От монолита к микросервисам: история и практика

Райффайзен

Обо мне

Константин Густов

архитектор,
Райффайзенбанк

10+ лет опыта в разработке
konst.gustov@gmail.com

Темы

I. Причины перехода от монолита к микросервисам

Темы

I. Причины перехода от монолита к микросервисам
II. Что мешало переходу

Темы

I. Причины перехода от монолита к микросервисам
II. Что мешало переходу
III. Наш переход от монолита к микросервисам

Темы

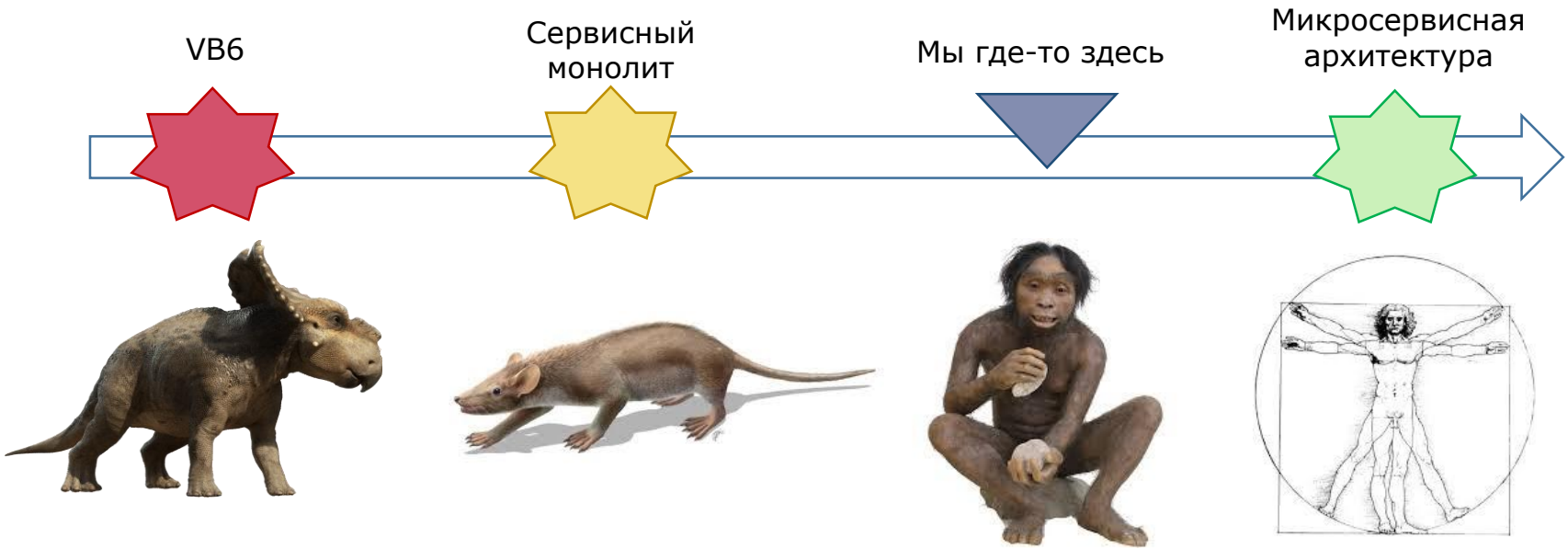
I. Причины перехода от монолита к микросервисам
II. Что мешало переходу
III. Наш переход от монолита к микросервисам
IV. Чего мы добились?

История

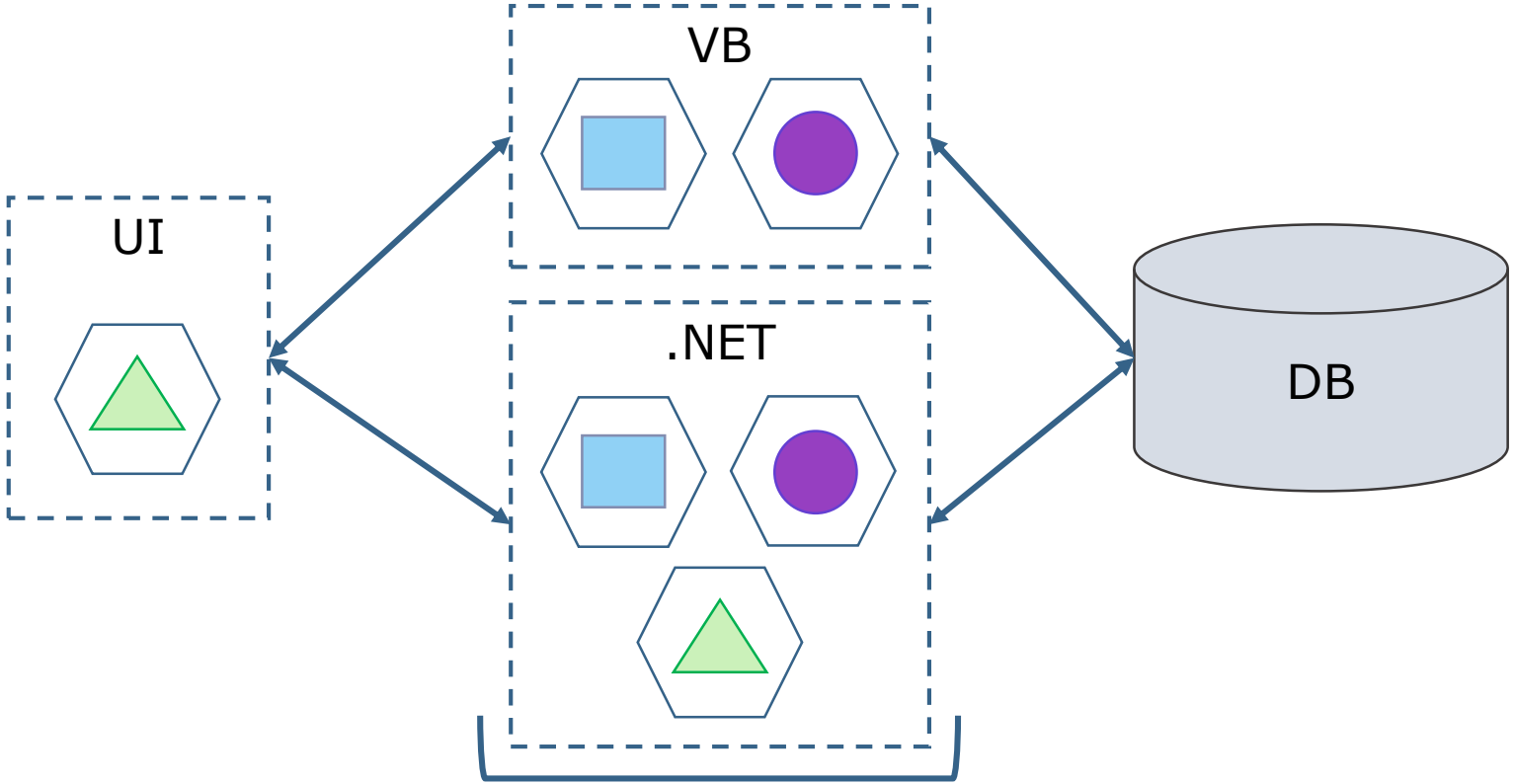
VB6

История

История

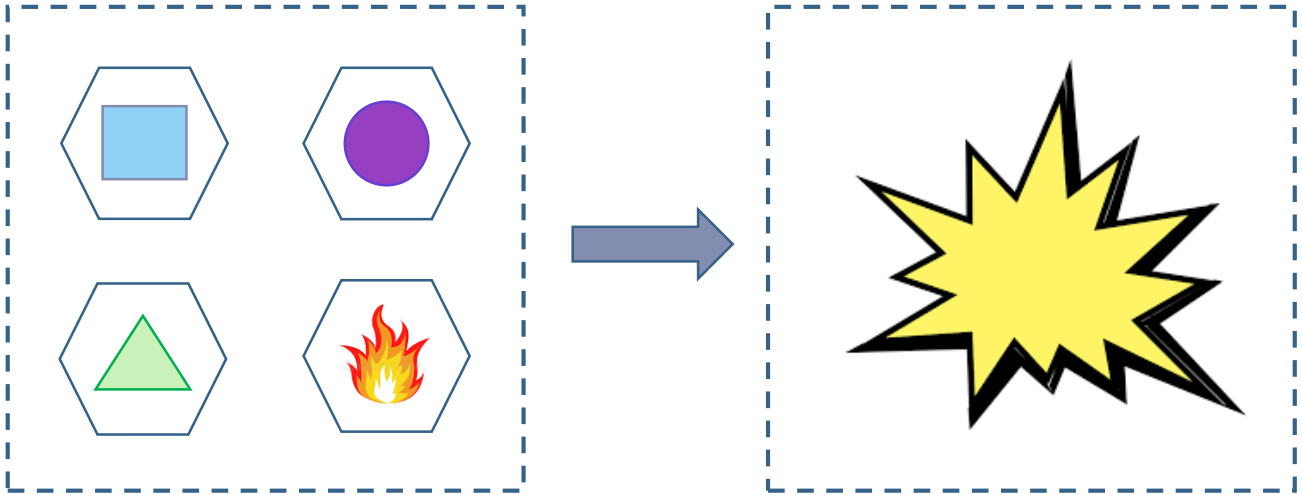

VB6
Сервисный монолит
Мы где-то здесь
Микросервисная архитектура

Архитектура. Начало



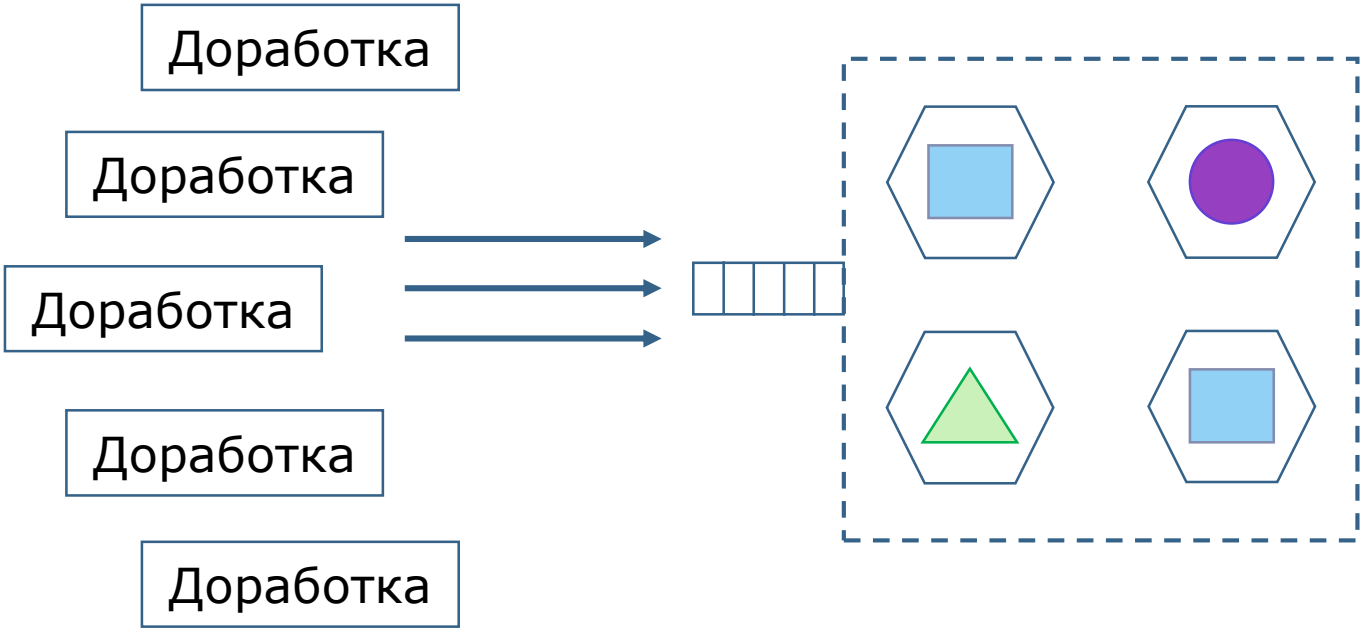
Недостатки решения

Единая точка отказа



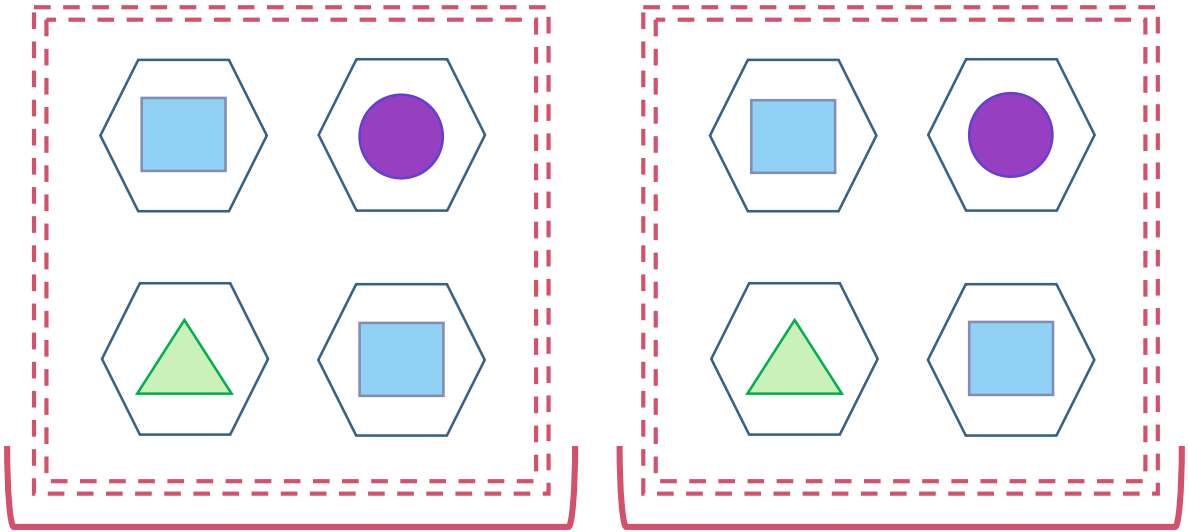
Недостатки решения

Очередь доработок



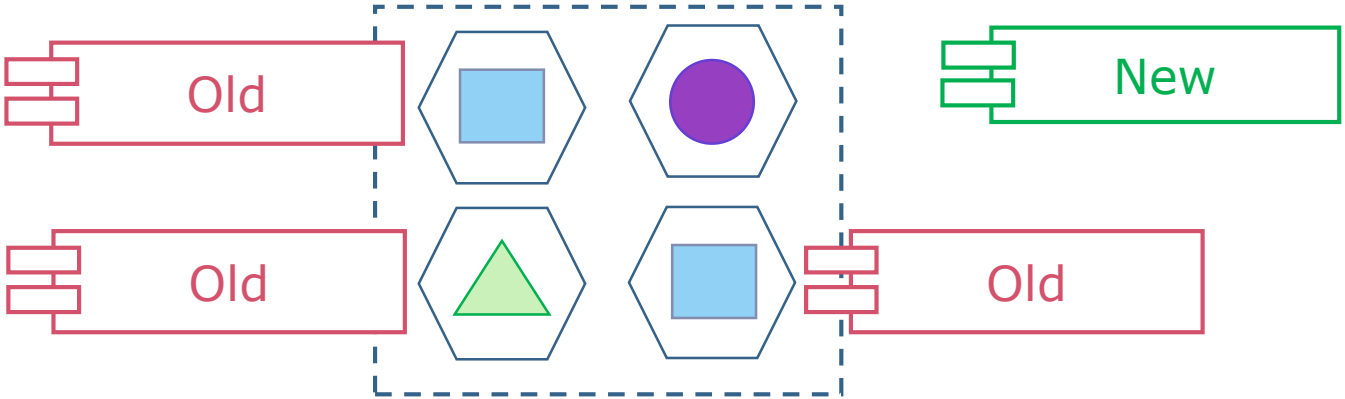
Недостатки решения

Неоптимальное использование ресурсов



Недостатки решения

Трудно внедрять современные технологии

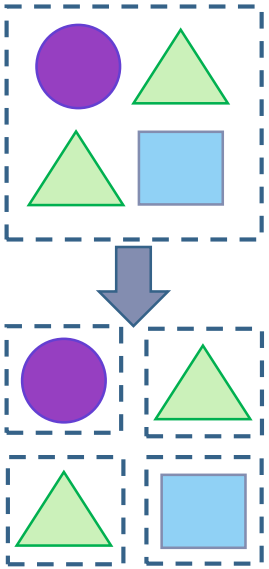


Недостатки решения

Сложность выдачи изменений

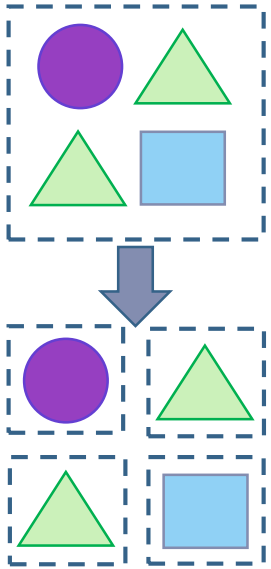
Ожидания от микросервисов

- Выдача компонентов по готовности



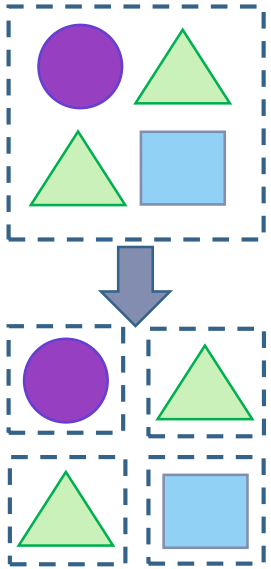
Ожидания от микросервисов

- Выдача компонентов по готовности
- Небольшие продуктовые команды



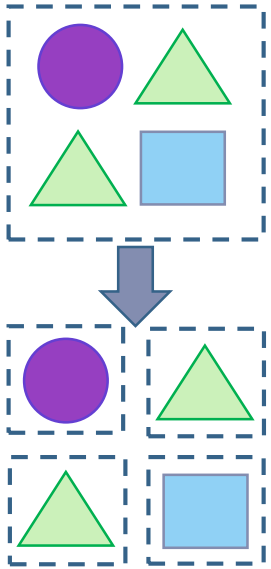
Ожидания от микросервисов

- Выдача компонентов по готовности
- Небольшие продуктовые команды
- Изоляция сервисов в отдельных процессах



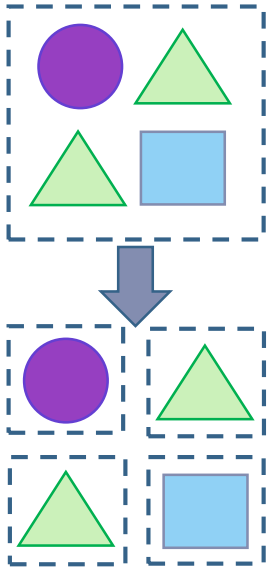
Ожидания от микросервисов

- Выдача компонентов по готовности
- Небольшие продуктовые команды
- Изоляция сервисов в отдельных процессах
- Гибкость развертывания



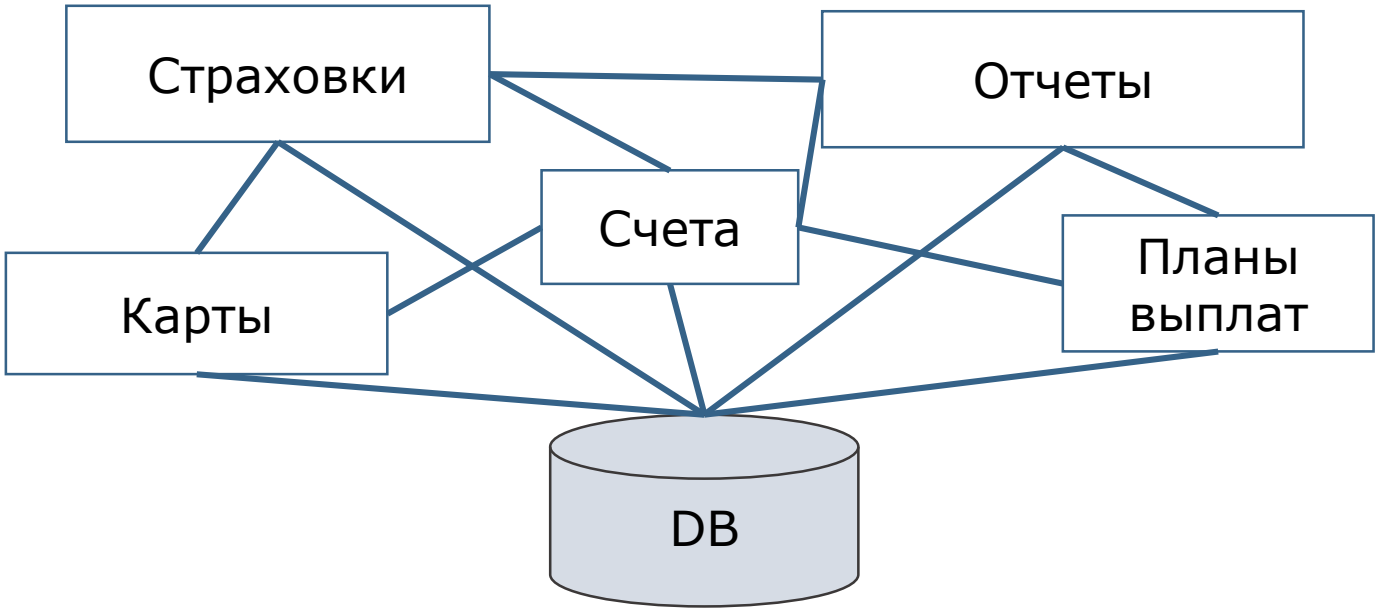
Ожидания от микросервисов

- Выдача компонентов по готовности
- Небольшие продуктовые команды
- Изоляция сервисов в отдельных процессах
- Гибкость развертывания
- Использование новых технологий



Проблема

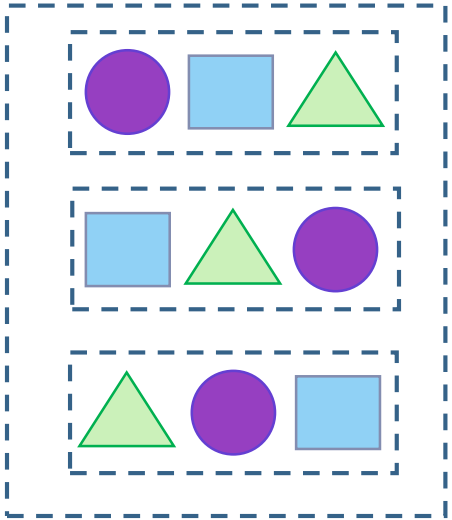
Связи между компонентами



Проблема

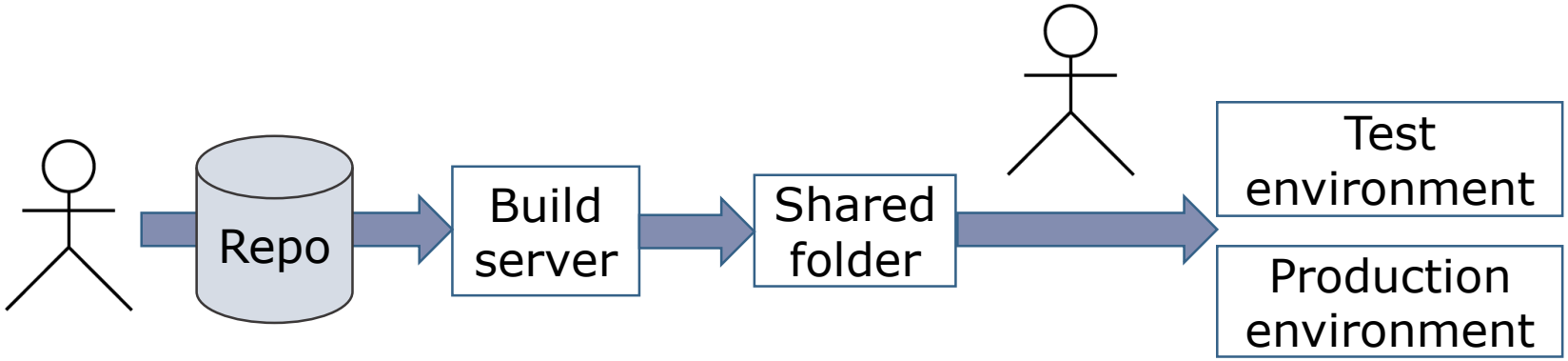
Размер решения:

- 500+ проектов
- 700+ тыс. строк кода



Проблема

Отсутствие необходимой инфраструктуры



Как перейти от монолита к микросервисам

1. Выделение микросервисов

Как перейти от монолита к микросервисам

1. Выделение микросервисов
2. Работа с БД

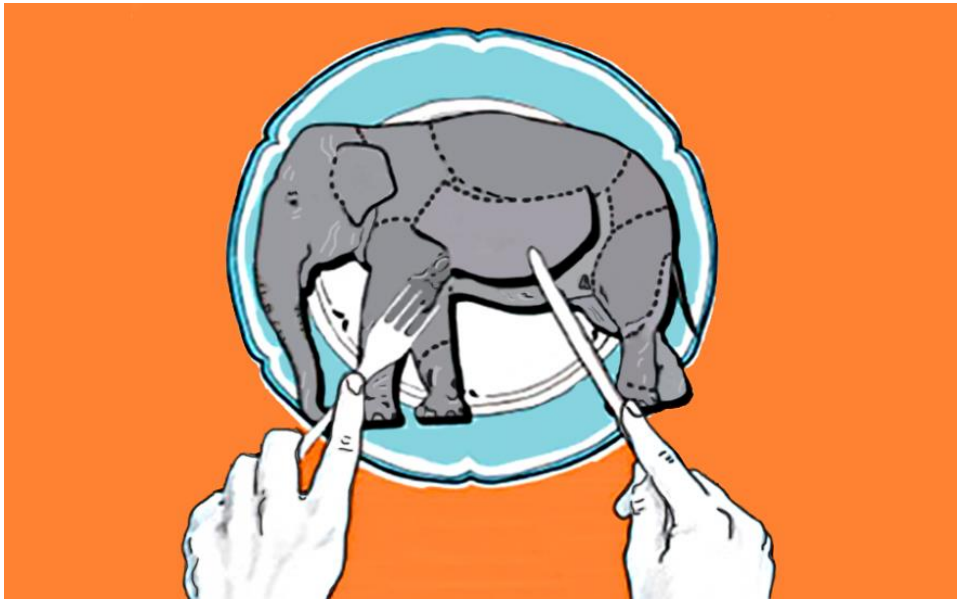
Как перейти от монолита к микросервисам

1. Выделение микросервисов
2. Работа с БД
3. Работа с исходным кодом

Как перейти от монолита к микросервисам

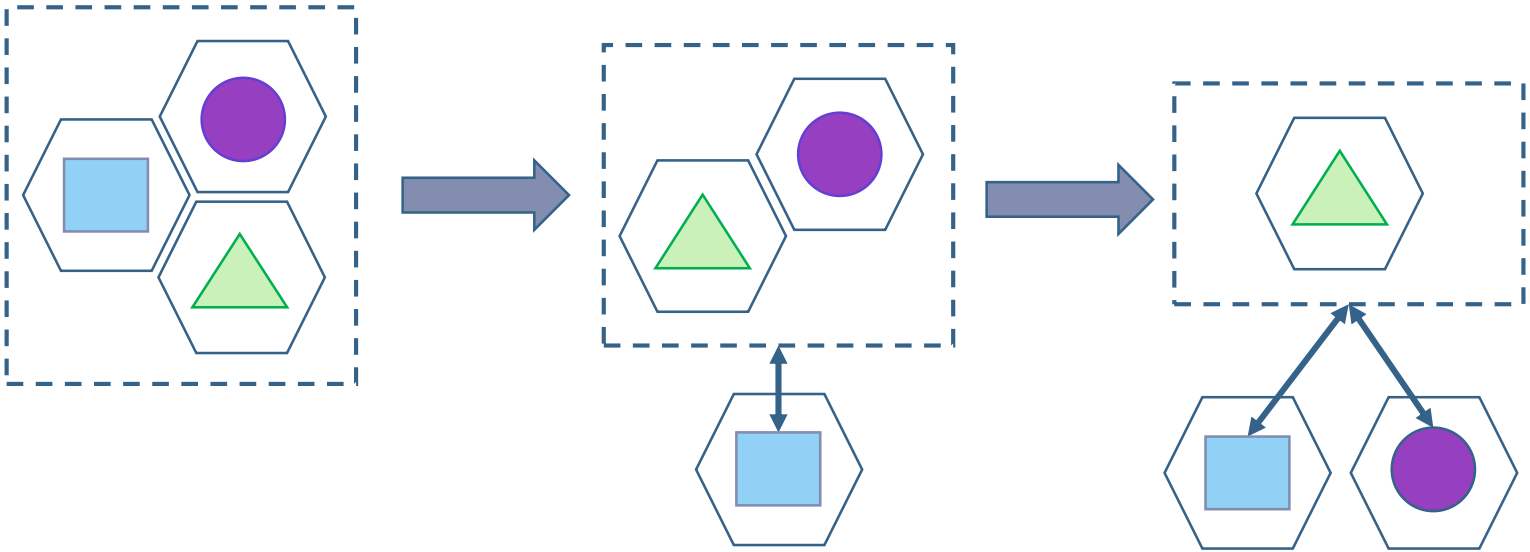
1. Выделение микросервисов
2. Работа с БД
3. Работа с исходным кодом
4. Проблемы инфраструктуры

Выделение микросервисов



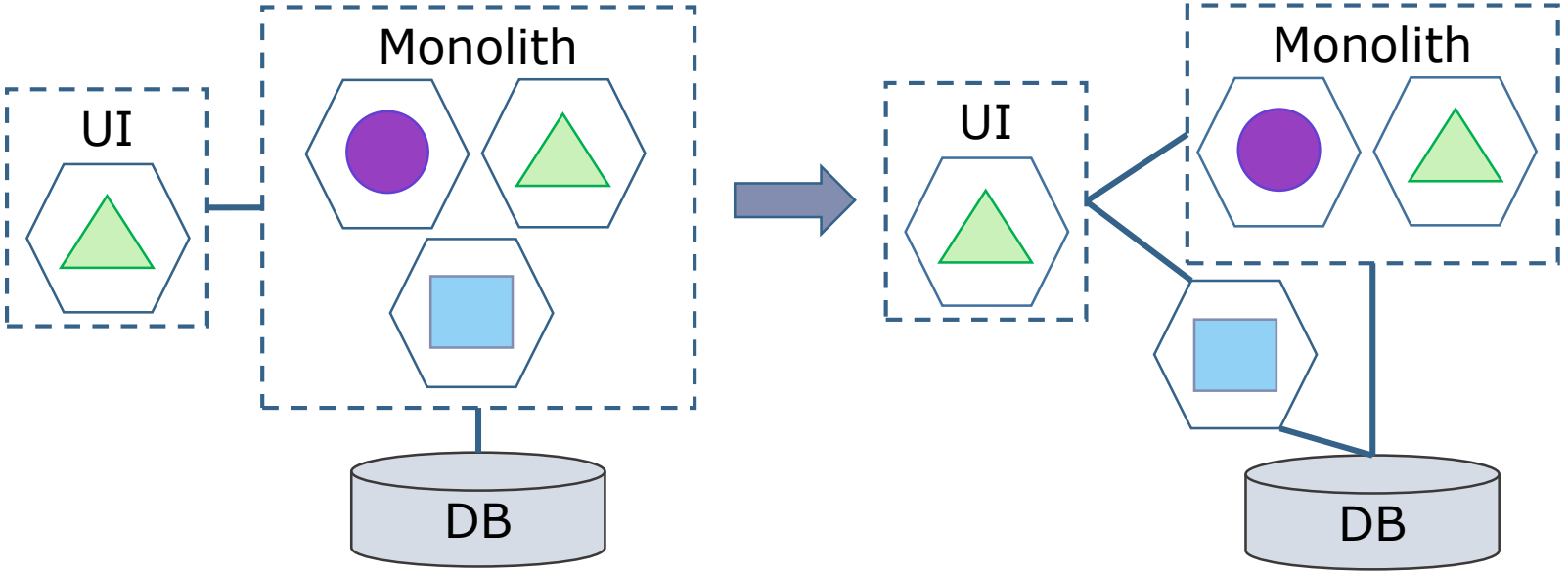
Выделение микросервиса

Разделение монолита – итерационный процесс

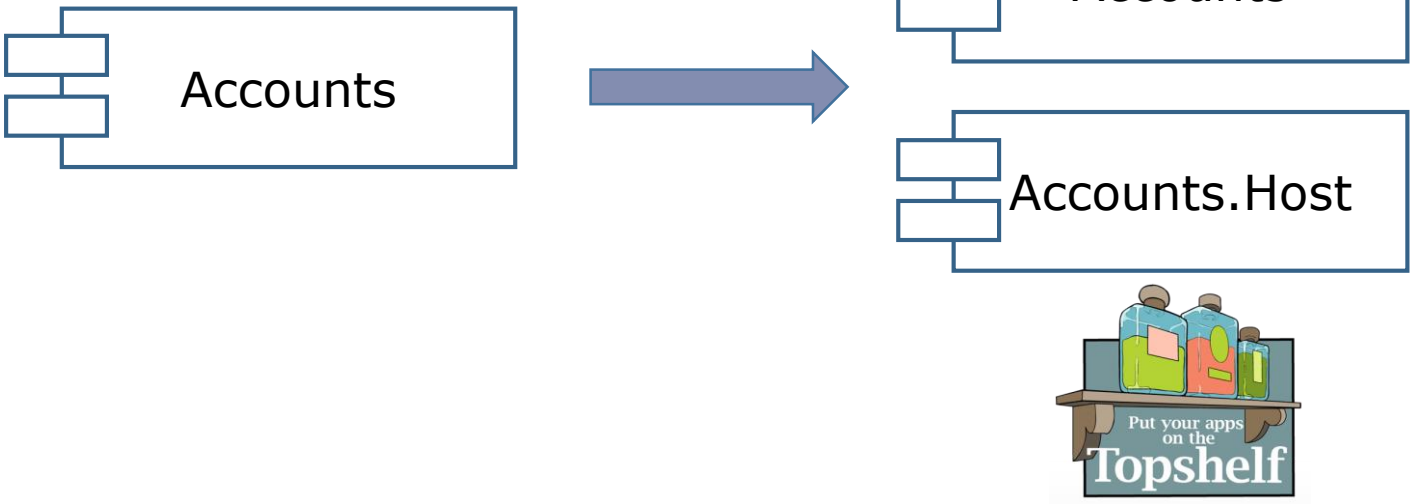


Способы выделения

Выносить существующие модули как сервисы



Сборка сервиса

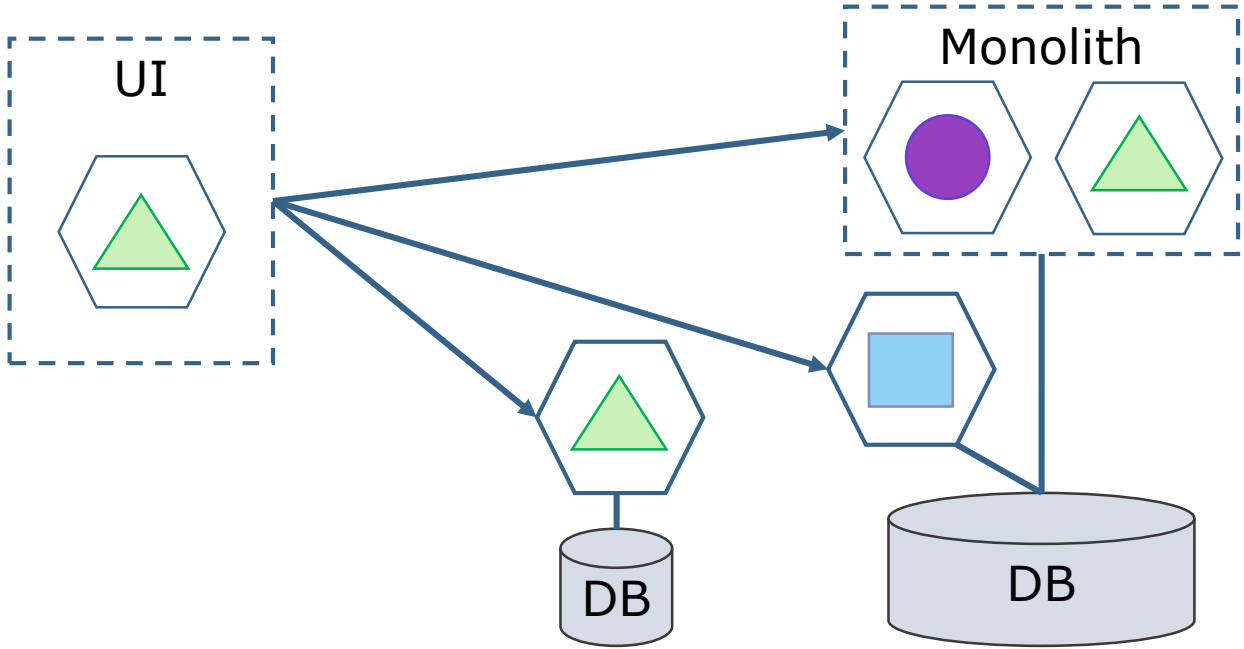


Запуск сервиса

```
namespace RBA.Services.Accounts.Host
{
  internal class Program
  {
    private static void Main(string[] args)
    {
      HostRunner<Accounts>.Run("RBA.Services.Accounts.Host");
    }
  }
}
```

Способы выделения

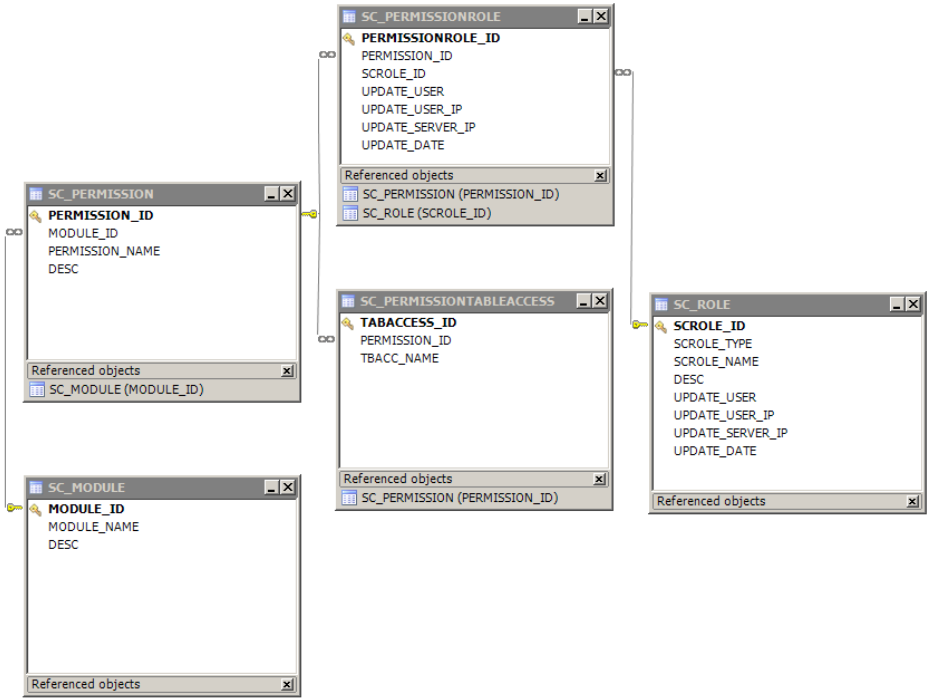
Для решения новых задач создавать микросервисы



Сервис авторизации

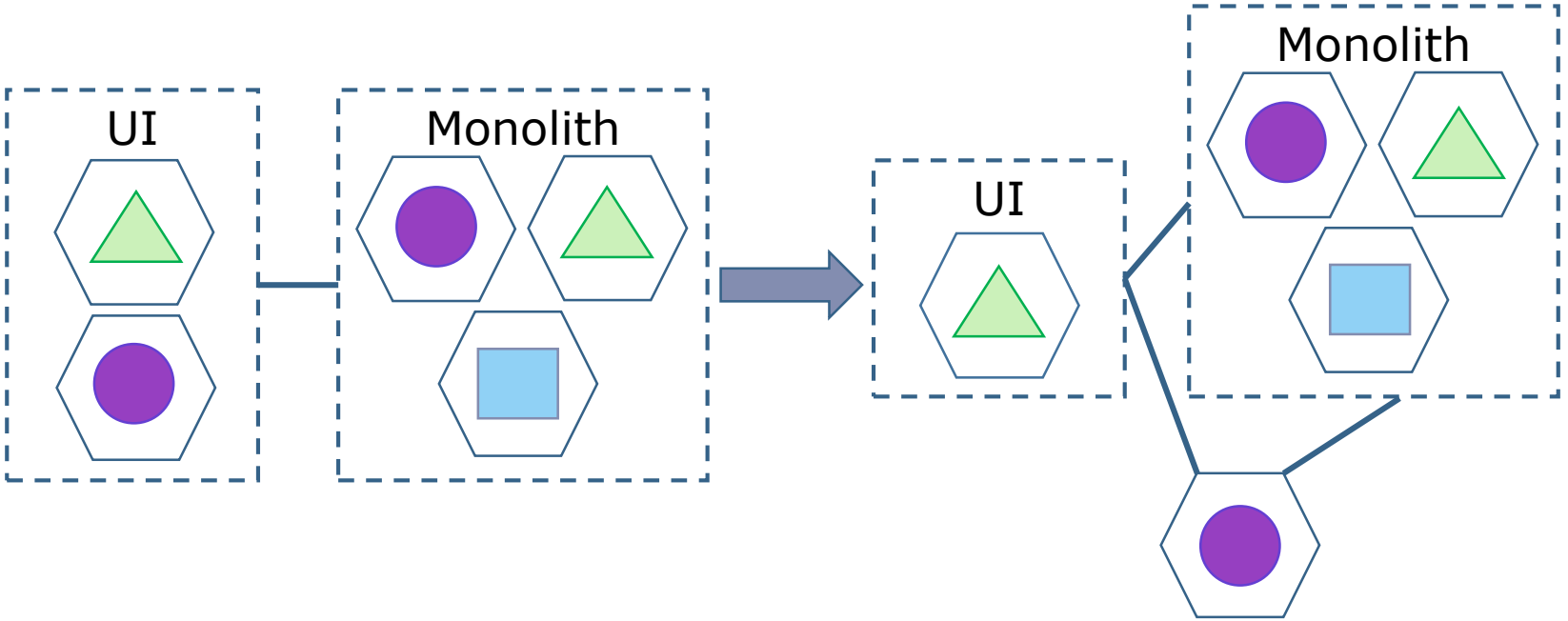
Security

GET	/api/users/current/permissions
GET	/api/users/{login}/permissions
GET	/api/users/current/profile
GET	/api/users/{login}/profile
GET	/api/users/{login}/full-profile
GET	/api/users/current/full-profile



Способы выделения

Выносить бизнес-логику с UI



Пример бизнес-логики на UI

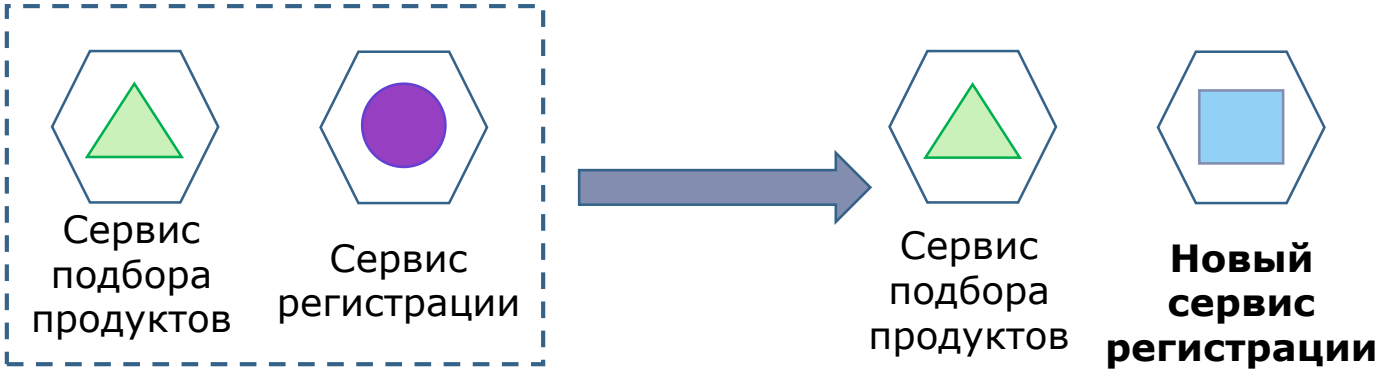
```
var accounts = _service.GetClientAccounts(cnum.ToInt(), (int)accType)
                                .AsEnumerable();
if (cardProduct != null && cardProduct.CardType == CardTypes.CreditCard)
{
  var cardProduct =_dictionaries.GetPrimeProfile(cardProduct.PrimeProfileAccount);
  accounts = accounts.Where(a => _dictionaries
                                   .GetPrimeProfile(a.D_CC_INTEREST_ID)
                                   .PRIME_TEMPLATE_CODE == cardProduct.Code);
}

var cards = _service.GetAllClientCards(cnum);

foreach (var account in accounts)
  currentAccount.Add(GetAccountDto(account, cards));
```

Business logic

UI logic

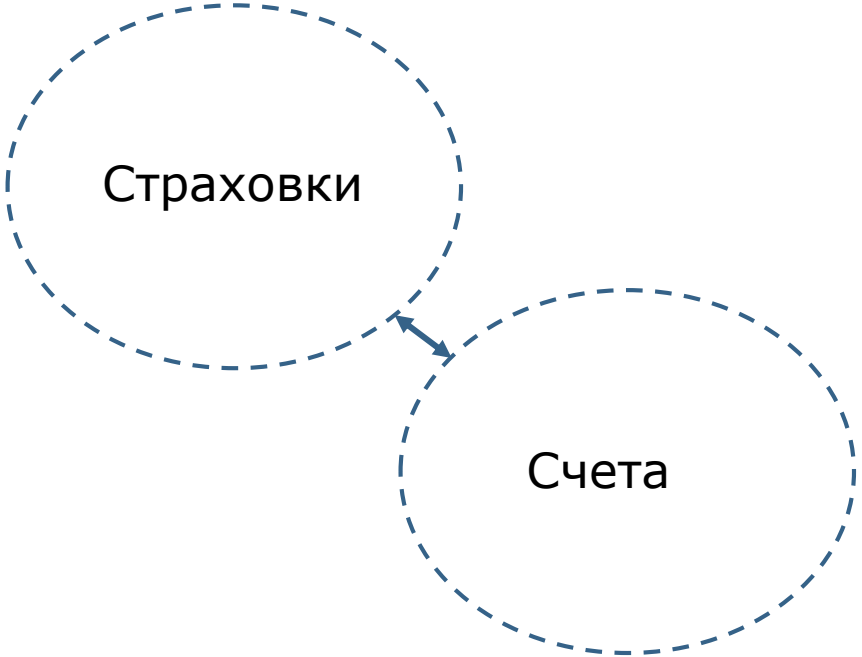
Способы выделения

Выносить существующие функции с рефакторингом

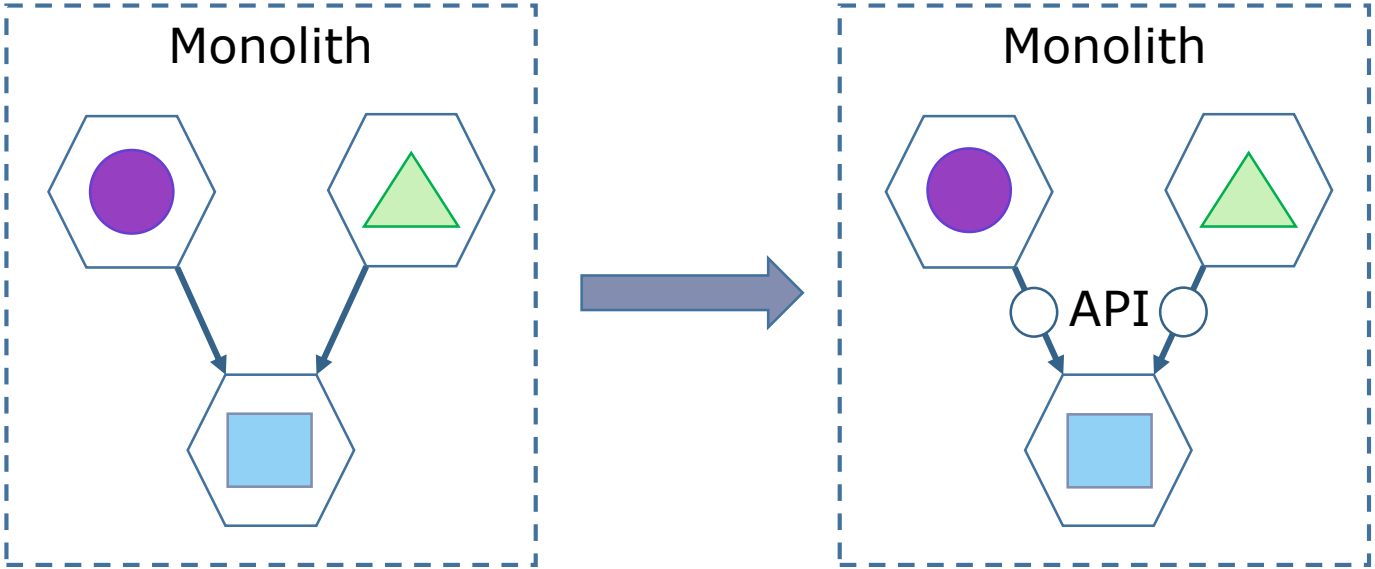


Выделение микросервиса

Для существующих функций определяются ограниченные контексты



Создание anticorruption layer



Создание anticorruption layer

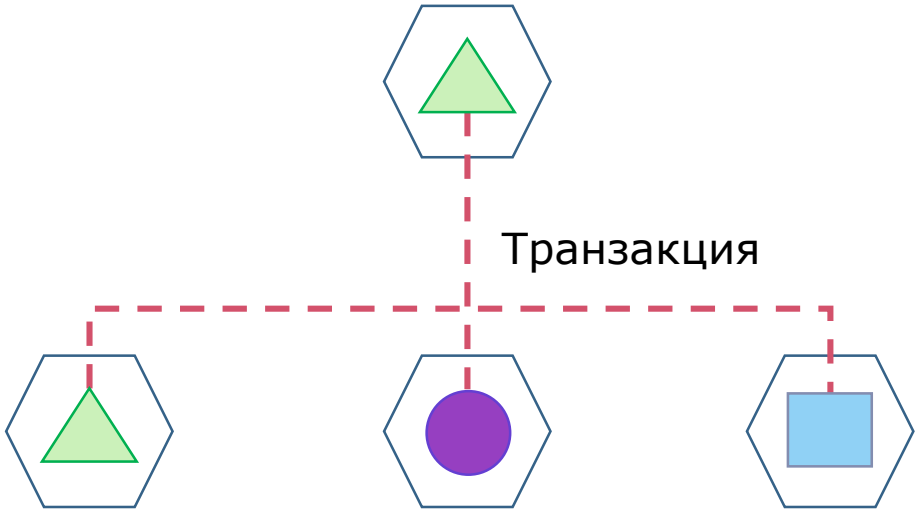
```
private void CreateCard(ICustomerAccountData data, ICardHolder cardHolder,
ICardData cardData)
{
  if (data.ContractAccount != null && data.ContractAccount.IsCARD_IDNull())
  {
    data.IsThereLoan = _customerFileData.IsThereLoan ?? false;
    cardData = _mapperFactory.Create().Map(data, cardData);

    var card = _cardsLogics.CreateCard(cardHolder, cardData);
    _cardsLogics.RegisterCard(card);
    SetCardId(data.Index, card.CardId);
    _cards.SaveChanges();
  }
}
```

Создание anticorruption layer

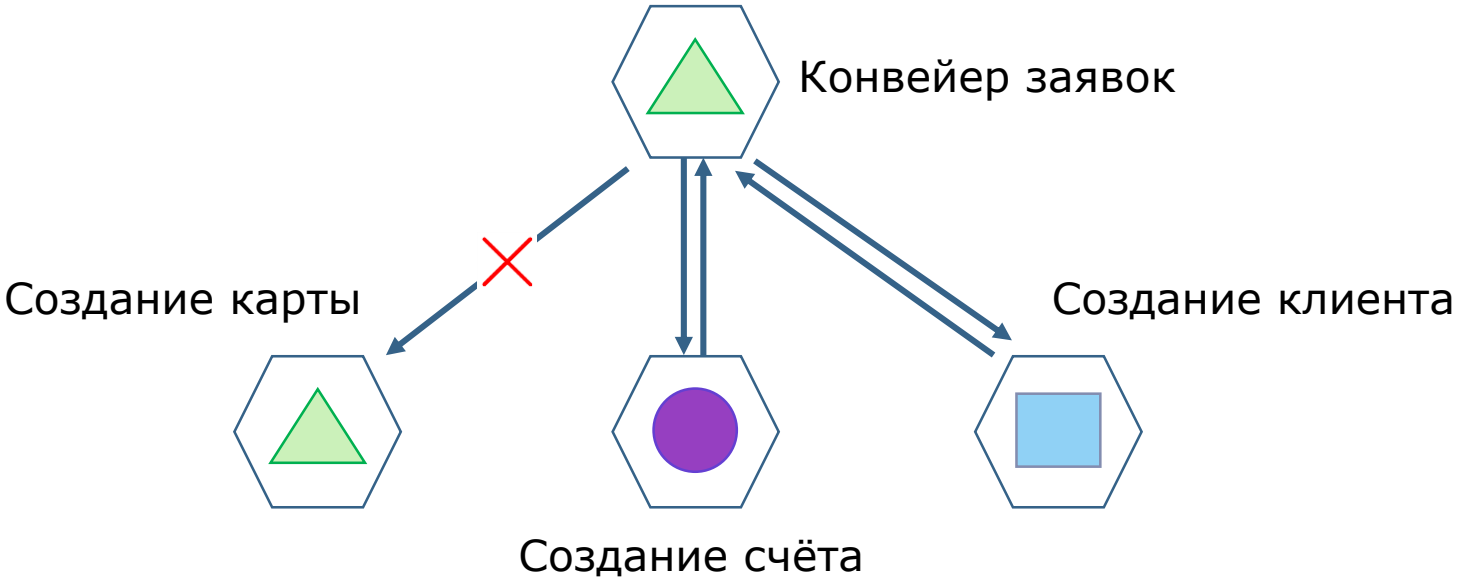
```
private void CreateCard(ICustomerAccountData data, ICardHolder cardHolder,
ICardData cardData)
{
  if (data.ContractAccount.Return(x => x.IsCARD_IDNull()))
  {
    var limit = CalculationLimit(data, cardHolder);

    var request = CreateCardRequestFactory.Create(cardData, cardHolder, limit);
    var card = _cardService.Execute(p => p.CreateCard(request));

    if (data.ContractAccount != null)
    {
       data.ContractAccount.CARD_ID = card.CardId;
    }
    SetCardId(data.Index, card.CardId);
  }
}
```

Выделение микросервиса

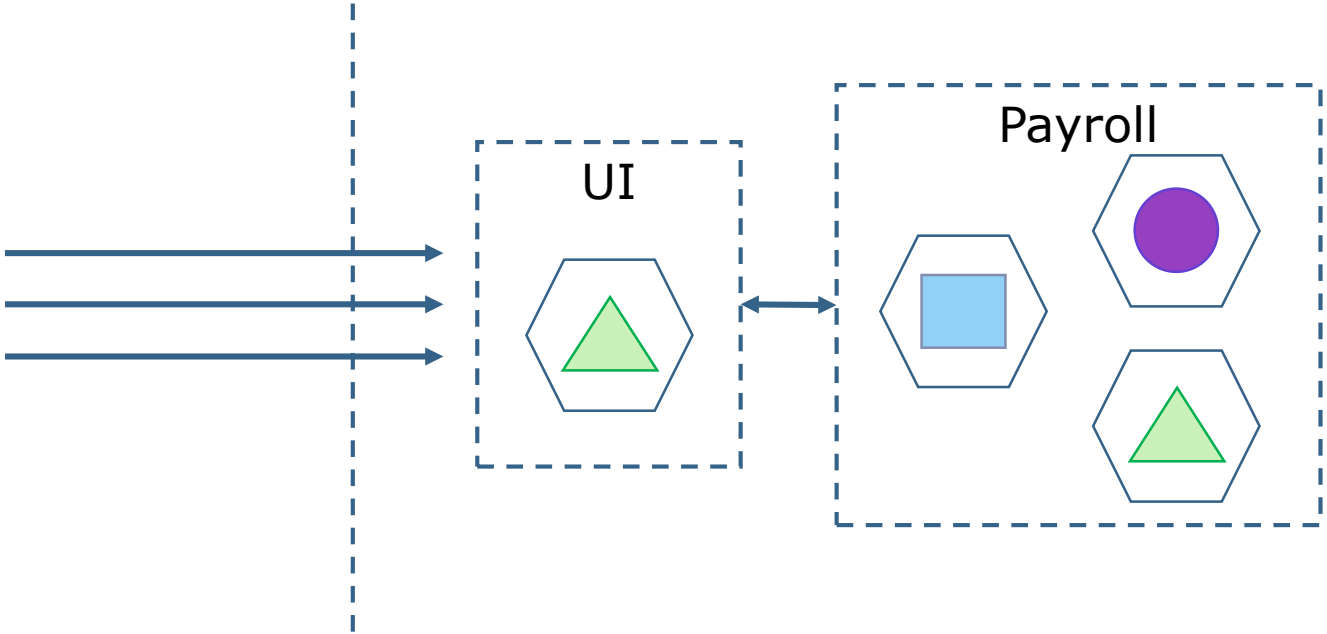
Не распределяйте транзакции между микросервисами



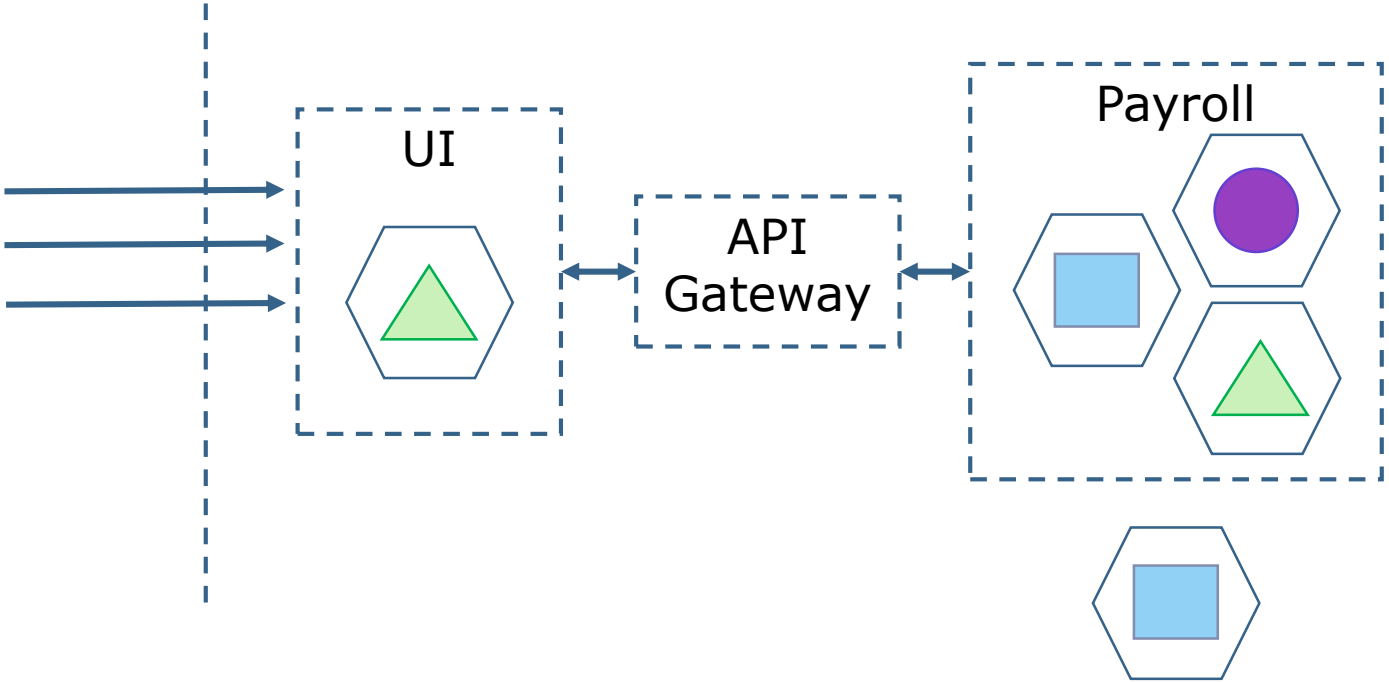
Итоговая согласованность



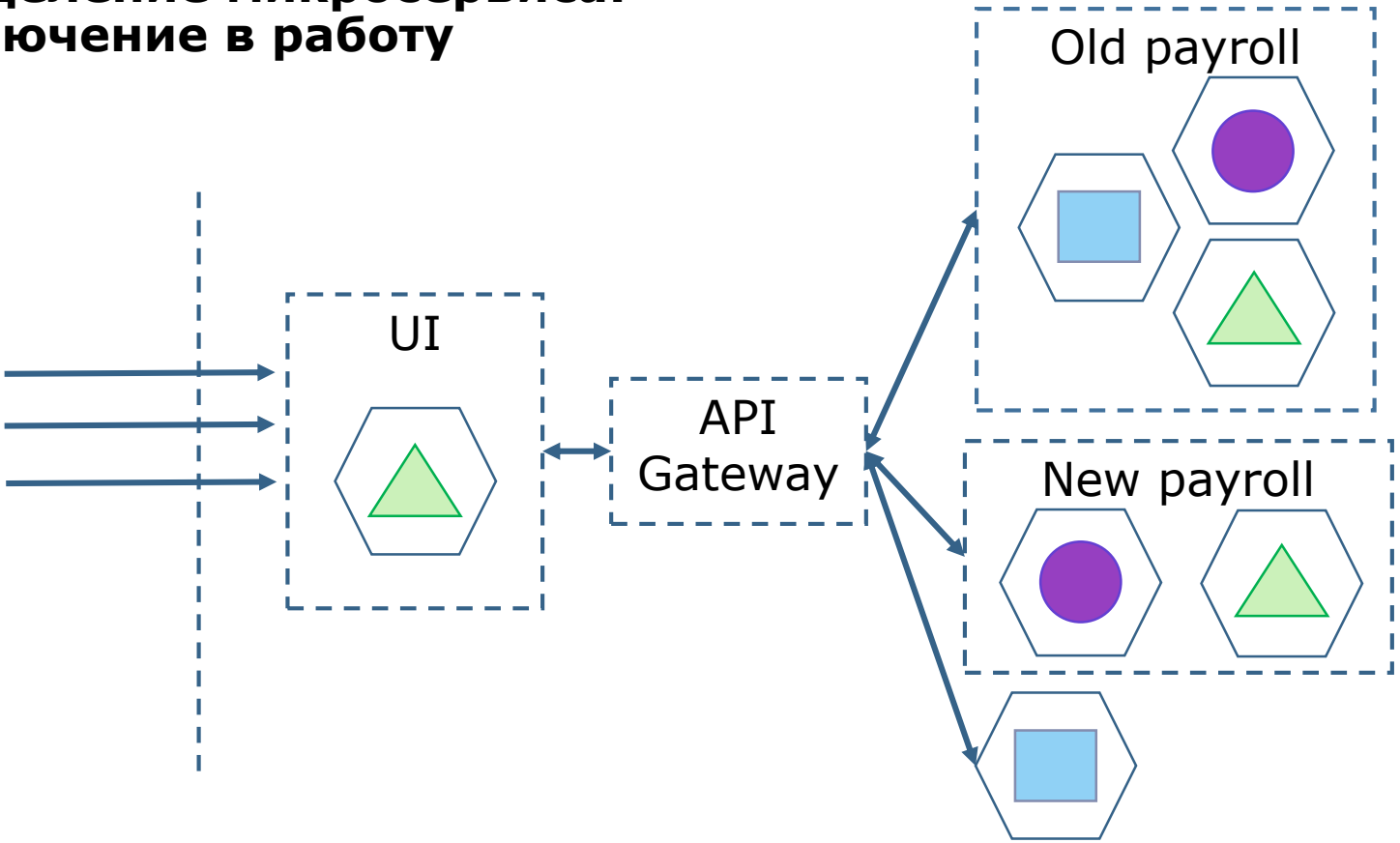
Выделение микросервиса: as is



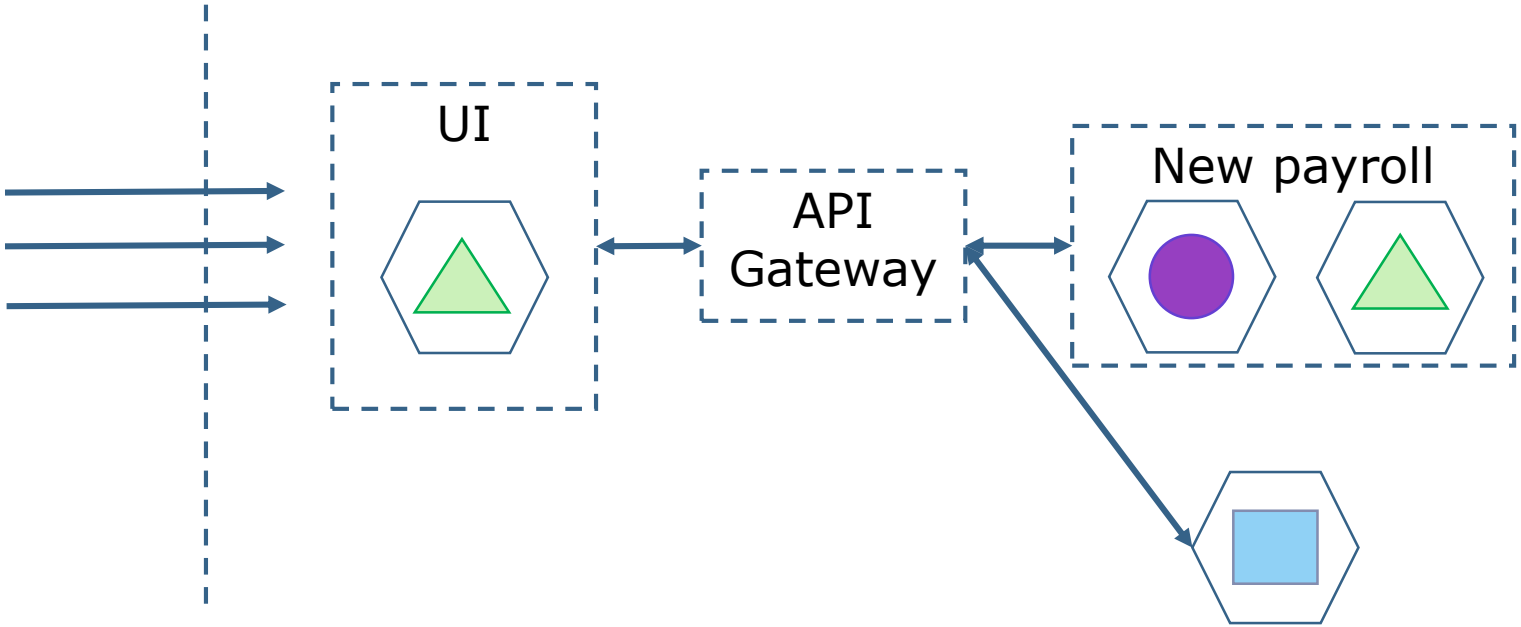
Выделение микросервиса: новый сервис



Выделение микросервиса: включение в работу

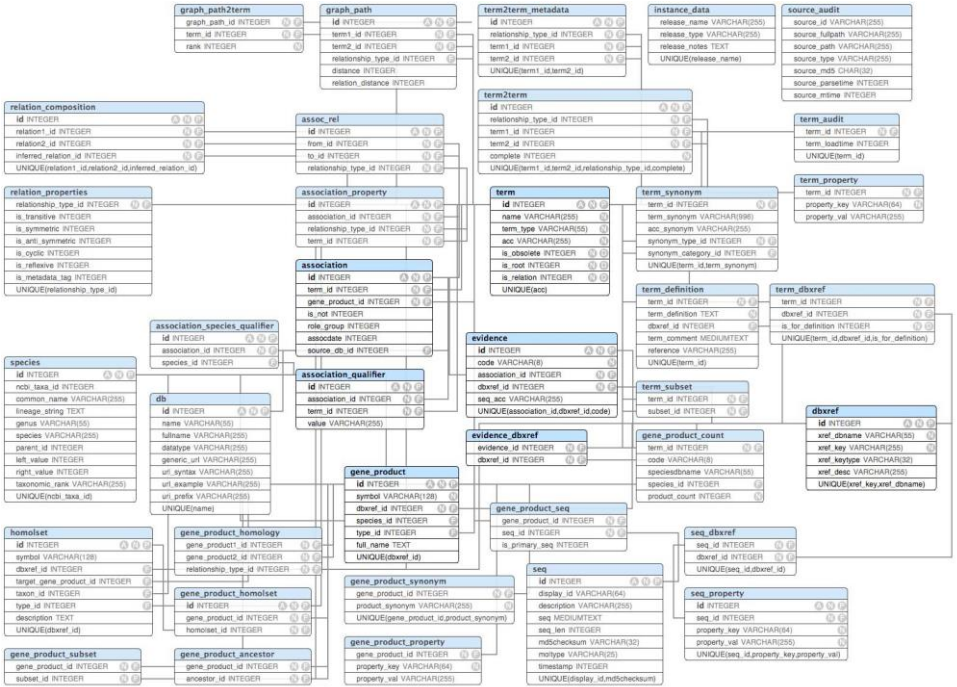


Выделение микросервиса: выведение старого монолита

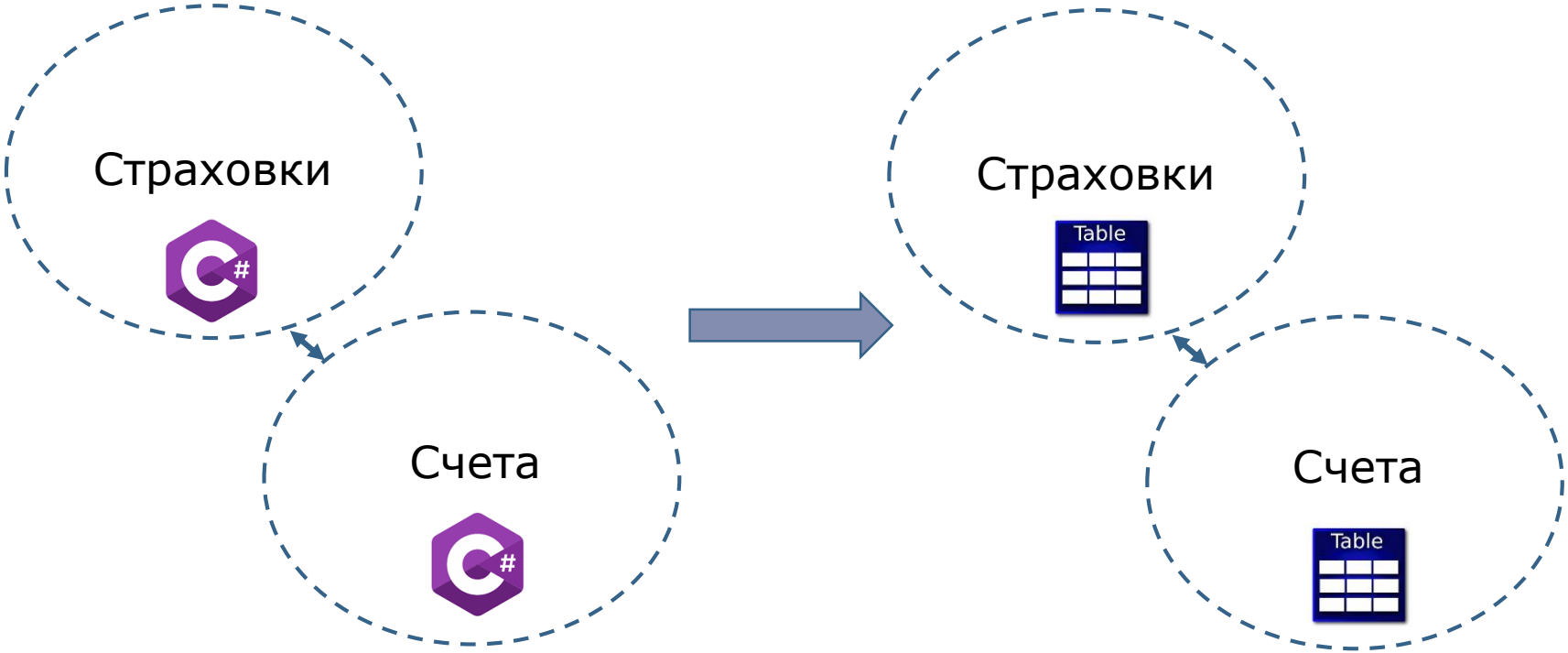


Работа с БД

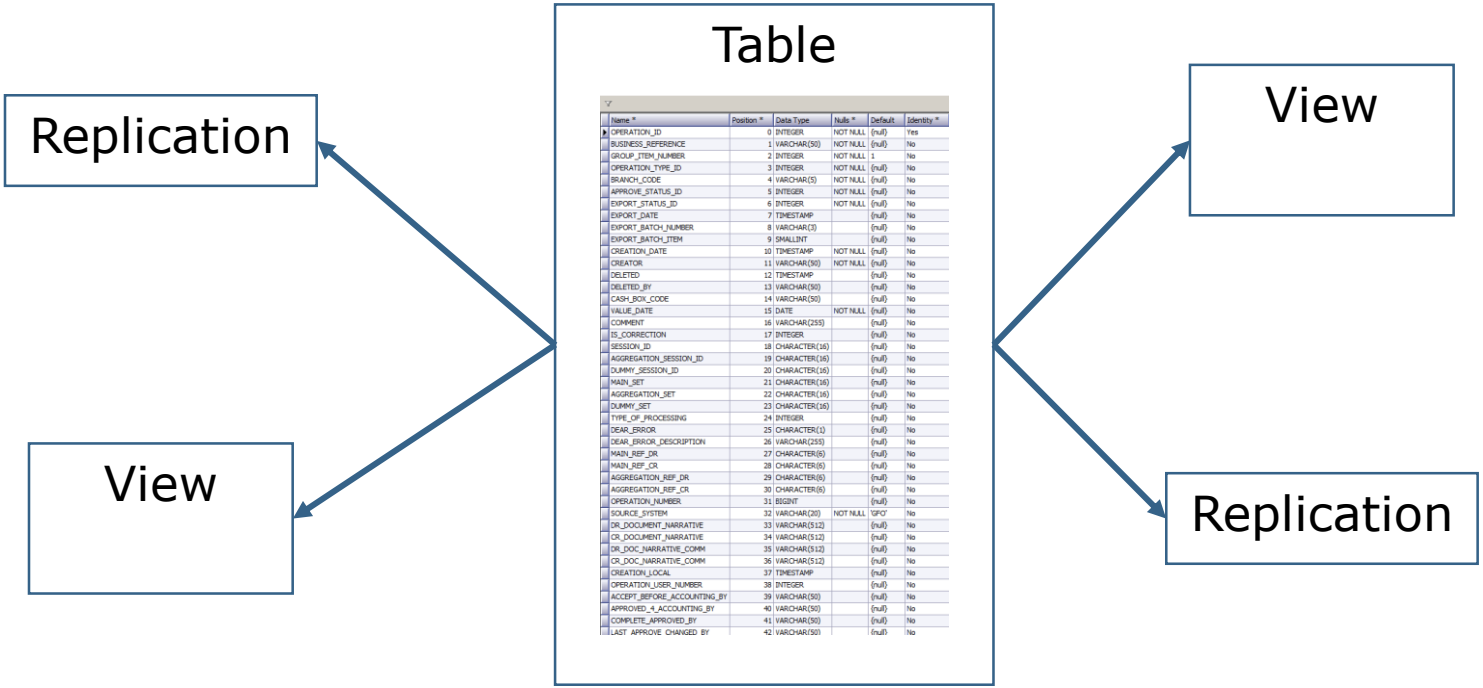
Структура БД



Ограниченные контексты БД



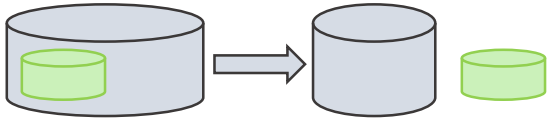
Что мешает разделению



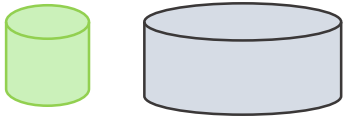
Name *	Position *	Data Type	Nulls *	Default	Identity *
OPERATION_ID	0	INTEGER	NOT NULL	(null)	Yes
BUSINESS_REFERENCE	1	VARCHAR(50)	NOT NULL	(null)	No
GROUP_ITEM_NUMBER	2	INTEGER	NOT NULL	1	No
OPERATION_TYPE_ID	3	INTEGER	NOT NULL	(null)	No
BRANCH_CODE	4	VARCHAR(5)	NOT NULL	(null)	No
APPROVE_STATUS_ID	5	INTEGER	NOT NULL	(null)	No
EXPORT_STATUS_ID	6	INTEGER	NOT NULL	(null)	No
EXPORT_DATE	7	TIMESTAMP		(null)	No
EXPORT_BATCH_NUMBER	8	VARCHAR(3)		(null)	No
EXPORT_BATCH_ITEM	9	SMALLINT		(null)	No
CREATION_DATE	10	TIMESTAMP	NOT NULL	(null)	No
CREATOR	11	VARCHAR(50)	NOT NULL	(null)	No
DELETED	12	TIMESTAMP		(null)	No
DELETED_BY	13	VARCHAR(50)		(null)	No
CASH_BOX_CODE	14	VARCHAR(50)		(null)	No
VALUE_DATE	15	DATE	NOT NULL	(null)	No
COMMENT	16	VARCHAR(255)		(null)	No
IS_CORRECTION	17	INTEGER		(null)	No
SESSION_ID	18	CHARACTER(16)		(null)	No
AGGREGATION_SESSION_ID	19	CHARACTER(16)		(null)	No
DUMMY_SESSION_ID	20	CHARACTER(16)		(null)	No
MAIN_SET	21	CHARACTER(16)		(null)	No
AGGREGATION_SET	22	CHARACTER(16)		(null)	No
DUMMY_SET	23	CHARACTER(16)		(null)	No
TYPE_OF_PROCESSING	24	INTEGER		(null)	No
DEAR_ERROR	25	CHARACTER(1)		(null)	No
DEAR_ERROR_DESCRIPTION	26	VARCHAR(255)		(null)	No
MAIN_REF_DR	27	CHARACTER(6)		(null)	No
MAIN_REF_CR	28	CHARACTER(6)		(null)	No
AGGREGATION_REF_DR	29	CHARACTER(6)		(null)	No
AGGREGATION_REF_CR	30	CHARACTER(6)		(null)	No
OPERATION_NUMBER	31	BIGINT		(null)	No
SOURCE_SYSTEM	32	VARCHAR(20)	NOT NULL	'GFO'	No
DR_DOCUMENT_NARRATIVE	33	VARCHAR(512)		(null)	No
CR_DOCUMENT_NARRATIVE	34	VARCHAR(512)		(null)	No
DR_DOC_NARRATIVE_COMM	35	VARCHAR(512)		(null)	No
CR_DOC_NARRATIVE_COMM	36	VARCHAR(512)		(null)	No
CREATION_LOCAL	37	TIMESTAMP		(null)	No
OPERATION_USER_NUMBER	38	INTEGER		(null)	No
ACCEPT_BEFORE_ACCOUNTING_BY	39	VARCHAR(50)		(null)	No
APPROVED_4_ACCOUNTING_BY	40	VARCHAR(50)		(null)	No
COMPLETE_APPROVED_BY	41	VARCHAR(50)		(null)	No
LAST_APPROVE_CHANGED_BY	42	VARCHAR(50)		(null)	No

Способы разделения БД

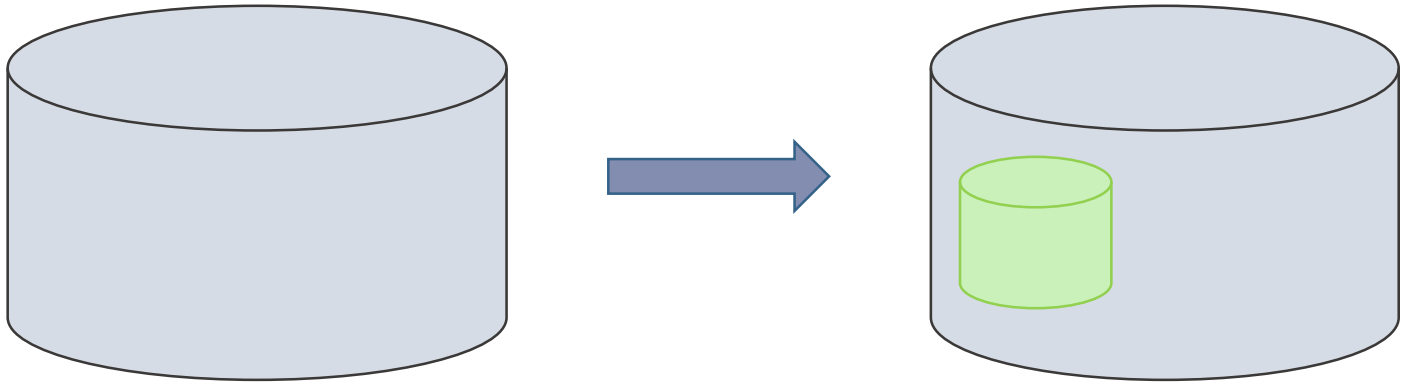
I. Отделение существующих таблиц



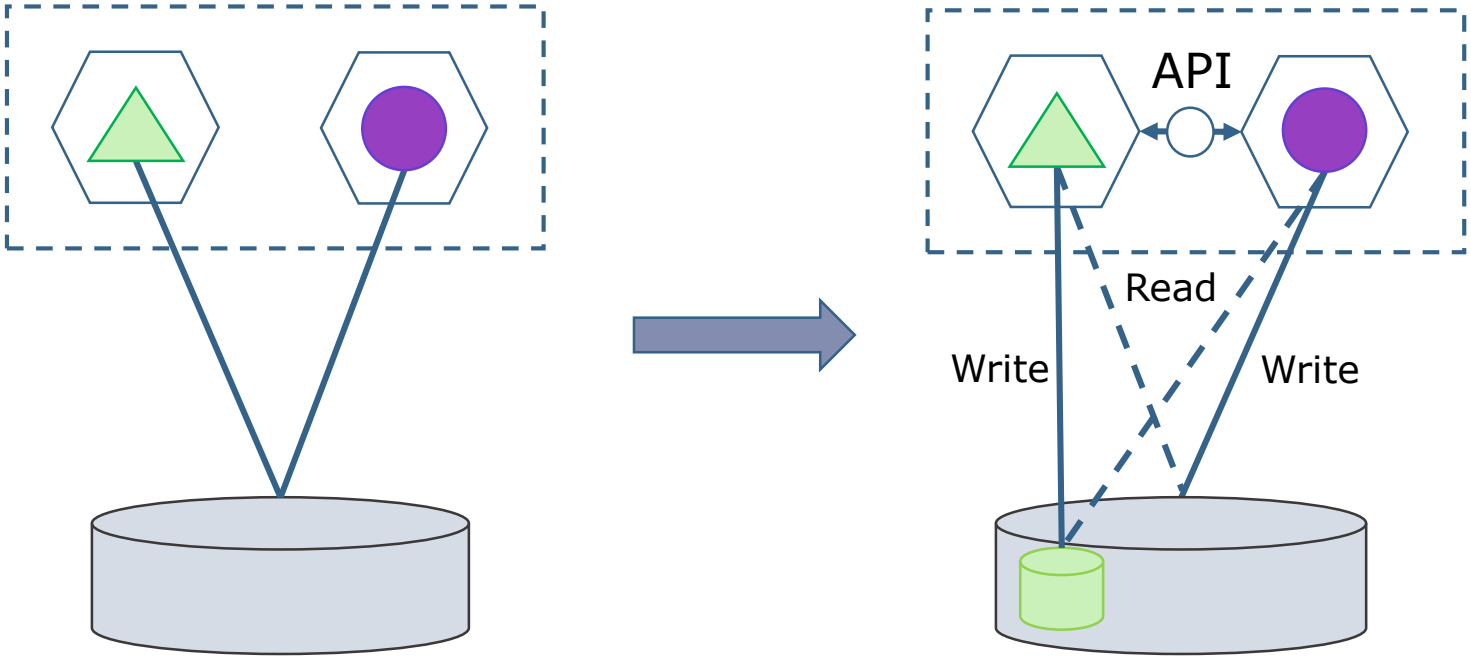
II. Отделение с переработкой



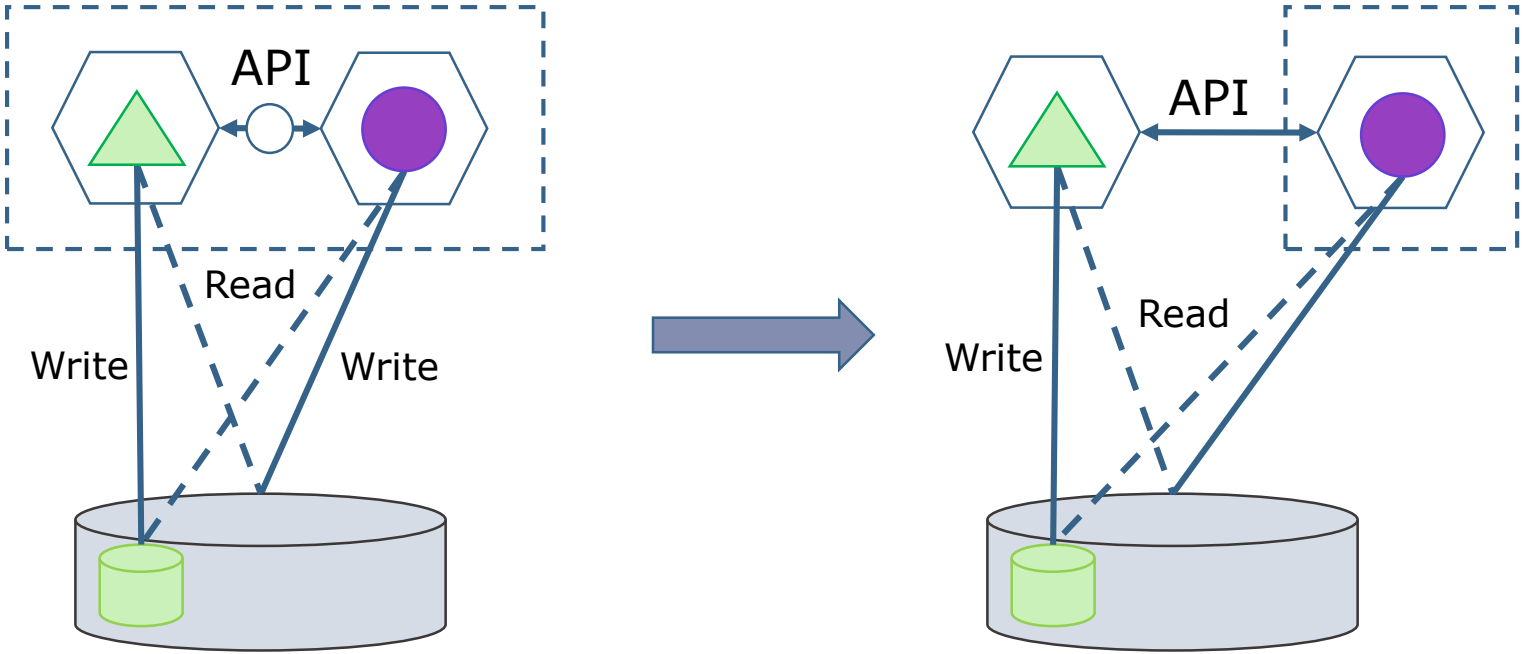
I. Выделение таблиц сервиса



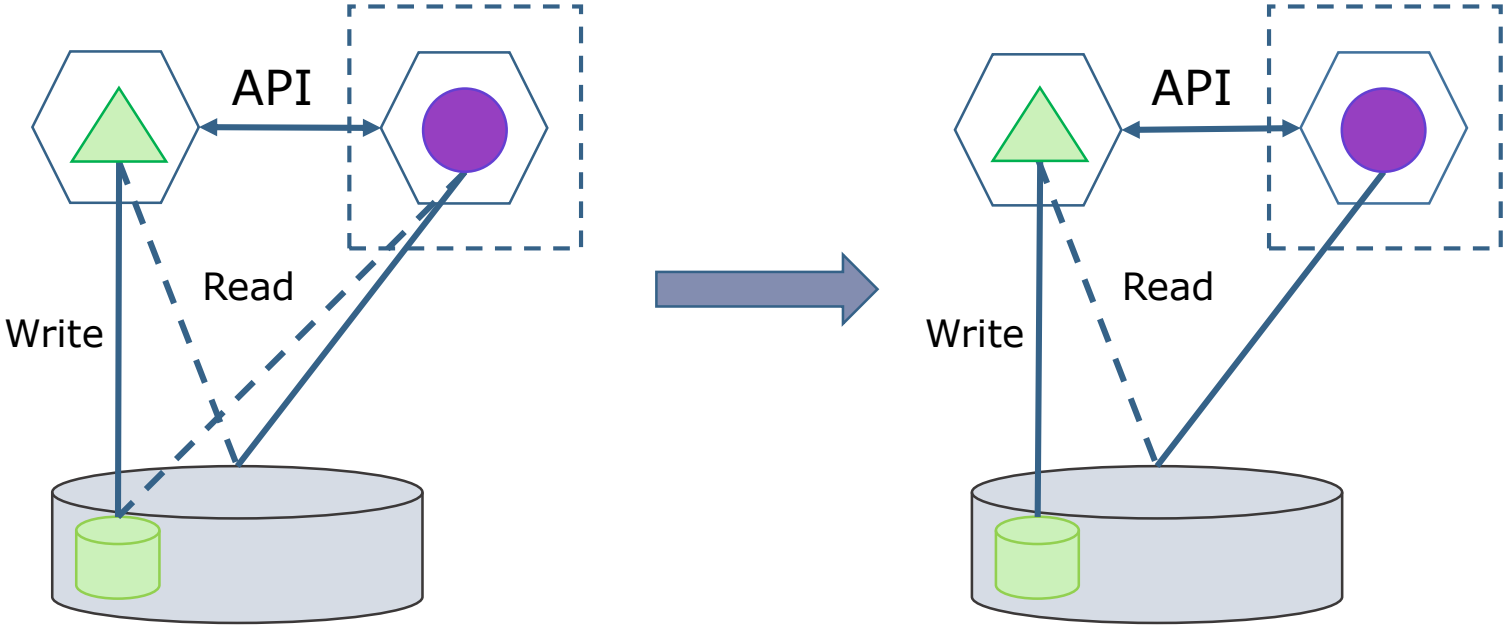
I. Выделение сервиса



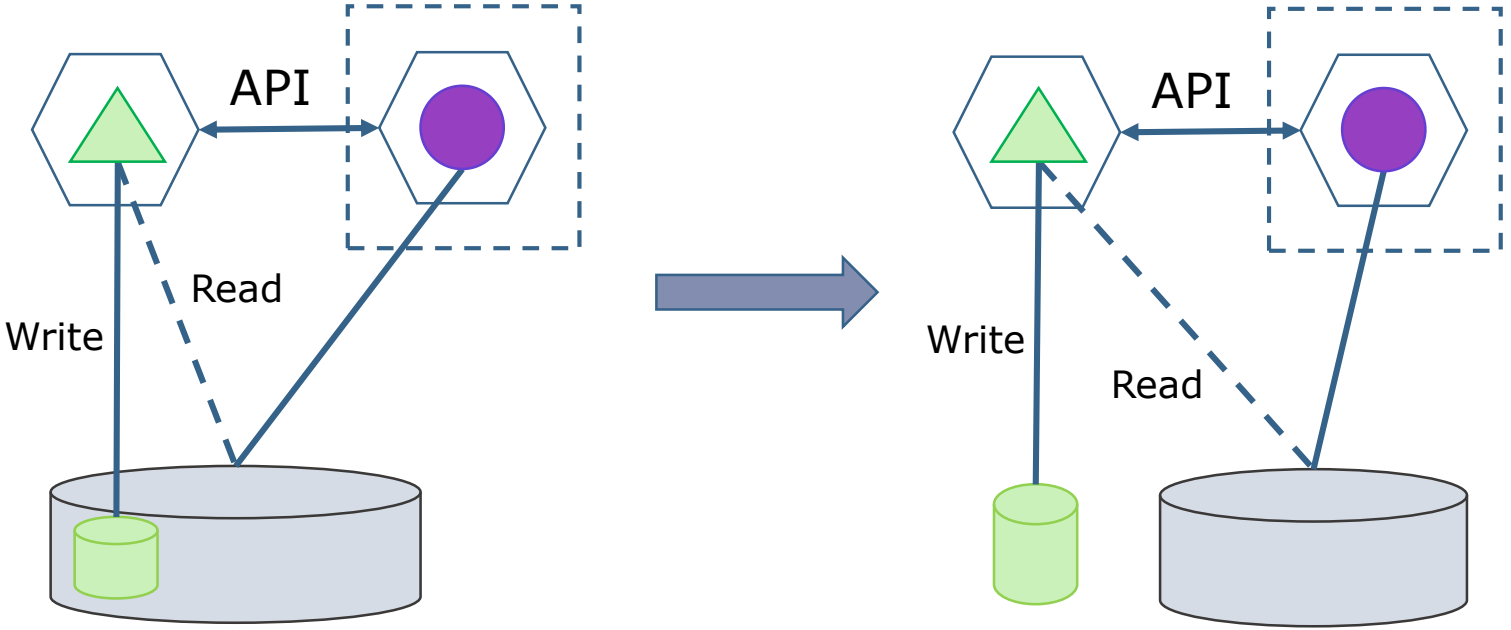
I. Выделение сервиса



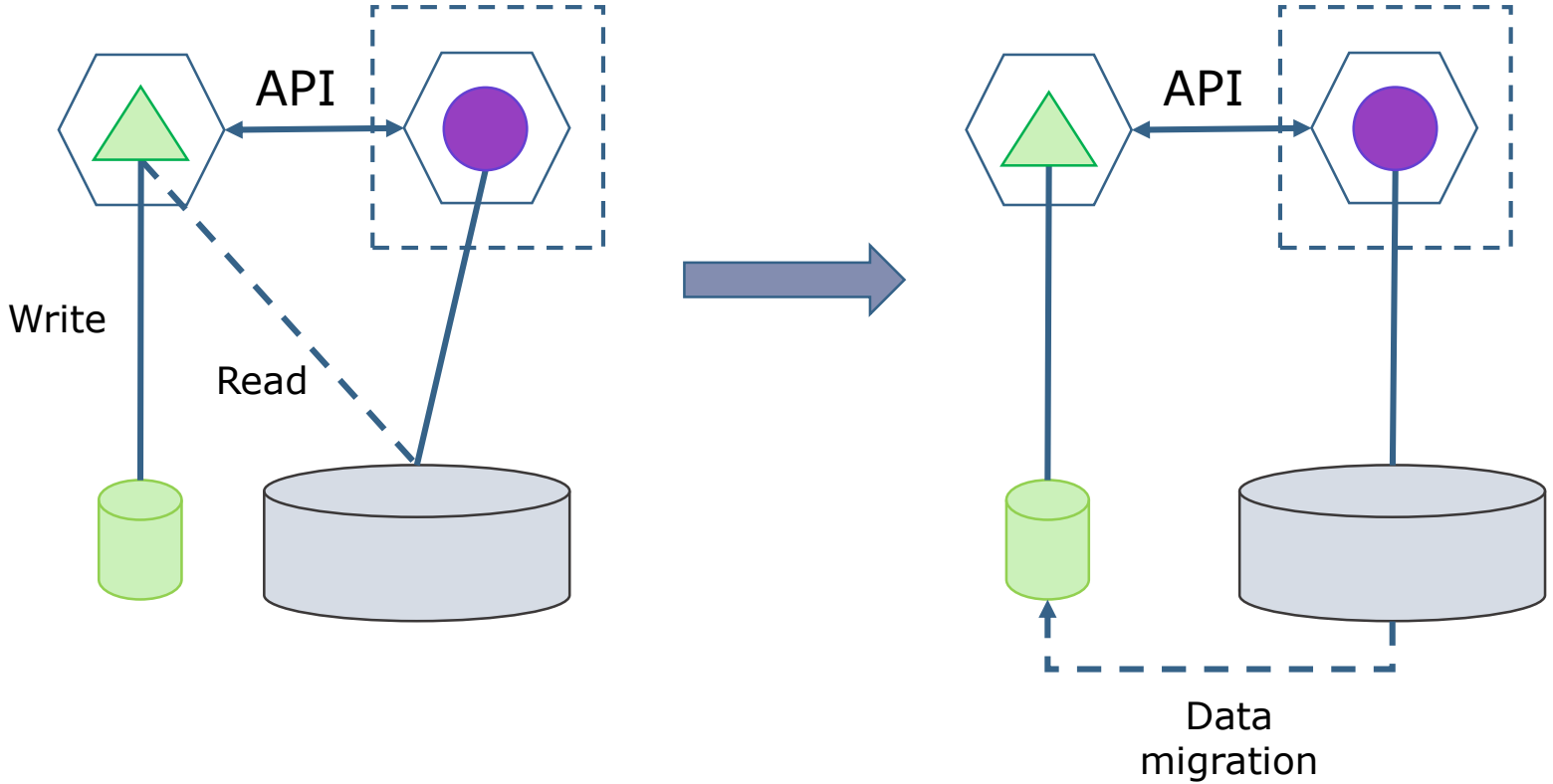
I. Выделение сервиса



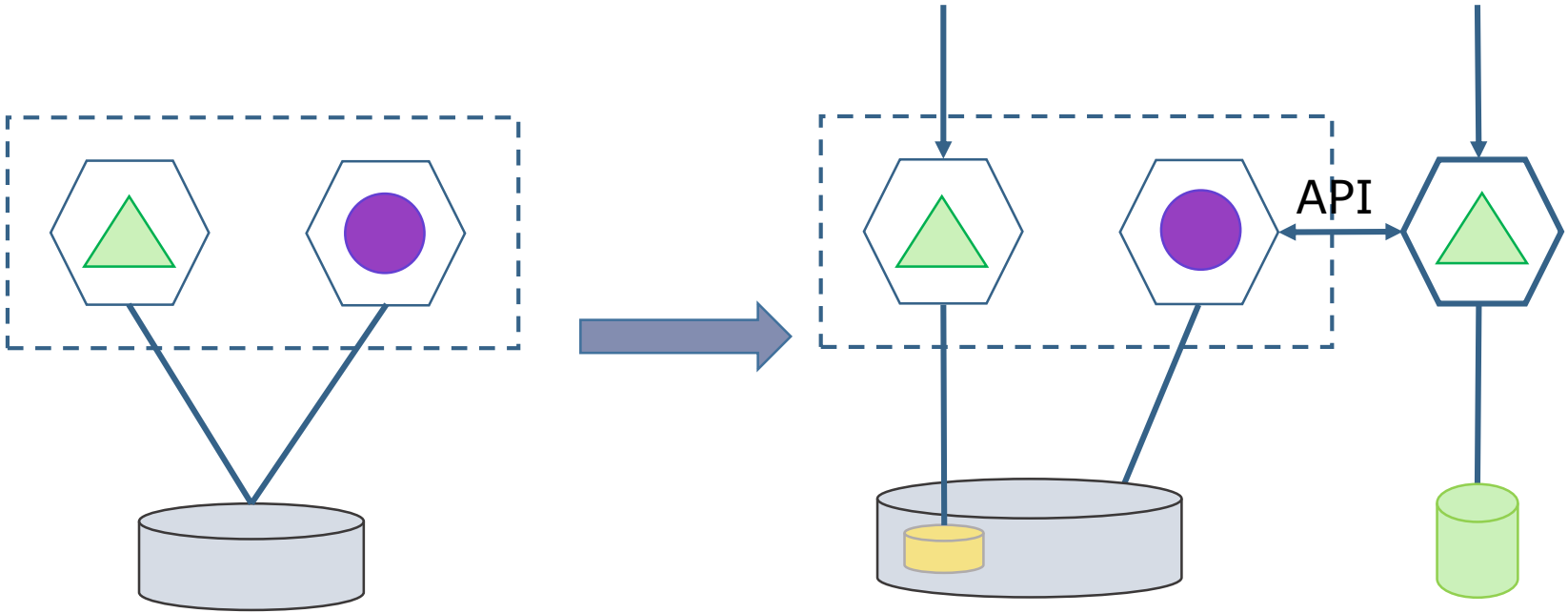
I. Выделение собственной БД



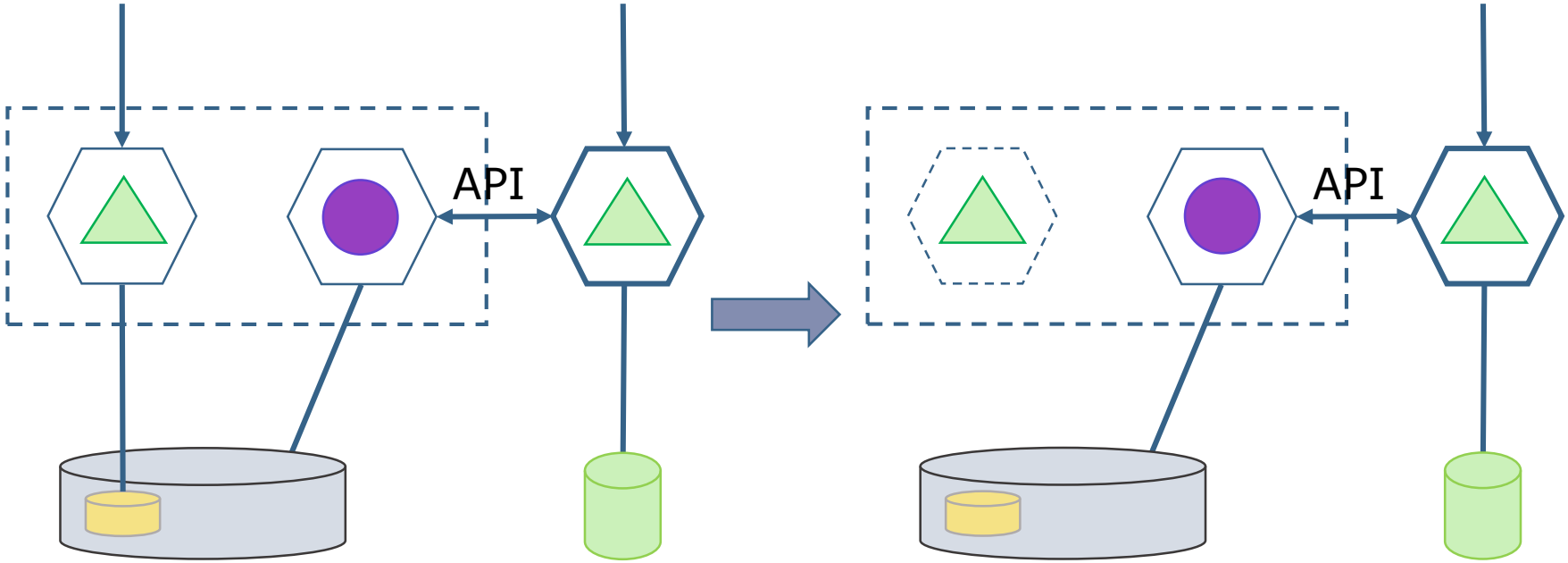
I. Миграция данных



II. Отделение с переработкой

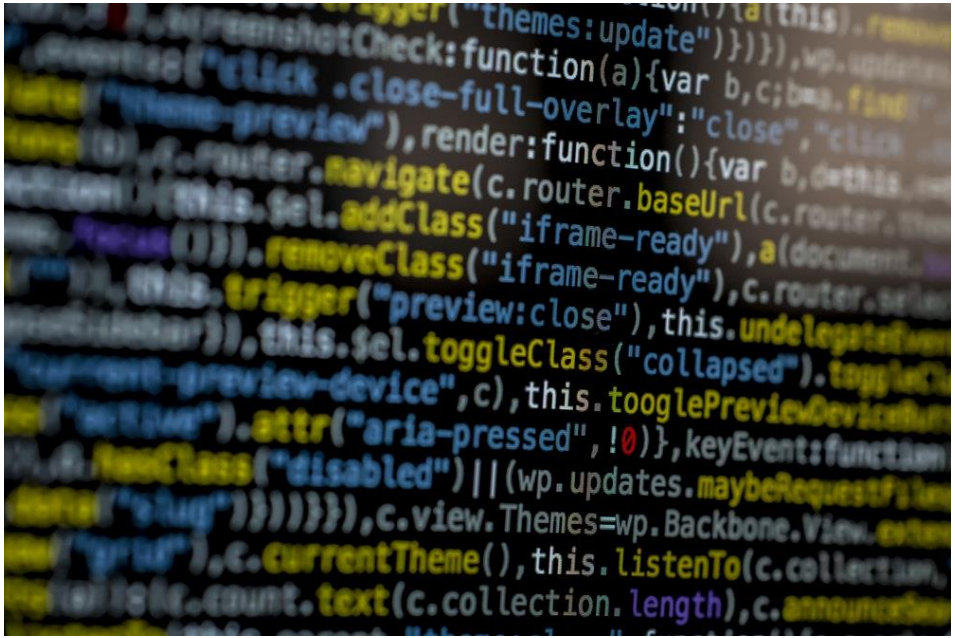


II. Отделение с переработкой

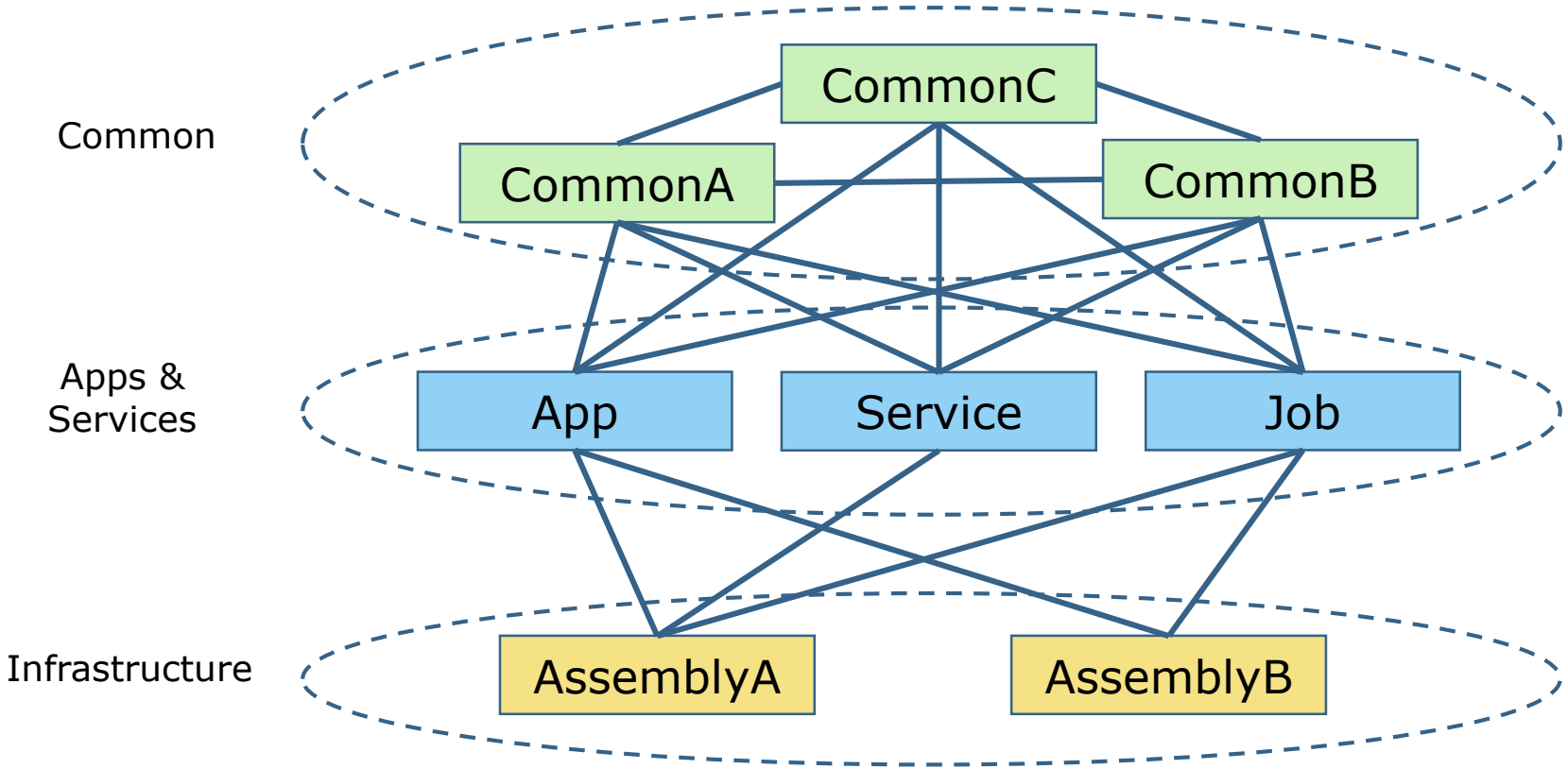


II. Удаление старой структуры

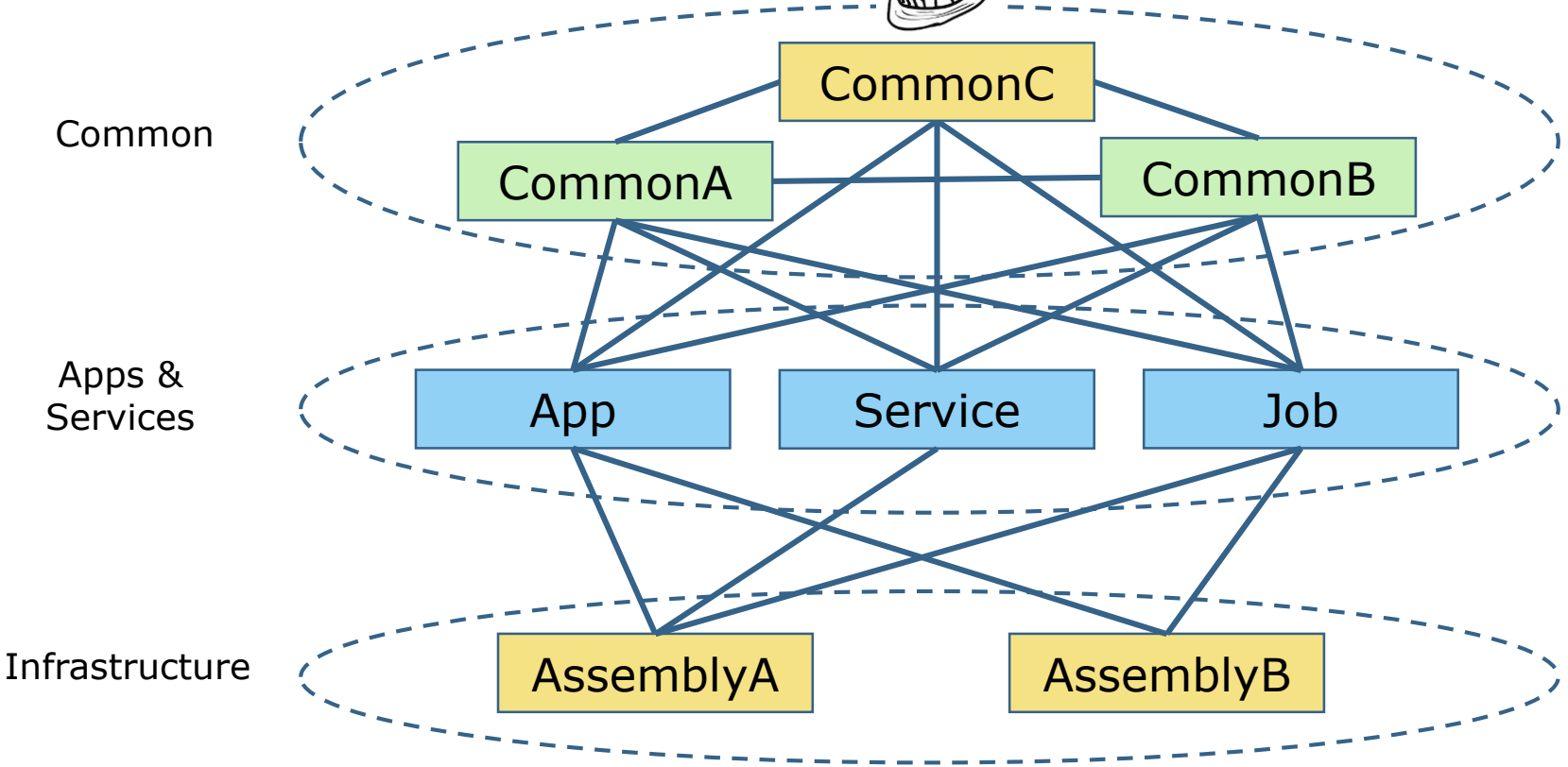
Работа с исходным кодом



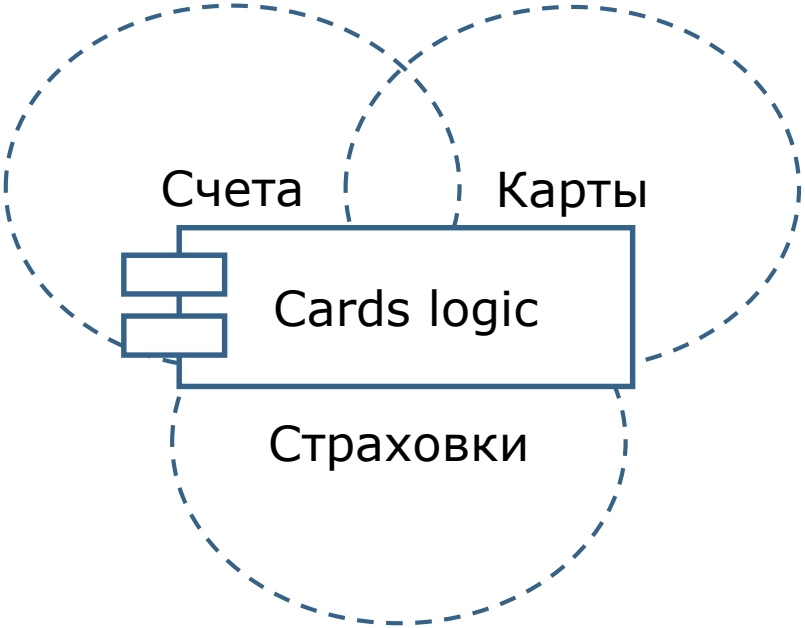
Структура решения



Структура решения

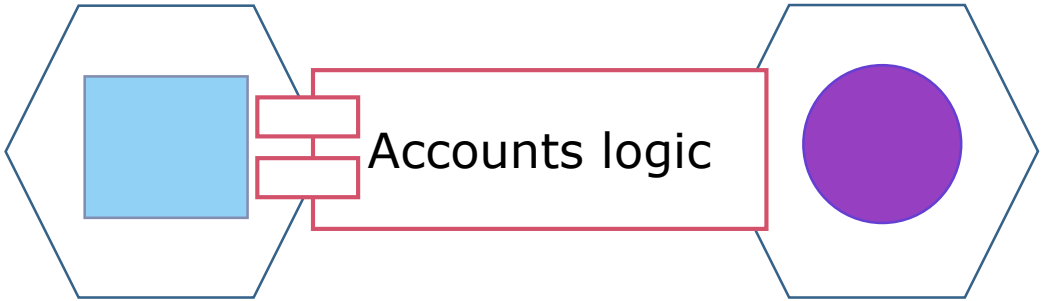


Ограниченные контексты



Разделение кода

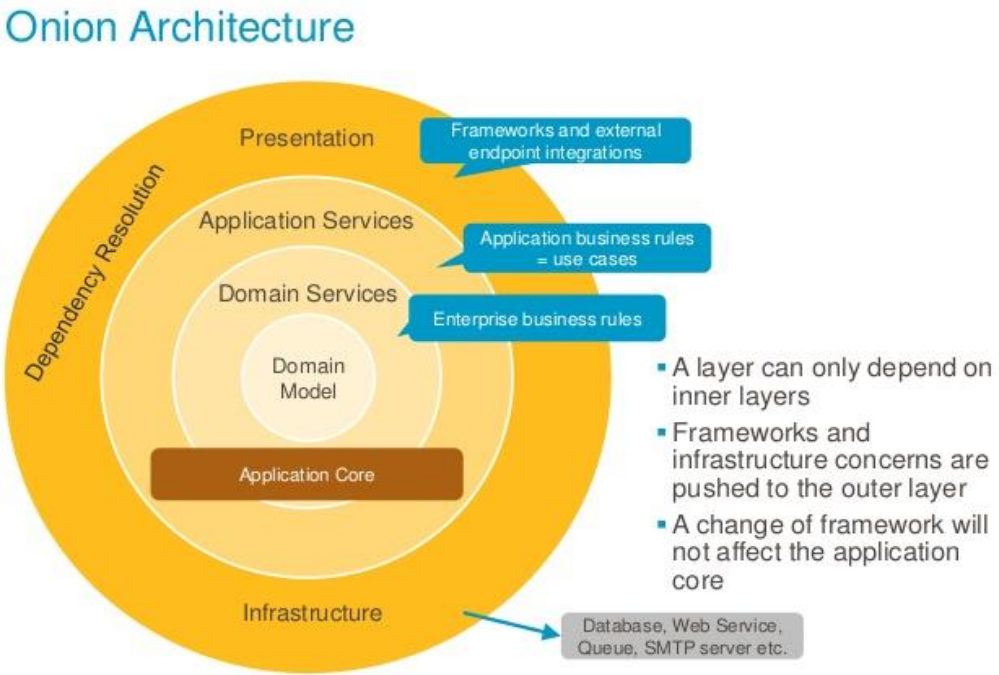
Не использовать общие сборки с бизнес-логикой



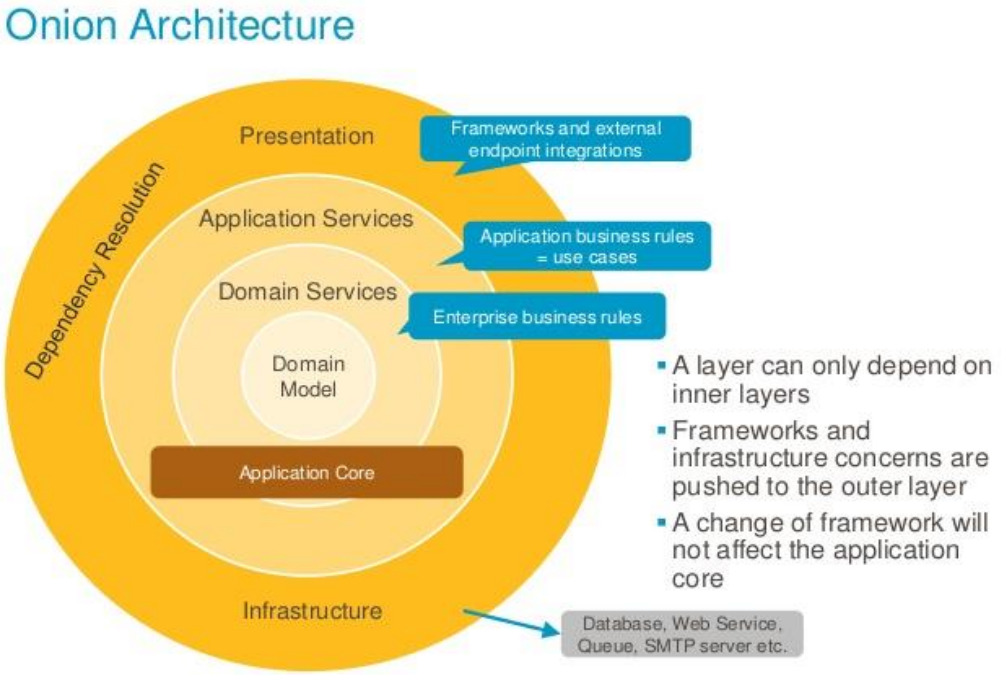
Разделение кода

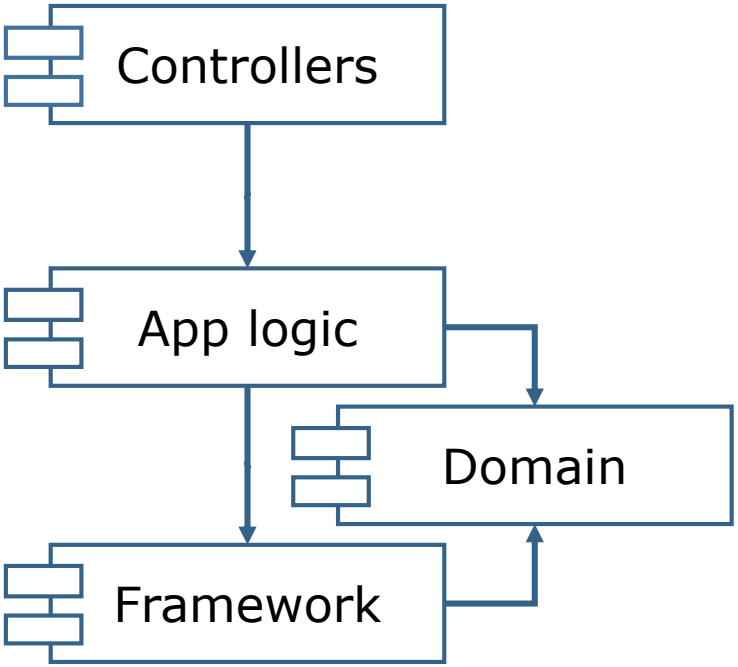
Использовать общие технические сборки

Структура новых сервисов

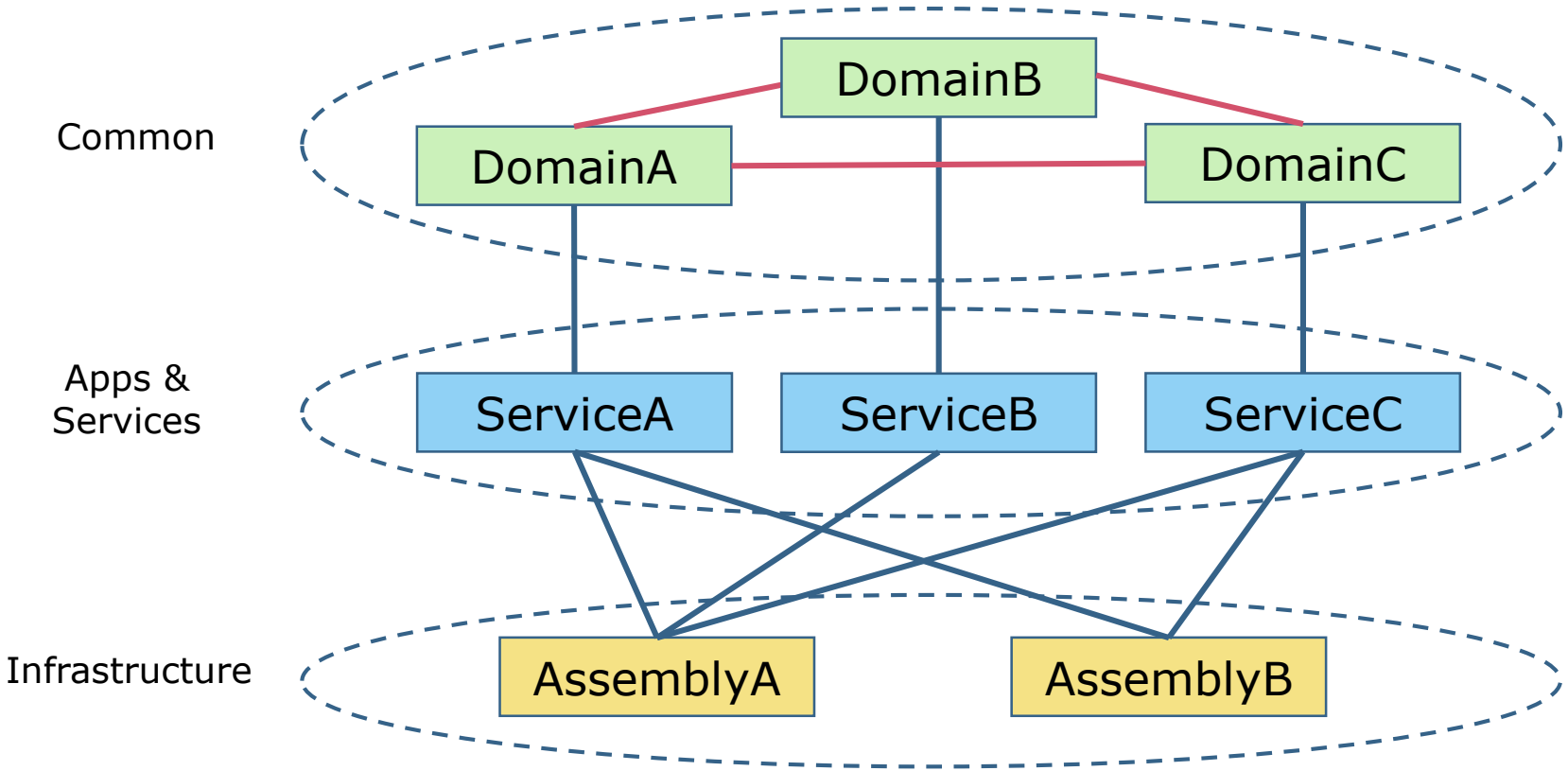


Структура новых сервисов

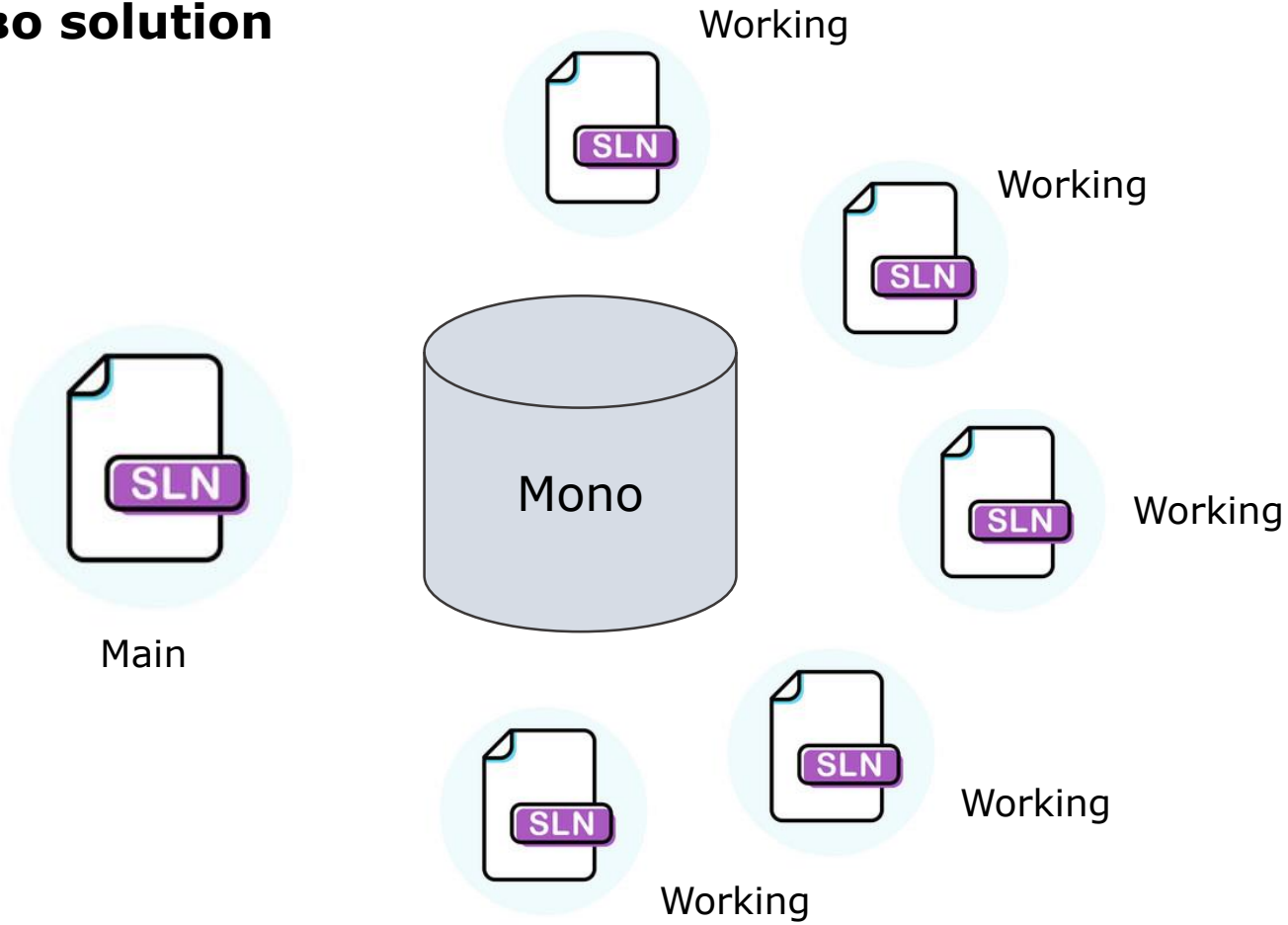




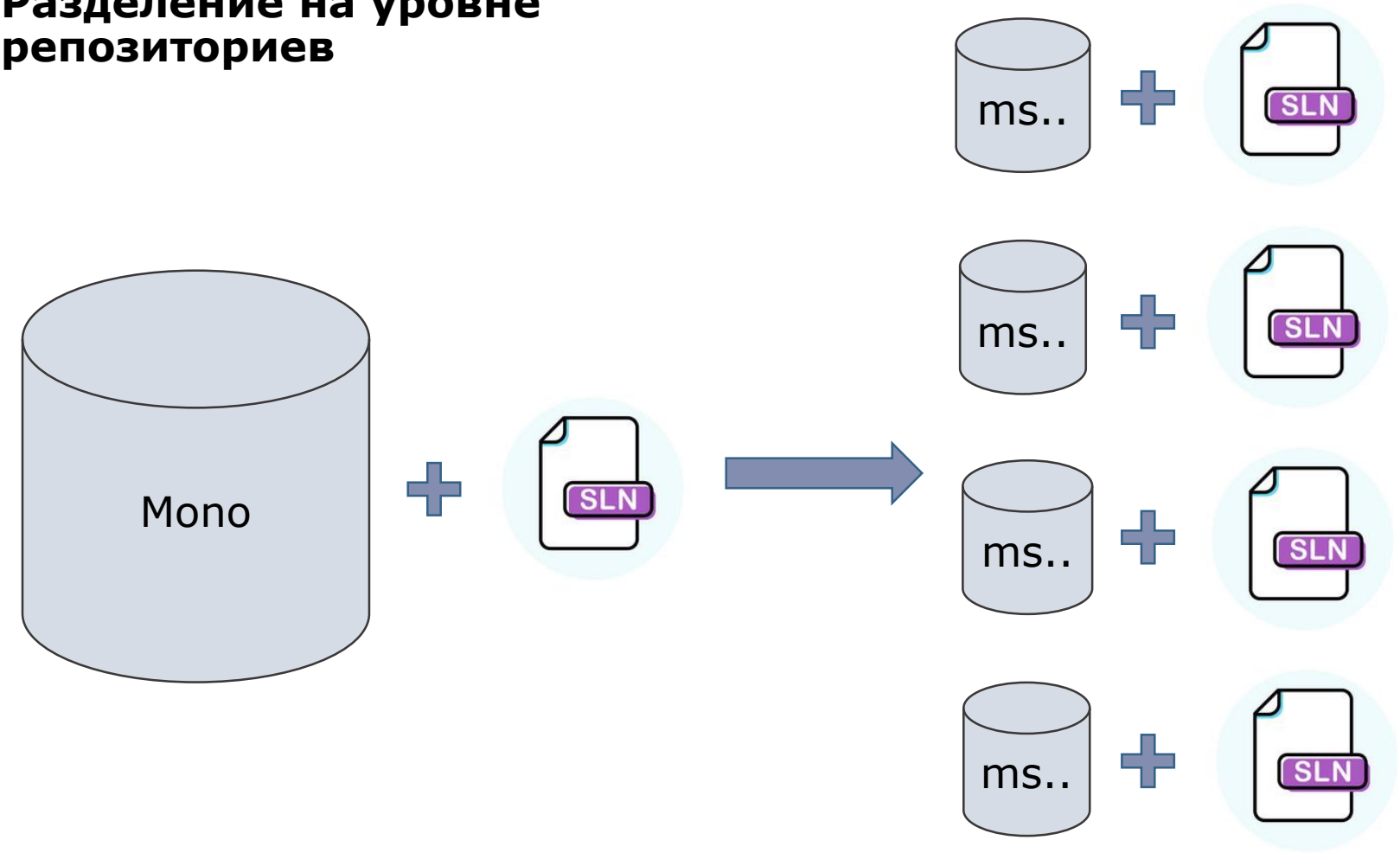
Структура решения



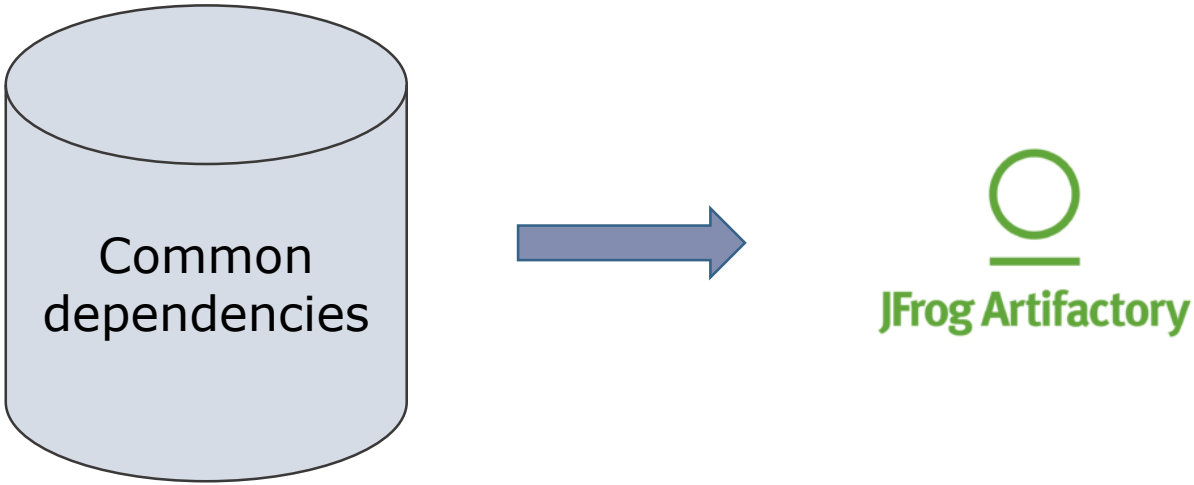
Множество solution



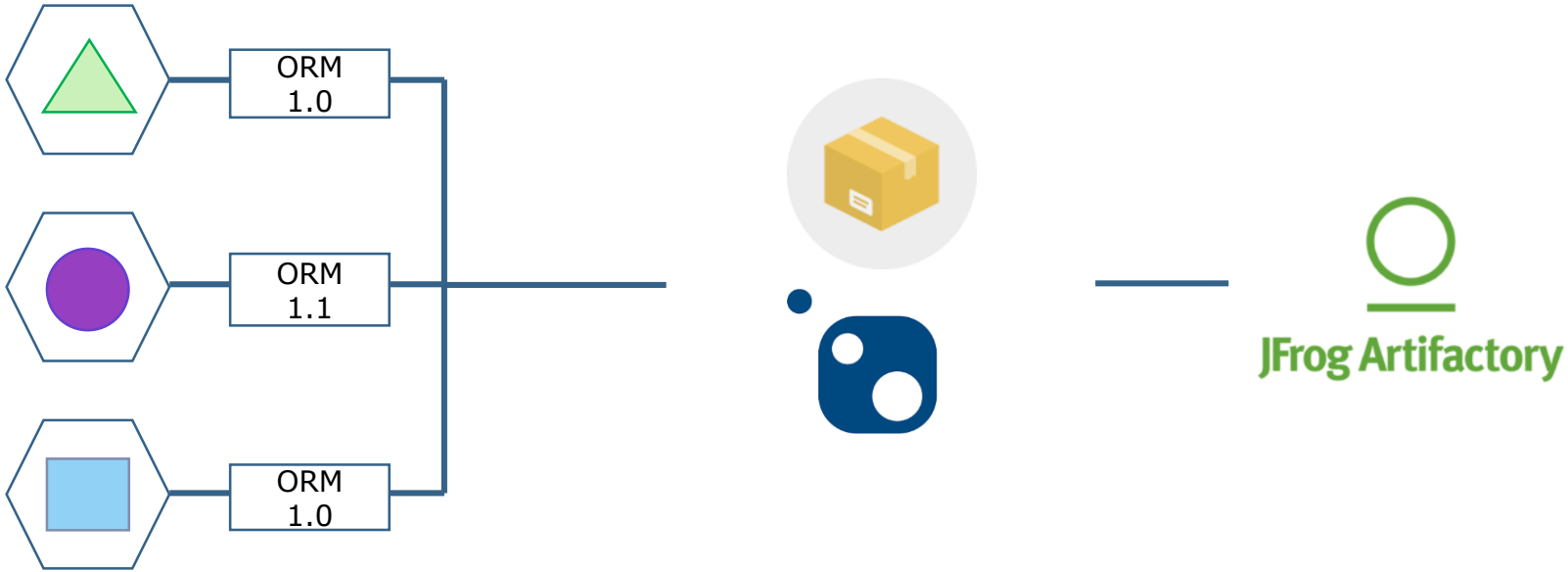
Разделение на уровне репозиториев



Репозиторий для зависимостей

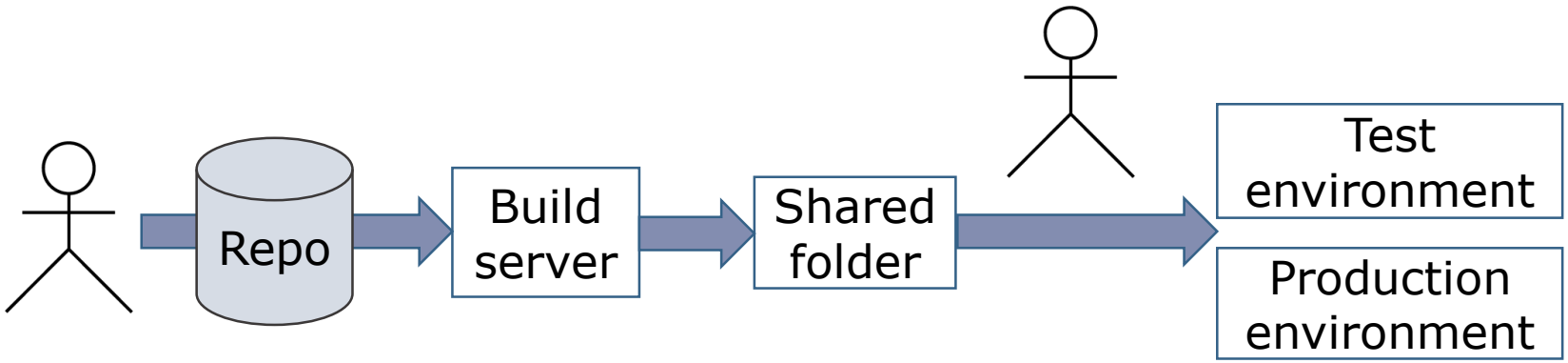


Общие зависимости



Проблемы инфраструктуры

Ручная установка на среды



CI/CD


Система контроля версий
Сервер контроля версий
Сервер сборки и деплоя
Менеджер репозиториев
Средство автоматизации
Go live
git
Bitbucket
Bamboo
LIQUIBASE
JFrog Artifactory
ANSIBLE

Раздельное логирование

2014-07-11 12:55:07,584
[5][admin]- Request

2014-07-11 12:55:08,341
[12][admin]- Action

Параметры логирования

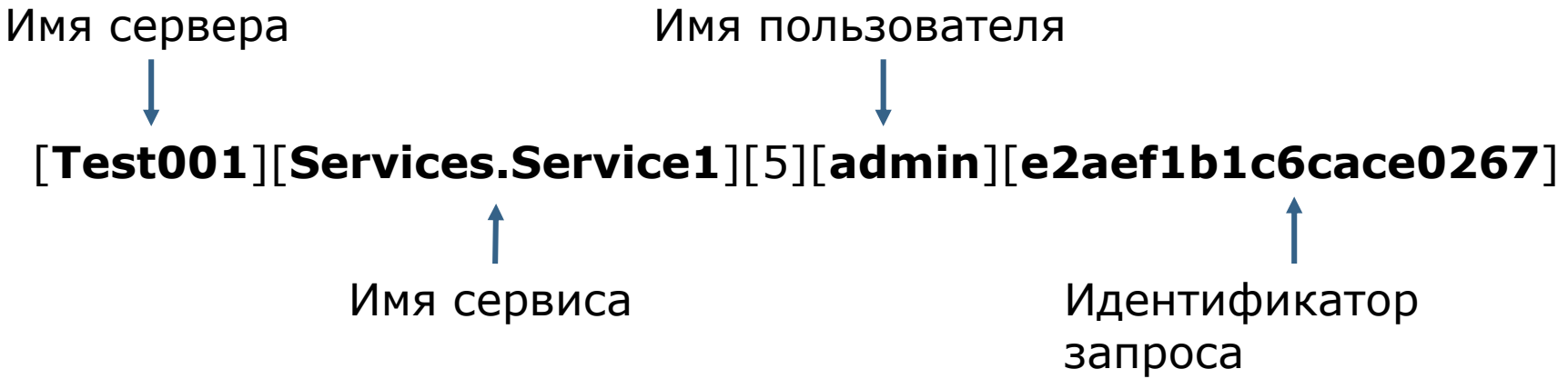
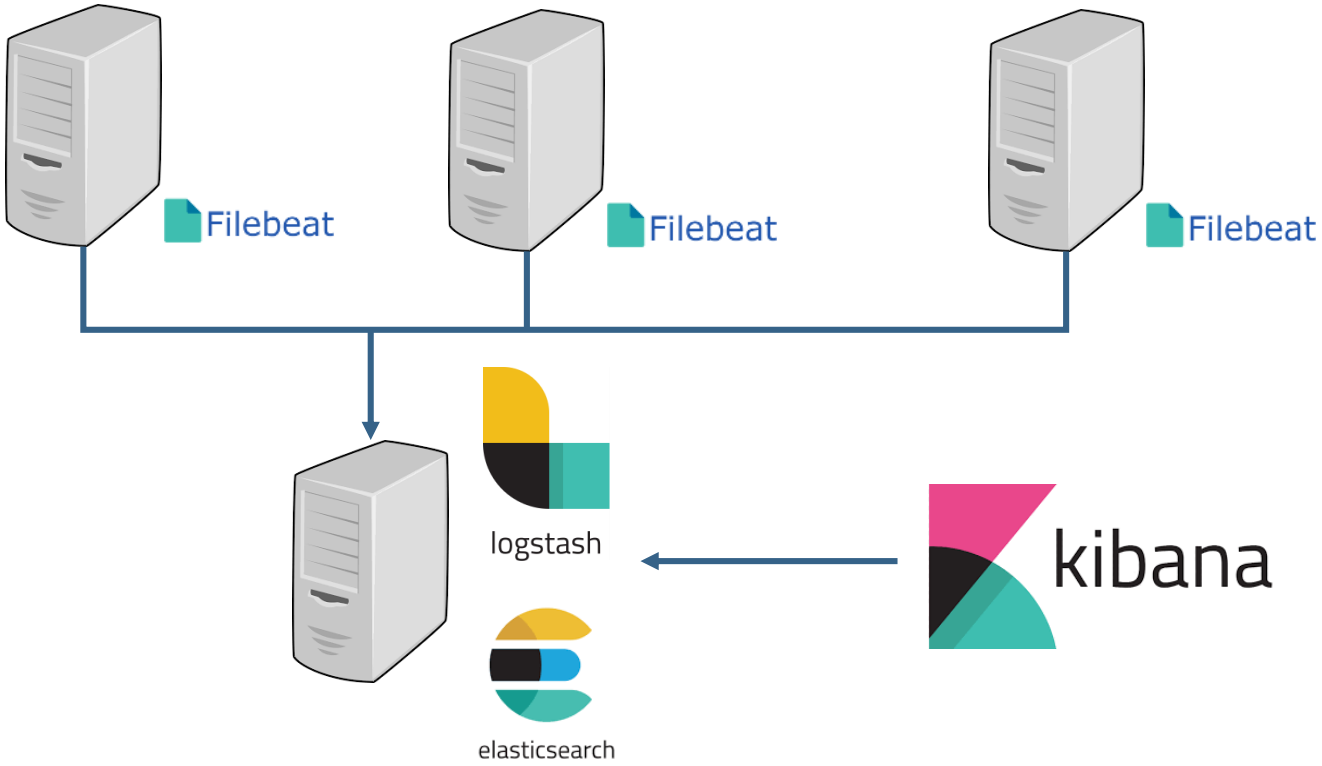
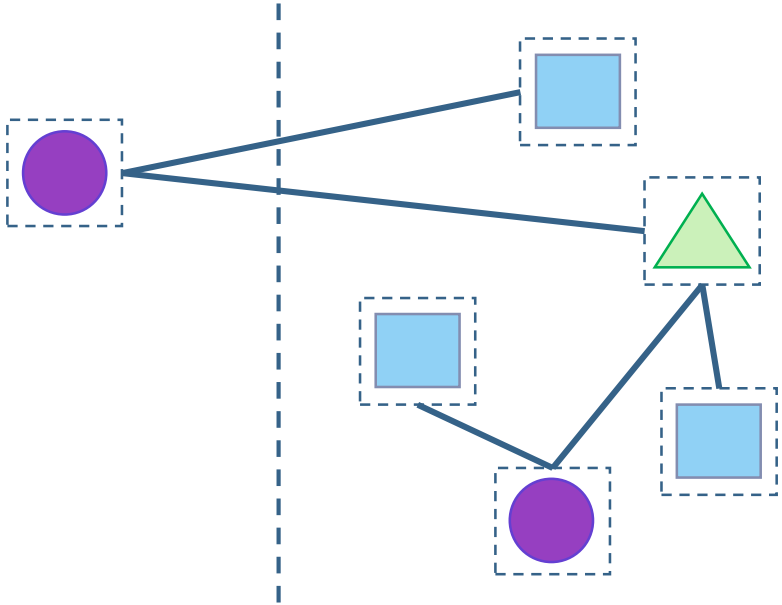


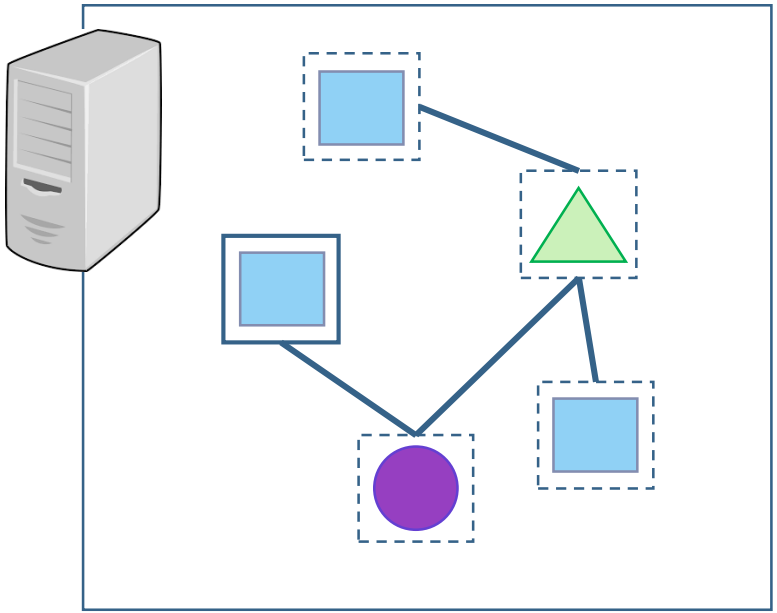
Схема логирования



Тестирование и отладка связанных сервисов



Локальная отладка



Локальная машина

Отладка с серверной частью

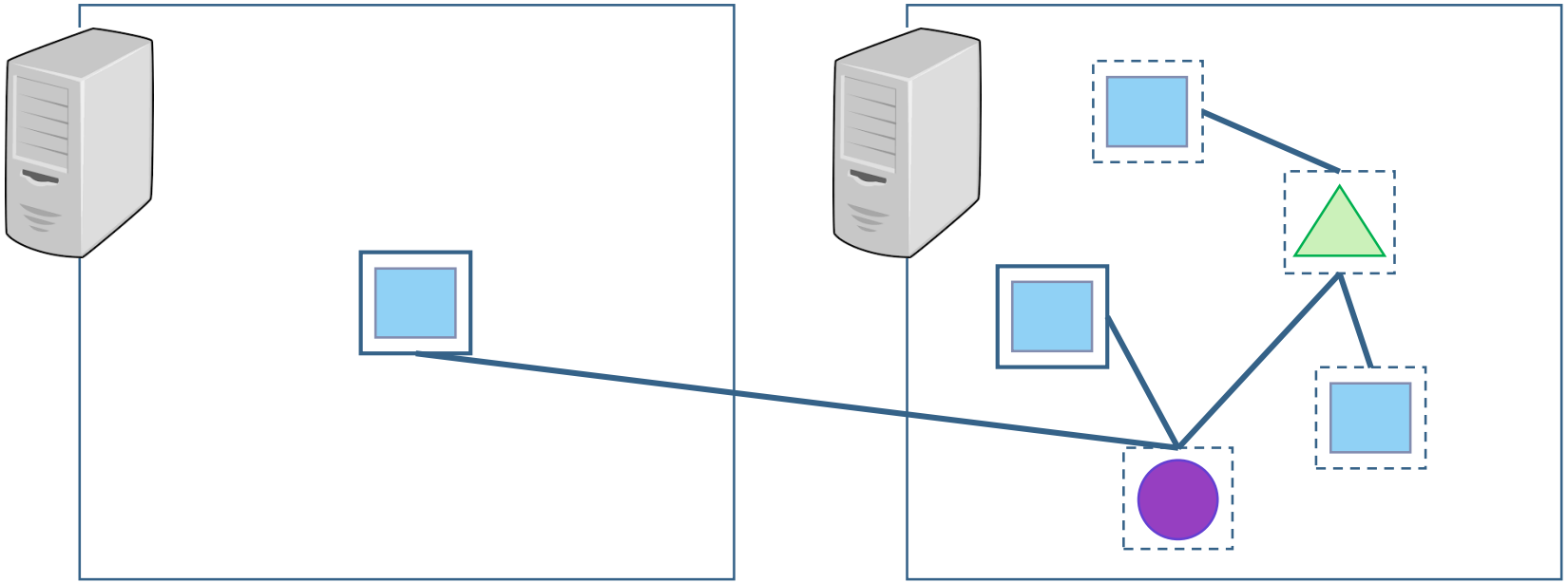
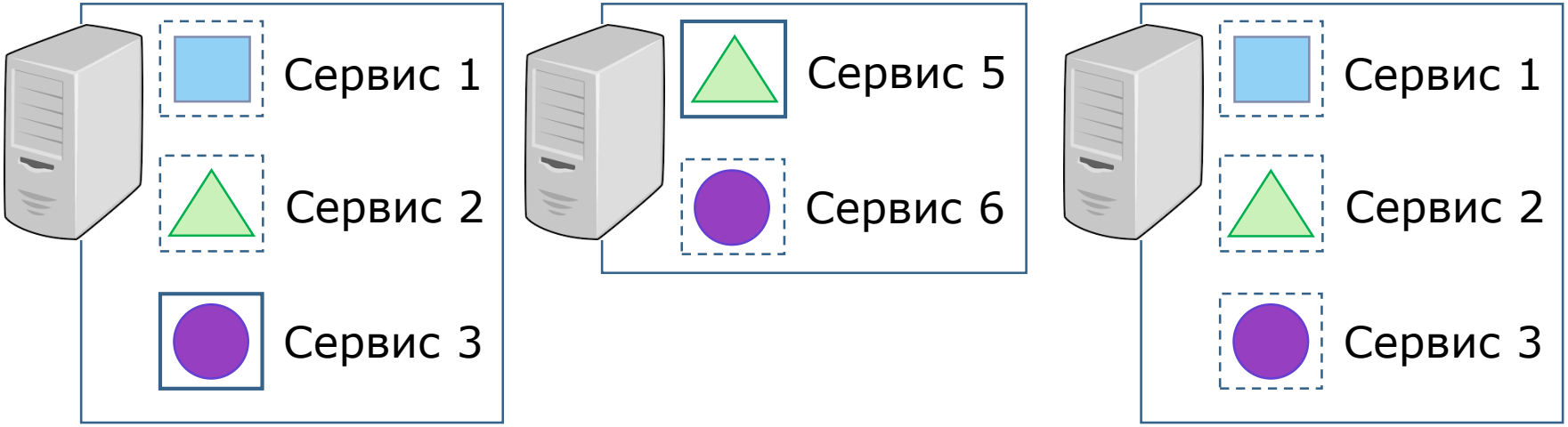
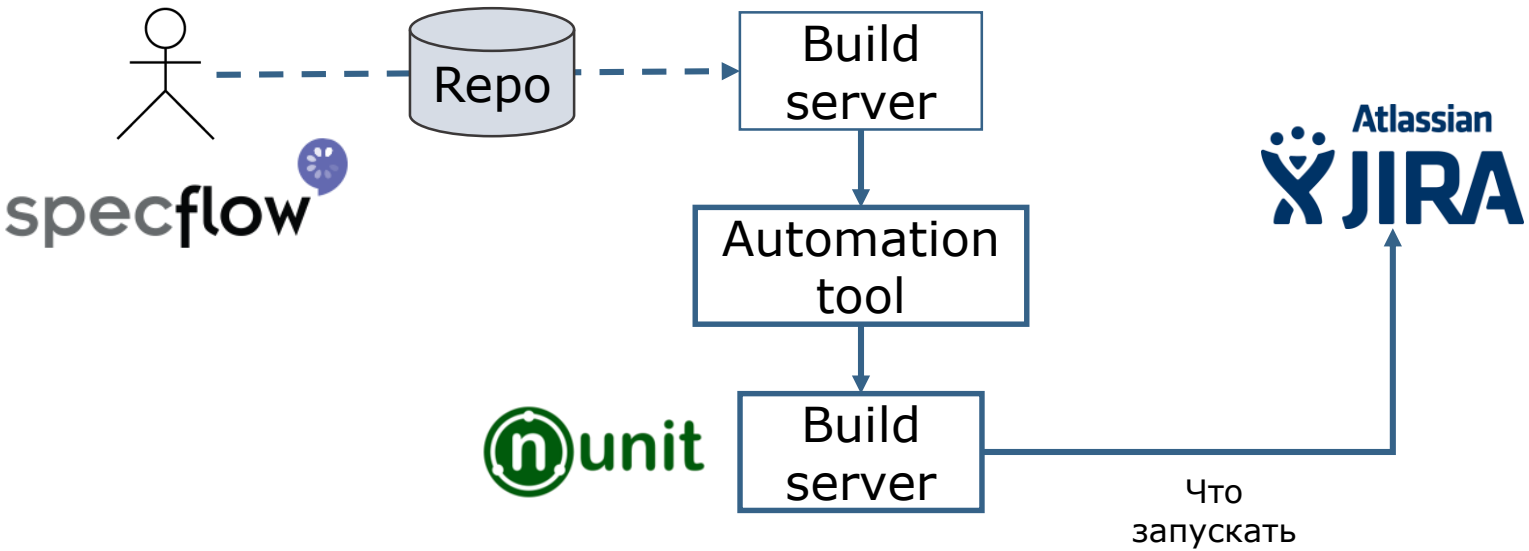


Схема тестовых серверов

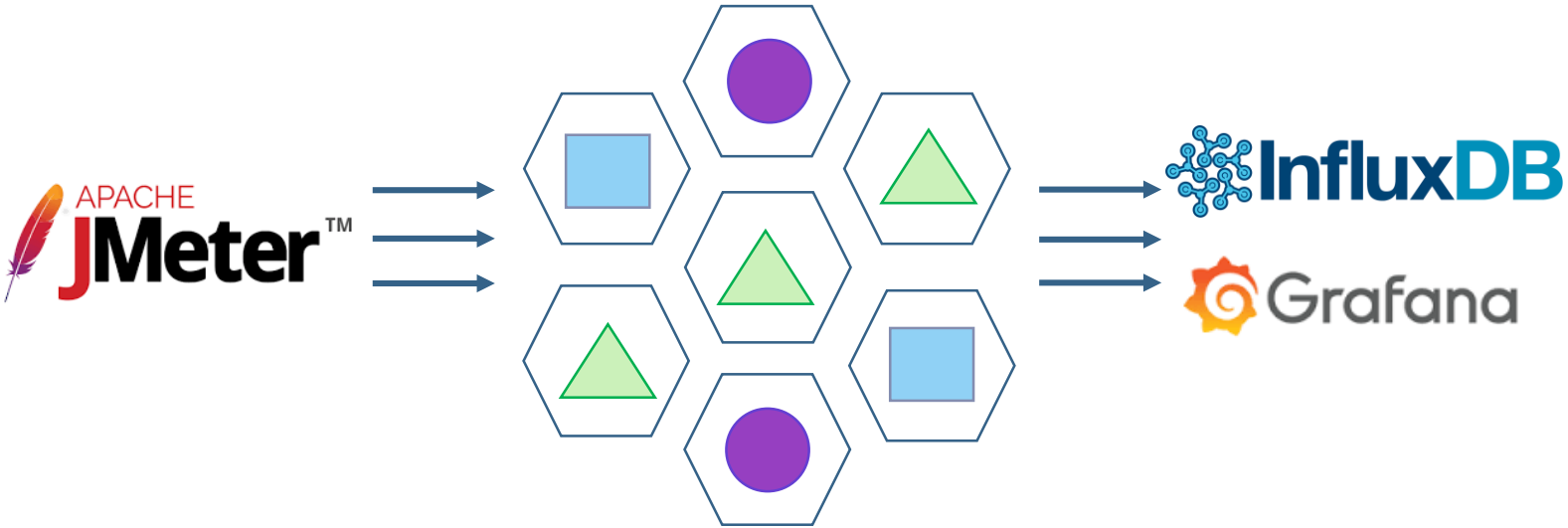


Like prod

Организация тестирования



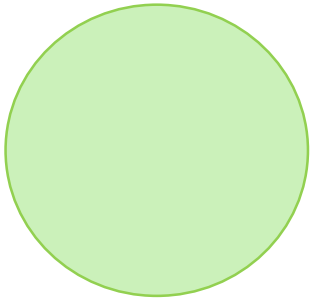
Нагрузочное тестирование



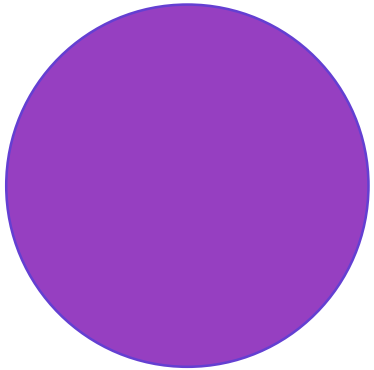
Чего мы добились?

Результаты разделения

У нас больше нет понятия «релиз»



Количество поставок



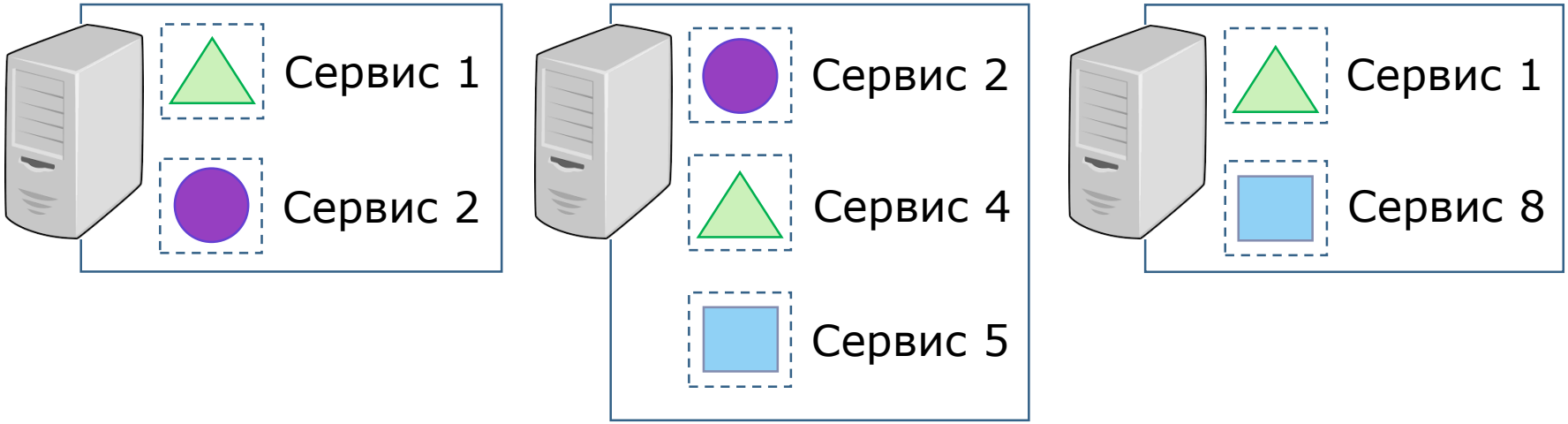
Рабочие дни

Результаты разделения

В системе нет фатальных сбоев

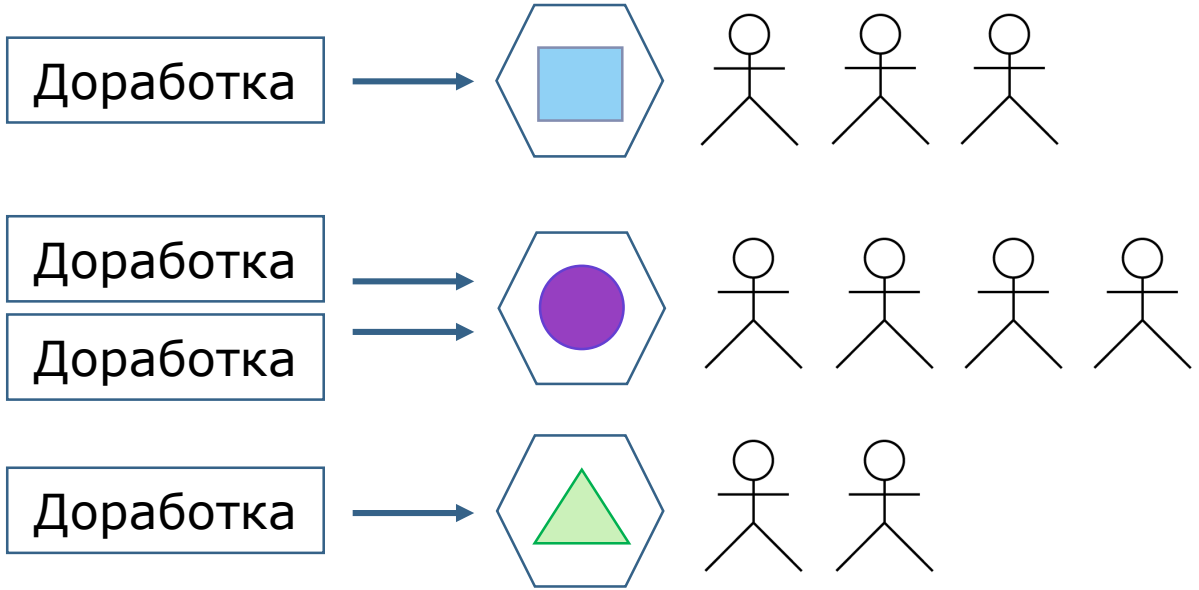
Результаты разделения

Легко управлять схемой развертывания



Результаты разделения

Независимая разработки компонентов



Резюме

- Микросервисы хорошо подходят для декомпозиции сложных систем

Резюме

- Микросервисы хорошо подходят для декомпозиции сложных систем
- Разделение на микросервисы дает не только технические, но и организационные преимущества

Резюме

- Микросервисы хорошо подходят для декомпозиции сложных систем
- Разделение на микросервисы дает не только технические, но и организационные преимущества
- Выделение микросервиса – итеративный процесс

Резюме

- Микросервисы хорошо подходят для декомпозиции сложных систем
- Разделение на микросервисы дает не только технические, но и организационные преимущества
- Выделение микросервиса – итеративный процесс
- При выделении микросервиса выгоднее переписать код, чем отделять часть legacy

Резюме

- Микросервисы хорошо подходят для декомпозиции сложных систем
- Разделение на микросервисы дает не только технические, но и организационные преимущества
- Выделение микросервиса – итеративный процесс
- При выделении микросервиса выгоднее переписать код, чем отделять часть legacy
- Затраты на поддержку могут оказаться выше, чем бонусы

Спасибо!

konst.gustov@gmail.com