



gRPC Forensics: Investigating Problems in Production

Thomas Rampelberg



Hello gRPC Conf!

Thomas Rampelberg

Software Engineer @ Buoyant

 @grampelberg

 grampelberg





Honest Status Page

@honest_update

Follow

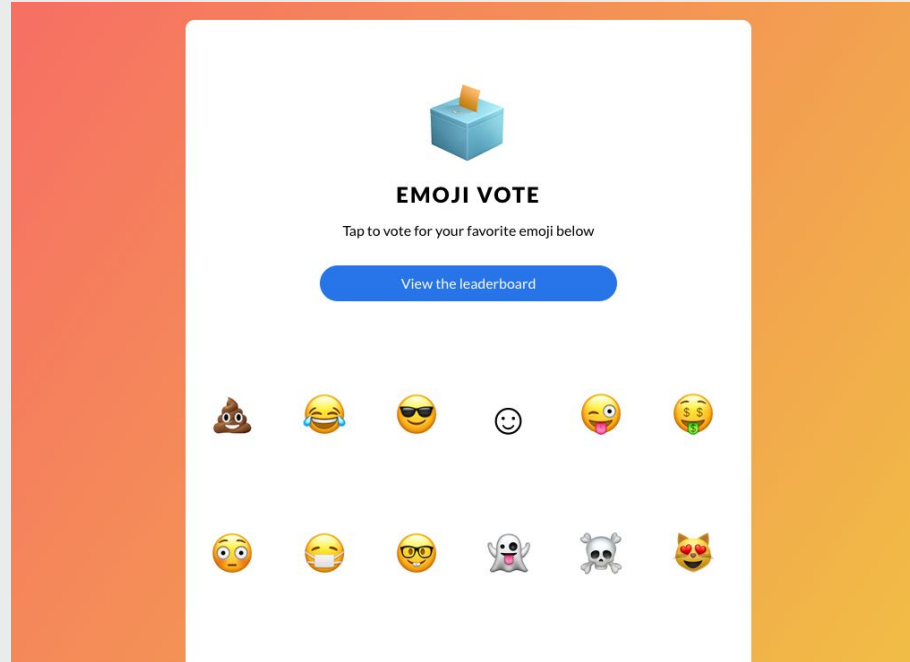


We replaced our monolith with micro services so that every outage could be more like a murder mystery.

4:10 PM - 7 Oct 2015



Get your votes in



<https://emoji.15d.io>

What problem needs to be solved?





Alerting

- Downtime
- Exceptions
- Service Level Agreements





Forensics

- Memory leaks
- Performance issues
- Environment specific failures



White Box Forensics





Exceptions

```
func readIntoBytes(filename string) ([]byte, error) {  
    file, err := static.Templates.Open(filename)  
    if err != nil {  
        return nil, err  
    }  
    defer file.Close()  
  
    buf := new(bytes.Buffer)  
    buf.ReadFrom(file)  
  
    return buf.Bytes(), nil  
}
```



Application Instrumentation

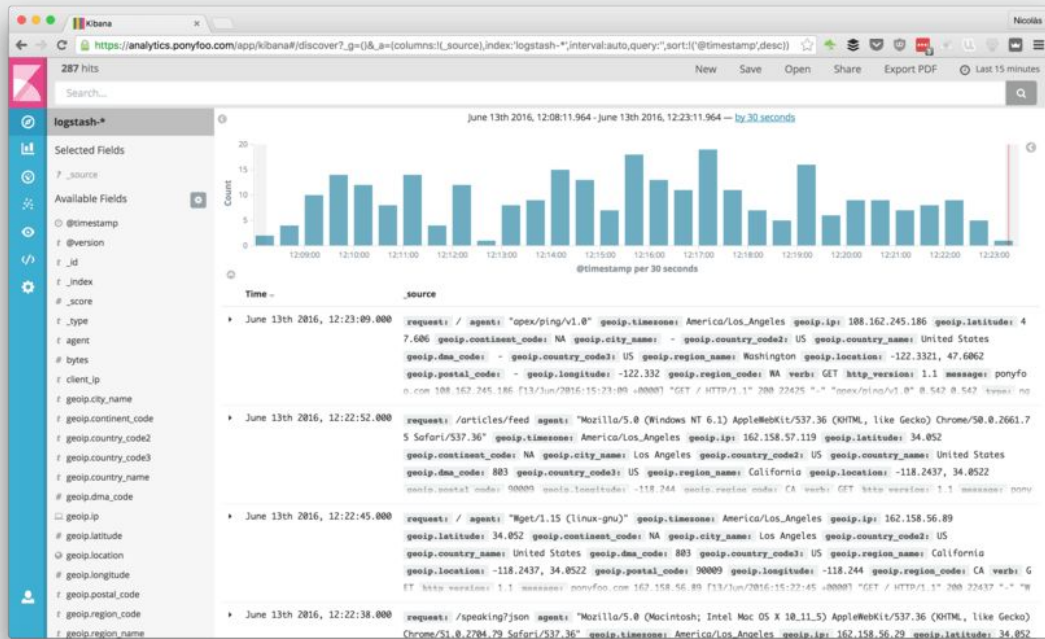
```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.9864e-05
go_gc_duration_seconds{quantile="0.25"} 5.2299e-05
go_gc_duration_seconds{quantile="0.5"} 7.1593e-05
go_gc_duration_seconds{quantile="0.75"} 0.000151335
go_gc_duration_seconds{quantile="1"} 0.003560545
go_gc_duration_seconds_sum 0.058749902
go_gc_duration_seconds_count 254
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 59
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.10.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 6.619704e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.465679856e+09
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.50257e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 6.977458e+06
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time that the garbage collector has been using since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 3.924477433282826e-05
```

```
func main() {
    recordMetrics()

    http.Handle("/metrics", promhttp.Handler())
    http.ListenAndServe(":2112", nil)
}
```



Log Aggregation



Howto: <https://kubernetes.io/docs/tasks/debug-application-cluster/logging-elasticsearch-kibana/>

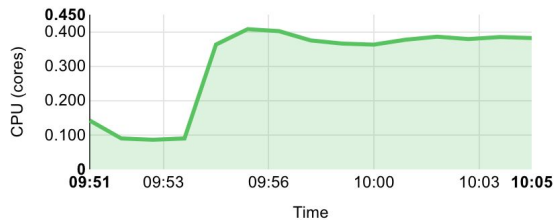
Black Box Forensics



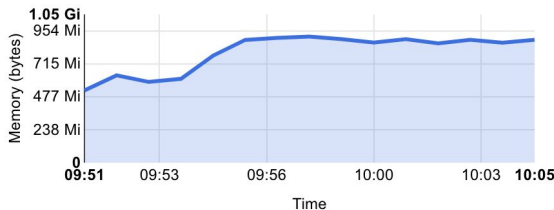
System/Process Metrics

- CPU
- Memory
- Networking
- Disk

CPU usage



Memory usage ⓘ



☀ Prometheus

- Kubernetes Standard
- Time Series
- Alerts
- Extra Metrics



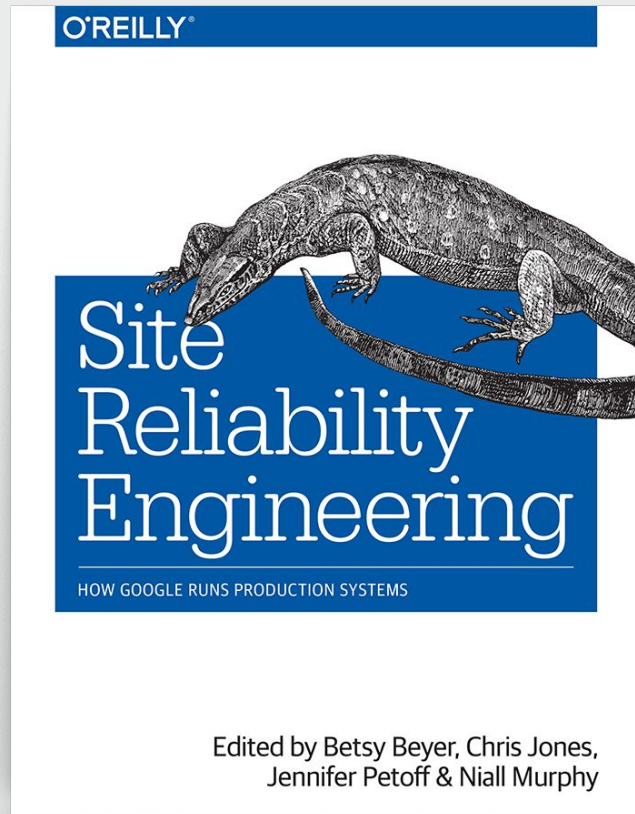
Install: <https://github.com/helm/charts/tree/master/stable/prometheus>
Tracked metrics: https://github.com/prometheus/node_exporter#enabled-by-default



Golden Signals

- Latency
- Traffic
- Errors
- Saturation

<http://bit.ly/golden-signals>





Every request matters

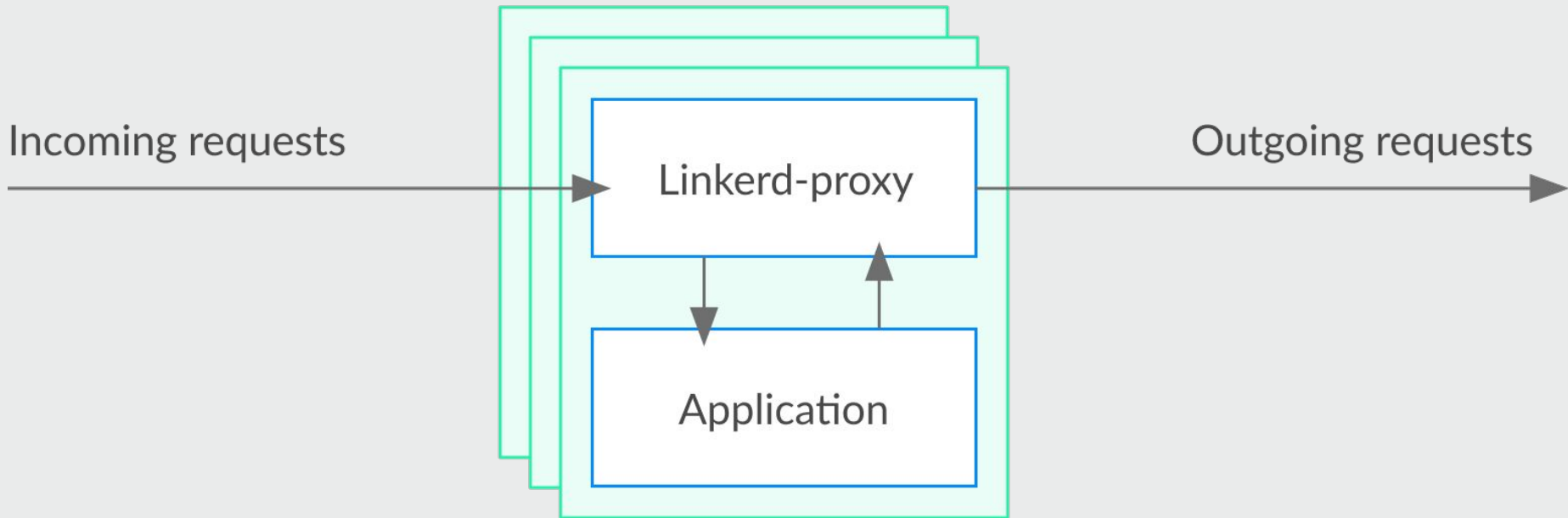
- Tail latency is important
- Users see responses
- Latency is not normally distributed

Site	# of requests	page loads that would experience the 99%'lie [(1 - (.99 ^ N)) * 100%]
amazon.com	190	85.2%
kohls.com	204	87.1%
jcrew.com	112	67.6%
saksfifthavenue.com	109	66.5%
--	--	--
nytimes.com	173	82.4%
cnn.com	279	93.9%
--	--	--
twitter.com	87	58.3%
pinterest.com	84	57.0%
facebook.com	178	83.3%
--	--	--
google.com (yes, that simple noise-free page)	31	26.7%
google.com search for "http requests per page"	76	53.4%

Linkerd: Microservice Forensics

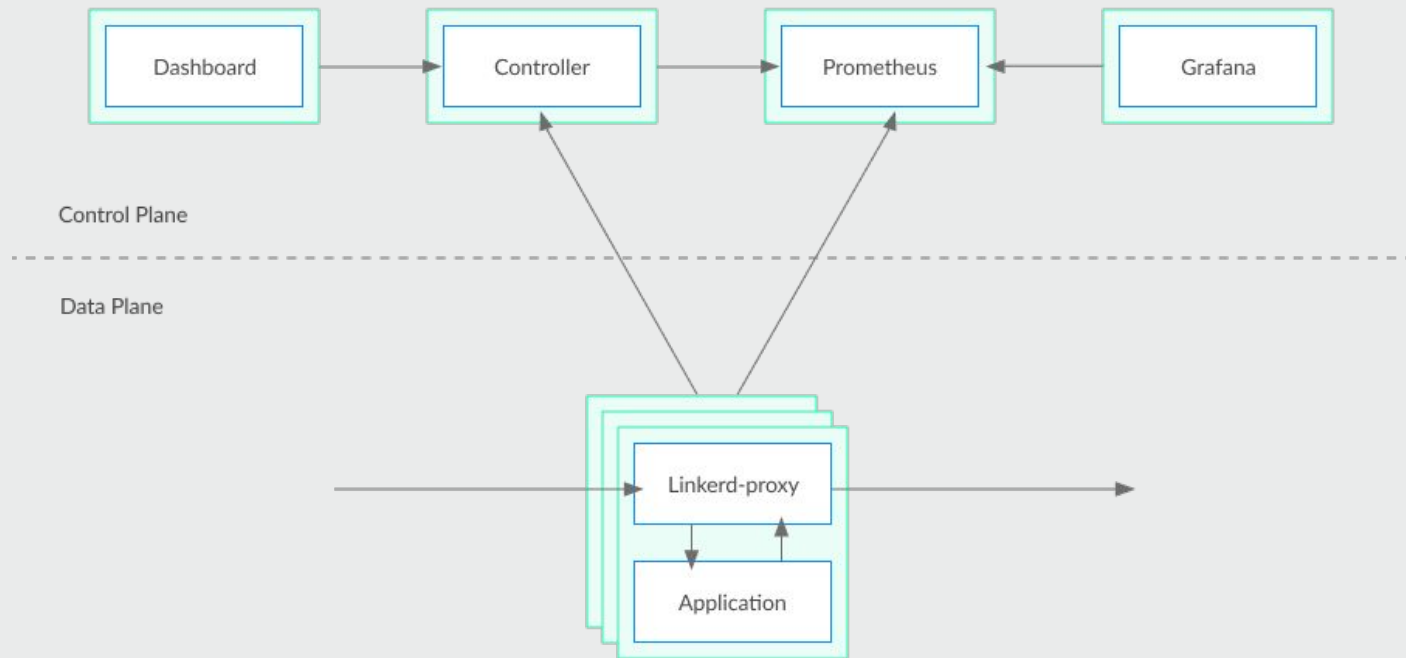


What is Linkerd?



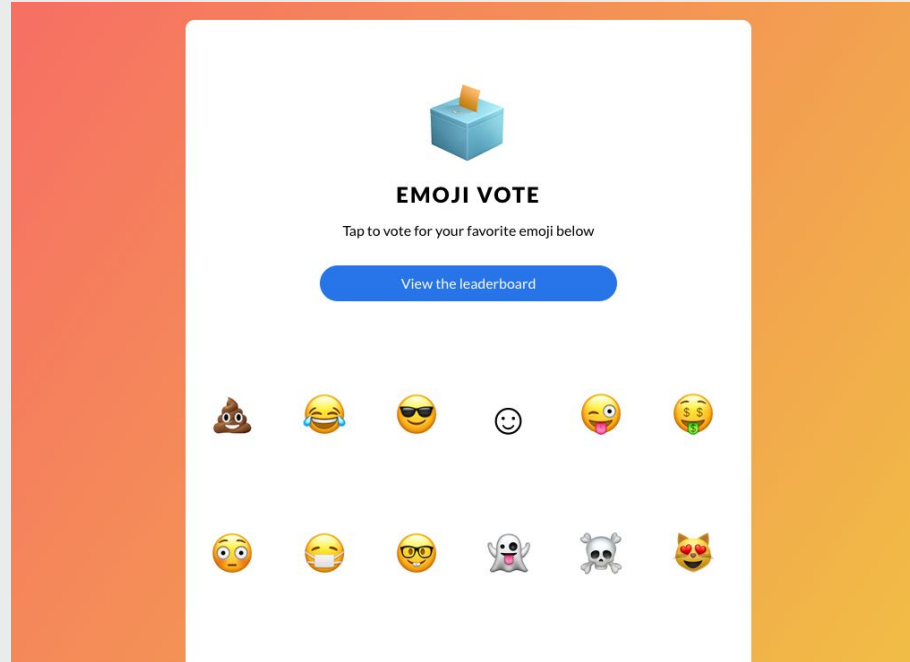


Architecture





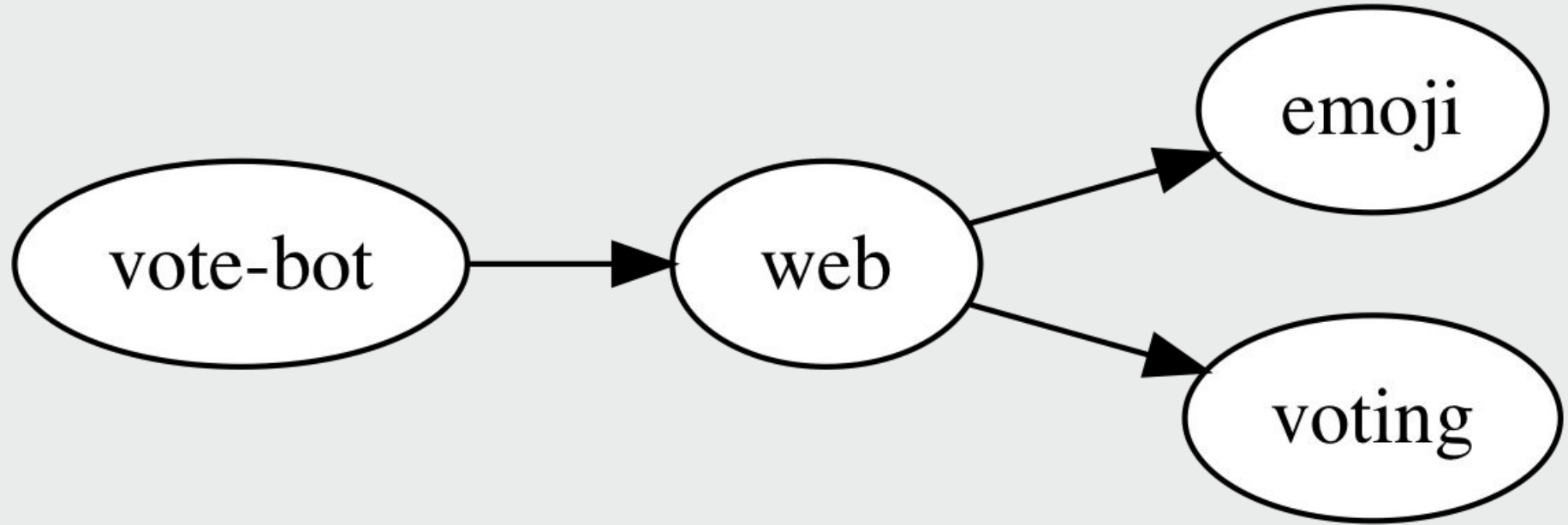
Get your votes in



<https://emoji.15d.io>



Architecture



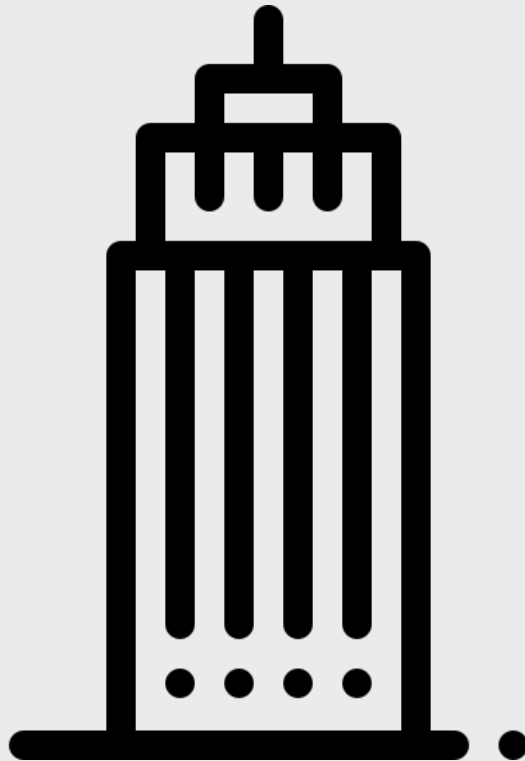
Demo!





Demo

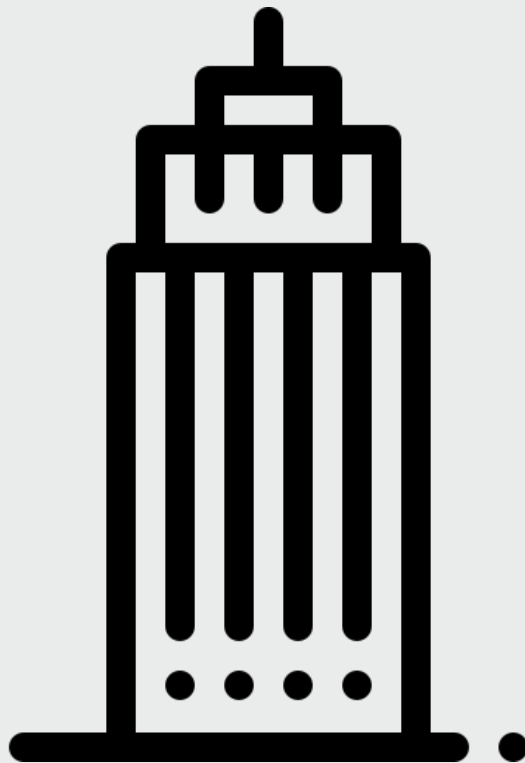
- ☐ Install Linkerd control plane
- ☐ Add emoji voto to data plane
- ☐ Instrument method calls





Demo

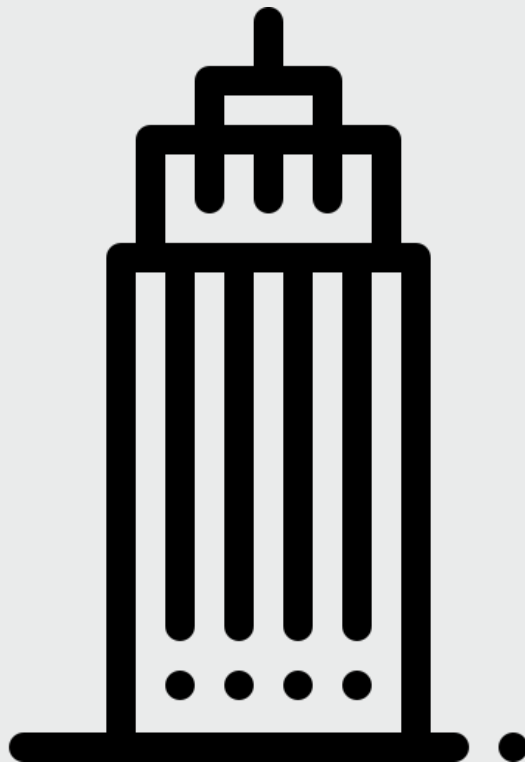
- ☒ Install Linkerd control plane
- ☐ Add emojiivoto to data plane
- ☐ Instrument method calls





Demo

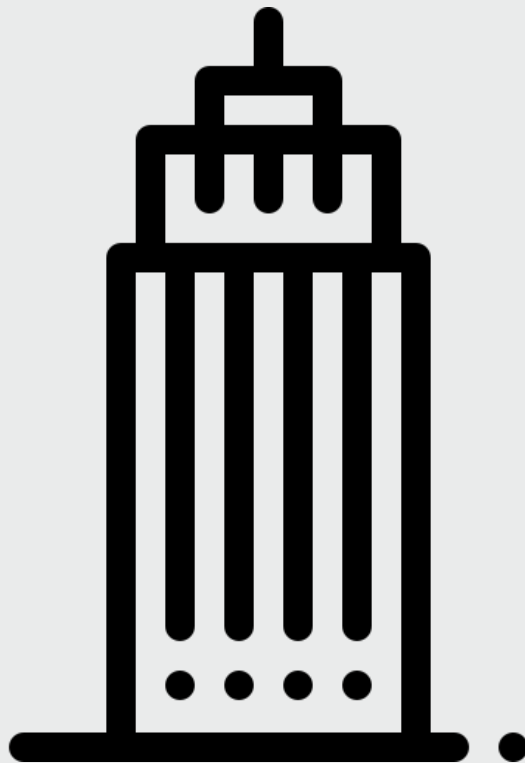
- ☒ Install Linkerd control plane
- ☒ Add emoji voto to data plane
- ☐ Instrument method calls





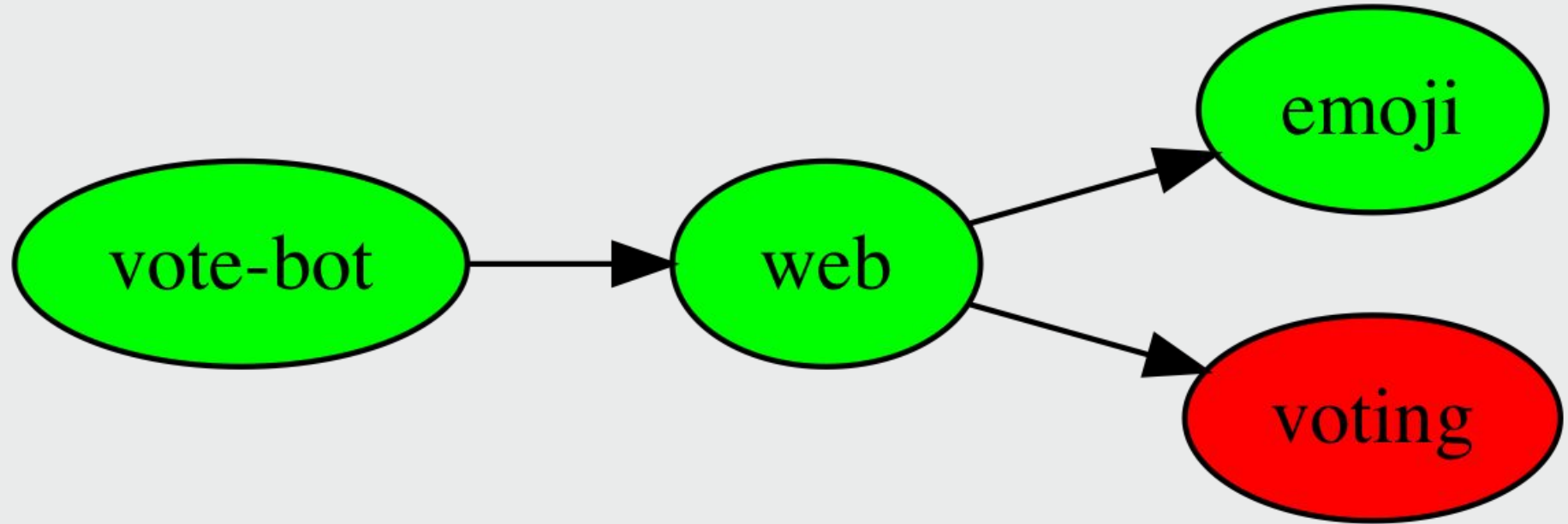
Demo

- ✓ Install Linkerd control plane
- ✓ Add emoji voto to data plane
- ✓ Instrument method calls





The Culprit



Slides

<http://bit.ly/grpcconf2019>

Tutorial

<http://bit.ly/linkerd-tutorial>

Get Started!

<https://bit.ly/linkerd-get-started>
