BUOYANT

Creators of LINKERD

What's Coming in Linkerd 2.13

Flynn, Technical Evangelist for Linkerd
William Morgan, CEO of Buoyant





BUOYANT

Creators of LINKERD



flynn@buoyant.io
@flynn on slack.linkerd.io



william@buoyant.io
@wm on Twitter



What's on the agenda?

- ➔ Today is a little heavier on slides than normal for SMA, and a little lighter on workshop (there's a lot to cover!)
- ➔ Overview of upcoming Linkerd 2.13 features
- ➔ **Workshop: dynamic request routing!**

What do you need for the workshop?



- A Kubernetes cluster. I'll be using `k3d`, but the type of cluster doesn't much matter.
- To follow along, you'll also need `kubectl`, `bat`, and the `linkerd CLI`.
- Check out the workshop source at

  https://github.com/BuoyantIO/service-mesh-academy/tree/main/sneak-peek-2-13

BUOYANT
Creators of LINKERD


KubeCon
CloudNativeCon
Europe 2023
LINKERDDAY
EUROPE

Our first-ever Linkerd Day!

The only KubeCon co-lo that is 100% end user talks!

- Zero-Cost Linkerd at Adidas: Impossible Is Nothing - **Adidas**
- Cost-Efficient Multi-Cluster Monitoring - **Finleap / BWI**
- Securing 1/3 of Norway's Annual State Budget **- NAV**
- From Zero to Prod in 2 Months with a Team of One - **Pento / StickerMule**
- The Negatives to a Per-Host Service Mesh, the Sidecar Model Being a More Ideal Solution for Providing a More Robust Security Boundary - **Raft**
- How a Global Logistics Company Migrated Clouds with Zero Downtime - **large logistics company**
- Increasing Platform Reliability and Performance in One Week with a Service Mesh - **Salt Security**
- Lack of Features Is a Feature for This German Automaker - **LeanNet**

BUOYANT
Creators of LINKERD

2.13 Overview



Coming in Linkerd 2.13...

- Dynamic request routing
- Circuit breaking
- FIPS support (commercial)
- Free health and vulnerability monitoring via Buoyant Cloud
- Azure marketplace offering

Release date: Soon! End of March / first week of April

DYNAMIC REQUEST ROUTING

"I want full control over how requests are routed in my Kubernetes cluster."

Dynamic request routing

In 2.12 and earlier...

- TrafficSplits (via SMI extension) are the primary way of dynamically controlling routing
- These allow coarse-grained routing behavior, based on the service name and percentage of traffic to split

Examples:

- Route 1% of traffic to `Foo` to `Foo-new`, and 99% to `Foo-orig` (progressive delivery)
- Route 100% of traffic to `Foo` to `Foo-west` (multi-cluster / failover)

BUOYANT
Creators of LINKERD

Dynamic request routing

- In 2.13, route traffic based on (almost) any attribute of the request:
  - Headers!
  - Verbs!
  - (But not body)
- Configured with the Gateway API, not with SMI components
  - The same HTTPRoute object introduced in Linkerd 2.12 for fine-grained authz policy
- Examples:
  - Sticky sessions
  - Per-user canaries
  - Sharding per region

BUOYANT
Creators of LINKERD

Dynamic request routing: Canary routing

```yaml
apiVersion: policy.linkerd.io/v1beta2
kind: HTTPRoute
metadata:
 name: color-canary
 namespace: faces
spec:
 parentRefs:
   - name: color
     kind: Service
     group: core
     port: 80                          # Match port numbers with what's in the Service resource
 rules:
 - backendRefs:
   - name: color
     port: 80
     weight: 50                        # Adjust the weights to control balancing
 - backendRefs:
   - name: color2
     port: 80
     weight: 50
```

BUOYANT
Creators of LINKERD

Dynamic request routing: A/B Routing

```yaml
apiVersion: policy.linkerd.io/v1beta2
kind: HTTPRoute
metadata:
 name: smiley-a-b
 namespace: faces
spec:
 parentRefs:
   - name: smiley
     kind: Service
     group: core
     port: 80
 rules:
 - matches:
   - headers:
     - name: "x-faces-user"          # X-Faces-User: testuser gets the B side of the test
       value: "testuser"
   backendRefs:
     - name: smiley2
       port: 80
 - backendRefs:
   - name: smiley
     port: 80
```

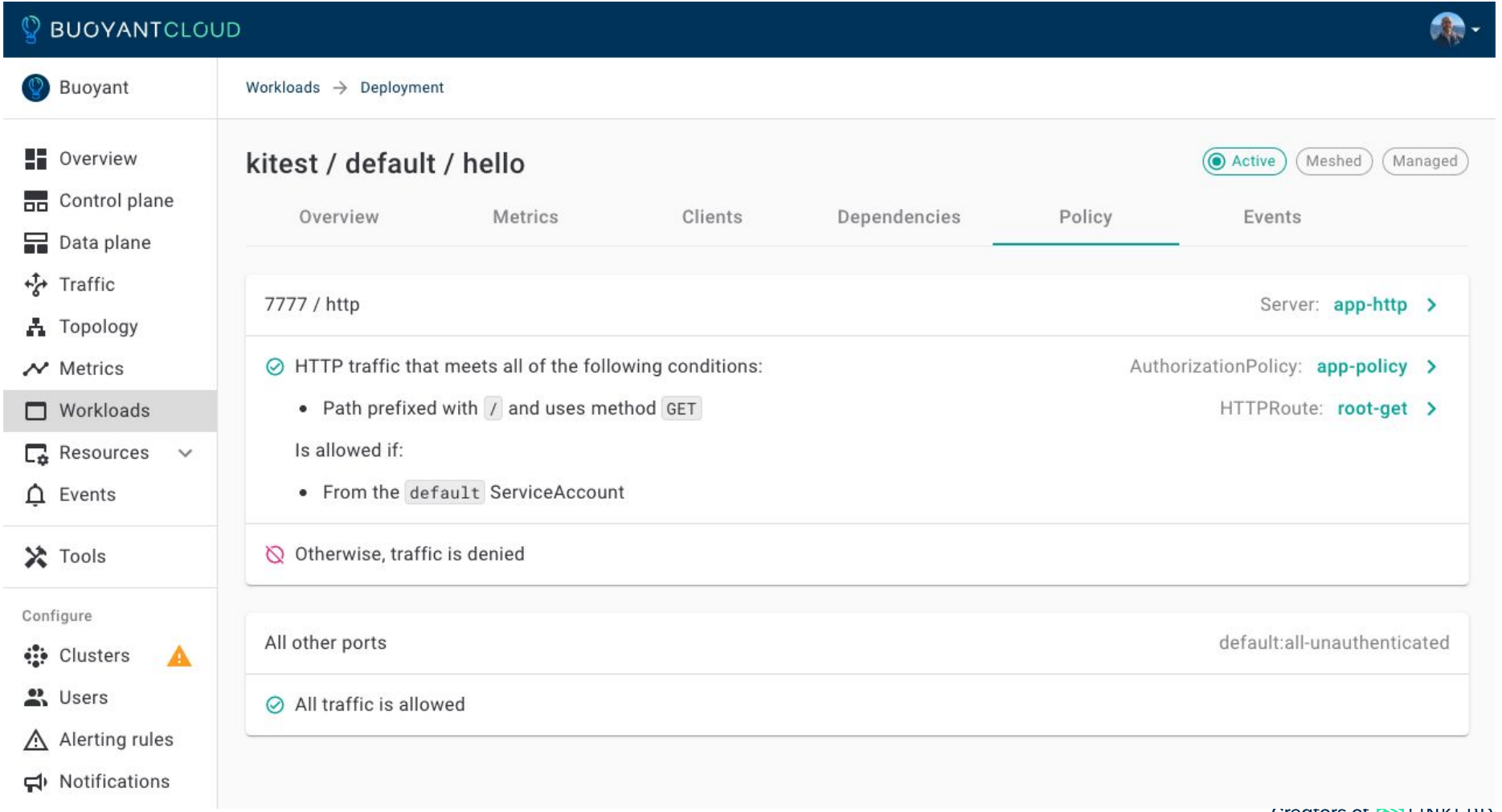
BUOYANT
Creators of LINKERD

A note on Gateway API

- Originally designed for ingress, but a great fit for service meshes too!
  - Lots of ongoing work in the GAMMA initiative to adapt it to meshes
- As of 2.12, Gateway API is the core configuration mechanism in Linkerd for talking about "classes of HTTP traffic" (including gRPC)
  - E.g. auth policy, dynamic request routing
- It's powerful, flexible, and... already lives on your cluster!
  - (Tiny caveat: we are copying the resource to the `policy.linkerd.io` APIGroup, for $reasons, until later releases)
- We expect this to deprecate SMI in the long run.
- We expect this to deprecate ServicePolicy in the long run.

Dynamic Request Routing Gotchas

- Only one big one: ServiceProfile and HTTPRoute **don't compose**
  - If you have a ServiceProfile that defines routes, it will always take precedence over the HTTPRoute
  - **This will be the case for the foreseeable future**
- Over time, HTTPRoute will be able to take over the things that ServiceProfile currently does
  - In 2.13, HTTPRoute is still limited: this is still early days for the Gateway API in Linkerd!


BUOYANTCLOUD
Buoyant
Workloads → Deployment
Overview
Control plane
Data plane
Traffic
Topology
Metrics
Workloads
Resources
Events
Tools
Configure
Clusters
Users
Alerting rules
Notifications
kitest / default / hello
Active
Meshed
Managed
Overview
Metrics
Clients
Dependencies
Policy
Events
7777 / http
Server: app-http
HTTP traffic that meets all of the following conditions:
AuthorizationPolicy: app-policy
Path prefixed with / and uses method GET
HTTPRoute: root-get
Is allowed if:
From the default ServiceAccount
Otherwise, traffic is denied
All other ports
default:all-unauthenticated
All traffic is allowed

CIRCUIT BREAKING

"Stop sending compounding traffic to a dying service."

Circuit breaking

- Detects when a workload endpoint is failing, and stops delivering traffic to it.
  - E.g. too many consecutive failures
  - E.g. Success rate too low
- Turn traffic back *on* when things recover.
- Configured with annotations... for now
  - More options in the future
  - Trying to make this as simple as possible!

Circuit breaking (possible) examples:

- Break the circuit after four consecutive request failures:

  `config.alpha.linkerd.io/circuit-breaker-consecutive-failures: 4`

- Break the circuit after the success rate drops below 80%:

  `config.alpha.linkerd.io/circuit-breaker-success-rate: 80`

- Try reenabling traffic after 30 seconds:

  `config.alpha.linkerd.io/circuit-breaker-backoff-ttl: 30s`

- **None of these are yet set in stone**: this is the current direction, but things may look a little different in stable-2.13.0.

BUOYANT
Creators of LINKERD

Health monitoring, vulnerability alerting, and upgrade assistance with Buoyant Cloud

"I want to be a professional about my service mesh deployment."

Making Linkerd healthy, secure, and up-to-date

We have discovered that:

- **Most** Linkerd deployments are not up-to-date.
- **Most** Linkerd adopters do not have a way to get notifications about security vulnerabilities
- **Most** Linkerd deployments have issues—sometimes major ones!

Linkerd is fundamentally a tool for adding security and reliability to your system—but Linkerd *cannot fulfill this mission unless it itself is healthy, secure, and up-to-date*.

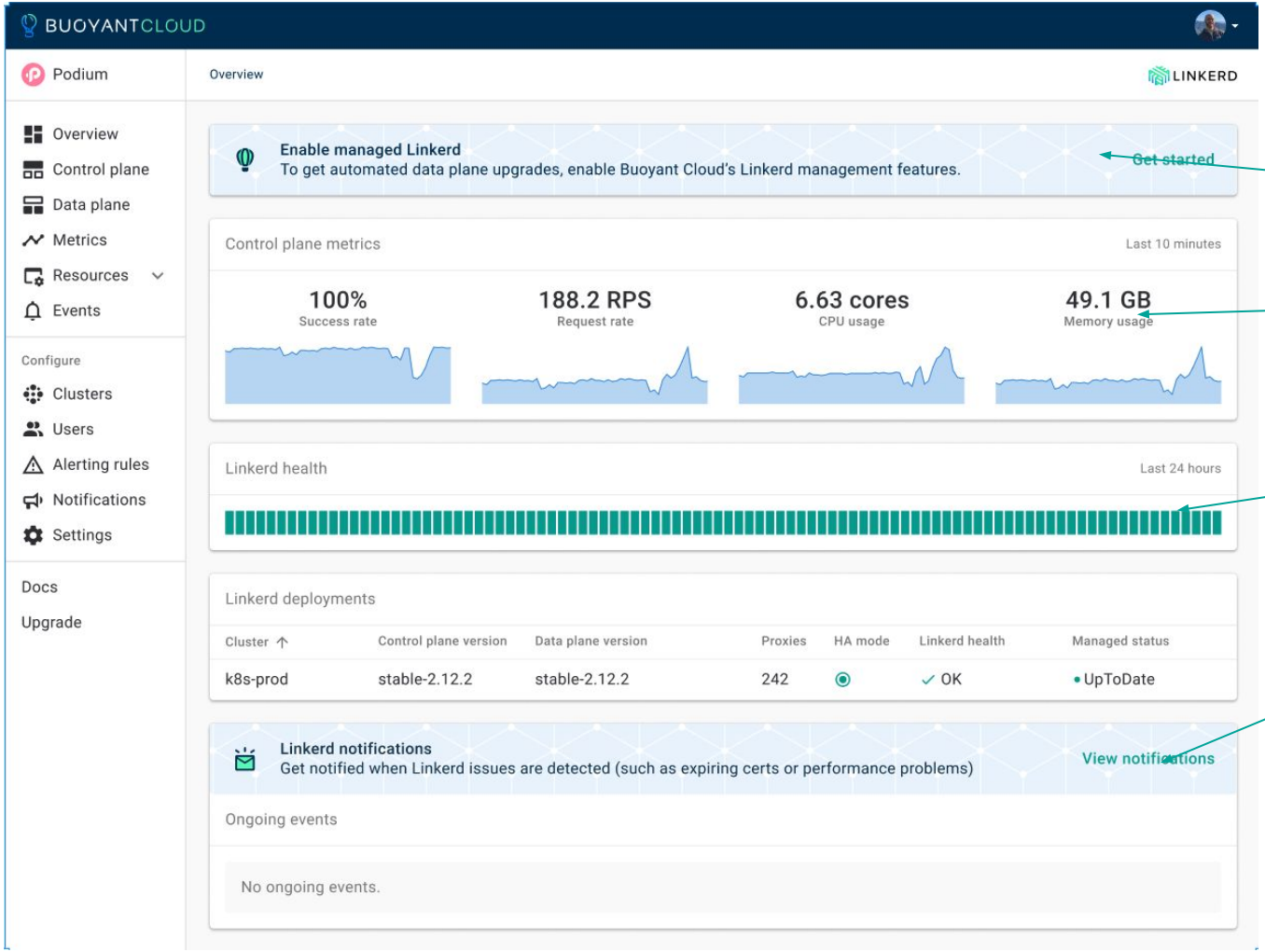
We're opening Buoyant Cloud to the community

As of Linkerd 2.13, Buoyant Cloud will have a free tier that includes:

- **Continuous health monitoring** (including cert alerts)
- **Vulnerability alerting** (e.g. there is a CVE, you need to upgrade)
- **Upgrade assistance** — automatically upgrade data plane proxies when control plane is upgraded
- **TLS rotation assistance**—automatically bounce proxies when a new trust anchor bundle is configured.

Completely free up to 5 clusters. Unlimited users.

BUOYANT
Creators of LINKERD



Automated data plane upgrades

At-a-glance overview across all clusters

Continuous health and vulnerability monitoring

Send alerts to Slack or email

BUOYANT
Creators of LINKERD

FIPS-140-2 SUPPORT

"I need to run my application for the US Government."

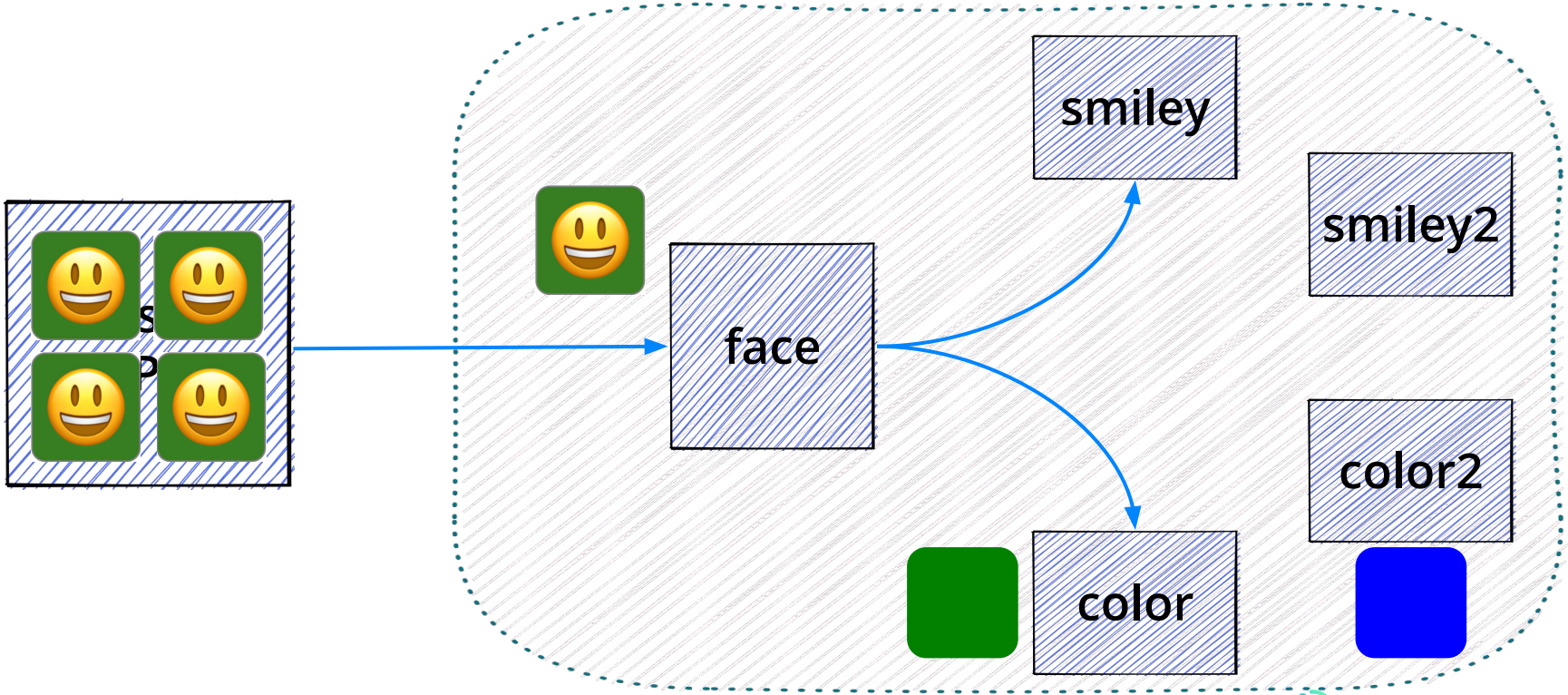
FIPS-140-2 support

- A compliance thing. Lots to say here, but basically if you run cryptography code for the US government (e.g. for mTLS), you need to follow certain rules
  - These rules are arguably worse for security.
  - But they are non-negotiable for compliance.
- Commercial offering. Contact Buoyant for details.

Demo Architecture

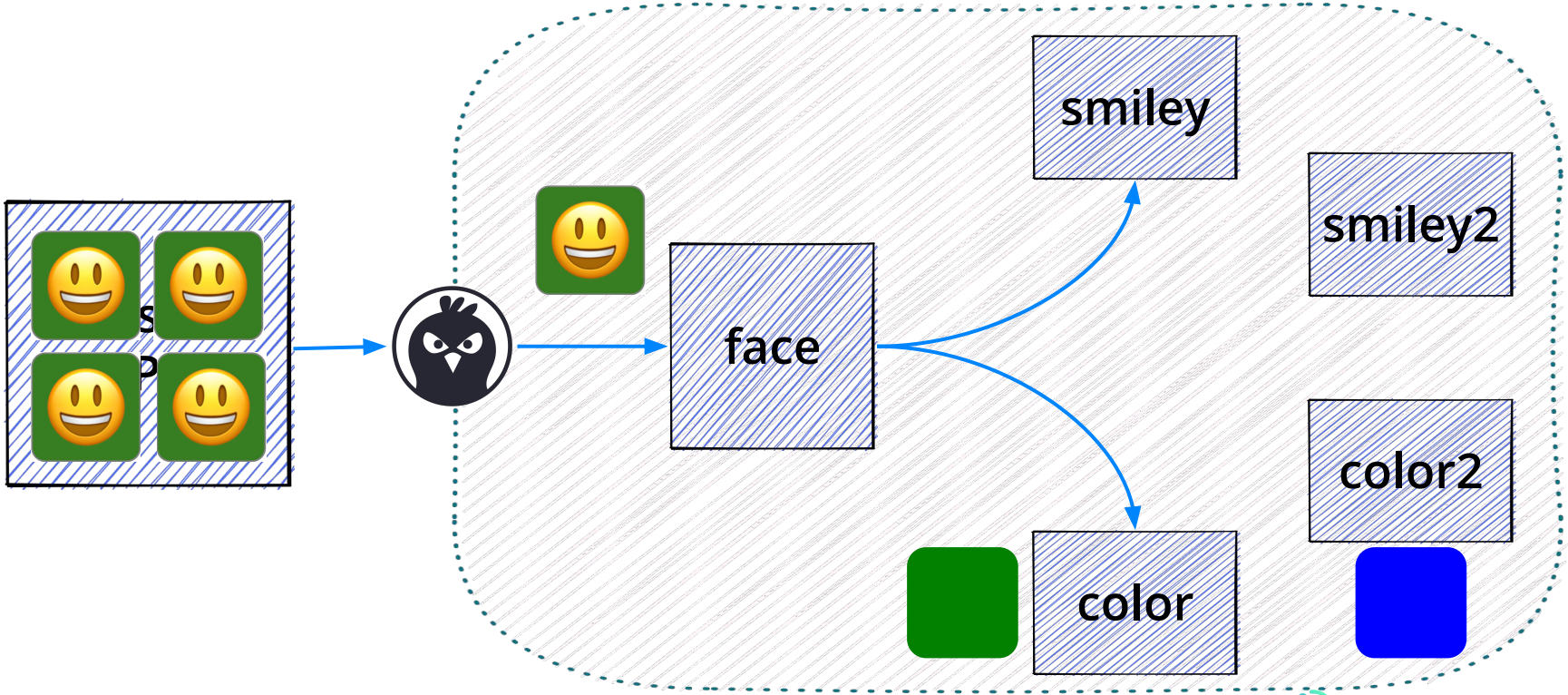


Demo Architecture



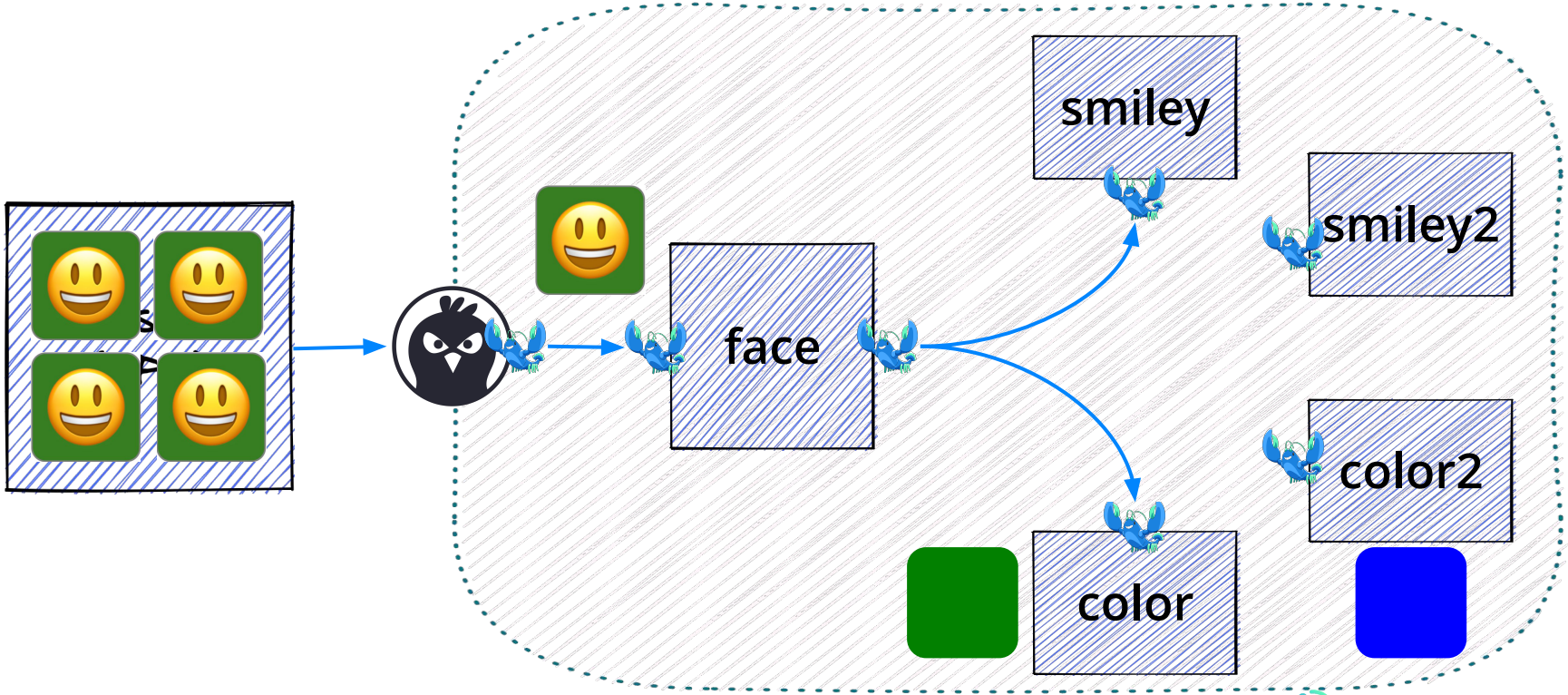
BUOYANT
Creators of LINKERD

Demo Architecture



BUOYANT
Creators of LINKERD

Demo Architecture



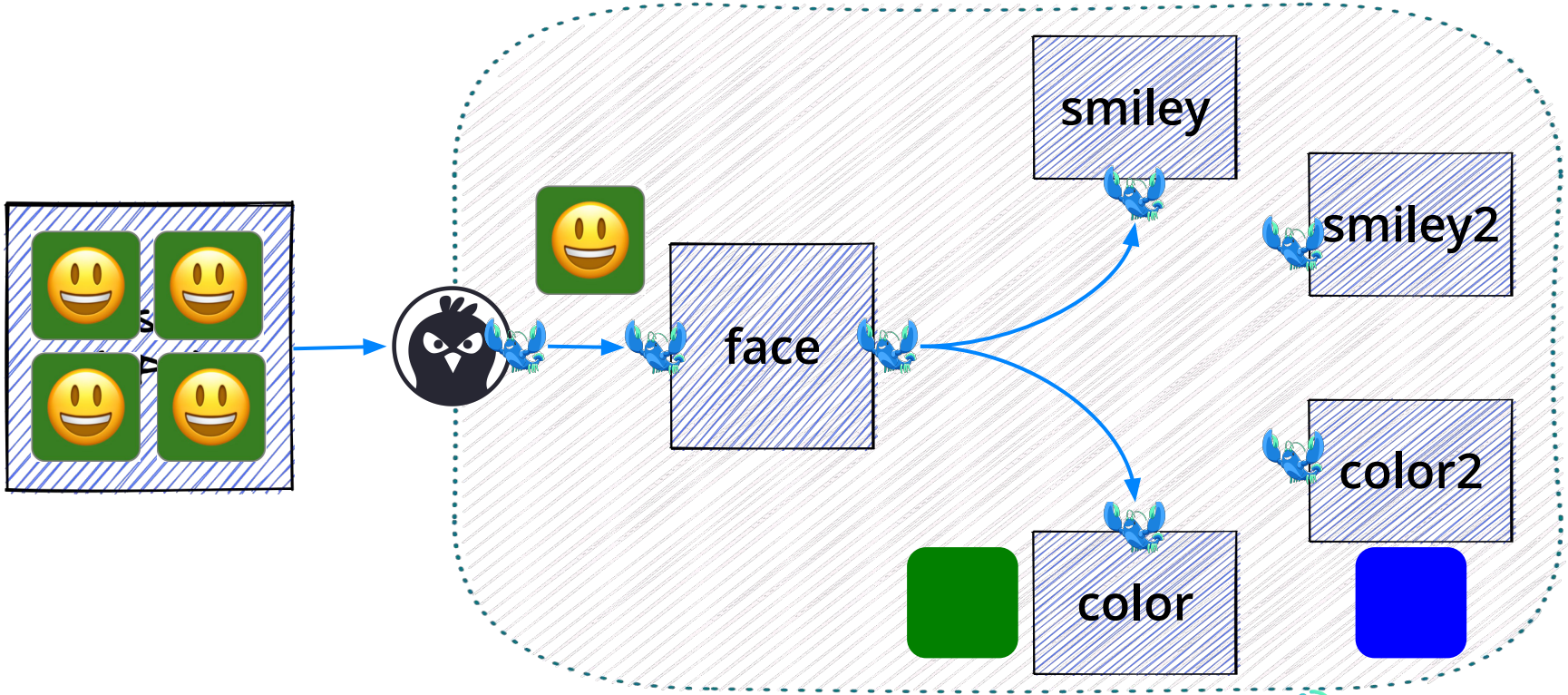
BUOYANT
Creators of LINKERD

Demo Time



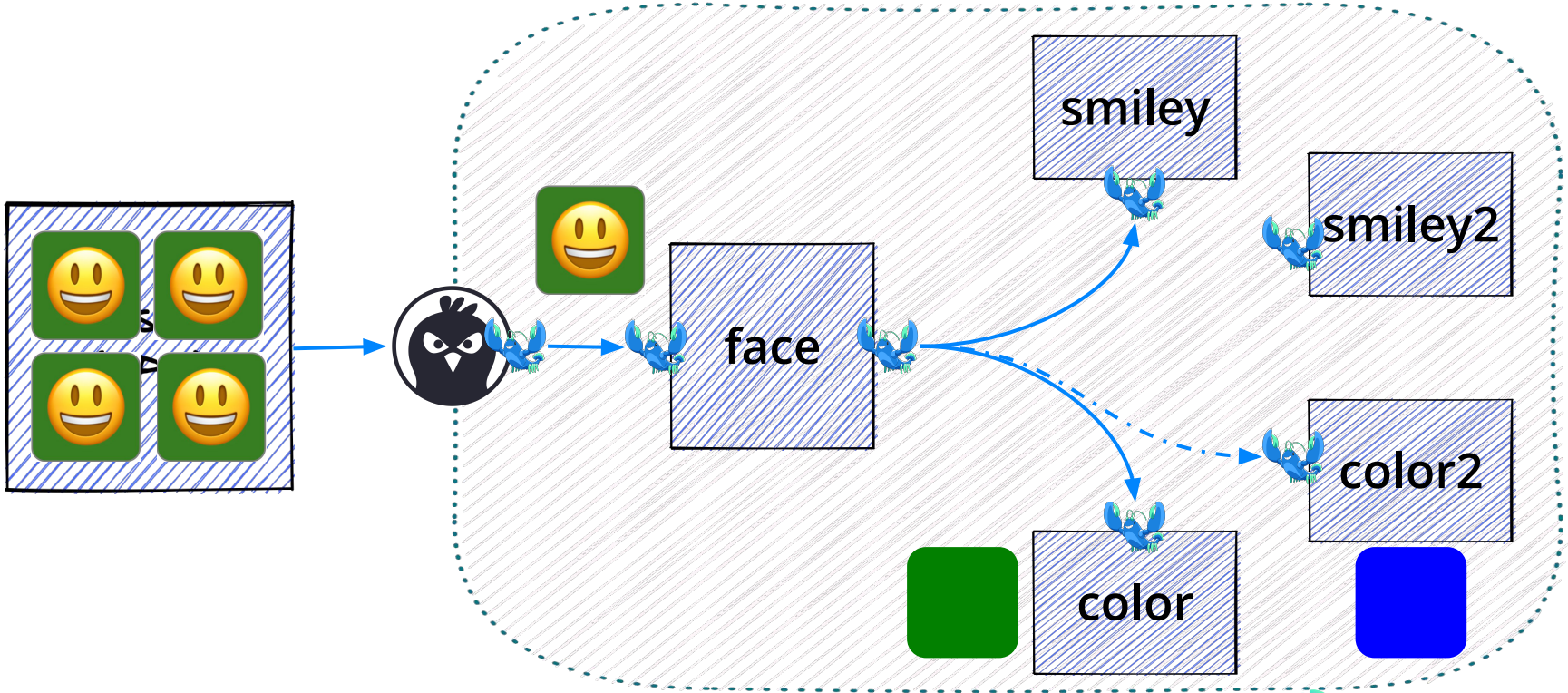


Demo Architecture



BUOYANT
Creators of LINKERD

Demo Architecture



BUOYANT
Creators of LINKERD

Q&A



KUBE CRASH

The KubeCon Amsterdam Warm-Up

Virtual. Free. 100% Open Source.

March 29-30, 2023



Ricardo Torres
Chief Engineer of Open Source,
Boeing



Solomon Hykes
Founder of Docker



Viviane Costa
Lead Platform Engineer
Super Evil Megacorp



Chris Voss
Sr. Software Engineer, **Xbox**

Register at **kubecrash.io**

Follow us on Twitter for updates @KubeCrash

Get fully managed LINKERD on any Kubernetes cluster

Buoyant Cloud automates, maintains, and monitors your open source Linkerd deployment, right on your very own clusters.



Request a demo
buoyant.io/demo

BUOYANT'S
SERVICE MESH
ACADEMY

The creators of
LINKERD

Monthly hands-on, engineer-focused training from the creators of the service mesh

May 18: Circuit Breakers & Dynamic Request Routing Deep Dive



Sign up at
buoyant.io/sma



BUOYANT

Creators of LINKERD

Thanks much!



flynn@buoyant.io
@flynn on slack.linkerd.io



william@buoyant.io
@wm on Twitter

