Модульный BFF

МАРГАРИТА КОЗЫРЕВА

· СТАРШИЙ РАЗРАБОТЧИК · ЛАБОРАТОРИЯ КАСПЕРСКОГО

kaspersky

Маргарита Козырева

> Senior Frontend в Kaspersky

Маргарита Козырева

- Senior Frontend в Kaspersky
- 7+ лет во фронте

Маргарита Козырева

- Senior Frontend в Kaspersky
- 7+ лет во фронте
- Строила дизайн-системы с нуля

Маргарита Козырева

- Senior Frontend в Kaspersky
- 7+ лет во фронте
- Строила дизайн-системы с нуля
- Работала в VK, Positive Tech, X5, Raif

Маргарита Козырева

- Senior Frontend в Kaspersky
- 7+ лет во фронте
- Строила дизайн-системы с нуля
- Работала в VK, Positive Tech, X5, Raif
- Ушла из тимлидов в сеньоры

МАРШРУТ ДОКЛАДА

Что будет в докладе

1 — Модульный frontend

МАРШРУТ ДОКЛАДА

Что будет в докладе

1. Модульный frontend
2. Проблема общего BFF

kaspersky

МАРШРУТ ДОКЛАДА

Что будет в докладе

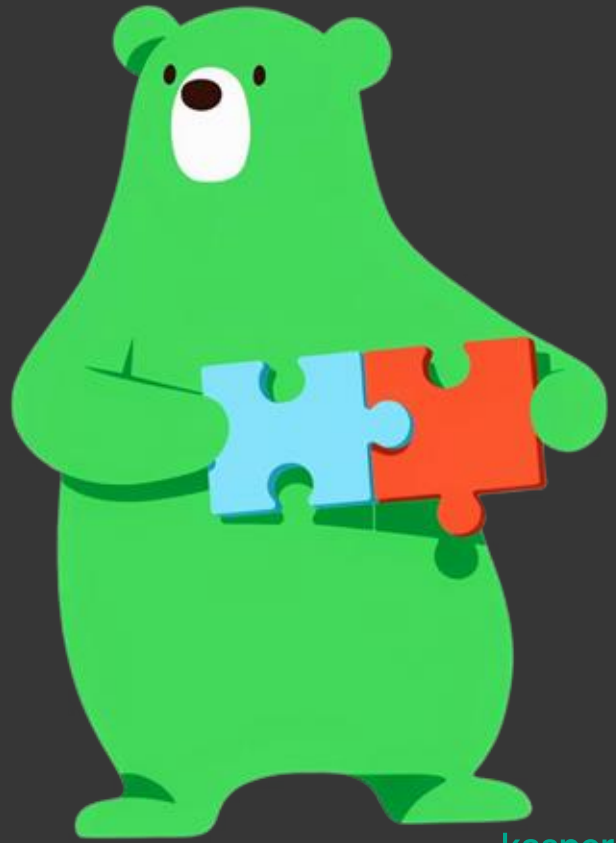
1. Модульный frontend
2. Проблема общего BFF
3. Модульный BFF

Это экспертный доклад

kaspersky

MICROFRONTEND

Microfrontend архитектура



kaspersky

Изолируем предметные области

MICROFRONTEND

- независимый деплой
- меньше связанности
- контракты между командами

У каждого модуля свои зависимости

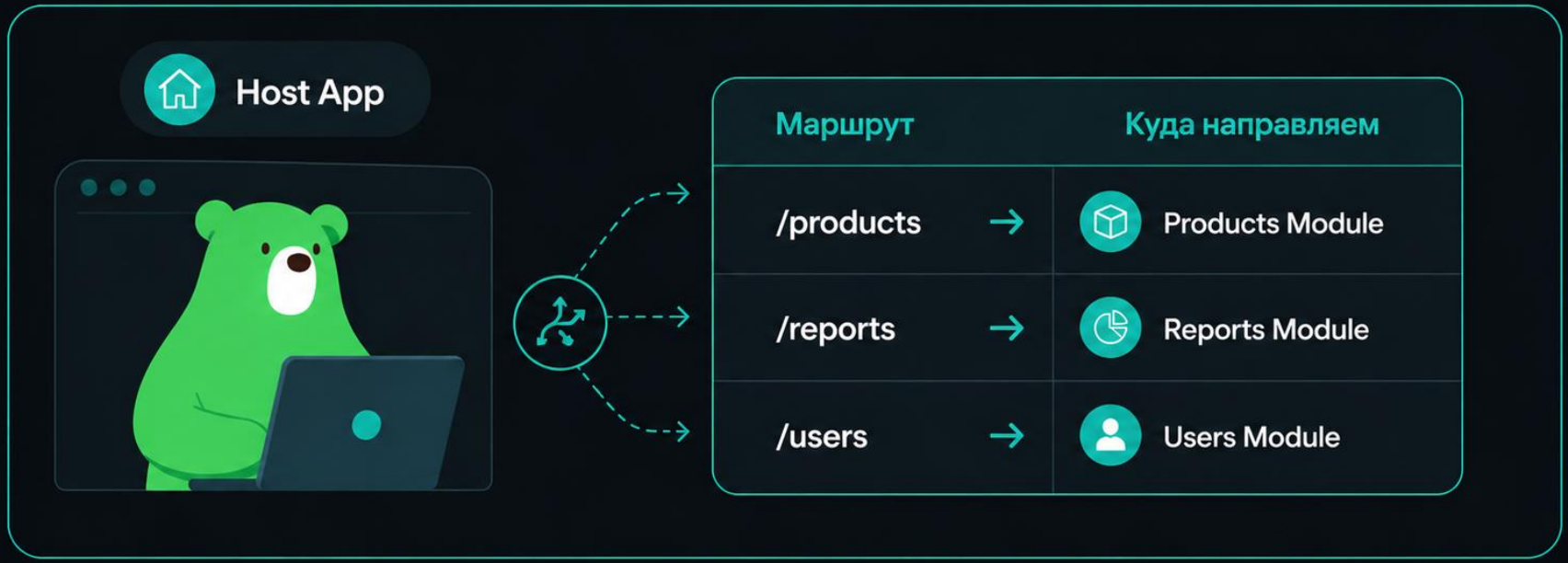
MICROFRONTEND

Внутри — свобода, снаружи — общий контракт

kaspersky

Маршрутизация живёт в Host

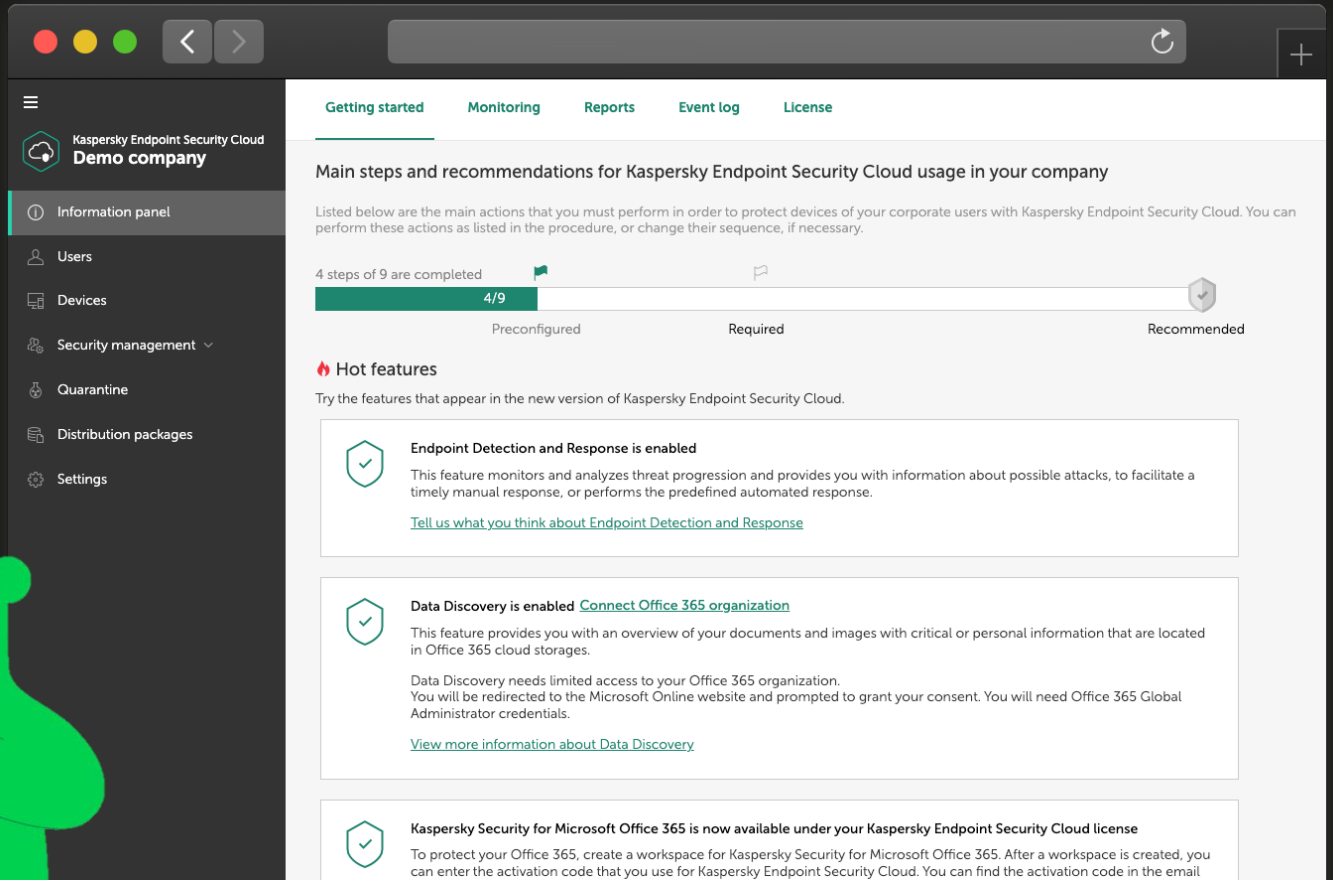
MICROFRONTEND



Пользователь переходит по маршрутам,
а Host решает, какой модуль показать

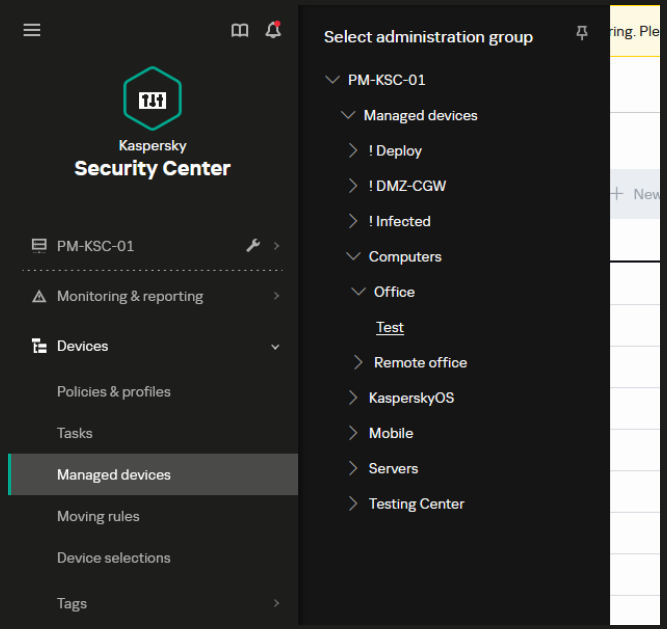
система для управления корпоративной защитой

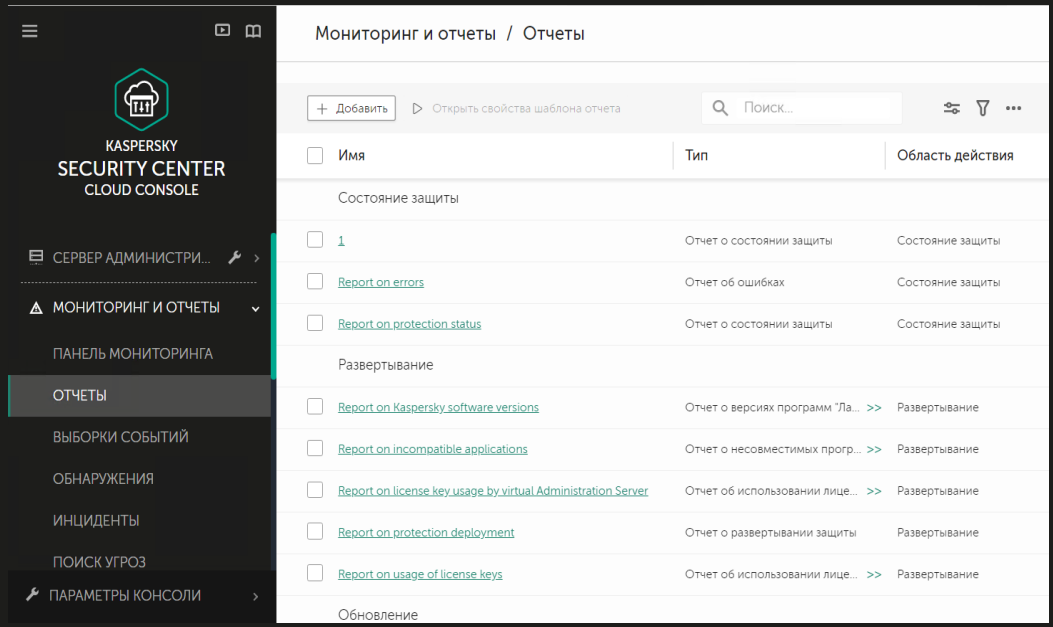
20+ микрофронтендов, каждый с отдельной зоной ответственности



kaspersky

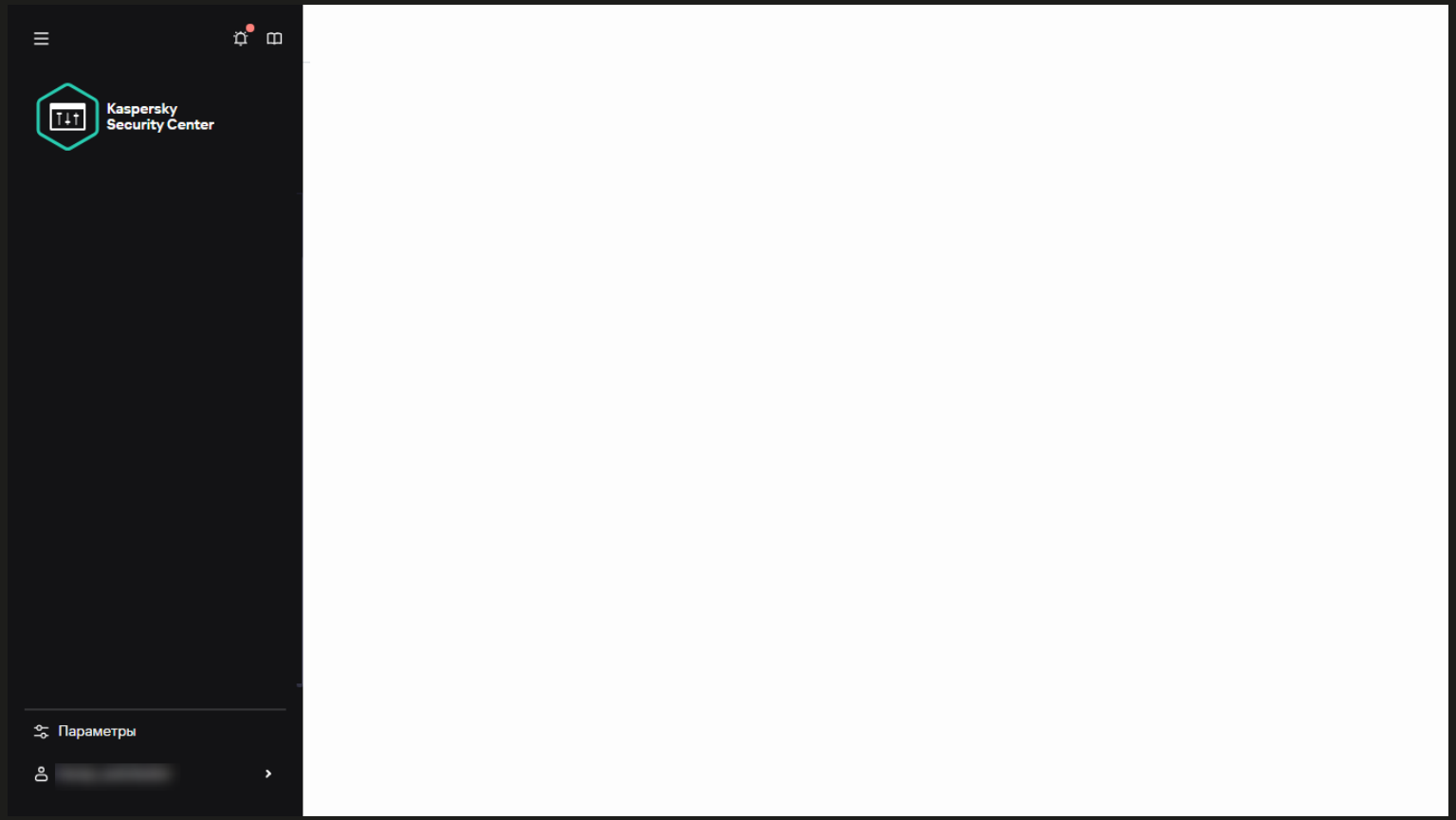
единый хост, связывающий микрофронтенды (плагины)

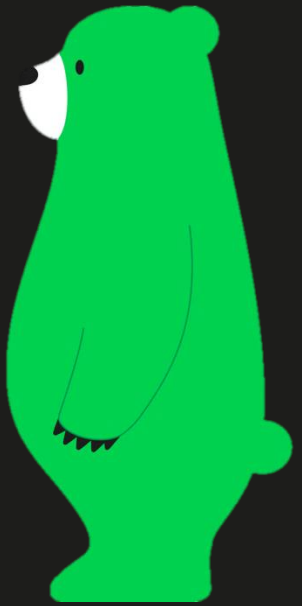




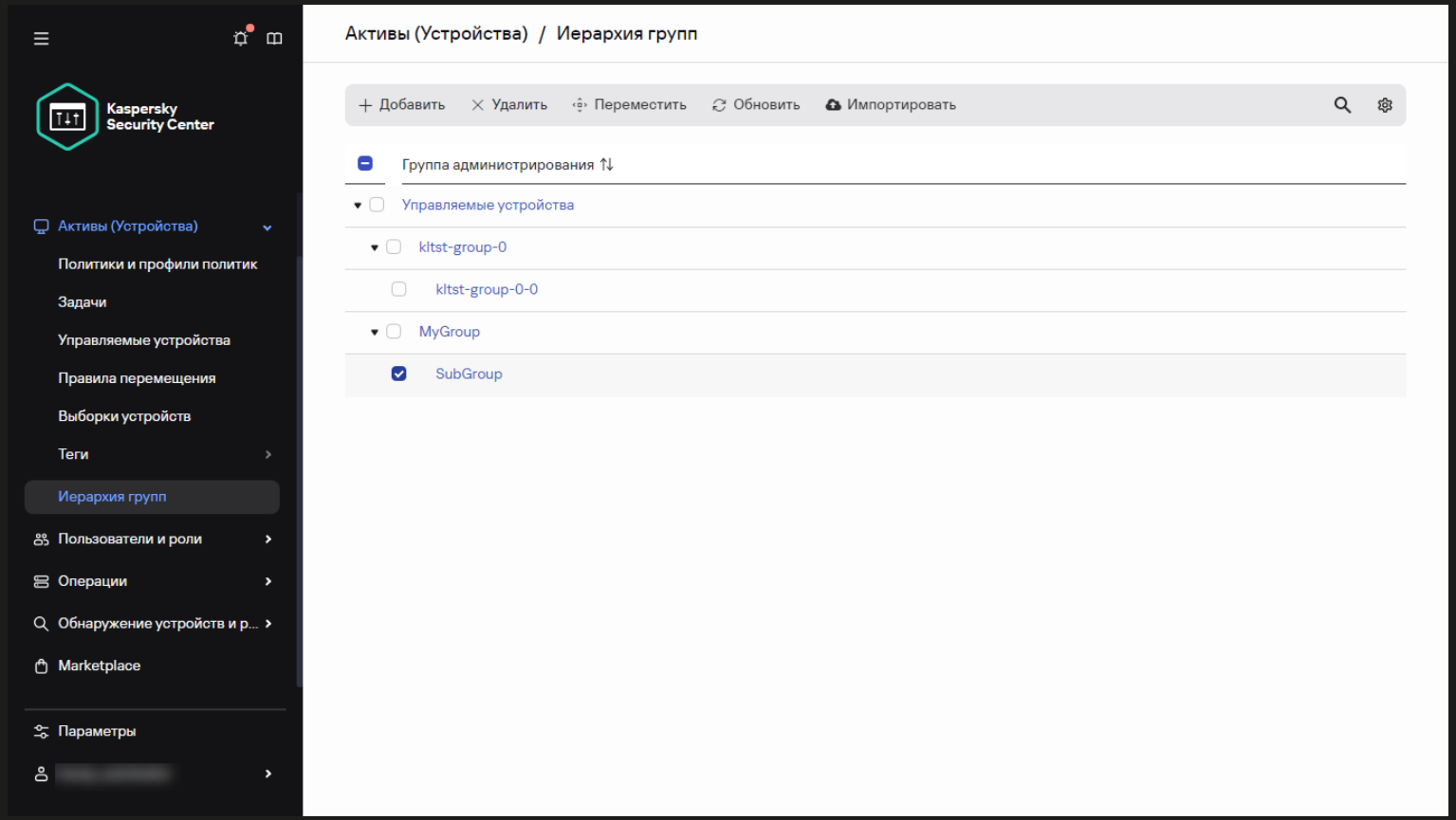
kaspersky

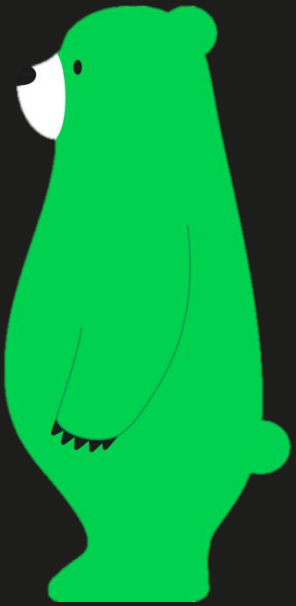
Компоненты платформы





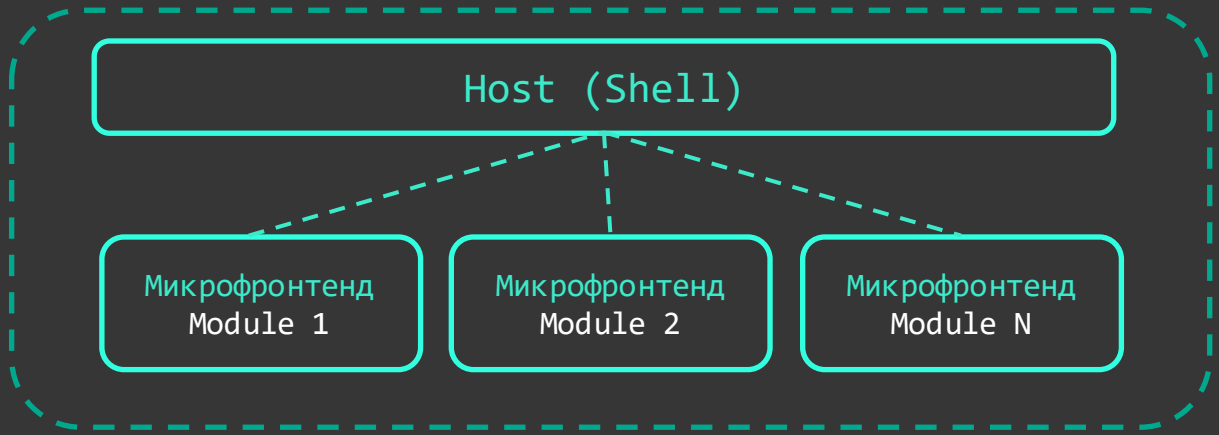
Компоненты платформы





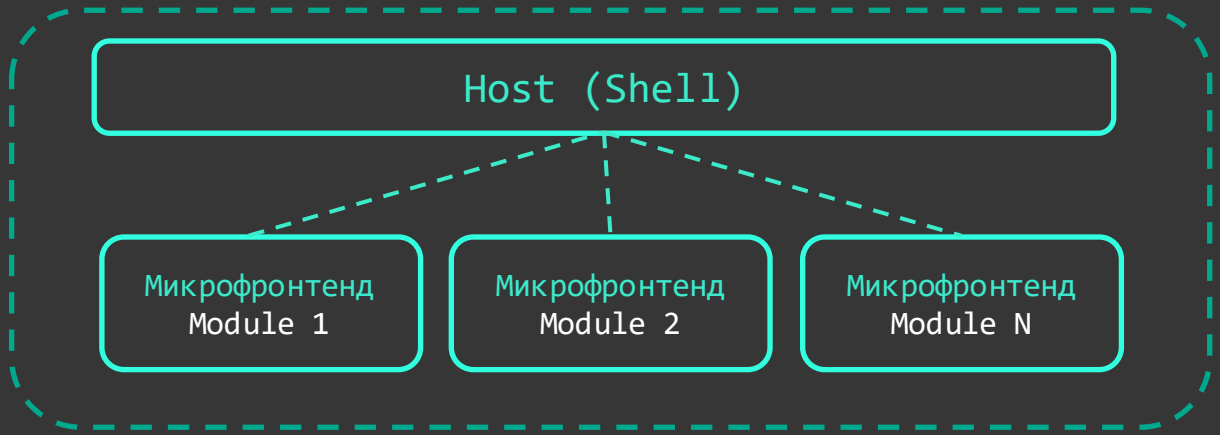
О ЧЁМ ЭТОТ ДОКЛАД

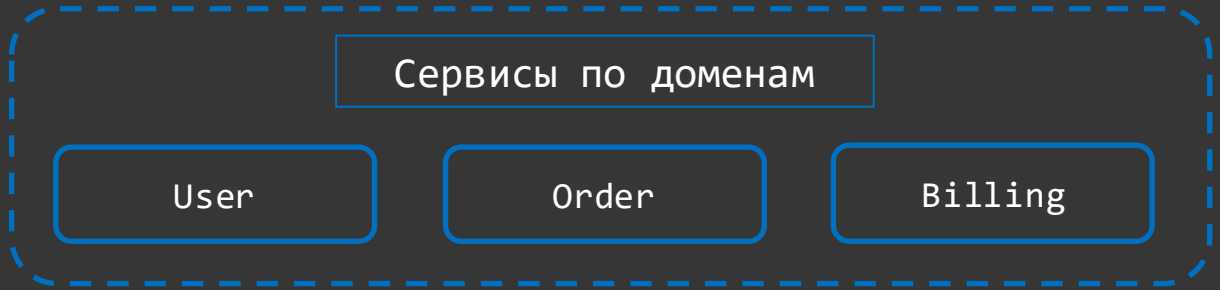
Модульность уже есть на клиенте



О ЧЁМ ЭТОТ ДОКЛАД

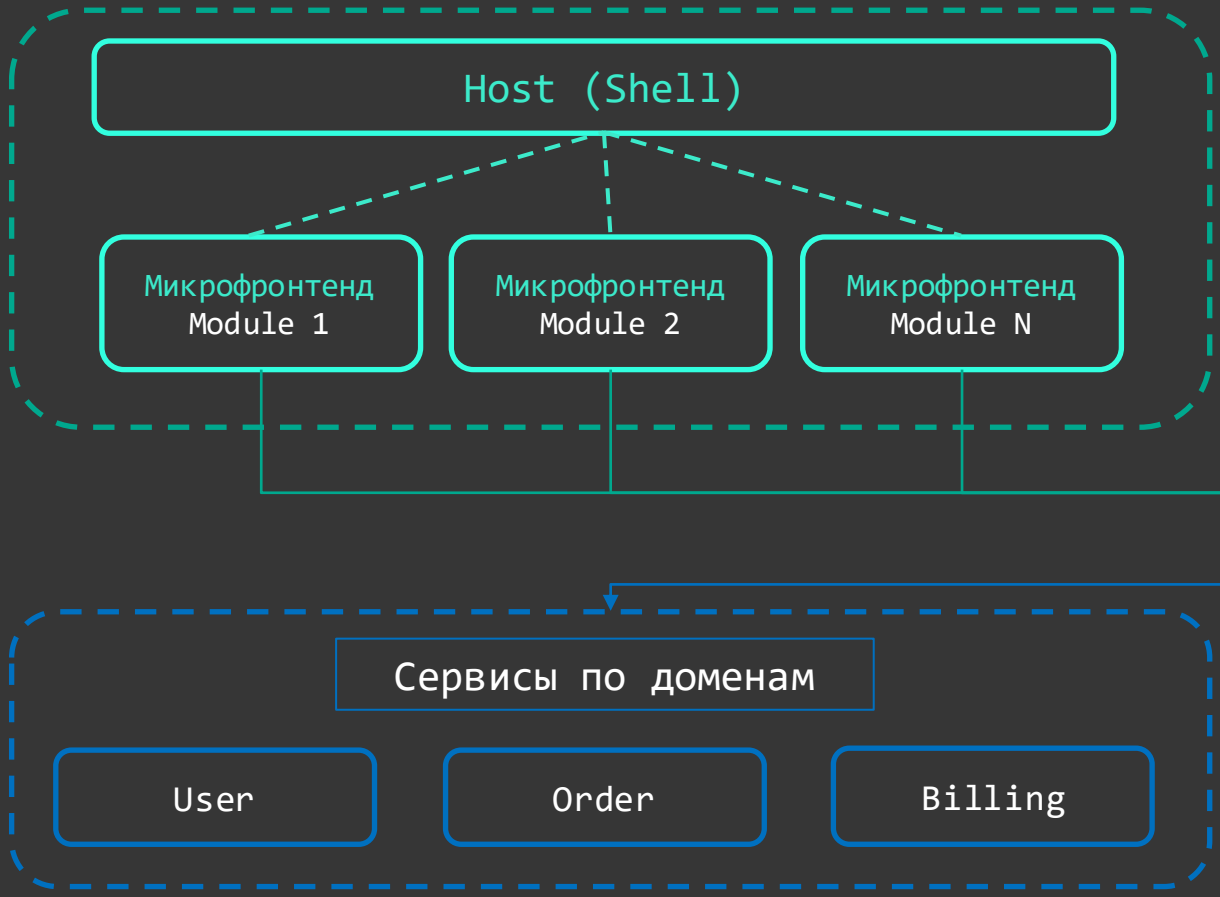
Модульность уже есть на клиенте





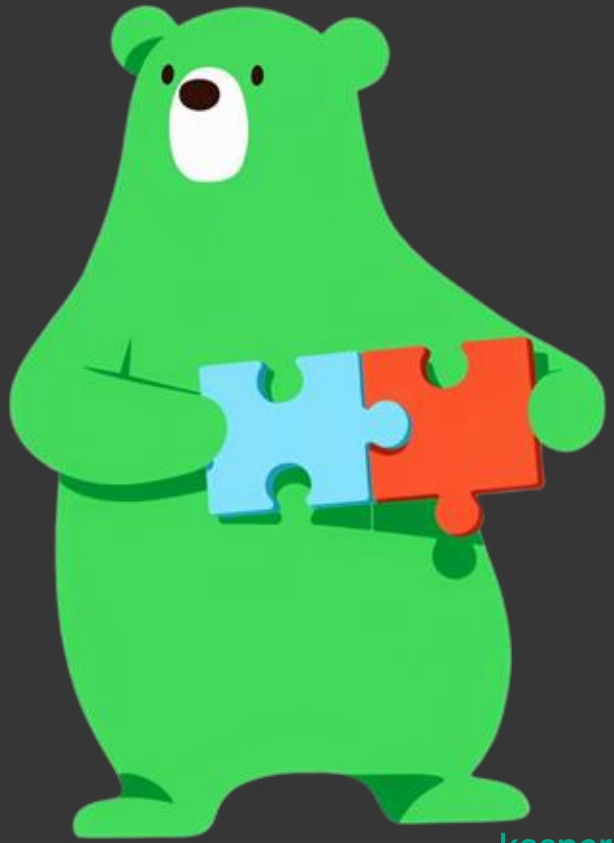
О ЧЁМ ЭТОТ ДОКЛАД

Модульность уже есть на клиенте



BFF

А зачем нам BFF?



ПОЧЕМУ ПОЯВЛЯЕТСЯ BFF?


```
async function loadDashboard(userId: string) {
  const user = await api.users.getUser(userId)
  const orders = await api.orders.getByUser(userId)
  const billing = await api.billing.getBalance(userId)
  return {
    title: user.name,
    ordersCount: orders.total,
    balance: billing.amount
  }
}
```

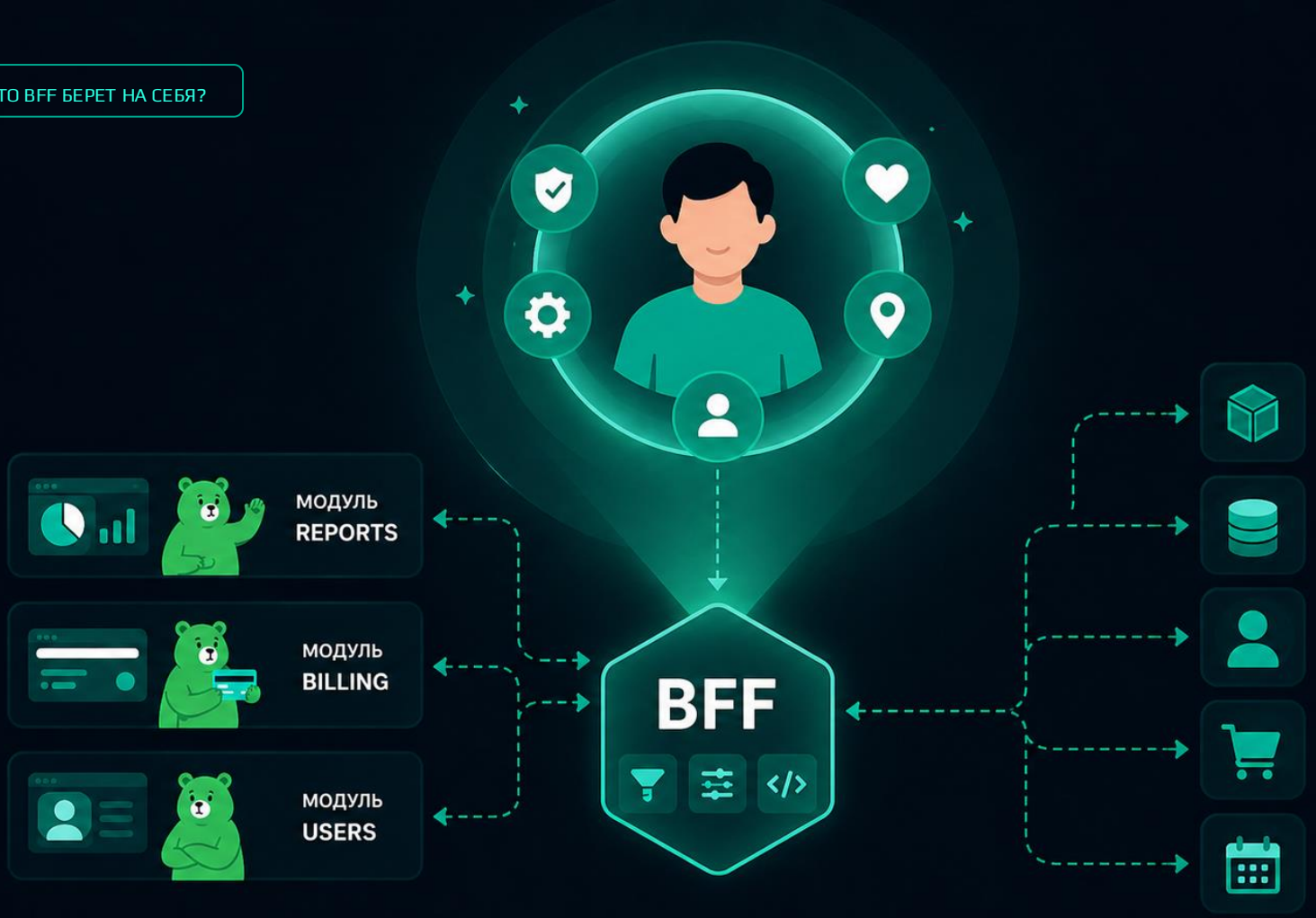
kaspersky

ПОЧЕМУ ПОЯВЛЯЕТСЯ BFF?


```
async function getDashboardView(userId: string) {  // Плохой вариант: доменный сервис знает про UI-экран
  const user = await users.getById(userId)
  const orders = await orders.getByUser(userId)
  return {
    title: user.name,
    subtitle: `${orders.total} active orders`,
    webActions: buildWebActions(user),
    mobileActions: buildMobileActions(user)
  }
}
```

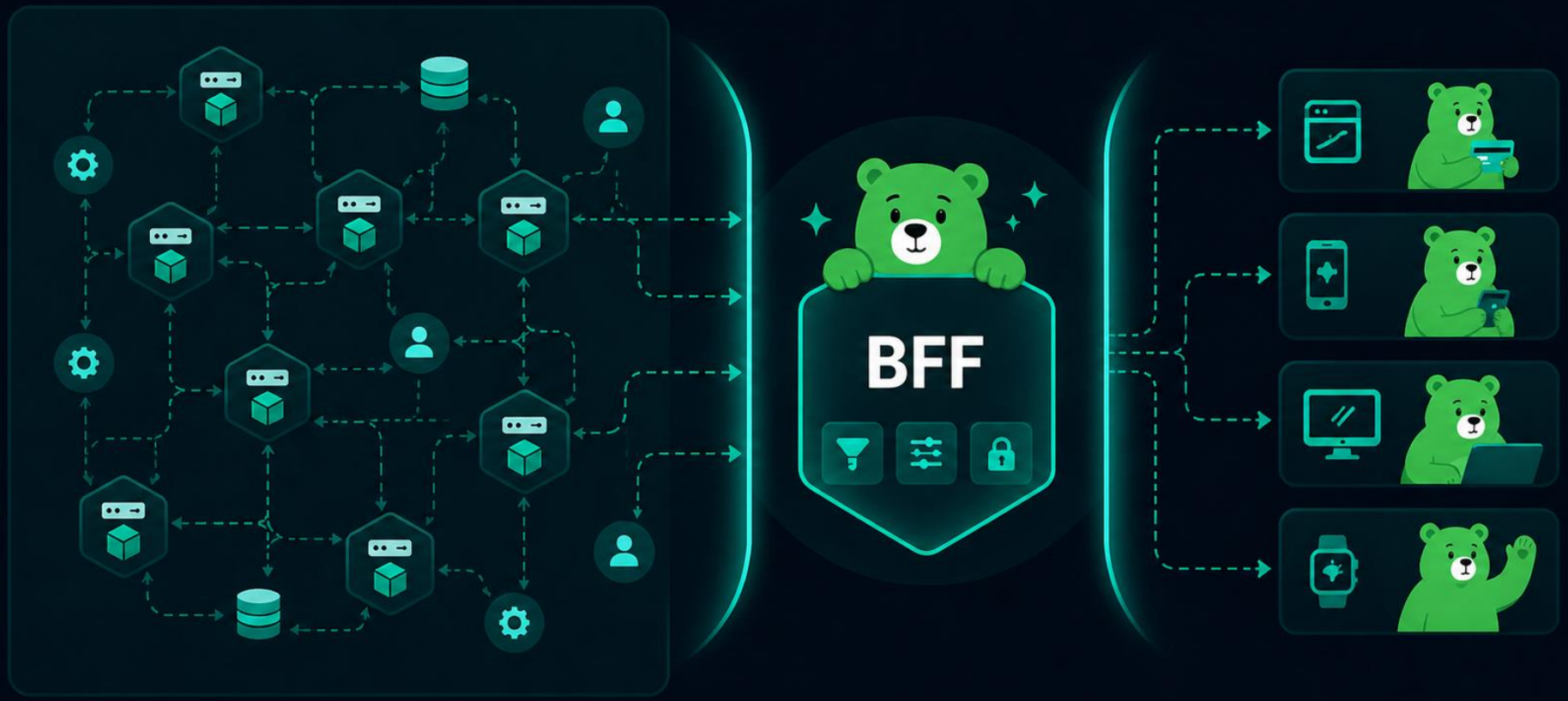
kaspersky

ЧТО BFF БЕРЕТ НА СЕБЯ?



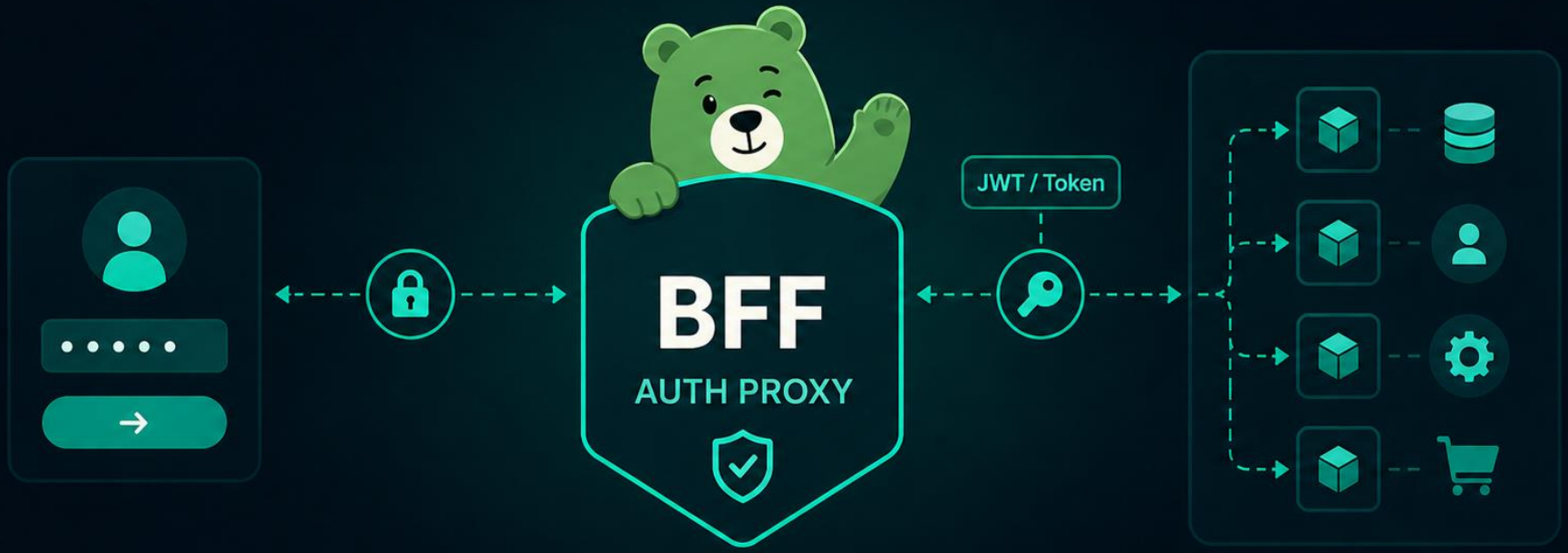
kaspersky

ЧТО BFF БЕРЕТ НА СЕБЯ?



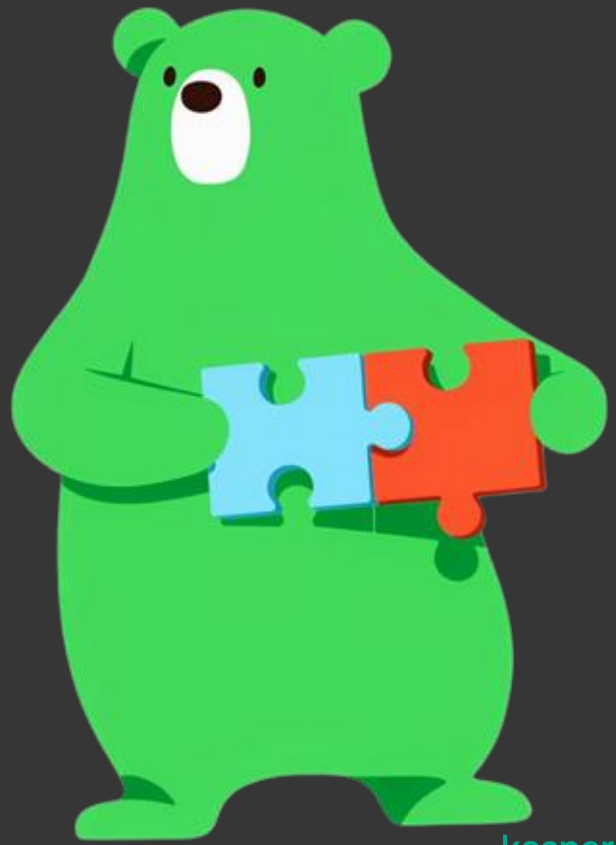
kaspersky

ЧТО BFF БЕРЕТ НА СЕБЯ?



BFF

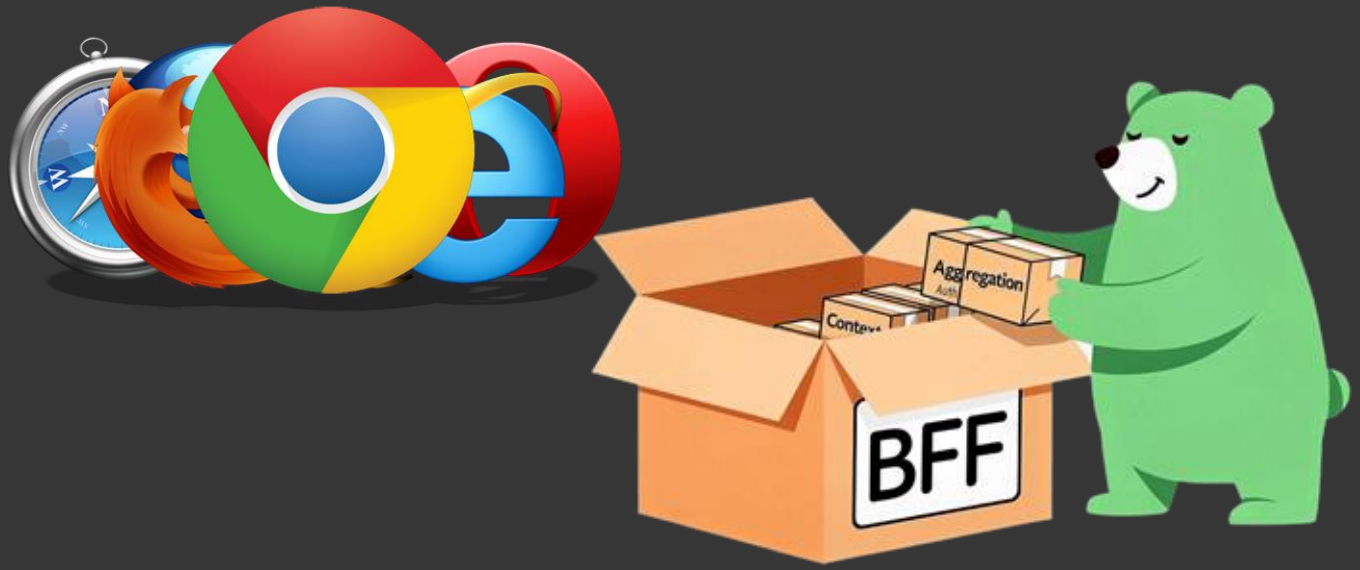
Не весь трафик идёт через BFF



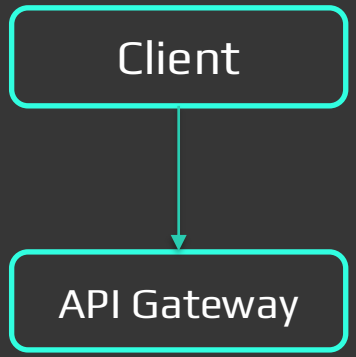
kaspersky

BFF

Client

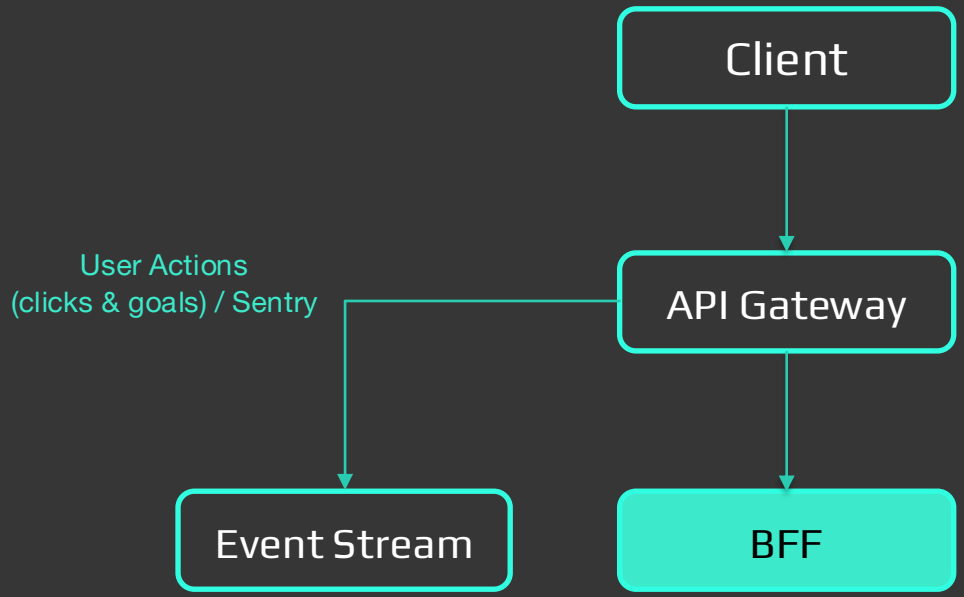


BFF



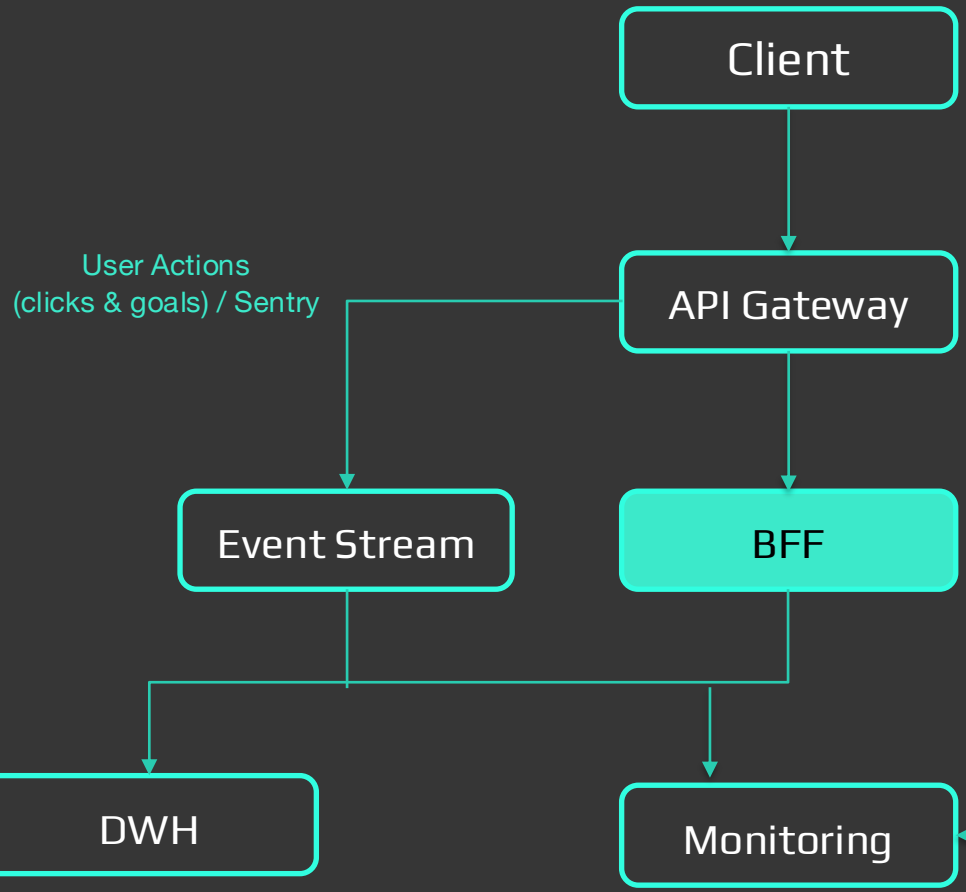
kaspersky

BFF


Client
User Actions
(clicks & goals) / Sentry
API Gateway
Event Stream
BFF

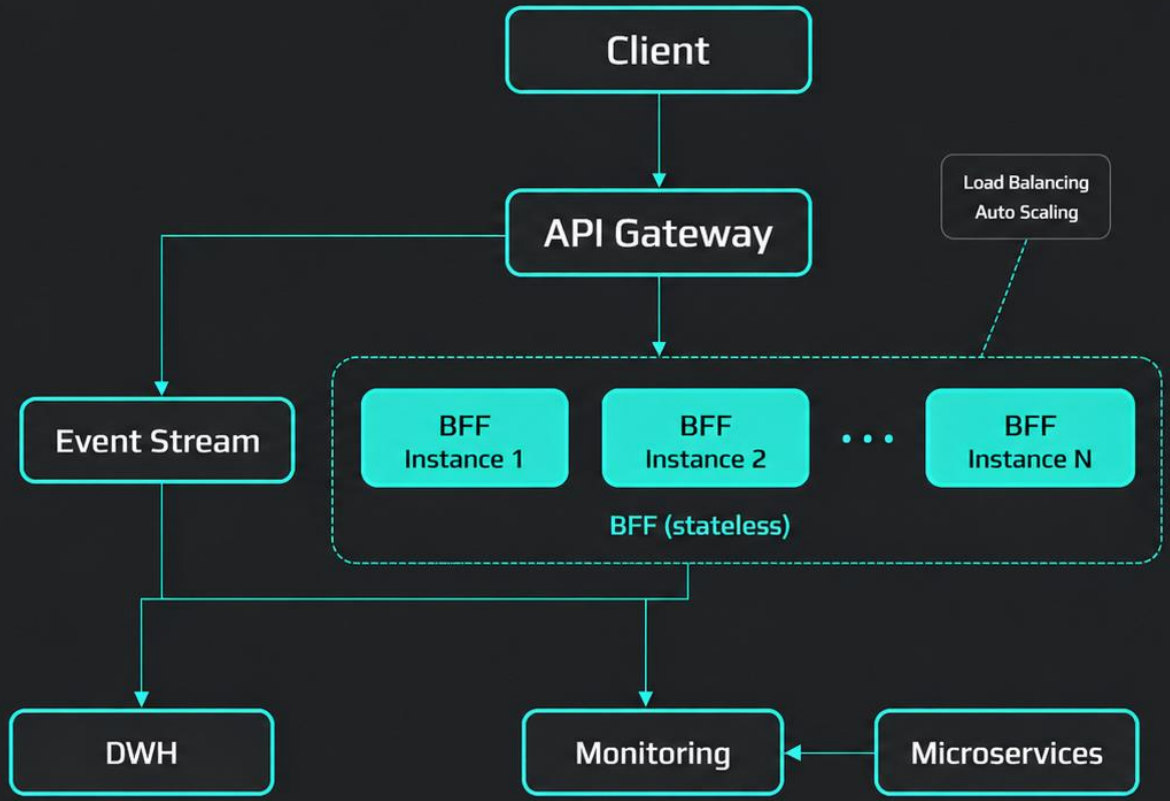

Aggregation
BFF

BFF



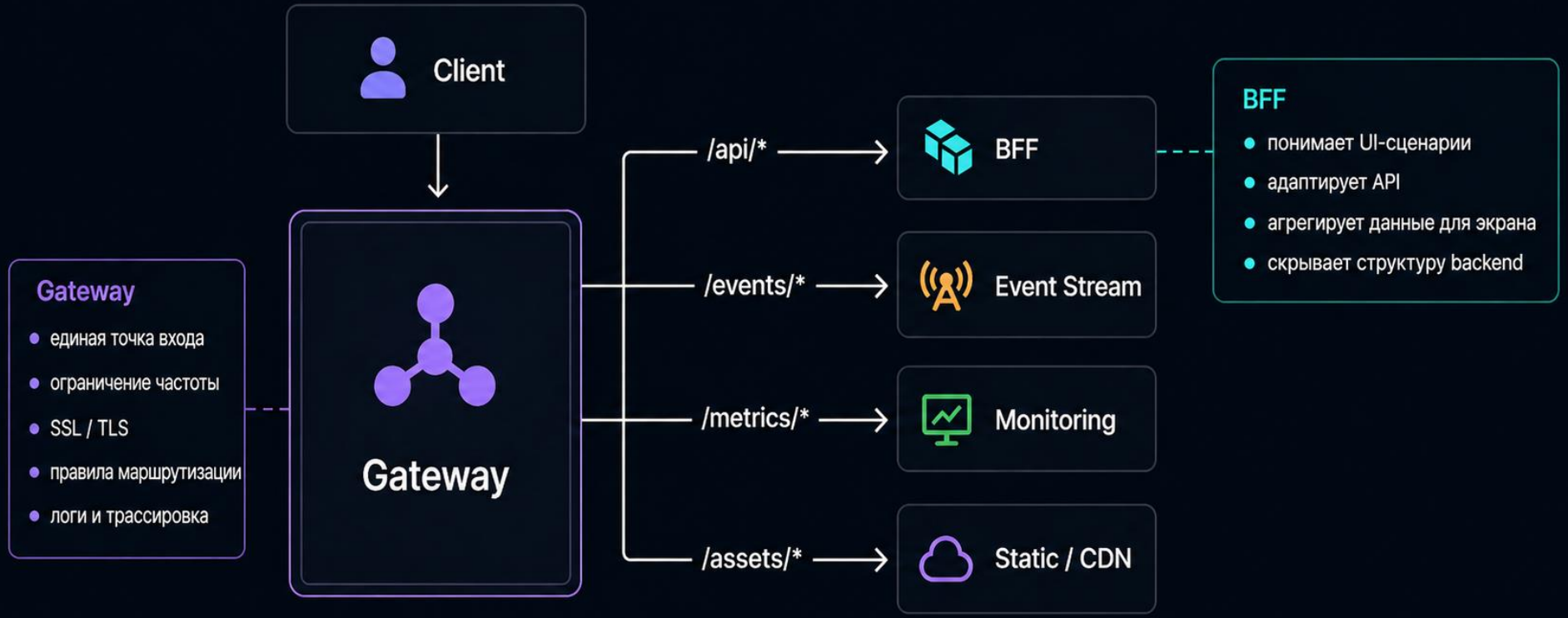
kaspersky

BFF МАСШТАБИРУЕТСЯ

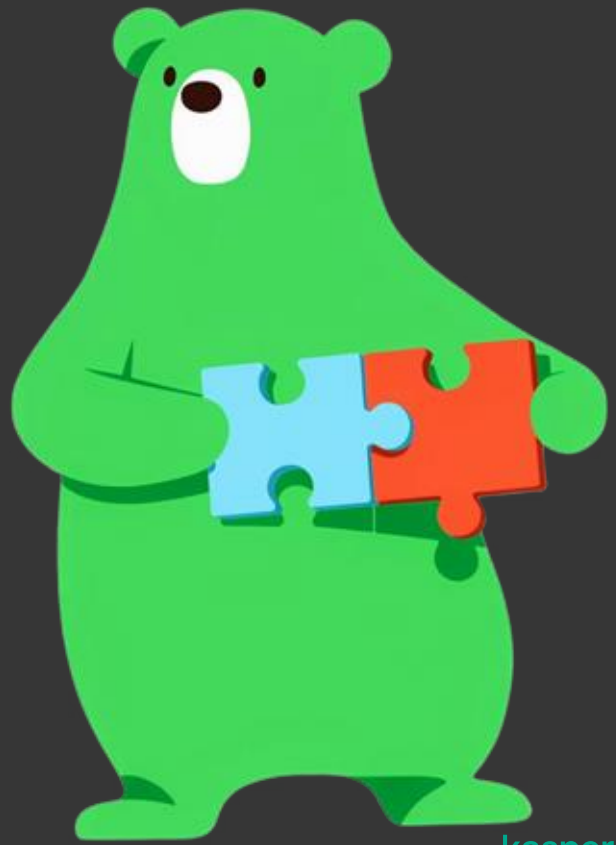


BFF

API Gateway?

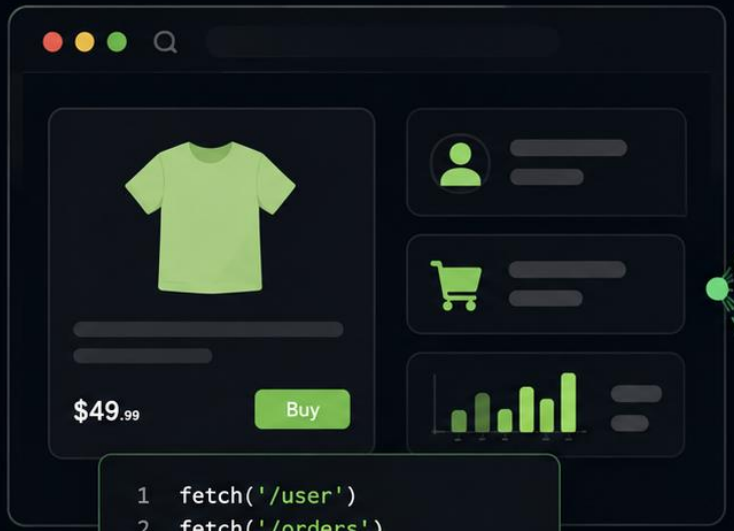


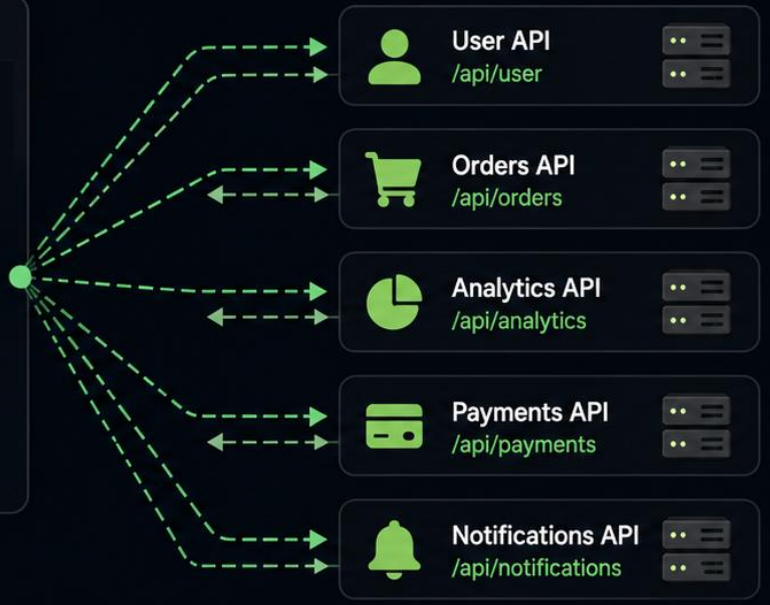
BIF 1.0 vs BIF 2.0



BIF VS BFF

BIF 1.0






```
1  fetch('/user')
2  fetch('/orders')
3  mergeData(...)
4  adaptResponse(...)
5  renderUI(...)
```

Frontend собирает сценарий сам

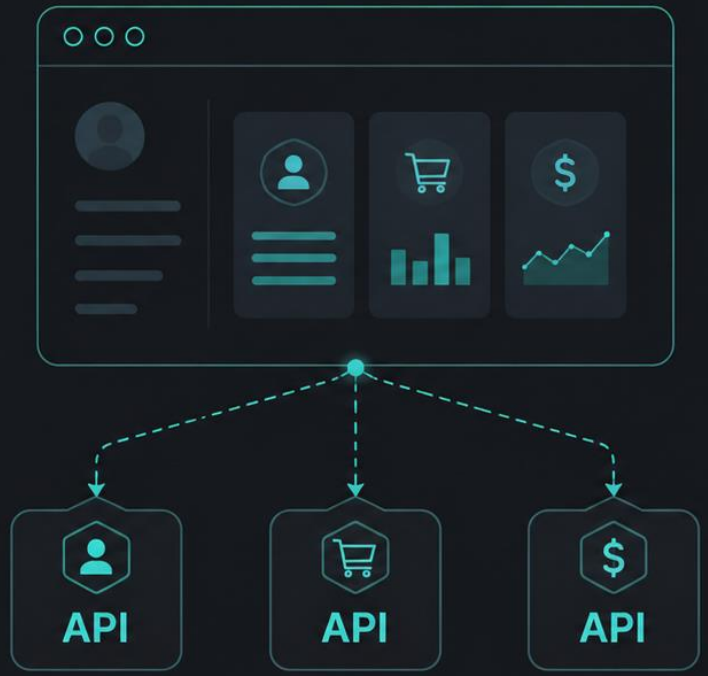
- Агрегация данных
- Адаптация контрактов
- Тесная связь с бэком
- Хаос API

kaspersky

BIF VS BFF

BIF 1.0

```
async function loadDashboard(userId: string) {
  const [user, orders, billing] = await Promise.all([
    api.get(`/users/${userId}`),
    api.get(`/orders?userId=${userId}`),
    api.get(`/billing/${userId}`)
  ])
  return {
    name: user.name,
    ordersCount: orders.total,
    balance: billing.amount
  }
}
```

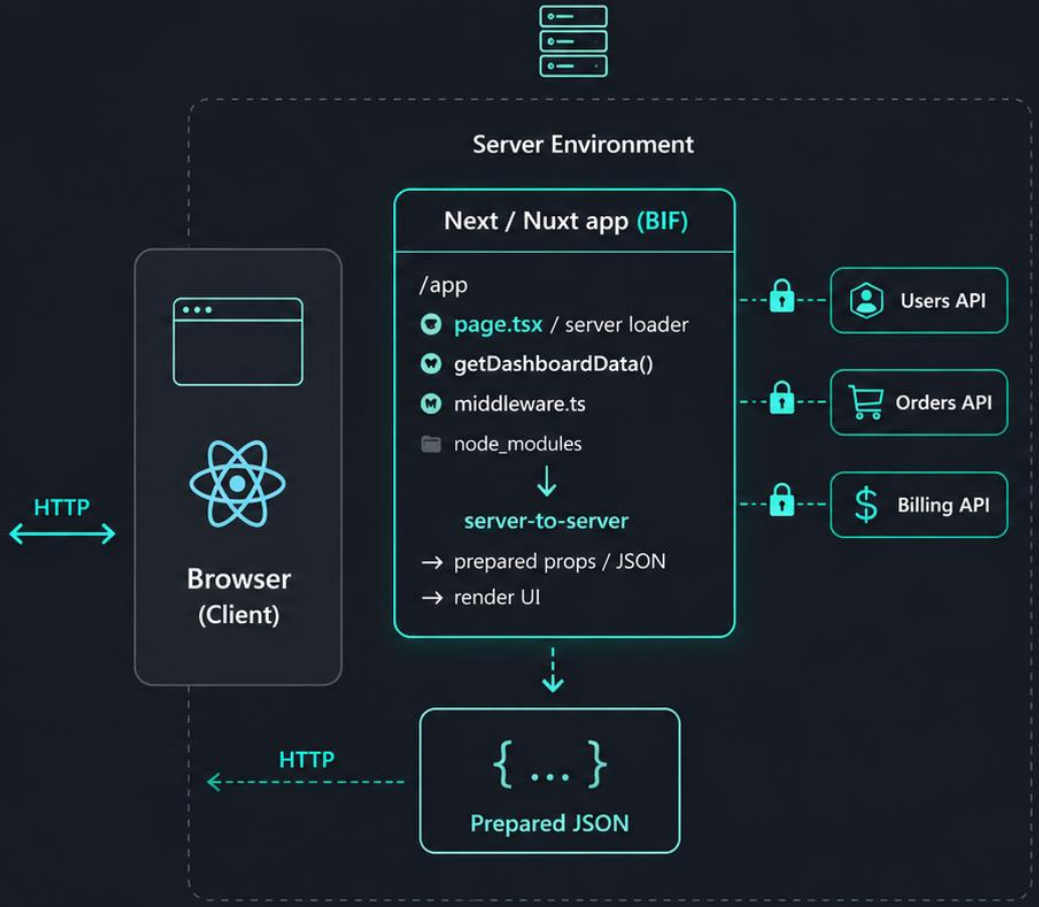


BIF VS BFF

BIF 2.0

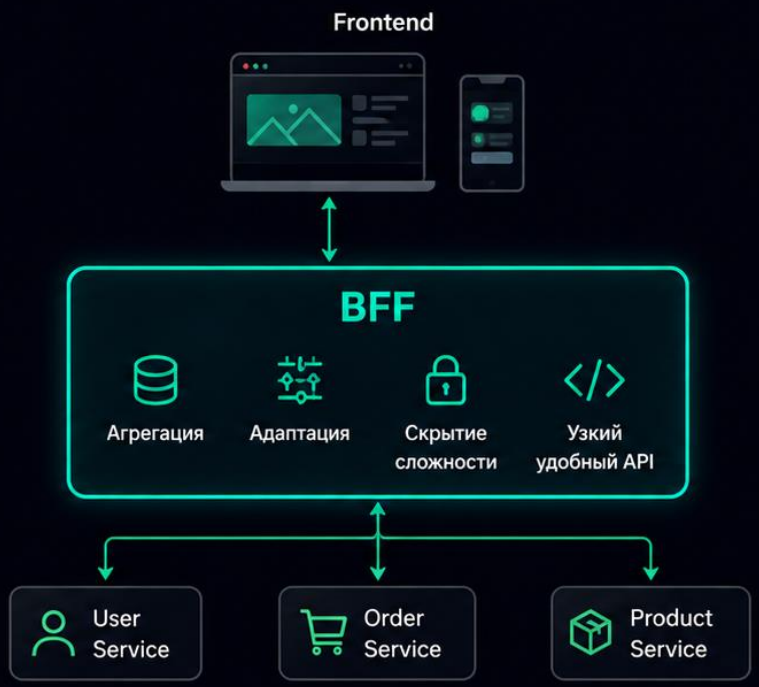
```
export default async function DashboardPage(userId) {
  const user = await getUser(userId)
  const orders = await getOrders(userId)
  const balance = await getBalance(userId)

  return (
    <Dashboard
      data={{
        name: user.name,
        ordersCount: orders.total,
        balance: balance.amount,
      }}
    />
  )
}
```



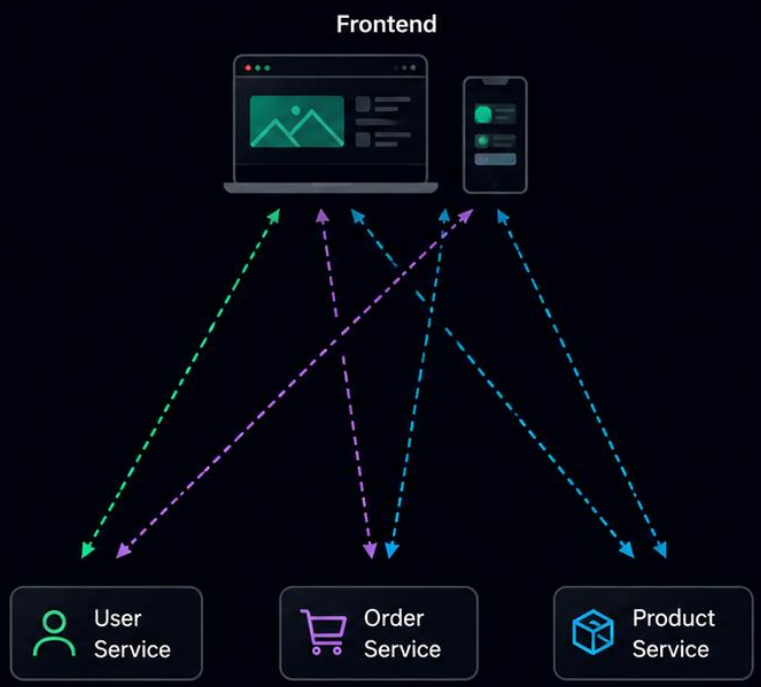
kaspersky

С BFF



Frontend работает с пользовательским сценарием

БЕЗ BFF

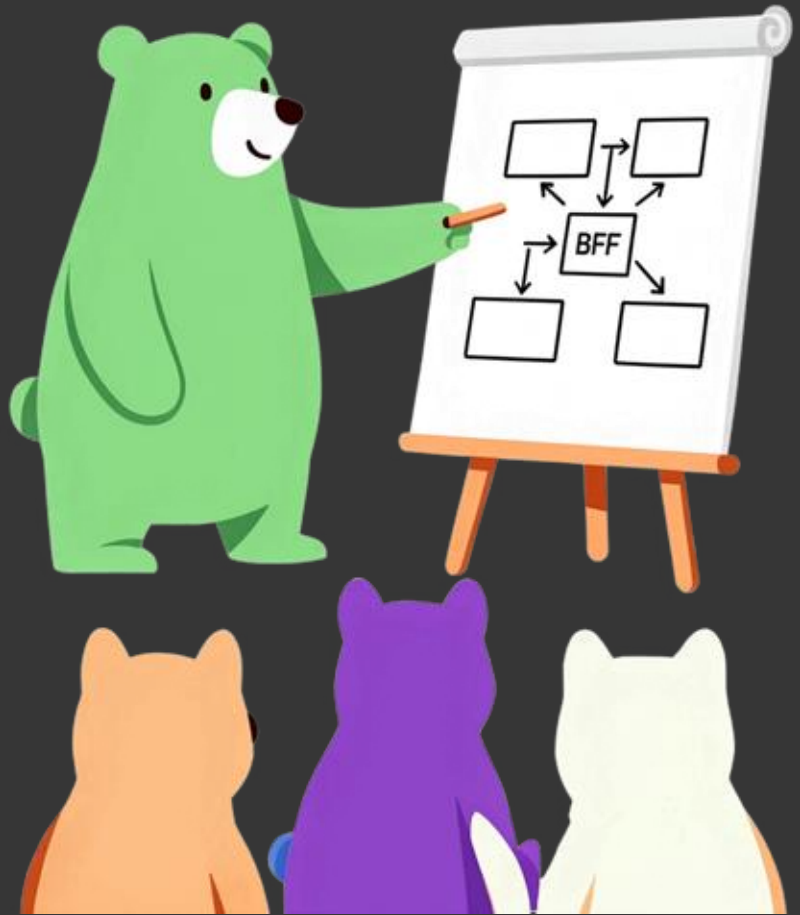


Frontend работает с деталями интеграции

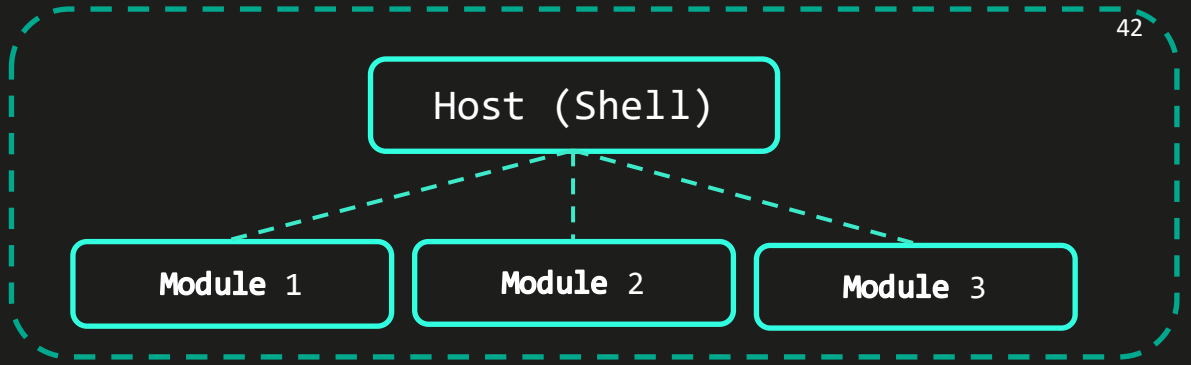
kaspersky

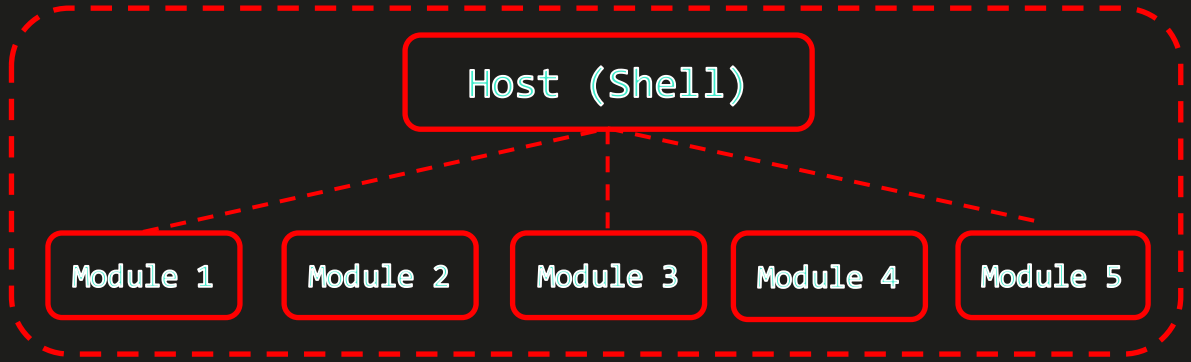
ОБЩИЙ BFF

Пока система небольшая — общий BFF выглядит разумно



- Единая точка входа
- Общий контекст
- Простое сопровождение


Host (Shell)
Module 1
Module 2
Module 3


Host (Shell)
Module 1
Module 2
Module 3
Module 4
Module 5

ОБЩИЙ BFF

Host

Микрофронтенд 1

Микрофронтенд 2

Микрофронтенд 3

Общий BFF

Сервисы по доменам

User

Order

Billing

kaspersky

ОБЩИЙ BFF

Host

Микрофронтенд 1

Микрофронтенд 2

Микрофронтенд 3

Общий BFF

Сервисы по доменам

User

Order

Billing

kaspersky

ОБЩИЙ BFF

Host

Микрофронтенд 1

Микрофронтенд 2

Микрофронтенд 3

Общий BFF

Сервисы по доменам

User

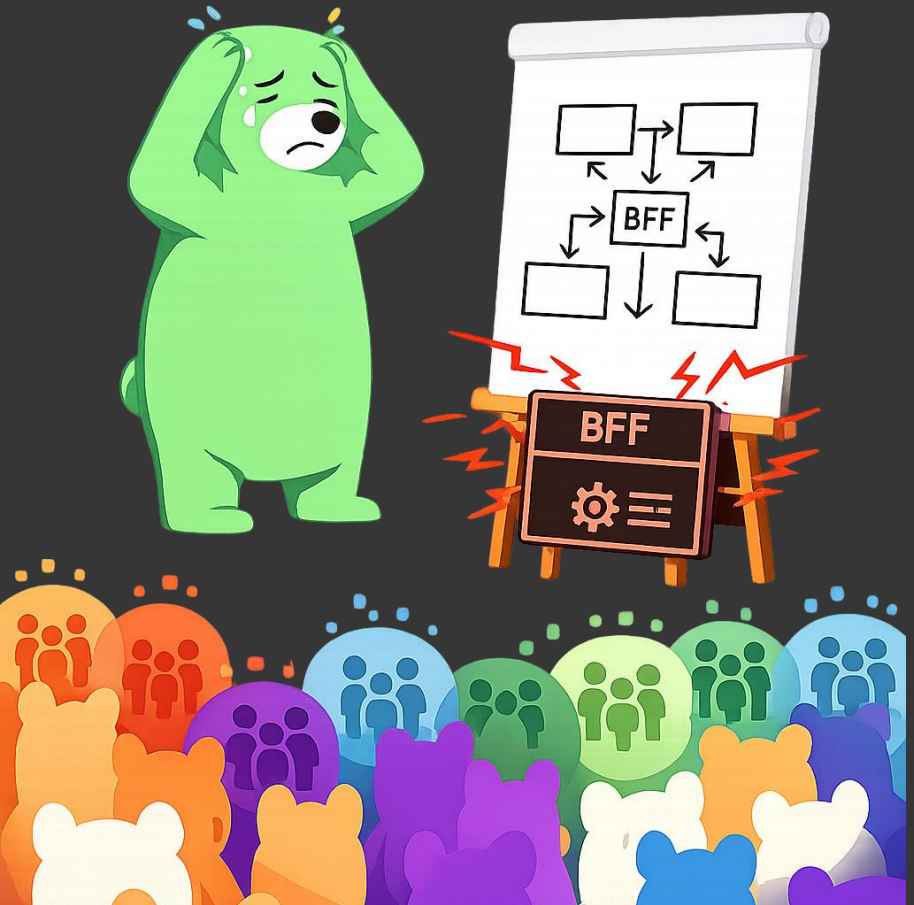
Order

Billing

kaspersky

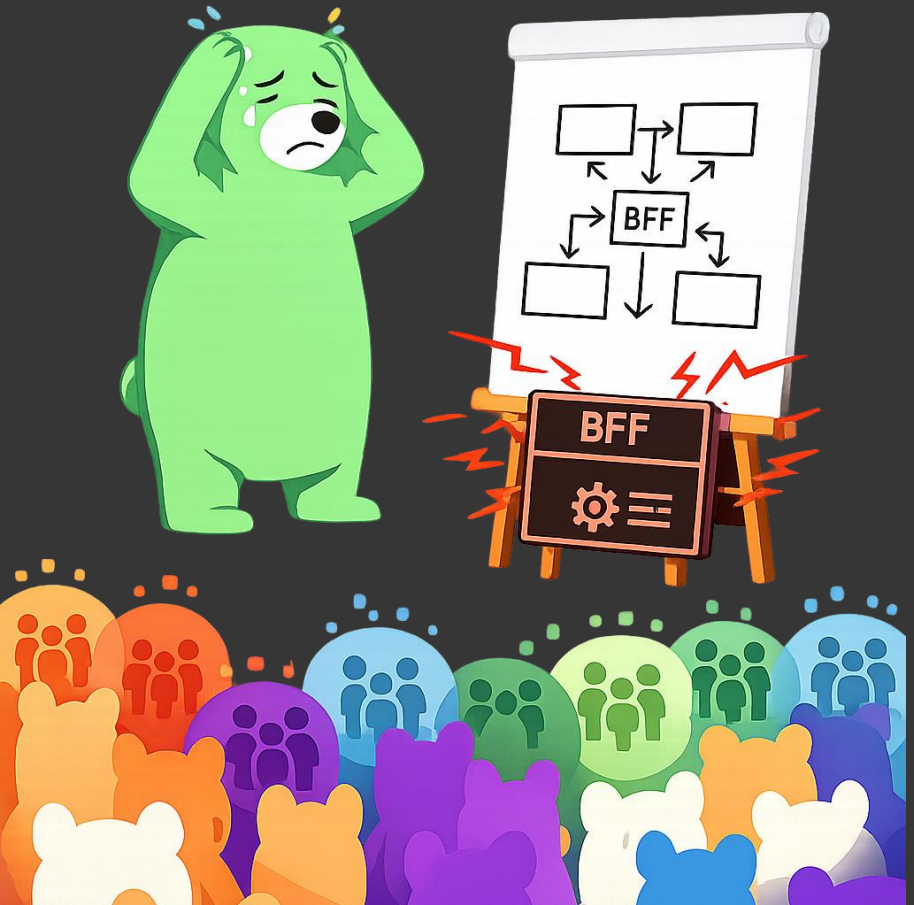
ОБЩИЙ BFF

Больше согласований



kaspersky

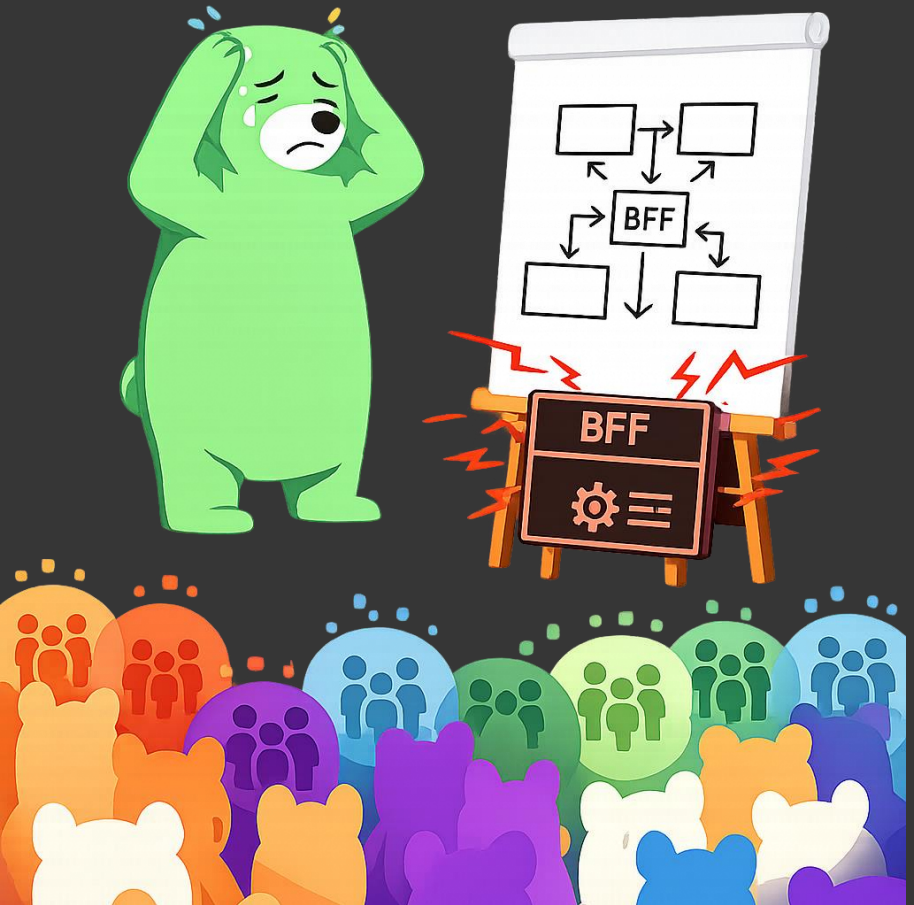
ОБЩИЙ BFF



Больше согласований

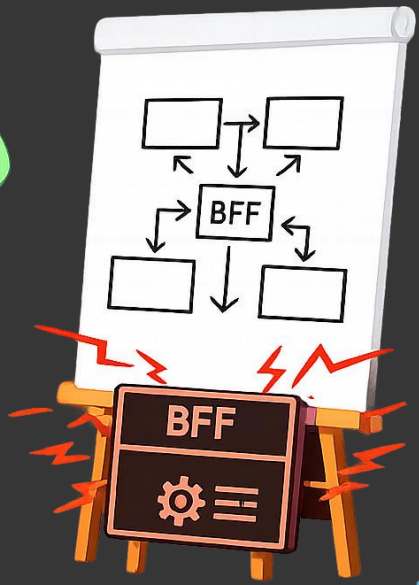
Размытые границы

ОБЩИЙ BFF



- Больше согласований
- Размытые границы
- Дорогие релизы

ОБЩИЙ BFF



- Больше согласований
- Размытые границы
- Дорогие релизы
- Риск задеть соседа

kaspersky

ОБЩИЙ BFF

On-Prem поставки

один продукт — разные конфигурации

Клиент A
Лицензия: Базовая
Модули:

Клиент B
Лицензия: Стандарт
Модули:

Клиент C
Лицензия: Расширенная
Модули:

Монолитный BFF

```
if (license === 'base') { ... }
if (moduleA.enabled) { ... }
if (featureX) { ... }
if (config.variant === 'custom') { ... }
 // ... много условий и флагов
...
```

Проблемы монолитного BFF

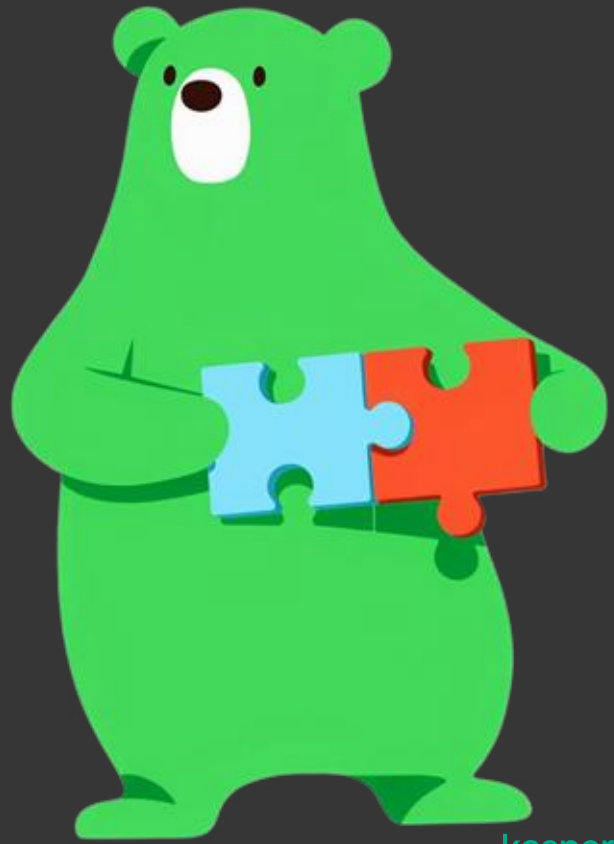
- Приходится вырезать куски под поставку
- Или оставлять лишний код и обрастать условиями и feature flags
- Серверный слой подстраивается под каждую конфигурацию
- Часть модулей клиенту вообще не нужна

Итог: больше сложность, выше стоимость поддержки, медленнее изменения

kaspersky

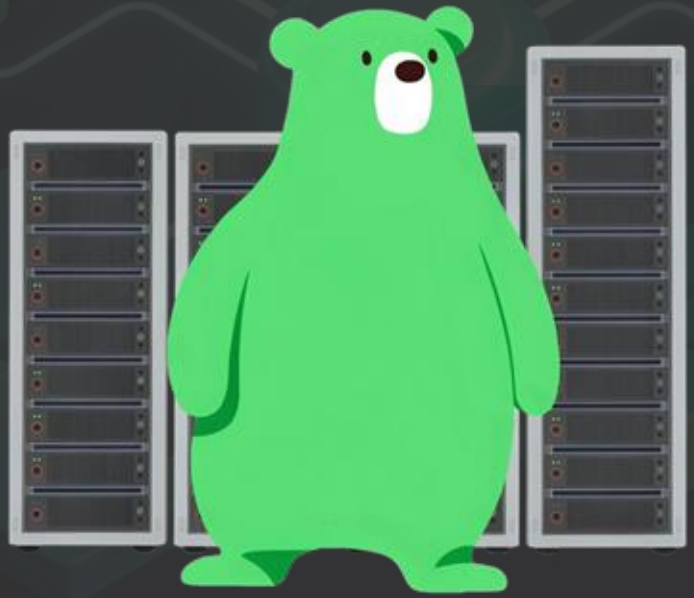
BIF VS BFF

Какое решение ?!

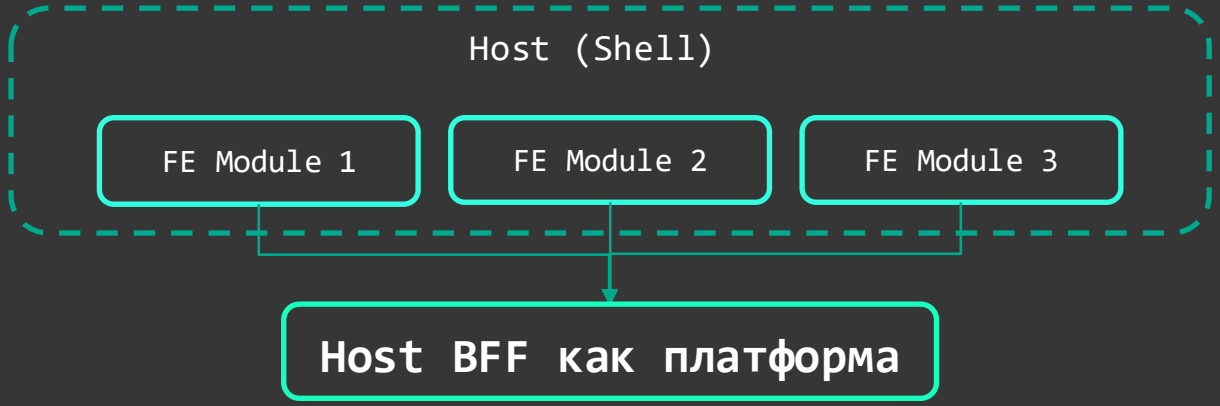


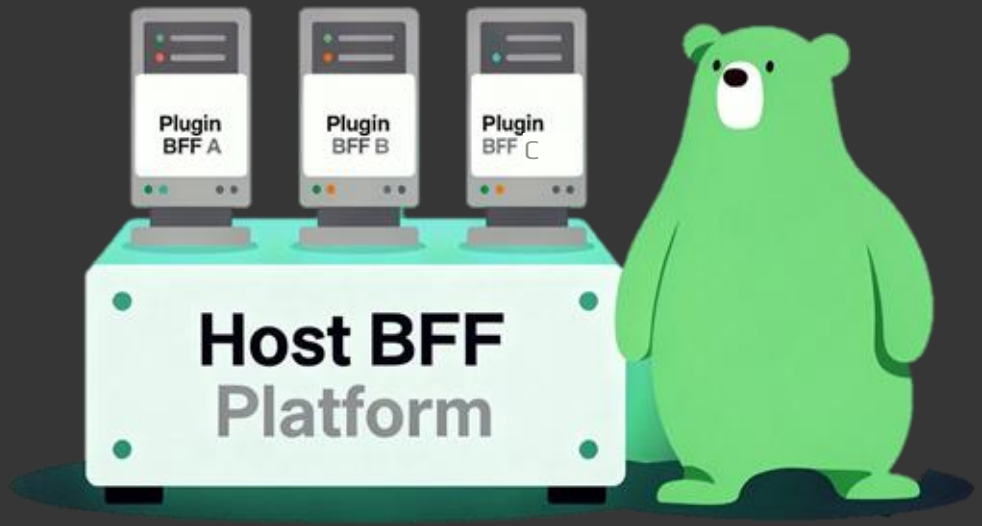
kaspersky

Host BFF как платформа



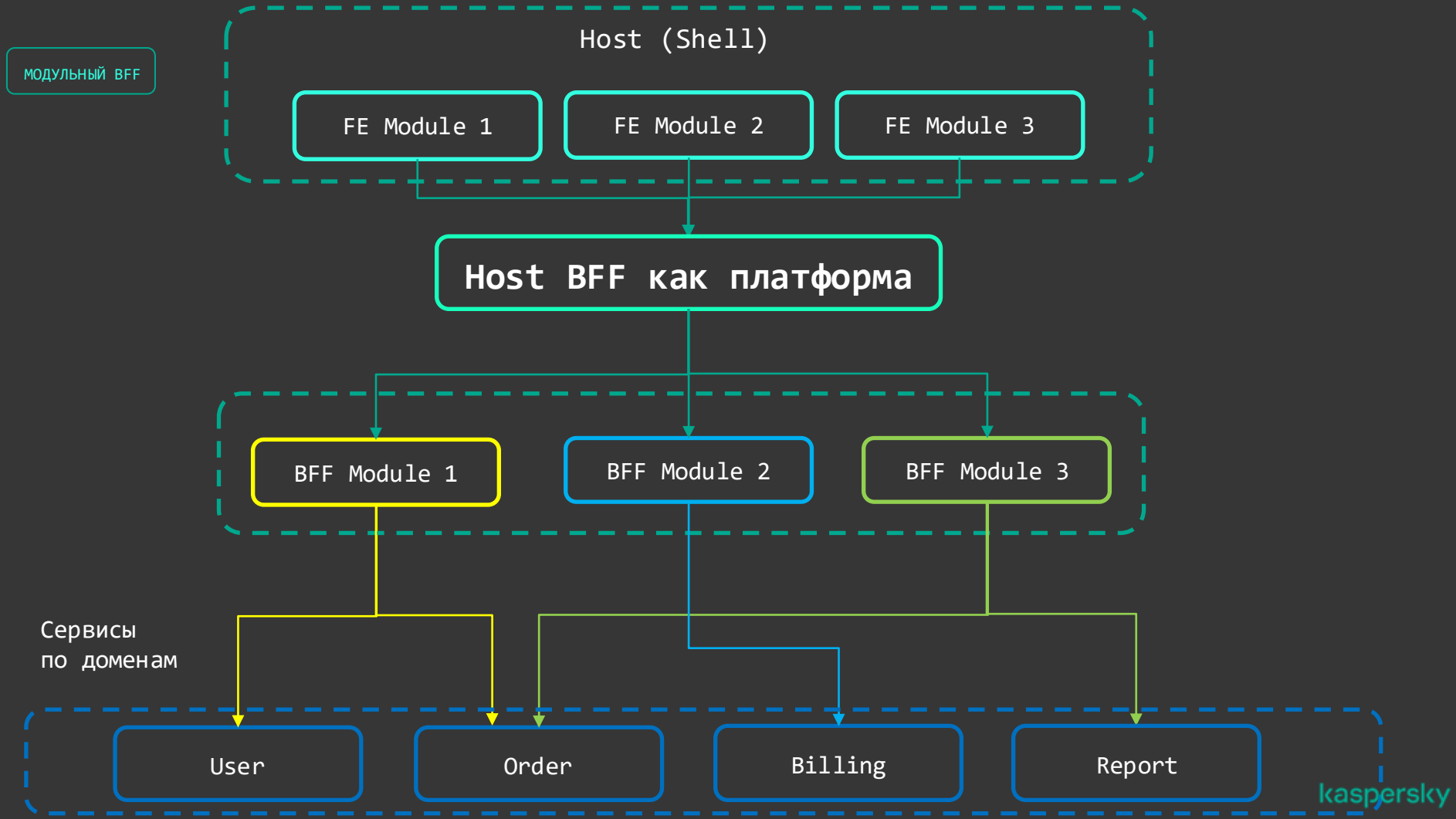
МОДУЛЬНЫЙ BFF



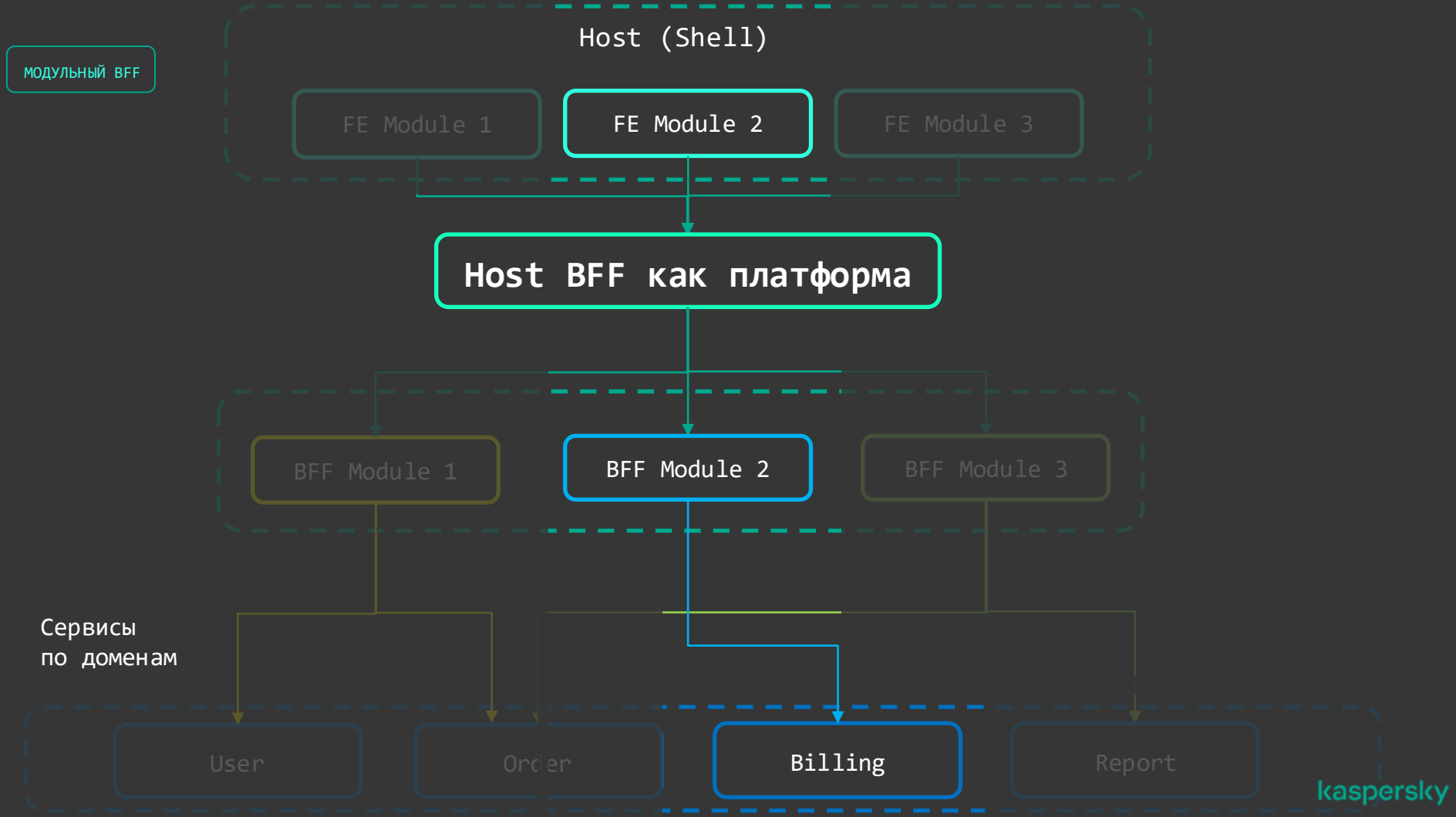


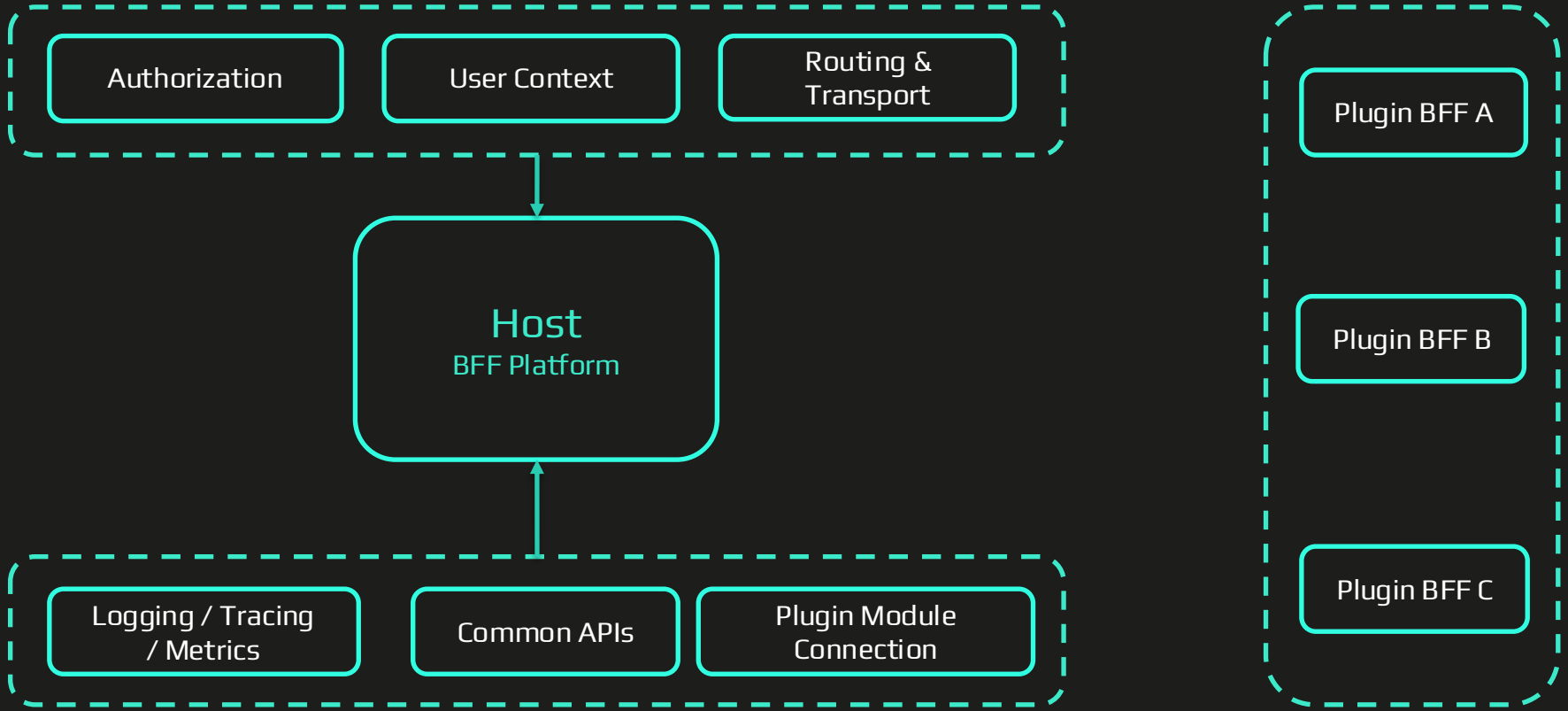
kaspersky

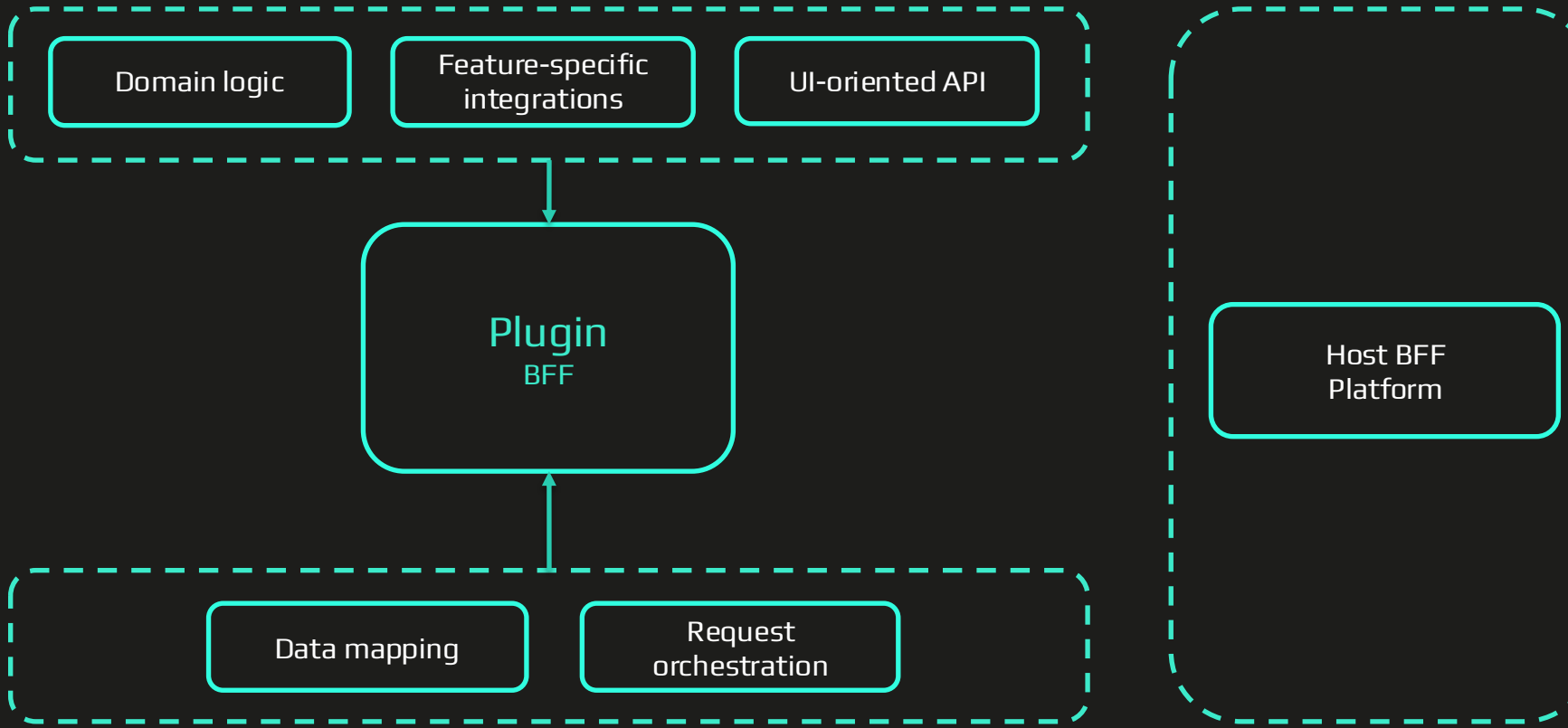
МОДУЛЬНЫЙ BFF



kaspersky

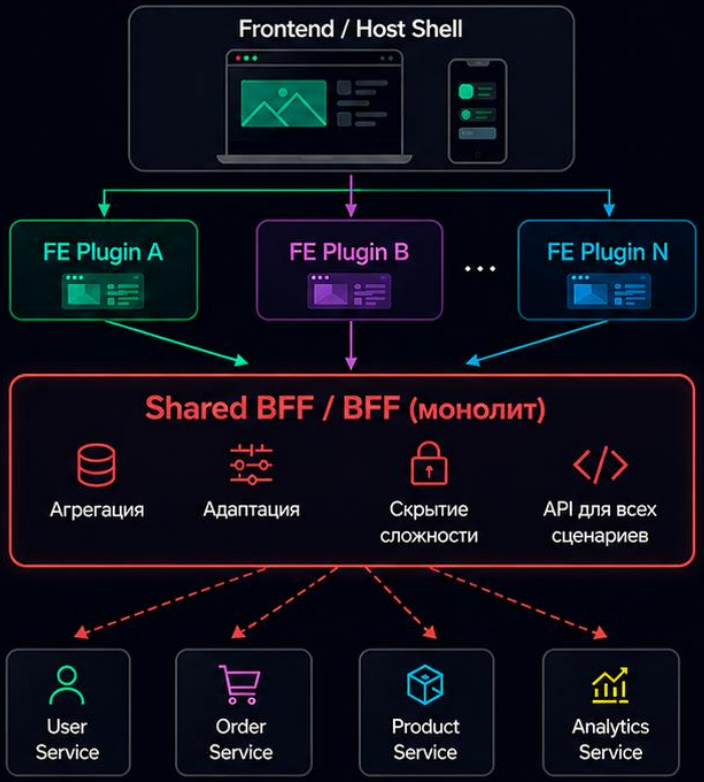

МОДУЛЬНЫЙ BFF
Host (Shell)
FE Module 1
FE Module 2
FE Module 3
Host BFF как платформа
BFF Module 1
BFF Module 2
BFF Module 3
Сервисы
по доменам
User
Order
Billing
Report
kaspersky


Authorization
User Context
Routing & Transport
Host
BFF Platform
Logging / Tracing / Metrics
Common APIs
Plugin Module Connection
Plugin BFF A
Plugin BFF B
Plugin BFF C


Domain logic
Feature-specific integrations
UI-oriented API
Plugin
BFF
Data mapping
Request orchestration
Host BFF Platform

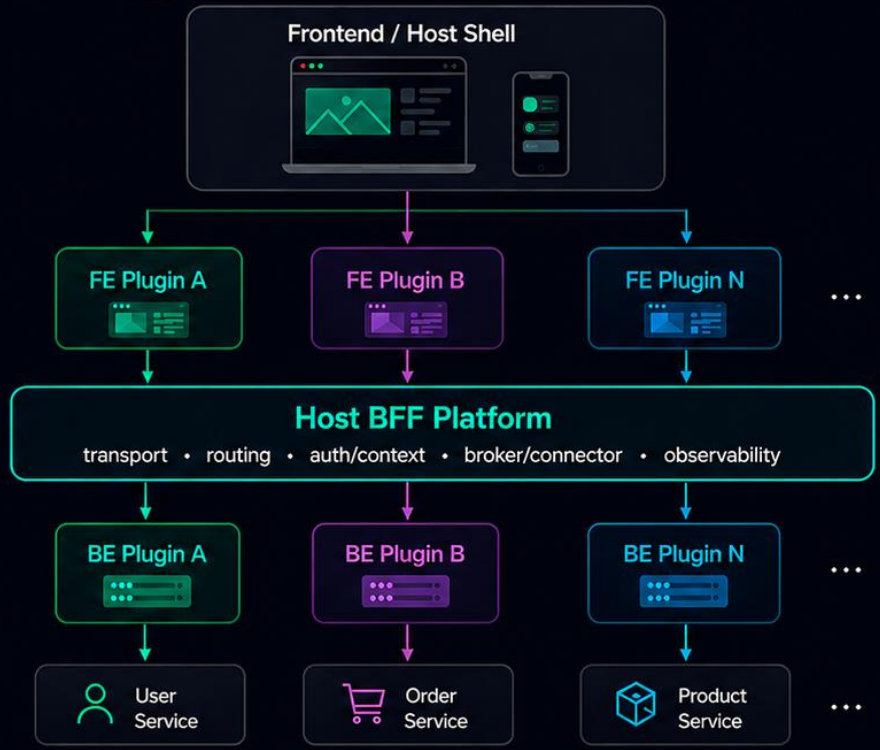
АРХИТЕКТУРА

Общий BFF



Frontend уже модульный,
но серверный слой всё ещё общий

Модульный BFF



Host остаётся общим слоем,
но доменная логика уезжает в BE Plugin команд

kaspersky

Как это выглядит на реальном модуле

config.json

```
{
  "plugin": {
    "id": "vulnerability_management",
    "version": "1.0.%REV%",
    "integration": {
      "routes": {
        "getter": "handlers.getRoutes",
        "prefix": "vulnerability-management"
      }
    }
  }
}
```

Client API

```
export const userApi =
api.injectEndpoints({
  endpoints: (build) => ({
    getAllPermissions: build.query({
      queryFn: (args) =>
        createQueryFn(
          server.user.getAllPermissions
        )(args)
    })
  })
})
```

Server API Contract

```
export type UserServiceSchema = {
  getAllPermissions: () => Promise<{
    data: Permissions
  }>
}
```

FE Plugin видит простой typed API,
а host-платформа берёт на себя транспорт и маршрутизацию

МОДУЛЬНЫЙ BFF

1

Build-time

Команда собирает артефакт плагина: клиентский bundle, серверный build, manifest, config.json, схемы интеграции

kaspersky

МОДУЛЬНЫЙ BFF

1
Build-time

Команда собирает артефакт плагина: клиентский bundle, серверный build, manifest, config.json, схемы интеграции

2
Deploy

Артефакты плагина публикуются независимо от основной платформы

kaspersky

МОДУЛЬНЫЙ BFF

1
Build-time

Команда собирает артефакт плагина: клиентский bundle, серверный build, manifest, config.json, схемы интеграции

2
Deploy

Артефакты плагина публикуются независимо от основной платформы

3
Runtime: Load

Host подхватывает артефакт, загружает клиентские bundles через init-product-plugins

kaspersky

МОДУЛЬНЫЙ BFF

1
Build-time

Команда собирает артефакт плагина: клиентский bundle, серверный build, manifest, config.json, схемы интеграции

2
Deploy

Артефакты плагина публикуются независимо от основной платформы

3
Runtime: Load

Host подхватывает артефакт, загружает клиентские bundles через init-product-plugins

4
Runtime: Register

Plugin backend делает announce config + announce API, host обновляет реестр

kaspersky

МОДУЛЬНЫЙ BFF

1
Build-time

Команда собирает артефакт плагина: клиентский bundle, серверный build, manifest, config.json, схемы интеграции

2
Deploy

Артефакты плагина публикуются независимо от основной платформы

3
Runtime: Load

Host подхватывает артефакт, загружает клиентские bundles через init-product-plugins

4
Runtime: Register

Plugin backend делает announce config + announce API, host обновляет реестр

5
Runtime: Route

Host начинает маршрутизировать трафик в зарегистрированный plugin server

kaspersky

Почему нельзя просто положить всё в конфиг?

АРХИТЕКТУРА

config.json

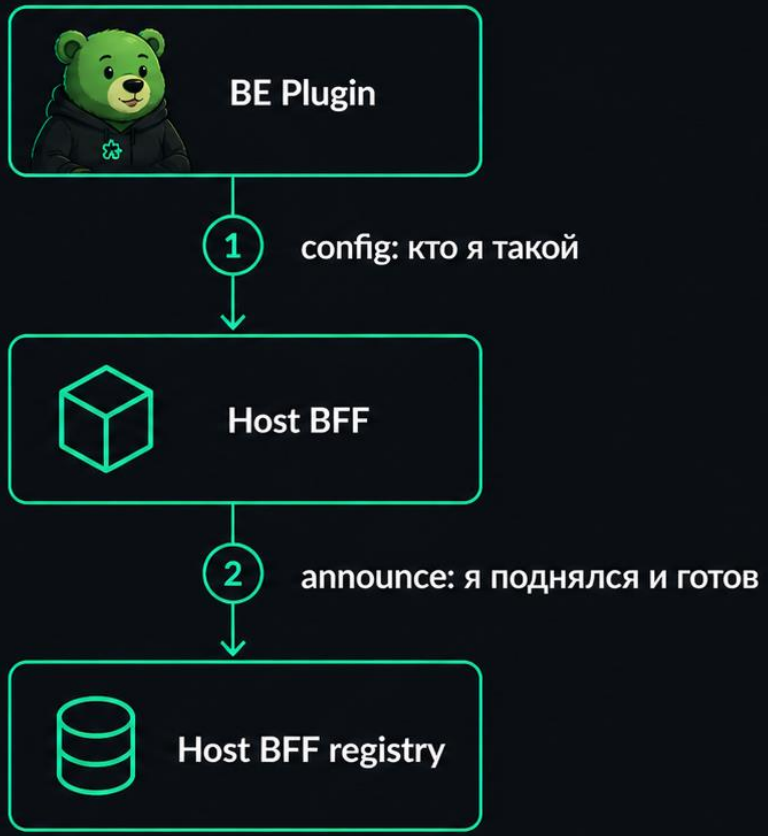
- id
- version
- entrypoints
- settings

А сейчас модуль:

- поднялся?
- подключился?
- живой?
- готов принимать вызовы?

Как Host узнаёт, что модуль готов

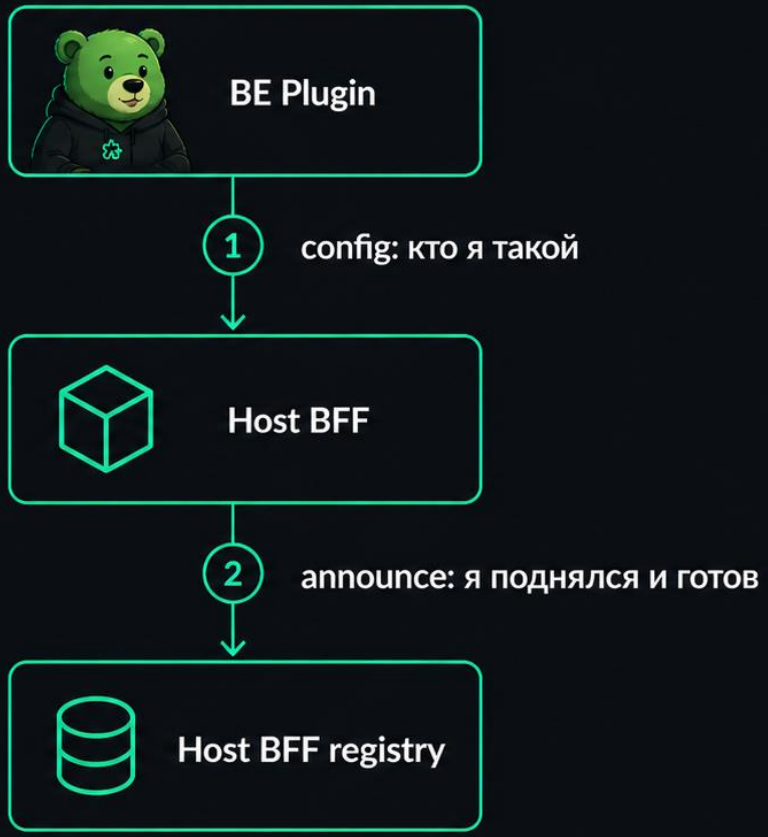
АРХИТЕКТУРА




```
const config = {
   id: 'cookie-module',
   version: '1.0.0',
   integration: {
      service: {
         hasServerFunctions: true
      }
   }
}

connector.announceConfig(config)
connector.announceApi(['getCookies', 'setCookie'])
```

Как Host узнаёт, что модуль готов




```
const config = {
   id: 'cookie-module',
   version: '1.0.0',
   integration: {
      service: {
         hasServerFunctions: true
      }
   }
}

connector.announceConfig(config)
connector.announceApi(['getCookies', 'setCookie'])
```

* Коннектор - это адаптер над брокером сообщений

PLUGIN BFF

Как это выглядит на клиенте?

plugin-client.ts TS

```ts
// Просто вызов метода своего модуля

const assets = await pluginServerApi.getAssets()

const vulns = await pluginServerApi.getVulnerabilities({ ... })
```

АРХИТЕКТУРА

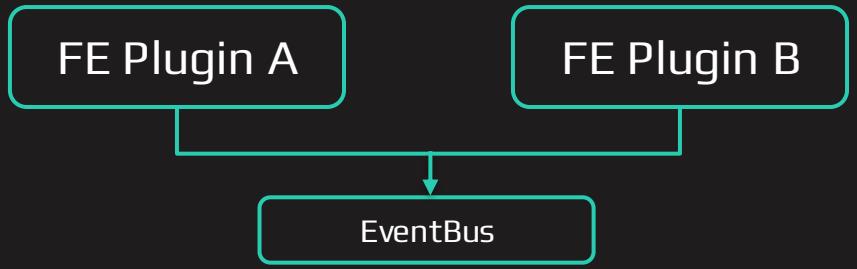
FE Plugin A

FE Plugin B

Как всё это устроено внутри?

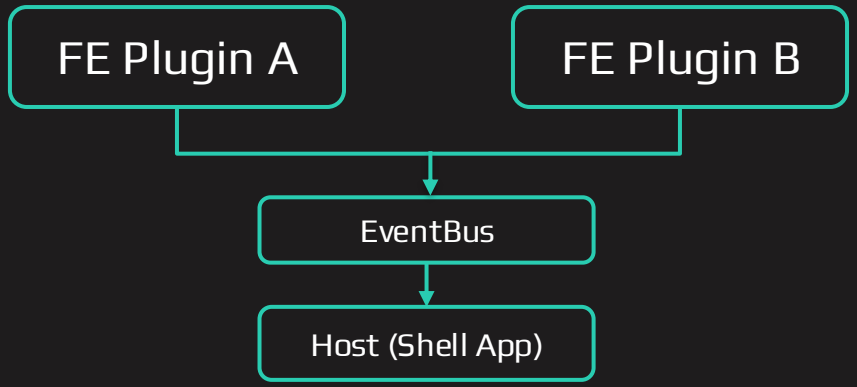
АРХИТЕКТУРА

Как всё это устроено внутри?



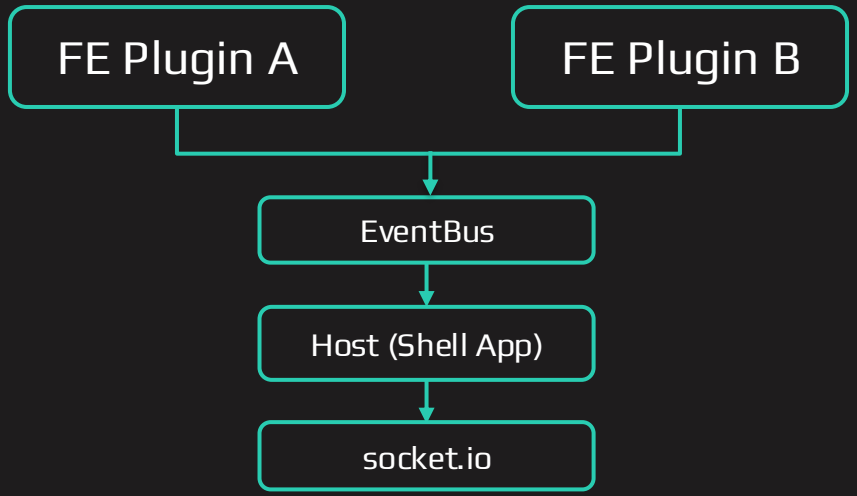
АРХИТЕКТУРА

Как всё это устроено внутри?



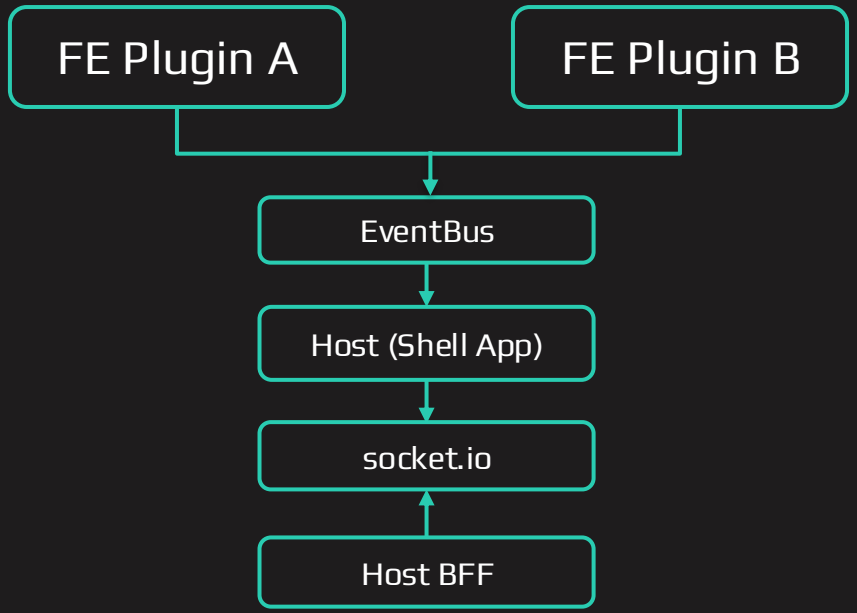
АРХИТЕКТУРА

Как всё это устроено внутри?



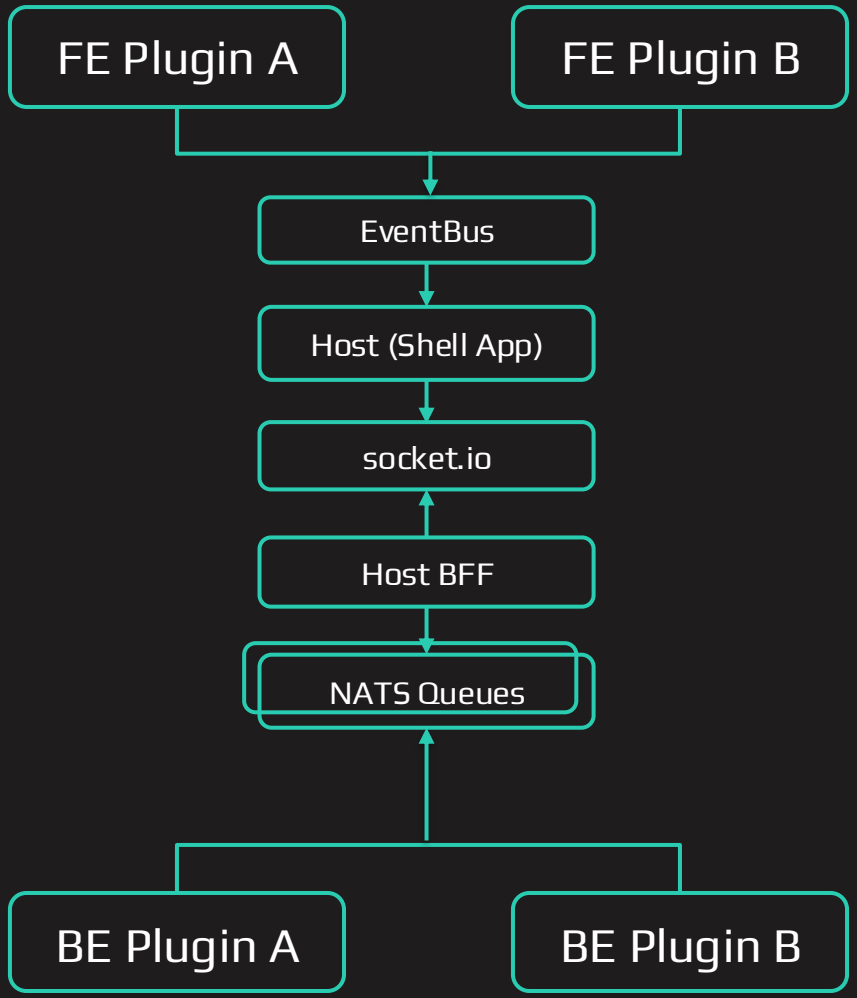
АРХИТЕКТУРА

Как всё это устроено внутри?



АРХИТЕКТУРА

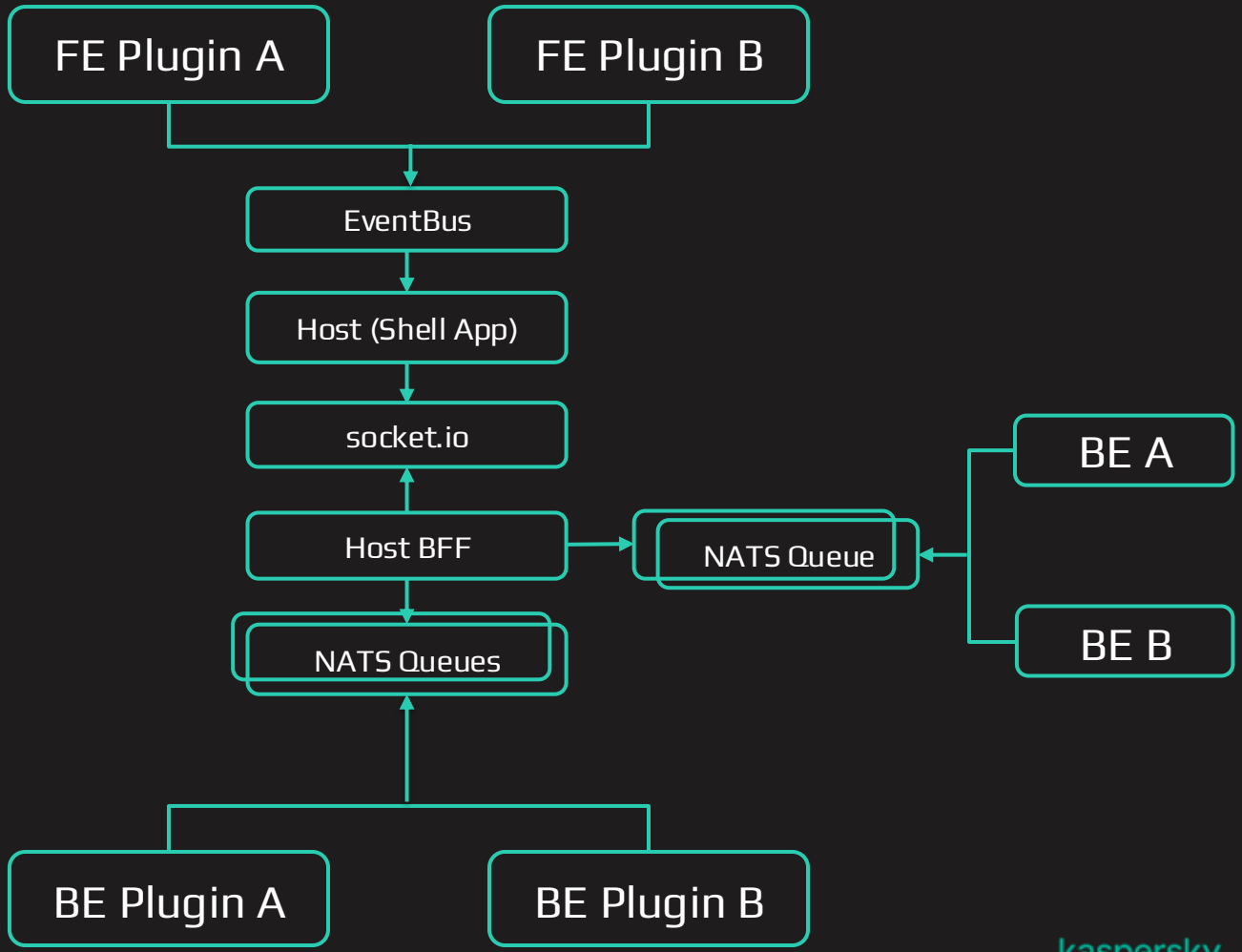
Как всё это устроено внутри?



kaspersky

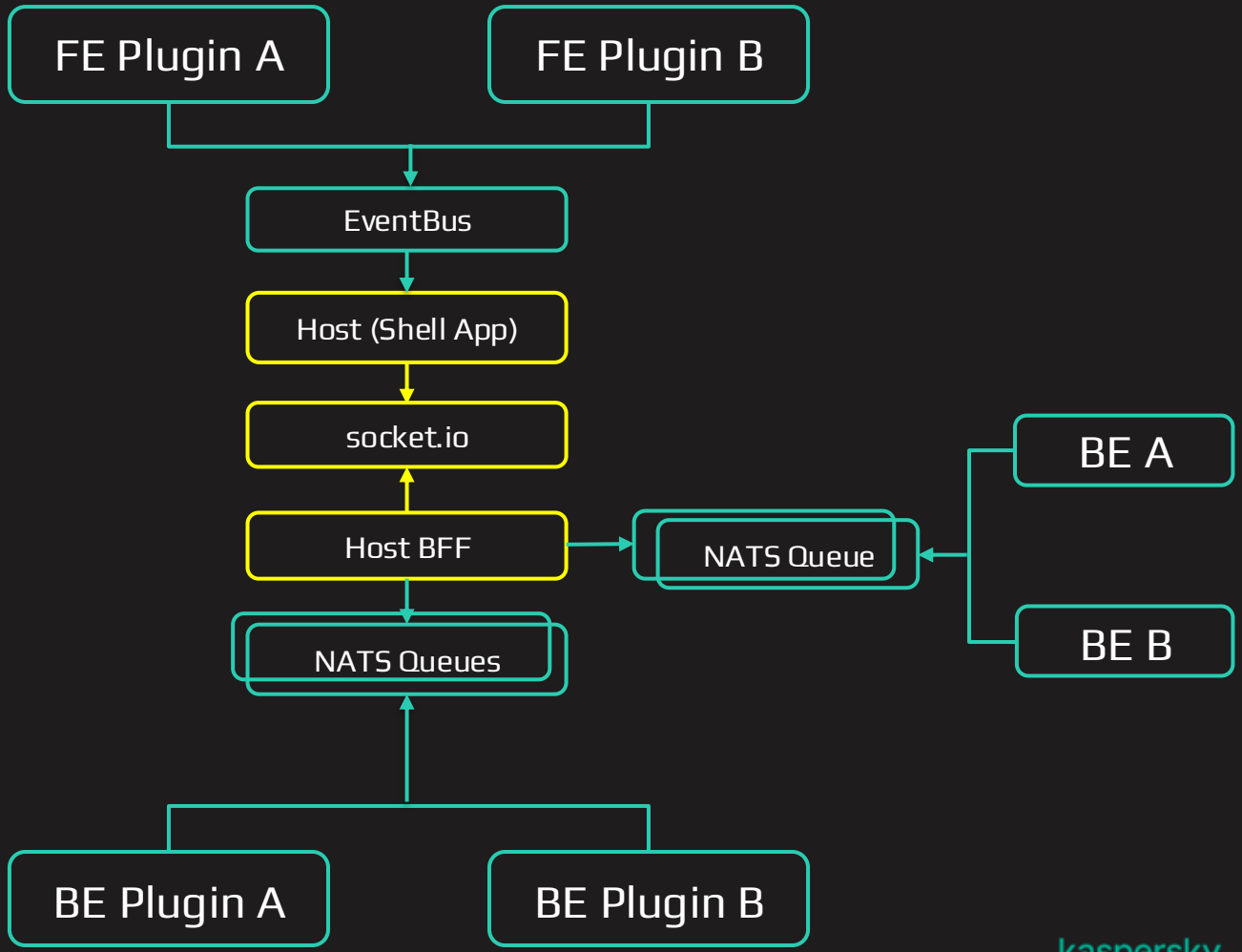
АРХИТЕКТУРА

Как всё это устроено внутри?



АРХИТЕКТУРА

Как всё это устроено внутри?



Как браузер общается с Host

· постоянный канал
· контекст сессии
· восстановление сессии


```
{
  "method": "getCookies",
  "params": { "userId": "42" },
  "reqId": "req_12345"
}
```

универсальный вызов
вместо отдельной REST-ручки

АРХИТЕКТУРА


```
{
  moduleId: 'cookie-module',
  method: 'getCookies',
  params: { userId },
  reqId: 'req_12345'
}
```

Много эндпоинтов
каждый метод — отдельный API

Сложно поддерживать
разные схемы, версии, документация

Плохо масштабируется
сложно развивать и эволюционировать

kaspersky

АРХИТЕКТУРА

Вызов метода как обычно

```
await api.getCookies({
  userId
})
```

Формируется универсальное сообщение (RPC)

```
{
  moduleId: 'cookie-module',
  method: 'getCookies',
  params: { userId },
  reqId: 'req_12345'
}
```

Host маршрутизирует в нужный plugin

Host

смотрит на moduleId и method

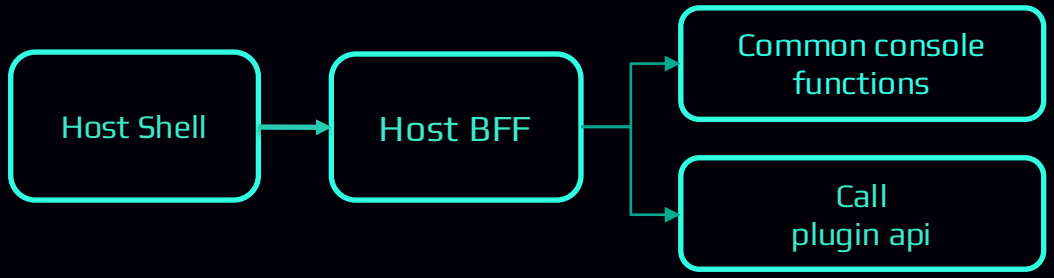
Plugin выполняет handler

BE Plugin
(cookie-module)
выполняет нужный handler

Результат возвращается обратно к клиенту

kaspersky

Host – единая точка обработки вызовов



host.service.ts

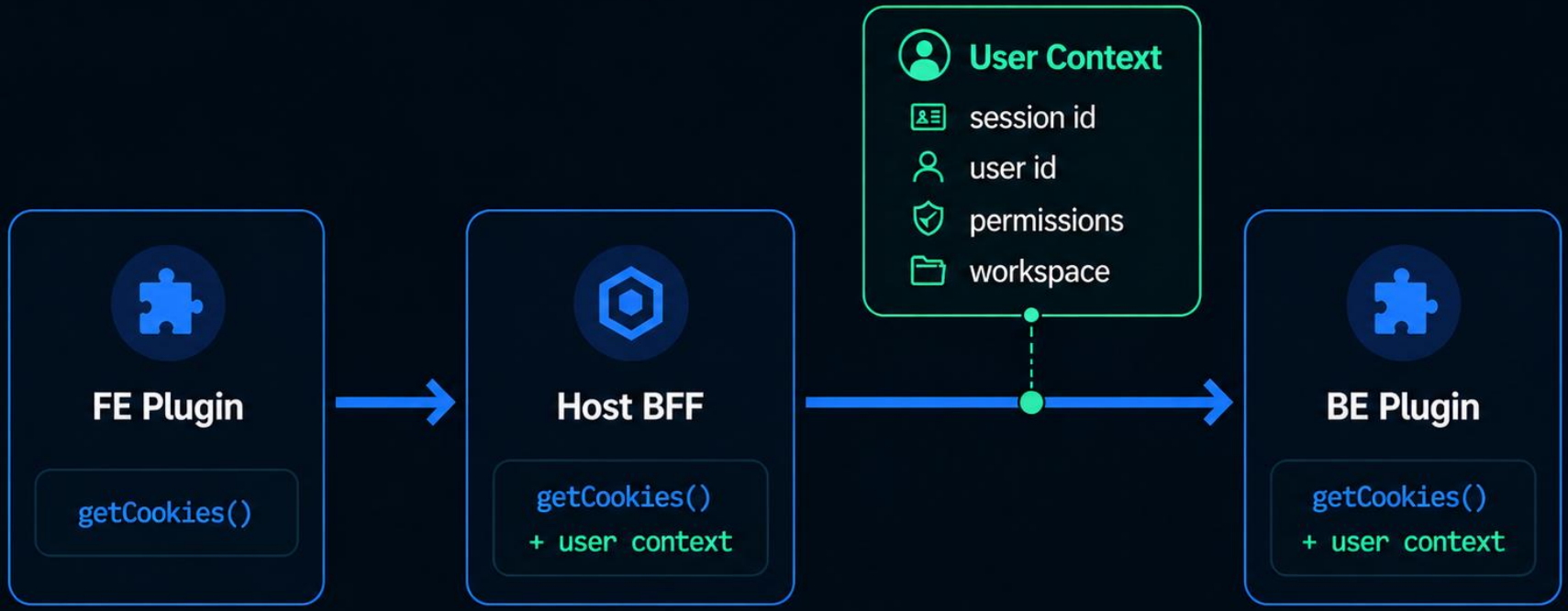
```ts
// После подключения Host становится центральной точкой обработки вызовов.

onRequest((request) => {

  if (isPlatformCall(request)) {
    return handlePlatformCall(request)
  }
  return callPluginApi(request)
})
```

АРХИТЕКТУРА | BFF

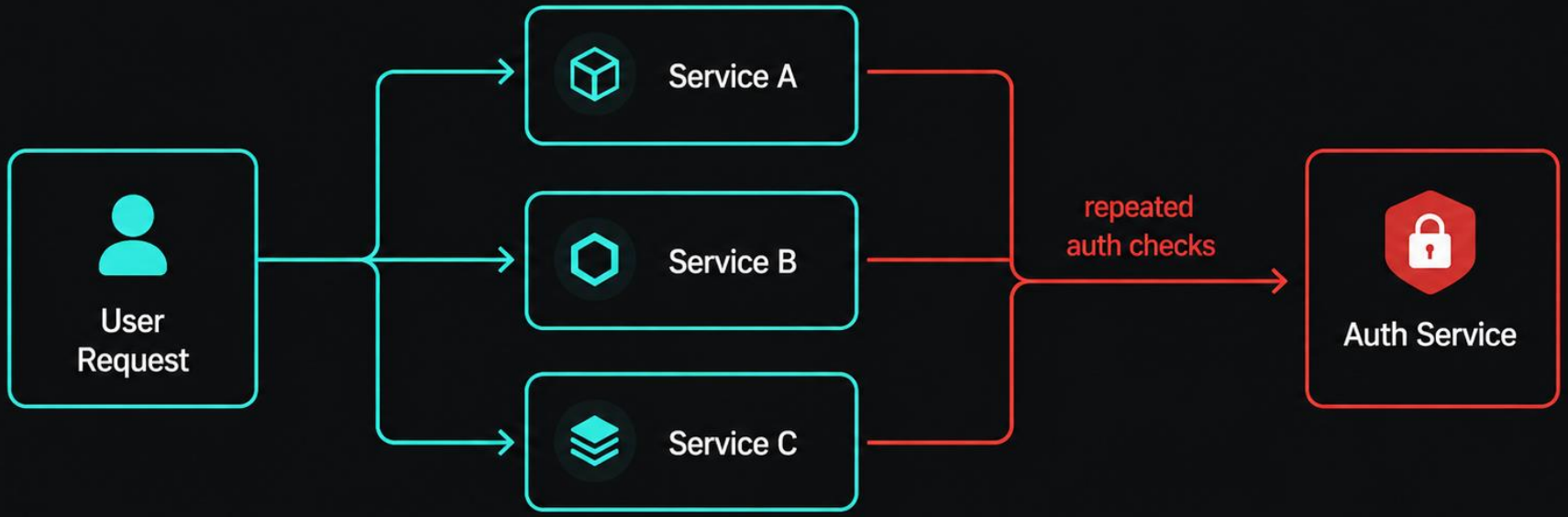
А что едет вместе с запросом?



BE Plugin получает не только название метода,
но и пользовательский контекст, чтобы выполнить запрос от имени конкретного пользователя.

Почему backend’у неудобно работать с session id

АРХИТЕКТУРА

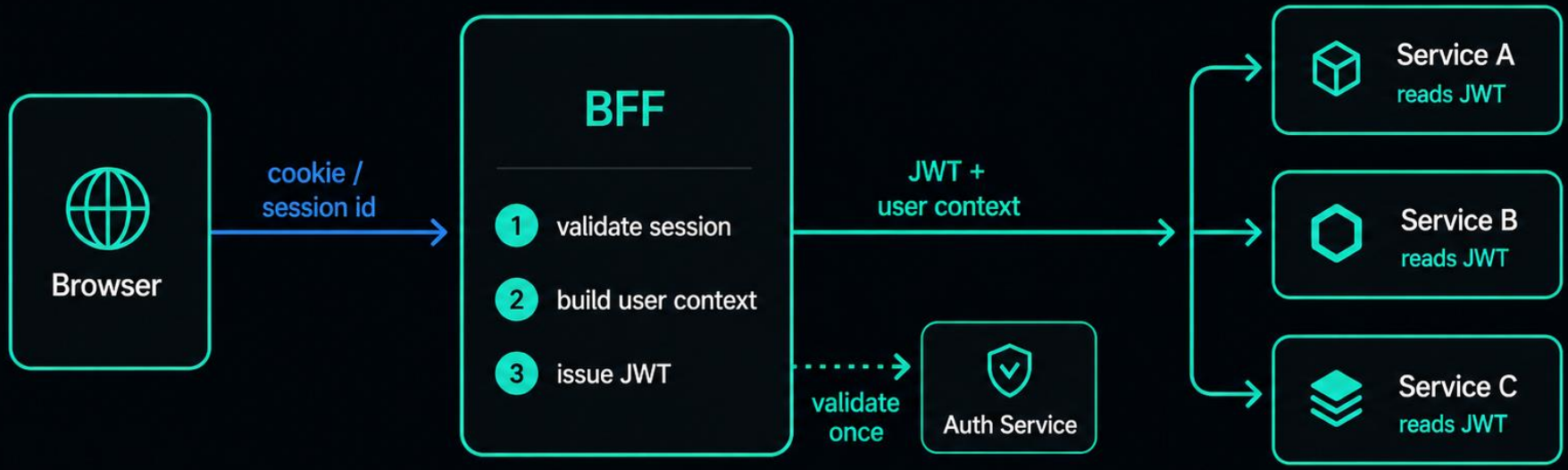


Лишние сетевые вызовы → latency → нагрузка на Auth Service

АРХИТЕКТУРА

BFF как точка обмена session → JWT

Один раз проверяем сессию, дальше передаём внутренний токен

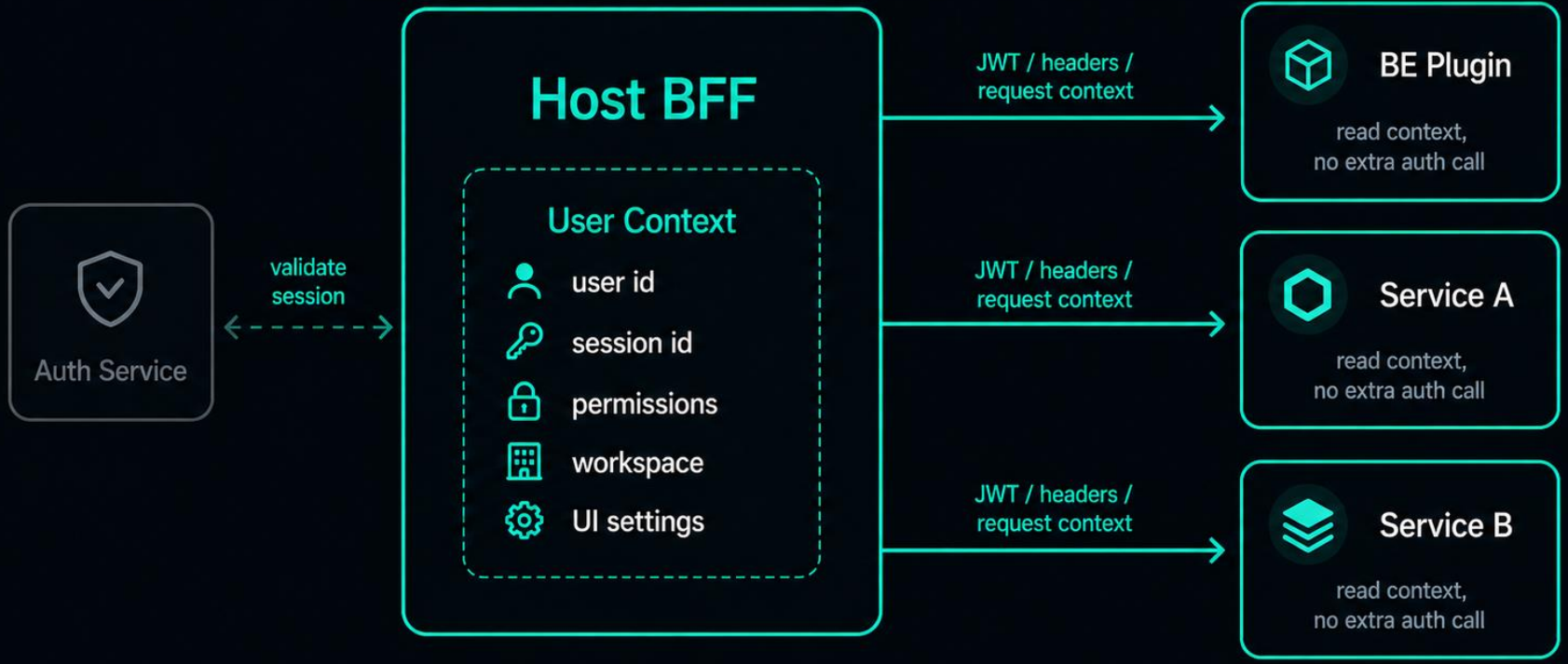


Меньше повторных auth-запросов → ниже latency → проще backend-контракты

АРХИТЕКТУРА

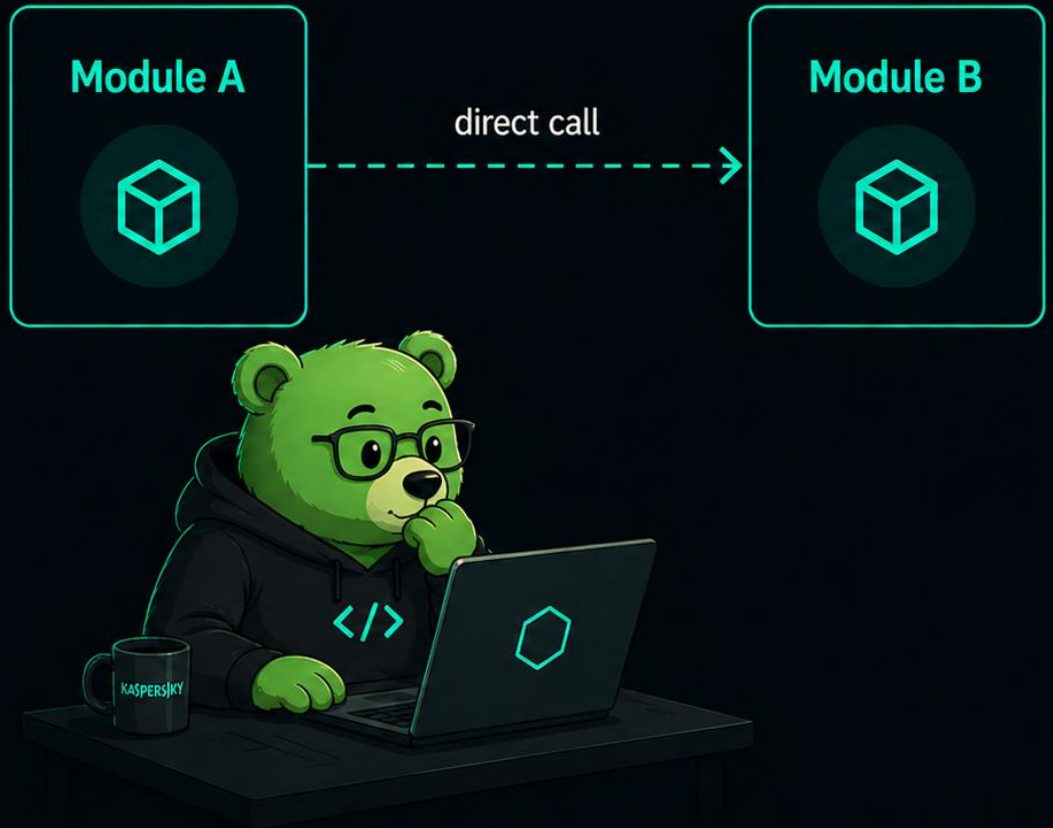
BFF собирает пользовательский контекст

Один контекст для FE Plugin, BE Plugin и backend-сервисов



Прямые зависимости между модулями — скрытая проблема

АРХИТЕКТУРА



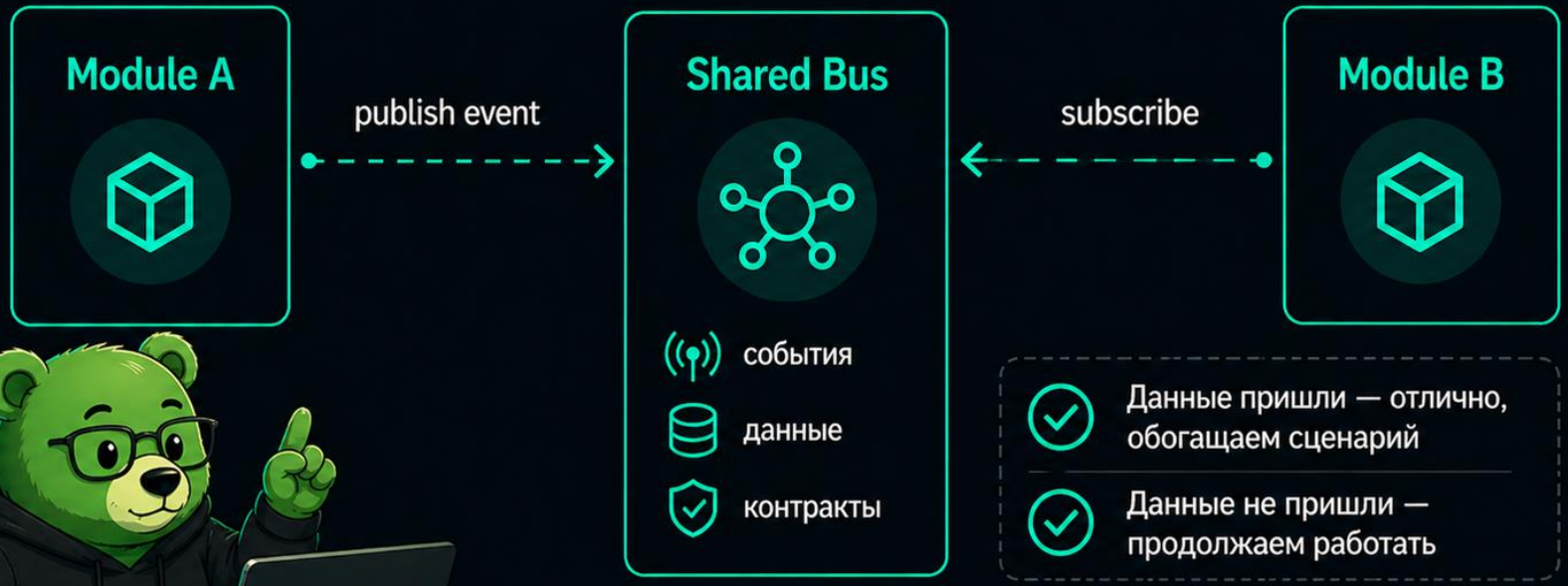
Hidden coupling:

- чужой контракт становится вашей зависимостью
- релизы начинают цепляться
- команды ломают друг друга

kaspersky

Взаимодействие модулей — через явный контракт, платформу или события

АРХИТЕКТУРА



Нет прямой зависимости → меньше связности → устойчивее релизы

kaspersky

Что даёт модульный BFF командам

АРХИТЕКТУРА

Общий BFF

Модульный BFF

Модульность **снижает связность** не только в коде, но и между командами

kaspersky
kaspersky

Что даёт модульный BFF командам

АРХИТЕКТУРА

Общий BFF

больше согласований

Модульный BFF

Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

АРХИТЕКТУРА

Общий BFF

больше согласований

Модульный BFF

изолированный контракт

Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

АРХИТЕКТУРА

Общий BFF

- больше согласований
- общие интеграционные точки

Модульный BFF

- изолированный контракт

Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

АРХИТЕКТУРА

Общий BFF

- больше согласований
- общие интеграционные точки

Модульный BFF

- изолированный контракт
- внутри своей зоны ответственности

Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

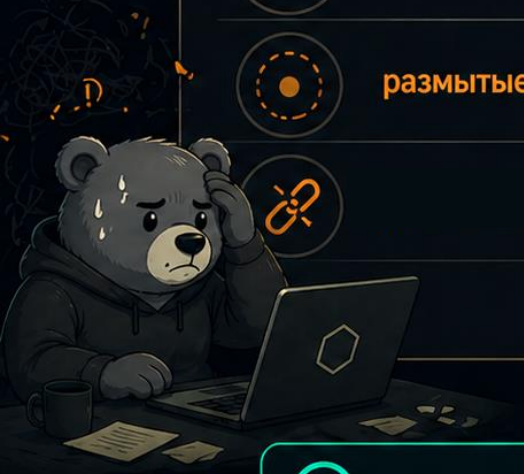
АРХИТЕКТУРА

Общий BFF

- больше согласований
- общие интеграционные точки
- размытые границы

Модульный BFF

- изолированный контракт
- внутри своей зоны ответственности



Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

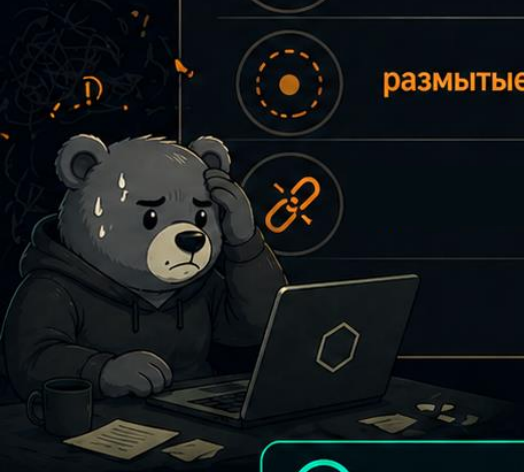
АРХИТЕКТУРА

Общий BFF

- больше согласований
- общие интеграционные точки
- размытые границы

Модульный BFF

- изолированный контракт
- внутри своей зоны ответственности
- границы строго определены за счёт контрактов



Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

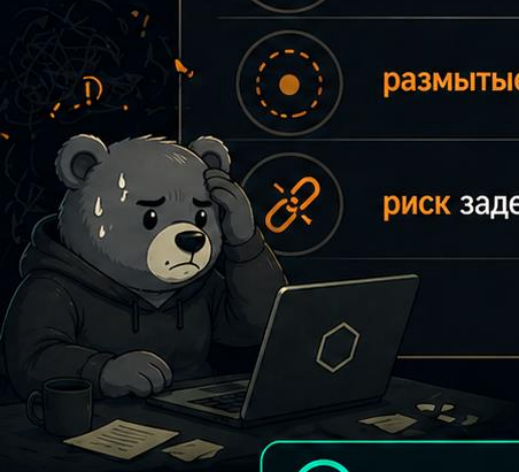
АРХИТЕКТУРА

Общий BFF

- больше согласований
- общие интеграционные точки
- размытые границы
- риск задеть соседа

Модульный BFF

- изолированный контракт
- внутри своей зоны ответственности
- границы строго определены за счёт контрактов



Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

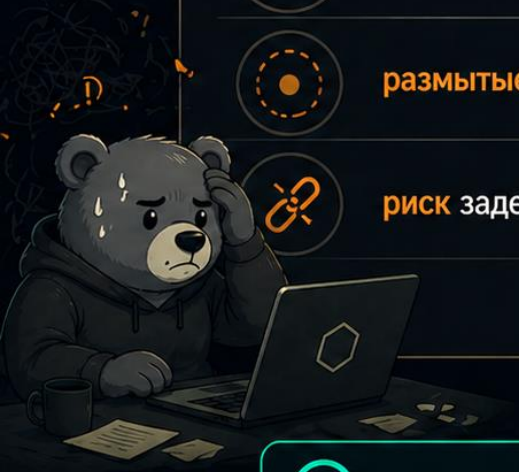
АРХИТЕКТУРА

Общий BFF

- больше согласований
- общие интеграционные точки
- размытые границы
- риск задеть соседа

Модульный BFF

- изолированный контракт
- внутри своей зоны ответственности
- границы строго определены за счёт контрактов
- минимизируем риск задеть соседа
  лишний раз не ходим к соседям



Модульность снижает связность не только в коде, но и между командами

kaspersky

Что даёт модульный BFF командам

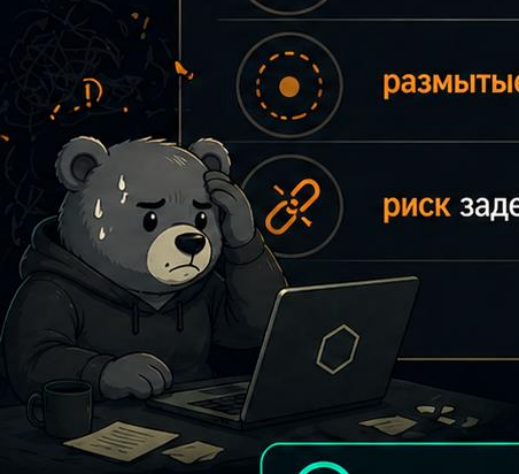
АРХИТЕКТУРА

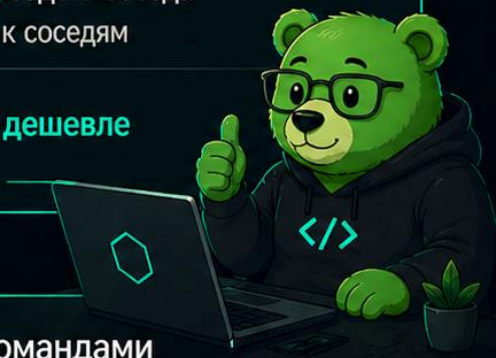
Общий BFF

- больше согласований
- общие интеграционные точки
- размытые границы
- риск задеть соседа

Модульный BFF

- изолированный контракт
- внутри своей зоны ответственности
- границы строго определены за счёт контрактов
- минимизируем риск задеть соседа
  лишний раз не ходим к соседям
- релизы становятся дешевле





Модульность снижает связность не только в коде, но и между командами

kaspersky

Но бесплатно это не достаётся

TRADE-OFF

Модульный BFF даёт независимость, но добавляет платформенные обязанности

01

Больше инфраструктуры

broker
connectors
registry
healthcheck

02

Контракты и версии

API-схемы
совместимость
правила сборки

03

Совместимость

разные версии модулей
+ одна платформа

04

Наблюдаемость

tracing
по всей цепочке вызова

05

Безопасность

контроль модулей, методов и прав

Модульный BFF требует зрелой платформенной поддержки и дисциплины команд

kaspersky

Но бесплатно это не достаётся

TRADE-OFF

Модульный BFF даёт независимость, но добавляет платформенные обязанности

01

Больше инфраструктуры

broker
connectors
registry
healthcheck

02

Контракты и версии

API-схемы
совместимость
правила сборки

03

Совместимость

разные версии модулей
+ одна платформа

04

Наблюдаемость

tracing
по всей цепочке вызова

05

Безопасность

контроль модулей, методов и прав

Модульный BFF требует зрелой платформенной поддержки и дисциплины команд

kaspersky

Но бесплатно это не достаётся

TRADE-OFF

Модульный BFF даёт независимость, но добавляет платформенные обязанности

01
Больше инфраструктуры

broker
connectors
registry
healthcheck

02
Контракты и версии

API-схемы
совместимость
правила сборки

03
Совместимость

разные версии модулей
+ одна платформа

04
Наблюдаемость

tracing
по всей цепочке вызова

05
Безопасность

контроль модулей, методов и прав

Модульный BFF требует зрелой платформенной поддержки и дисциплины команд

kaspersky

Но бесплатно это не достаётся

TRADE-OFF

Модульный BFF даёт независимость, но добавляет платформенные обязанности

01
Больше инфраструктуры

broker
connectors
registry
healthcheck

02
Контракты и версии

API-схемы
совместимость
правила сборки

03
Совместимость

разные версии модулей
+ одна платформа

04
Наблюдаемость

tracing
по всей цепочке вызова

05
Безопасность

контроль модулей, методов и прав

Модульный BFF требует зрелой платформенной поддержки и дисциплины команд

kaspersky

Но бесплатно это не достаётся

TRADE-OFF

Модульный BFF даёт независимость, но добавляет платформенные обязанности

01
Больше инфраструктуры

broker
connectors
registry
healthcheck

02
Контракты и версии

API-схемы
совместимость
правила сборки

03
Совместимость

разные версии модулей
+ одна платформа

04
Наблюдаемость

tracing
по всей цепочке вызова

05
Безопасность

контроль модулей, методов и прав

Модульный BFF требует зрелой платформенной поддержки и дисциплины команд

kaspersky

Но бесплатно это не достаётся

TRADE-OFF

Модульный BFF даёт независимость, но добавляет платформенные обязанности

01

Больше инфраструктуры

broker
connectors
registry
healthcheck

02

Контракты и версии

API-схемы
совместимость
правила сборки

03

Совместимость

разные версии модулей
+ одна платформа

04

Наблюдаемость

tracing
по всей цепочке вызова

05

Безопасность

контроль модулей, методов и прав

Модульный BFF требует зрелой платформенной поддержки и дисциплины команд

kaspersky

Когда модульный BFF оправдан

Overkill

- маленький продукт
- одна или две команды
- мало интеграций
- нет модульности в UI
- синхронизация между командами дешёвая

Когда модульный BFF оправдан

Overkill

- маленький продукт
- одна или две команды
- мало интеграций
- нет модульности в UI
- синхронизация между командами дешёвая

Оправдано

- реальная модульность в UI
- много независимых команд
- разные темпы развития модулей
- общий BFF тормозит релизы
- on-prem поставки с разными конфигурациями

Когда модульный BFF оправдан

Overkill

- маленький продукт
- одна или две команды
- мало интеграций
- нет модульности в UI
- синхронизация между командами дешёвая

Оправдано

- реальная модульность в UI
- много независимых команд
- разные темпы развития модулей
- общий BFF тормозит релизы
- on-prem поставки с разными конфигурациями

Подход окупается там, где независимость команд уже стала реальной потребностью

Если UI уже плагинный —

рано или поздно возникает вопрос:

почему BFF всё ещё монолитный?

BFF
?

Экспериментируем в модулях, стабилизируем в платформе

Всё общее со временем становится платформой

Архитектура | Enterprise-проблемы | Frontend | Платформы | Путешествия и горы

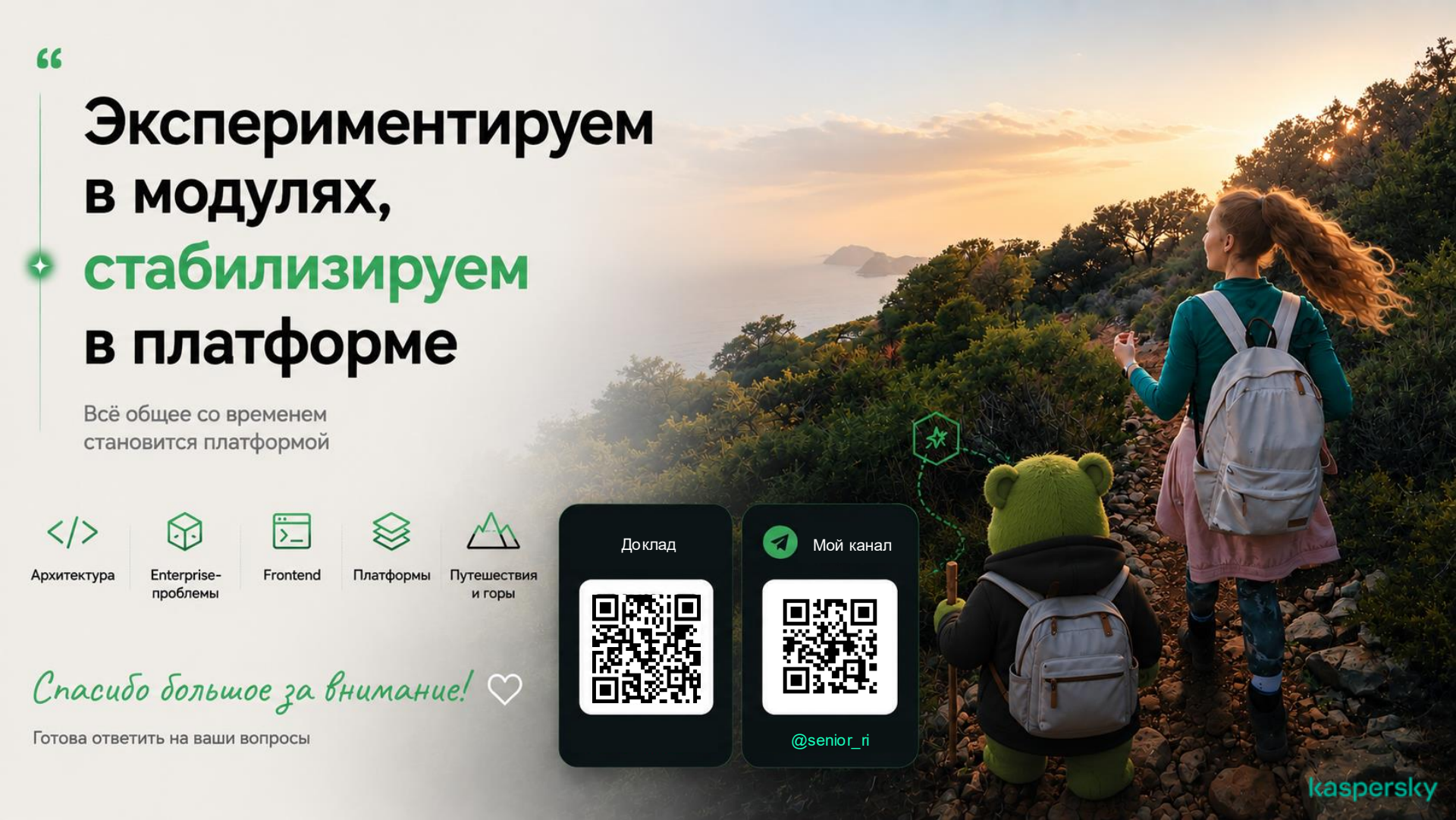
Спасибо большое за внимание!

Готова ответить на ваши вопросы

Доклад

Мой канал

@senior_ri



kaspersky