Skillbox

Service Discovery & Configuration

Шаблон Chassis

Андрей Гордиенков

Solution Architect

ABAX

Skillbox

На прошлом уроке

- Управление конфигурацией
- Разбор примеров применения

На этом уроке

- Шаблон Chassis
- Решение типовых проблем

Проблема

```
Метод(параметры){
  Проверяем параметры
  Проверка прав доступа
  Логируем что-то
  Трейсинг
  Получение данных
  Логируем снова
  Полезный код метода
  Логируем снова
  Высылаем метрики
  Отдаём результат
}
```

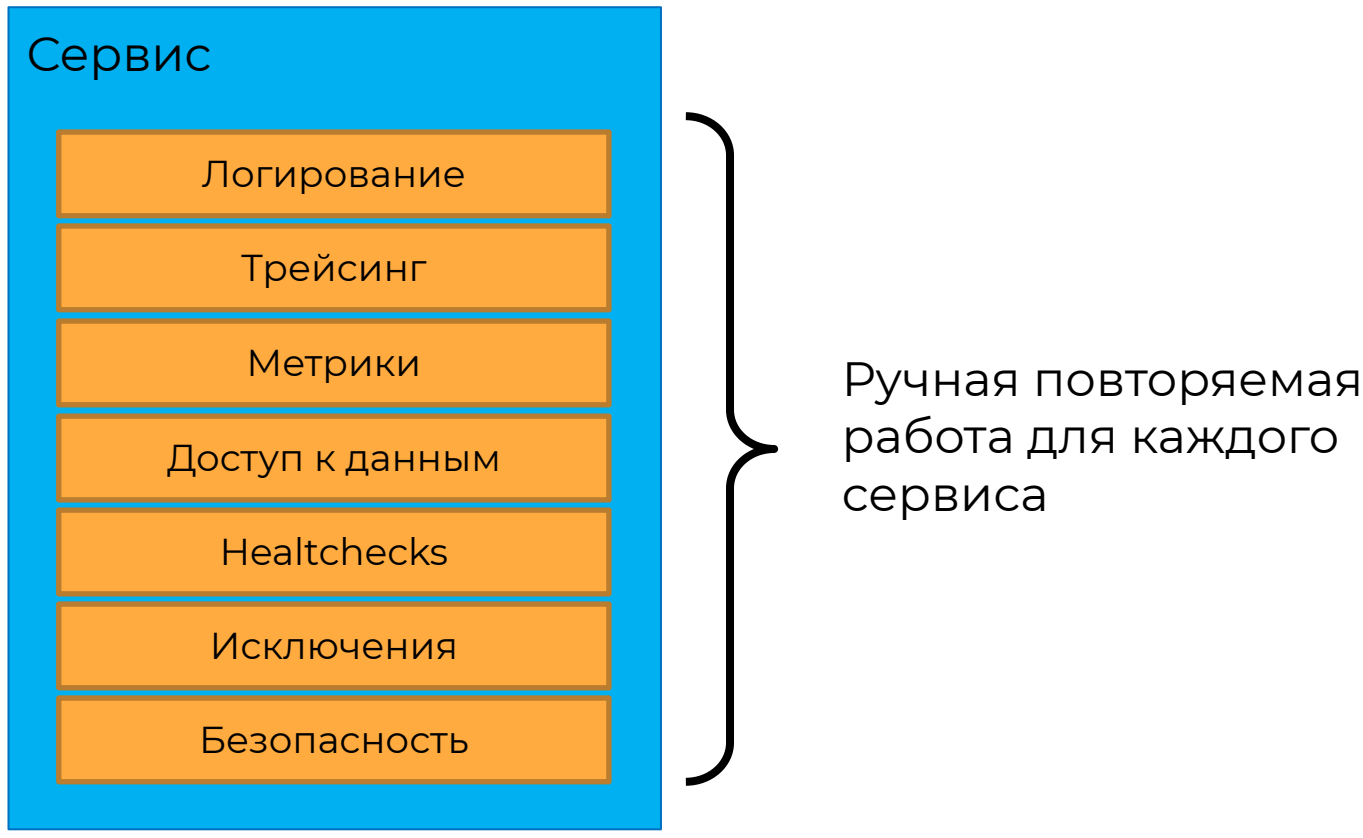
Проблема

```
Метод(параметры){
   Проверяем параметры
   Проверка прав доступа
   Логируем что-то
   Трейсинг
   Получение данных
   Логируем снова
   Полезный код метода
   Логируем снова
   Высылаем метрики
   Отдаём результат
}
```

Типичная реализация содержит много служебного кода.

Сервисы

- Добавить зависимость
- Правильно настроить
- Прописать использование



Skillbox

Шаблон Chassis

Позволяет сэкономить время и силы, используя стандартный набор уже настроенных инфраструктурных оснасток для сервиса.

Мотивация

- Создание нового сервиса должно быть быстрым и простым
- Подключение сквозных (cross-cutting) модулей должно быть лёгким:
  - логирование
  - работа с исключениями
  - перехватчики
  - безопасность
  - общие сервисы для вашего окружения

Решение

Использовать (или создать и использовать) легковесный фреймворк для конфигурации служебного сквозного кода.

Можно ограничиться общим загрузочным модулем.

Пример

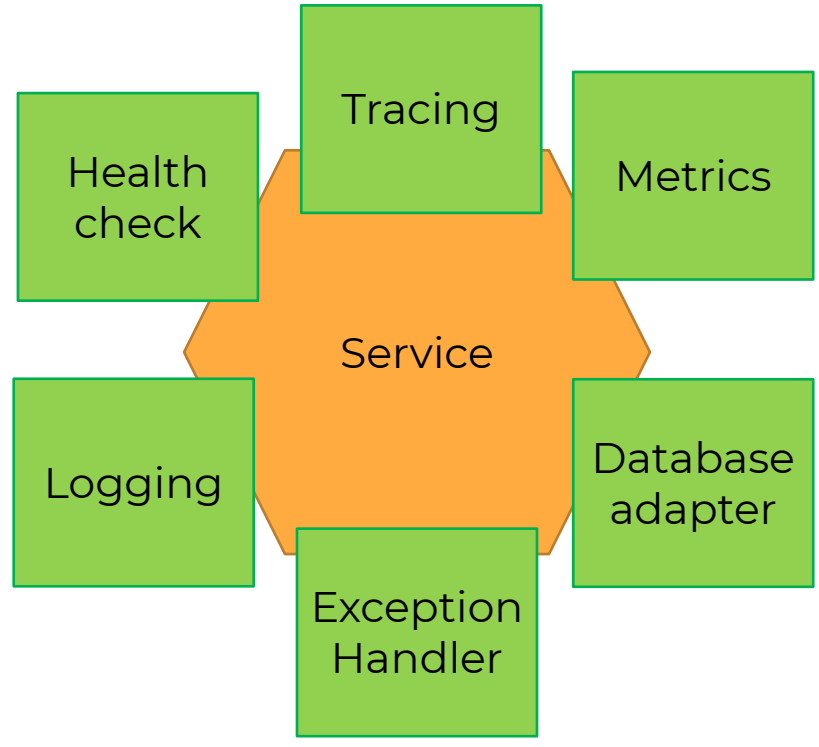
```
services.AddMessageProcessor(
    (sp, builder) ⇒ builder
        .Client(
            KafkaBrokerClient.Builder
                .Configure(Configuration)
                .Create()
        )
        .Topics( ... )
        .Module(new SomeModule( ... ))
        .StopOnUserException()
        .ProcessorName(ApplicationKey)
        .SubscriptionName( ... )
        .BatchingStrategy(BatchingStrategy.None)
        .Buffer( ... )
        .UseDatadogTracing(TracePriority.Normal, "kafka", ApplicationKey)
);

services.AddHealthChecks(Configuration);
services.AddInfrastructure(Configuration);
services.AddLogging(Configuration);
```

Skillbox

Шаблон Chassis

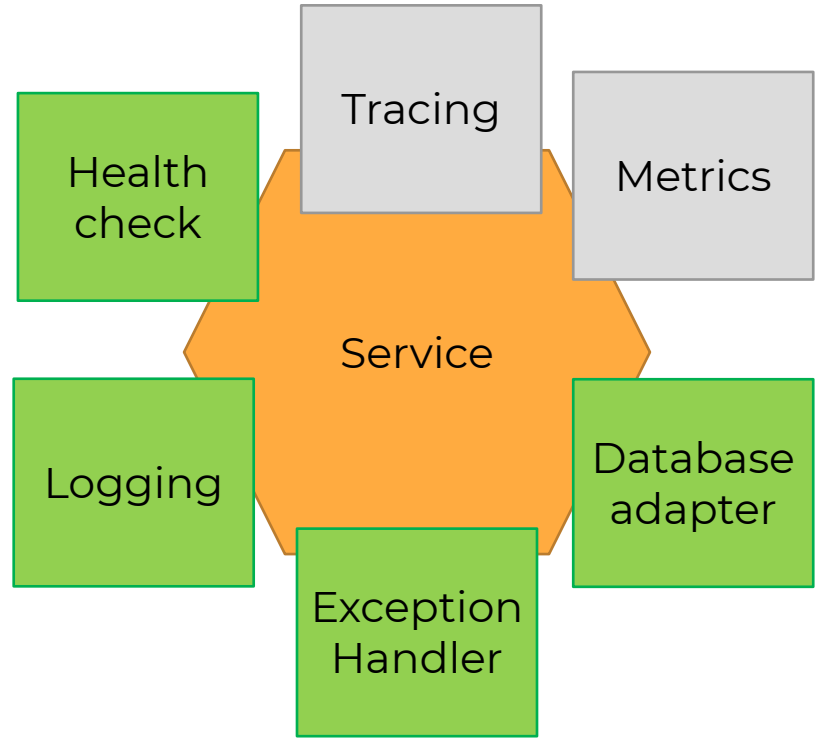
Слабосвязанный конструктор модулей, не связанный с бизнес-логикой:



Skillbox

Шаблон Chassis

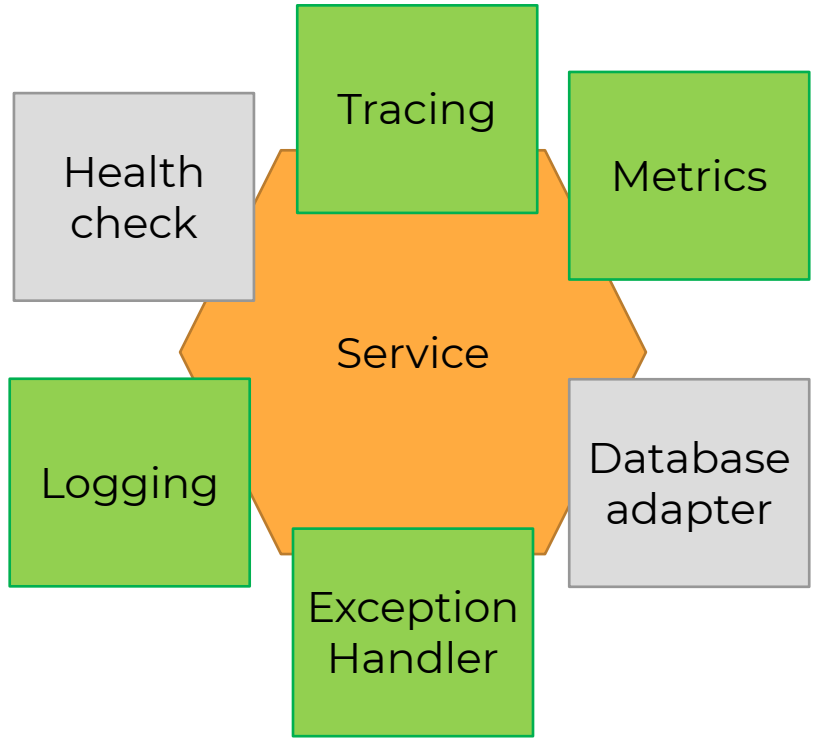
Слабосвязанный конструктор модулей, не связанный с бизнес-логикой:



Skillbox

Шаблон Chassis

Слабосвязанный конструктор модулей, не связанный с бизнес-логикой:



Шаблон Chassis — это «конструктор», упрощающий начальную конфигурацию сервисов.

Что выделять?

Инфраструктурные составляющие вашего кода, которые повторяются в большинстве сервисов, чаще всего это настройка.

Чего опасаться:

- делать методы «Подключи всё (конфиг)»
- делать реализацию слишком умной, включая туда автоматические методы
- делать конфиг-файлы слишком гибкими, так что составление такого файла уже можно назвать программированием

Плюсы

- Скорость и удобство создания сервисов с нуля
- Не надо снова и снова писать одни и те же тесты интеграции для базовых вещей

Минусы

- Модули для шаблона надо написать/выбрать
- Модули надо поддерживать
- Модули ориентированы на типовые сценарии использования

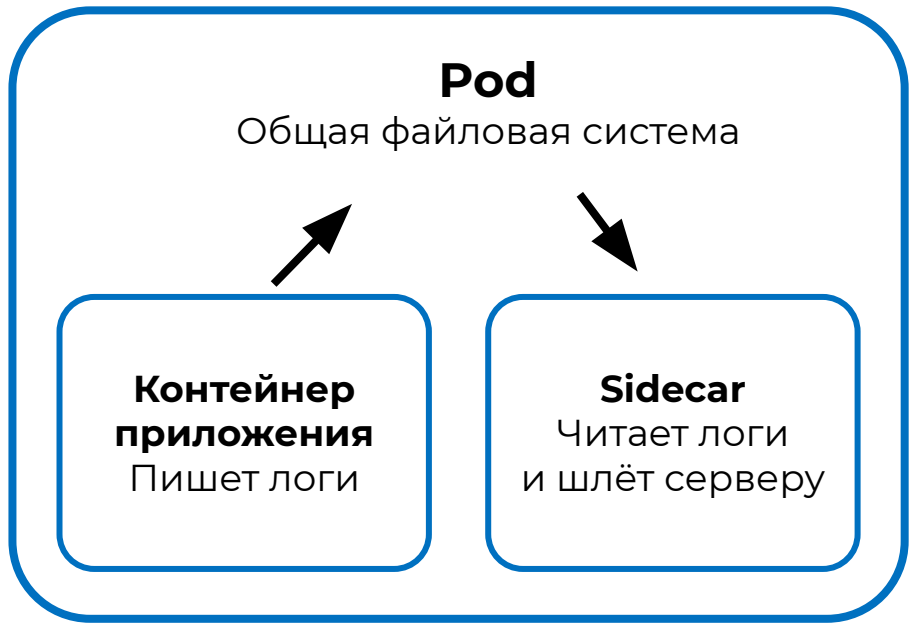
Skillbox

Шаблон Sidecar

У приложения есть «внешний» помощник, которому делегируется задача, не связанная с основной бизнес-логикой.

Примеры применения:

- утилиты логирования
- агенты мониторинга
- сервисы синхронизации
- «смотрители»

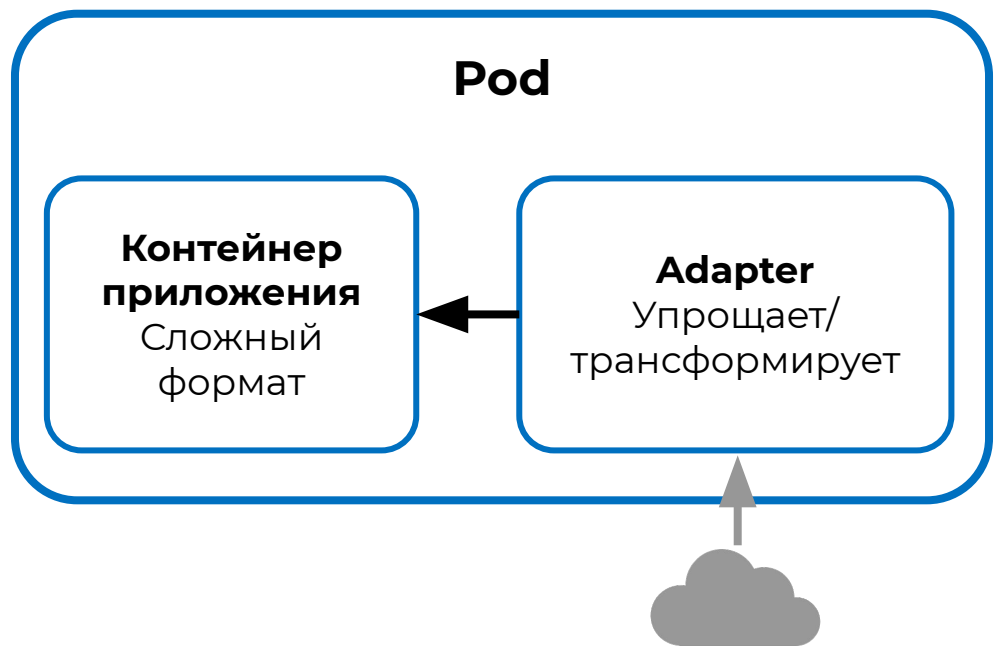


Skillbox

Шаблон Adapter

Шаблон Adapter в данном контексте нормализирует
и стандартизирует телеметрию приложения в зависимости
от потребителя.

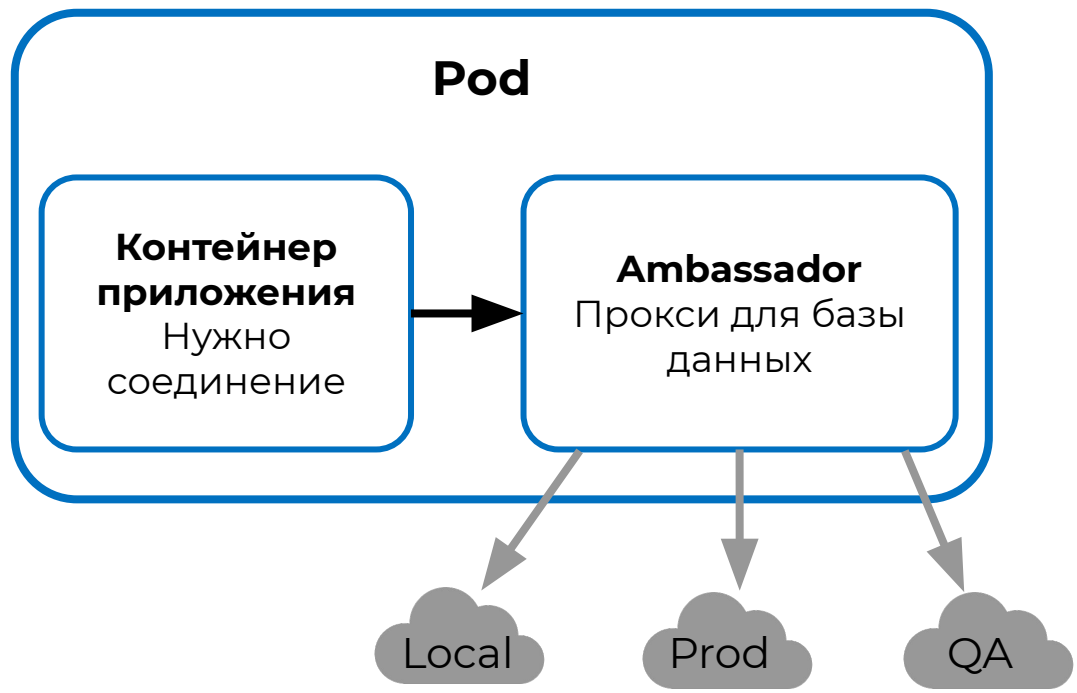
Не требуется менять внутренний код приложения или сложность
его настройки при смене потребителя.



Шаблон Ambassador

Шаблон Ambassador является посредником между приложением и внешними сервисами.

Приложение всегда работает с localhost, что упрощает разработку. Окружение деплоя будет подкладывать нужного представителя во время создания сервиса.



Chassis vs Sidecar

Открытость к настройке и изменениям в самом приложении	Закрытость реализации, можно пользоваться только открытым API
Функции дополнений вызываются приложением и контролируются им же	Относительная независимость функционирования

Выводы

- Шаблон Chassis может сэкономить время и силы для создания новых сервисов при соблюдении высокой гранулярности компонентов шаблона
- Не пишите фреймворк/bootstrap заранее, сделайте его на основе 4–5 уже созданных сервисов
- Для микросервисной архитектуры существует шаблон Sidecar, а для решения похожих проблем используются его вариации

Выводы модуля

- Процесс ввода и вывода сервисов из работы становится нетривиальной задачей, если счёт сервисов пошёл на десятки. Service Discovery позволяет решить эту задачу
- Контроль конфигураций сервисов можно осуществлять централизованно
- Контроль конфигураций — это не только строки подключений, но и возможность гибко управлять предоставляемым функционалом
- Активная разработка новых сервисов может быть облегчена и стандартизирована с помощью шаблона Chassis

Skillbox

Спасибо за внимание!