

# Что такое service mesh простыми словами / Хабр

 [habr.com/ru/companies/slurm/articles/705914](https://habr.com/ru/companies/slurm/articles/705914)

## Что такое service mesh простыми словами

Всё больше компаний переходят на микросервисы. Такой выбор вполне оправдан: при должной реализации они решают множество проблем монолита. За последние несколько лет микросервисная архитектура сильно эволюционировала и обросла вспомогательными технологиями, одна из которых service mesh. В статье разберём, какую роль service mesh играет в развёртываниях микросервисов и как помогает упростить работу разработчиков.

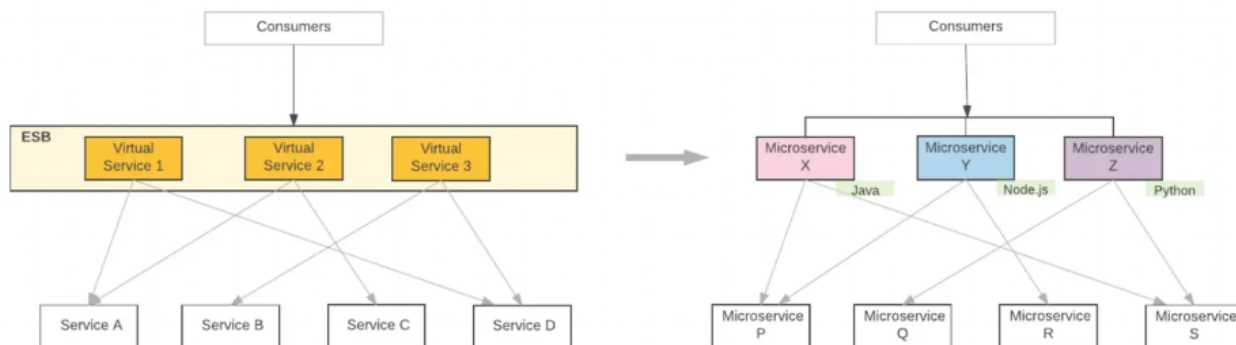


## Зачем нужен service mesh

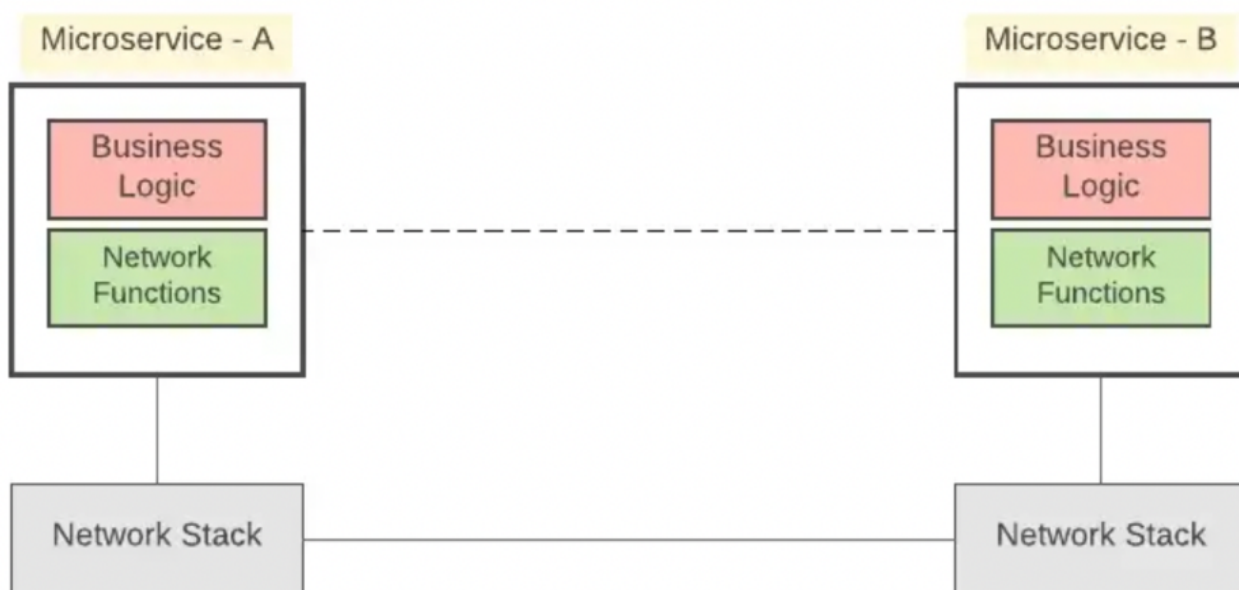
Service mesh — инструмент, позволяющий контролировать, как разные части приложения обмениваются данными между собой. В отличие от других систем управления, он представляет собой отдельный слой инфраструктуры, встроенный прямо в приложение. Этот слой документирует, насколько хорошо взаимодействуют микросервисы, поэтому становится проще оптимизировать это взаимодействие и избегать простоев по мере роста приложения.

Многие думают, что микросервисы уже являются решением всех проблем, которые возникали с монолитом. Однако, наблюдая за реализацией микросервисной архитектуры в реальном мире, мы обнаруживаем, что большинство функций, поддерживаемых централизованной шиной (ESB), теперь реализованы на уровне микросервисов. То есть мы решаем один и тот же набор фундаментальных проблем, просто делаем это в разных измерениях.

С архитектурой ESB вы можете легко использовать встроенные возможности для создания виртуальных сервисов и функциональных возможностей, которые полезны для межсервисного взаимодействия:



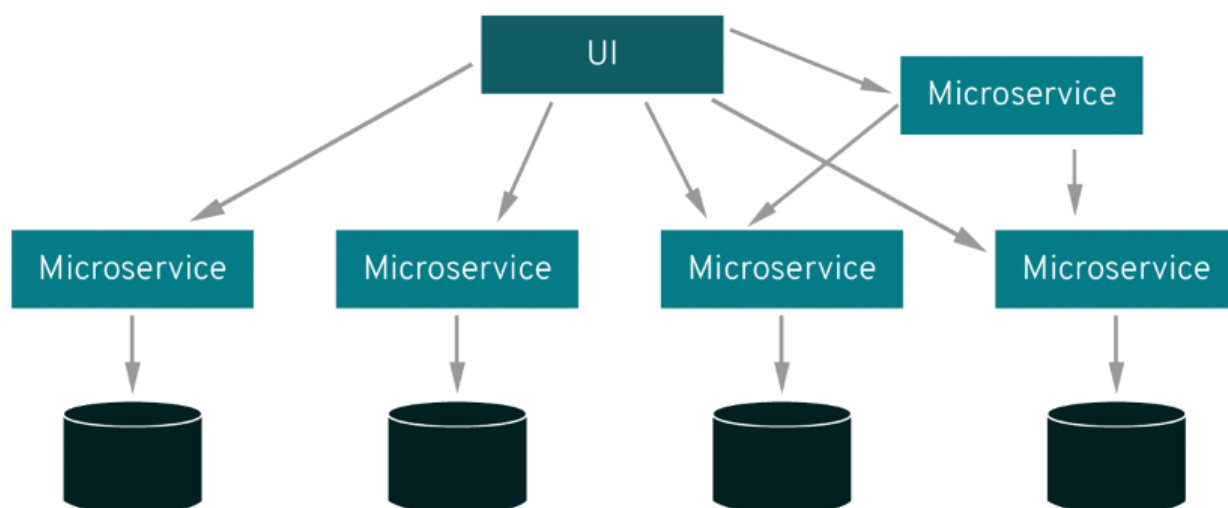
Когда вы реализуете тот же сценарий с использованием микросервисов, у вас больше нет централизованного уровня интеграции ESB. Вы должны реализовать все функциональные возможности на уровне микросервисов.



Самая сложная задача при реализации микросервисной архитектуры заключается не в создании самих сервисов, а в обеспечении связи между ними. Для выполнения своей функции одному сервису может потребоваться запросить данные у нескольких других микросервисов. Но что, если некоторые микросервисы будут перегружены запросами? Здесь на помощь приходит service mesh — технология направляет запросы от одного сервиса к другому, оптимизируя совместную работу всех частей приложения.

**Разве микросервисы уже не делают это?**

Архитектура микросервисов позволяет разработчикам вносить изменения в сервисы приложения без необходимости полного повторного развёртывания. В отличие от разработки приложений в других архитектурах, отдельные микросервисы создаются небольшими командами с возможностью гибкого выбора собственных инструментов и языков программирования. По сути, микросервисы создаются независимо, взаимодействуют друг с другом и могут выходить из строя по отдельности, не приводя к сбою во всем приложении.



Связь между сервисами — то, благодаря чему микросервисная архитектура работает. Логику, управляющую этой связью, можно закодировать в каждом сервисе и без уровня service mesh, но по мере усложнения взаимодействия сервисов ценность service mesh возрастает. Для облачных приложений с микросервисной архитектурой service mesh — это способ объединения большого количества отдельных сервисов в функциональное приложение.

## Как это работает

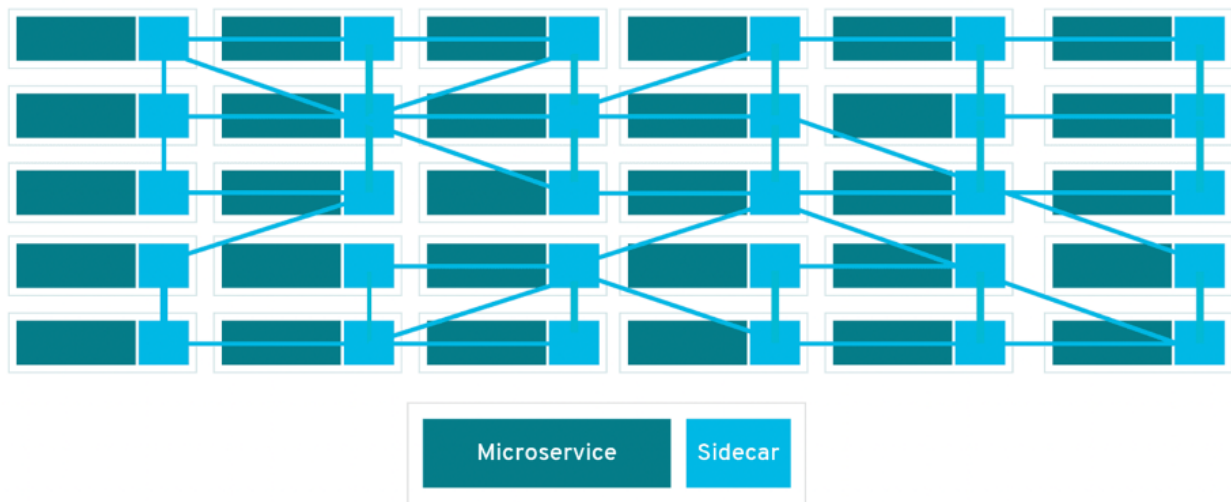
Service mesh не вводит новые функциональные возможности в среду выполнения приложения — приложения в любой архитектуре нуждались в правилах, определяющих, как запросы доставляются из точки А в точку Б. Отличие service mesh, заключается в том, что эта технология выводит логику, управляющую взаимодействием между сервисами, на уровень инфраструктуры.

Service mesh встраивается в приложение в виде массива сетевых прокси.



В service mesh запросы маршрутизируются между микросервисами через прокси-серверы на их собственном уровне инфраструктуры. Отдельные прокси-серверы в service mesh иногда называют «сайдкарами», так как они работают вместе с каждым

сервисом, а не внутри него. В совокупности прокси-серверы, отделенные от каждого сервиса, образуют сеть:



Без service mesh разработчикам сложнее фокусироваться на бизнес-целях, приходится тратить больше времени на диагностику и исправление сбоев, поскольку логика, управляющая взаимодействием микросервисов, скрыта внутри каждого сервиса.

## Разные реализации

---

Linkerd и Istio — популярные реализации service mesh с открытым исходным кодом. Они имеют схожую архитектуру, но разные механизмы. [Здесь](#) можно почитать о сравнении инструментов.

## Как service mesh оптимизирует коммуникации

---

Каждый новый сервис усложняет коммуникационную среду и создаёт новые точки отказа. В сложной микросервисной архитектуре становится практически невозможно определить причину проблемы без service mesh.

Service mesh фиксирует все аспекты взаимодействия между сервисами в качестве показателей производительности. Если в каком-то сервисе происходит сбой, service mesh собирает данные о том, сколько времени прошло до успешной повторной попытки. По мере накопления данных о времени сбоя можно подготовить правила для определения оптимального времени ожидания перед повторным запуском сервиса. Это гарантирует, что система не будет перегружена ненужными повторными попытками.

## Плюсы и минусы

---

Кратко перечислим основные плюсы и минусы service mesh.

---

### Плюсы

### Минусы

---

---

— стандартные функции реализованы вне кода микросервиса и могут использоваться повторно

— есть решение большинства проблем микросервисной архитектуры: распределённая трассировка, ведение журнала, безопасность, контроль доступа и др.

— больше свободы, когда дело доходит до выбора языка реализации микросервисов: не нужно беспокоиться, поддерживает ли язык или библиотека для создания функций сетевого приложения

— наличие service mesh резко увеличивает количество экземпляров среды выполнения, которые есть в данной реализации микросервиса

— service mesh решает только проблемы взаимодействия между сервисами, однако помимо этого существуют и другие проблемы вроде сложной маршрутизации, которые тоже необходимо решать в бизнес-логике микросервиса

— технология service mesh ещё слишком молода для крупномасштабного развёртывания

---

## Коротко о главном

Service mesh решает ключевые проблемы, связанные с реализацией микросервисной архитектуры. Технология позволяет сосредоточиться на бизнес-логике и не тратить время на сетевые функции между сервисами. Благодаря service mesh микросервис не взаимодействует напрямую с другими сервисами. Кроме того, обеспечивается встроенная поддержка сетевых функций: отказоустойчивости, маршрутизации, контроля доступа и др. При этом service mesh не зависит от языка: поскольку микросервис всегда находится на вершине стандартных протоколов (HTTP1.x/2.x, gRPC), вы можете использовать любые технологии — они все равно будут работать с service mesh.

Материал основан на статьях [«What's a service mesh?»](#) и [«Service Mesh for Microservices»](#).