
KubeCon
CloudNativeCon
North America 2023


KubeCon | CloudNativeCon

North America 2023

Service Mesh Journey at DoorDash: The Good, the Bad, and the Ugly

Hochuen Wong
DoorDash

Hochuen Wong
Software Engineer, Doordash

Doordash:

- Joined Core Infrastructure Org in 2020
- Compute and Traffic Infrastructure Teams

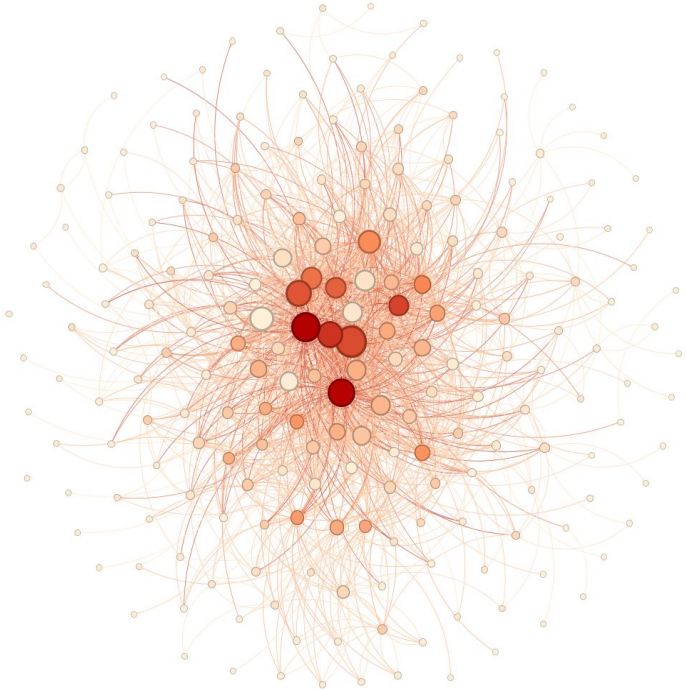
Prior to Doordash:

- Distributed systems for Machine Learning

Contact:

- https://www.linkedin.com/in/hochuenw/

Why Service Mesh?

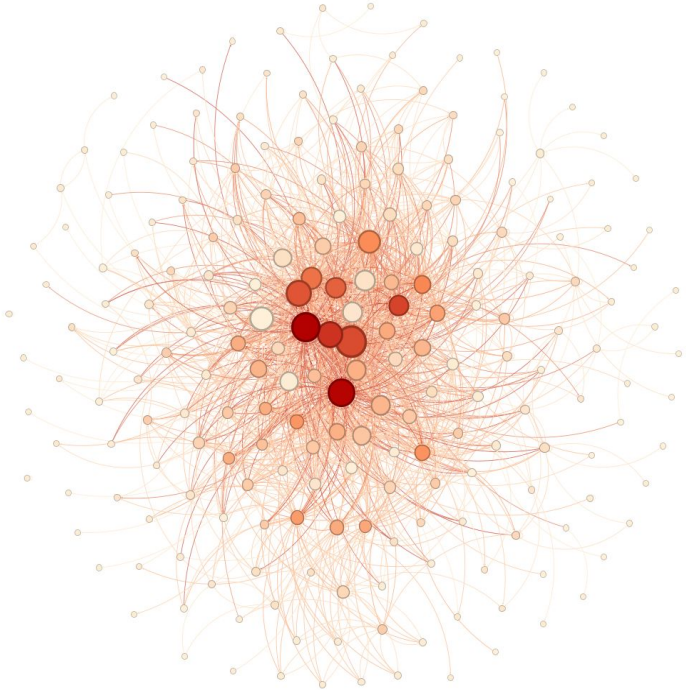


DoorDash service graph in 2023

WHY SERVICE MESH?

Move to microservices

- Started microservice journey in 2019
- 50+ microservices by the end of Q1 2021
- Classic microservice challenges:
  - Observability and debugging
  - Service to service communication
  - Authentication and authorization
  - Standardization
  - ...



DoorDash service graph in 2023

WHY SERVICE MESH?

Move to microservices

- Started microservice journey in 2019
- 50+ microservices by the end of Q1 2021
- Classic microservice challenges:
  - Observability
  - Networking
  - Debugging
  - Authentication and authorization
  - Standardization
  - ...
- **Service Mesh journey began in Q2 2021**

Initial Design

Requirements

SCALABILITY

- Support the largest Kubernetes cluster with 2000+ nodes

FLEXIBILITY

- Support DoorDash Consul-based multi-cluster architecture

MATURITY

- Successful user stories
- Community

CONFIGURATION & DOCUMENTATION

- Easy to configure
- Documentation

FEATURES

- Observability
- Security
- Reliability
- Traffic Management

Requirements

SCALABILITY

- Support the largest Kubernetes cluster with 2000+ nodes

FLEXIBILITY

- Support DoorDash Consul-based multi-cluster architecture

MATURITY

- Successful user stories
- Community

CONFIGURATION & DOCUMENTATION

- Easy to configure
- Documentation

FEATURES

- Observability
- Security
- Reliability
- Traffic Management

Requirements

SCALABILITY

- Support the largest Kubernetes cluster with 2000+ nodes

FLEXIBILITY

- Support DoorDash Consul-based multi-cluster architecture

MATURITY

- Successful user stories
- Community

CONFIGURATION & DOCUMENTATION

- Easy to configure
- Documentation

FEATURES

- Observability
- Security
- Reliability
- Traffic Management

Requirements

SCALABILITY

- Support the largest Kubernetes cluster with 2000 nodes

FLEXIBILITY

- Support DoorDash Consul-based multi-cluster architecture

MATURITY

- Successful user stories
- Community

CONFIGURATION & DOCUMENTATION

- Easy to configure
- Documentation

FEATURES

- Observability
- Security
- Reliability
- Traffic Management

Requirements

SCALABILITY

- Support the largest Kubernetes cluster with 2000 nodes

FLEXIBILITY

- Support DoorDash Consul-based multi-cluster architecture

MATURITY

- Successful user stories
- Community

CONFIGURATION & DOCUMENTATION

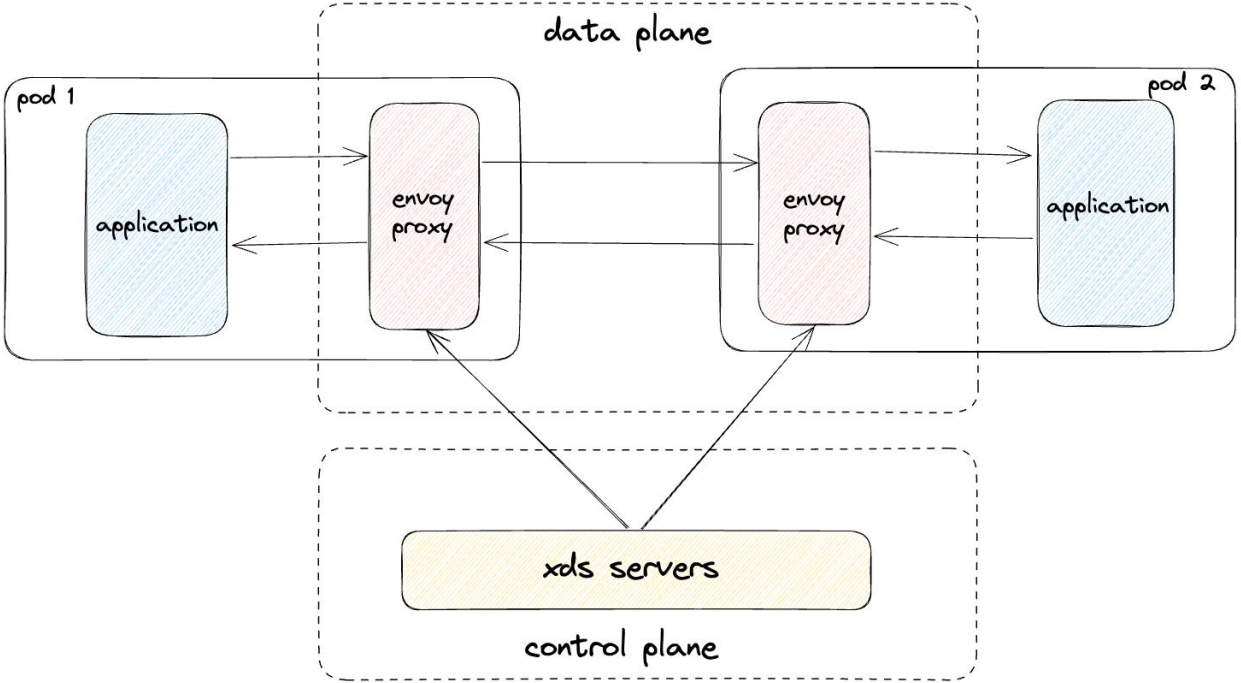
- Easy to configure
- Documentation

FEATURES

- Observability
- Security
- Reliability
- Traffic Management

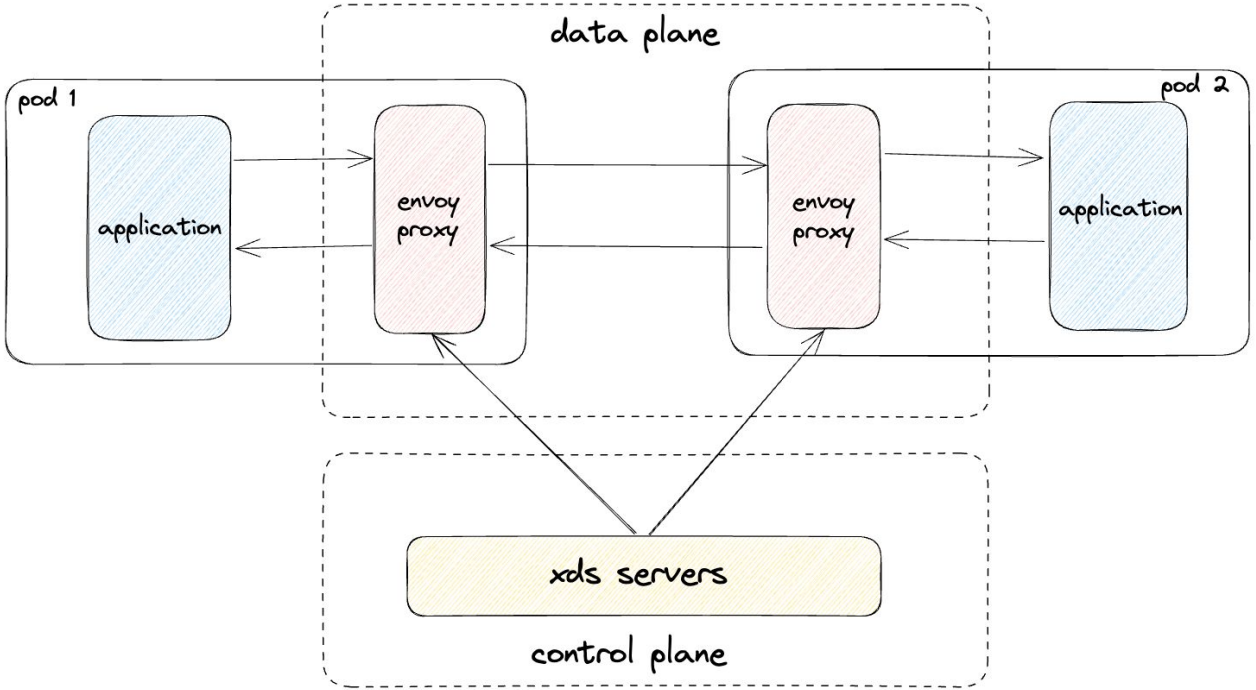
Architecture

- K8s mutating webhook
- iptables
- Data Plane
  - Envoy sidecar container
  - HTTP1/HTTP2/gRPC
- Control plane
  - Custom version



Architecture

- Data Plane
  - Envoy sidecar container
- Control plane
  - Custom version
  - Consul for EDS
  - K8s custom resource for other resources
- Adoption
  - Onboard everyone
  - Minimum features

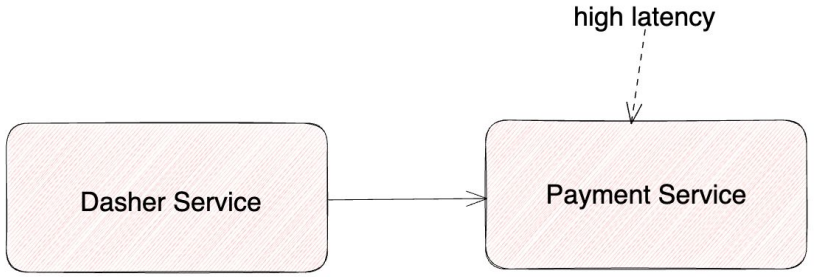


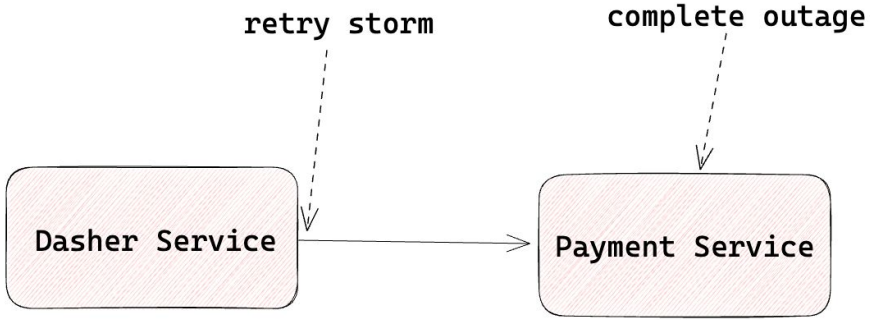
Sitewide Outage

SITE-WIDE OUTAGE

What happened

- **CASCADING FAILURE (RCA)**

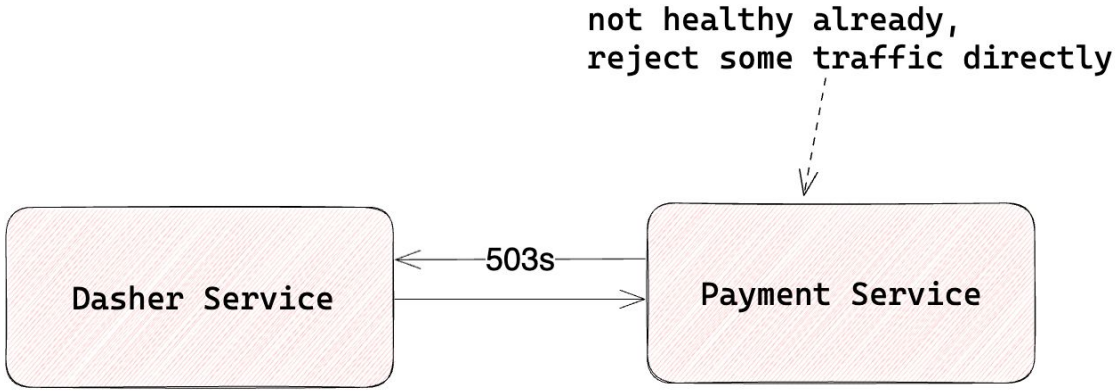




SITE-WIDE OUTAGE

What should have happened

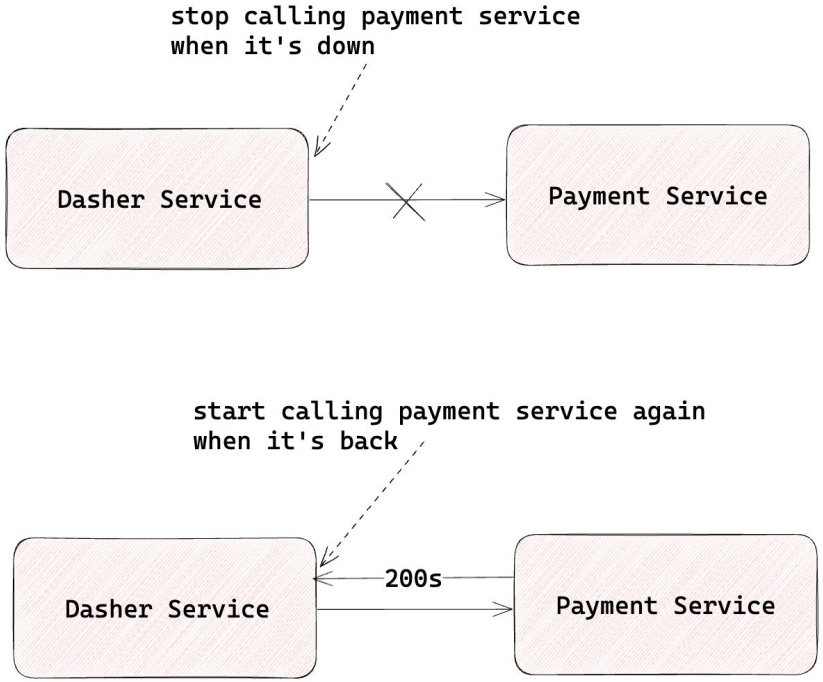
- **LOAD SHEDDING**
- CIRCUIT BREAKING



SITE-WIDE OUTAGE

What should have happened

- LOAD SHEDDING
- CIRCUIT BREAKING



SITE-WIDE OUTAGE

Code freeze.

SITE-WIDE OUTAGE

Code freeze.

A project called Service Mesh?

- Just reviewed the initial design doc
- No operational experience with running Envoy
- No control plane

SITE-WIDE OUTAGE

Code freeze.

A project called Service Mesh?

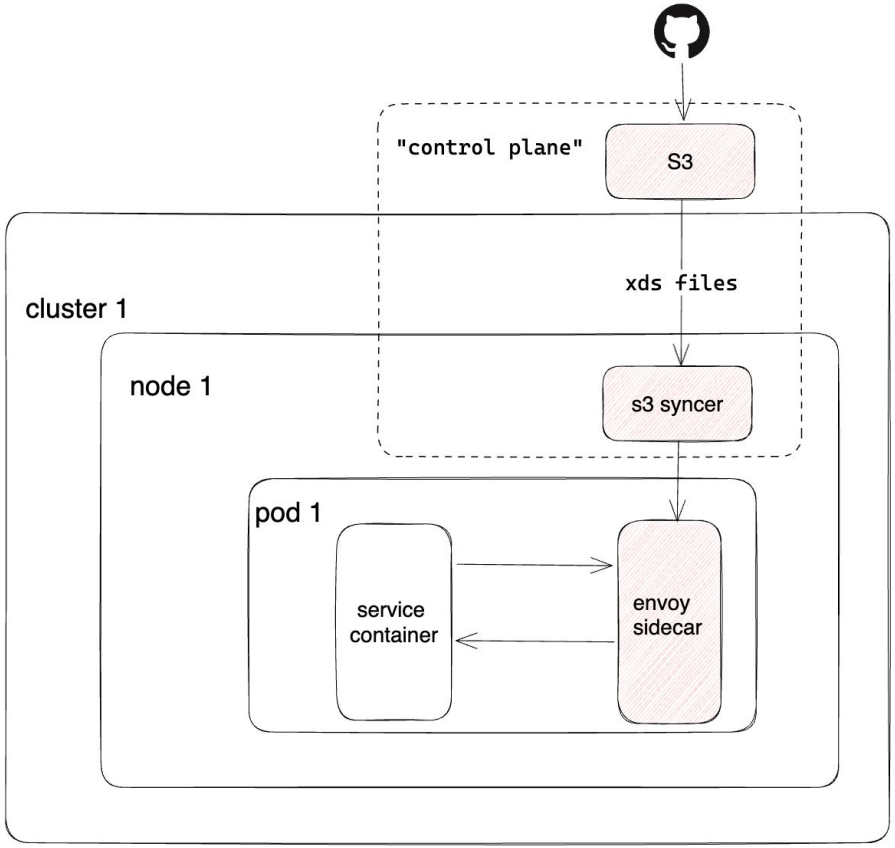
Shift Priority !

Design after the outage

Design after the outage

Implementation

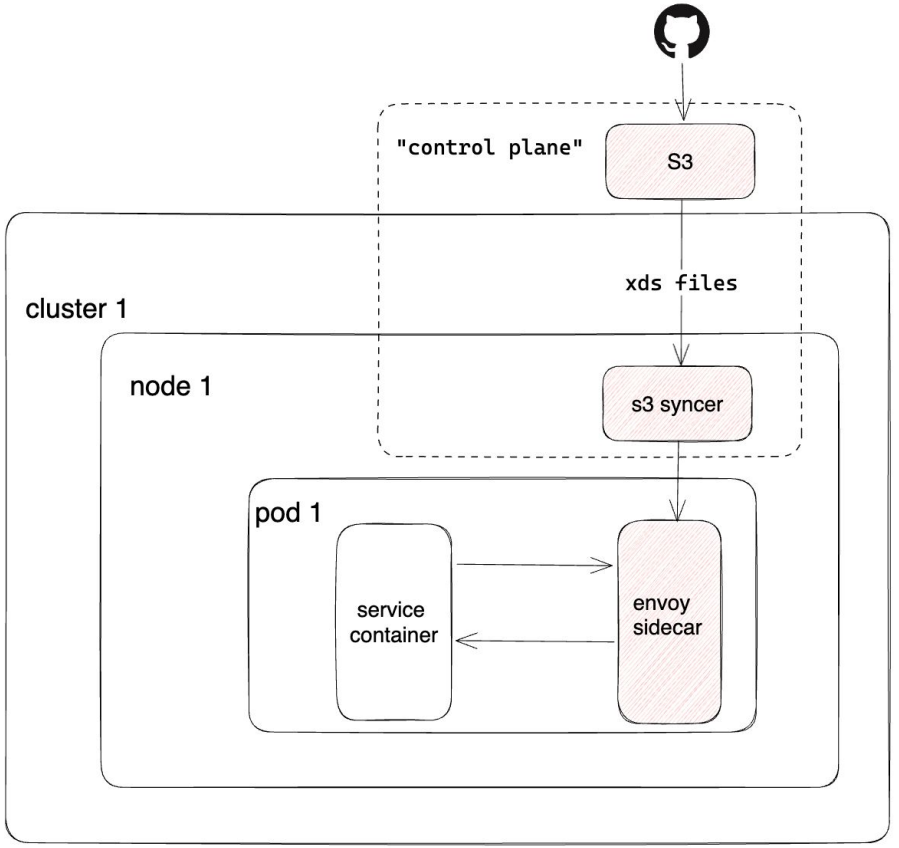
- Envoy sidecar injector & iptables
- Configuration
  - File-based dynamic configurations
  - HTTP passthrough proxy
- Features
  - Adaptive concurrency (load shedding)
  - Outlier detection (circuit breaking)



INITIAL VERSION

A project called ~~Service Mesh~~ Envoy Sidecar

- Primitive configuration management
- Focus on the data plane and two reliability features



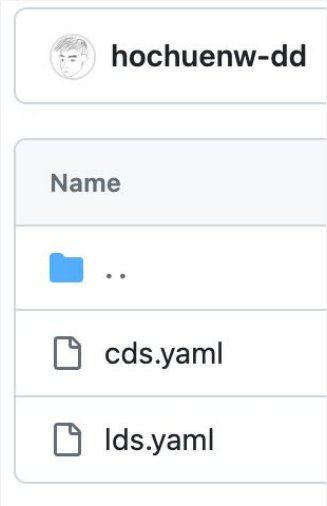
Onboarding First Customers

ONBOARDING FIRST CUSTOMERS

Steps to onboard one service

- label namespace
- label deployment
- create RAW(!) Envoy configurations (1K+ lines)

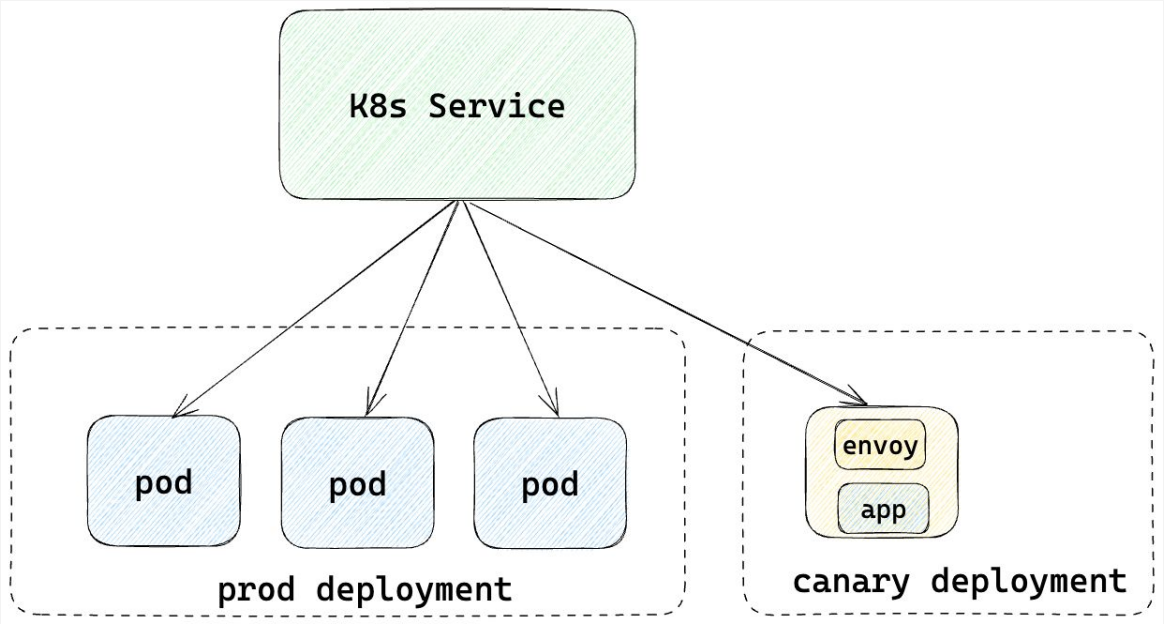
```
labels:
  envoy-sidecar-injector.doordash.com/enabled: "true"
```



ONBOARDING FIRST CUSTOMERS

Rollout Strategy

- Canary deployment
- Two week bake-in period



ONBOARDING FIRST CUSTOMERS

Did we make the right decision?

LEARNINGS

- ✔ Dream Big, Start Small
- ✔ Solve real-world problems

General Availability!

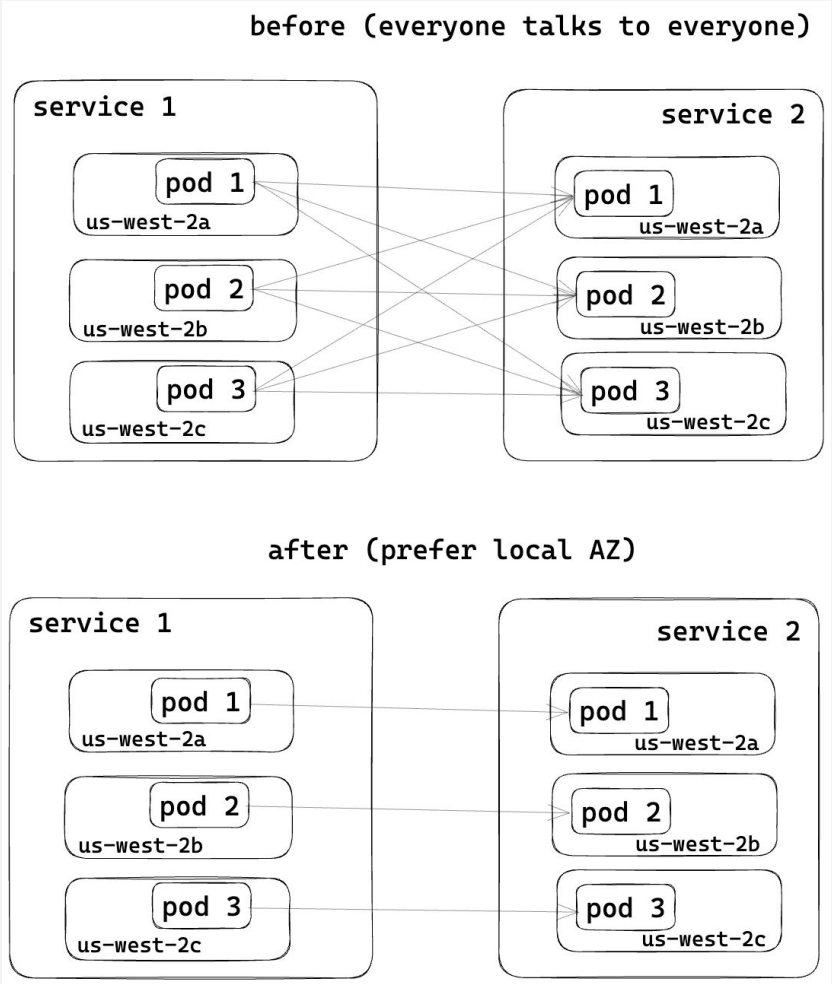
WHAT WE HAVE IMPROVED
SINCE ONBOARDING FIRST CUSTOMERS

- Configuration management
  - Templated Envoy configurations
- Observability
  - Common observability dashboard
  - Common alert module and runbook
- Operational experience
  - Support for more programing languages
  - Onboard more services

More features

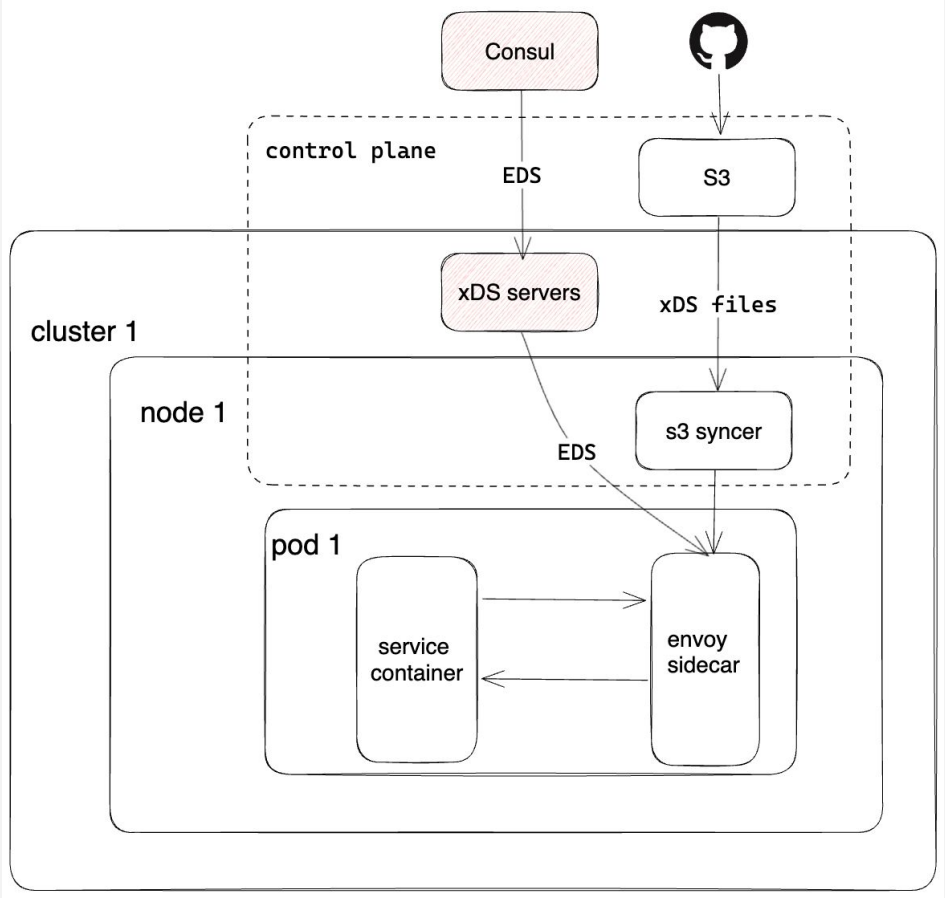
More features

- Zone-aware Routing
  - Route egress traffic to its local zone
  - Ensure traffic is still balanced
  - Save data transfer cost
  - Reduce impact of AZ outage
  - Performance



More features

- Zone-aware Routing
  - Evolved configuration management systems
  - Introduced API-based dynamic configurations



More features

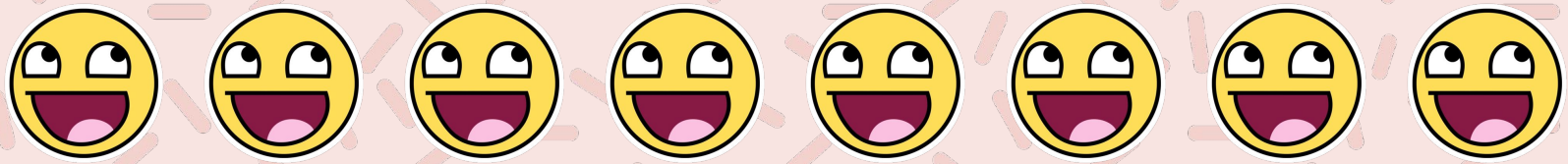
- There are many more feature request
  - Zone-aware Routing
  - Client-side load balancing
  - Sandbox traffic routing
  - Slow Start Mode
  - Traffic redirection
  - Global rate limiting
  - ...
- Deeper understanding of customer pain points

LEARNINGS

✔ Co-develop your solution with initial customers

Adopt 100 Services!

Four more years!



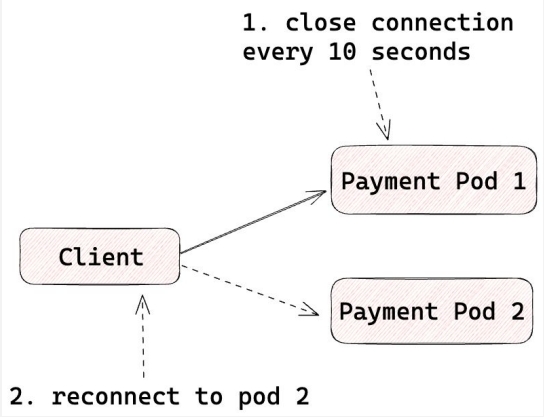
Q4 2022: Prepare for large scale onboarding

Prepare for large scale onboarding

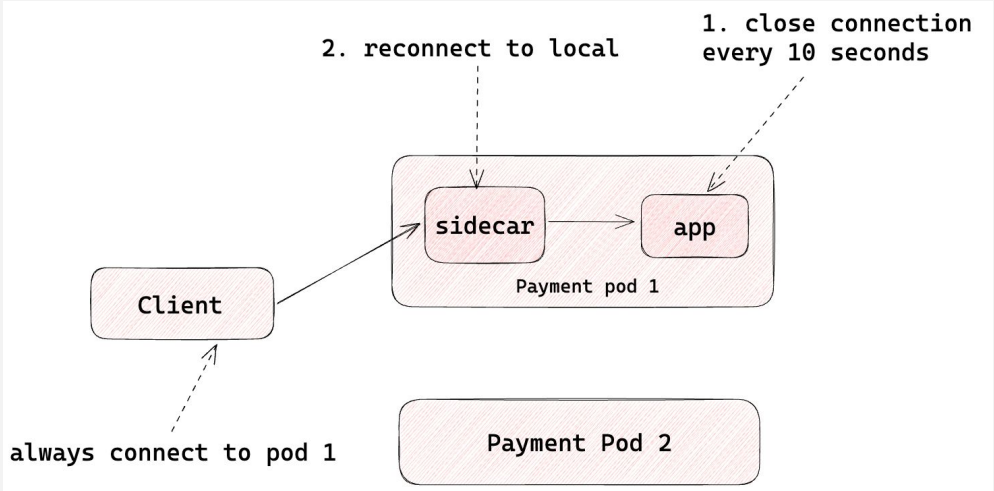
Unknowns discovered

- Unnoticed client behaviors became apparent
- First consumer payment service as an example

Before



After



Prepare for large scale onboarding

Unknowns discovered

- There are many more examples...
- But it slowed down as more services being onboarded

LEARNINGS

- ✔ Envoy sidecar is not transparent
- ✔ Expect the unexpected
- ✔ Take a bet at the right time

ACTION ITEMS

- ✔ Switch to use gradual rolling update approach

Prepare for large scale onboarding

Challenges in Developer Experience

✘ The manual onboarding process and Envoy resource tuning, led by developers, required a learning curve

LEARNINGS

✔ Decentralized onboarding and manual Envoy resource tuning doesn't scale

ACTION ITEMS

✔ Infra team owns onboarding and Envoy resource management

✔ Streamline the onboarding process

✔ Pre-generate all Envoy configs and labels

Prepare for large scale onboarding

Challenges in Developer Experience

✘ Caused confusion:

- Onboarding status
- Envoy configs
- Architecture

ACTION ITEMS

✔ Invest in educating/enabling service owners

Prepare for large scale onboarding

Challenges in Developer Experience

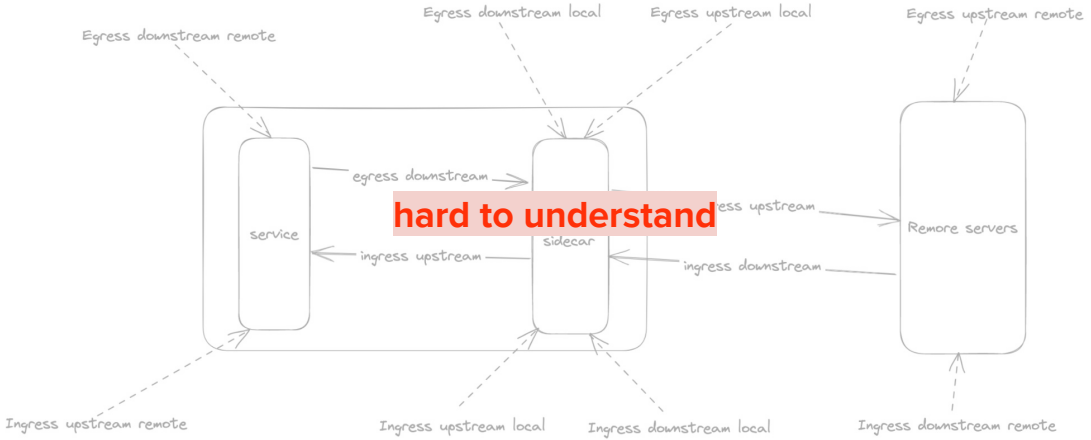
✘ Complexity of observability features

LEARNINGS

✔ Have a product mindset / customer obsessed

ACTION ITEMS

✔ Simplify the observability dashboards



Prepare for large scale onboarding

Challenges in Developer Experience

✘ Debugging challenges

ACTION ITEMS

✔ Availability SLOs

✔ Introducing distributed tracing with OpenTelemetry

Prepare for large scale onboarding

Challenges in Developer Experience

- ✘ Many features are only enabled when egress clusters are defined
- ✘ Building an accurate service graph is not trivial

ACTION ITEMS

- Get accurate service graph from tracing data
- Build a tool to generate egress configs

LARGE SCALE ONBOARDING

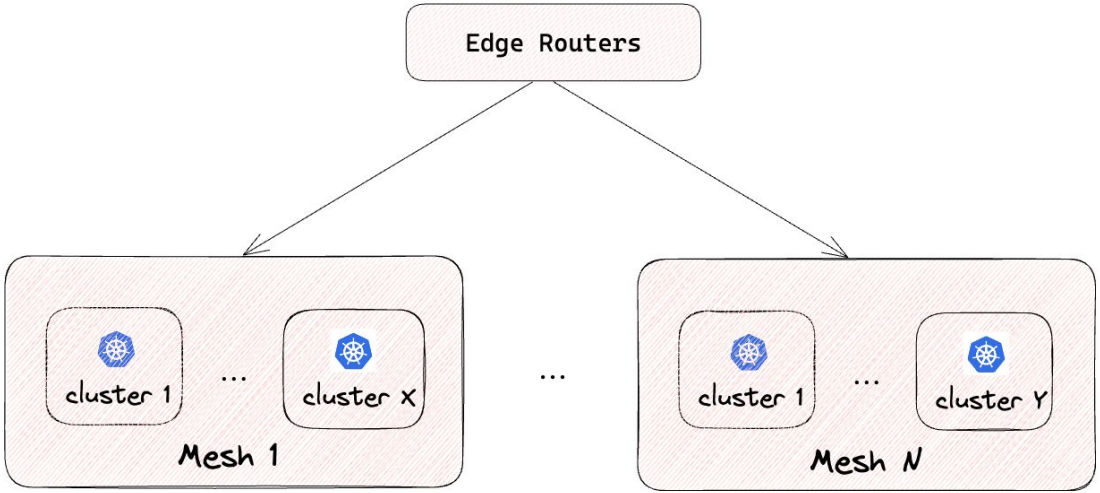
Large scale adoption

- Onboarding was much faster
- Uncovered a few unknowns
- Manageable additional maintenance responsibilities for the infrastructure team
- Features are widely adopted

Current State

WHERE WE'RE AT TODAY!

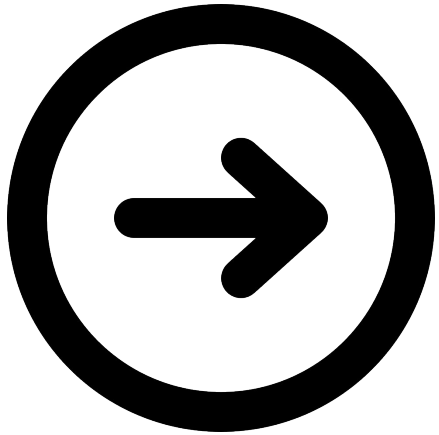
- 10 production Kubernetes clusters
- 500+ microservices
- 5 isolated mesh
- Multiple clusters with one mesh
- 10K pods
- 5M RPS



Current and future work

WHAT'S NEXT!

- Developer Velocity
  - Improve configuration management experience
  - Unify the experience of using observability features
- Efficiency
  - Compute
  - Metrics
- Traffic Simplification
  - Refactor multi-cluster service discovery infrastructure
- Feature enrichment



Recap!

BEFORE YOU START!

- ✔ Understand your requirements and use cases
- ✔ Co-develop your solution with initial customers

WHEN WORKING WITH ENVOY

- ✔ Sidecar is not always transparent
- ✔ Expect the unexpected

Recap (continued)!

WHEN ONBOARDING SERVICES!

- ✔ Test things gradually at the beginning
- ✔ Make well-informed bets (at the right time)
- ✔ Decentralized onboarding doesn't scale
- ✔ Streamline/Automate the onboarding process

WHEN DELIVERING PRODUCT TO THE REST OF THE TEAM

- ✔ Envoy metrics can overwhelm service owners
- ✔ Have a product mindset / customer obsessed
- ✔ Invest in training/enabling service owners
- ✔ Increase velocity through more user friendly solutions

Thank you

Please scan the QR Code above to leave feedback on this session