

1. Покажите на архитектурной диаграмме, где будут располагаться API Gateways. Обоснуйте свое решение.
   1. Светло-зеленым показаны апи для сервисов. Они отвечают за предоставление-изменения данных в рамка одного конкретного сервиса. «Склад», «Логистика» и «Производство» - сокращены и включают в себя группу сервисов, устройств и иже с ними. Каждый из них может рассматриваться как отдельная система и вопрос интеграции будет открыт весьма широко. Условно примем, что все эти подсистемы находятся в рамках одной локальной сети. Так же за аксиому считается, что всякие сканеры не нужны в остальной сети вне своей подсети (датчики с производства на складе смотреть не надо). Все необходимые данные даются и запрашиваются через соответствующие api.
   2. Коротки пунктирные линии сокращают связь «все ко всем», чтобы не перегружать стрелочками схему.
   3. Под «менеджером» следует понимать любое должностное лицо фирмы с соответствующими правами. Это может быть продажник, видящий личные данные клиентов, или администратор, видящий нагрузку на сервисы в режиме реального времени и имеющий инструменты удаленно их перегрузить.
   4. Основных API Gateway предлагаю 4:
      1. Api Доступа, отвечающий за взаимодействие внешней и внутренней сети. Его задачи – фильтровать траффик, отсекать непонятные запросы, намекать юзерам, что не надо по 100500 раз запрашивать данные, меня циферки в адресной строке браузера.
      2. Api для удалённых сервисов. Хоть и предположили, что все наше находится у нас в одной локальной сети, но высоковероятно появление «удаленного склада», «представительства в другом конце города» и т.д. Для интеграции безопасным образом удаленной и основной внутренних сетей нужна эта api. Что-то вроде безопасного proxy. Основная задача та же – обеспечение безопасности, отсечение нежелательной и непонятной активности.
      3. Api доступа соответствующей сущности (3) Менеджер и (4) Клиент. Данные для менеджера и клиента очень разнятся, ровно, как и права доступа, а также необходимая пропускная способность и требования к защите. Разделение позволит четко отделить мух от котлет, а также повысить безопасность (достаточно сменить формат запросов и сразу будет видно откуда запрос пришёл, а откуда бы он ДОЛЖЕН БЫЛ прийти.)
         1. Дистрибьютор – просто очень большой клиент. Ему можно красивую галочку в личном кабинете нарисовать, но технически он всё та же сущность «Клиент».
2. Для всех добавленных API Gateway опишите, какой функционал они будут предоставлять и по возможности представьте качественные характеристики этого функционала (то есть с цифрами).
   1. Api для взаимодействия с внешней средой (номера 1 и 2).
      1. Защита. Брутфорс, попытка залезть куда не надо попытка перегрузить основные сервисы кучей запросов, залезть посередине. Два запроса без учетки не на авторизацию за 5 минут – блок по связке юзерагент-айпи на 5 минут. Алярм админам, что кто-то исследует то, что не нужно.
      2. В (2) – проверка аутоинтефикации, авторизации. Это канал технический, все запросы четко известны заранее. Любое отклонение – блокировка и алярм, что что-то в системе завелось вредное. Так же не всем сервисам можно лезть куда угодно удаленно. Например, складу не нужно лезть в финансовые операции, ему нужен только список Продктов к выдаче.
   2. Api для доступа сущностей
      1. Проверка аутентификации, авторизации.
         1. Тройной провал ввода пароля для (3) за 5 минут – блокировка учетки и разблокировать – только через админов. Тройной запрос на запрещенные по правам страницы за 5 минут – блокировка учетки. Для (4) то же самое, только разблокировка учетки самостоятельно через 5 минут.
      2. Предоставление данных от внутренних сервисов по соответствующим запросам согласно правам доступа. Попытка 3 запроса без прав – блок до звонка админам (3) или на 5 минут (4). Попытка 10 запросов за 10 секунд – блок или бан.
      3. Передача запроса на изменение данных соответствующим внутренним сервисам. Попытка 2 запросов без прав – блок до звонка админам (3) или на 5 минут (4). Попытка 10 запросов за 10 секунд – блок или бан.

Получается система api, предоставляющая и контролирующая доступ извне, при этом четко отделяющая функцию защиты от функции обмена данными.

Можно масштабировать как ввширь (балансировщики трафика), так и ввысь (выделение больше ресурсов). Желательно иметь хотя бы 2-3 экземпляра каждого на форсмажоры, но кто ж денег и мощей даст…

Нагрузка распределяется более равномерно, и внезапная смерть одной из api не убьет всю систему, а только ту часть, за которую отвечает. В частности, даже при смерти (1) – сайт просто не сможет предоставлять актуальные Продукты и совершать сделки, кеш никто не отменял, как и локальные БД. При смерти (2) – не будет только интеграции подсистем, но работать-то они будут. Как восстановится связь – данные согласуются.