LINKERD

An open source **service mesh** and CNCF project.

- 🔥 **36+** months in production
- 🔥 **3,500+** Slack channel members
- 🔥 **10,000+** GitHub stars
- 🔥 **100+** contributors
- 🔥 **Weekly** edge releases
- 🔥 **6-8 week** stable release cadence

NORDSTROM CHASE

GoDaddy Walmart

STRAVA BIGCOMMERCE

Expedia OfferUp

ebay Cisco webex

COMCAST planet.

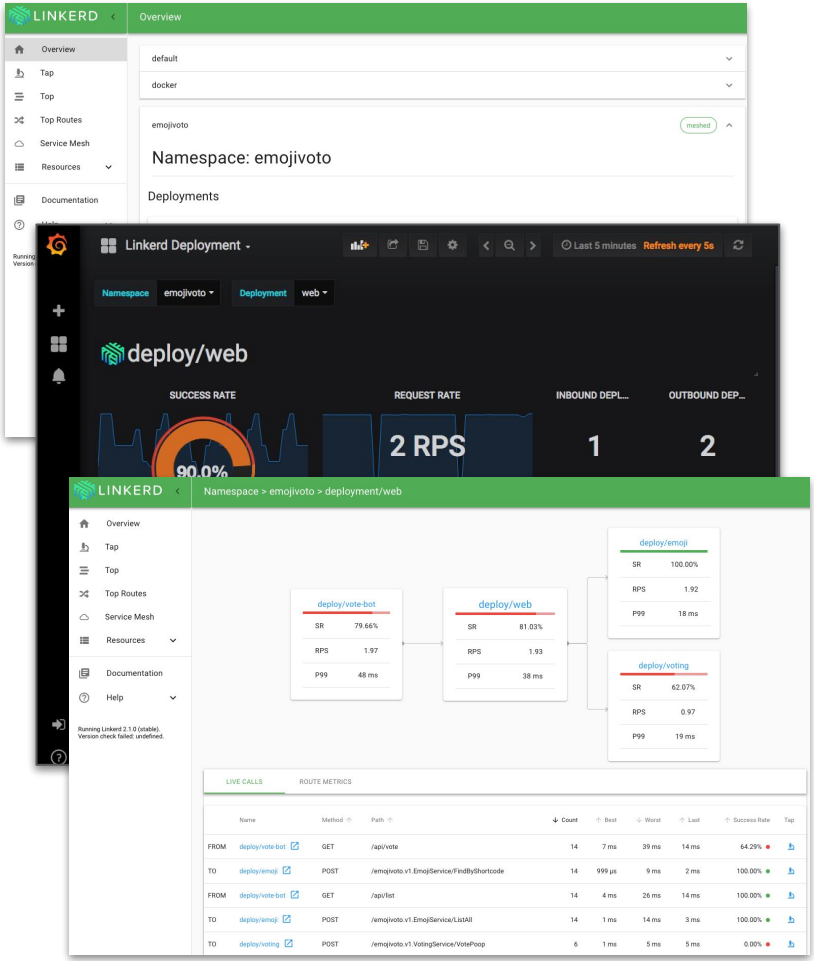
What does Linkerd do?

⚡ **Observability:** Service-level *golden metrics*: success rates, latencies, throughput. Service topologies.

⚡ **Reliability:** Retries, timeouts, load balancing, circuit breaking*

⚡ **Security:** Transparent mTLS, cert management and rotation, policy*

In an ultralight package focused on **operational simplicity** first and foremost.



Why should I care?

Linkerd gives **...**

... **platform owners** (SREs, architects) 👉 **Who?**

... the **observability**, **reliability**, and **security** primitives 👉 **What?**

... that are **critical** for cloud native architectures 👉 **Why?**

... with **no developer involvement**! 👉 **The magic**

Linkerd doesn't just solve technical problems, it solves **socio-technical problems**: by decoupling them developers, it gives platform owners control over their destiny. 💪

How is Linkerd designer?

In short, "do less, not more":

💪 **Just works:** Zero config, out of the box, for any Kubernetes app

💪 **Ultralight:** Introduce the bare minimum perf and resource cost

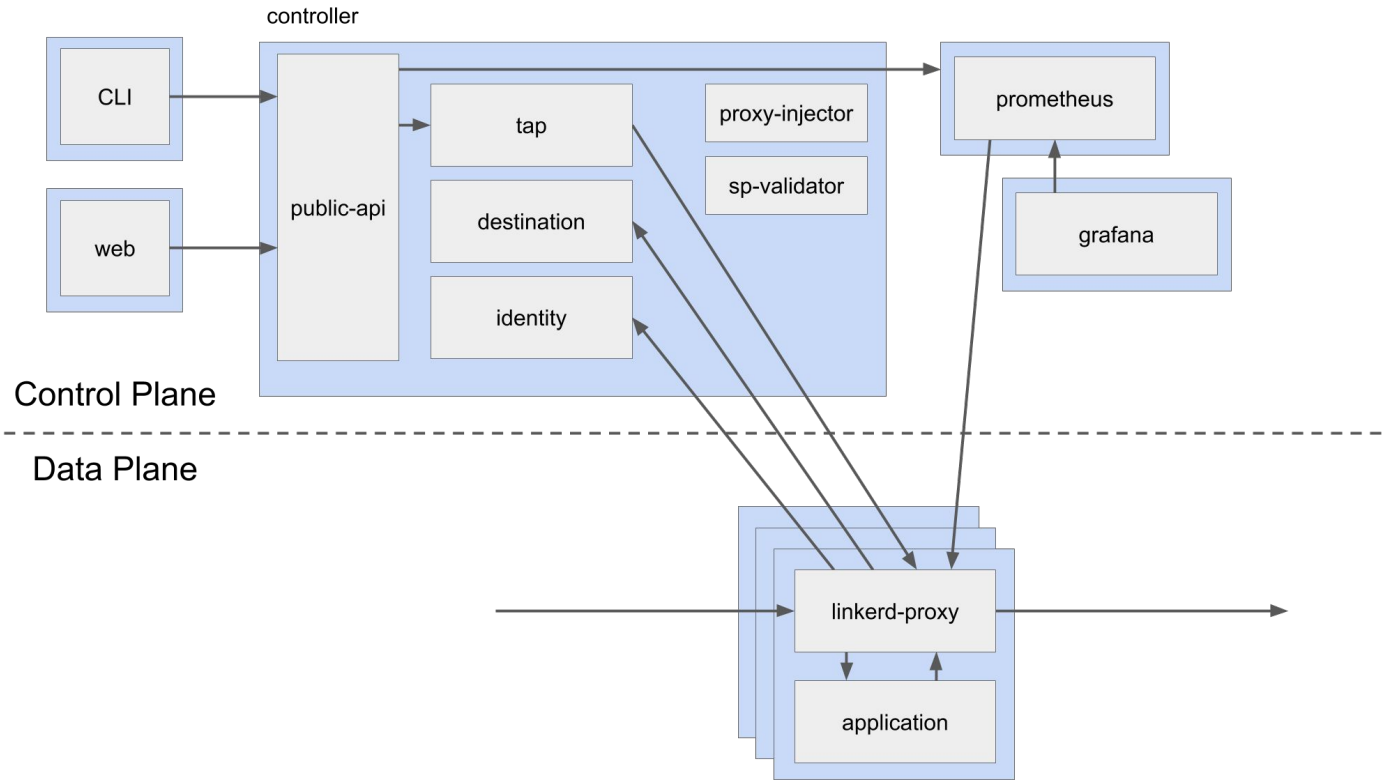
💪 **Simple:** Reduce operational complexity in every possible way

Control plane: Go. ~200mb RSS (excluding metrics data). (Repo: linkerd/linkerd2).

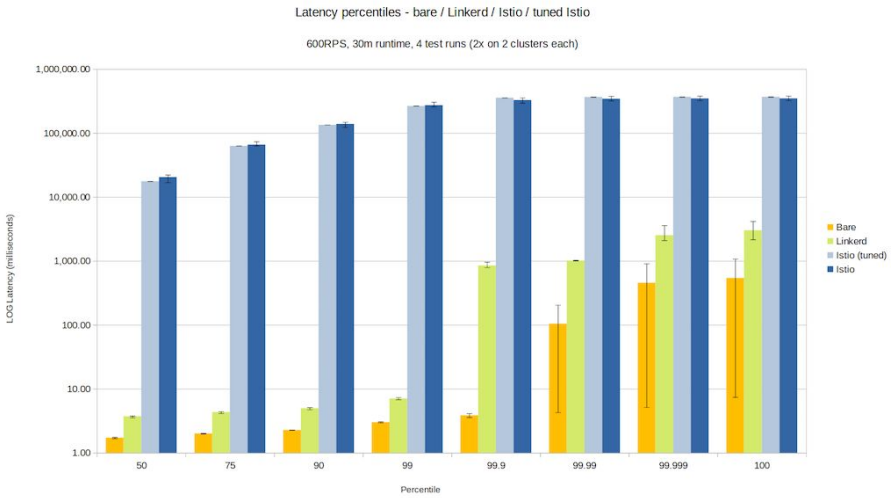
Data plane: Rust. <10mb RSS, <1ms p99 (!!!). (Repo: linkerd/linkerd2-proxy)

Background reading: Linkerd v2: How Lessons from Production Adoption Resulted in a Rewrite of the Service Mesh (InfoQ)

Linkerd 2.x architecture

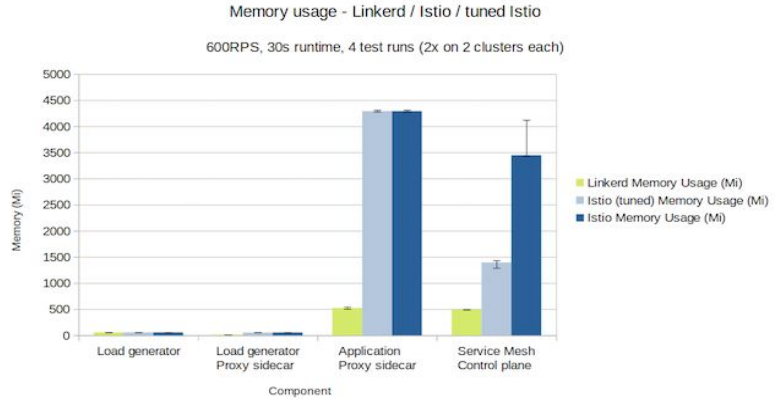


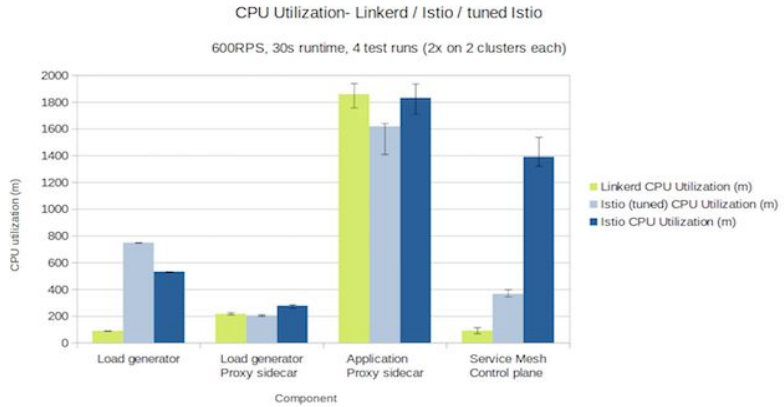
How fast/small is it?



Tl;dr: really fast. Not as fast as "no service mesh", but *significantly* smaller and faster than Istio.

Source: https://kinvolk.io/blog/2019/05/performance-benchmark-analysis-of-istio-and-linkerd/





Roadmap

As of 2.6 (Oct 2019):

- Distributed tracing, traffic shifting (blue/green, canaries), telemetry, retries, timeouts, proxy auto-injection, mTLS on by default for all HTTP, Helm charts, live header sampling that obeys RBAC, and much much more

On the roadmap for 2.7 (EOY 2019):

- mTLS for all all TCP traffic, thorough mTLS auditing, control plane cert rotation

2020:

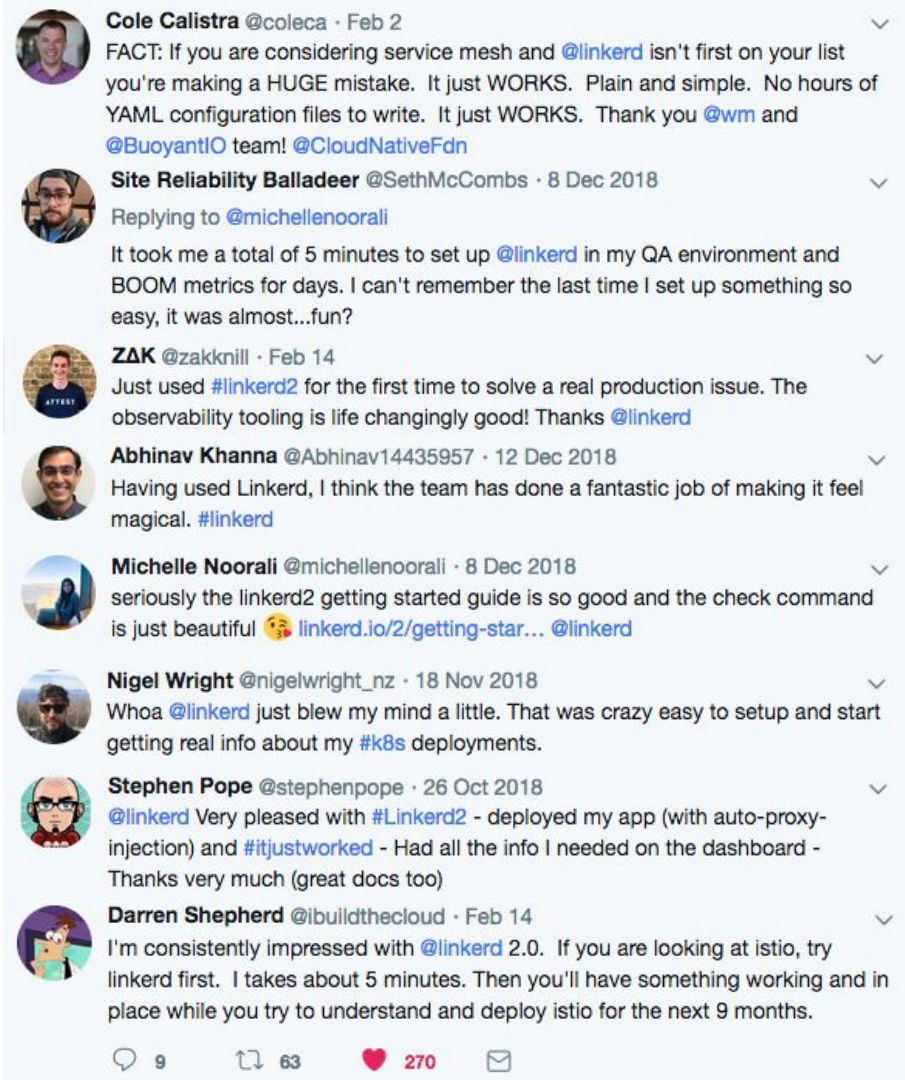
- Policy, mTLS enforcement, mesh expansion, multi-cluster, and lots lots more.

Get involved!

- 💚 Development is all on GitHub
- 💚 Thriving community in the Slack
- 💚 Formal announcements on the CNCF mailing lists
- 💚 Monthly community calls
- 💚 Formal 3rd-party security audits

Linkerd has a friendly, welcoming community! Join us!

Linkerd is 100% Apache v2 licensed, owned by a neutral foundation (CNCF), and is committed to open governance.



Appendix: History of Linkerd

Two parallel branches of development:

- 🚀 **Linkerd 2.x:** ultralight, zero-config, Kubernetes-first (active)
- 🚀 **Linkerd 1.x:** JVM-based and multi-platform (maintenance)