There's a Bug in My Service Mesh! What Do You Do When the Mesh is At Fault?

LINKERD + PAYBASE®

Ana Calin

Systems Engineer @ Paybase

@AnaMariaCalin

@calinah

Risha Mars

Software Engineer @ Buoyant

@marzipan

@rmars

Mmm...what's cooking?

Mushrooms & trees

Meshes - what is Linkerd?

♀

Mysteries - symptoms of the bug

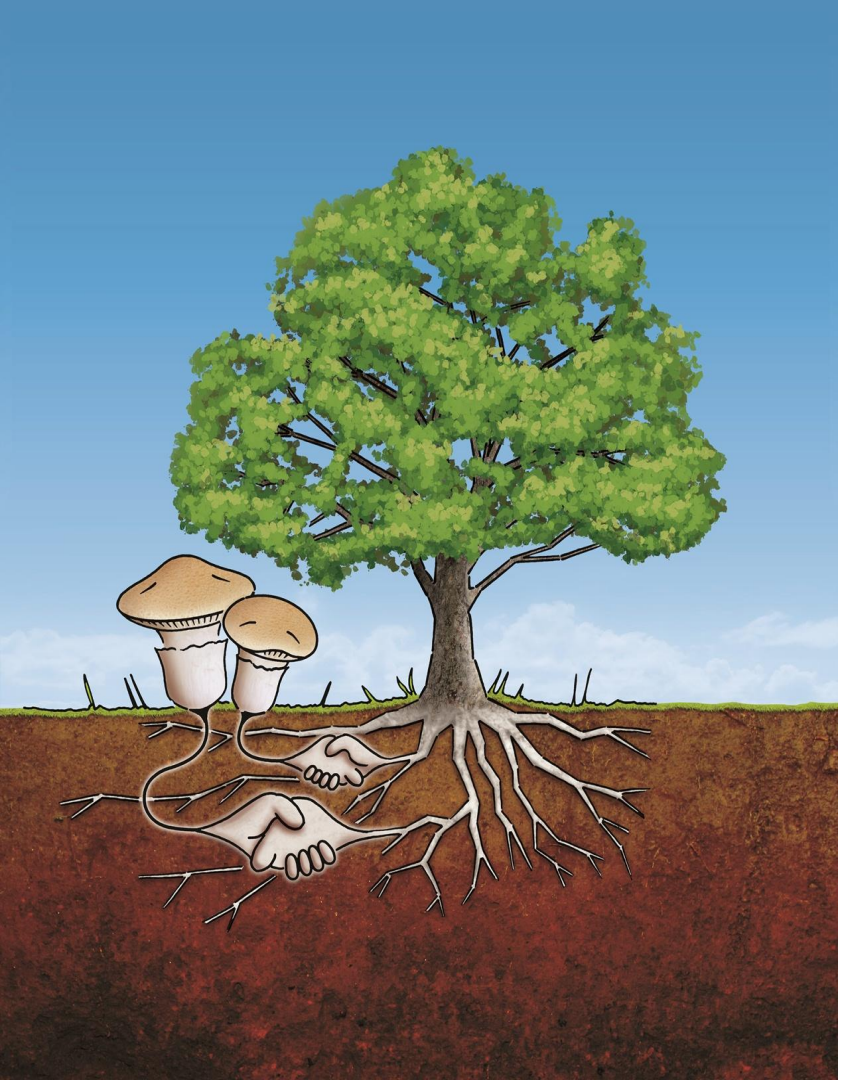
Microscopes - finding the bug

Mastery! - understanding how to find bugs!

Interacting with an OSS project

Users ~~vs~~ Maintainers

and

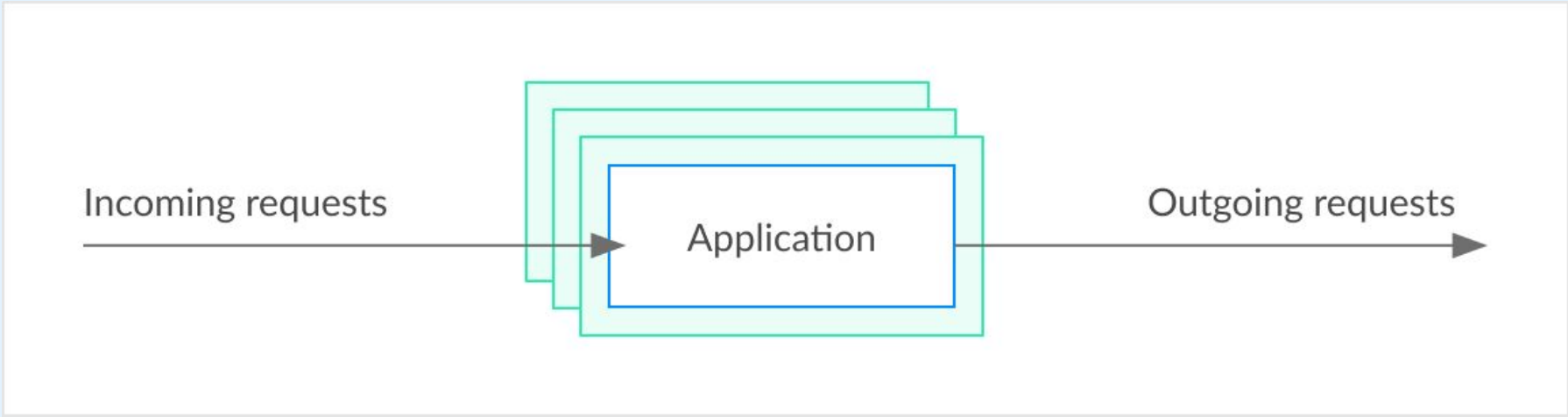


Users & Maintainers of OSS

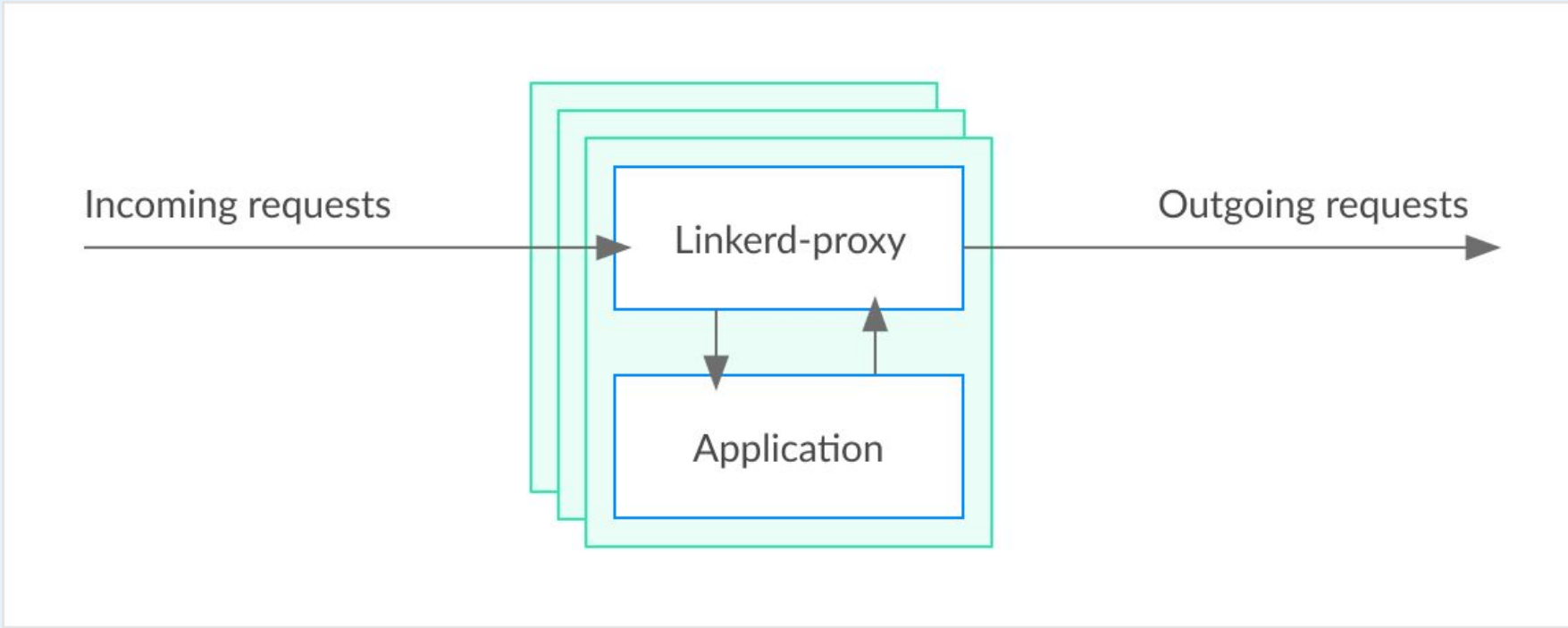
- Users and maintainers should be in a symbiotic relationship
- An OSS project grows based on user feedback, user testing and user contributions
- Some OSS projects are not funded and the work is not paid for so be nice - approach everything in a blame-free way
- Effective communication can open opportunities to learn and fix bugs fast and painlessly
- Being nice is nice

What is Linkerd?

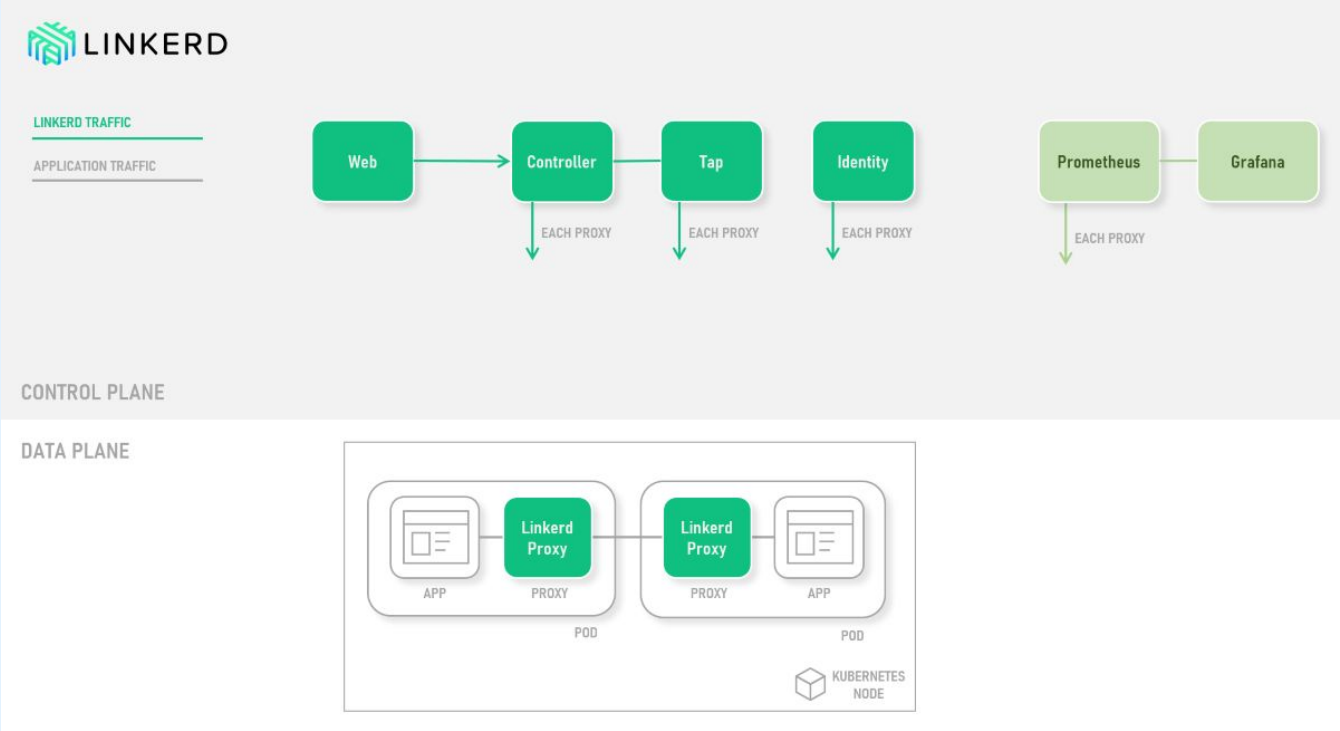
Your application



Your application + data plane



Traffic flow in a meshed application



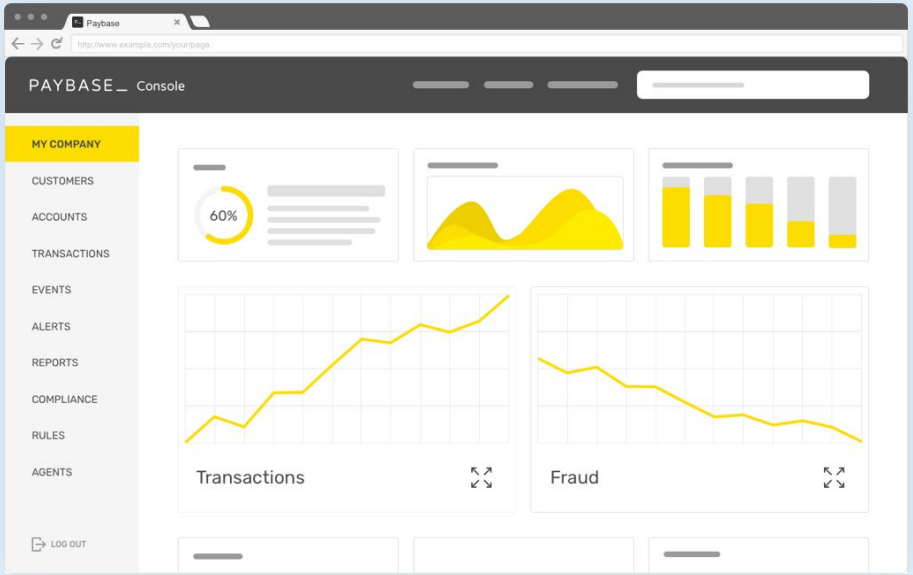
https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-about

Service meshes are awesome

- Automatic mTLS between your meshed services
- Telemetry and Monitoring
- Distributed Tracing
- HTTP, HTTP/2, and gRPC Proxying
- Latency-aware load balancing
- Retries and Timeouts
- TCP Proxying and Protocol Detection

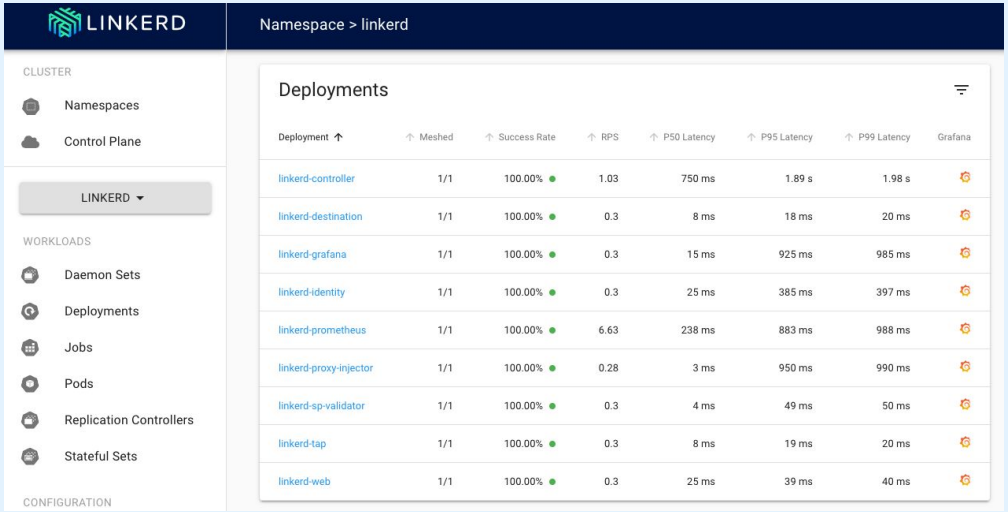
PAYBASE®_

- API driven payments platform
- B2B – marketplace, gig/sharing economies, crypto, sophisticated payment flows
- Highly regulated



Why Linkerd?

- ~80% OSS, 100% running on K8s
- 50+ microservices
- gRPC load balancing for scalability
- Distributed tracing

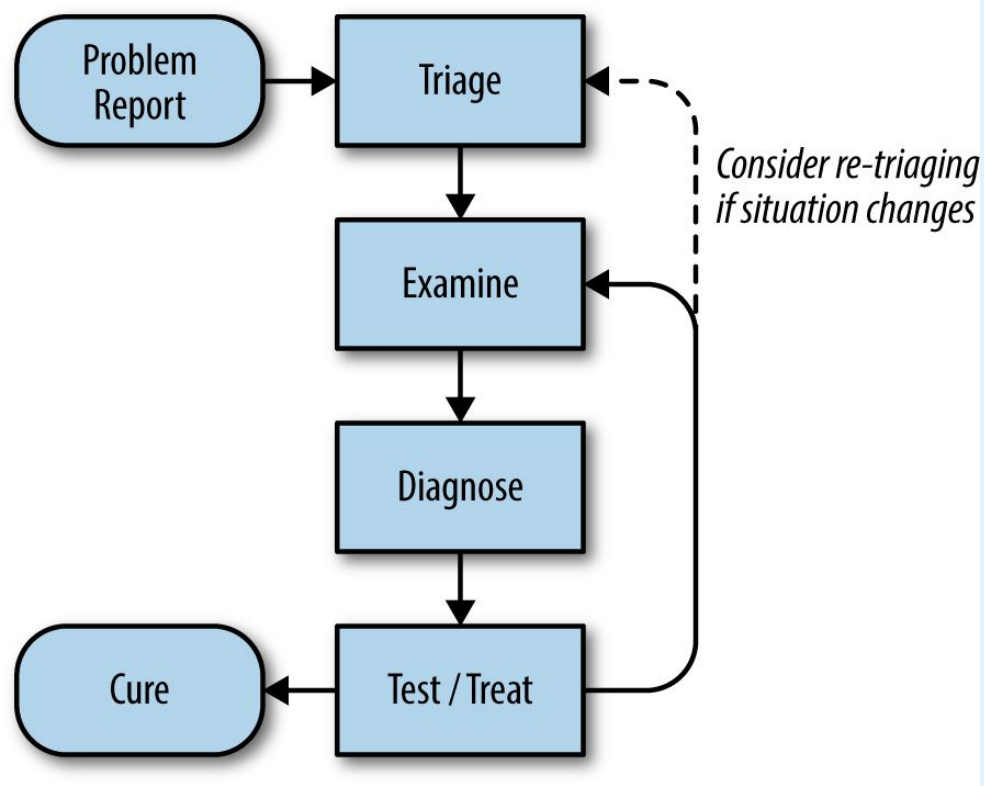


What we expected ***vs*** what we got

A framework for troubleshooting

“Be warned that being an expert is more than understanding how a system is supposed to work. Expertise is gained by investigating why a system doesn't work.”

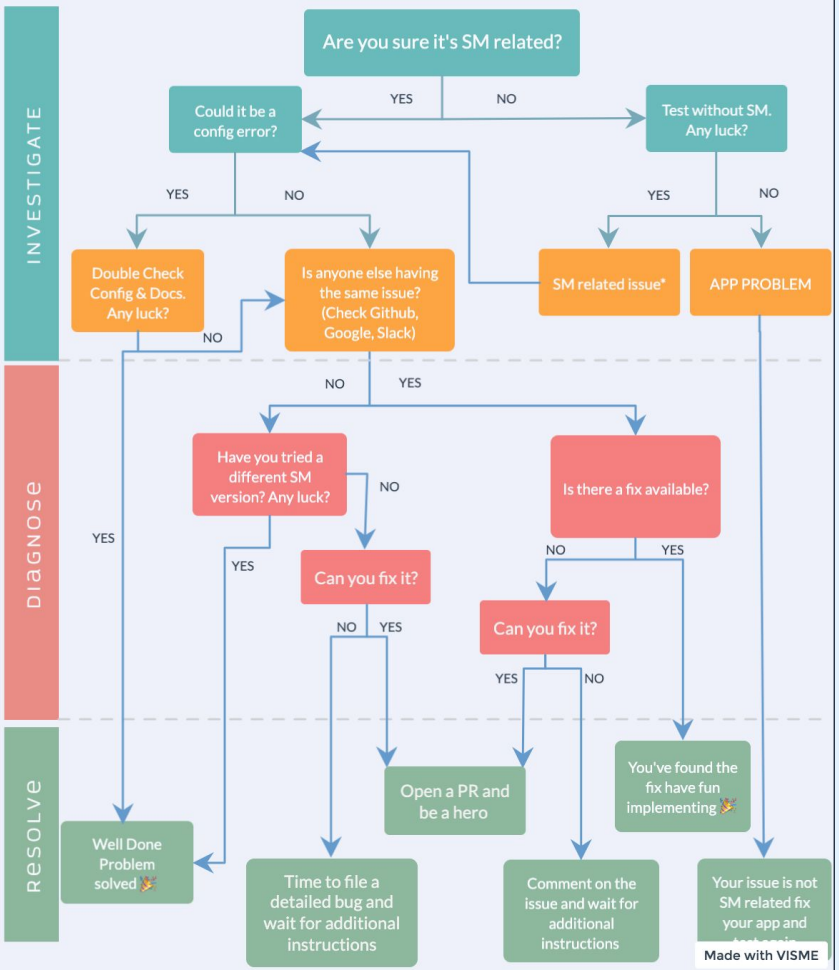
Brian Redman



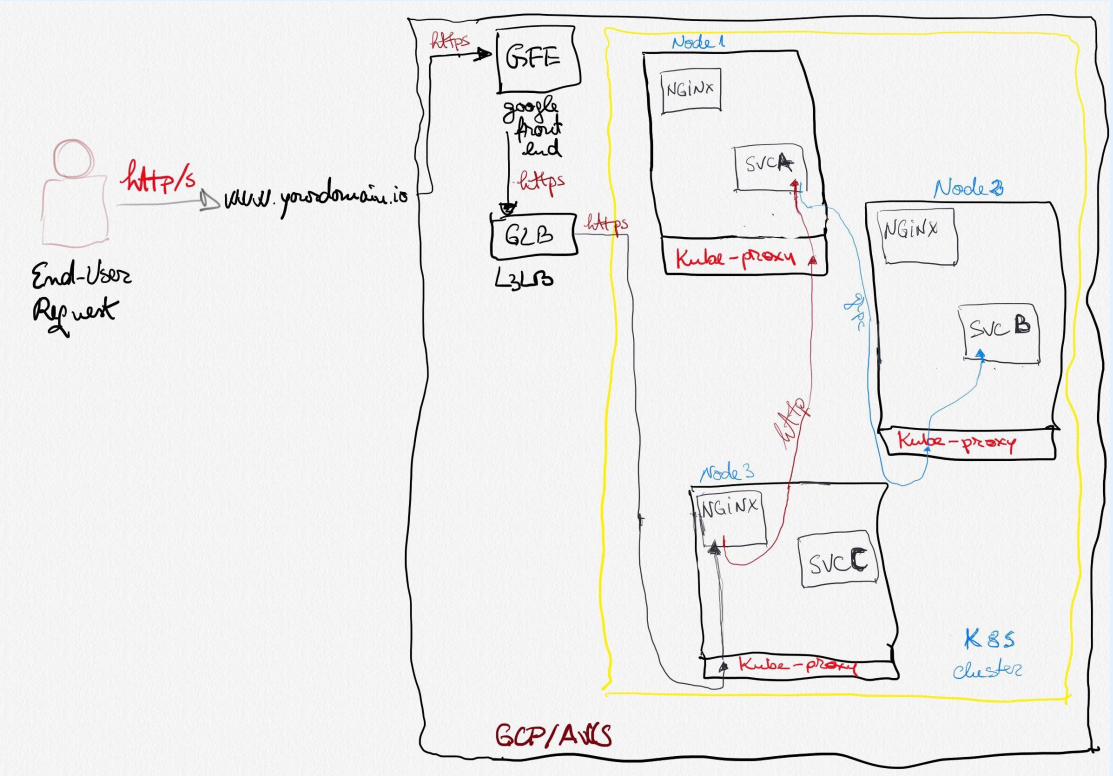
Site Reliability Engineering: How Google Runs Production Systems, Chapter 12

Service Mesh Errors?

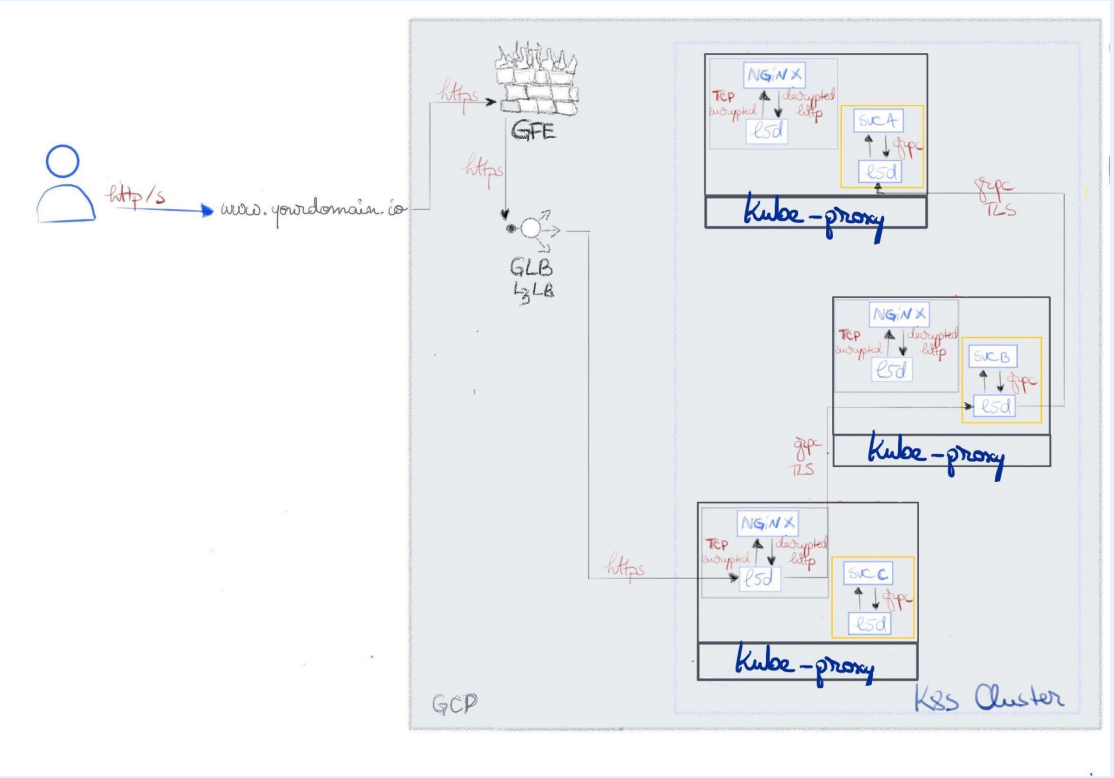
This is the Flow Chart You're Looking For



Diagnose: System without Linkerd

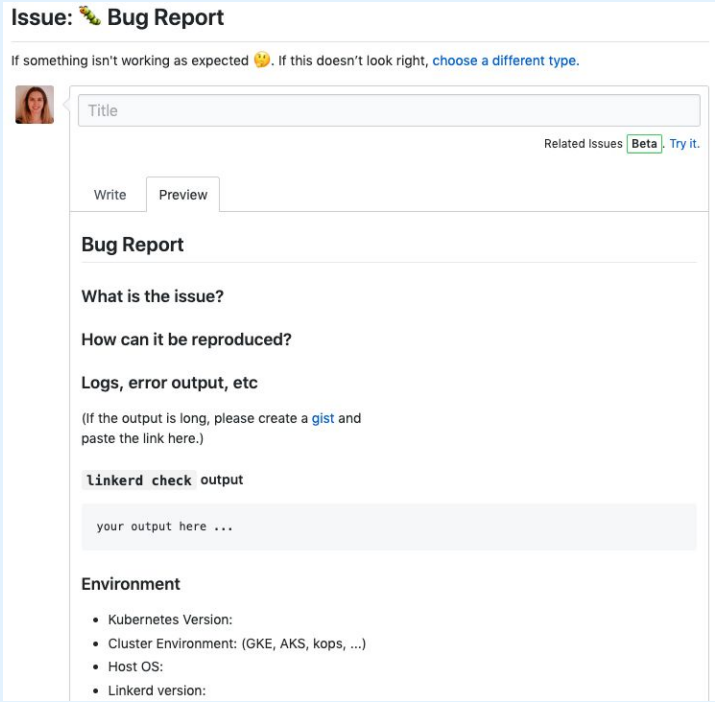


Diagnose: System **with** Linkerd

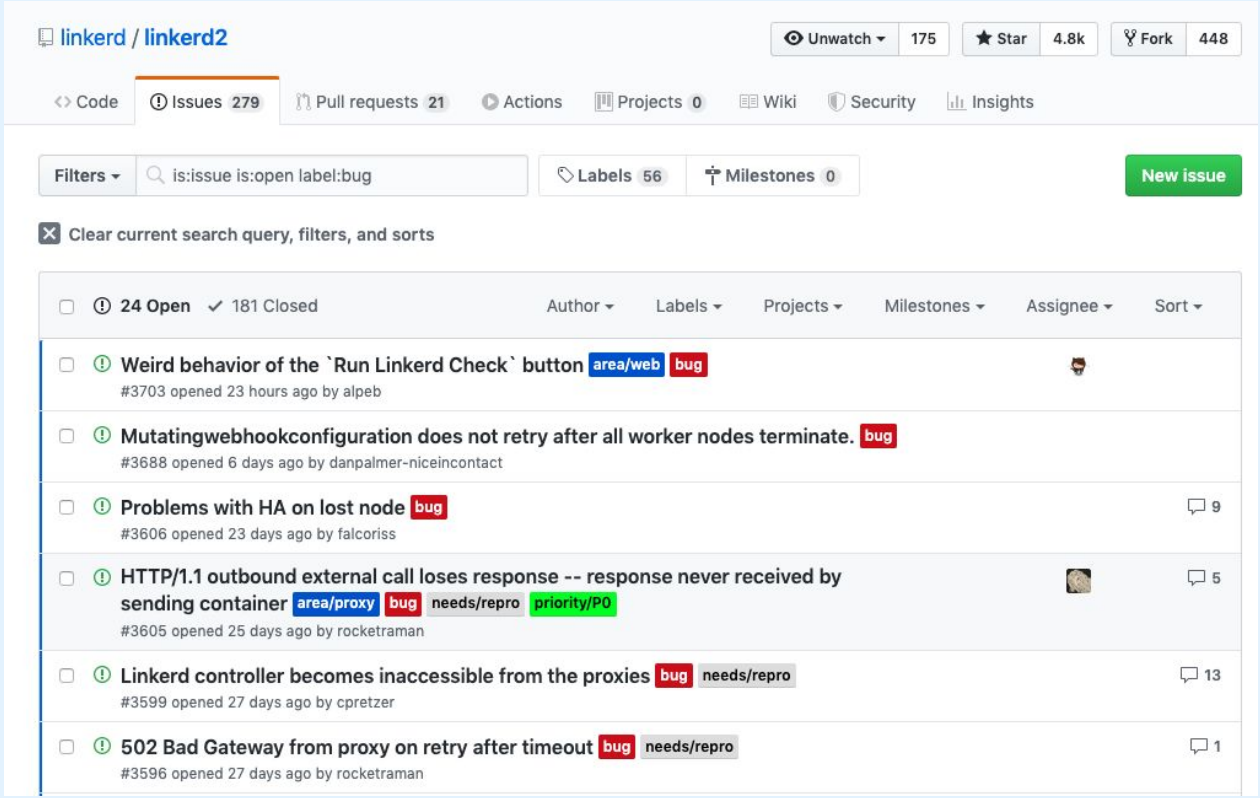


Problem Report: Filing an Issue

- Follow the recommendations
- Attach log outputs
- Don’t assume people know your system – be as detailed as possible
- Include what you’ve already tried
- This is universal advice any time to ask For external help (regardless of medium)



Triage

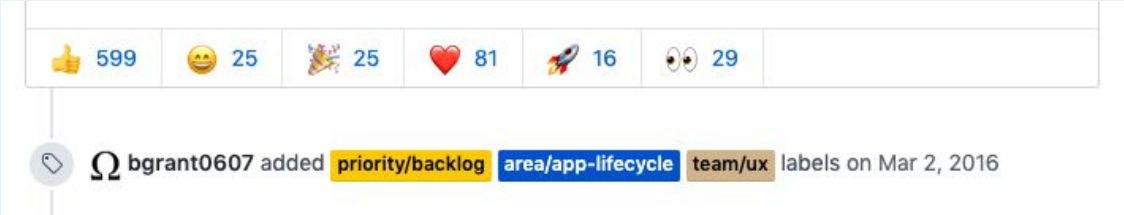

linkerd / linkerd2
Unwatch 175
Star 4.8k
Fork 448
Code
Issues 279
Pull requests 21
Actions
Projects 0
Wiki
Security
Insights
Filters
is:issue is:open label:bug
Labels 56
Milestones 0
New issue
Clear current search query, filters, and sorts
24 Open
181 Closed
Author
Labels
Projects
Milestones
Assignee
Sort
Weird behavior of the `Run Linkerd Check` button area/web bug
#3703 opened 23 hours ago by alpeb
Mutatingwebhookconfiguration does not retry after all worker nodes terminate. bug
#3688 opened 6 days ago by danpalmer-niceincontact
Problems with HA on lost node bug
#3606 opened 23 days ago by falcoriss
9
HTTP/1.1 outbound external call loses response -- response never received by sending container area/proxy bug needs/repro priority/P0
#3605 opened 25 days ago by rocketraman
5
Linkerd controller becomes inaccessible from the proxies bug needs/repro
#3599 opened 27 days ago by cpretzer
13
502 Bad Gateway from proxy on retry after timeout bug needs/repro
#3596 opened 27 days ago by rocketraman
1

Triage: Reproducibility

- Is the problem clearly stated?
- Are there enough details to reproduce the bug?
- What is the smallest reproducible test case?
- Can we reproduce it without having your whole architecture in play?

Triage: Impact

- How bad is the bug?
  - Does it impact security? Usability?
- Are multiple users experiencing this bug?
  - If a user files a good bug report, other users can comment if they are also experiencing it!



Diagnosing the Bug

Where could the bug be?

Is it…

- in the application?
- in the application's dependencies?
- in the Linkerd control plane? (Golang)
- in the Golang dependencies?
- in the Linkerd-proxy? (Rust)
- in the Rust dependencies?
- in Kubernetes?

Diagnosing the Bug

- The initial bug report contained proxy and application logs
  - Application logs (`kubectl logs -f deploy/foo -n bar`)
  - Proxy logs (`linkerd logs`)
  - There were protocol errors on requests that had gRPC metadata in the headers
- We asked for further detail: `linkerd tap`
  - Examined the requests between services in the application
- We dived even deeper: `tcpdump`
  - Looked at the raw TCP packets
  - Saw that headers were being split across two frames
  - This is unusual, as headers typically only take up one frame

Understanding the Bug(s)!

🤔 HTTP/2 in the Linkerd service mesh

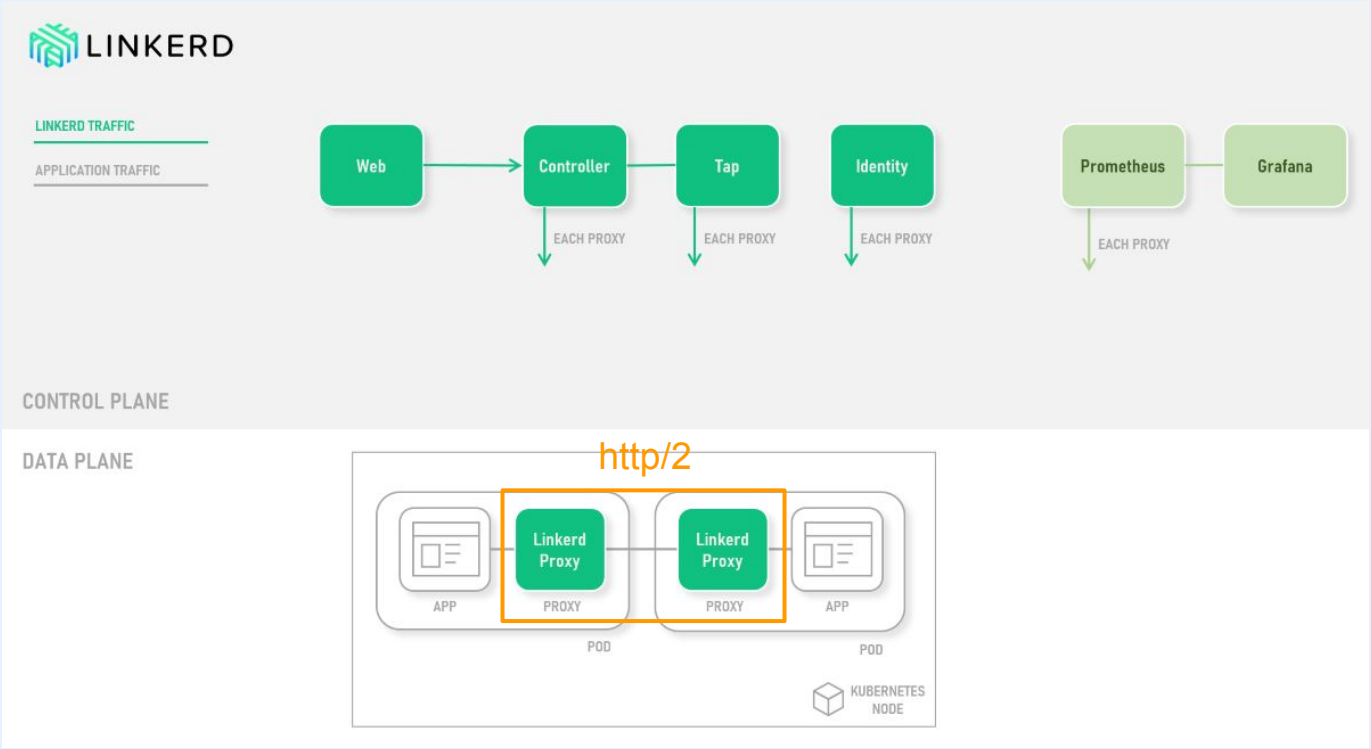
🤔 HTTP/2 frame types and header compression

Understanding the Bug(s)!

🤓 **HTTP/2 in the Linkerd service mesh**

🤔 HTTP/2 frame types and header compression

HTTP/2 in the Linkerd service mesh



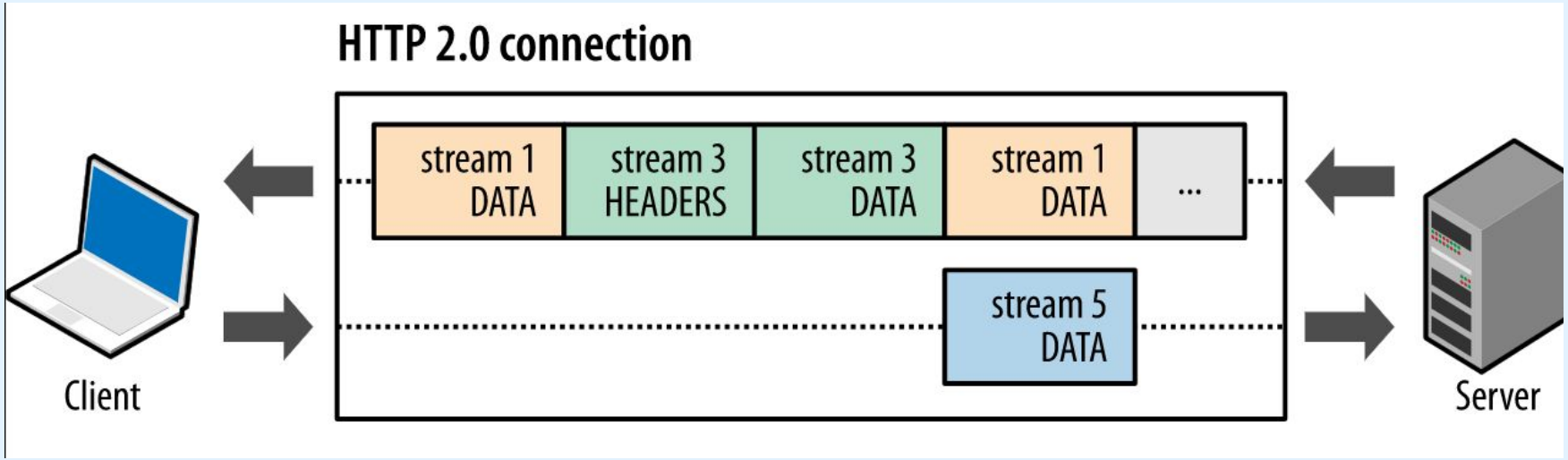
https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-about

Understanding the Bug(s)!

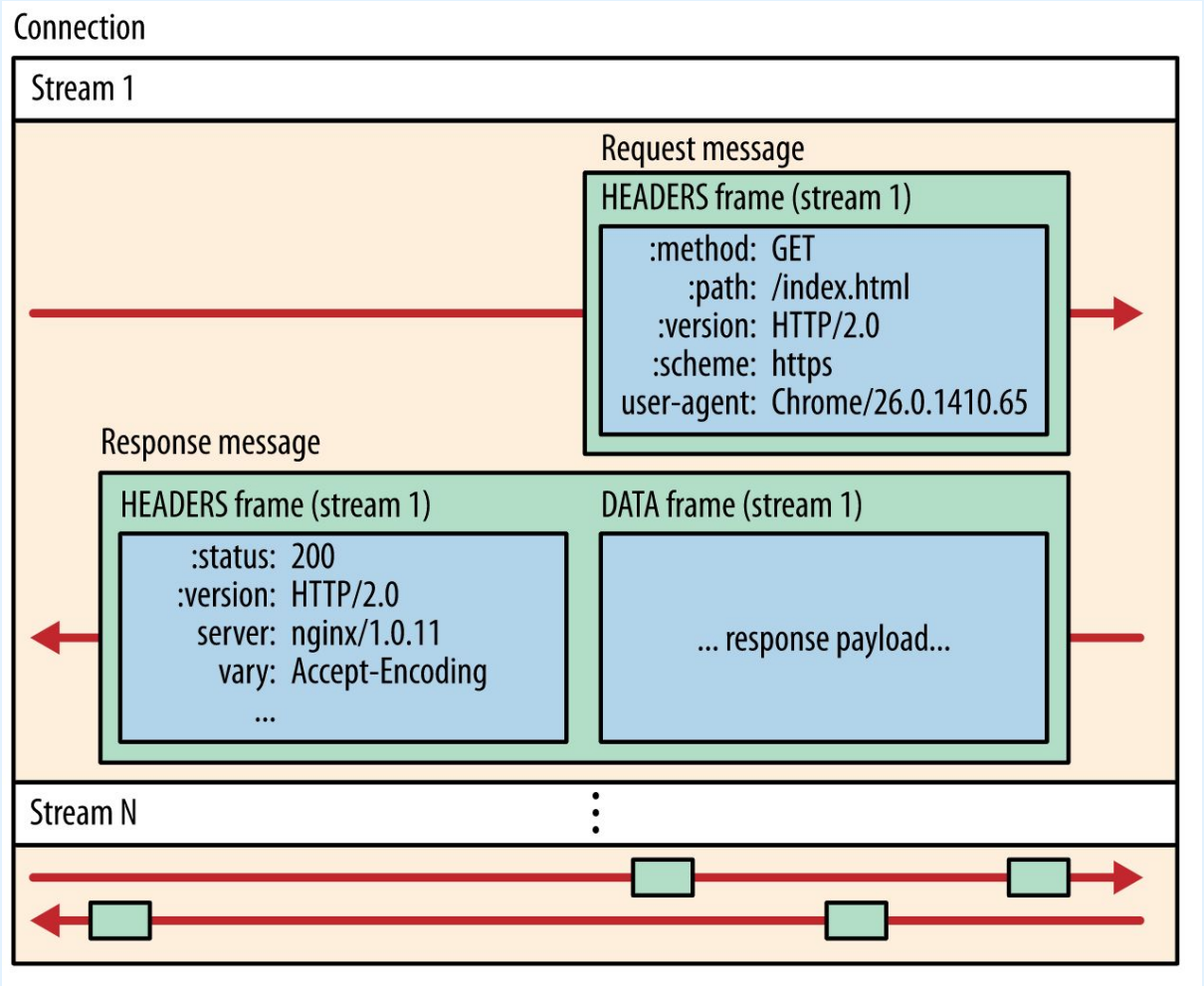
🤔 HTTP/2 in the Linkerd service mesh

🤓 **HTTP/2 frame types and header compression**

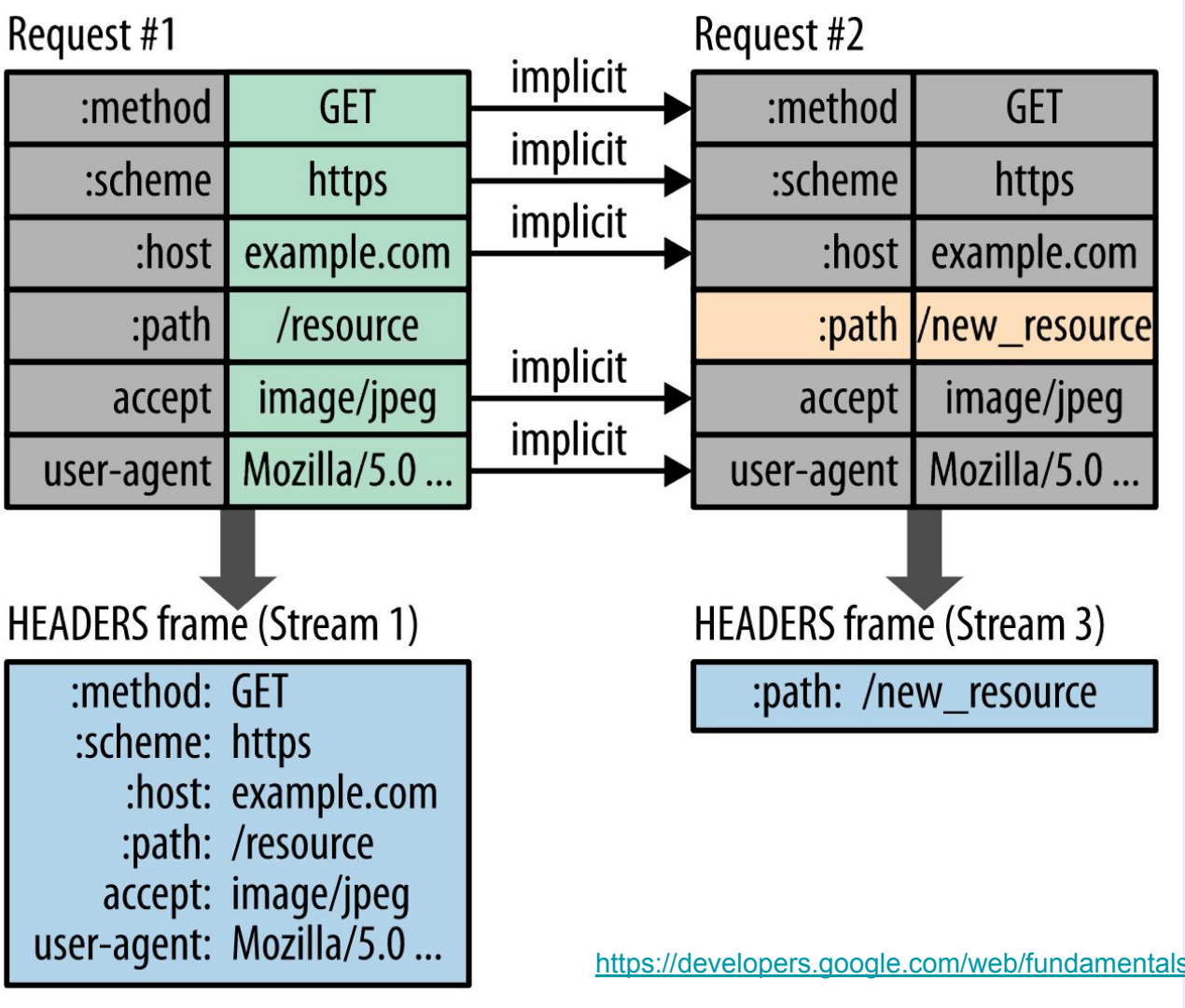
HTTP/2: multiplexing, frame types



https://developers.google.com/web/fundamentals/performance/http2



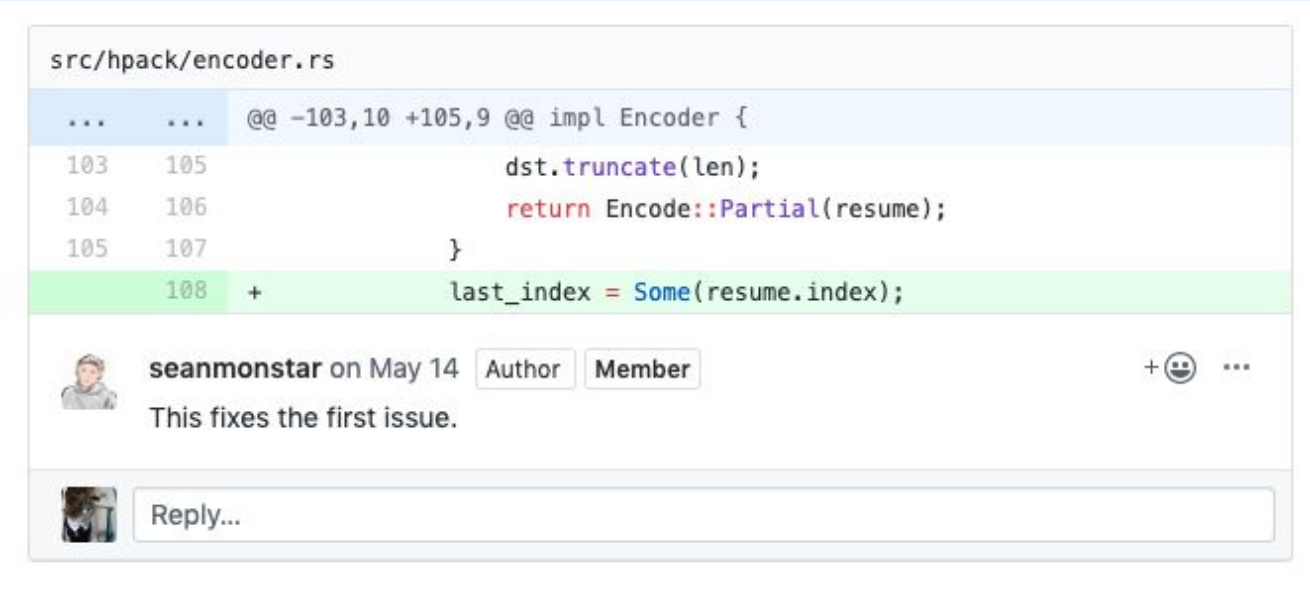
https://developers.google.com/web/fundamentals/performance/http2



https://developers.google.com/web/fundamentals/performance/http2

Bug #1: Continuation Frame Panic

The code panicked when a CONTINUATION frame contained a repeated header



Bug #2: Evicted Table Header Index

We were looking up repeated headers using the wrong index

src/hpack/table.rs

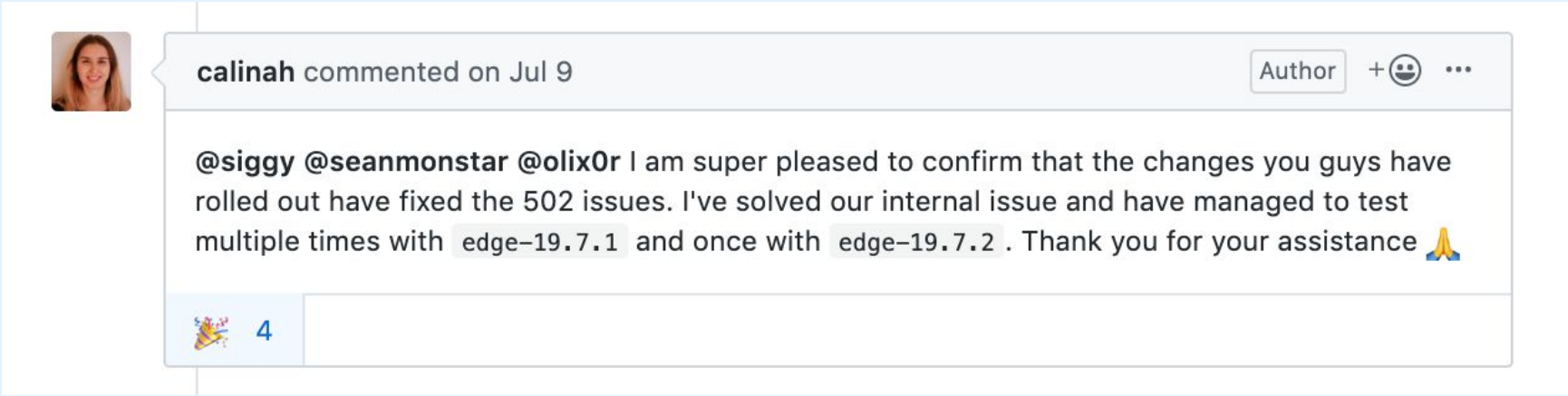
```
...  ...  @@ -125,7 +125,7 @@ impl Table {
125  125          Indexed(idx, ..) => idx,
126  126          Name(idx, ..) => idx,
127  127          Inserted(idx) => idx + DYN_OFFSET,
128       -       InsertedValue(idx, _) => idx,
     128  +       InsertedValue(_name_idx, slot_idx) => slot_idx + DYN_OFFSET,
```

seanmonstar on May 14 Author Member

This makes the next header (that has no name, so using this index) point at the new index, not at the old name index, which may have been evicted.

Reply...

The bugs are fixed!



Aftermath:
Linkerd diagnostic improvements!

- A debug sidecar container
  - Deploy the container into a failing pod to diagnose problems
  - The debug image contains `tshark`, `tcpdump`, `lsof`, and `iproute2`
  - Once installed, it automatically logs all traffic with `tshark`
- More visibility into application traffic with `linkerd tap`
  - Can now also view request bodies
- Tracing in the Rust libraries
  - Increased visibility into the libraries we depend on

Useful commands for troubleshooting

```
# increase log_level &/or skip inbound/outbound ports

$ kubectl get deploy -o yaml | linkerd inject --proxy-log-level=debug,linkerd2_proxy=debug --skip-
outbound-ports 8529,5432,8015 --skip-inbound-ports 8529,5432,8015 - | kubectl apply -f -

# enable linkerd debug container

$ kubectl get deploy $deploy_name -o yaml | linkerd inject --enable-debug-sidecar --manual - | kubectl
apply -f -

# run tcpdump in debug container

$ kubectl exec -it pod_name linkerd-debug -- tcpdump -i any -s 65535 -w out.pcap

# listen to traffic stream for a resource

$ linkerd tap deploy/web

# display pods (CPU/Memory/Storage) usage

$ kubectl top pods
```

Summary

There was more than one bug! The bugs were deep in the stack!

All got fixed fairly quick due to:

- Detailed bug reports
- Space to test with/without linkerd , different versions
- Used those log suggestions
- Looked through code / other bugs

\+

PAYBASE®

Linkerd has a **friendly, welcoming** community! Join us!


github.com/linkerd


slack.linkerd.io


@linkerd

FROM YOUR FRIENDS AT


BUOYANT

Slides at **bit.ly/bug-in-my-mesh**

bit.ly/bug-in-my-mesh

References

https://developers.google.com/web/fundamentals/performance/http2

https://http2.github.io/http2-spec/#FrameHeader

https://tools.ietf.org/html/rfc7541

https://linkerd.io/2/tasks/using-the-debug-container

https://buoyant.io/2017/04/25/whats-a-service-mesh-and-why-do-i-need-one

https://linkerd.io/2/reference/architecture

https://docs.microsoft.com/en-us/azure/aks/servicemesh-linkerd-about