

# Service Mesh Basics with



LINKERD

# Charles Pretzer



@charpretz



cpretzer



@charles



charles@buoyant.io

# Agenda



Service Mesh Overview



Deploying the Linkerd Service Mesh



SMI

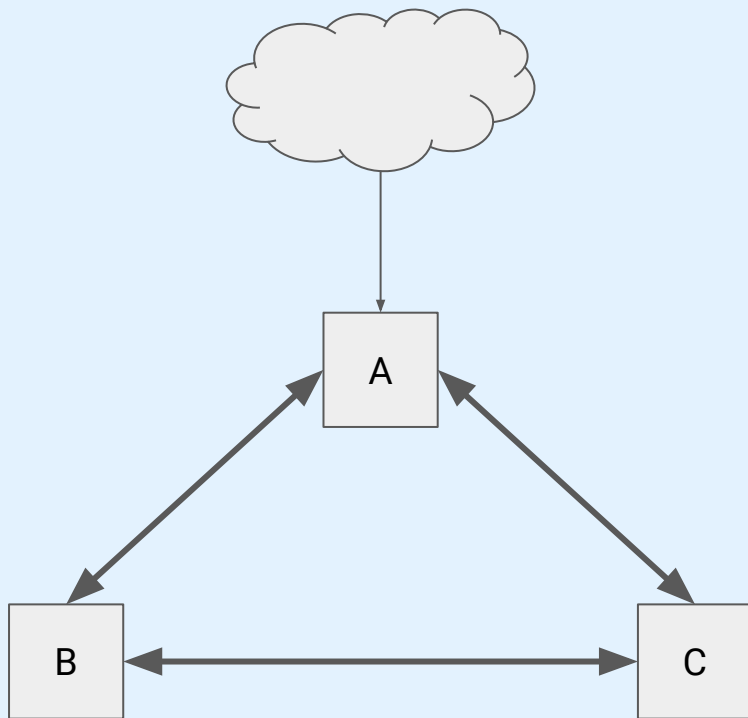
Splitting Traffic

Questions/Discussion

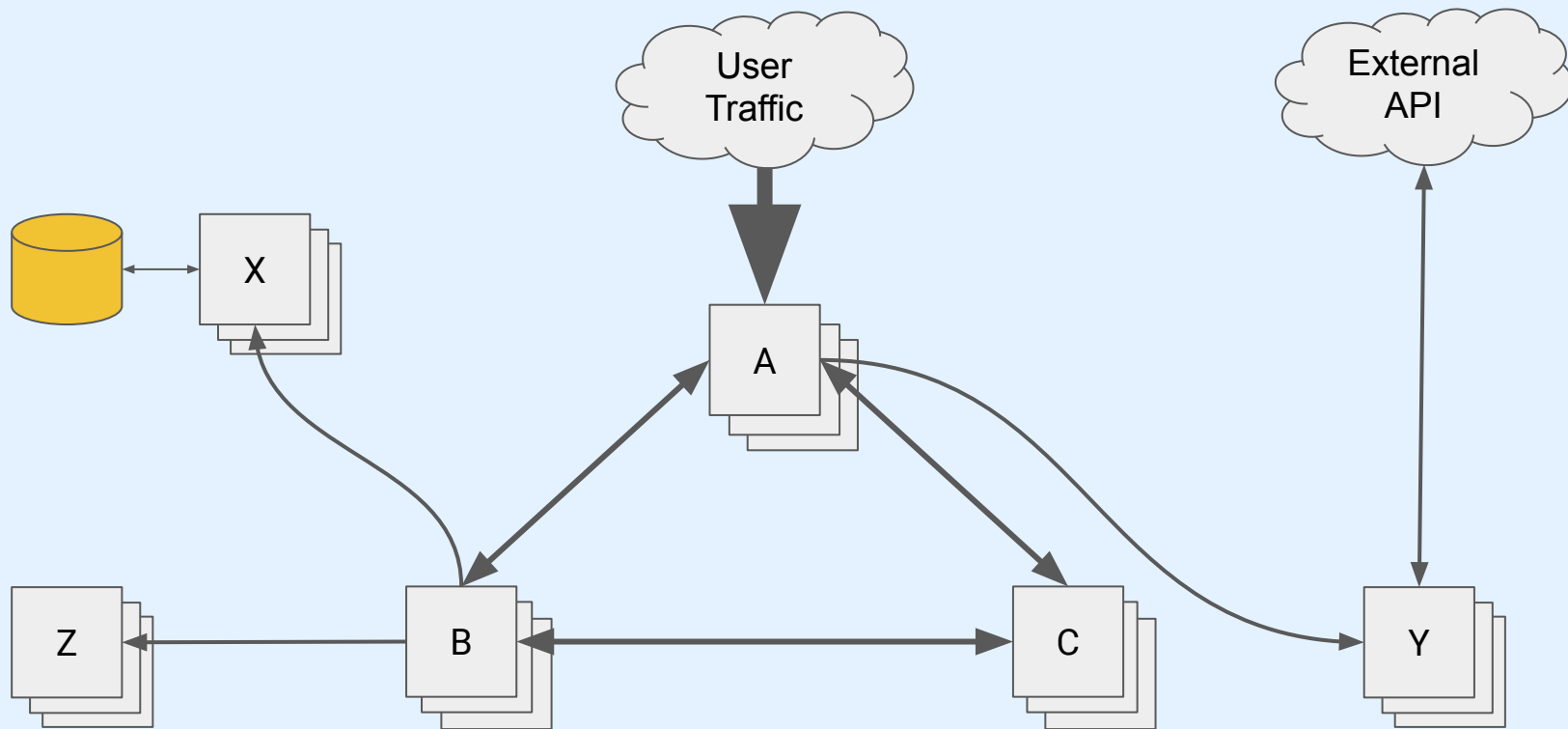


# Service Mesh Overview

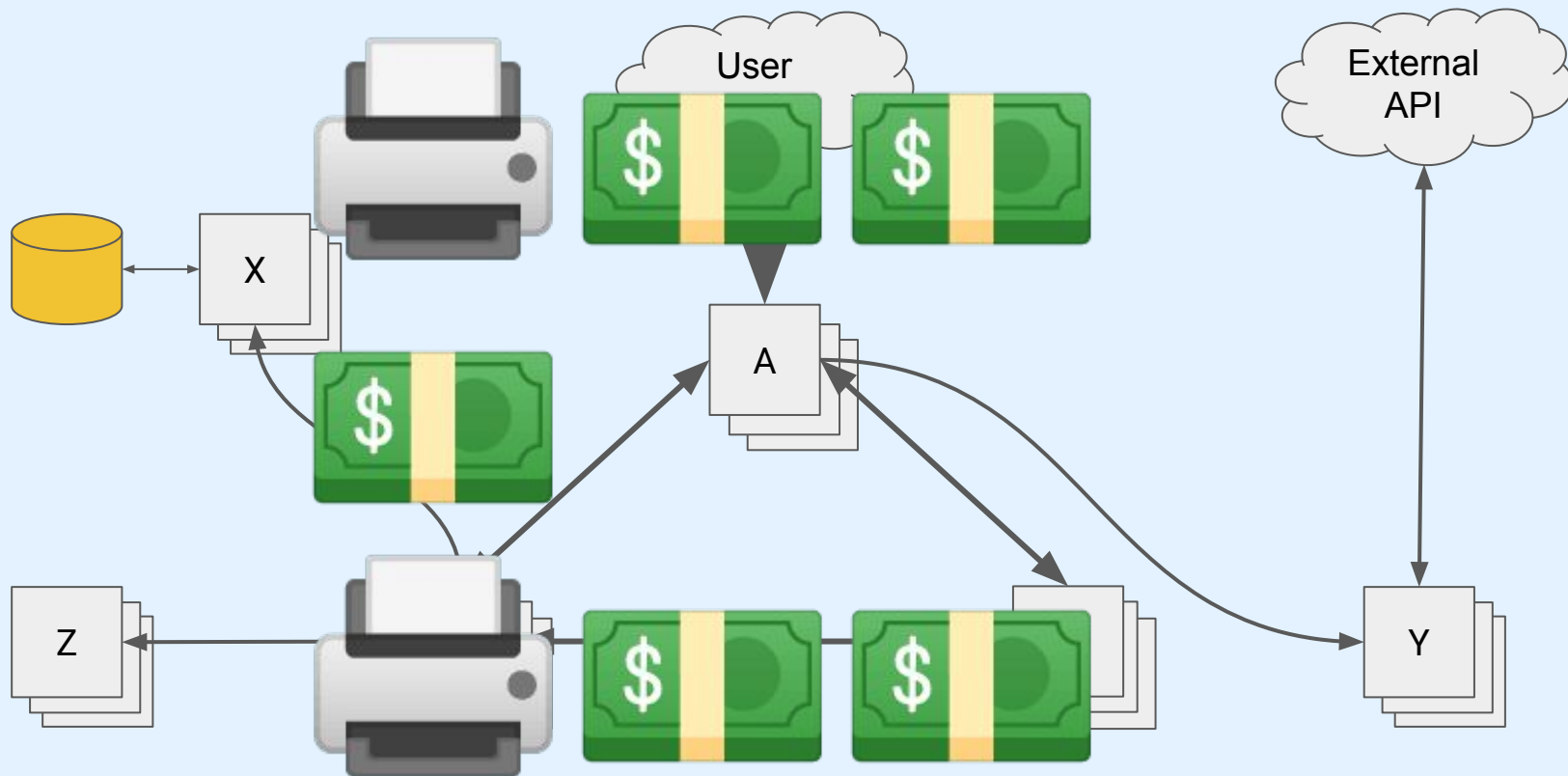
# What's a service mesh?



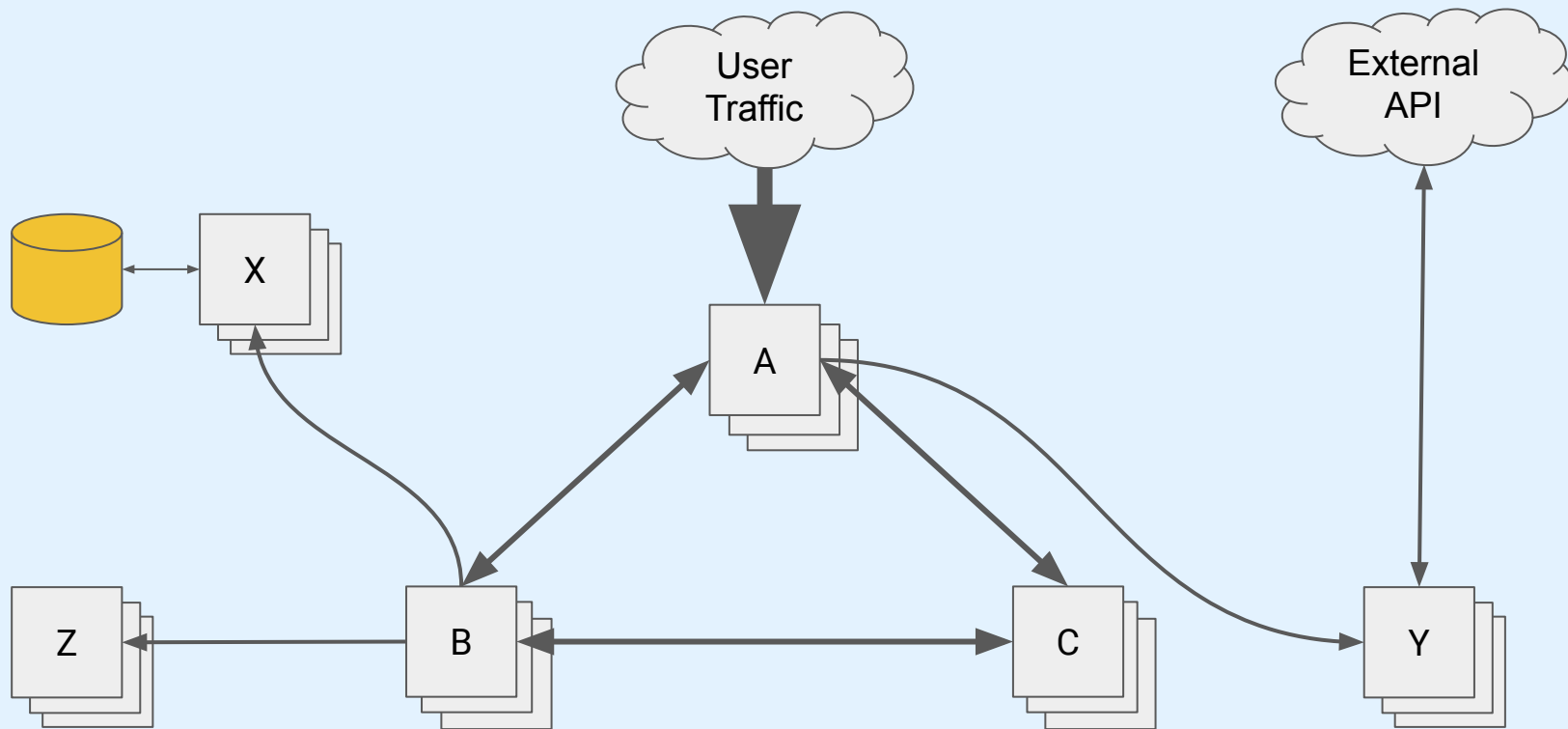
# Service Mesh: The basics



# Service Mesh: The basics

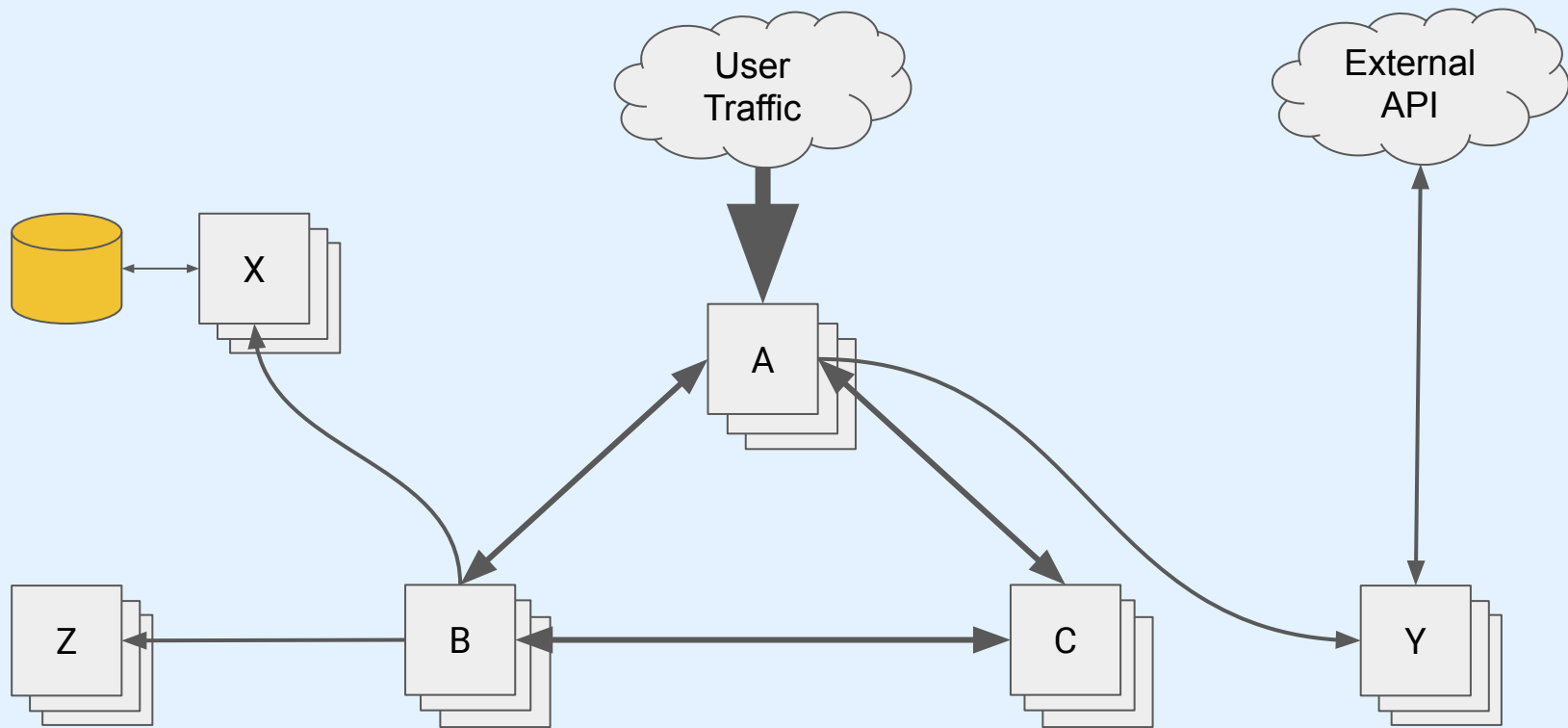


# Service Mesh: The basics

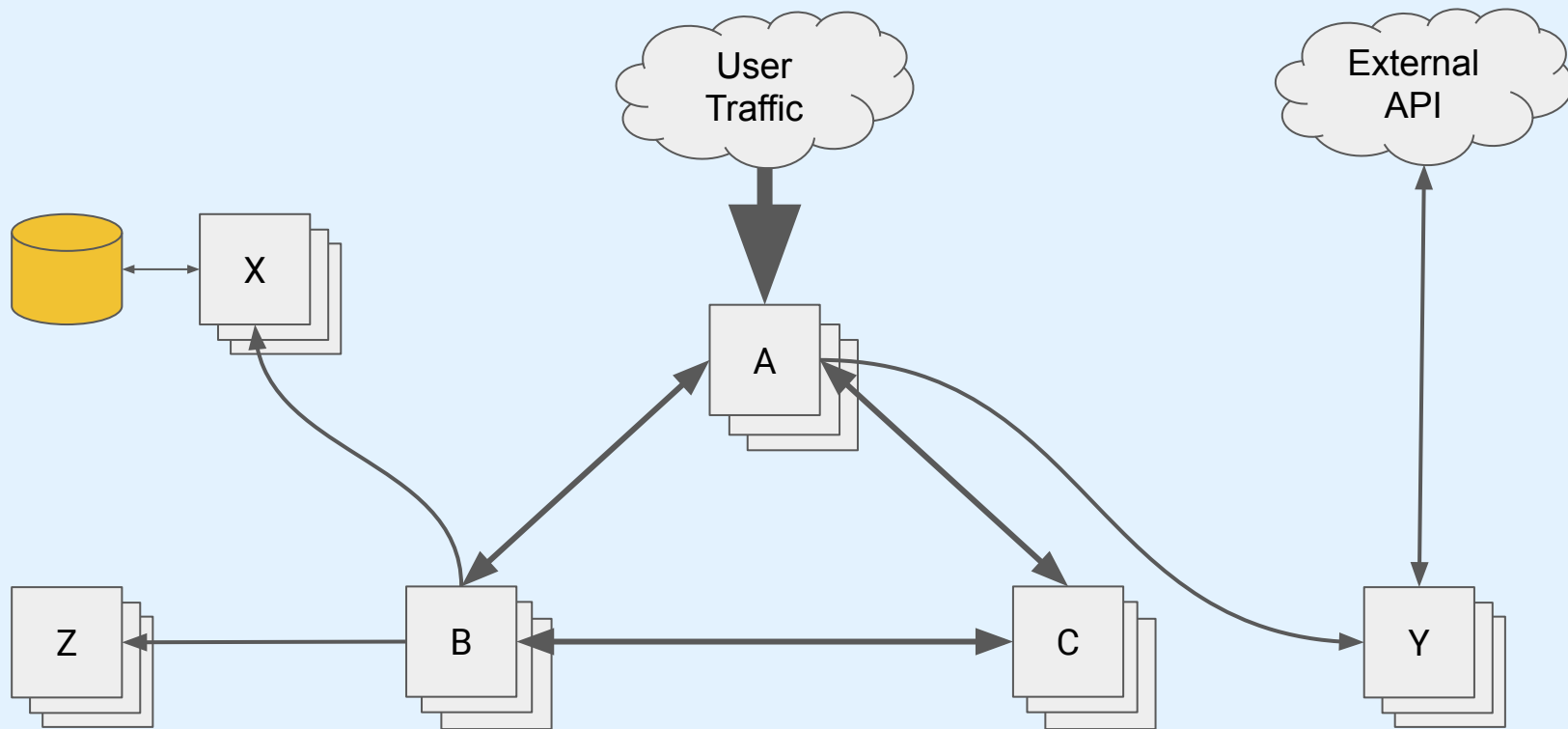




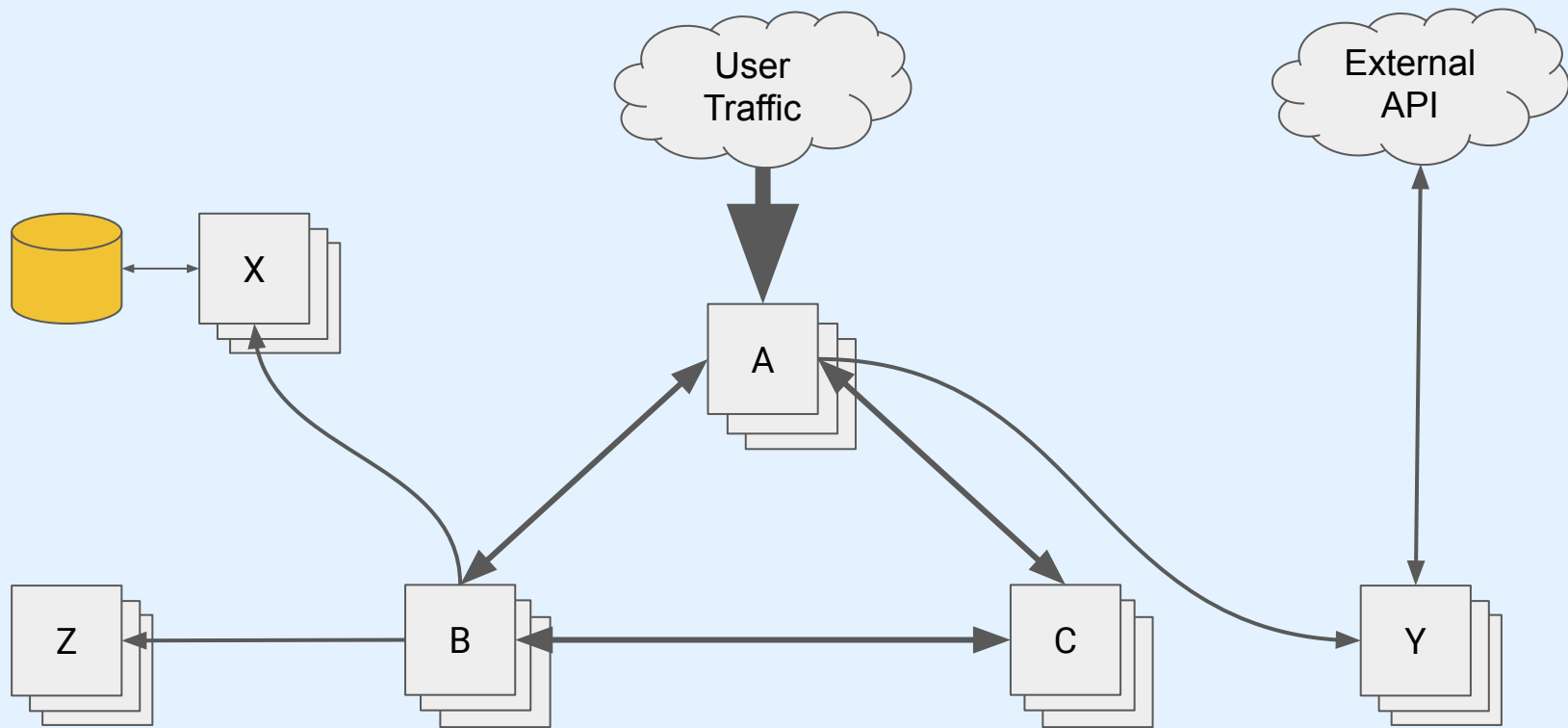
# Service Mesh: The basics



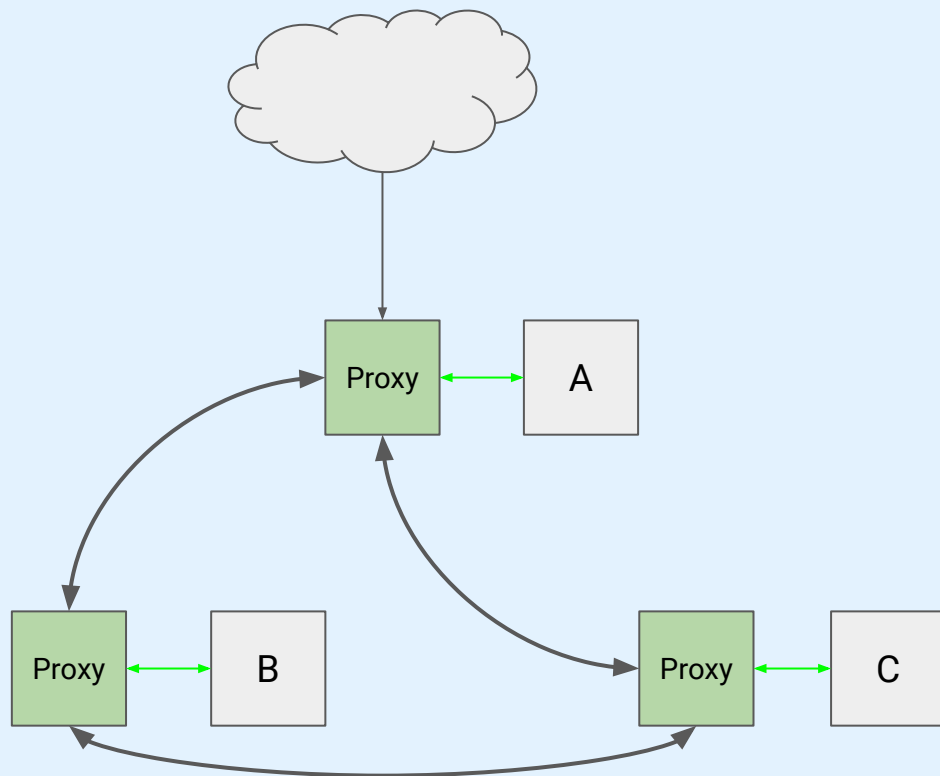
# Service Mesh: The basics



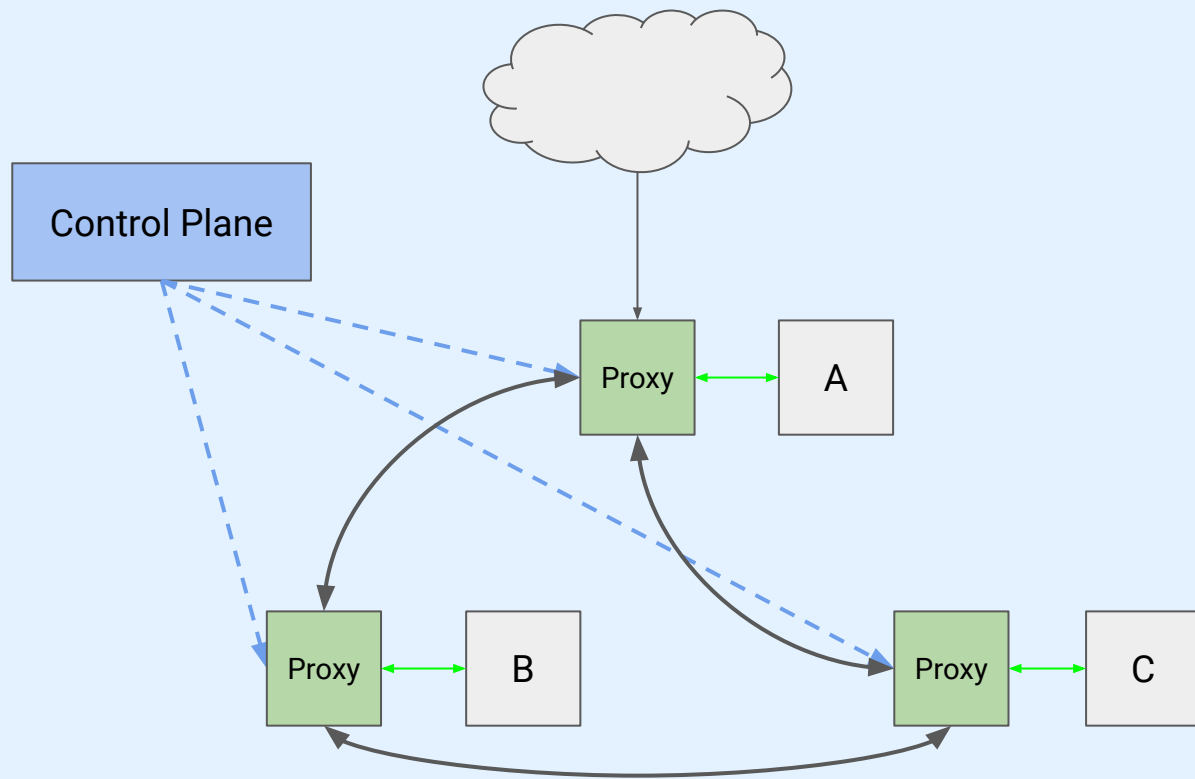
# Service Mesh: The basics



# Service Mesh: Data Plane



# Service Mesh: Control Plane



# Linkerd

- Observability: Collecting actionable traffic metrics
- Security: Encrypting traffic between services
- Reliability: Ensuring services are available
- Traffic Management: Routing traffic to services

# Linkerd: Observability

- Rich traffic metrics: "Golden Metrics"
  - Request rate, Success rate, latency
  - Across many dimensions
- Request inspection
- Distributed Tracing

# Linkerd: Security

- Cryptographic identity
  - mTLS between services
  - On by default
  - Transparent



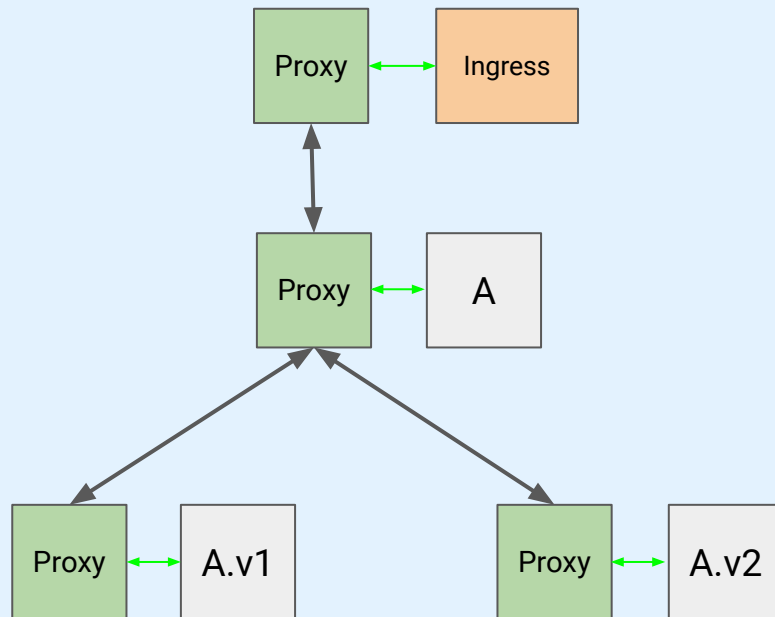
# Linkerd: Reliability

- Latency aware load balancing
- Retries
- Timeouts

# Linkerd: Traffic Management

- Traffic Split

- Introduced in 2.4.0
- Enables canary and blue/green deployments





Do I need a service mesh?

# Service Mesh Characteristics

- Observability: Collecting actionable traffic metrics
- Security: Encrypting traffic between services
- Reliability: Ensuring services are available
- Traffic Management: Routing traffic to services

# Service Mesh Characteristics: Observability

- Observability: Collecting actionable traffic metrics
  - APM Instrumented in code via libraries (NewRelic, DataDog, etc.)

# Service Mesh Characteristics: Security

- Security: Encrypting traffic between services
  - mTLS: certificate management and distribution
  - service code must be modified to use and validate certs

# Service Mesh Characteristics

- Reliability: Ensuring services are available
  - Service code must include logic for retries and timeouts
  - Load balancing is difficult

# Service Mesh Characteristics

- Traffic Management: Routing traffic to services
  - Service code must be able to discover other services and make itself discoverable
    - This pattern lends itself to a global component



# Cognitive Load

- 9-12 Services is the limit for any team to manage on their own
  - Debugging the graph of services is not straightforward
  - Logs are no longer in one linear file
    - Interpolating and matching timestamps across logs
  - Tracking aggregates requires a bespoke dashboard



# Installing Linkerd



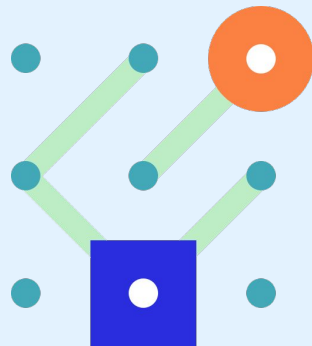
# Service Mesh Interface (SMI)

# Service Mesh Interface

**Announced in May 2019, Service Mesh Interface (SMI) is a specification for service meshes that run on Kubernetes. It defines a common standard that can be implemented by a variety of providers.**

Service Mesh Interface provides:

- A standard interface for meshes on Kubernetes
- A basic feature set for the most common mesh use cases
- Flexibility to support new mesh capabilities over time
- Space for the ecosystem to innovate with mesh technology



# Service Mesh Interface Concepts

**Initial specifications for the top three service mesh features covering the most common service mesh capabilities:**

- **Traffic policy** – apply policies like identity and transport encryption across services
- **Traffic telemetry** – capture key metrics like error rate and latency between services
- **Traffic management** – shift and weight traffic between different services

# SMI: CNCF Donation

**A proposal has been made to donate the Service Mesh Interface specification to the CNCF sandbox!**

## Service Mesh Interface (SMI) proposal #337



lachie83 opened this issue 5 days ago · 9 comments · May be fixed by [#336](#)



lachie83 commented 5 days ago

+ 😊 🚩 Tip ...

We would like to propose [Service Mesh Interface \(SMI\)](#) specification for donation as a sandbox project in the CNCF.

The SMI proposal was presented during the CNCF sig-network, January 16, 2020. The proposal deck can be found [here](#).

Please find associated proposal PR - [#336](#)



10



11



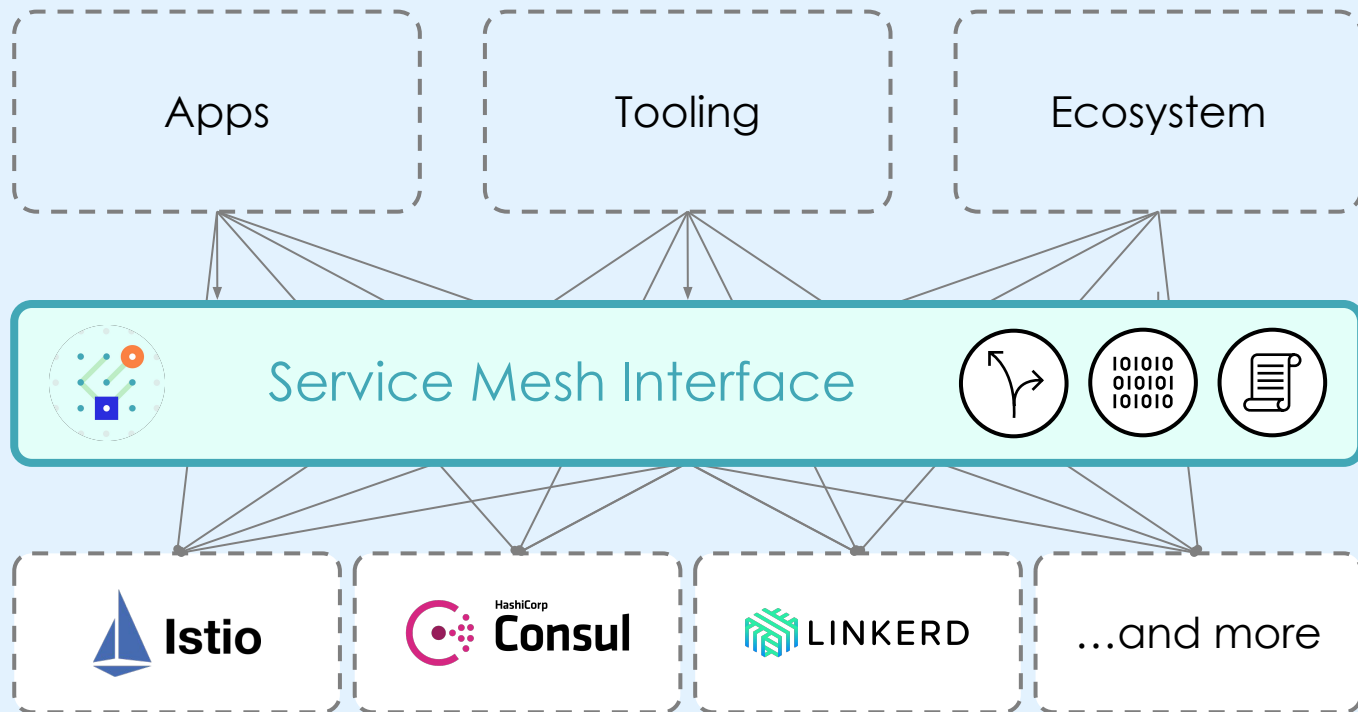
9



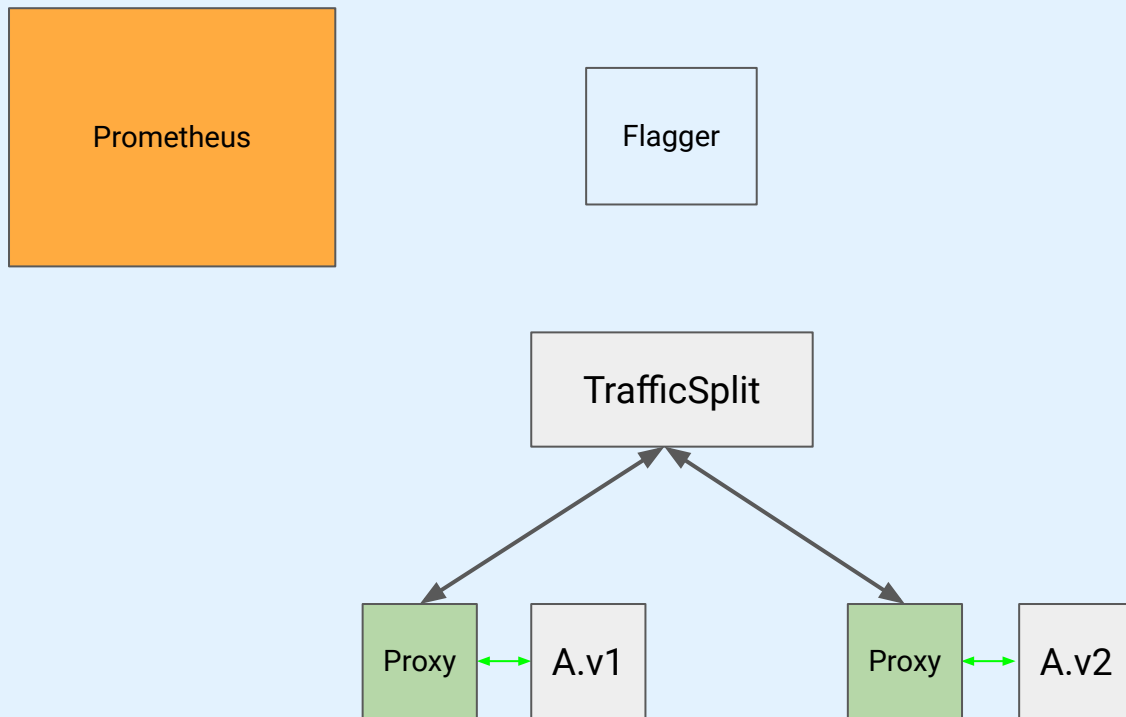
8

<https://github.com/cncf/toc/issues/337>

# Service Mesh Interface

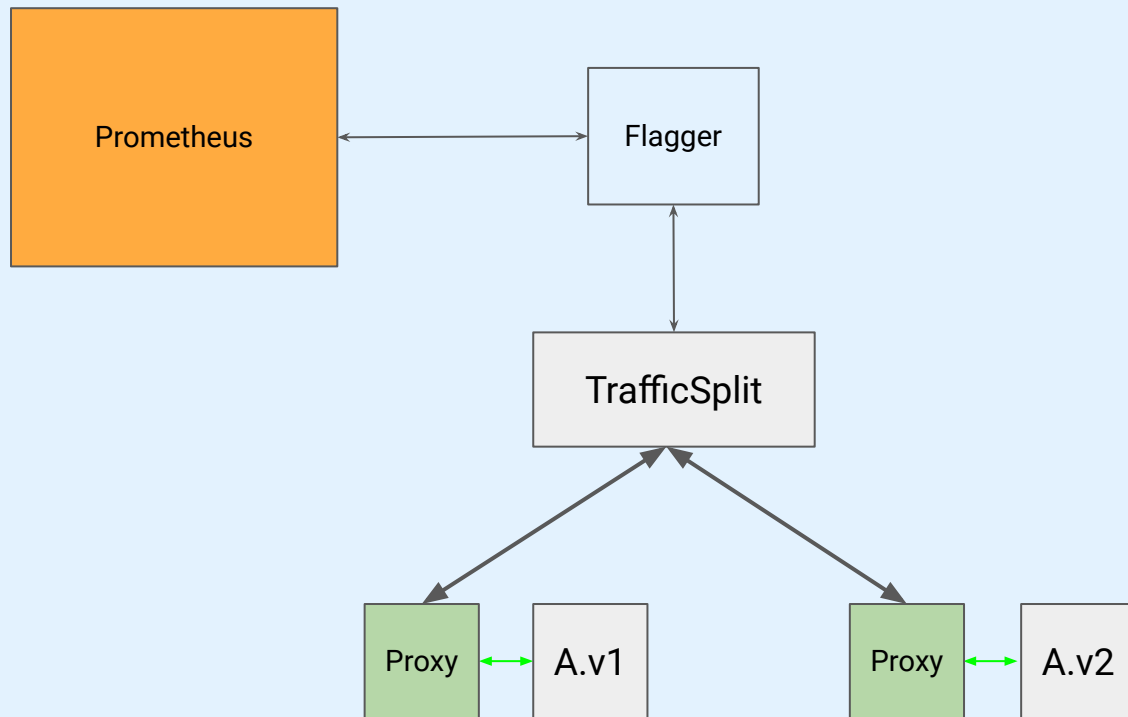


# Flagger Architecture

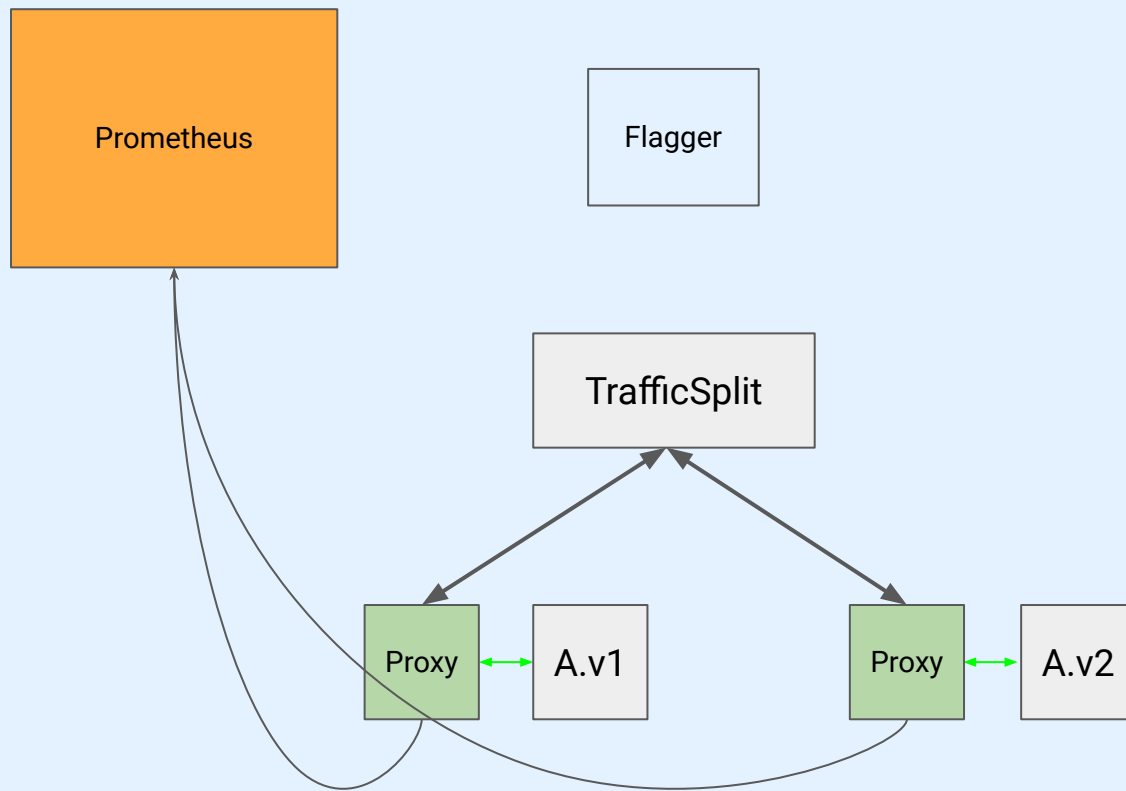




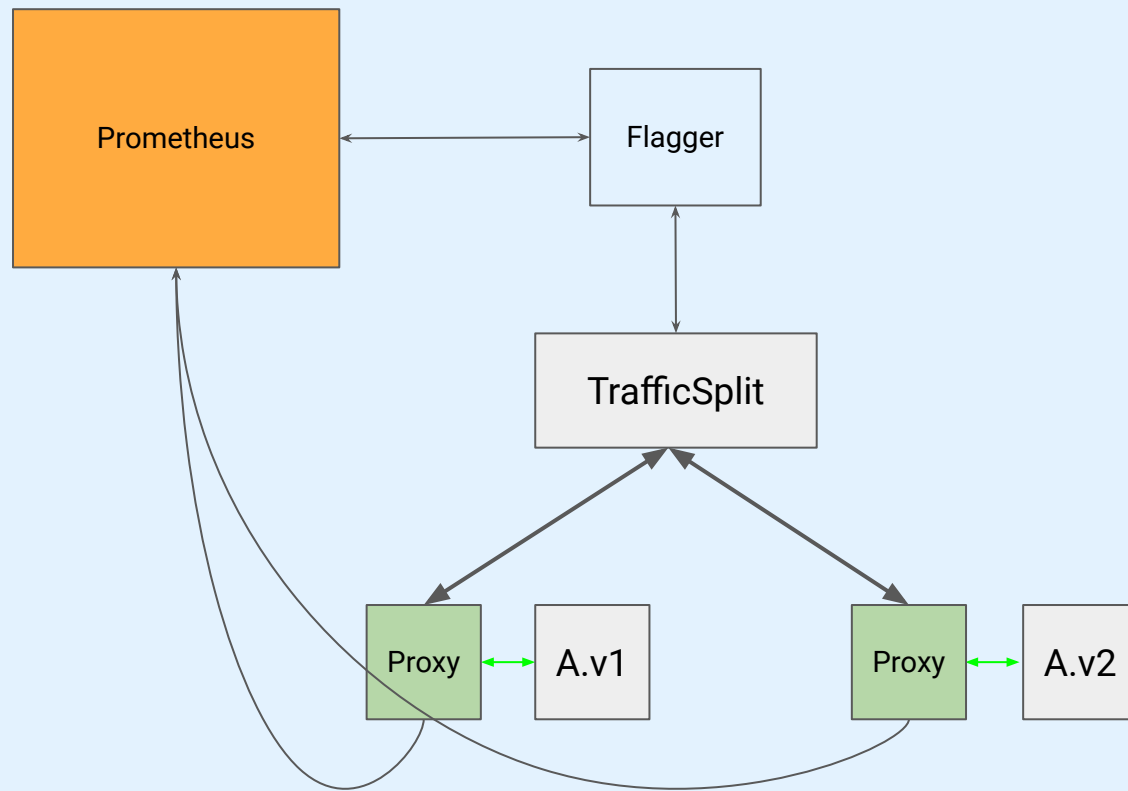
# Flagger Architecture



# Flagger Architecture



# Flagger Architecture





Q & A



Join our community!



[github.com/linkerd](https://github.com/linkerd)



[slack.linkerd.io](https://slack.linkerd.io)



[@linkerd](https://twitter.com/linkerd)

FROM YOUR FRIENDS AT



BUOYANT