Skillbox

API Gateway

Андрей Гордиенков

Solution Architect

ABAX

Skillbox

Типы API Gateway: как выбрать, антипаттерны

Андрей Гордиенков
Solution Architect
ABAX

Skillbox

В прошлом модуле

Функциональные части API Gateway.

В этом модуле узнаем

- Типы API Gateway
- Как выбрать подходящий продукт
- Антипаттерны

Skillbox

Таксономия API Gateway

В отрасли применяют различные требования к API Gateway, поэтому выделяют несколько общих типов применения:

- традиционные шлюзы уровня предприятия
- мини- и микросервисные шлюзы
- Шлюзы Service Mesh

Традиционные шлюзы

- Ориентированы на управление бизнес-ориентированного API
- Строгая модель управления жизненным циклом API: релиз, поддержка, монетизация на большом масштабе сервисов
- Большая часть решений коммерческая
- Зависимость от собственного хранилища данных
- Требуют высокой доступности платформы для работы

Skillbox

Микросервисные шлюзы

- Основная ответственность — управление входящим трафиком к бэкенду
- Слабая поддержка управления жизненным циклом API
- Обычно бесплатные решения
- Устанавливаются как отдельный компонент и используют инфраструктуру для управления

Шлюз Service Mesh

- Обычно ответственны только за перенаправление трафика к Service Mesh
- Тесно связан с сетью сервисов и использует её возможности для управления перенаправлением

Skillbox

Сравнение типов таксономий

Основное предназначение

- **Традиционный API Gateway**

  Предоставление доступа, композиция и управление внутренним бизнесом API

- **Микросервисный API Gateway**

  Предоставление доступа и слежение за внутренними сервисами

- **Service Mesh Gateway**

  Предоставление доступа к внутренним сервисам сети

Сравнение типов таксономий

Публикация API

- **Традиционный API Gateway**

  Команда управления API использует специальное административное приложение для регистрации и обновления доступных сервисов и их API

- **Микросервисный API Gateway**

  Команда поддержки API использует декларативное описание API, и оно является частью процесса разворачивания новых сервисов или их обновления

- **Service Mesh Gateway**

  Команда обслуживания обновляет сеть сервисов с помощью декларативного описания во время процесса разворачивания

Сравнение типов таксономий

Мониторинг API

- **Традиционный API Gateway**

  Ориентирован на администраторов и следит за высокоуровневыми метриками для бизнеса (количество вызовов API по клиентам, отчёты по ошибкам API и т. д.

- **Микросервисный API Gateway**

  Ориентирован на разработчиков, т. е. в фокусе latency, ошибки трафика, затухание сервисов

- **Service Mesh Gateway**

  Основной акцент на метриках платформы: утилизация ресурсов, затухание компонентов, ошибки

Сравнение типов таксономий

Отладка системы

- **Традиционный API Gateway**

  Запуск шлюза с дополнительным логированием, апробация решения на staging окружении

- **Микросервисный API Gateway**

  Используется более детальный мониторинг, подключается «теневой трафик» и/или «канареечные» релизы для проверки проблемы

- **Service Mesh Gateway**

  Используется более детальный мониторинг, отладка производится с помощью специализированных инструментов (Squash, Telepresence)

Сравнение типов таксономий

Тестирование

- **Традиционный API Gateway**

  Набор окружений: QA, Staging, Prod. Автоматические интеграционные тесты, многоступенчатый процесс одобрений при деплое API. Используется клиентоориентированное версионирование для стабильности и совместимости (SemVer)

- **Микросервисный API Gateway**

  «Канареечные» релизы для динамического тестирования. Версионирование ориентировано на разработчиков для управления обновлениями

- **Service Mesh Gateway**

  «Канареечные» релизы для динамического тестирования

Сравнение типов таксономий

Возможности локальной разработки

- **Традиционный API Gateway**

  Локальное разворачивание (Vagrant, Docker) для имитации «боевого» окружения, используются фреймворки для мокирования шлюзов

- **Микросервисный API Gateway**

  Локальное разворачивание на основе платформы оркестрации (Kubernetes)

- **Service Mesh Gateway**

  Разворачивание сети сервисов на основе платформы оркестрации (Kubernetes)

Антипаттерны

Смешивание средств для управления трафиком.

API Gateway имеет тенденцию слишком тесно интегрироваться с бэкенд-сервисами.

- API Gateway Loopback: «Service Mesh Lite»
- API Gateway в роли ESB
- Бесконечные API Gateways

Skillbox

API Gateway Loopback: «Service Mesh Lite»

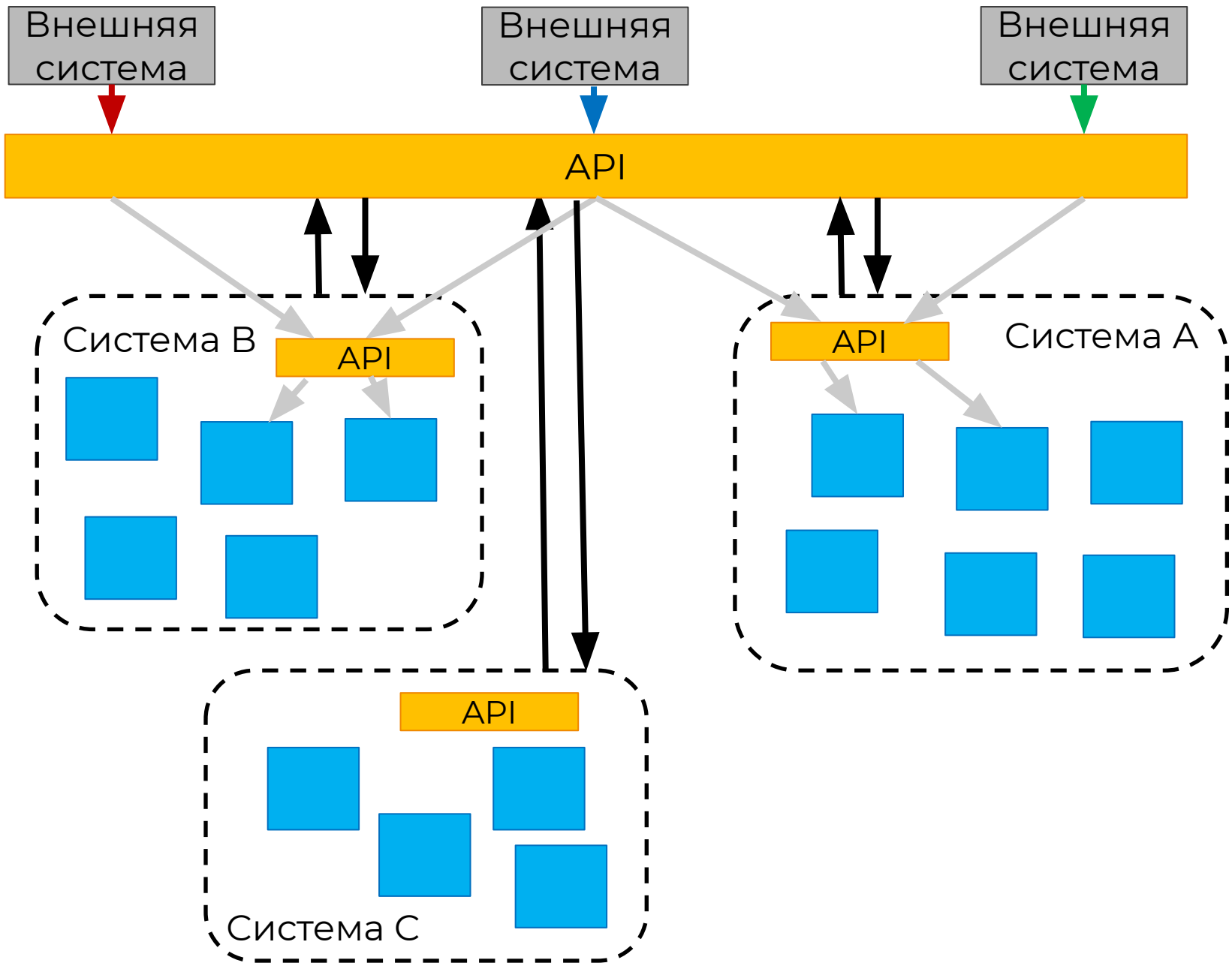
Контекст:

малое количество сервисов и один API Gateway отвечает за обнаружение сервисов и обработки входящего и исходящего трафика.

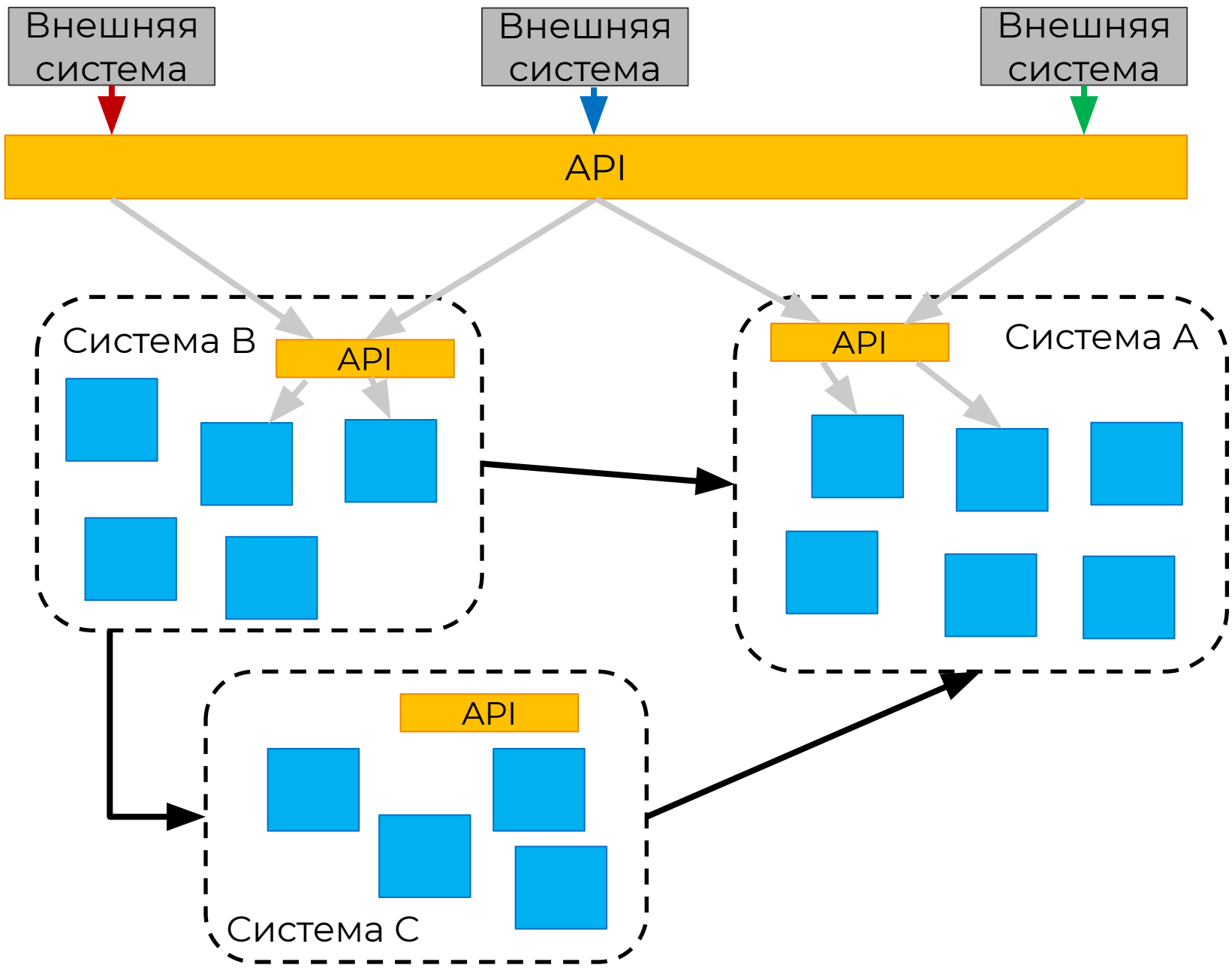
Проблемы:

- с ростом количества сервисов трафик между сервисами будет создавать излишнюю нагрузку на API Gateway (безопасность, скорость, сам трафик стоит денег)
- «бутылочное горлышко» и единая точка отказа, ставящая под угрозу работу всей системы разом

Антипаттерн



Как надо


Внешняя система
Внешняя система
Внешняя система
API
Система B
API
API
Система A
API
Система C

Skillbox

API Gateway в роли ESB

Контекст:

почти все современные API Gateway поддерживают расширение функционала с помощью плагинов.

Проблемы:

- велик соблазн дописать свои плагины с внедрением бизнес-логики для работы с трафиком/сообщениями. Это ведёт к «хрупкости» решения и может создавать эффект «кругов на воде» при изменении конфигурации
- сложность с обновлениями, потому что надо учитывать множество факторов

Skillbox

Бесконечные API Gateways

Контекст:

API Gateways создаются для каждой системы, департамента, типа трафика и взаимодействия. Например, шлюз безопасности трафика, авторизации, логирования, департамента А, департамента Б.

Проблемы:

- усилия для изменения шлюзов и координации несоразмерны пользе от каждого такого шлюза
- сложность распределения ответственности
- проблема производительности

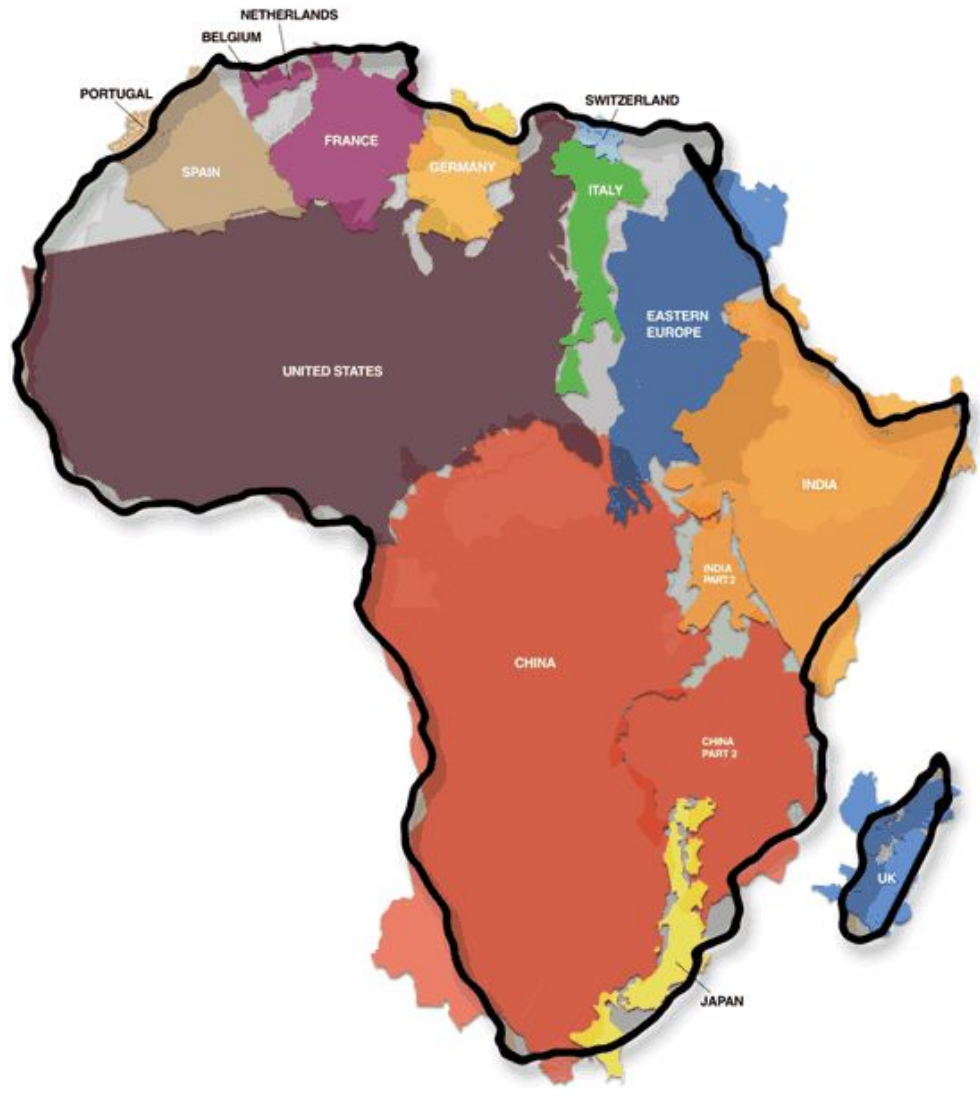
Skillbox

Как выбрать API Gateway?

Осознать собственные требования к API Gateway.

Сбивают с пути:

- новые технологии с серебряными пулями
- магия маркетинга
- продажная документация



Требования

- Уменьшение связи между фронтендом и бэкендом
- Упрощение использования API клиентами за счёт объединения и/или трансляции протоколов сервисов на бэкенде
- Защита API от чрезмерного использования и злоупотреблений с помощью обнаружения и устранения угроз
- Понимание того, как потребляются API и как работают базовые системы
- Управление API как продуктами, т. е. требования к управлению жизненным циклом API
- Монетизация API, включая потребности в управлении учётными записями, выставлении счетов и оплате

Знайте свои ограничения

Команда

- Знания команды, организация, динамика
- Закон Конвея

Технологии

- Что используется сейчас?
- Какие есть возможности миграции на новые технологии?
- Каков план поддержки выбранной технологии?

План развития

- Некоторые технологии и сервисы могут быть запрещены по разным соображениям

Купить или сделать?

- Цена владения
- Возможность реализации необходимого функционала на хорошем уровне
- Знание рынка поставщиков

Что обсудить?

- Знаем ли мы все наши требования, связанные с выбором API-шлюза, и расставили ли мы приоритеты?
- Знаем ли мы текущие технологические решения, которые были развёрнуты в этой области в организации?
- Знаем ли мы все наши командные и организационные ограничения?
- Изучили ли мы наш план развития организации в связи с этим решением?
- Честно ли мы подсчитали затраты «строить или покупать»?
- Изучили ли мы текущий технологический ландшафт и знаем ли мы обо всех доступных решениях?
- Проконсультировались ли мы со всеми заинтересованными сторонами и проинформировали ли их об анализе и принятии решения?

Выводы

- API-шлюзы бывают разных типов. Нет одного подходящего всем
- При внедрении API-шлюза есть ловушки и надо о них помнить
- Выбирать надо на основе реальных потребностей, а не обещаний маркетологов

Выводы модуля

- API Gateway призван упростить доступ к внутренним сервисам и дать контекст использования
- API Gateway может многое, но надо осознанно подходить к делегируемой ответственности
- При выборе учитывать таксономию API Gateway, ограничения, связанные с командой, технологиями и планами развития организации

Skillbox

Спасибо за внимание!