Service Mesh

Memi Lavi
www.memilavi.com

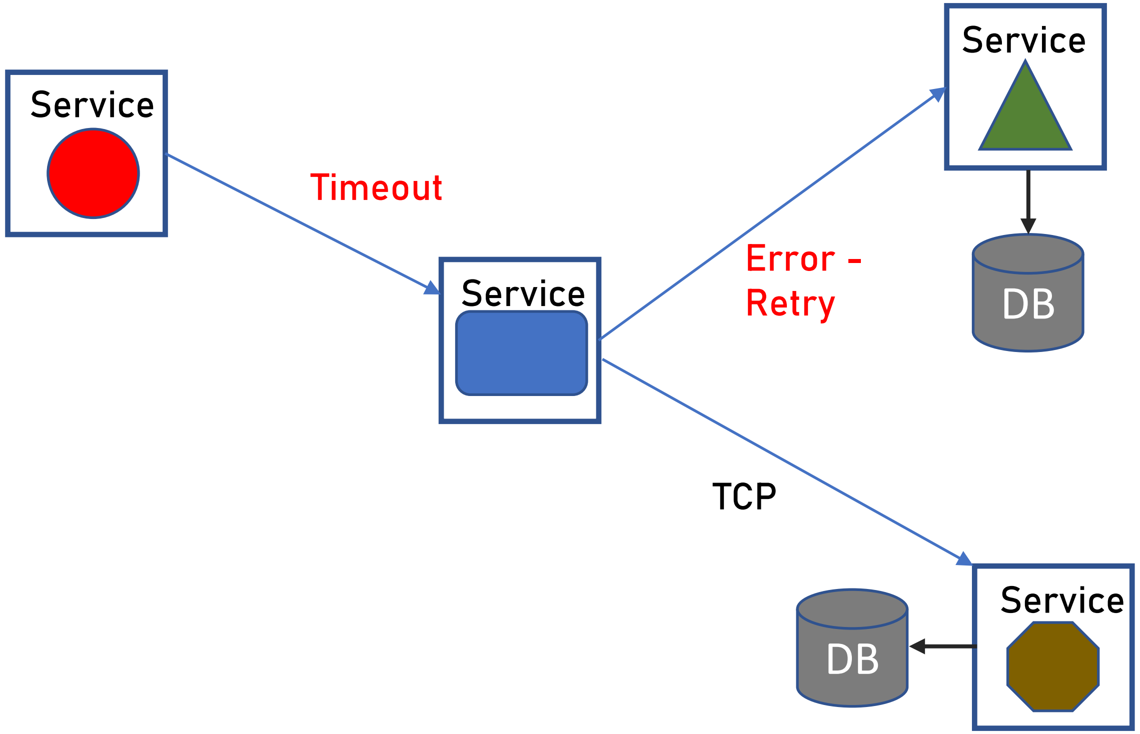
Service Mesh

- Manages all service-to-service communication
- Provides additional services
- Platform agnostic (usually…)

Problems Solved by Service Mesh

- Microservices communicate between them a lot
- The communication might cause a lot of problems and challenges:
    - Timeouts
    - Security
    - Retries
    - Monitoring

Problems Solved by Service Mesh



Service Mesh

- Software Components that sit near the service and manage all service-to-service communication
- Provides all communication services
- The service interacts with the service mesh only

Service Mesh Services

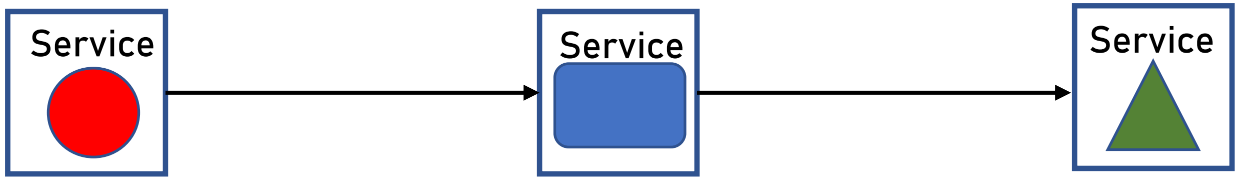
- Protocol conversion
- Communication security
- Authentication
- Reliability (timeouts, retries, health checks, circuit breaking)
- Monitoring
- Service Discovery

Service Mesh Services

- Testing (A/B testing, traffic splitting)
- Load balancing

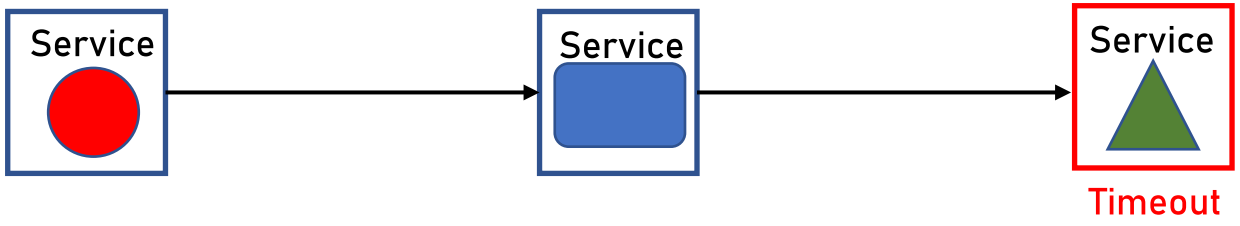
Circuit Breaker

- Prevents cascading failures when a service fails



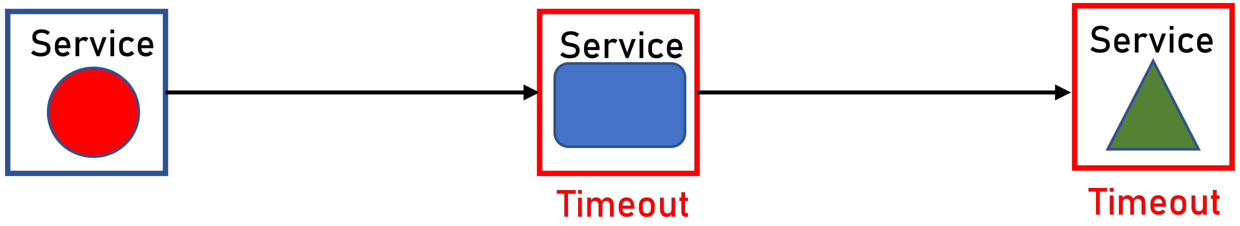
Circuit Breaker

- Prevents cascading failures when a service fails



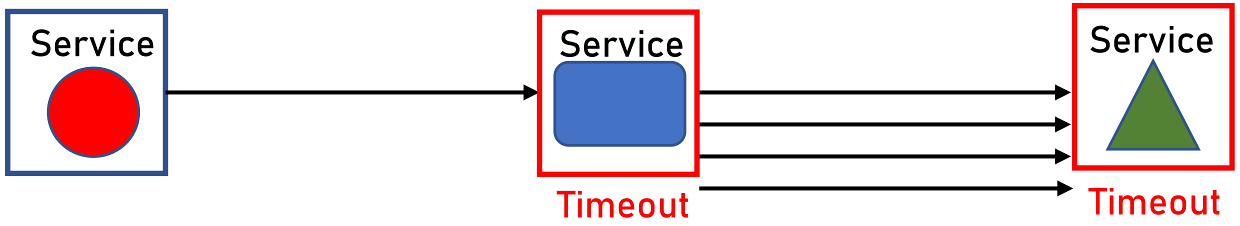
Circuit Breaker

- Prevents cascading failures when a service fails



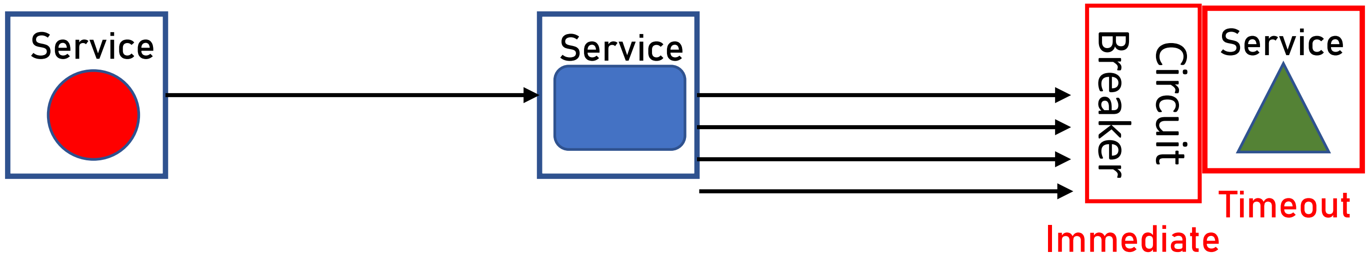
Circuit Breaker

- Prevents cascading failures when a service fails



Circuit Breaker

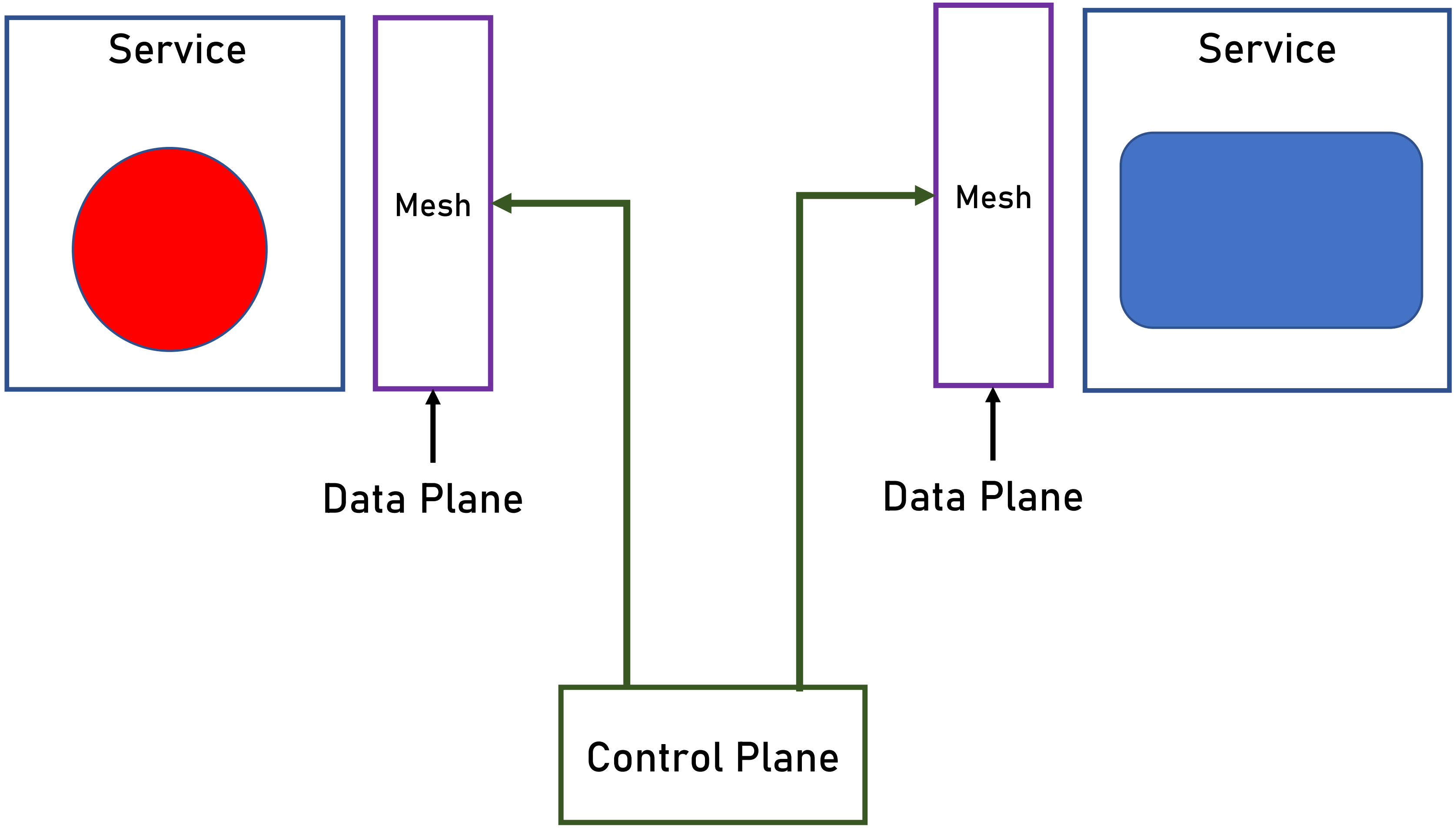
- Prevents cascading failures when a service fails



Service Mesh

- In short:
  - Service's developers need not handle communication aspects when using Service Mesh
  - Focus on the business, not the plumbing

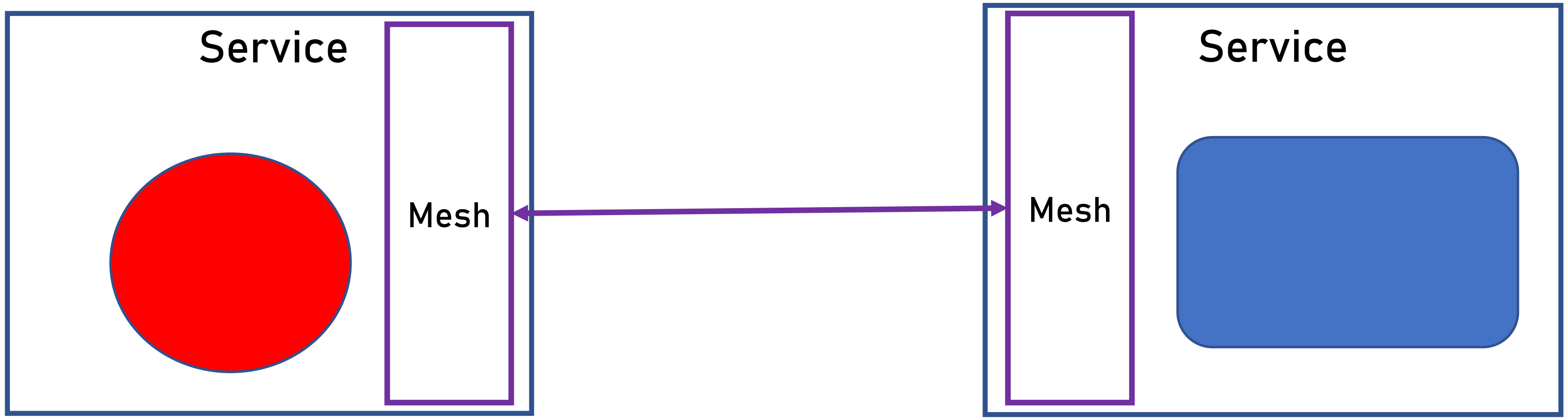
Service Mesh Architecture



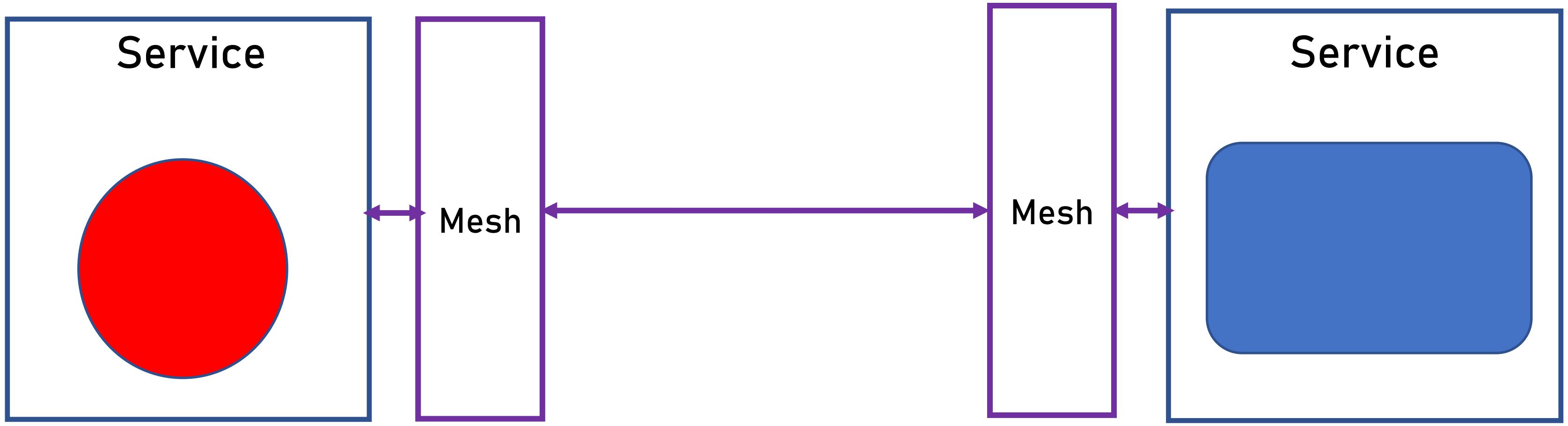
Types of Service Mesh

- Two main types:
  - In-Process
  - Sidecar

In-Process



Sidecar



In-Process vs Sidecar

In-Process

- Performance

Sidecar

- Platform agnostic
- Code agnostic

More popular

Products and Implementations

- There are quite a few Service Mesh implementations
- Some in-process, most sidecar
- Most free, some aren’t
- DO NOT develop your own

Sidecar

In-Process

Should You Use Service Mesh?

- Only if:
  - You have a lot of services…
  - Which communicate a lot with each other
  - Or you have a complex communication requirement with various protocols or brittle network