
KubeCon
CloudNativeCon
Europe 2025
LONDON BUS

Grafana Labs

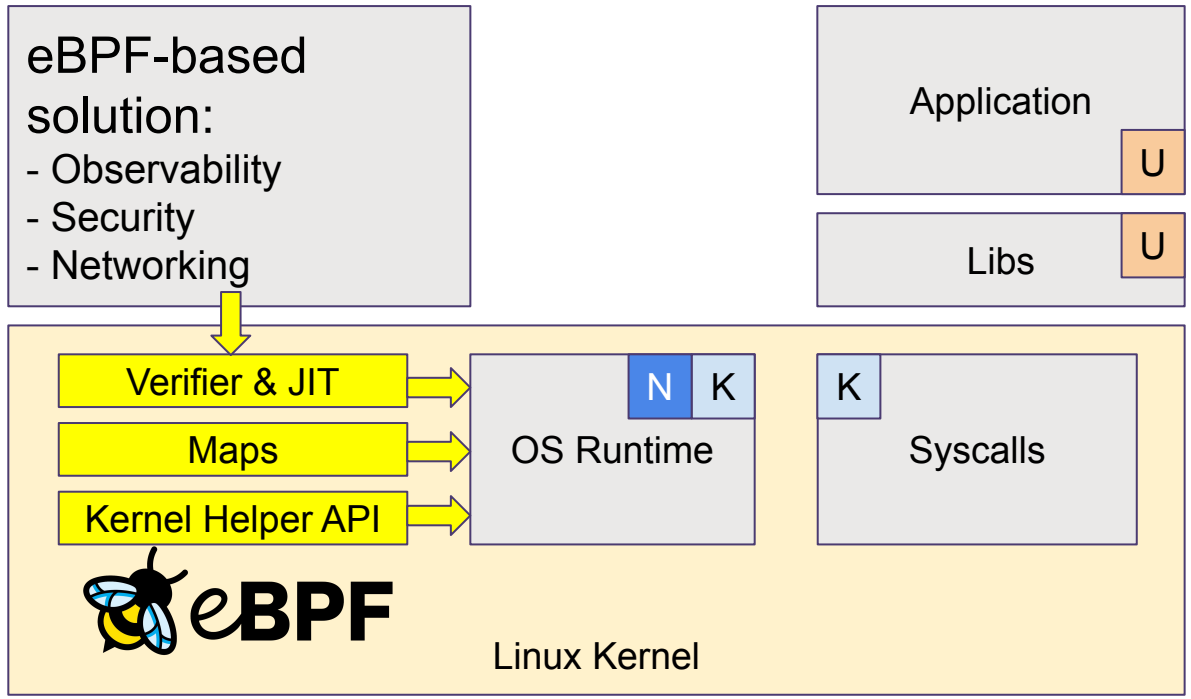
KubeCon | CloudNativeCon
Europe 2025

Using eBPF for non-invasive, instant network monitoring

Marc Tudurí
Staff Software Engineer - Grafana Labs

Mario Macías
Staff Software Engineer - Grafana Labs

eBPF at a glance



- K Kprobes
- U UProbes
- N Network programs

… long etc

eBPF at a glance: observability

- No need to rebuild your code
- No need to redeploy your services
- Native performance (eBPF JIT)
- Safety (eBPF preverification)

- eBPF != “magic”
  - Requires API-level knowledge of instrumented targets
  - Requires binary-level knowledge of data
  - eBPF programs are limited in size and functionalities

Grafana Beyla

Grafana's approach to zero-code automatic instrumentation and network monitoring

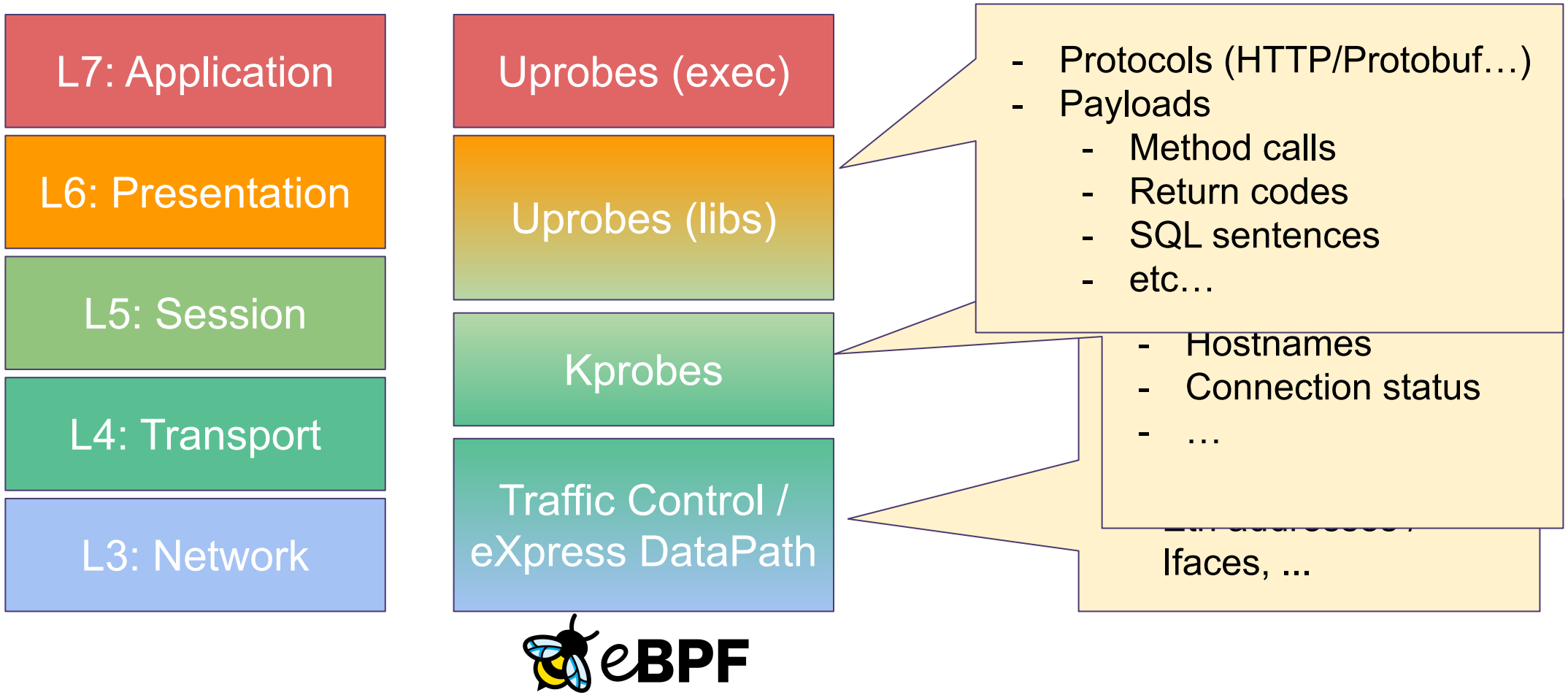
Metrics:

- beyla_network_flow_bytes (L3-L4)
- beyla_network_inter_zone_bytes (L3-L4)
- Application-level metrics (L5-L7, OTEL spec)
  - HTTP/s, gRPC, Kafka, Redis, SQL…

Traces:

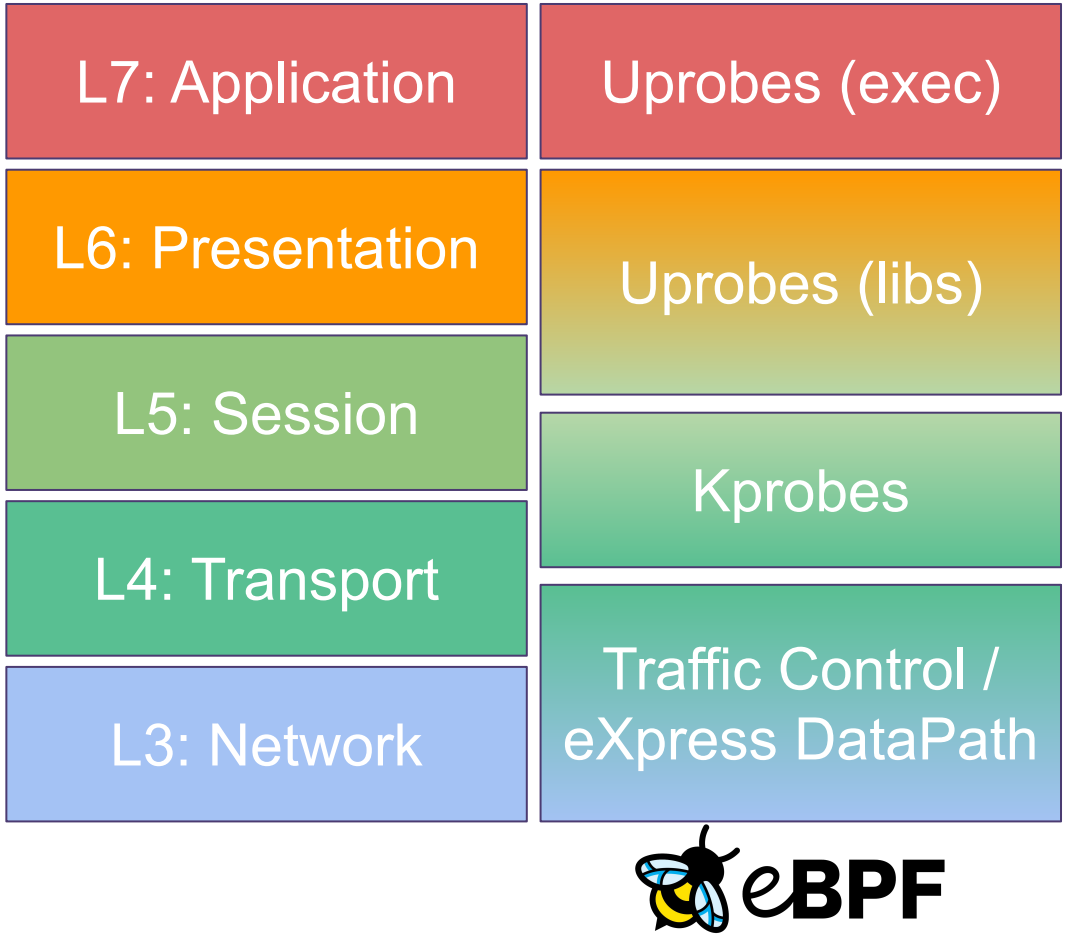
- L5-L7 application-level traces (OTEL)

How to instrument your network

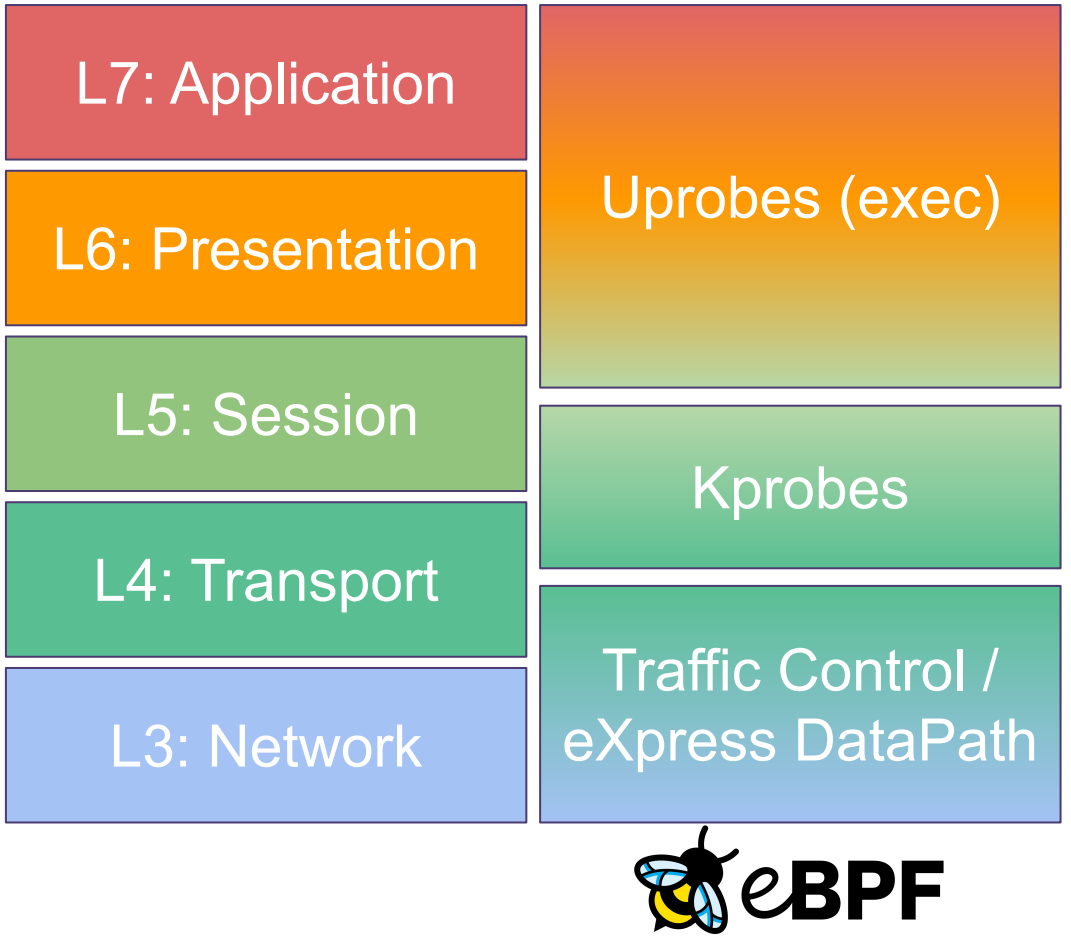


Instrumentation is platform-dependant

C, Rust, Python…



Go, Java…



We have all the puzzle pieces…

Joining pieces for network metrics

Packet events captured by Traffic Control

Time	Src IP	Src Port	Dst IP	Dst Port	…other…	Payload Length
98	…	…	…	…	…	…
100	10.0.0.4	54200	10.0.0.23	80	…	123
101	10.0.0.4	54201	10.0.0.36	3361	…	234
102	10.0.0.33	80	162.168.1.12	50342	…	322
103	10.0.0.4	54200	10.0.0.23	80	…	1234
102	10.0.0.33	80	162.168.1.12	50342	…	101
103	…	…	…	…	…	…

```
beyla_network_flow_bytes{
  src_ip=”10.0.0.4”, src_port=”54200”,
  dst_ip=”10.0.0.23”, dst_port=”80”
} 1357
```

L5-L7: joining pieces

Time	Source	Event
	…	… previous stuff …
12	Socket (kernel)	Connection start (ports 53672→443)
13	Socket (kernel)	Connection start (ports 56380→8080)
15	Socket library	Response Body: “HTTP/1.1 200 OK\n ….”
17	TLS library	Request Body: “GET / HTTP/1.1\nHost: …”
17	HTTP library	Request Body: “POST /users/1234/product HTTP/1.1\nHost: …”
22	Socket library	Content: “(binary stuff) SELECT * FROM Users WHERE….”
23	Golang HTTP uprobes	Request Body: “GET / HTTP/1.1\nHost: …”
	…	… more stuff …

L5-L7: joining pieces

The easy way: classic web servers

Time	Thread ID	Parent Thread	Source	Payload
100	1	…	connect	src/dst address/port, etc…
110	123	1	sock_send	GET /users HTTP/1.1 Host: kube-service.ns User-Agent: …
115			accept	src/dst address/port, socket fd…
130	321	888	sock_recv	(binary stuff…)
143	123	1	sock_recv	HTTP/1.1 200 content-type: text/html; etc…
147	123	1	sock_recv	(more stuff…)
116	321	888	sock_send	(stuff…)
118	123	1	sock_close	

- Client-Side request
- HTTP protocol
- GET /users
- Request payload size
- Status 200 OK
- Response payload time
- Total transaction time

L5-L7: joining pieces

KubeCon CloudNativeCon Europe 2025

Playing in hell mode: modern async servers

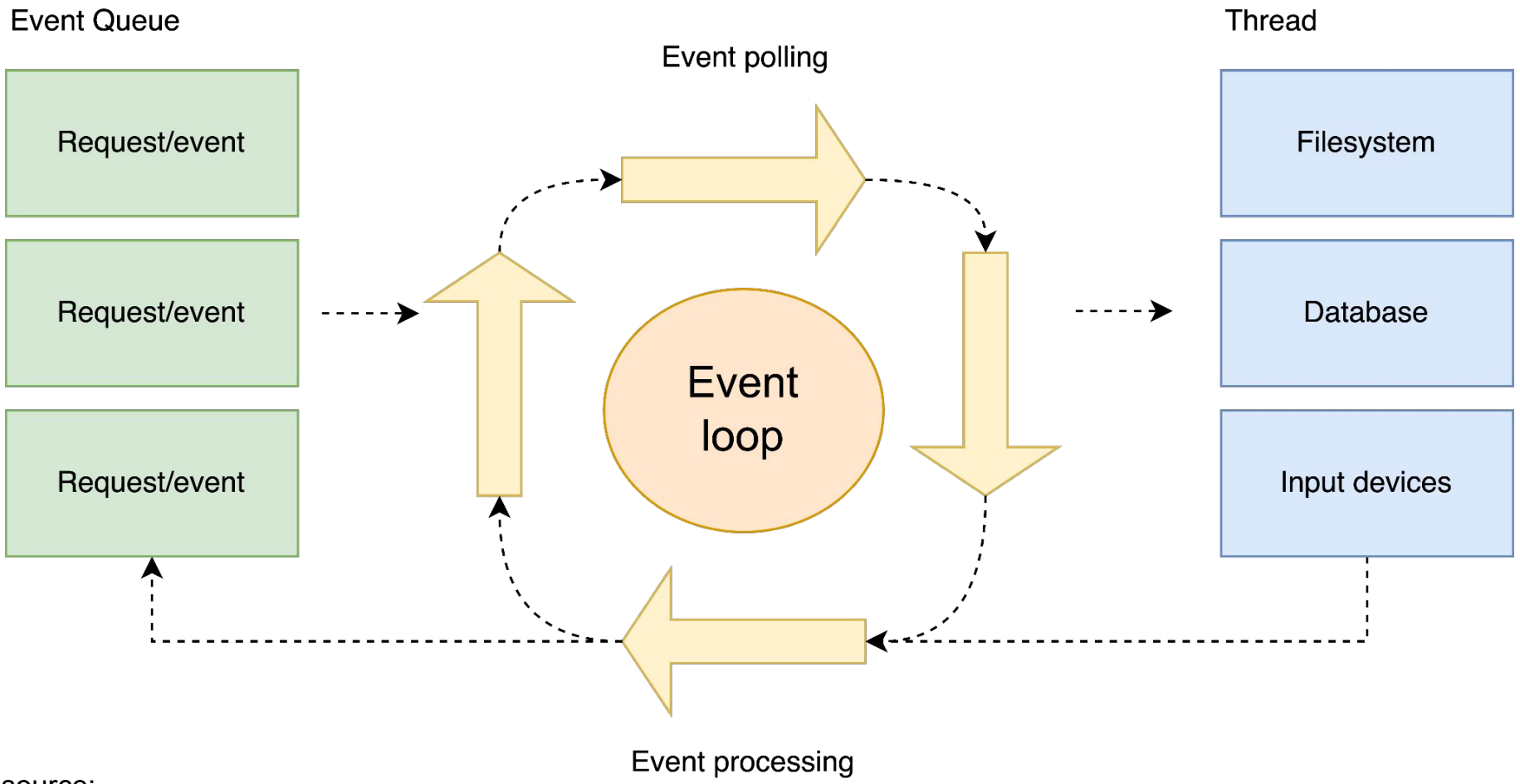
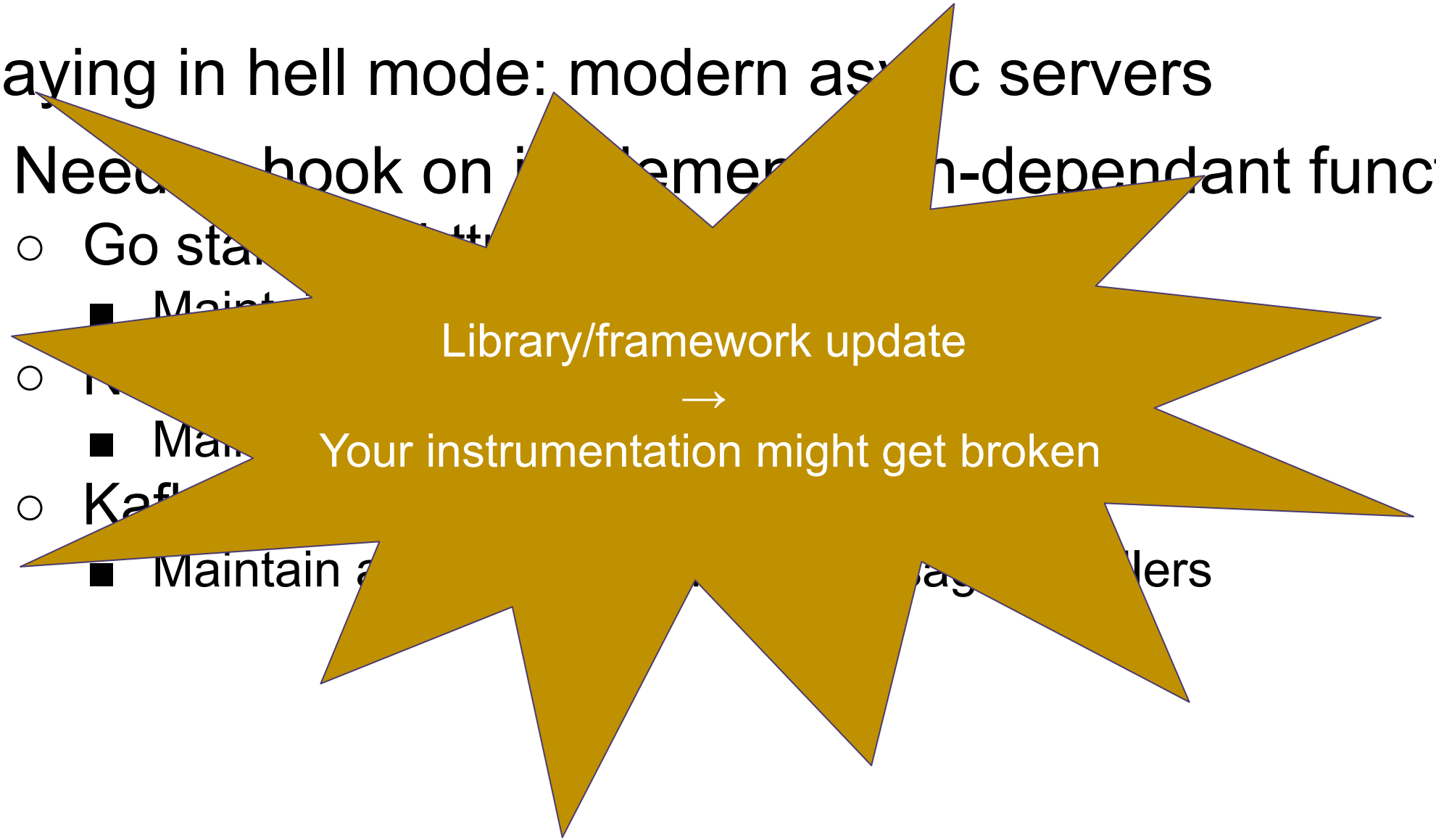


Image source:
https://tech.utugit.fi/soft/tools/lectures/dtek2054/2022/events/eventloop/

L5-L7: joining pieces

Playing in hell mode: modern asy c servers

- Need hook on i emen n-dependant functions
  - Go sta tt
    - Maint
  - K
    - Mai
  - Kaf
    - Maintain a ag lers



L3-L4: network-level metrics

- ✌️ Robust: Based on stable APIs and standard binary representations (TCP, UDP, IP… packets)

- 🤦 Basic information
  - Src/Dst endpoints
  - Packet count
  - Bytes sum

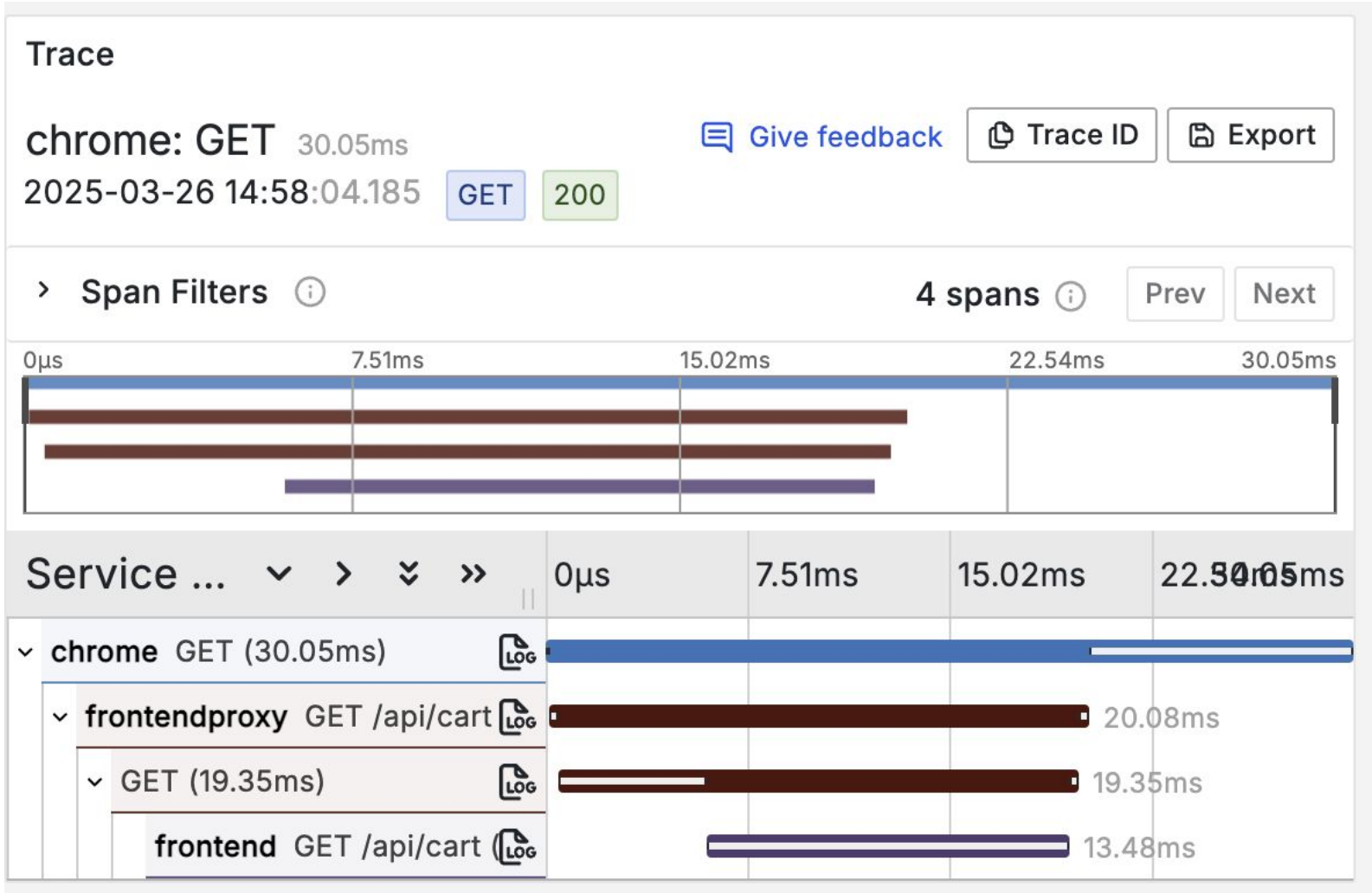
L5-L7: application-level metrics / traces

- ✌️ Rich information
  - HTTP/GRPC: methods, response codes, request times…
  - Other protocols: Kafka, Redis, SQL…

- 🤦 Need to implement explicit support to any new implementation of a protocol

- 😭 Relies on internal implementation details that can change with time

Trace context propagation



HTTP

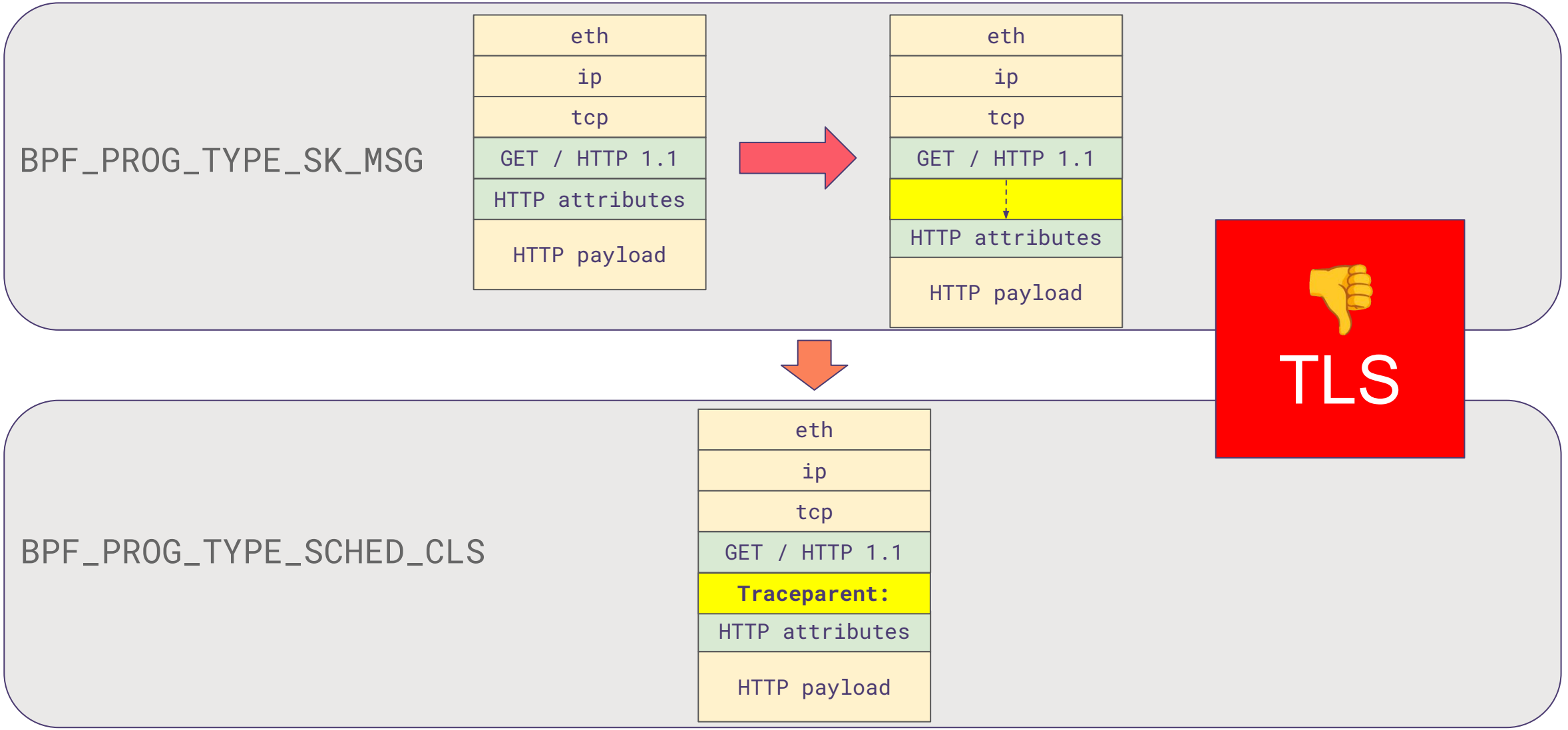
- Traceparent header

gRPC

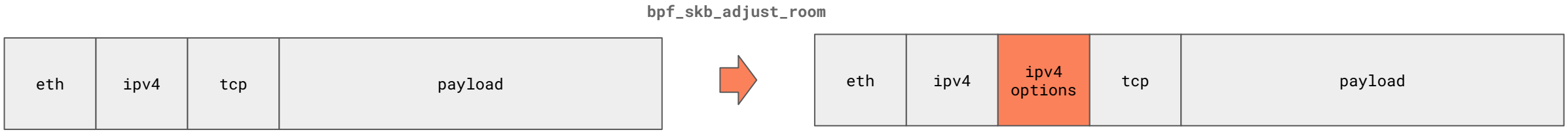
- Metadata header

Instrumentation SDKs need to explicitly read it from inbound requests and inject it in outbound

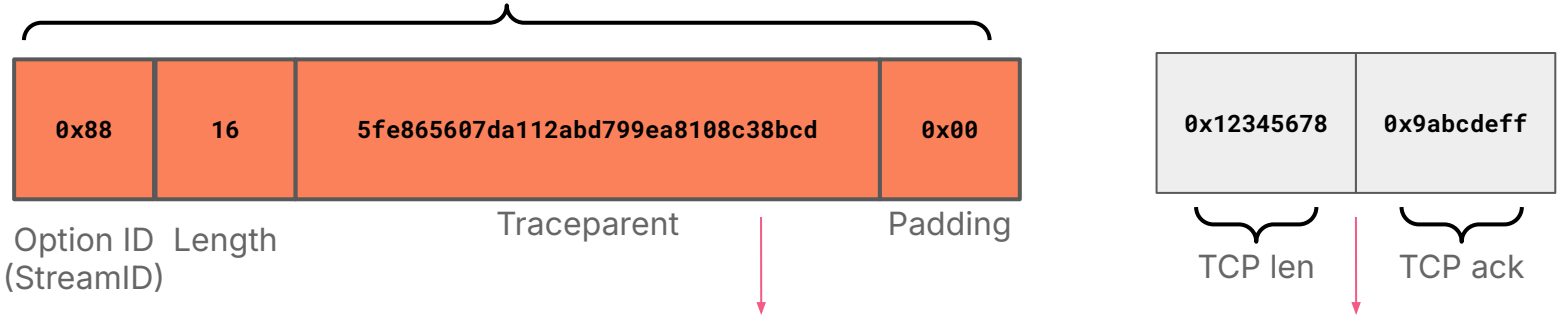
HTTP context propagation



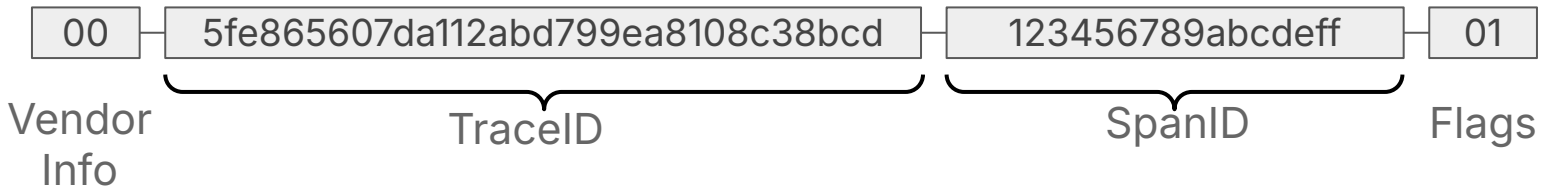
IP context propagation



Egress



Ingress



Grafana Labs

KubeCon | CloudNativeCon
Europe 2025

Kubernizing your data

What eBPF sees


```
beyla_network_flow_bytes{
  src_address=”10.0.0.4”,
  src_port=”54200”,
  dst_address=”10.0.0.23”,
  dst_port=”80”
} 1357

http_server_request_duration_sum {
  service_name=”java”,
  url_route=”/users/{id}/products”,
} 84578547
```

What K8s users need

KubeCon CloudNativeCon Europe 2025

```
beyla_network_flow_bytes{
  src_address=”10.0.0.4”,
  src_port=”54200”,
  dst_address=”10.0.0.23”,
  dst_port=”80”
} 1357

http_server_request_duration_sum {
  service_name=”java”,
  url_route=”/users/{id}/products”,
} 84578547
```

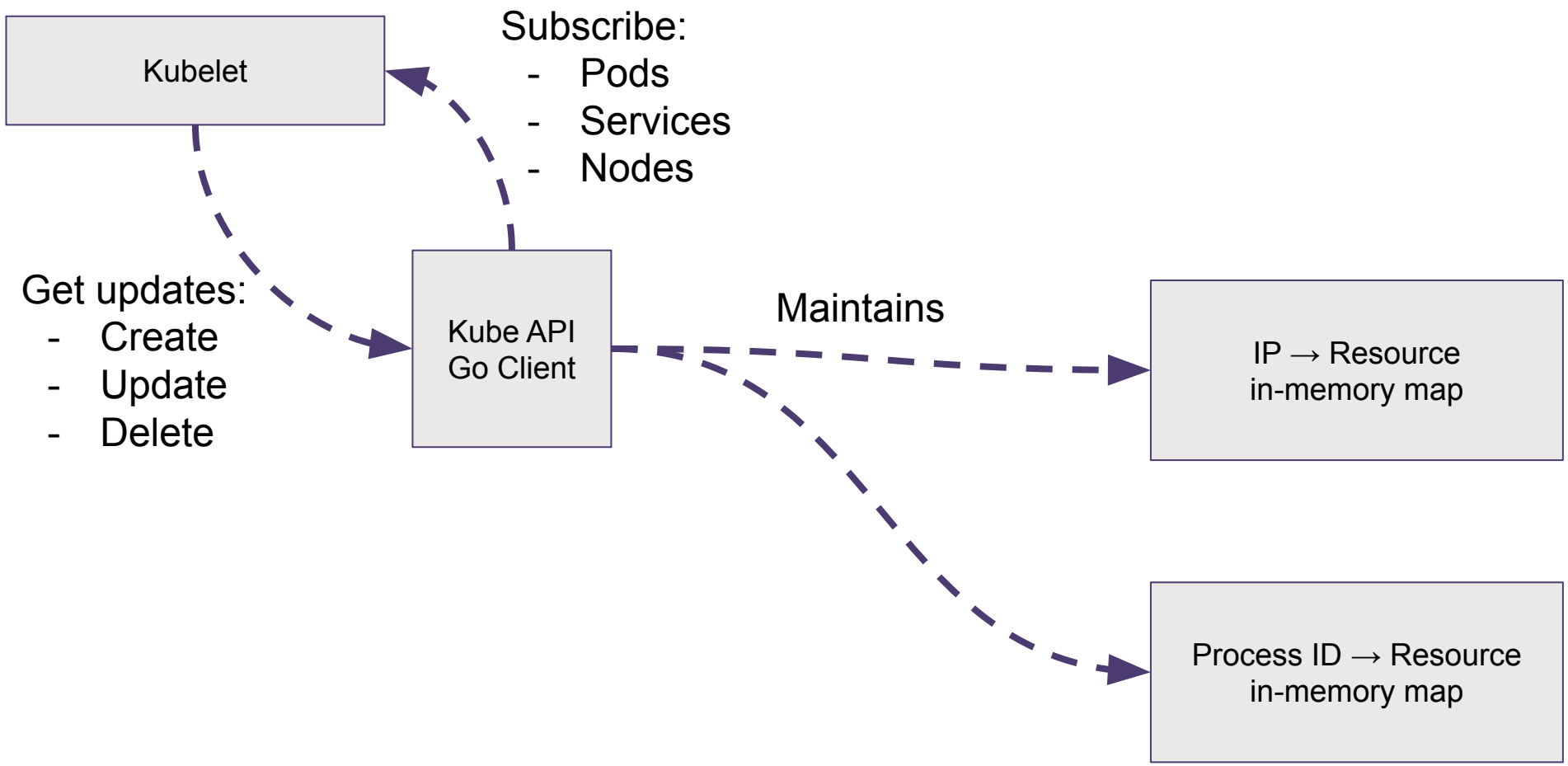
```
beyla_network_flow_bytes{
  k8s_src_owner=”frontend”,
  k8s_src_namespace=”app”
  k8s_dst_owner=”database”
  k8s_src_namespace=”storage”
} 1357

http_server_request_duration_sum {
  service_name=”inventory”,
  service_namespace=”backend”
  url_route=”/users/{id}/products”,
} 84578547
```

Grafana Beyla

Grafana Labs

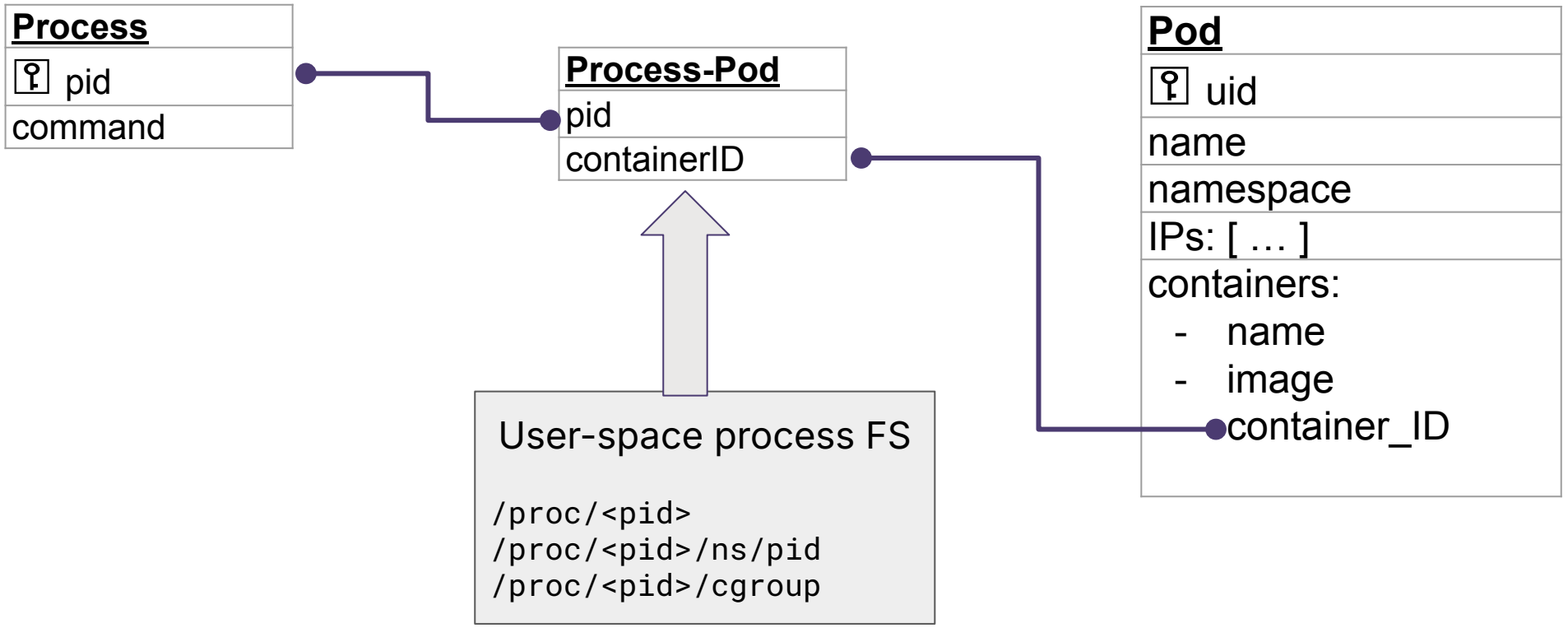
Informers to the rescue



Matching a process with a K8s Pod

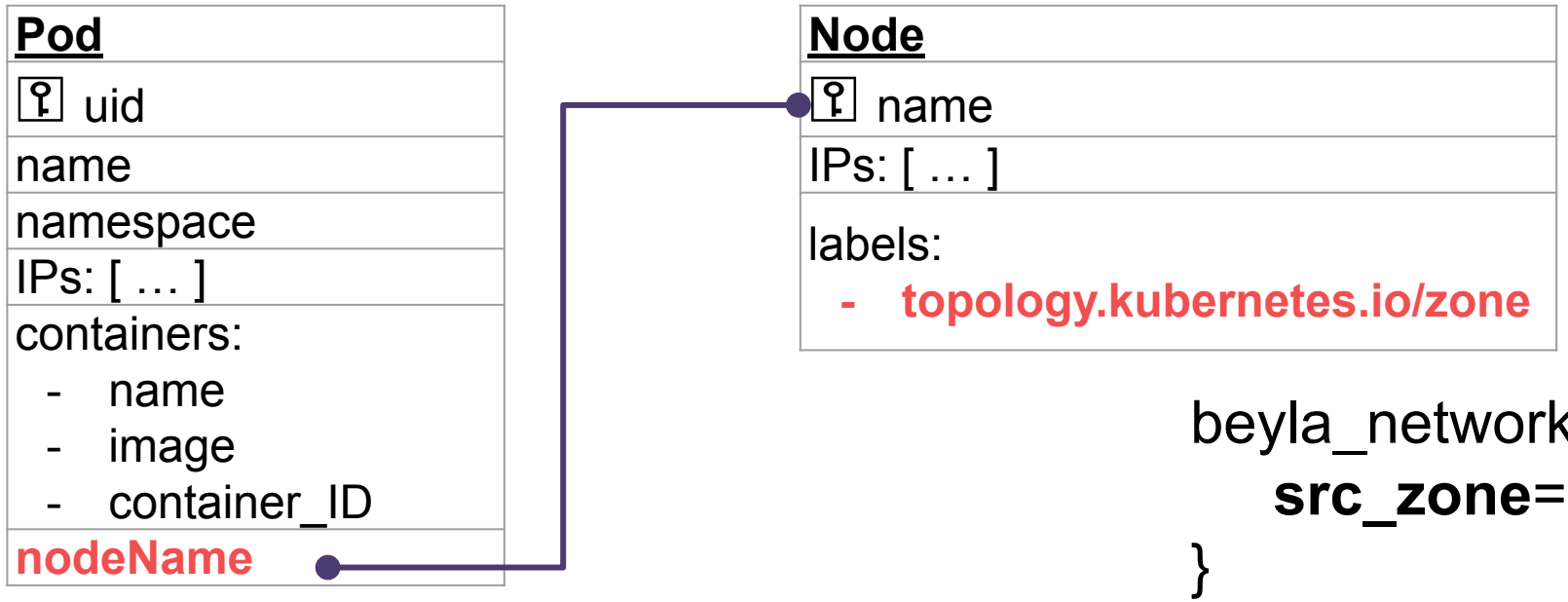
What eBPF sees

What K8s sees



Inter-zone traffic

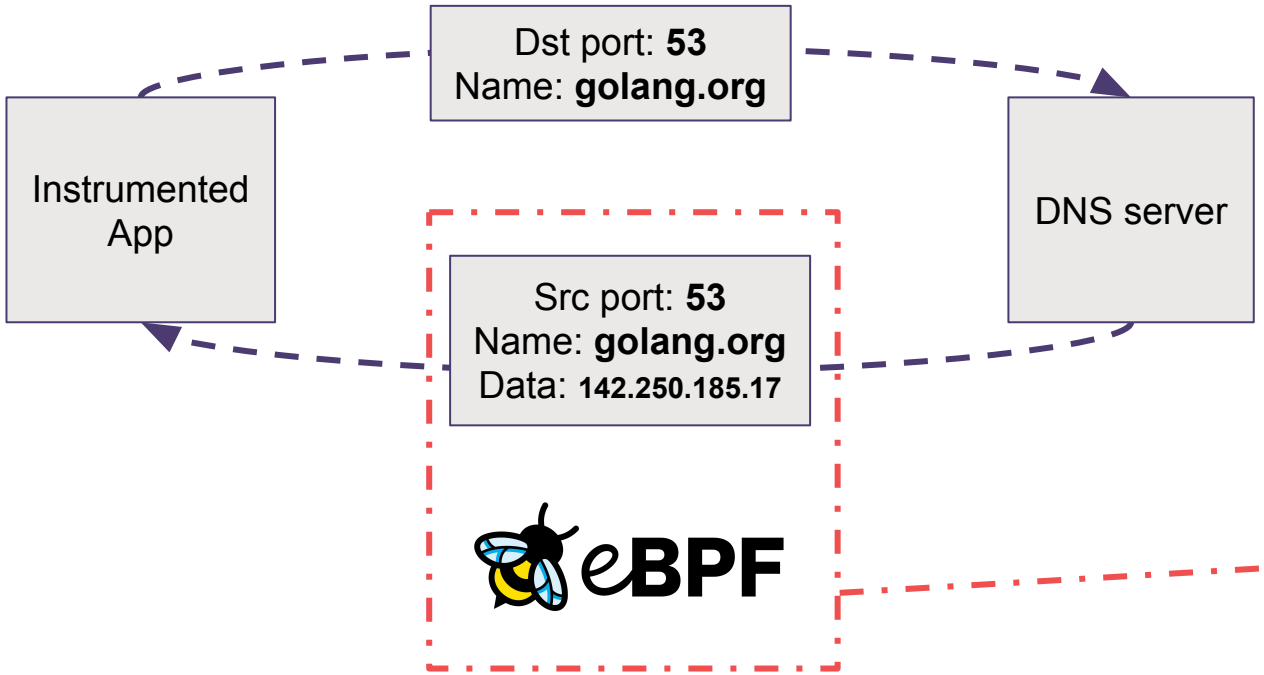
Traffic between Cloud Availability-Zones might incur in high extra costs




```
beyla_network_flow_bytes {
    src_zone="...", dst_zone="...",
}
beyla_network_inter_zone_bytes {
    src_zone="...", dst_zone="...",
}
```

External traffic: Reverse DNS


```
$ ping -c 1 golang.org
PING golang.org (142.250.185.17): 56 data bytes
…
$ nslookup 142.250.185.17
…
17.185.250.142.in-addr.arpa    name = mad41s11-in-f17.1e100.net.
```




```
network_flow_bytes{
  …
  dst_address="142.250.185.17",
  dst_name="golang.org",
} 1357
```

Dealing with cardinality

KubeCon CloudNativeCon Europe 2025

Attributes of Beyla metrics

```yaml
attributes:
  select:
    beyla_network_flow_bytes:
      include:
        - k8s.src.owner.*
        - k8s.dst.owner.*
        - src.zone
        - dst.zone
    sql_client_duration:
      # report all the possible attributes but db_statement
      include: ["*"]
      exclude: ["db_statement"]
    http_*:
      # report the default attribute set but exclude the Kubernetes Pod information
      exclude: ["k8s.pod.*"]
```

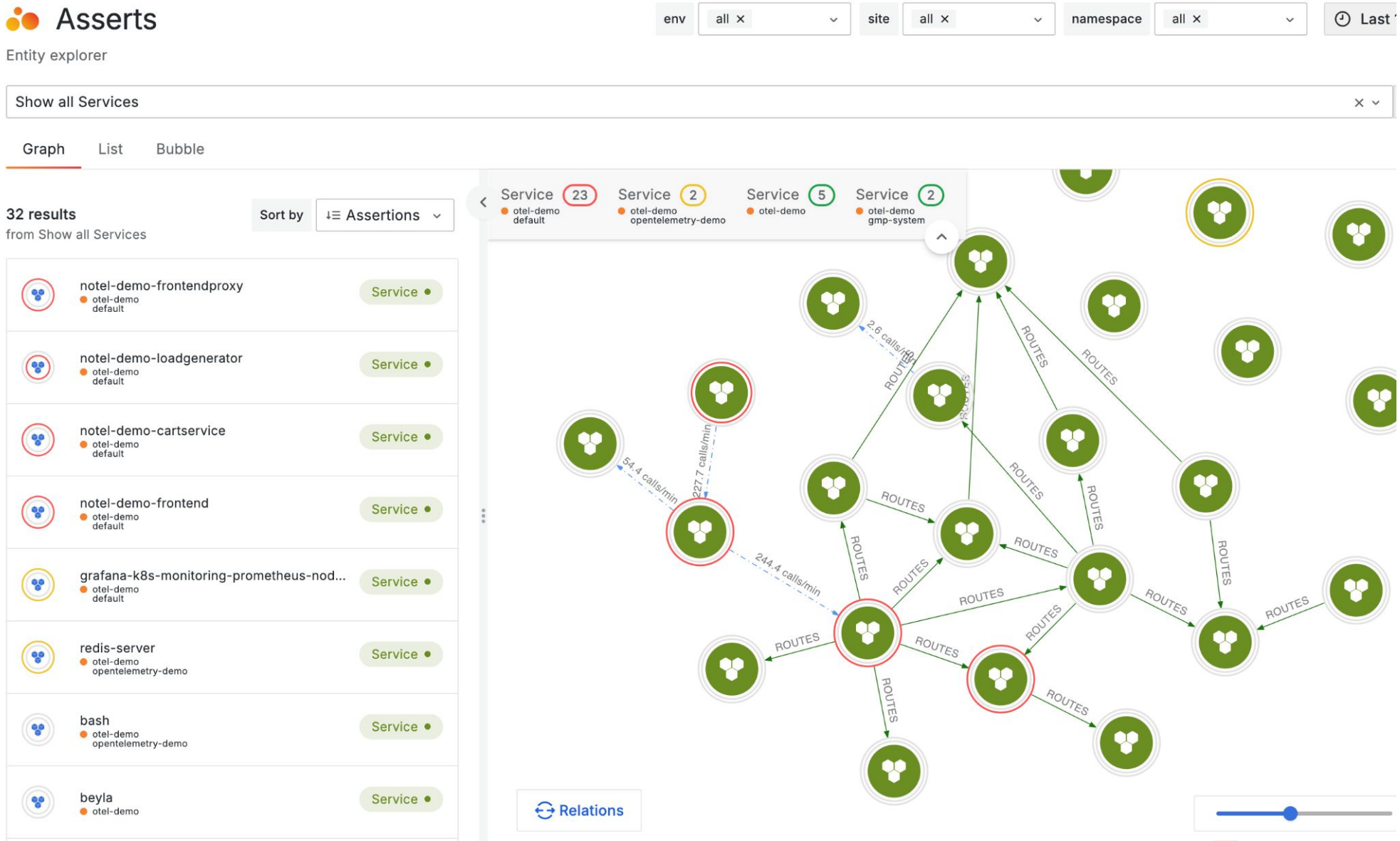
Application (all)	`k8s.statefulset.name`	shown if Kubernetes metadata is enabled
Application (all)	`k8s.replicaset.name`	shown if Kubernetes metadata is enabled
Application (all)	`k8s.statefulset.name`	shown if Kubernetes metadata is

`beyla.network.flow.bytes`	`client.port`	hidden
`beyla.network.flow.bytes`	`direction`	hidden
`beyla.network.flow.bytes`	`dst.address`	hidden
Traces (SQL, Redis)	`db.query.text`	hidden

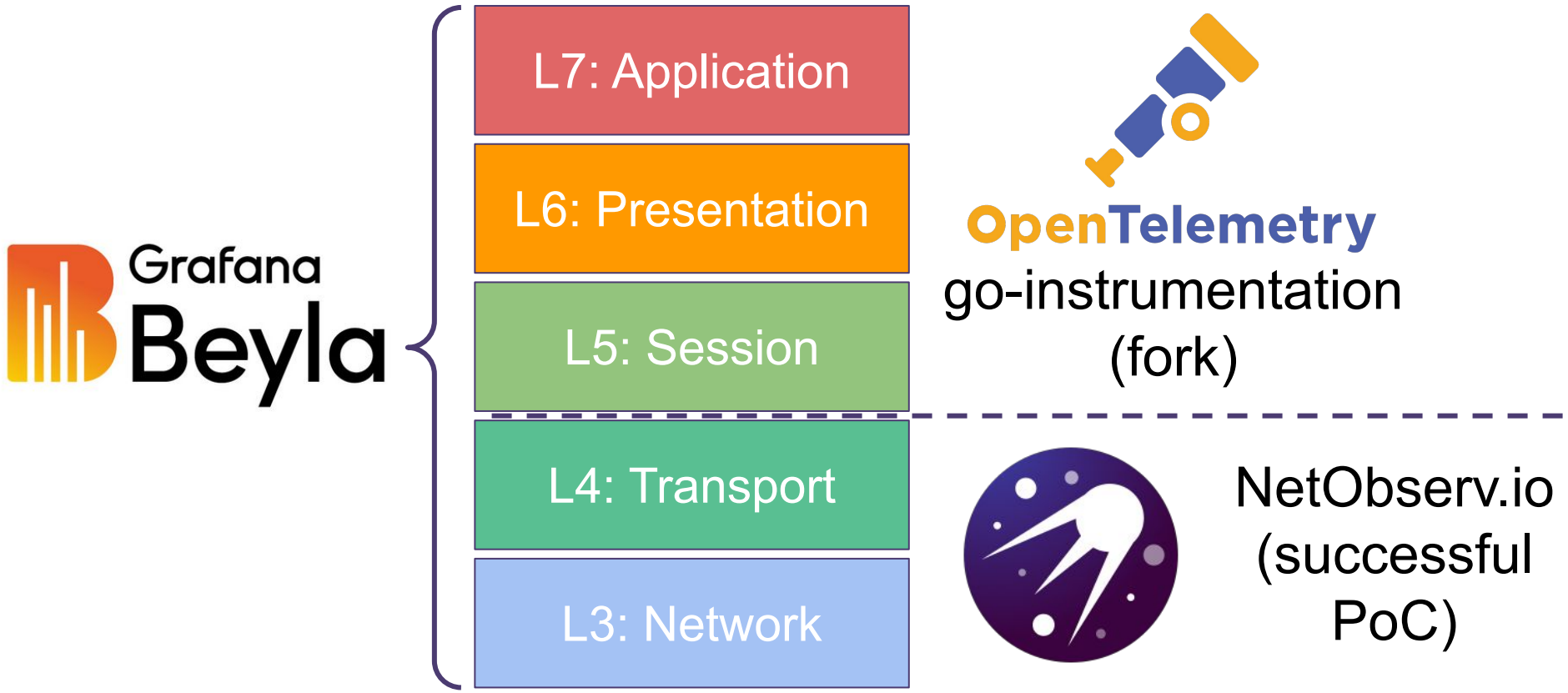
Beyla

Grafana Labs

Provide meaning to IPs and ports



Over the shoulders of giants



OSS Community: a core principle

open-telemetry / **community**

Code | Issues 97 | Pull requests 2 | Actions | Projects 4 | Security

[Donation Proposal]: Beyla, eBPF auto-instrumentation tool for metrics and traces #2406

Open

grcevski opened on Oct 23, 2024 · edited by grcevski — Edits

Description

Grafana Labs would like to offer the donation of Beyla to the OpenTelemetry project.

Beyla is a mature eBPF-based auto-instrumentation tool for OpenTelemetry

Assignees

No one assigned

Labels

area/donation

Grafana Labs

KubeCon | CloudNativeCon
Europe 2025

Using eBPF for non-invasive, instant network monitoring

Marc Tudurí
Staff Software Engineer - Grafana Labs

Mario Macías
Staff Software Engineer - Grafana Labs