The Good, the Bad, and the Meshy

Joe Searcy / T-Mobile

#IstioCon

IstioCon

@phenixblue

@jmsearcy

Who am I?

- **Staff Engineer** @ T-Mobile
- **Kubernetes** User since v1.4
- **Istio** User since v0.8.0
- **Kubernetes Community** Member/Contributor
- **Co-organizer** for Kubernetes Atlanta Meetup
- **Maintainer** of magtape.io

Part of a team

TKE Core Team
handles core Kubernetes platform

Platform Acceleration Team
Handles Meshy type things

Platform Shared Services Team
Handles shared tooling/observability

API Center for Enablement
Handles API and Service standards

Backstory

Cluster Growth

Complex App Landscape

Platform Diversity

Security

Simplification

Resiliency

Goals

Part of the platform

Lifecycle Automation

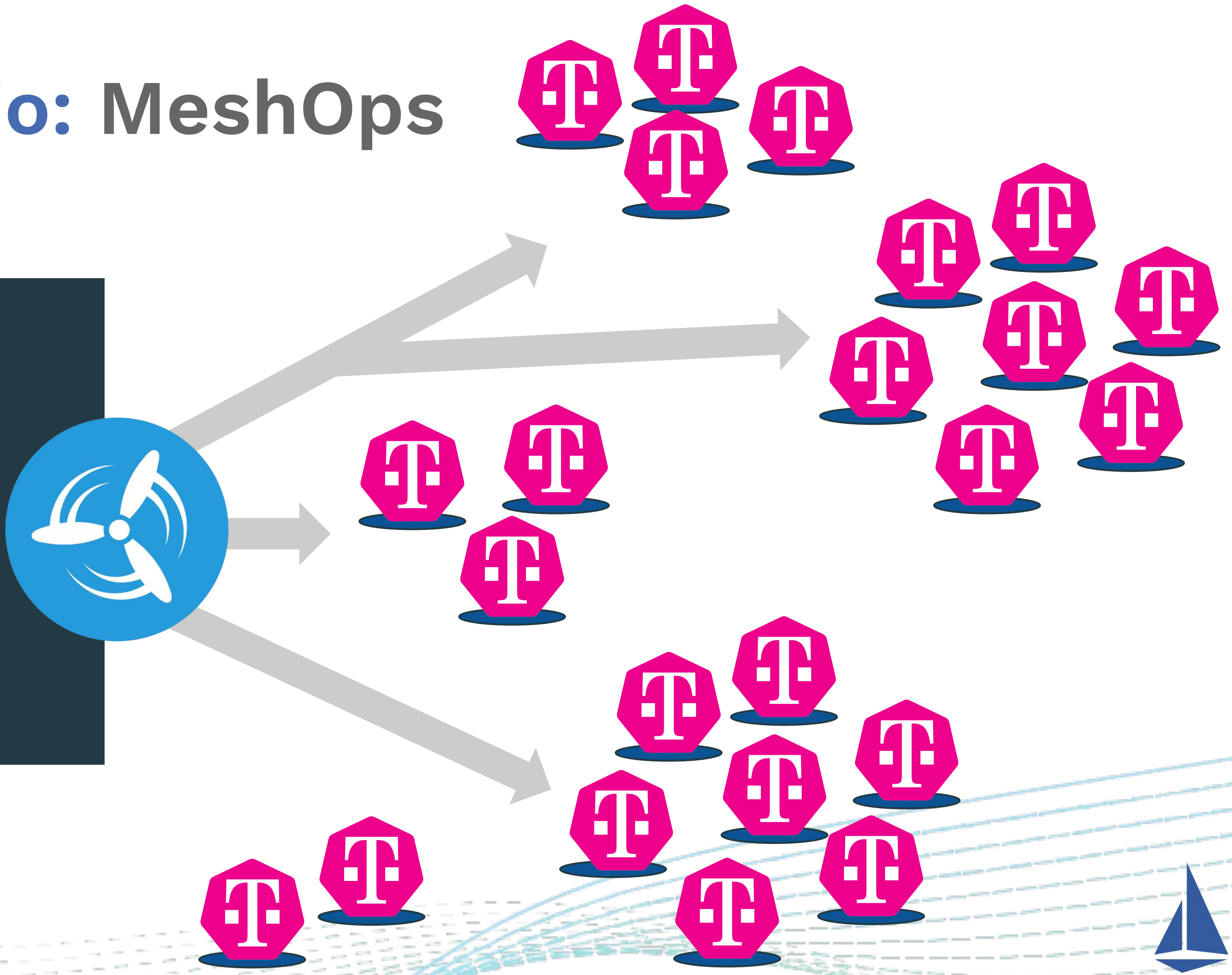
Opt-in UX

Mix of Mesh/Non-Mesh

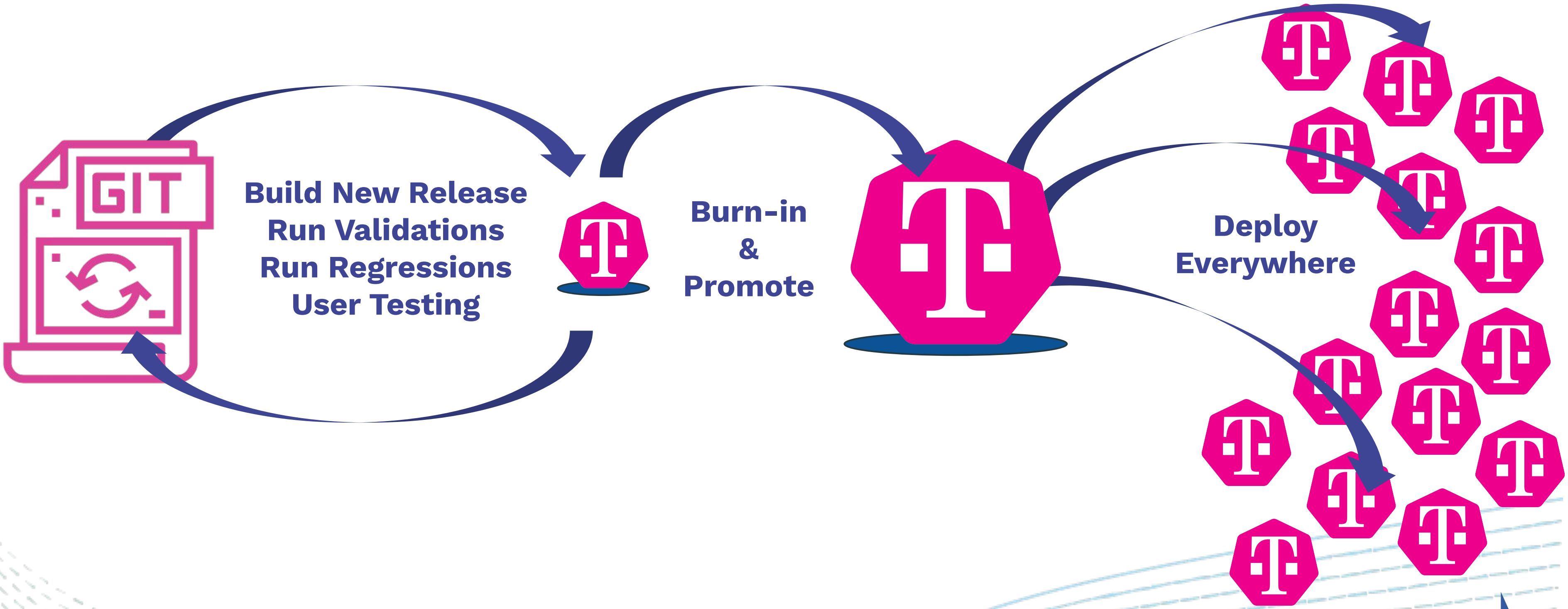
Iterative Adoption

How we Istio: MeshOps

```
component:
 name: istio
instances:
 global: "v1.7.7"
 environment:
    - sandbox: "v1.8.3"
    - non-production: "v1.8.2"
 region:
    - region1: skip
 clusters:
    - sandbox-cluster1: "v1.9.0"
```



How we Istio: Lifecycle Management

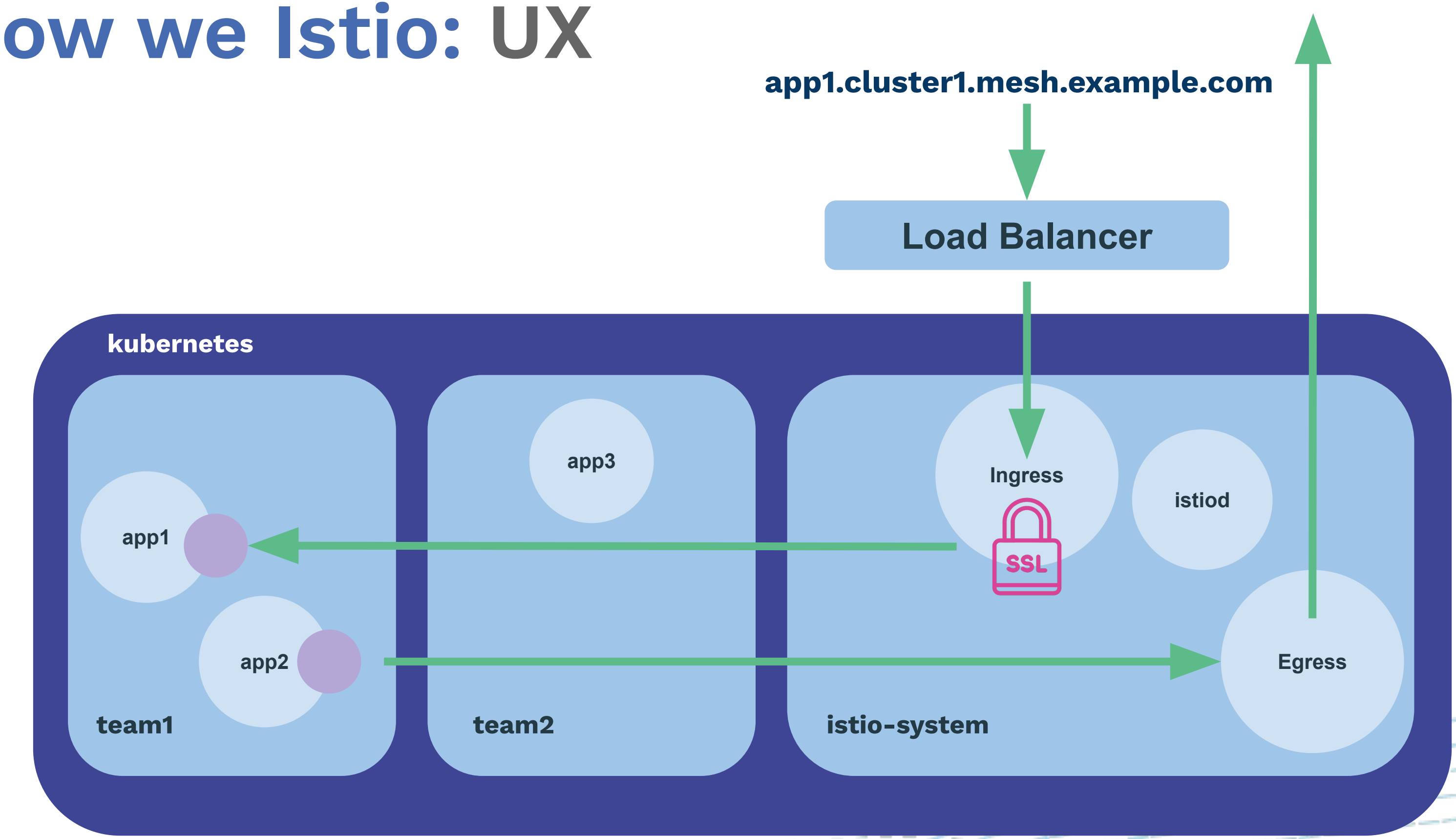


Build New Release
Run Validations
Run Regressions
User Testing

Burn-in
&
Promote

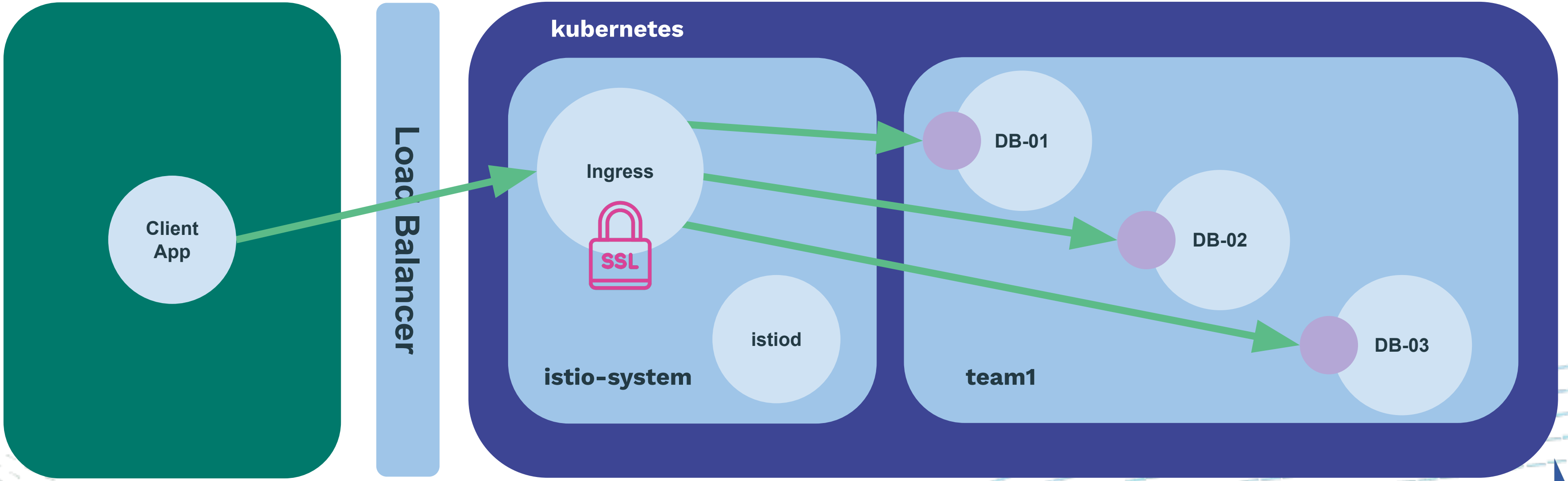
Deploy
Everywhere

How we Istio: UX

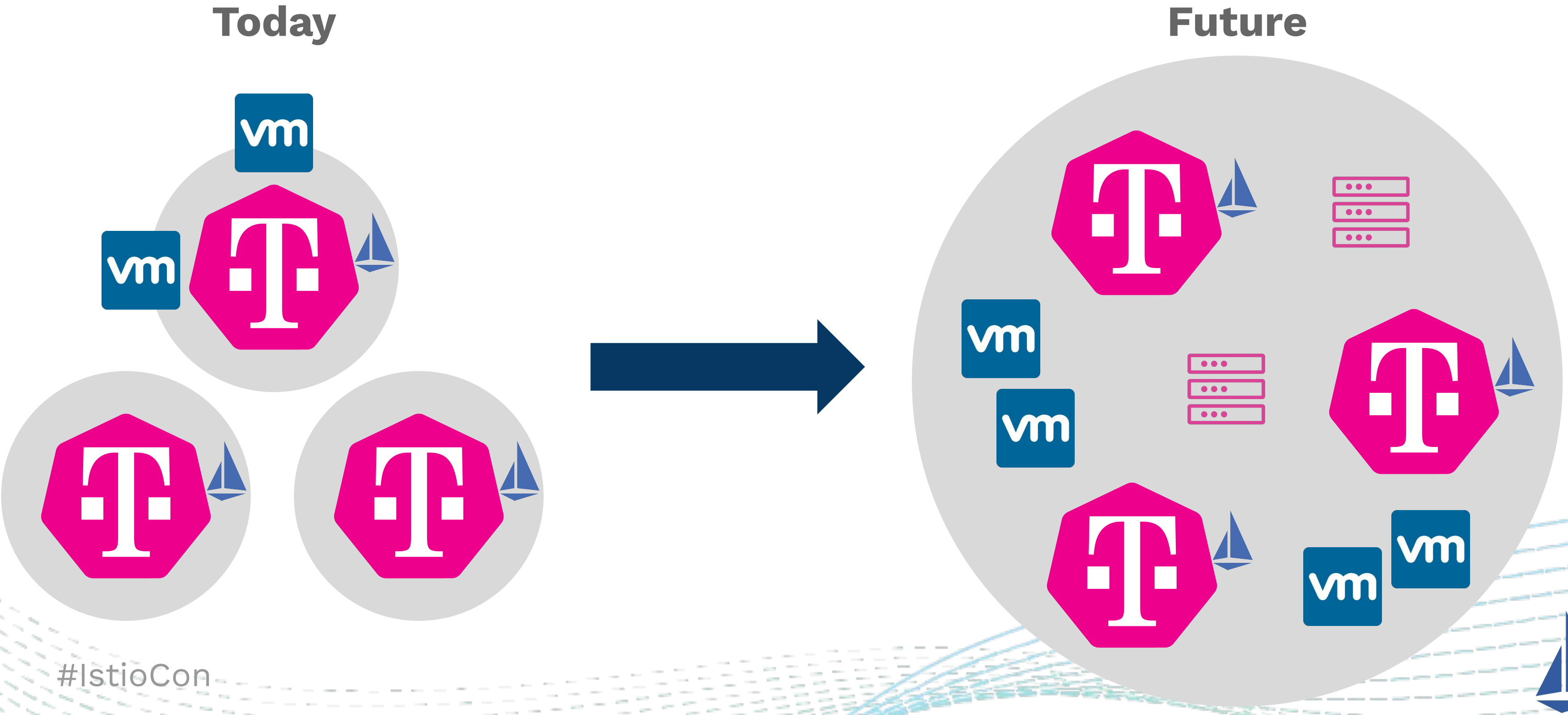


How we Istio: SNI Routing

db-01.cluster1.mesh.example.com:9042 → db-01.team1.svc.cluster.local:9042
db-02.cluster1.mesh.example.com:9042 → db-02.team1.svc.cluster.loca:9042
db-03.cluster1.mesh.example.com:9042 → db-03.team1.svc.cluster.loca:9042



Down with the Meshness



Today

Future

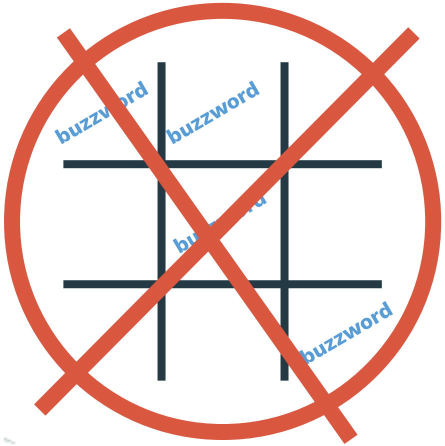
Ok...let’s get past the 🌈’s and 🦄’s

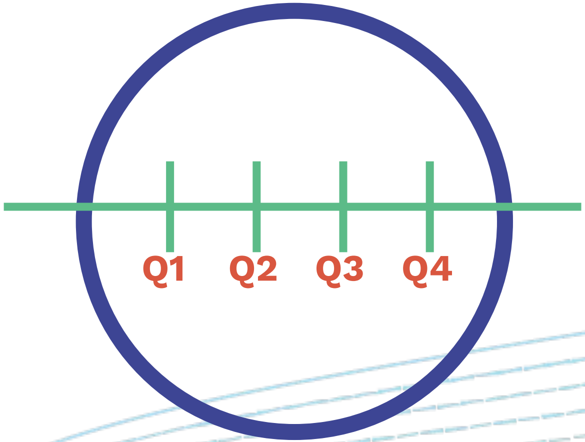
Adopting Istio enterprise wide didn’t come without headaches

- Overall complexity
- Stability of releases
- Consistency in the API’s
- Lack of hard...ok...even soft multi-tenancy
- Release/Support Cadence

What we've learned: Complexity

- Drive adoption based on need…not buzzword bingo/FOMO
- Define easy to consume patterns for your users
- Iterate on implementation…don't try to do all of the things!



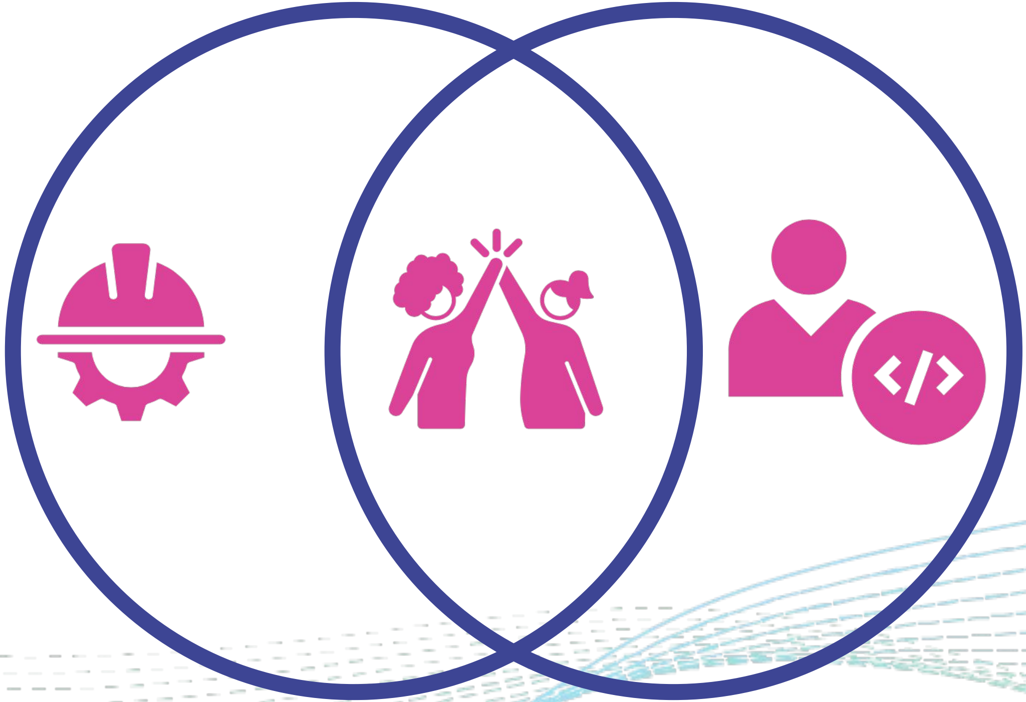


What we’ve learned: Stability

- Automate everything from the start
- Test....no really, test!
- Capture your failures
- Contribute

What we’ve learned: Usability

- Embed Platform teams with your Development teams



What we’ve learned: Risk Isolation

- Multiple control planes for Multi-Cluster Mesh
- Neighbors can be really noisy!
- Policy-as-code can help

Winning at this Mesh thing!

Observability (Metrics, Tracing, Visualization)

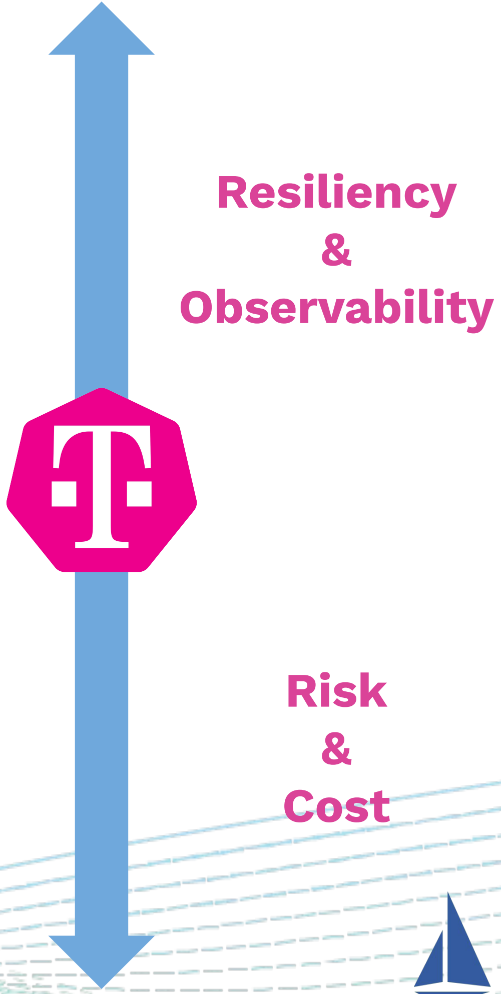
Security (mTLS, Service-to-Service AuthN/AuthZ)

Application Simplification (Focus on Business Logic)

Envoy Competency (Filters & WebAssembly)

50% reduction in MTTR

Savings in excess of 100,000 Engineering hours



Resiliency & Observability

Risk & Cost

Thank you!

@phenixblue

@jmsearcy

https://opensource.t-mobile.com

#IstioCon