

Skillbox

Архитектурные стили

**Заблуждения
о распределённых
вычислениях**

Павел Елисеев

Software Architect
Сбер

На прошлом занятии

- Ознакомились с архитектурой текущего решения.
- Поняли, что нужно двигаться в сторону распределённой архитектуры.

На этом занятии

- Поговорим о проблемах в распределённых системах на примере труда Fallacies of distributed computing (1994) Питера Дойча и других его коллег из Sun Microsystems.

Заблуждения

1. Сеть надёжна.
2. В Сети нет задержек.
3. Пропускная способность бесконечна.
4. Сеть безопасна.
5. Топология никогда не меняется.
6. Администратор всегда один.
7. Цена передачи данных — нулевая.
8. Сеть однородна.

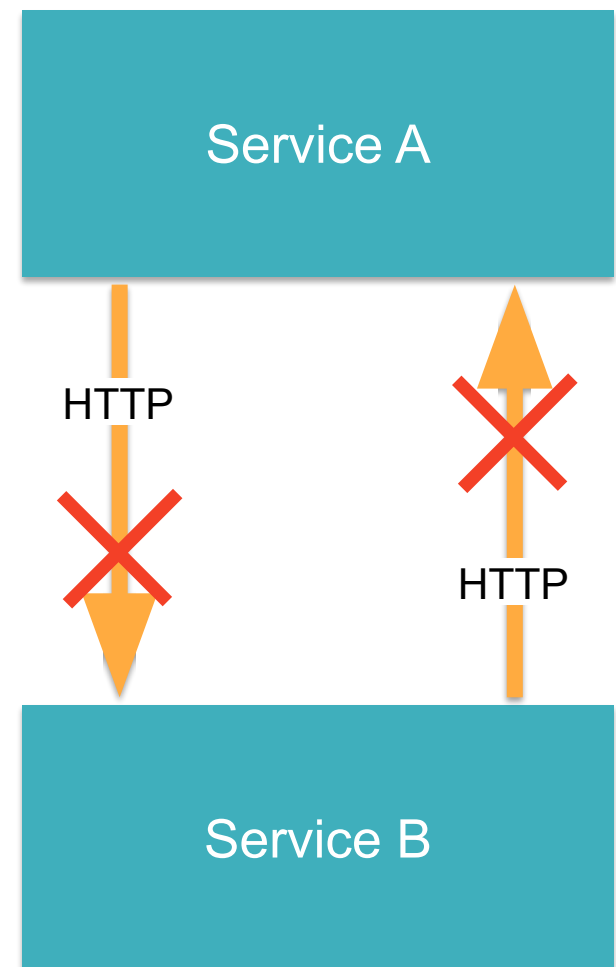


Сеть надёжна



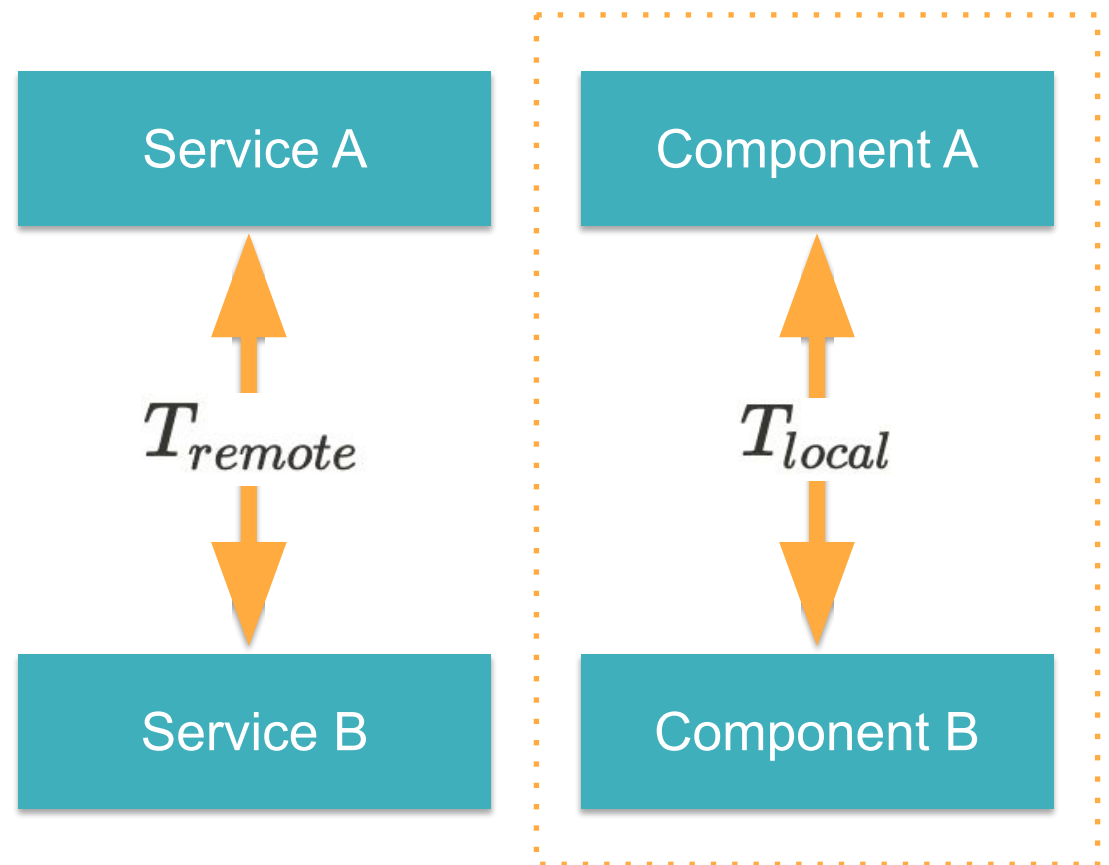
Сеть надёжна

- Все распределённые приложения общаются по Сети
- Сеть может отказать
- Нужно проектировать систему на отказ Сети:
 - Circuit Breaker
 - Health-check
 - Timeouts



В Сети нет задержек

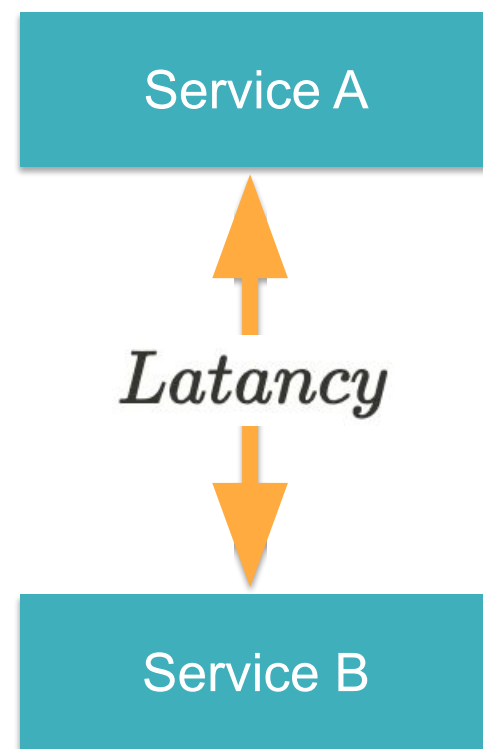
- Вызовы внутри одного приложения всегда быстрее удалённых вызовов
- Latency — одна из важнейших метрик в распределённой архитектуре



$$\underline{T_{remote} > T_{local}}$$

В Сети нет задержек

- Среднее значение latency между сервисами равно 200 миллисекунд
- Клиентский запрос проходит 10 сервисов
- $10 * 200 = 2\,000$ миллисекунд, или 2 секунды
- Стоит обращать внимание на 95-й и 99-й перцентили
- Если 95-й перцентиль = 3 000 миллисекунд, то 5% клиентских запросов обрабатываются за 30 секунд

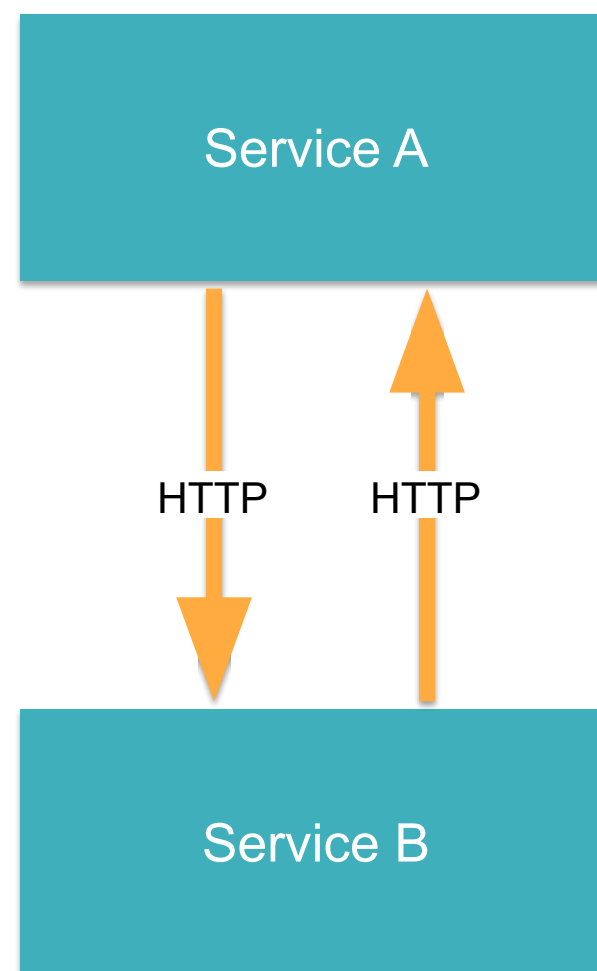


$$\overline{Latancy} = 200ms$$

$$Latancy_{(95\%)} = 3000ms$$

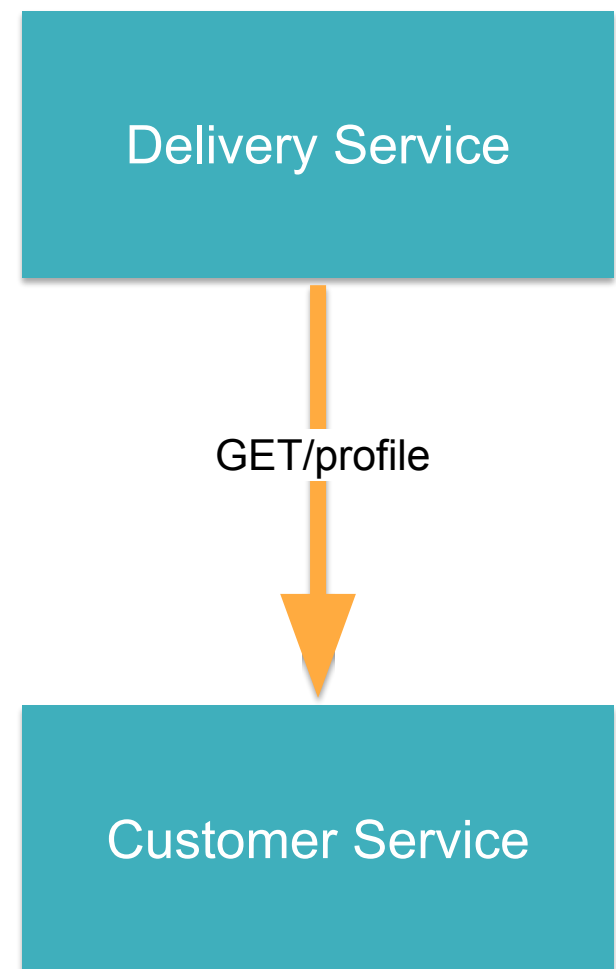
Пропускная способность бесконечна

- У Сети есть ограничения по пропускной способности
- Излишнее расходование пропускной способности приводит к ошибкам и увеличению latency



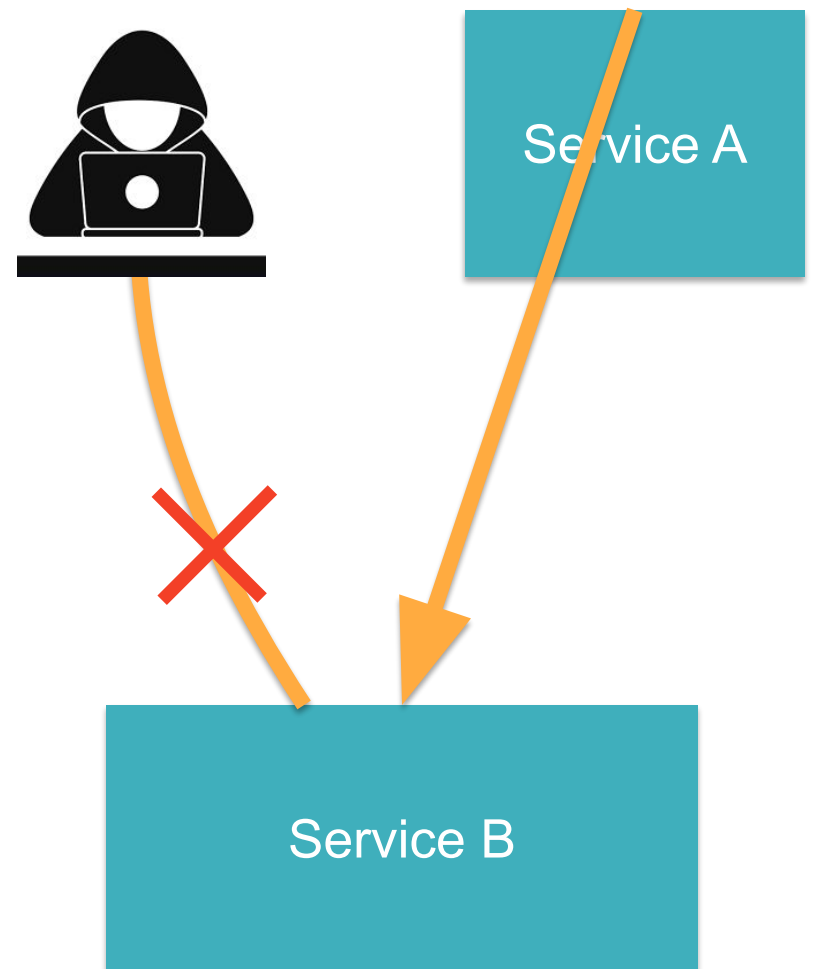
Пропускная способность бесконечна

- Сервису доставки нужен адрес клиента (100 byte)
- Customer Service возвращает весь профиль клиента (500 Kb)
- Интенсивность запросов — 1 000 RPS
- За одну секунду система расходует 0,5 Gb пропускной способности Сети



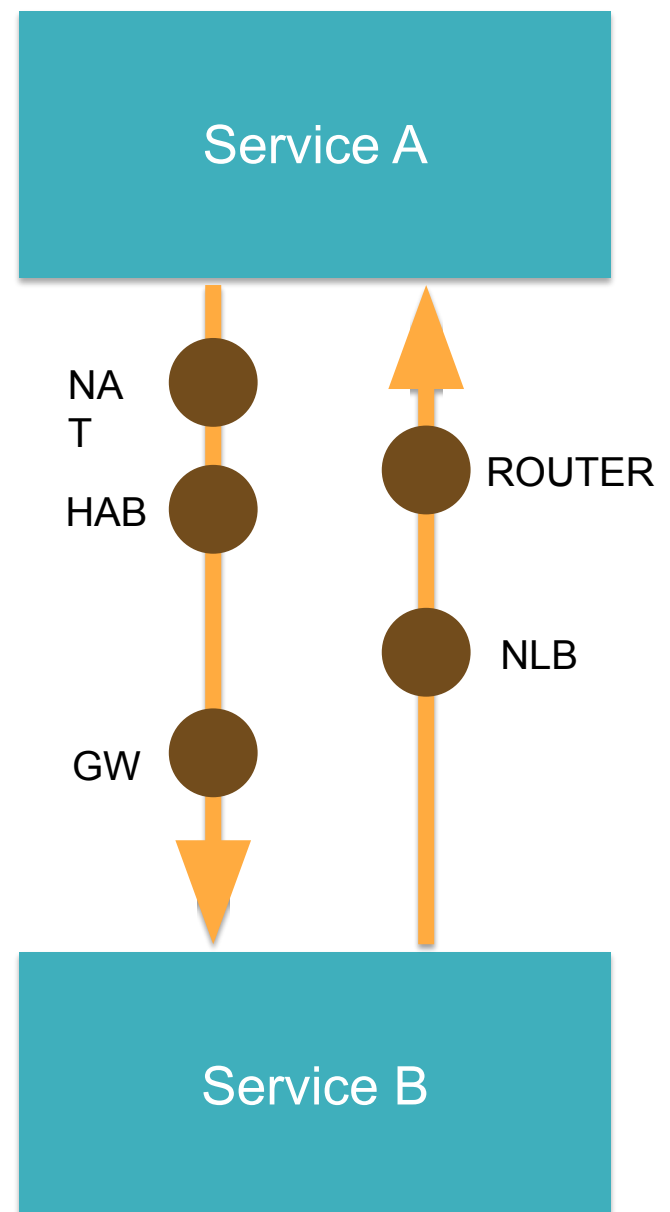
Сеть безопасна

- Сетевой трафик может быть перехвачен
- Без должной защиты доступ к сервисам имеет кто угодно
- Всегда есть компромисс между безопасностью и производительностью/удобством



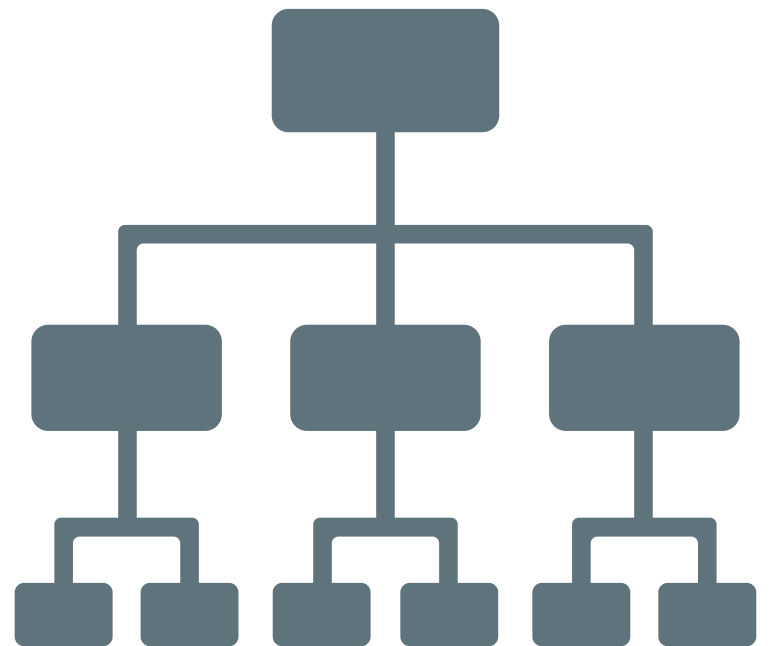
Топология никогда не меняется

- Между сервисом А и сервисом Б находится множество сетевых устройств
- Топология Сети постоянно меняется
- Архитектору нужно поддерживать связь с инфраструктурами и отделами, в частности с сетевыми администраторами



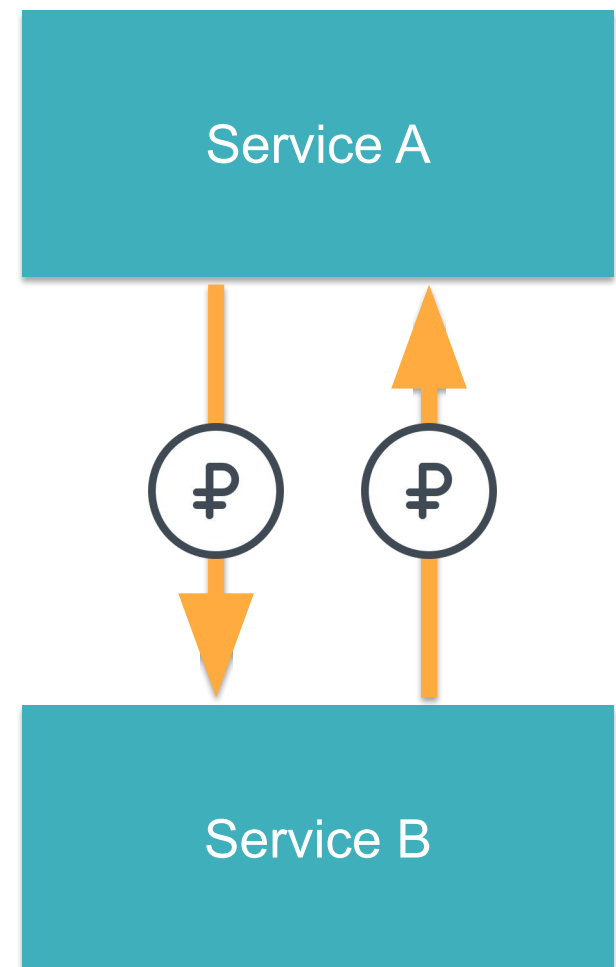
Администратор всегда один

- В больших компаниях много администраторов и отделов
- Когда проблема на стыке отделов, решить её сложно
- Сложность распределённых систем состоит в большом количестве коммуникаций



Цена передачи данных — нулевая

- Коммуникация по Сети всегда дороже локальной коммуникации
- «Дороже» — это про деньги
- Требуются дополнительные расходы на:
 - железо
 - серверы
 - шлюзы
 - балансировщики и прочее
- Cloud (IaaS) и Serverless помогают сэкономить



Сеть однородна

- В Сети работают разные устройства
- У этих устройств могут быть разные операционные системы



Выводы

- Рассмотрели эволюцию архитектуры.
- Выяснили, какие подходы можно использовать для построения монолитных приложений.
- Рассмотрели архитектуру текущего решения.
- Выяснили, какие существуют заблуждения касательно распределённых вычислений.

В следующем модуле

- Рассмотрим, какие существуют подходы для построения системы в распределённом стиле, и выберем один из них.