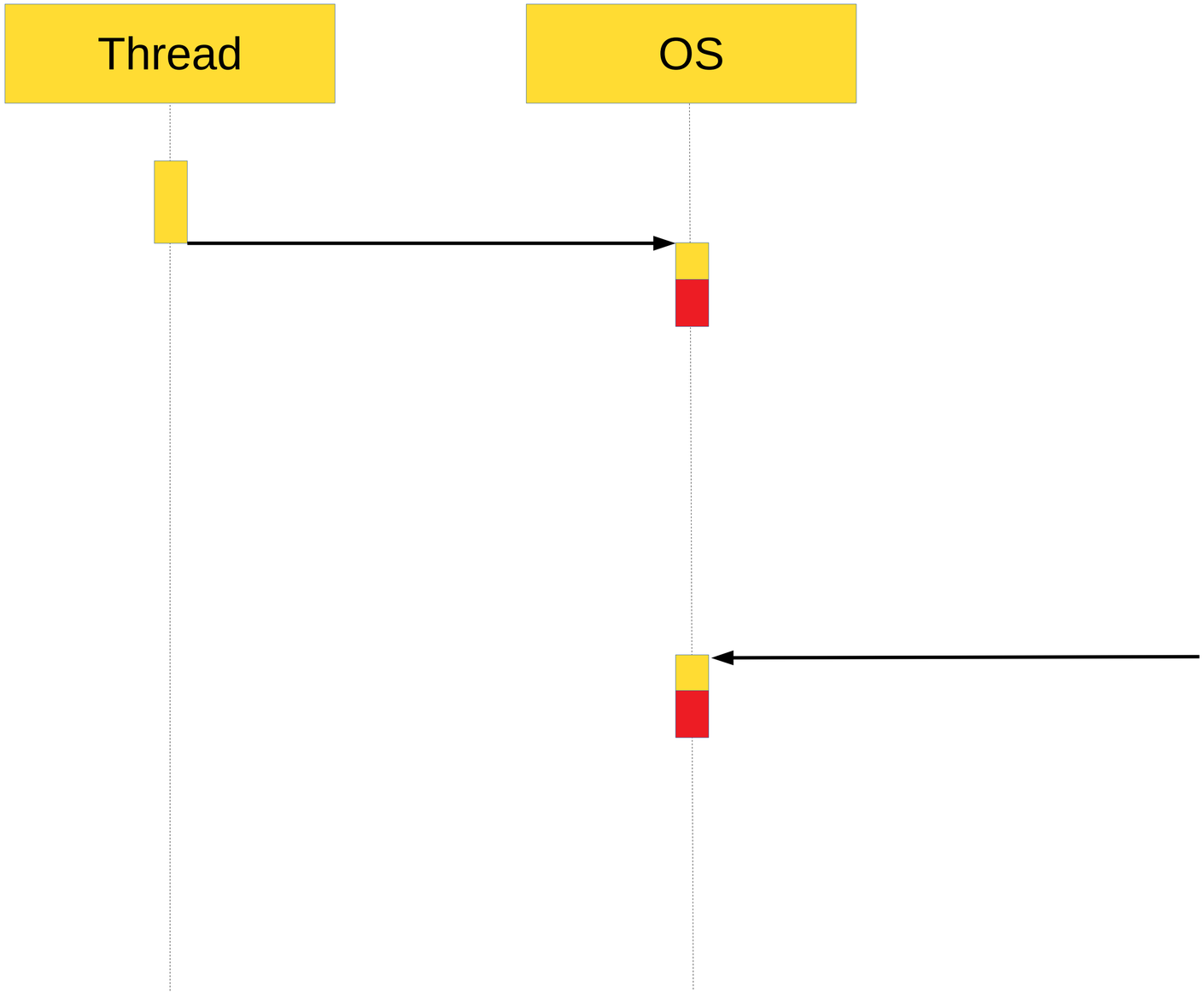
C++



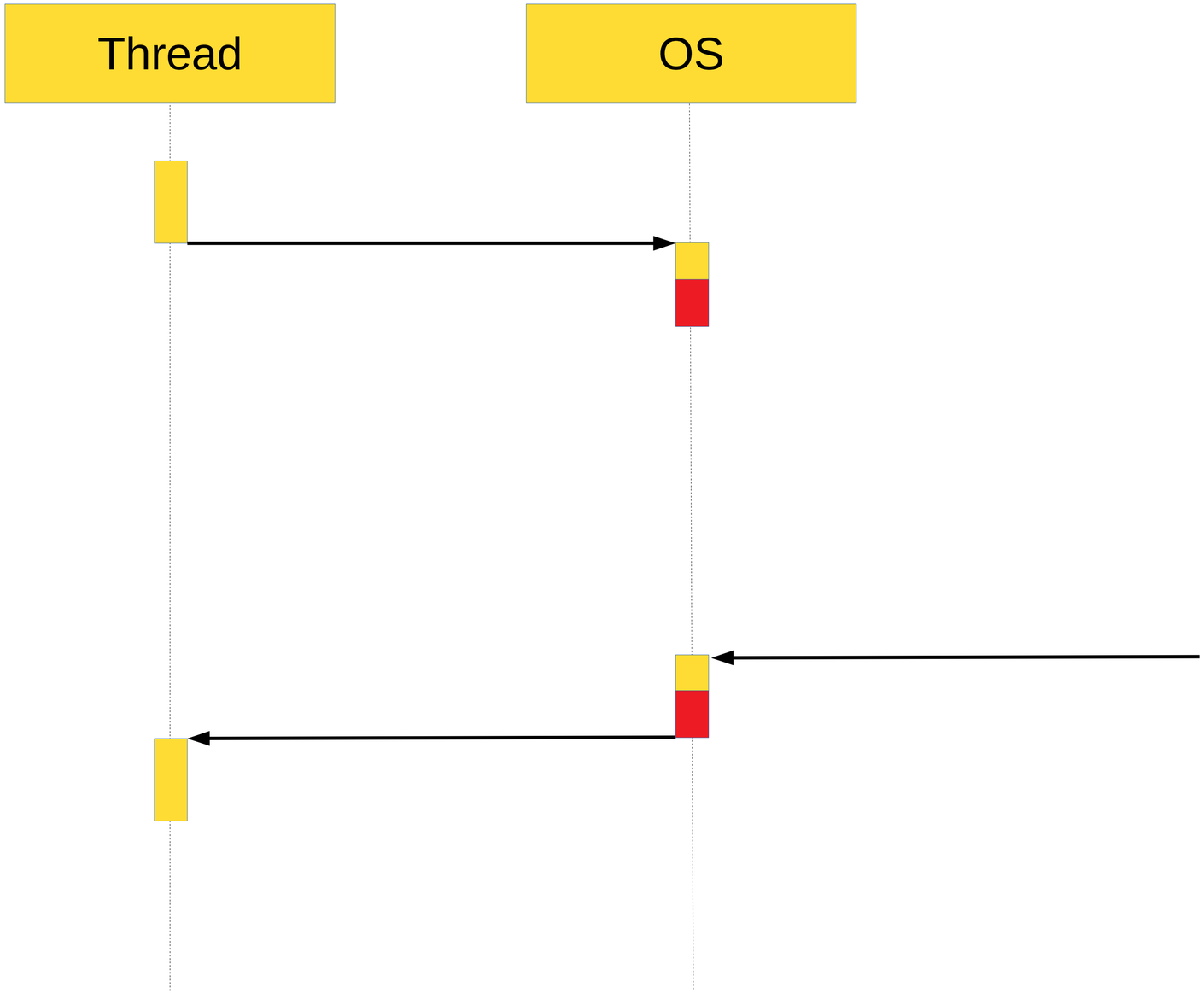
C++



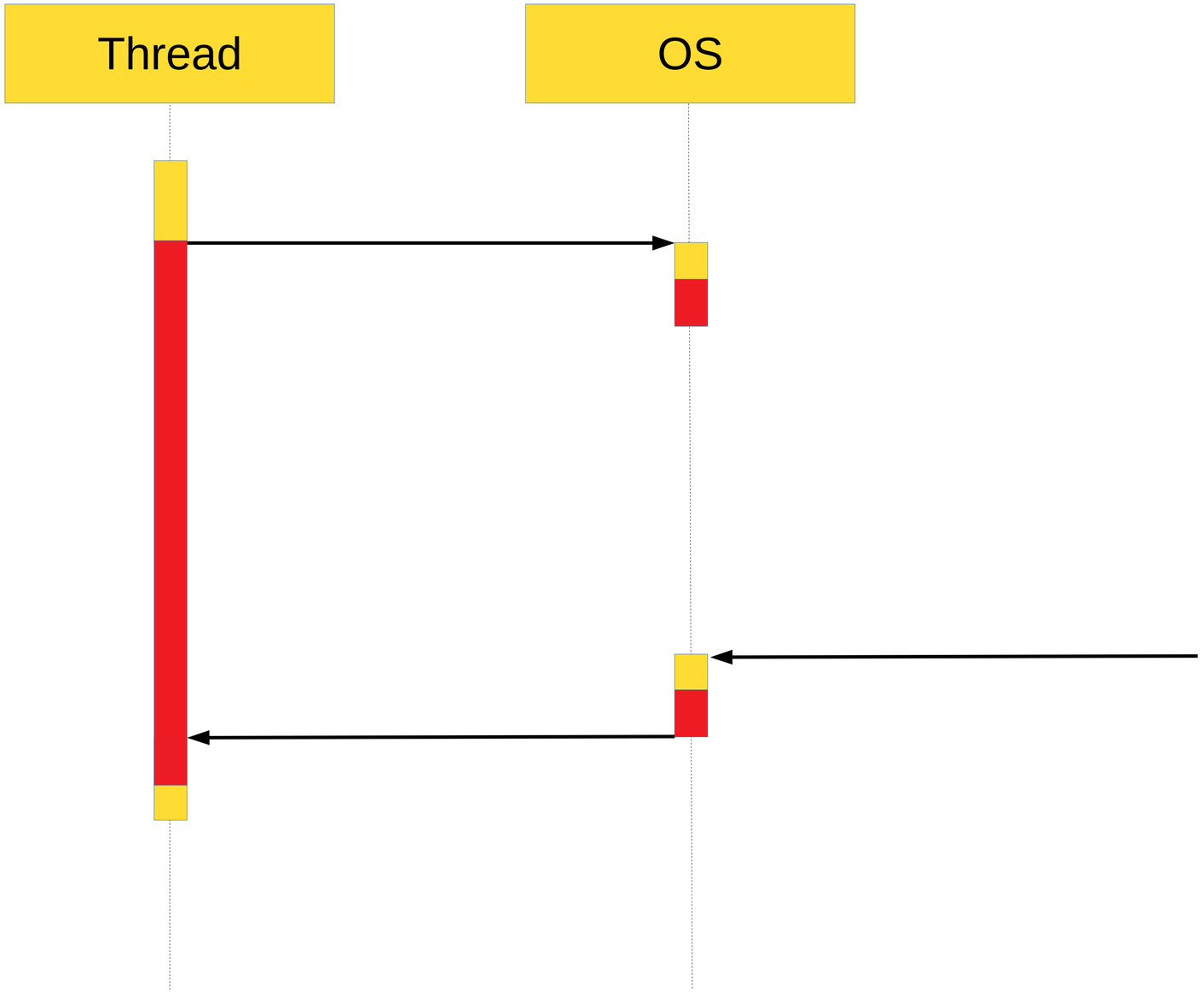
C++



C++



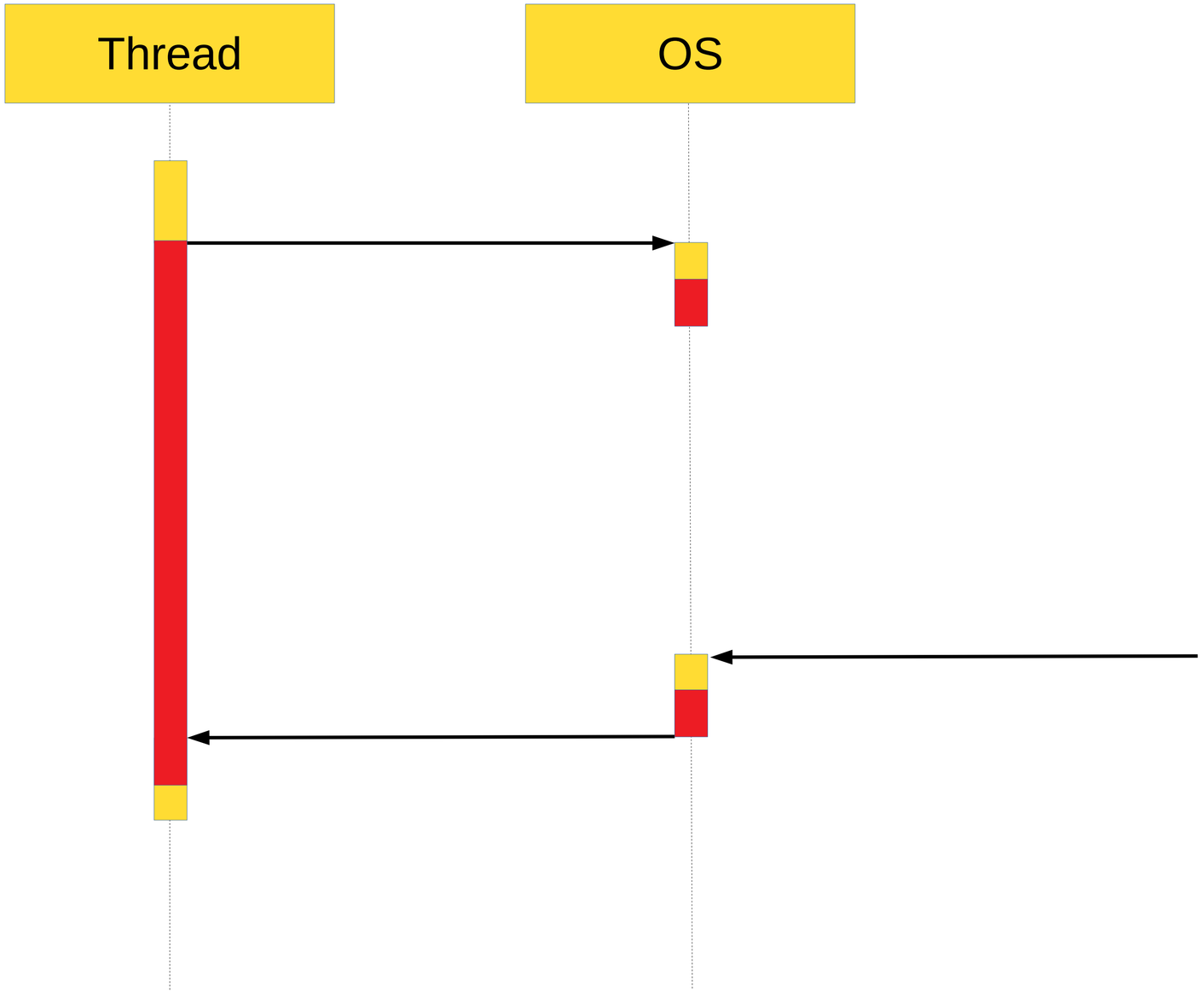
C++



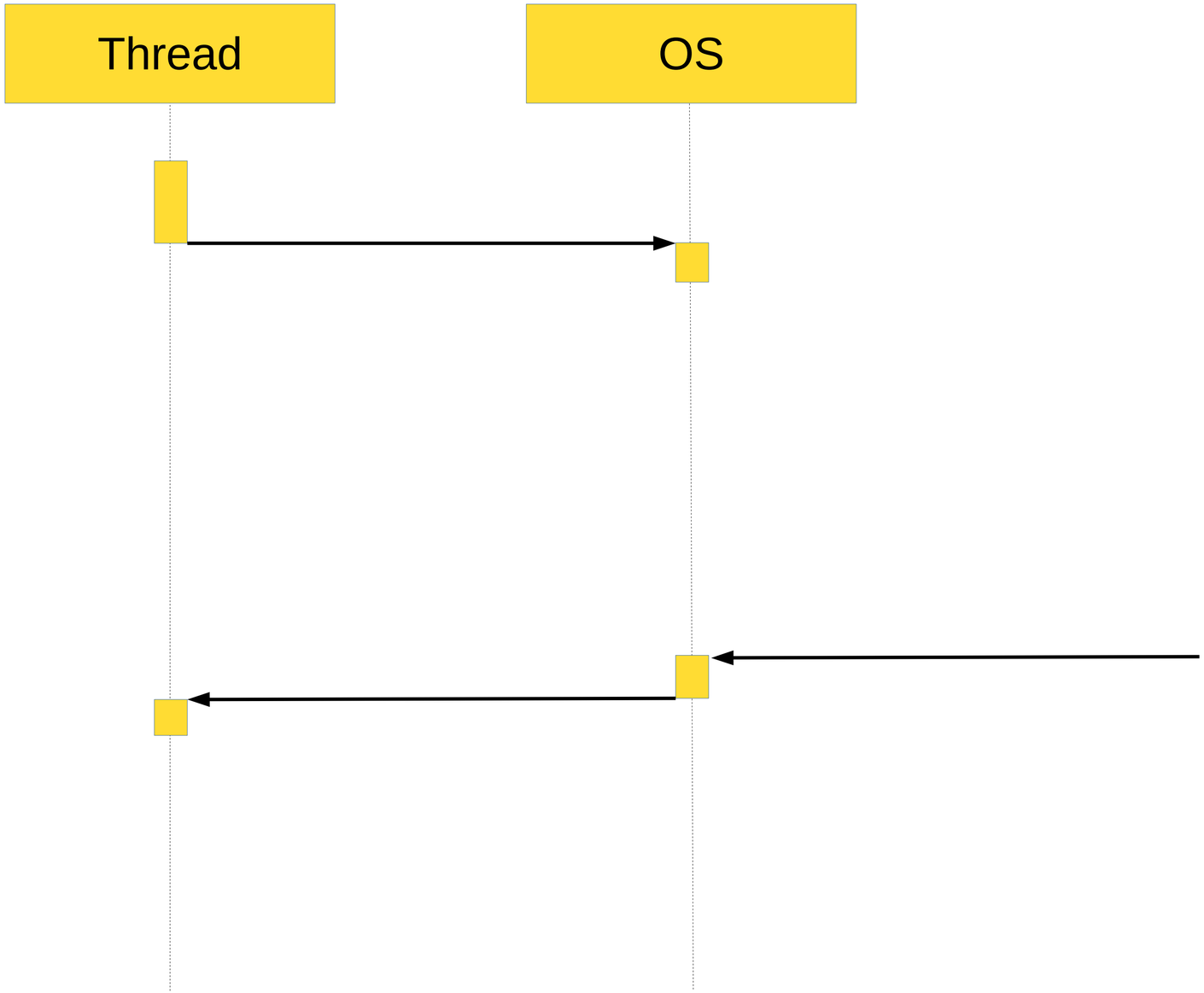
userver

```
// Асинхронное выполнение, std::thread переиспользуется
pg_cluster_->Execute(storages::postgres::ClusterHostType::kMaster,
                     "DELETE FROM key_value_table WHERE key=$1", key);
```

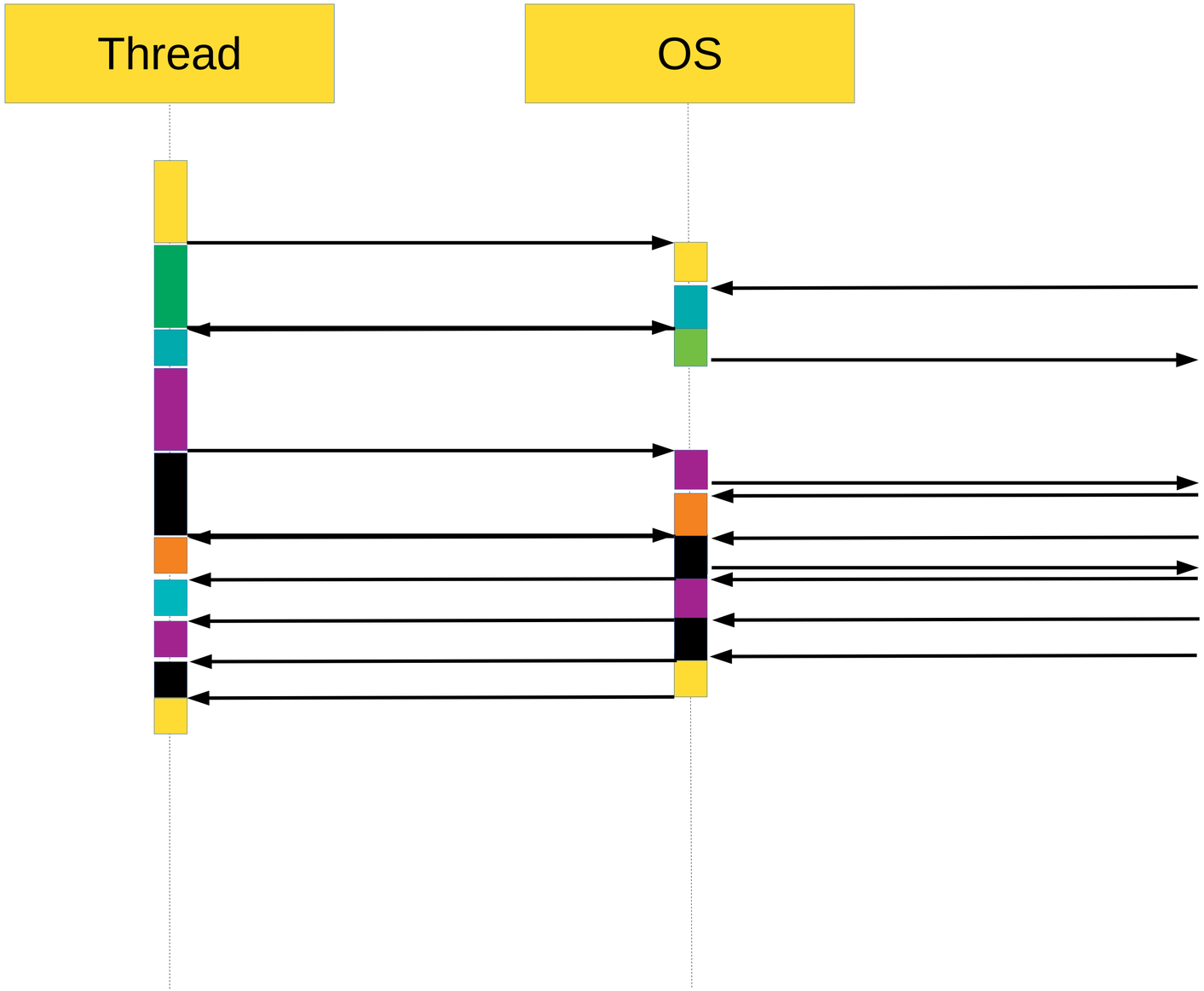
HE userver



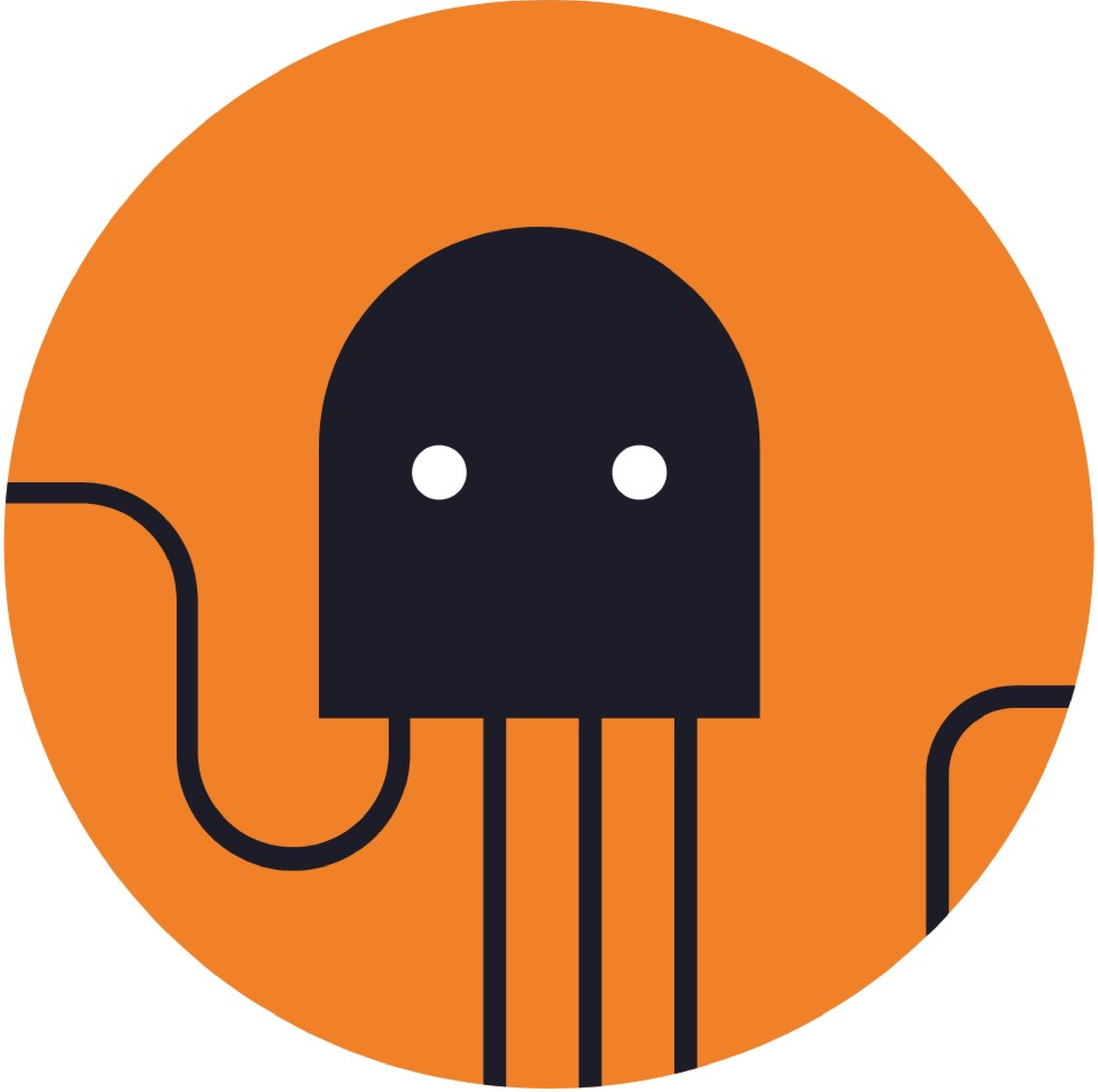
C++



C++



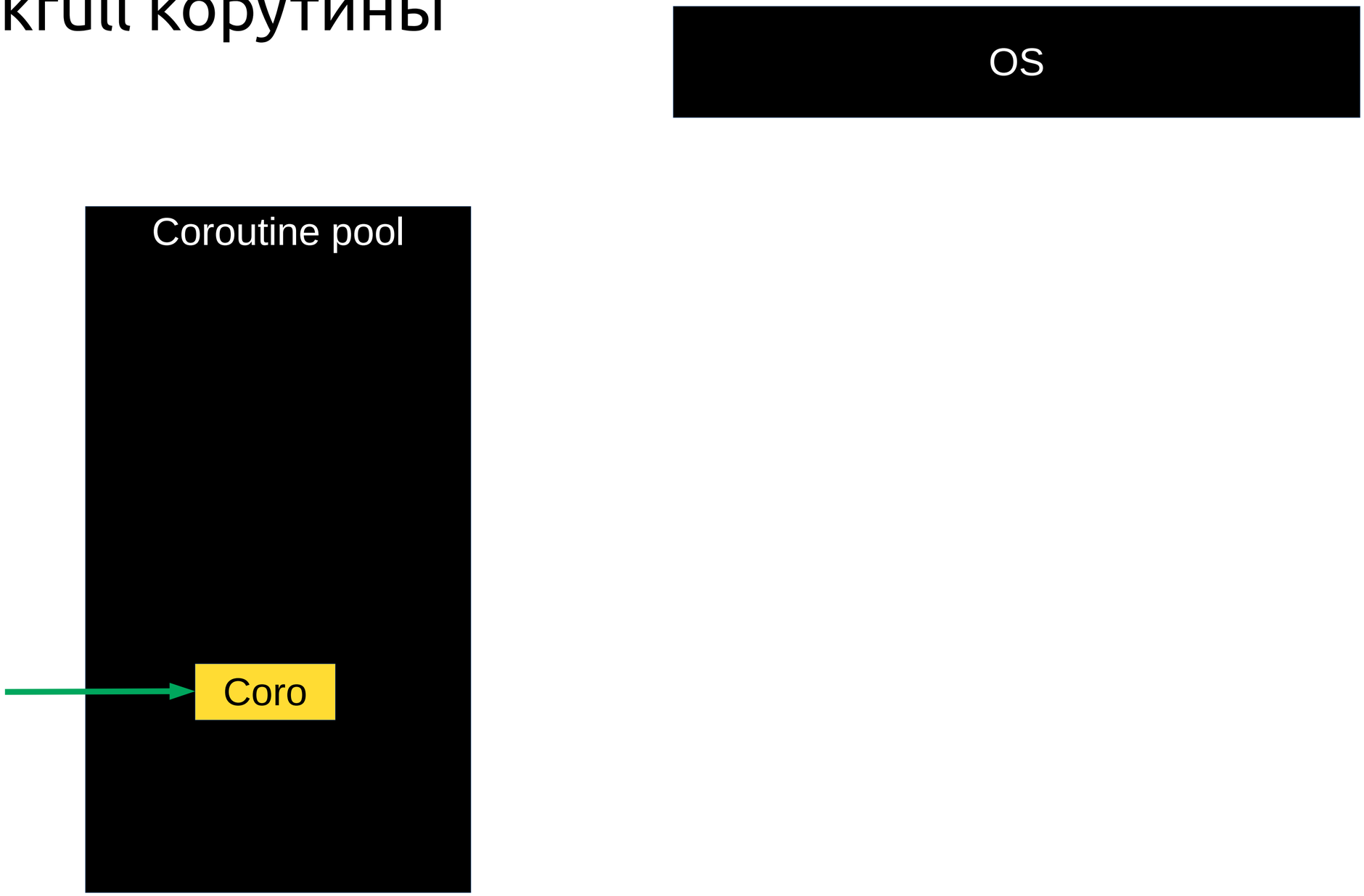
Как это сделать?



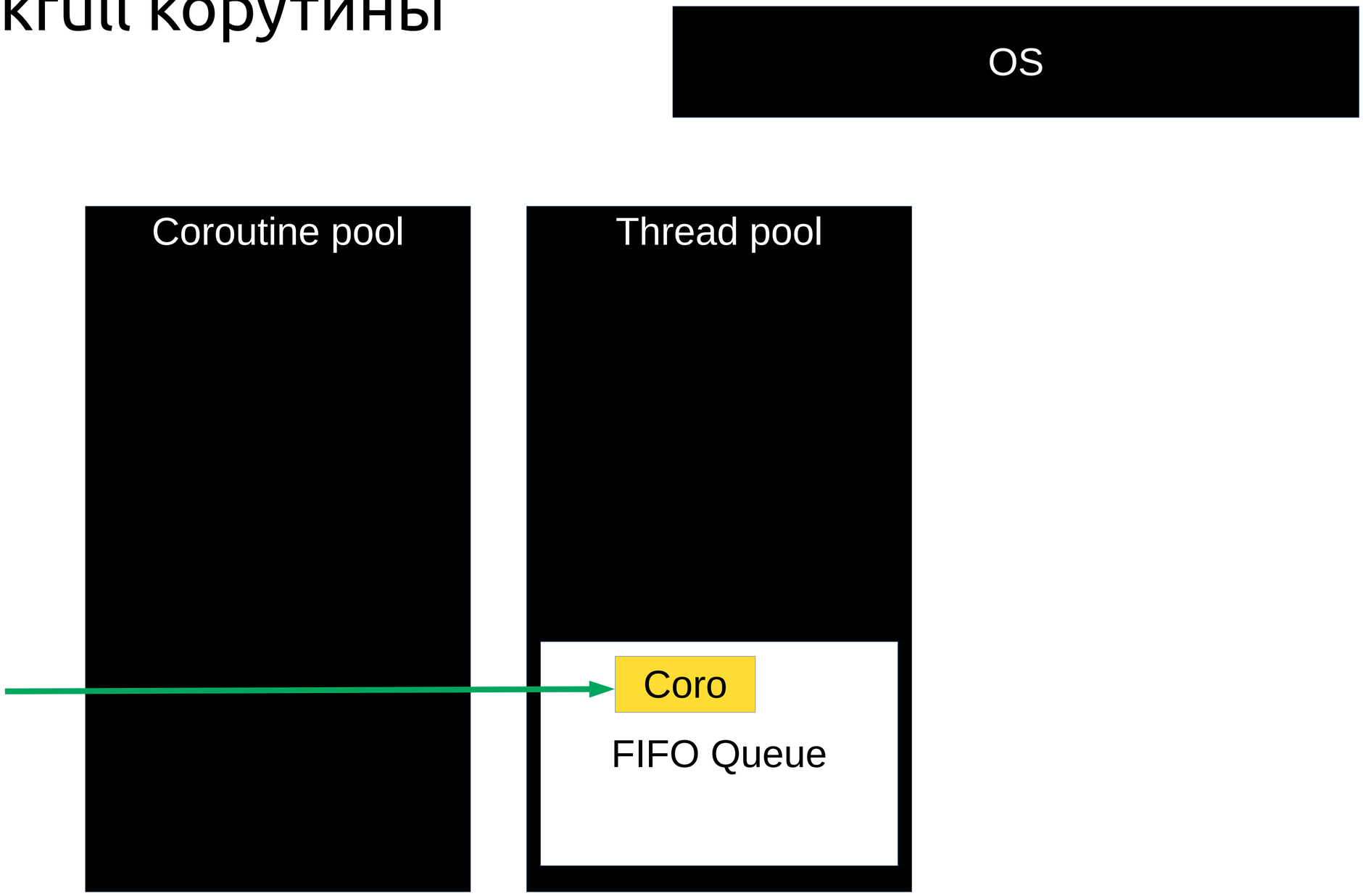
stackfull корутины

stackfull корутины

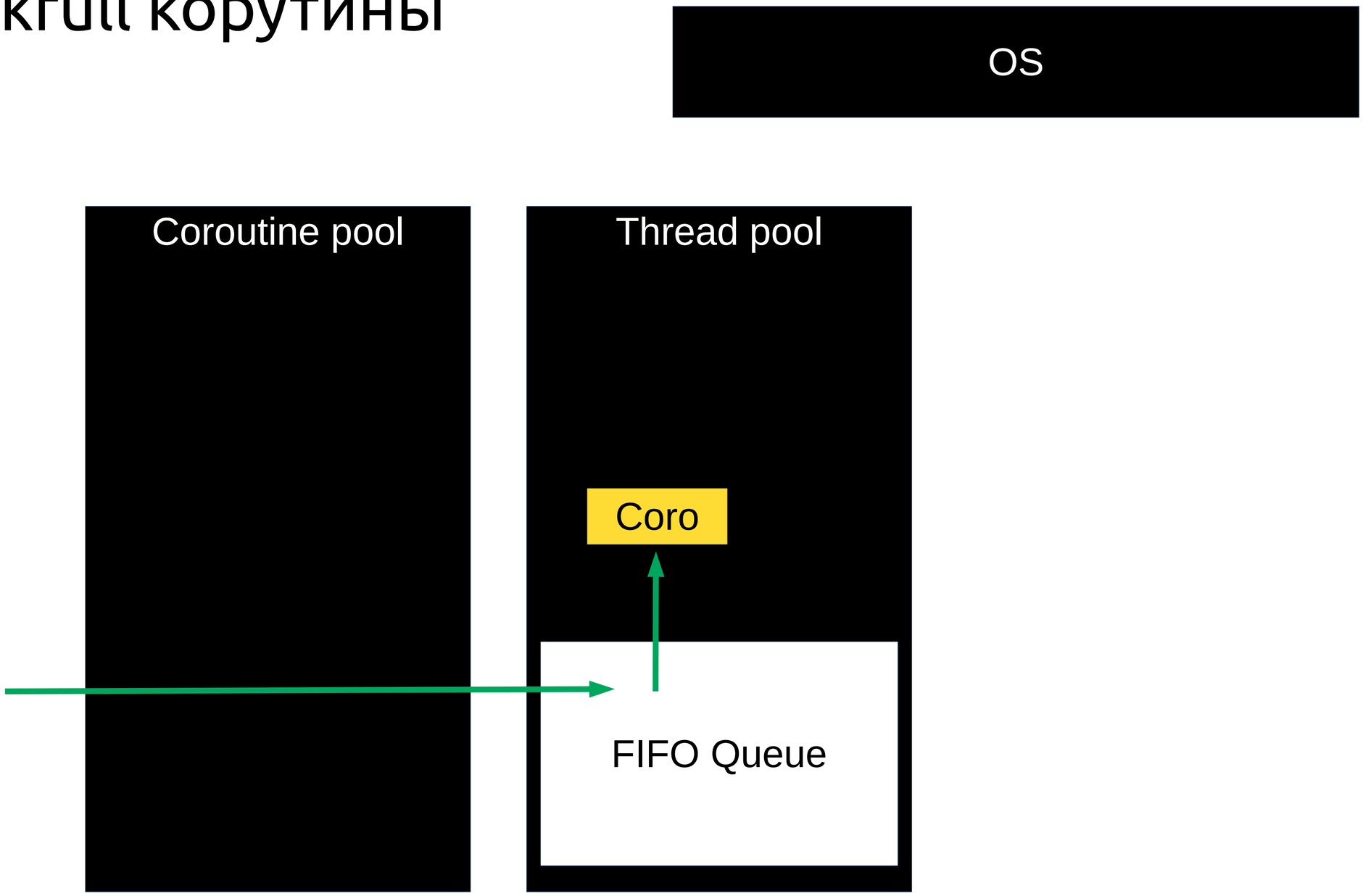
stackfull корутины



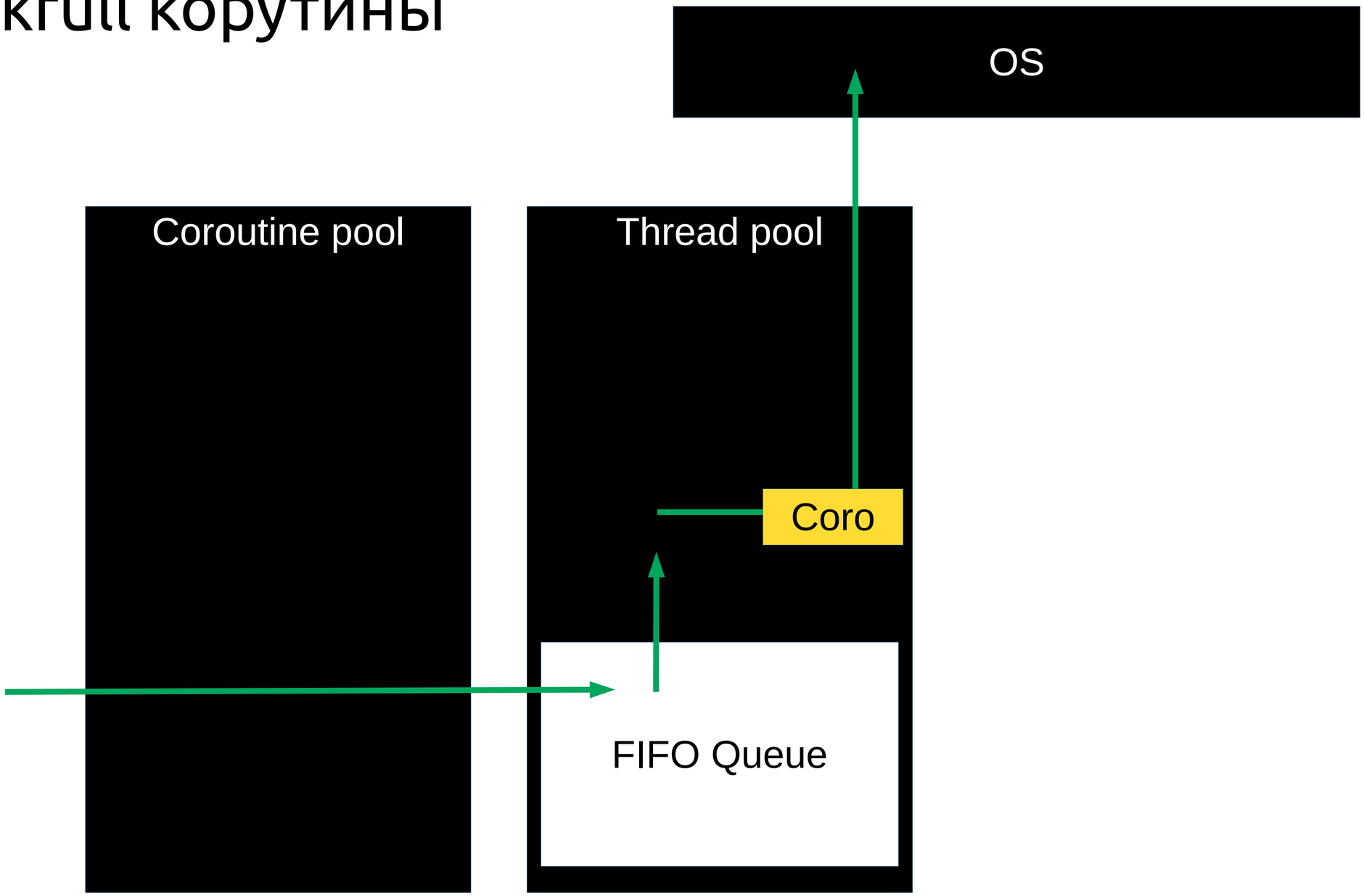
stackfull корутины



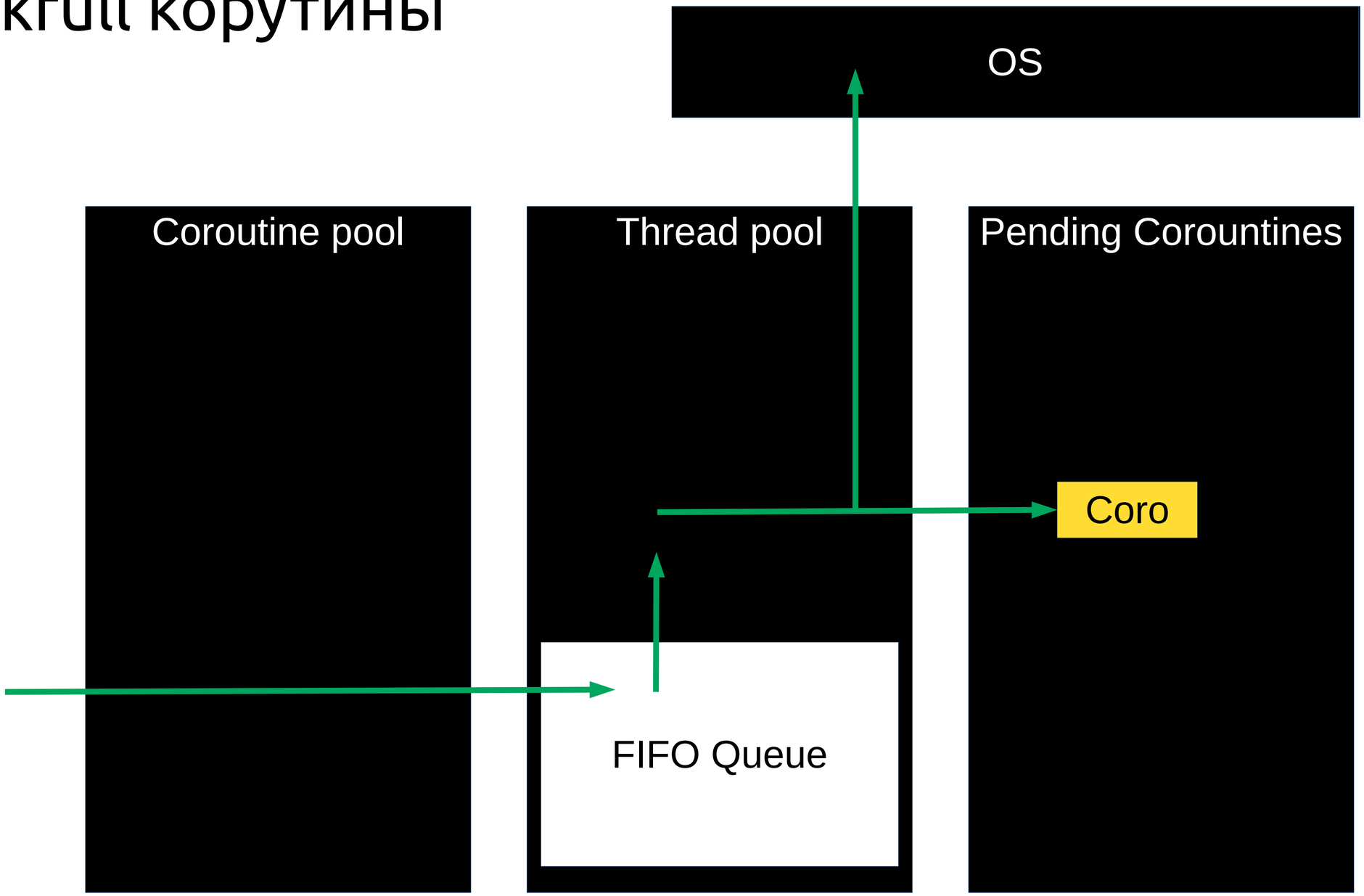
stackfull корутины



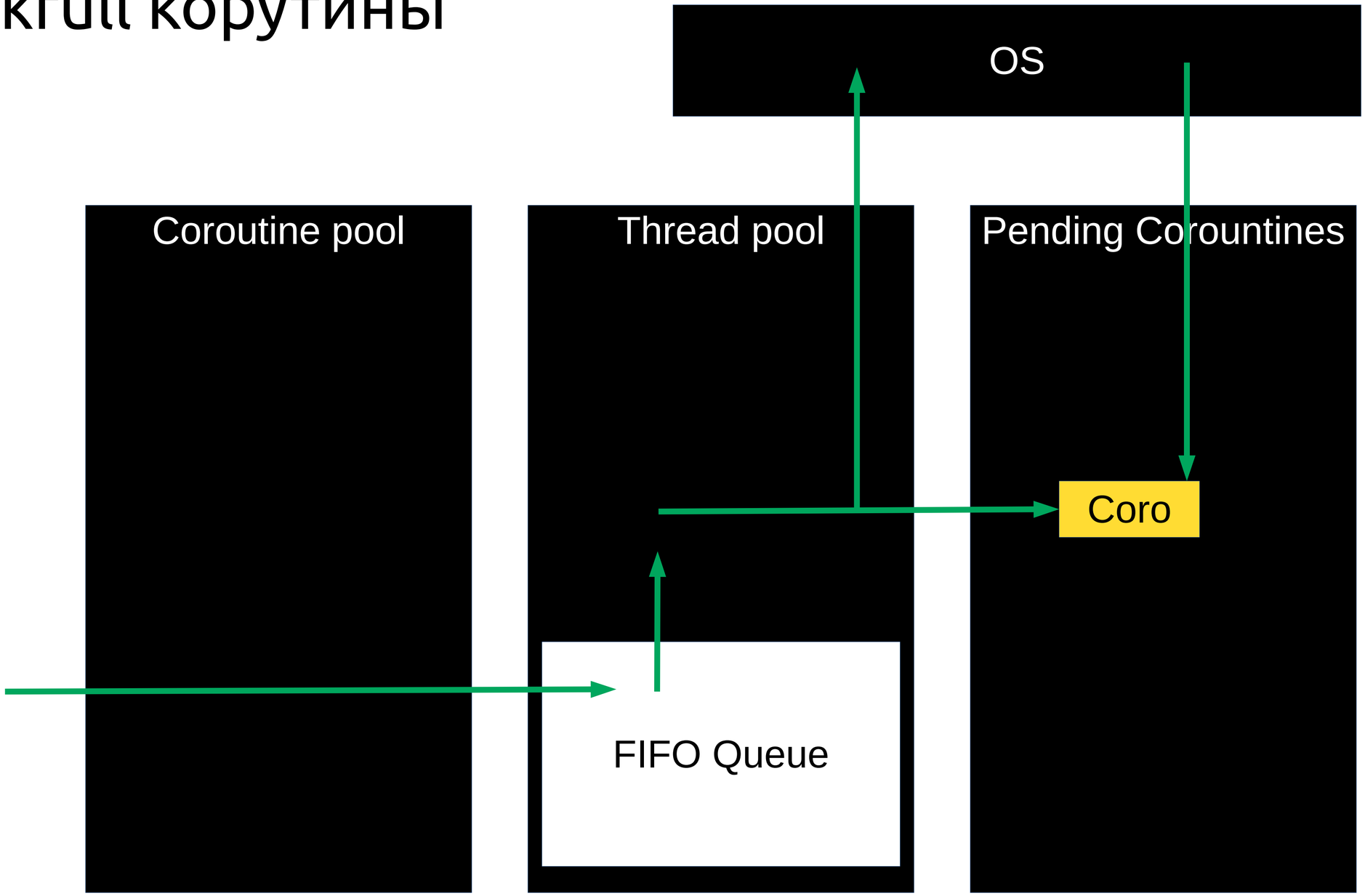
stackfull корутины



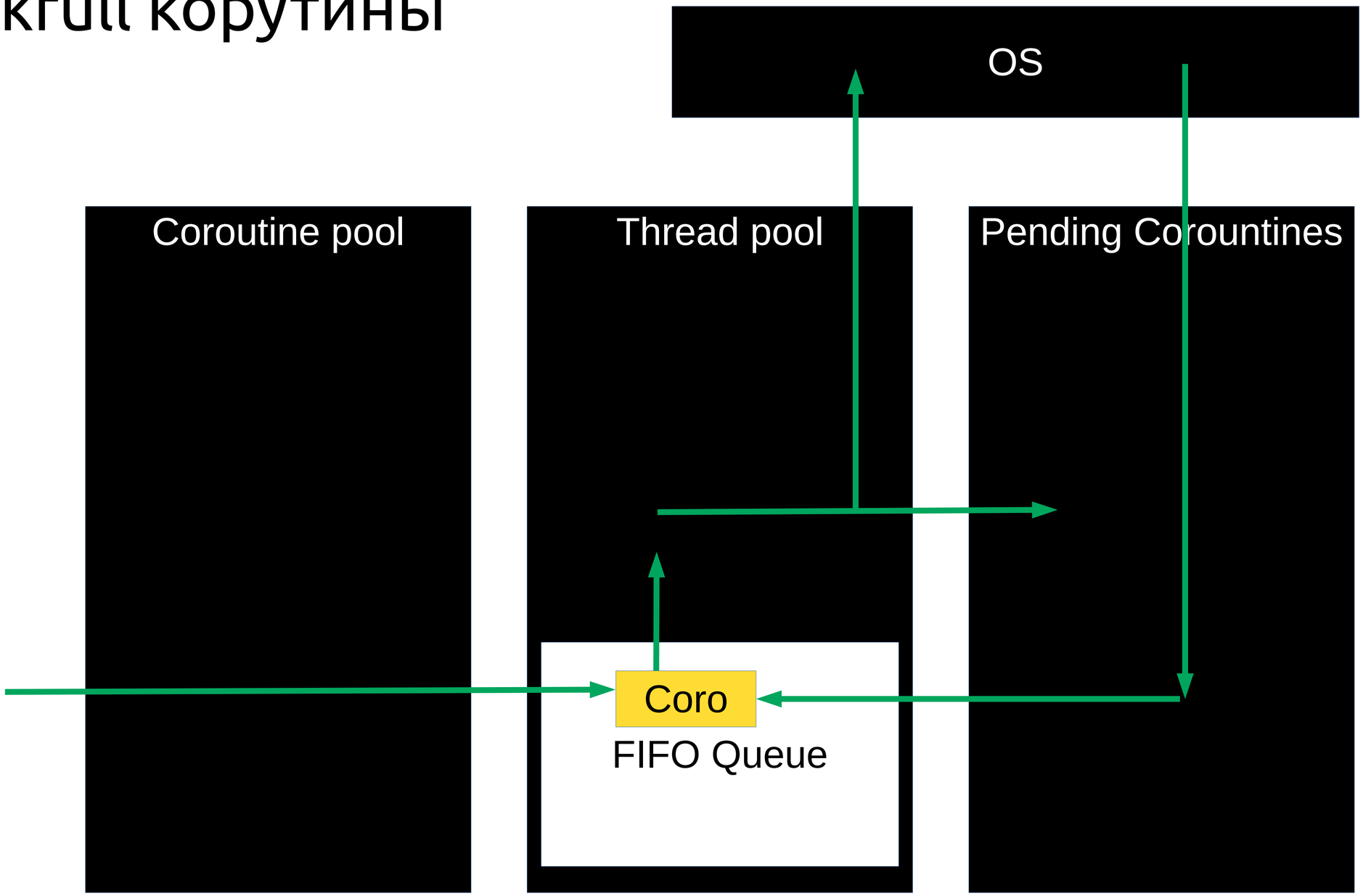
stackfull корутины



stackfull корутины



stackfull корутины



userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

userver

```
Response View::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>()};
}
```

HE userver

```
void View::Handle(Request&& request, const Dependencies& dependencies, Response
response) {
  dependencies.pg->GetCluster(
    [request = std::move(request), response](auto cluster)
  {
    cluster->Begin(storages::postgres::ClusterHostType::kMaster,
      [request = std::move(request), response](auto& trx)
    {
      const char* statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";
      psql::Execute(trx, statement, request.id,
        [request = std::move(request), response, trx = std::move(trx)](auto& res)
      {
        auto row = res[0];
        if (!row["ok"].As<bool>()) {
          if (LogDebug()) {
              GetSomeInfoFromDb([id = request.id](auto info) {
                  LOG_DEBUG() << id << " is not OK of " << info;
              });
```

HE userver

```
            }
            *response = Response400{};
          }
          psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar,
            [row = std::move(row), trx = std::move(trx), response]()
          {
            trx.Commit([row = std::move(row), response]() {
              *response = Response200{row["baz"].As<std::string>()};
            });
          });
        });
      });
    });
  }
```

Итог

Итого

Итого

- Все архитектуры хороши по своему

Итого

- Все архитектуры хороши по своему
  - Монолит весьма неплох

Итого

- Все архитектуры хороши по своему
  - Монолит весьма неплох
  - Микросервисы тоже норм

Итого

- Все архитектуры хороши по своему
  - Монолит весьма неплох
  - Микросервисы тоже норм
- Думайте над архитектурой балансеров

Итого

- Все архитектуры хороши по своему
  - Монолит весьма неплох
  - Микросервисы тоже норм
- Думайте над архитектурой балансеров
- Кеши + микросервисы = ❤

Итого

- Все архитектуры хороши по своему
  - Монолит весьма неплох
  - Микросервисы тоже норм
- Думайте над архитектурой балансеров
- Кеши + микросервисы = ❤
- Нужен толковый фреймворк для IO-bound приложений...

Итого

- Все архитектуры хороши по своему
  - Монолит весьма неплох
  - Микросервисы тоже норм
- Думайте над архитектурой балансеров
- Кеши + микросервисы = ❤
- Нужен толковый фреймворк для IO-bound приложений...

  ... и он скоро появится в открытом доступе

Спасибо

Полухин Антон

Эксперт-разработчик C++

antoshkka@gmail.com

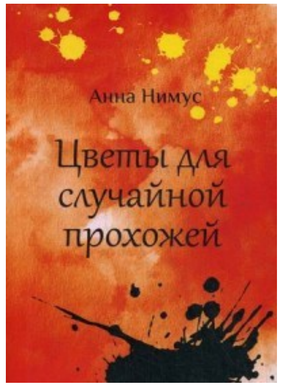
antoshkka@yandex-team.ru

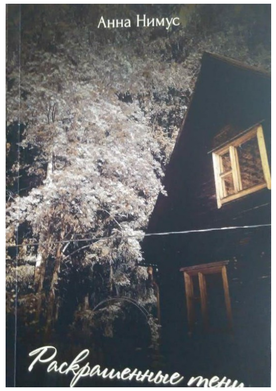
https://github.com/apolukhin

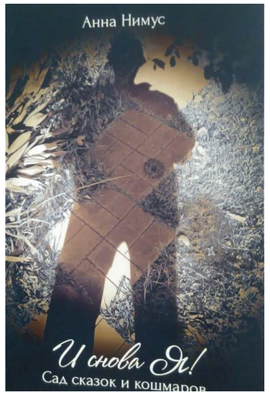
https://stdcpp.ru/

РГ21 C++ РОССИЯ

Спасибо


Анна Нимус
Цветы для случайной прохожей


Анна Нимус


Анна Нимус
И снова Я!
Сад сказок и кошмаров