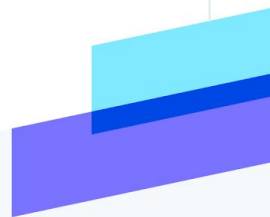


**stripe**

# **Service Mesh: A Hole in the Pocket?**



# Who Are We?



## **John Murray**

Service Networking Engineer, Stripe  
Occasional Envoy Contributor  
C++ Enthusiast

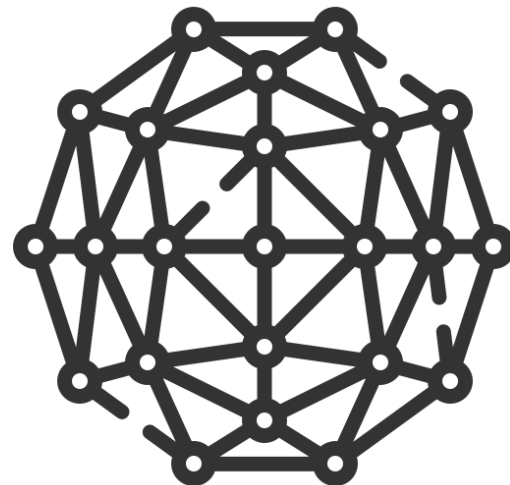


## **Venil Noronha**

Service Networking Engineer, Stripe  
Envoy and Istio Contributor  
Distributed Systems Enthusiast

# Promises of a Service Mesh

- **Simplified Application Networking**
  - ◆ Load Balancing
  - ◆ Circuit Breaking
  - ◆ Retries
- **Traffic Patterns**
  - ◆ Blue/Green Deployments
  - ◆ Traffic Splitting
  - ◆ Fault Injection
- **Security**
  - ◆ Mutual TLS (mTLS)
  - ◆ Role Based Access Control (RBAC)
- **Observability**
  - ◆ Metrics
  - ◆ Access Logs
  - ◆ Distributed Tracing



# Costs of Running a Service Mesh

→ **Explicit Costs**

The cost of running a group of proxies alongside application processes.

→ **Hidden Costs**

The cost not very apparent unless one looks at their service mesh very closely.

→ **Integration Costs**

The cost associated with integrating a service mesh with third-party services.

→ **Developer Costs**

The cost of having a service mesh layer which application developers now need to understand.

→ **Support Costs**

The cost of maintaining and operating a service mesh.

# Costs of Running a Service Mesh

## → Explicit Costs

### ◆ CPU

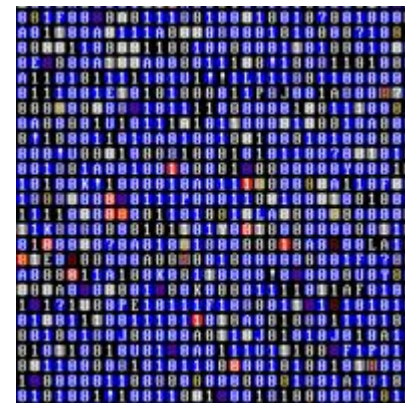
- Misconfigured quota

### ◆ Memory

- Misconfigured quota

### ◆ Latency

- Misconfigured concurrency



# Costs of Running a Service Mesh

## → Hidden Costs

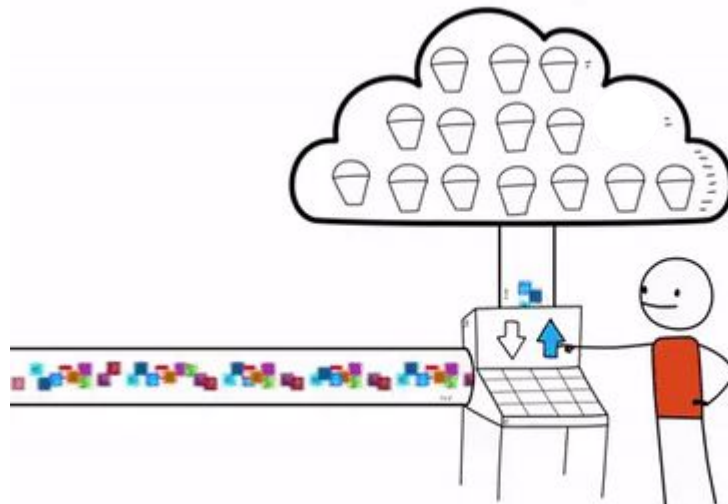
- ◆ Network bandwidth usage
  - Control plane traffic
- ◆ I/O costs
  - Public cloud policies
  - Feature specific: healthchecks, hedging, retries, etc.



# Costs of Running a Service Mesh

## → Integration Costs

- ◆ Metrics storage
  - Stats storage
  - Vendor costs
- ◆ Log storage
  - Vendor costs
- ◆ Distributed Tracing
  - Vendor costs



# Costs of Running a Service Mesh

## → Developer Costs

### ◆ Education

- How to react to various errors
- Code patterns to use and avoid
- Managing resource quotas





# Costs of Running a Service Mesh

## → Support Costs

- ◆ Maintenance
  - CVE fixes, patches, etc.
  - API compatibility and upgrades
- ◆ Operations
  - Data plane upgrades
  - Control plane upgrades
  - Cert rotations
- ◆ Troubleshooting
- ◆ Ownership of client libraries



# Strategies for Controlling Cost

## → Understand Defaults

Service meshes and proxies come with default configurations, and understanding it is key to control costs.

## → Access Log Sampling

Service meshes can generate a huge amount of access logs, and having a sampling strategy can help cut down storage costs.

## → Metrics

Service mesh proxies can generate a large number of stats, and simplifying it can help a great deal.

## → AZ-aware Routing

Cloud vendors have policies for networking and i/o, especially when it's across AZs, and there are ways to improve routing across AZs (Availability Zones).

# Strategies for Controlling Cost

## → Understand Defaults

- ◆ HTTP/2 configuration in Envoy
- ◆ Envoy concurrency
- ◆ Retry configuration
- ◆ Circuit breaker configuration
- ◆ Metrics



# Strategies for Controlling Cost

## → Access Log Sampling

- ◆ Typically see a lot of 2xx responses
- ◆ Can overwhelm the storage system
- ◆ Not all access log entries are interesting
- ◆ Usage of access logs (incidents v/s learning)
- ◆ Sample/filter logs depending on need

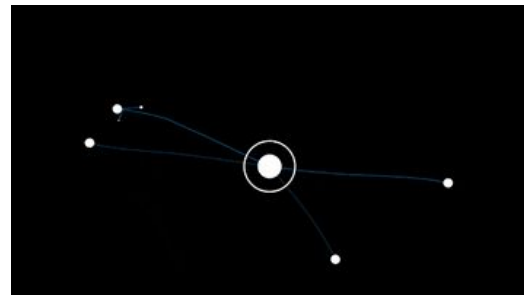




# Strategies for Controlling Cost

## → AZ-aware Routing

- ◆ Cloud vendors charge for traffic crossing AZ borders
- ◆ Need to clearly understand which services route across AZs
- ◆ Identify whether any cross-AZ control plane communication can be cut down
- ◆ Implement configurations to prefer local services over cross-AZ ones
- ◆ Implement health checks to ensure there are no outages when preferring local services



# Open Problems

## → Troubleshooting

How to simplify troubleshooting network v/s application issues when running a service mesh?

## → Ownership of Client Libraries

Does the networking team need to own client libraries? If so, where to draw the line on networking interfaces v/s business abstractions?

## → Developer Education

What can we do to educate application developers about service mesh behavior? Can service meshes be a 100% transparent to service owners?

## → Maintenance

How to simplify upgrades? How to minimize friction when migrating between different API versions?

# Summary

- Promises of a Service Mesh
- Costs of Running a Service Mesh
- Strategies for Controlling Cost
- Open Problems



Is it worth it?



**stripe**

**Thank You!**

**We're Hiring!**

John Murray | [murray@stripe.com](mailto:murray@stripe.com)  
Venil Noronha | [venil@stripe.com](mailto:venil@stripe.com)