



**Страх и ненависть: почему мы
пилим монолит без
использования микросервисов**

Войдите или зарегистрируйтесь

Вы сможете отслеживать статус заказа и получать персональные предложения

Вход или регистрация

Корзина 1730-0176-47372

Быстрое добавление: введите наименование или код това...

✓ Выделить все ✕ Удалить выбранные

Поделиться Скачать Распечатать



Конкурс покупок от Makita

код: 5195

Перфоратор Makita HR2470

♡ ✕

139 886 010 ₺

- 9 999 +



Самовывоз [Изменить](#)

м.Авиамоторная, 2-й Кабельный проезд, д. 1к1

Дата

Покупатель [Указать данные](#)

Оплата [Изменить](#)

Оплата по QR-коду

Введите промокод

9999 товаров • 47 595,24 кг 139 886 010 ₺

Итого 139 886 010 ₺

Оформить заказ

Монолит





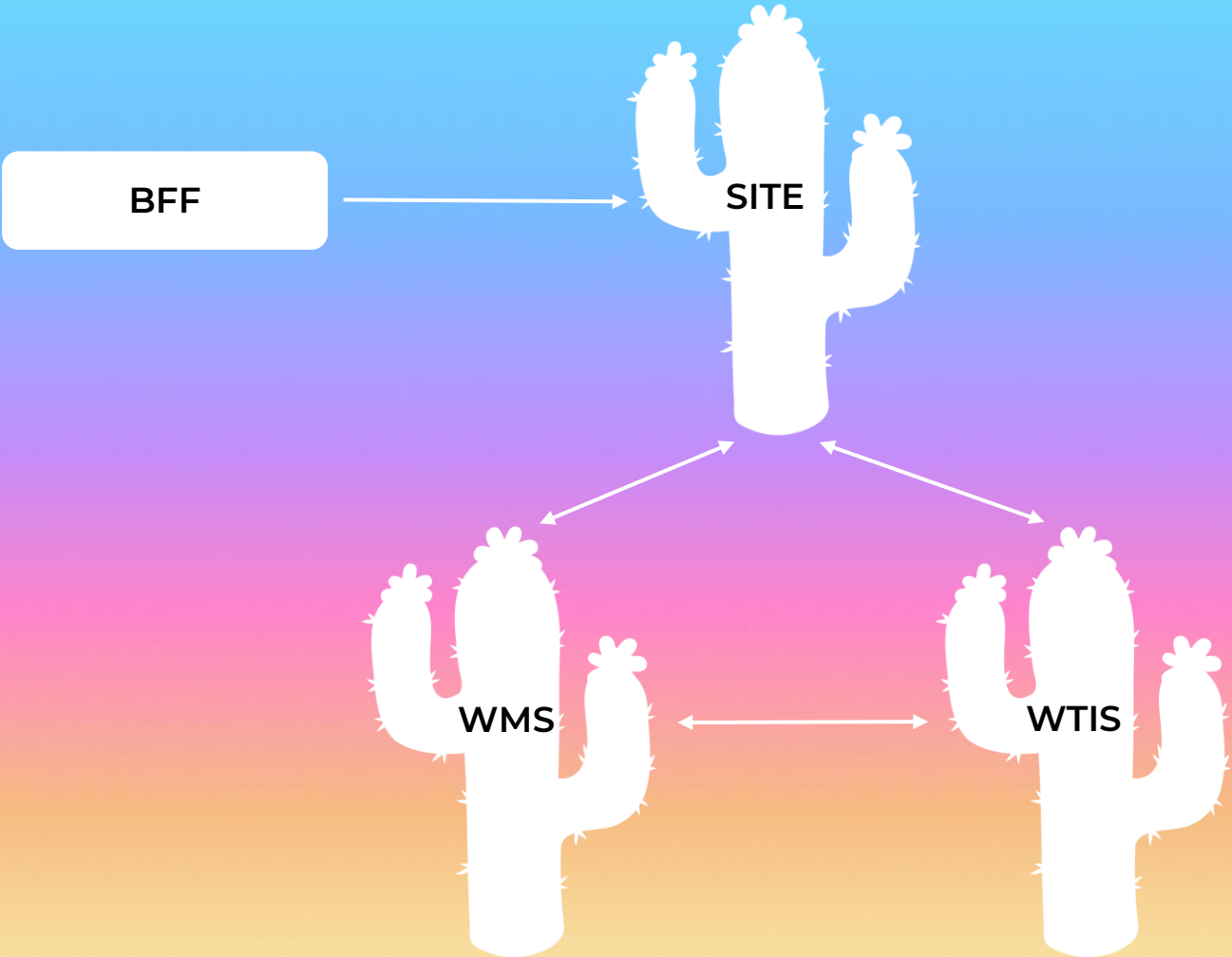


Монолит



Монолит





Почему нам не нравится этот парень?

1. Скорость загрузки страниц не восхищает
2. TTM непредсказуем
3. Код увлекает в черную дыру

1



Очки защитные
209 Р
1 шт. в наличии

Купить

2

Купить



3



4

40 минут

5



Очки защитные
209 Р
1 шт. в наличии

Купить

6

Купить



7



alarm!!!
на складе

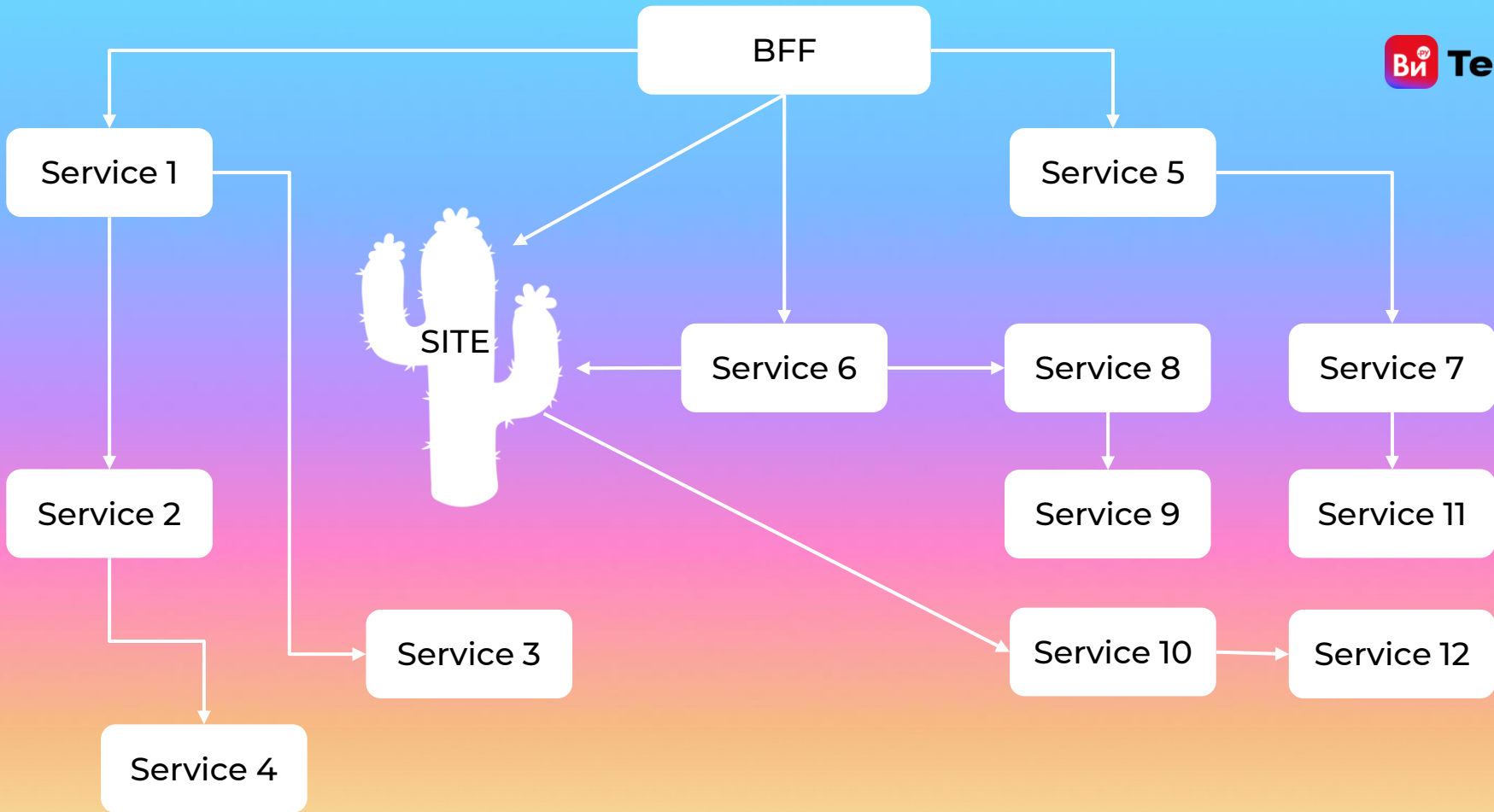
8



Очки защитные
209 Р
0 шт. в наличии

Закончился





Чем хороша команда распила монолита?

1. Сфокусирована только на распиле
2. Защищает продуктовые команды от непродуктивных активностей
3. Сервисы однообразны



У компании одна-единственная
доменная область, а сервисы
лишь проецируют ее модели на
себя

Доменная область

Товар:

- цена
- наличие
- возможность доставки
- поставщик
- аналоги

Сервис 1

Товар:
- цена

Сервис 2

Товар:
- наличие
- поставщик

Сервис 3

Товар:
- наличие
- цена

Сервис 4

Товар:
- наличие
- возможность доставки
- аналоги
- цена





Монолит

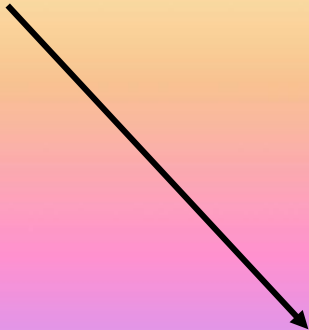
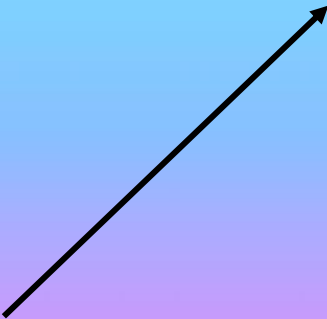
Микросервисы

Монолит



Микросервисы

Сервисы



Монолит

Микросервисы





SOA



Почему нам подошла SOA?

1. Сильно связанный код в монолите, который сложно дробить на микросервисы
2. Сроки создания микросервисов нам не подходят
3. Инфраструктурные и бюрократические ограничения














Чек-листы

- готовность high- и low-level-дизайнов
- согласовать с архитектурным комитетом технологии, которых нет на тех. радаре
- пройти проверку ИБ
- пройти проверку на соблюдение паттернов отказоустойчивости
- развернуть тестовые и интеграционные стенды с CI/CD
- проведение нагрузочного тестирования и привлечение SRE для анализа результатов
- написание инструкций и документации

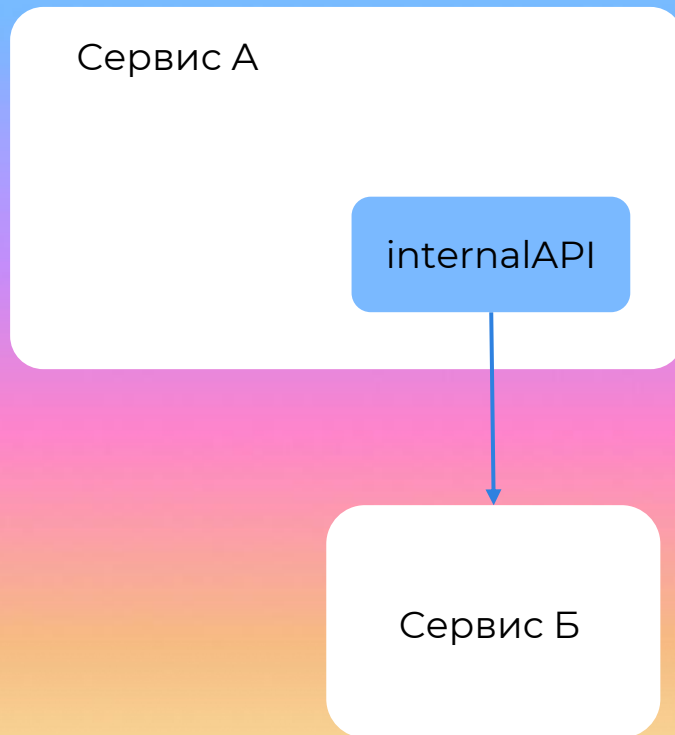
Закон Конвея:
организации проектируют
системы, которые копируют
структуру коммуникаций внутри
самой организации



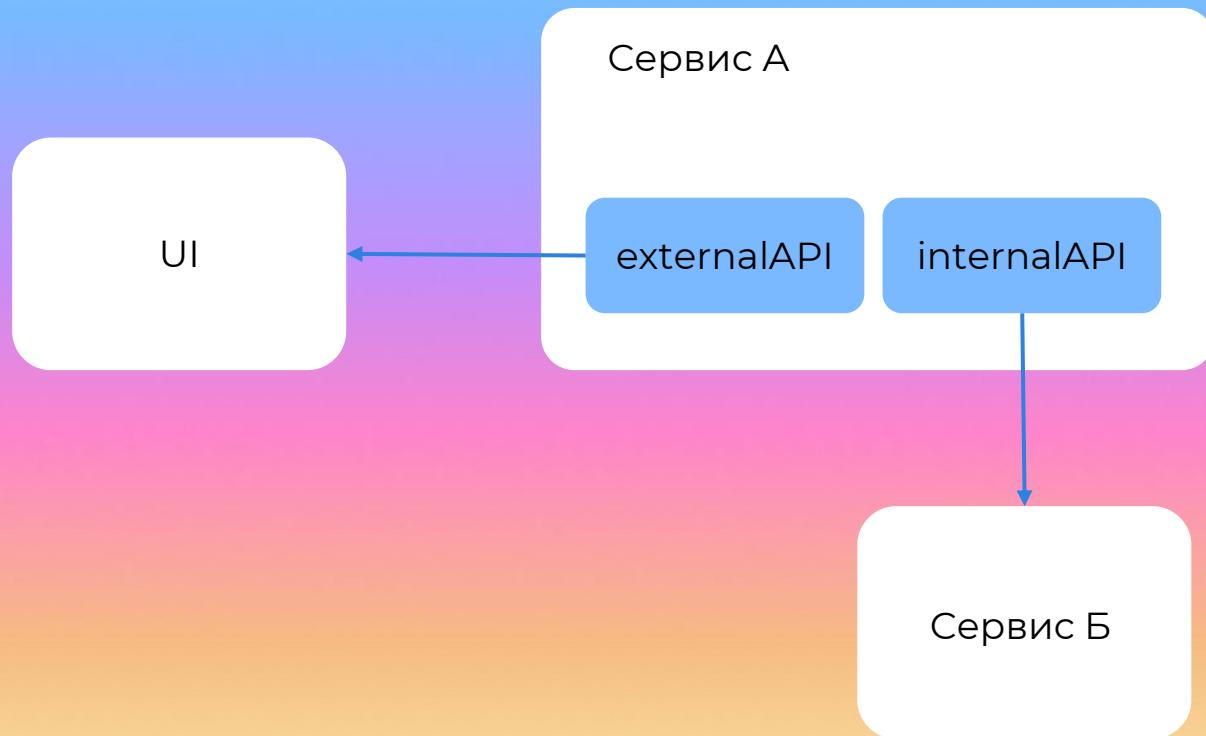
Монорепа

- ▼  cmd
- ▼  app
 -  main.go
- ▼  cli
 -  consume.go
 -  dlq.go
 -  grpc.go
 -  load.go
 -  migrate.go
 -  rest.go
 -  root.go
 -  sync.go
 -  testlock.go

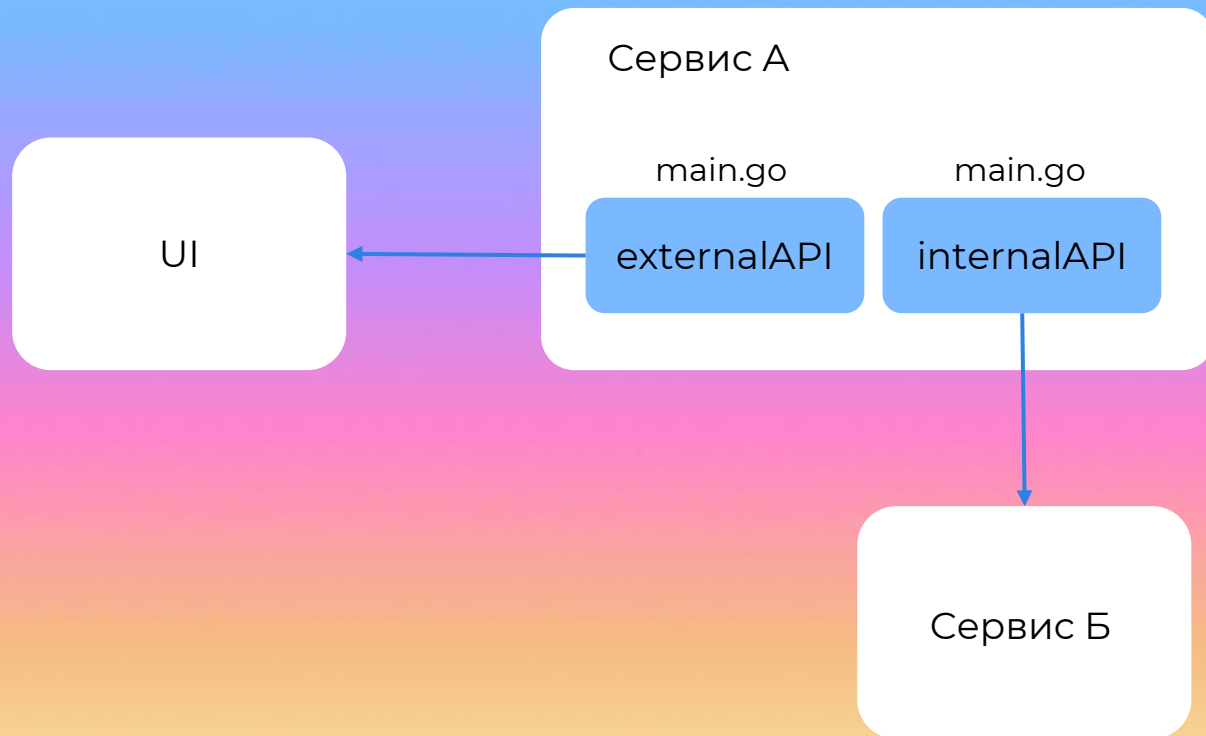
Общение через API



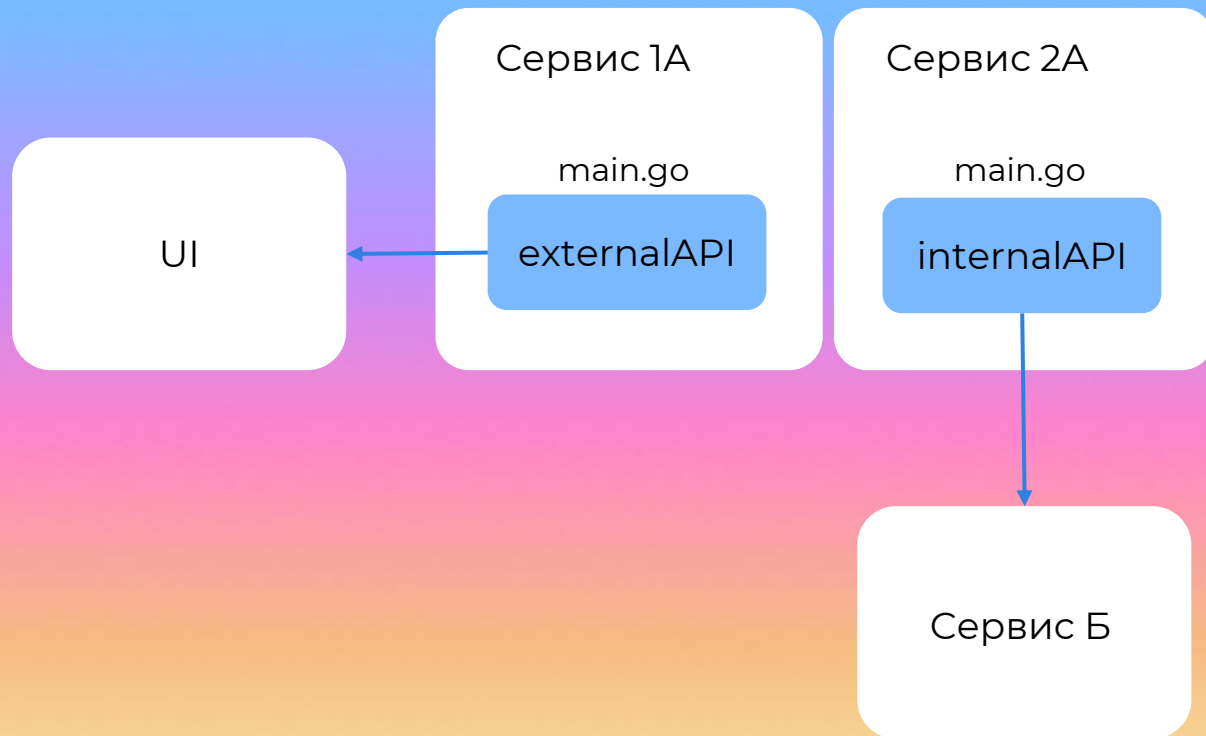
Что, если нужен UI?



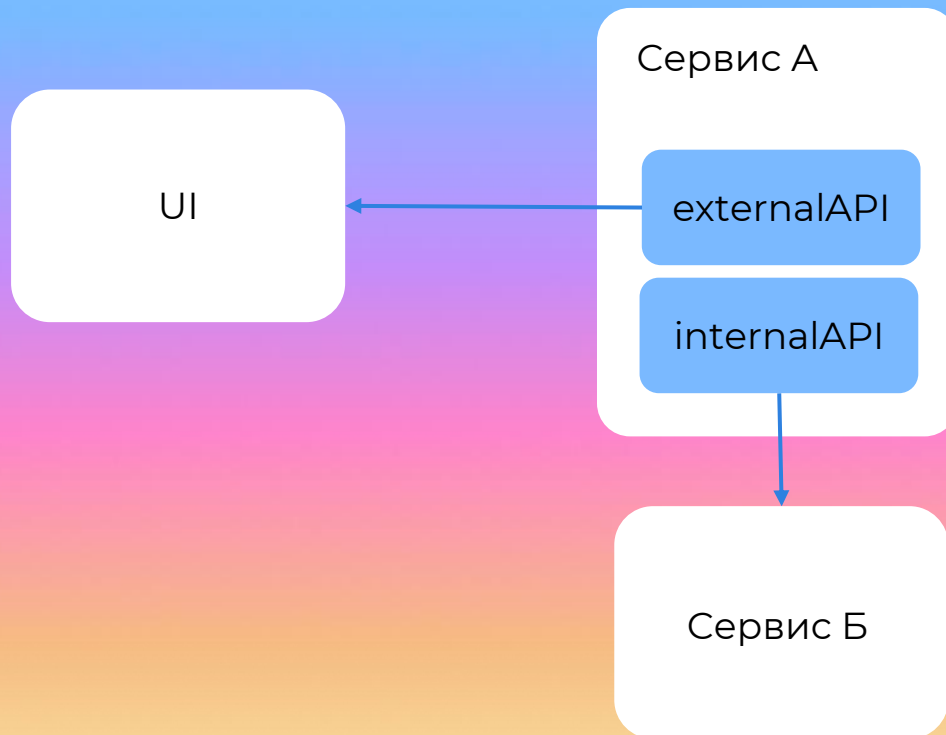
Пара main-файлов



Два отдельных сервиса



И все же монорепа



Запуск через Кобру

```

func Run(ctx context.Context, conf config.Config) error {
    root := &cobra.Command{
        RunE: func(cmd *cobra.Command, args []string) error {
            return cmd.Usage()
        },
    }

    root.AddCommand(
        grpcCmd(ctx, conf),
        migrateCmd(ctx, conf),
        consumeCmd(ctx, conf),
        restCmd(ctx, conf),
    )

    return errors.Wrap(root.ExecuteContext(ctx), "run application")
}

```

Команды запуска

```
func grpcCmd(ctx context.Context, conf config.Config) *cobra.Command {
    return &cobra.Command{
        Use:     "grpc",
        Short:  "start grpc server listening",
        RunE:   func(cmd *cobra.Command, args []string) error {...}
    }
}
```

```
func migrateCmd(ctx context.Context, conf config.Config) *cobra.Command {
    command := &cobra.Command{
        Use:     "migrate",
        Short:  "run db migrations",
        ValidArgs: []string{"postgres"},
        RunE:   func(cmd *cobra.Command, args []string) error {...}
    }
}
```

Преимущества распила на сервисы

1. Системный подход к процессу распила
2. Избегаем части сетевых вызовов
3. Не ограничиваемся в подходах
4. Имеем универсальный пайплайн

Недостатки распила на сервисы

1. Замоноличивание сервисов
2. Усложнение процессов и коммуникации










api	Replace broken link in api folder examples.	4 years ago
assets	chore: remove unnecessary .keep files	5 years ago
build	build config file location clarification	5 years ago
cmd	Update cmd/README w/updated example URLs	2 years ago
configs	fix a word mistake in writing	4 years ago
deployments	chore: remove unnecessary .keep files	5 years ago
docs	Dapr examples for cmd, docs, pkg and tools	5 years ago
examples	remove dead link	2 years ago
githooks	chore: remove unnecessary .keep files	5 years ago
init	chore: remove unnecessary .keep files	5 years ago
internal	Update README.md	10 months ago
pkg	Update README.md	10 months ago
scripts	Add missing newlines	5 months ago
test	chore: remove unnecessary .keep files	5 years ago
third_party	chore: remove unnecessary .keep files	5 years ago
tools	Dapr examples for cmd, docs, pkg and tools	5 years ago
vendor	Convert github.com/golang/go/wiki links to go.dev/wiki	5 months ago
web	more notes and notes in major directories	6 years ago
website	Capitalize the H in GitHub	4 years ago

Платформа

Гошники

api	Replace broken link in api folder examples	4 years ago
assets	chore: remove unnecessary .keep files	5 years ago
build	build config file location clarification	5 years ago
cmd	Update cmd/README w/updated example URLs	2 years ago
configs	fix a word mistake in writing	4 years ago
deployments	chore: remove unnecessary .keep files	5 years ago
docs	Docs examples for cmd, docs, pkg and tools	5 years ago
examples	remove dead link	2 years ago
githubko	chore: remove unnecessary .keep files	5 years ago
int	chore: remove unnecessary .keep files	5 years ago
internal	Update README.md	10 months ago
pkg	Update README.md	10 months ago
scripts	Add missing newlines	5 months ago
test	chore: remove unnecessary .keep files	5 years ago
third_party	chore: remove unnecessary .keep files	5 years ago
tools	Docs examples for cmd, docs, pkg and tools	5 years ago
vendor	Convert github.com/golang/gwiki links to go.dev/wiki	5 months ago
web	more notes and notes in major directories	6 years ago
website	Capitalize the H in GitHub	4 years ago

Сервис корзины и оформления

- ▼  internal
 - >  api
 - >  build
 - >  cart
 - >  config
 - >  infrastructure
 - >  migrations
 - >  mocks
 - >  order

Модуль корзины

- ▼ cart
 - ▼ api
 - > cartcontract
 - > grpc
 - ▼ broker
 - ▼ kafka
 - > consumers
 - > producers
 - ▼ informers
 - > cartinformers
 - > informersservice
 - > iteminformers
 - 👤 cartstateinfo.go
 - > metrics
 - > migrations
 - > models
 - > mutator
- ▼ redis
 - > consumer
 - > sync
- ▼ repository
 - > postgres
 - 👤 errors.go
- ▼ rest
 - > middlewares
 - > restapi
- ▼ services
 - > externalservices
 - > innerservices
 - > models
- > uuids

Сеньор

Сеньор



Чистая архитектура
решает наши проблемы

Модуль оформления

- ▼ folder order
 - ▼ folder api
 - > folder broker
 - > folder grpc
 - ▼ folder app
 - > folder services
 - > folder usecase
 - > folder valueobject
 - ▼ folder domain
 - > folder constants
 - > folder entity
 - > folder enum
 - > folder repository
 - > folder valueobject
 - ▼ folder infrastructure
 - > folder cartapi
 - > folder mutex
 - > folder pgsq
 - > folder services





God-object?

```
type Builder struct {  
    config config.Config  
  
    clients map[string]*http.Client  
  
    hamletClient      hamlet_api.AvailabilityClient  
    productInfoClient productinfo.ProductInfoServiceClient  
    fryClient         frysdk.FryClient  
    itcClient         itcsdk.Client  
    frodoClient       frodosdk.FrodoClient  
  
    shutdown      shutdown  
    healthcheck   healthcheck  
}
```

```

type Builder struct {
    config                config.Config
    grpcServer            *grpc.Server
    redis                 *redis7.Client
    redisBridge           *sync.Bridge
    pgConnectionMaster    *bun.DB
    pgConnectionReplica  *bun.DB
    shutdown              shutdown
    healthcheck           healthcheck
    mutexSvc              *mutex.Service
    cartUIDSvc            *cartuid.Service
    lotItemSvc            *lotitem.Service
    cartPermissionsSvc    *cartpermissions.Checker
    cartNumberGenerator  *cartnumber.Generator
    cartFinderSvc         *cartfinder.Service
    cartGetterSvc         *cartgetter.Service
    cartGetterSvcWithSync *cartsync.CartGetterDecorator
    pricerSvc             *priceservice.Service
    itemAvailabilitySvc   *itemavailability.Service
    itemMixerSvc          *itemmixer.Mixer
    promoCodeSvc          *promocode.Service
    itemsGetterSvc        *itemsgetter.Service
    cartSyncSvc           *cartsync.CartSync
    ownerRepo             *pgrepo.CartOwnerRepo
    cartItemRepo          *pgrepo.CartItemRepo
    cartRepo              *pgrepo.CartRepo
    cartNumberSequence   *pgrepo.CartNumberSequence
    scroogeExporterCli    *scrooge.ExporterClient
    scroogeExternalCli   scroogeExternalCli
    officerCli            officerCli
    cartLoyalCli          cartLoyalCli
    orderLoyalCli         orderLoyalCli
    fiasCache             fiasCache
    aregCli              aregCli
    druzCli              druzCli
    analyticSvc          analyticSvc
    fryCli               fryCli
    jwtService            jwtService
    promoService          promoService
    cartProductInfoSvc   cartProductInfoSvc
    orderProductInfoSvc  orderProductInfoSvc
    cartHamletSvc        cartHamletSvc
    orderHamletSvc       orderHamletSvc
    froxySvc              froxySvc
    siteDeliverySvc       siteDeliverySvc
    geocoderSvc           geocoderSvc
    geolocatorSvc        geolocatorSvc
    requiredAddressSvc   requiredAddressSvc
    addressGetter         addressGetter
    cartAPI               cartAPI
    orderCartAPIService  orderCartAPIService
    prometheusRegistry   prometheusRegistry
    *scroogesvc.ExternalClient
    *officer.Client
    *loyal.Service
    *orderloyal.Service
    *fiasgetter.FiasCache
    *areg.Service
    *druz.Client
    *analytics.UseCase
    *fry.Client
    *jwt.Service
    *promo.Service
    *cartproductinfo.Service
    *orderproductinfo.Service
    *carthamlet.Service
    *orderhamlet.Service
    *froxy.Service
    *sitedelivery.Service
    *geocoder.Chained
    *dadatageolocator.Adapter
    *requiredaddress.Service
    *getaddress.Usecase
    *cartcontract.API
    *ordercartapi.Service
    *prometheus.Registry
}

```

Экспорт функциональности

```
func (b *Builder) RedisConsumer() (*redisconsumer.WithGroupDestruction, error) {}
```

```
func (b *Builder) GRPCServer(ctx context.Context) (*grpc.Server, error) {}
```

```
func (b *Builder) HTTPServer(ctx context.Context) (*http.Server, error) {}
```

```
func (b *Builder) NewMonolithClient() *monolith.Client {}
```


Shutdown c builder

```

func grpcCmd(ctx context.Context, conf config.Config) *cobra.Command {
    return &cobra.Command{
        Use: "grpc",
        Short: "start grpc server listening",
        RunE: func(cmd *cobra.Command, args []string) error {
            builder := build.New(ctx, conf)
            ctx, cancel := context.WithCancel(ctx)
            defer cancel()

            go func() {
                builder.WaitShutdown(ctx) // Здесь обрабатываем шатдаун
            }()

            ...
        }
    }
}

```

Shutdown c builder

```

func dlqHandleCmd(ctx context.Context, cfg config.Config) *cobra.Command {
    cmd := &cobra.Command{
        Use: "handle",
        Short: "Re-handles selected messages",
        RunE: func(cmd *cobra.Command, args []string) error {
            ctxWithCancel, cancel := context.WithCancel(ctx)
            defer cancel()

            builder := build.New(ctxWithCancel, cfg)
            defer builder.Shutdown(ctxWithCancel) // И здесь обрабатываем шатдаун
        }
    }
}

```

Структура shutdown

```
type shutdownFn func(context.Context) error
```

```
type shutdown struct {  
    fn []shutdownFn  
}
```

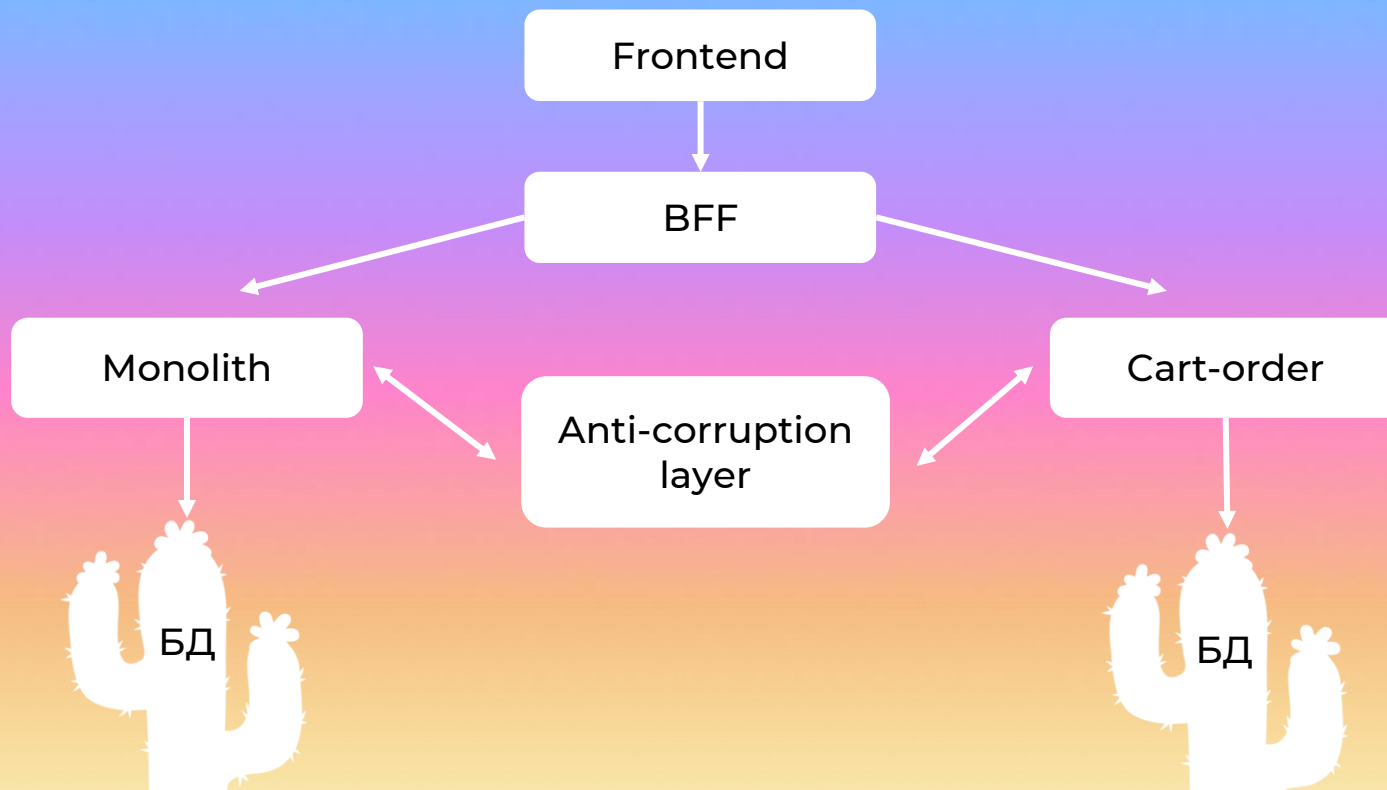
```
func (s *shutdown) add(fn shutdownFn) {  
    s.fn = append(s.fn, fn)  
}
```

```
func (s *shutdown) do(ctx context.Context) {  
    for i := len(s.fn) - 1; i >= 0; i-- {  
        if err := s.fn[i](ctx); err != nil {  
            zerolog.Ctx(ctx).Err(err).Send()  
        }  
    }  
}
```





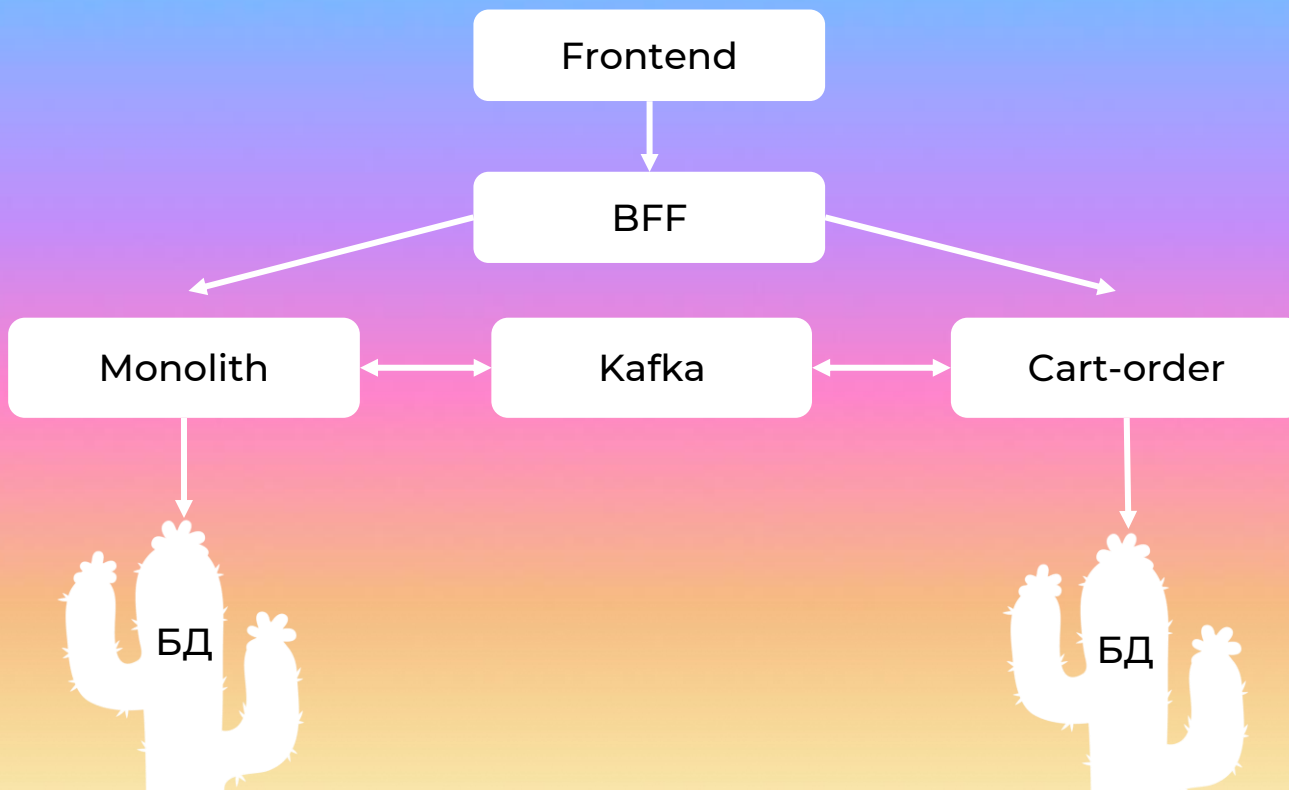
C anti-corruption layer



Почему не anti-corruption layer?

1. Дополнительная служба в нагрузку к остальным
2. Нет уверенности в консистентности данных

Kafka решает проблемы



Выводы об архитектуре

1. Сначала определите доменную область
2. Любое уменьшение связанности кода — благо
3. Ключевой критерий успеха — решение софтовых задач

Ориентируйтесь на
потребности бизнеса



YOU ARE NOW
LEAVING
FEAR & LOATHING



POP: 0

FOUNDED
1971

Микросервисы





CERTIFIED BY NO. 35996



MOTION PICTURE ASSOCIATION OF AMERICA



THIS PICTURE MADE UNDER
THE JURISDICTION OF



AFFILIATED WITH
A.F.L. - C.I.O. - C.L.C.

Вопрос-ответ

Презентация:

<https://bit.ly/FearInVi>

