From Chaos to Calm: Improving Service Mesh Reliability

KCD Chennai, July 2023

Hello!

I am **Atulpriya Sharma**

- Senior Developer Advocate @ InfraCloud Technologies
- CNCF Ambassador | CNCF Hyderabad Organizer
- Cloud Native, Kubernetes, DevOps
- Food & Travel Blogger @ socialmaharaj.com

@TheTechmaharaj | Atulpriya Sharma

Agenda

- Service Mesh
- Reliability & Resiliency
- Chaos Engineering
- Chaos Engineering in Service Mesh
- Demo

1 Service Mesh

The backstage crew that makes your microservices shine on stage, without any embarrassing wardrobe malfunctions.
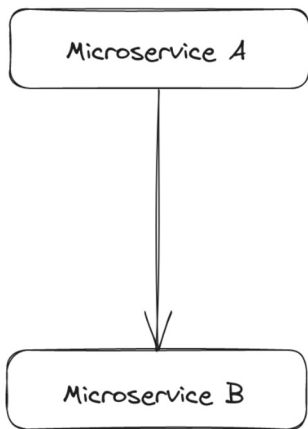
Service Mesh & The Need Of It

- Increasing adoption of microservices architecture

- Complex service architecture

  - Decentralized Communication
  - Limited Observability
  - Manual Resilience Mechanisms
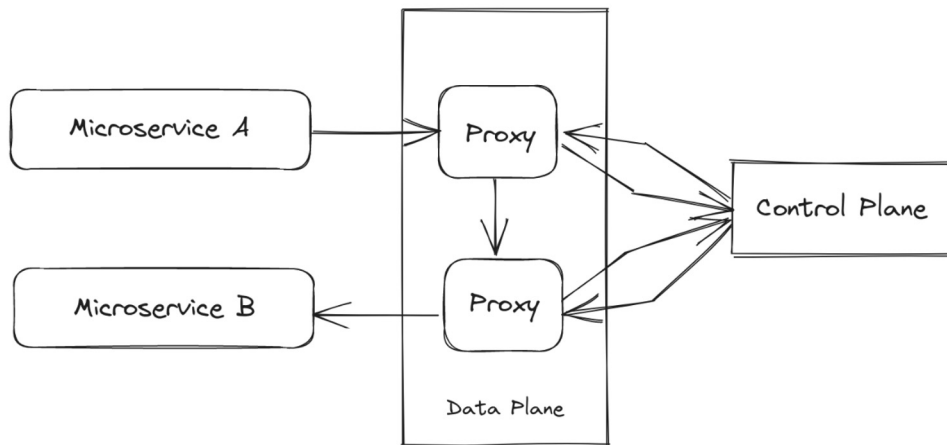 - Security Management
  - Scalability Issues

Service Mesh

Microservices

```
┌─────────────────────────┐
│      Microservice A      │
└─────────────────────────┘
             │
             │
             ▼
┌─────────────────────────┐
│      Microservice B      │
└─────────────────────────┘
```

Microservices + Service Mesh

```
┌──────────────┐      ┌───────────┐                 ┌──────────────┐
│ Microservice A│ ───▶ │   Proxy   │ ◀─┐             │ Control Plane │
└──────────────┘      └───────────┘   │             └──────────────┘
                            │          │
                            ▼          │
┌──────────────┐      ┌───────────┐   │
│ Microservice B│ ◀─── │   Proxy   │ ◀─┘
└──────────────┘      └───────────┘
                         Data Plane
```

Service mesh enables service-to-service interactions by ==abstracting away the complexities== of networking, routing, load balancing, and other cross-cutting concerns.
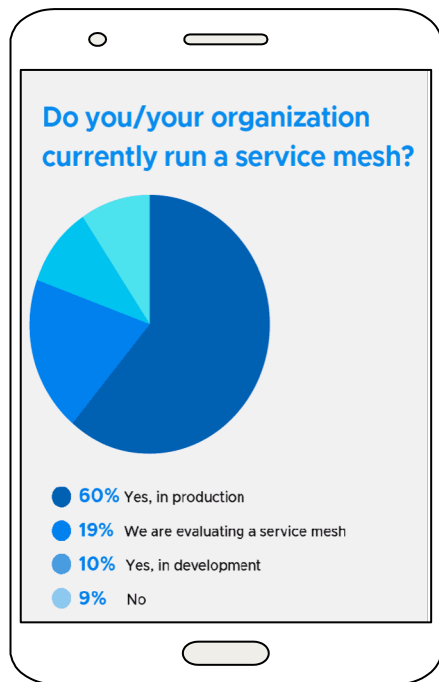
Benefits

- Service Discovery
- Traffic Routing
- Load Balancing
- Improved Observability
- Enhance Security
- Platform Independence

Service Mesh In Numbers

Do you/your organization currently run a service mesh?

- **60%** Yes, in production
- **19%** We are evaluating a service mesh
- **10%** Yes, in development
- **9%** No

Which of the following features and capabilities are driving your adoption of service mesh?
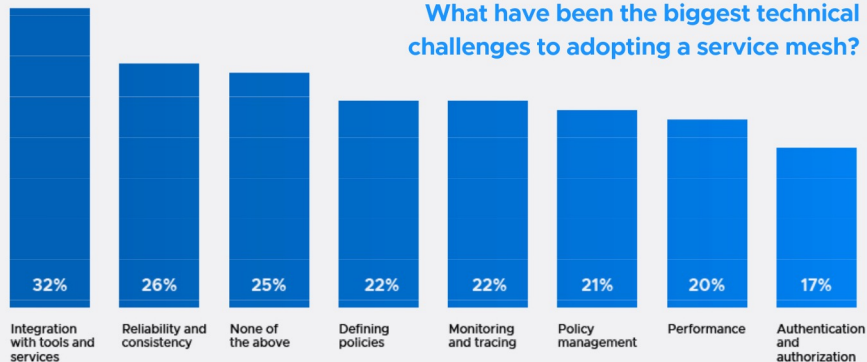
- **79%** Security — for example, mTLS
- **78%** Observability — for example, golden metrics
- **62%** Traffic management — for example, blue-green deploys
- **56%** Reliability — for example, request retries

What have been the biggest technical challenges to adopting a service mesh?

32%	26%	25%	22%	22%	21%	20%	17%
Integration with tools and services	Reliability and consistency	None of the above	Defining policies	Monitoring and tracing	Policy management	Performance	Authentication and authorization

Source: https://www.cncf.io/wp-content/uploads/2022/05/CNCF_Service_Mesh_MicroSurvey_Final.pdf

2 Reliability & Resiliency

Because trust issues are for relationships, not for your infrastructure.

Why Reliability?

- Traffic Surges and Load Testing
- Multi-Cloud Deployments
- Dependency Outages
- Autoscaling and Elasticity
- Disaster Recovery

Testing For Reliability

- ◉ Observability
- ◉ Load Testing
- ◉ Fault Injection
- ◉ Security Testing
- ◉ Chaos Engineering

3 Chaos Engineering

Making the world a better place, one broken system at a time.

Understanding Chaos Engineering

- **What:** The practice of purposefully breaking things to make systems stronger

- **When:** Netflix in the early 2000s

- **Why:** Robust and reliable systems

- **How:** Intentionally introducing controlled failures into a system to uncover vulnerabilities to improve resiliency

Role of Chaos In Distributed Systems

- The network is reliable.
- Latency is zero.
- Bandwidth is infinite.
- The network is secure.
- Topology doesn't change.
- There is one administrator.
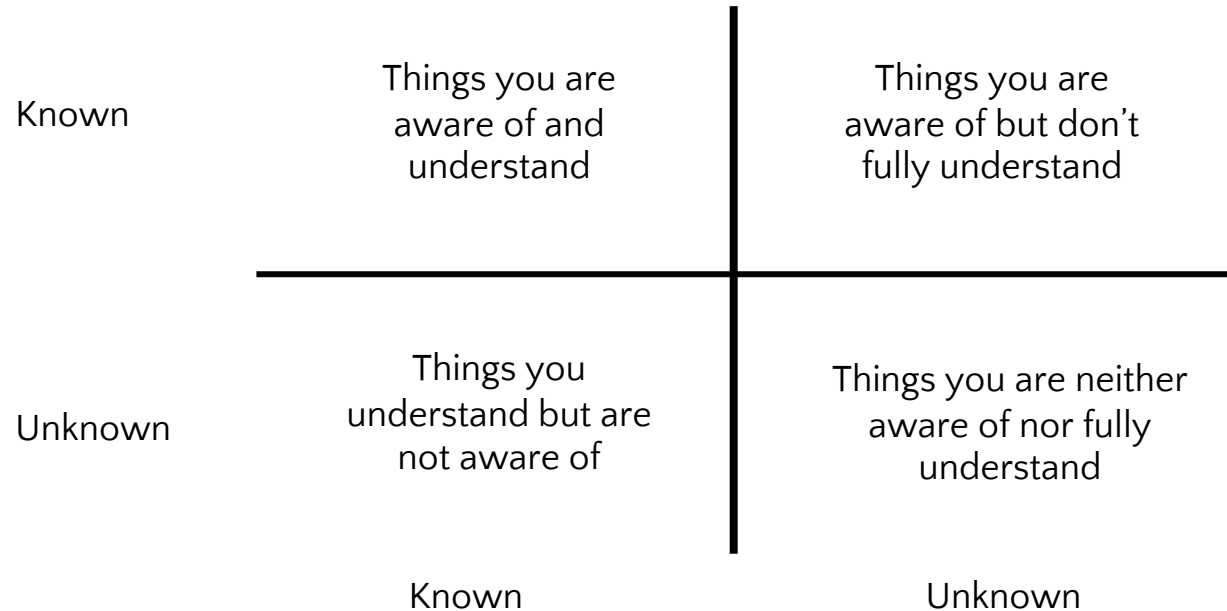- Transport cost is zero.
- The network is homogeneous.

Chaos Principles

- Hypothesize system behavior under failure scenarios.

- Define steady state as the baseline for comparison.

- Simulate real-world conditions and failure scenarios.

- Test in production-like environments

Designing Chaos Experiments

	Known	Unknown
Known	Things you are aware of and understand	Things you are aware of but don't fully understand
Unknown	Things you understand but are not aware of	Things you are neither aware of nor fully understand

Chaos Engineering Tools in CNCF

Technical 101

Chaos engineering tools and practices are critical to achieving high availability for your applications. Distributed systems are often too complex to be fully understood by any one engineer and no change process can fully predetermine the impact of changes on an environment. By introducing deliberate chaos engineering practices teams are able to practice and automate failure recovery. Chaos Mesh and Litmus Chaos are two CNCF tools in this space.

Buzzwords	CNCF Projects
• Chaos Engineering	• Chaos Mesh (incubating) • Chaosblade (sandbox) • Litmus (incubating)

Chaos Mesh
CNCF Incubating

Litmus
CNCF Incubating

ChaosToolkit   ChaosBlade   chaoskube   Gremlin   POWERFUL SEAL   steadybit   TRaaS CHAOS

Source: https://landscape.cncf.io/guide#observability-and-analysis--chaos-engineering

Chaos Engineering in Service Mesh

4

Because life is too short to play by the rules all the time, even for your microservices.

Chaos in Service Mesh

- ◎ Validate resilience

- ◎ Optimize performance and scalability

- ◎ Continuous improvement

- ◎ Validate assumptions

- ◎ Prepare for the unexpected

5 Demo

Who needs a stand-up comedy routine when you can entertain your audience with a hilarious demo fail or two?

What Next?

- Gather & Analyze Data

- Assess Impact

- Root Cause Analysis

- Documentation

- Remediate

- Repeat & Reiterate

6 # Questions?

A chance for me to say, "Your question, very good, I'll get back to you"

Thanks!

Connect With Me:

-> Twitter: @TheTechMaharaj

-> LinkedIn: Atulpriya Sharma

-> Email: atulpriya.sharma@infracloud.io

-> Blog: Socialmaharaj.com | @Atulmaharaj