Envoy Mobile:
From Server to Multiplatform Library

KubeCon - November 2019

Michael Schore
@goaway

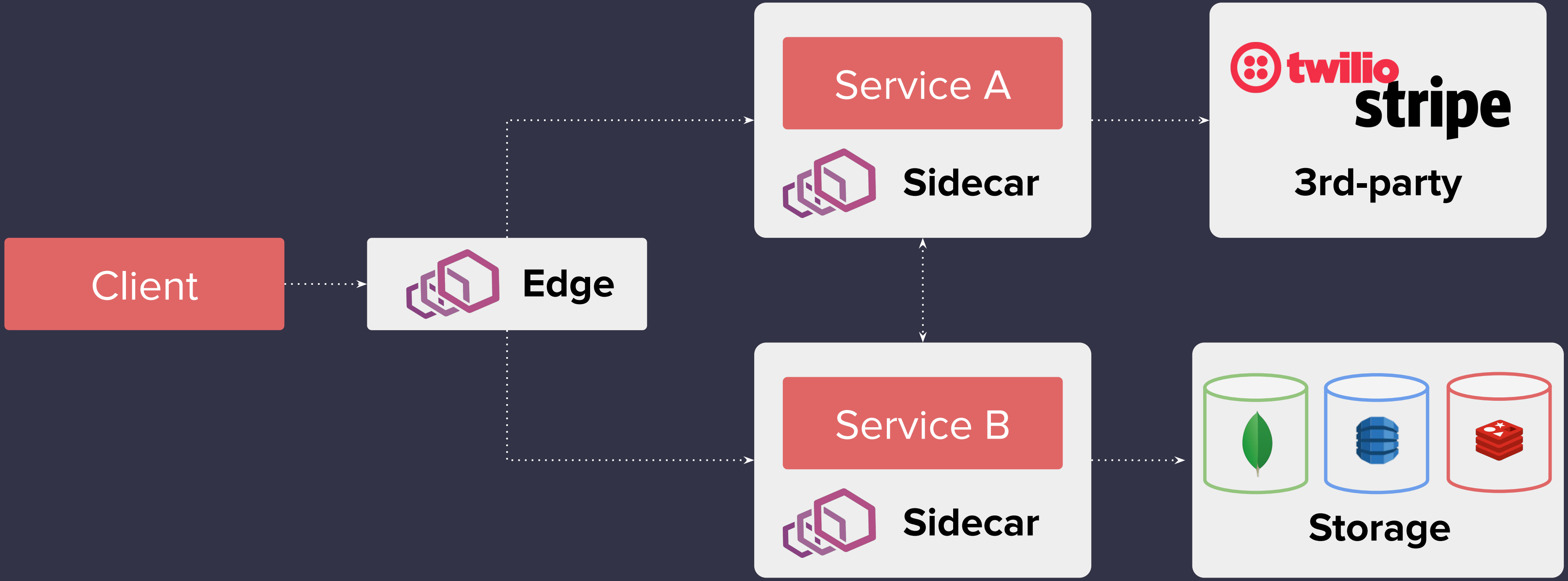
Jose Nino
@junr03

Agenda

- Why bring Envoy to Mobile?
- Envoy as a Library
- Where are we now?
- Onwards!

Why bring Envoy …to Mobile?

Topology 2.0: Universal Network Primitive



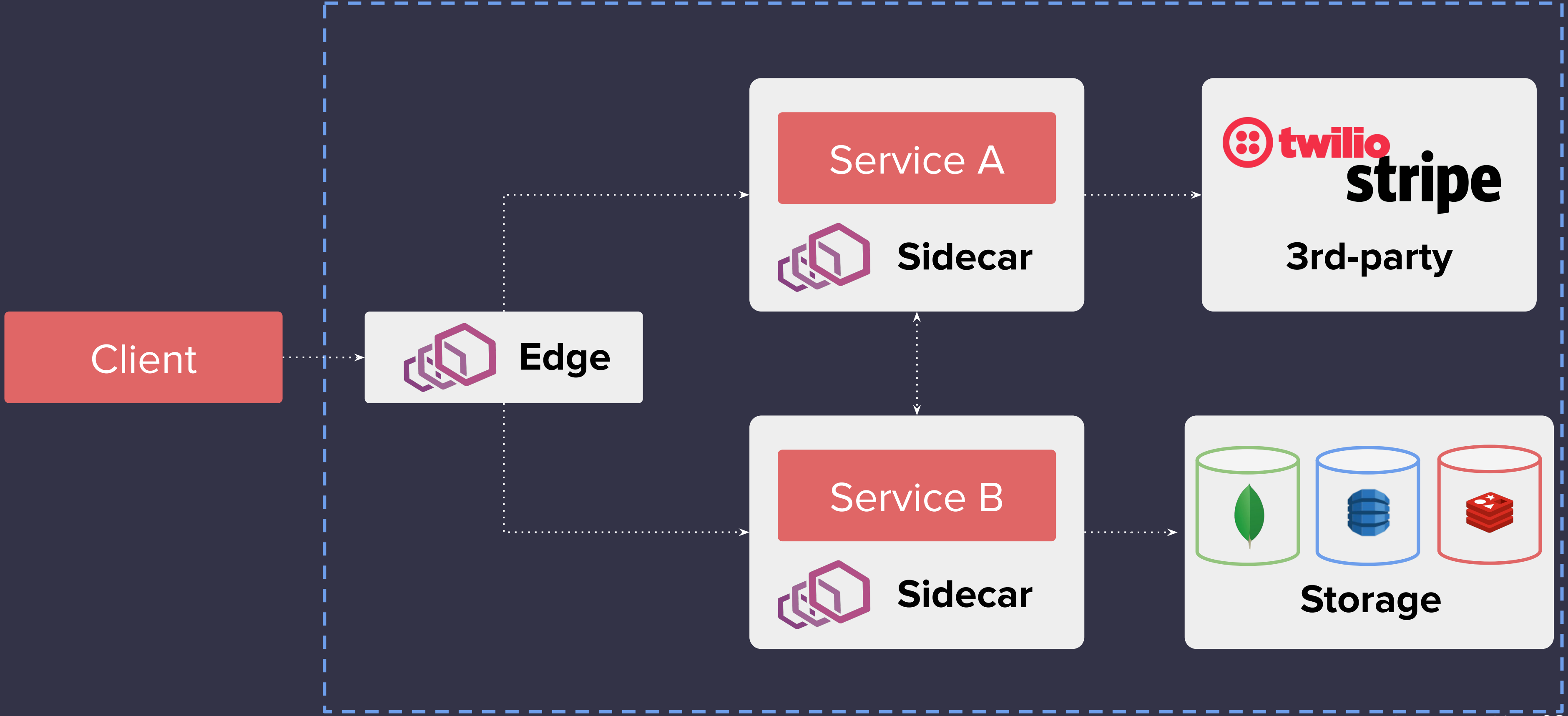
@goaway @junr03

What are we solving for?

Three 9s at the server-side edge is meaningless if the user of a mobile application is only able to complete the desired product flows a fraction of the time.

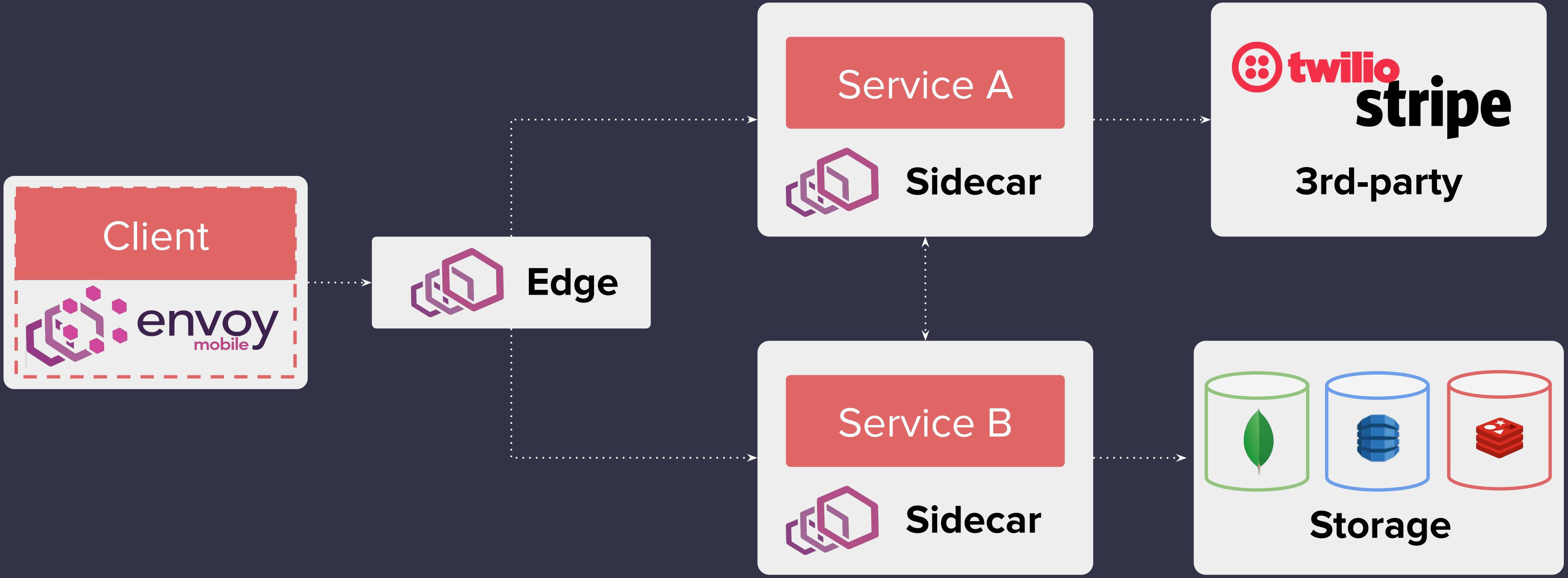
Performance	?	✓
Reliability	?	✓
Extensibility	?	✓
Observability	?	✓
Configuration API	?	✓

Topology 2.0: ~~Universal~~ Network Primitive



@goaway @junr03

Topology 3.0: Universal Network Primitive



@goaway @junr03

Standardizing infrastructure

envoy mobile

envoy

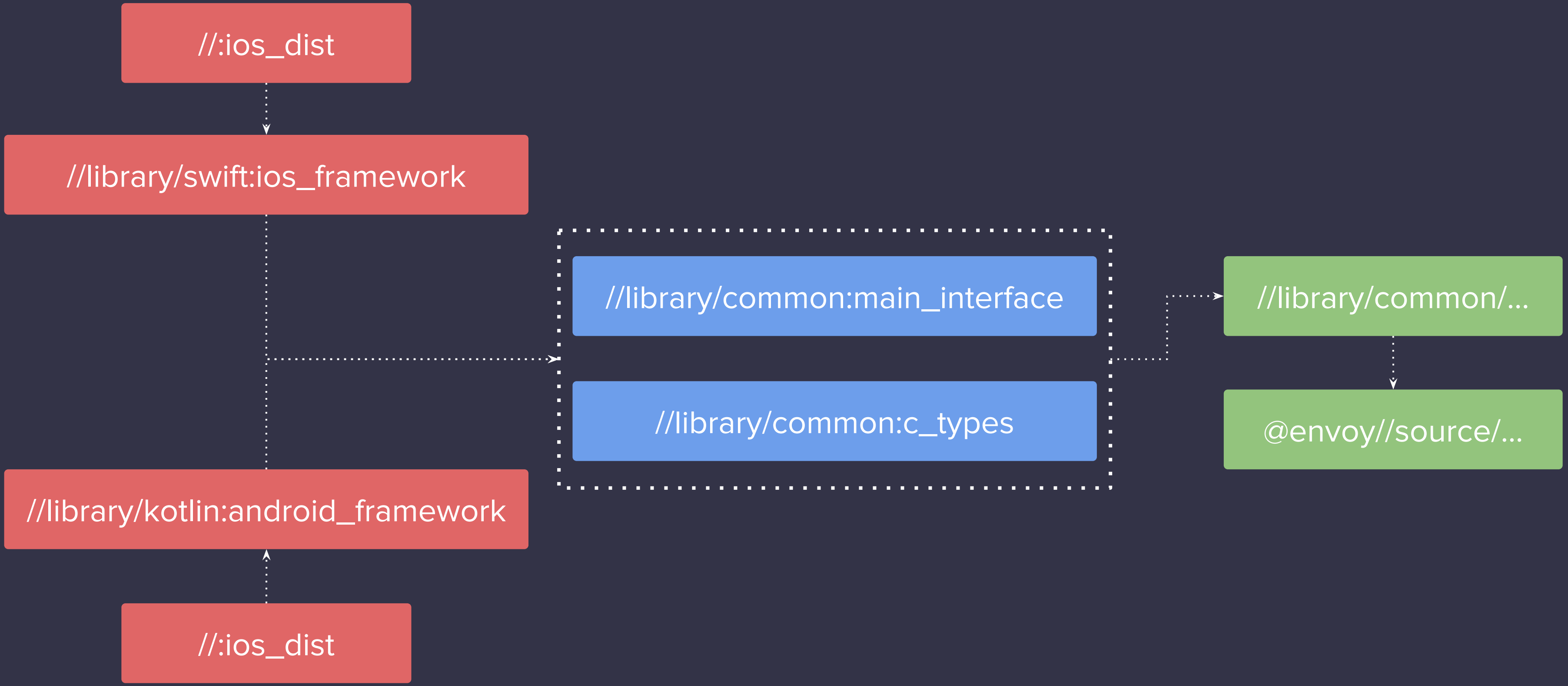
@goaway @junr03

Why is ~~world domination~~ standardization useful?

- Write once, deploy everywhere
- Common tooling for common problems
- Reduce cognitive load

Envoy as a Library

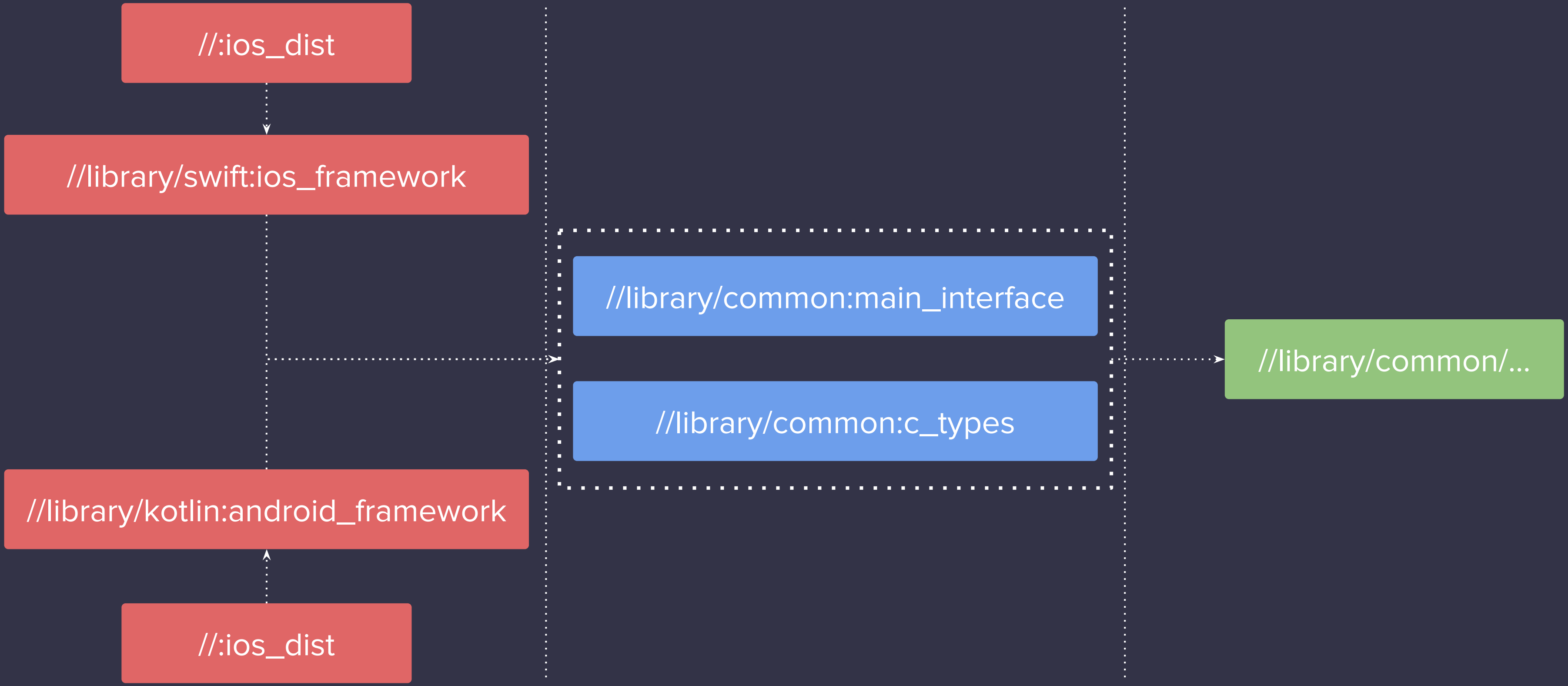
Build System



//:ios_dist
//library/swift:ios_framework
//library/common:main_interface
//library/common:c_types
//library/common/...
@envoy//source/...
//library/kotlin:android_framework
//:ios_dist

@goaway @junr03

Build System


//:ios_dist
//library/swift:ios_framework
//library/common:main_interface
//library/common:c_types
//library/common/...
//library/kotlin:android_framework
//:ios_dist

@goaway @junr03

API - Layered Design

Platform	Bridge	Native
Thin platform Code	bridging over C bindings	leveraging C++ native code.

How to run a process in an app?

picture of an engine (a very fast one)

Threading contexts

Application Threads

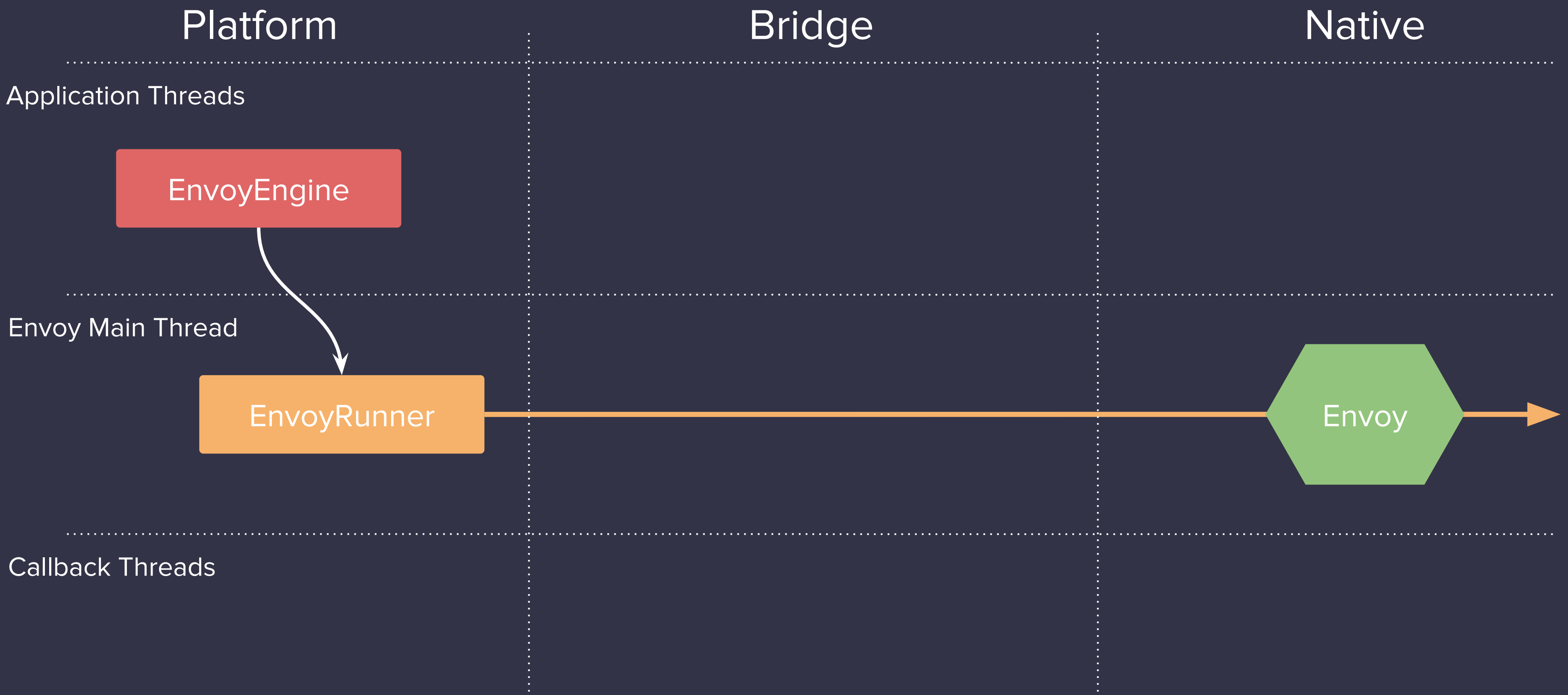
Envoy Main Thread

Callback Threads

Library Matrix

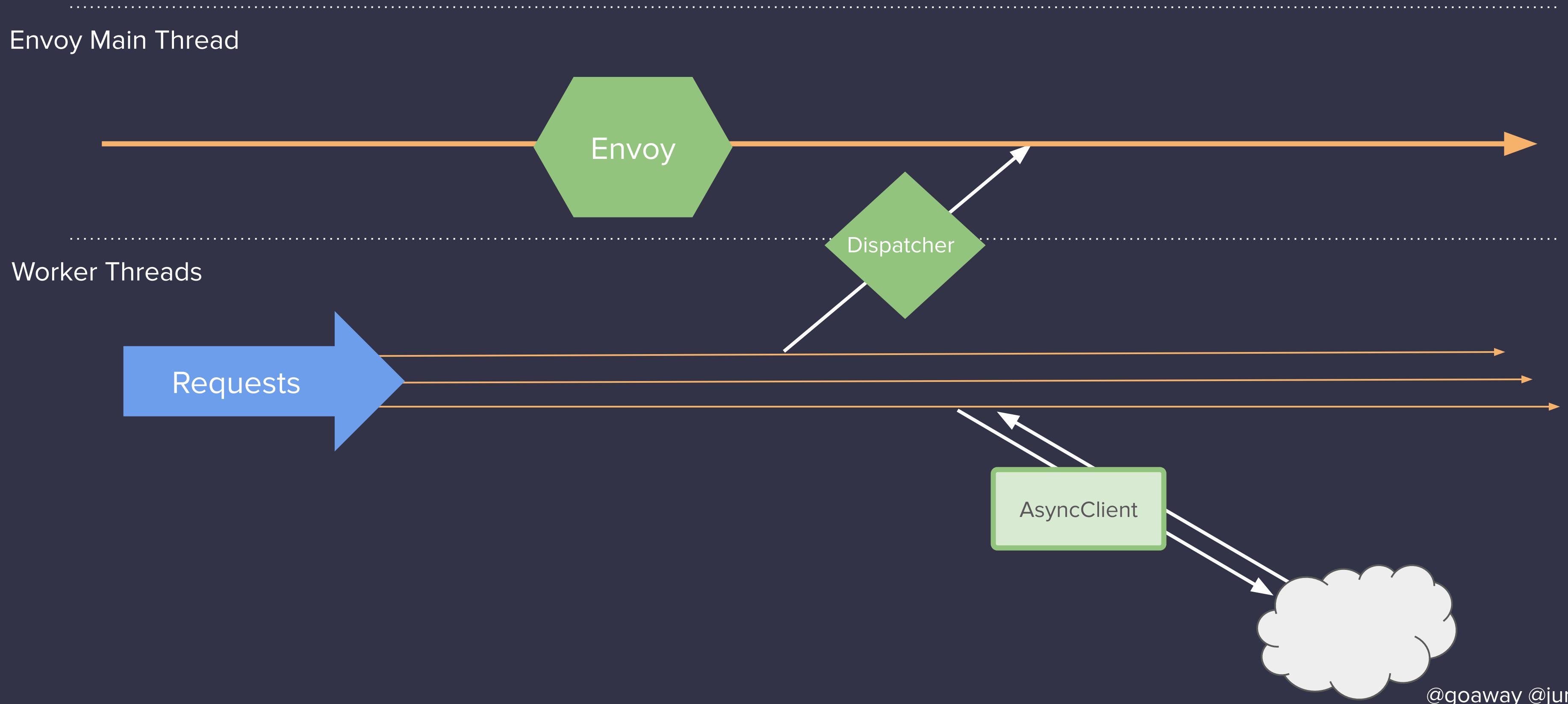
	Platform	Bridge	Native
Application Threads			
Envoy Main Thread			
Callback Threads			

Library Lifecycle - Running Envoy

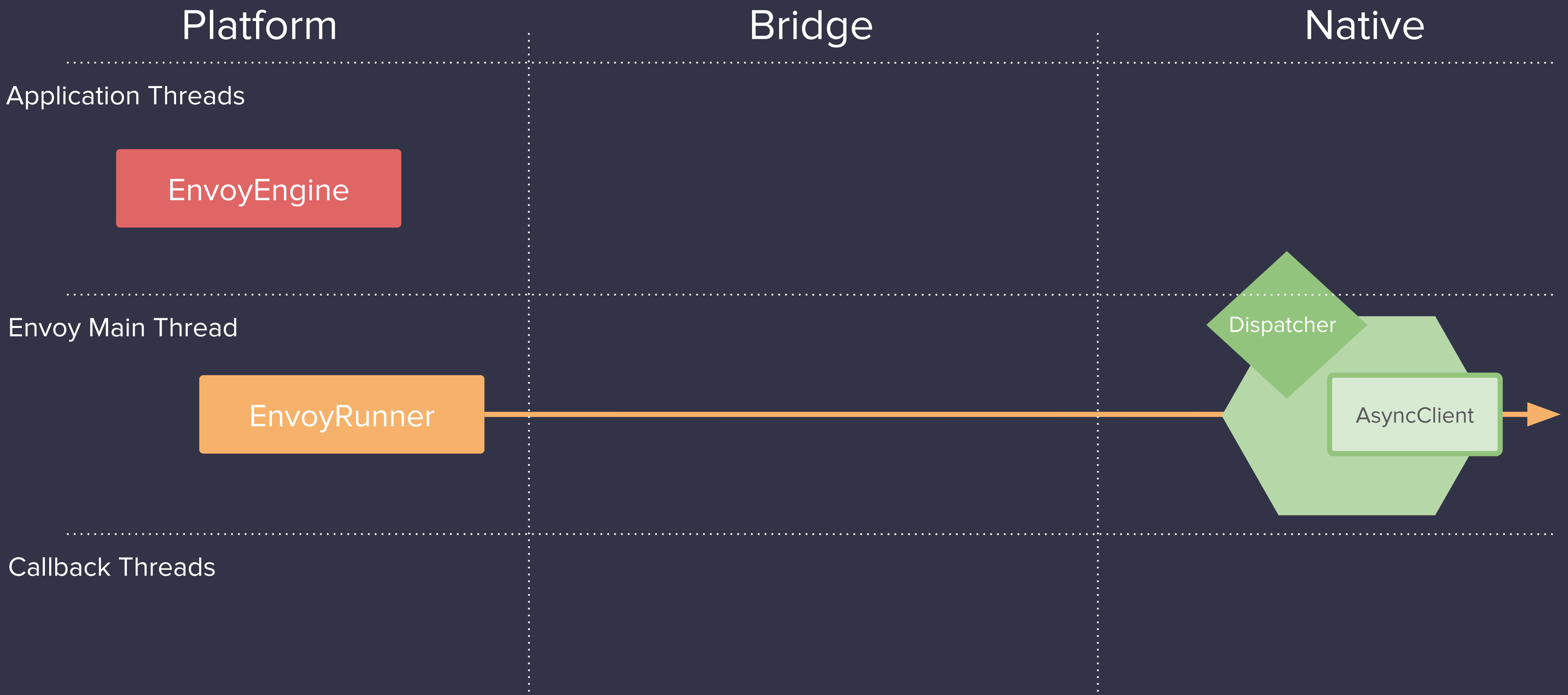


@goaway @junr03

Server Envoy

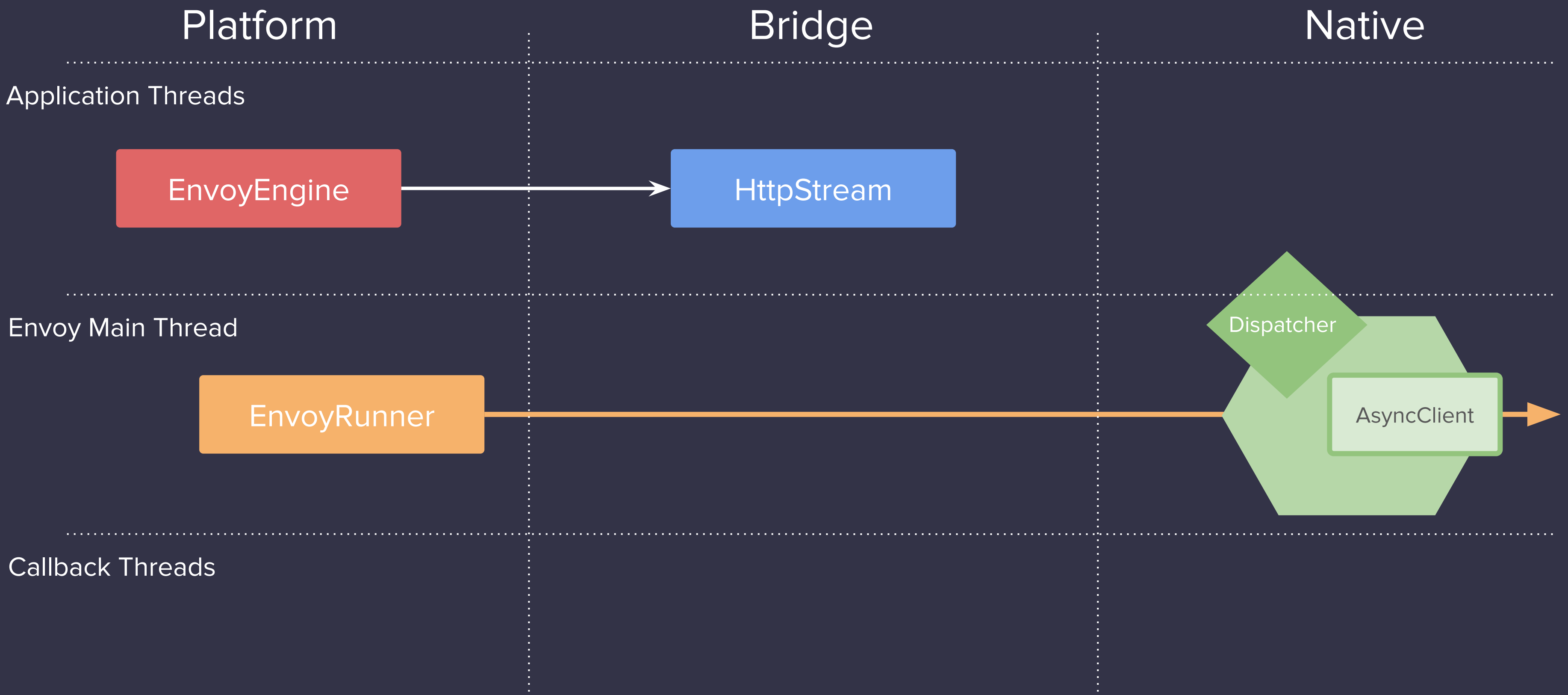


Library Lifecycle - using Envoy Constructs



@goaway @junr03

Library Lifecycle - starting a stream

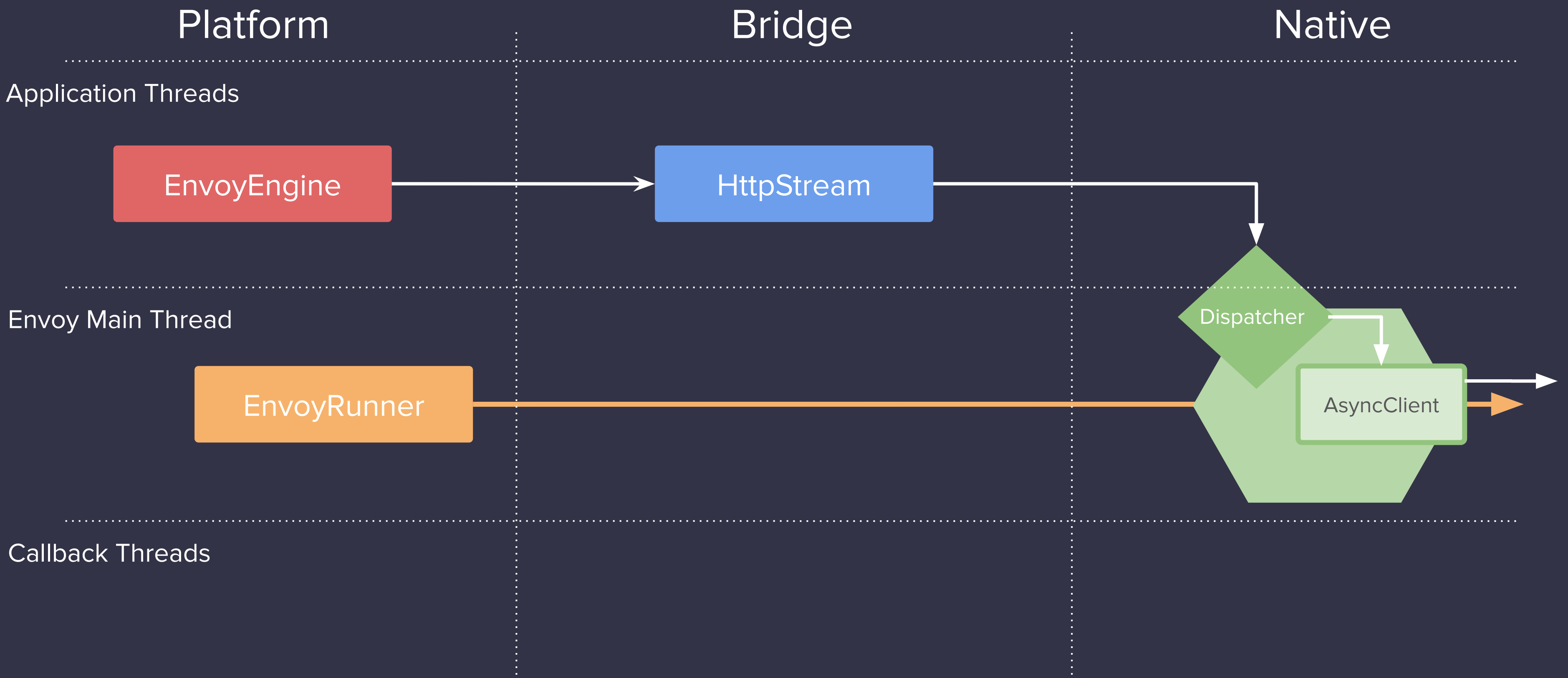


@goaway @junr03

Memory Management

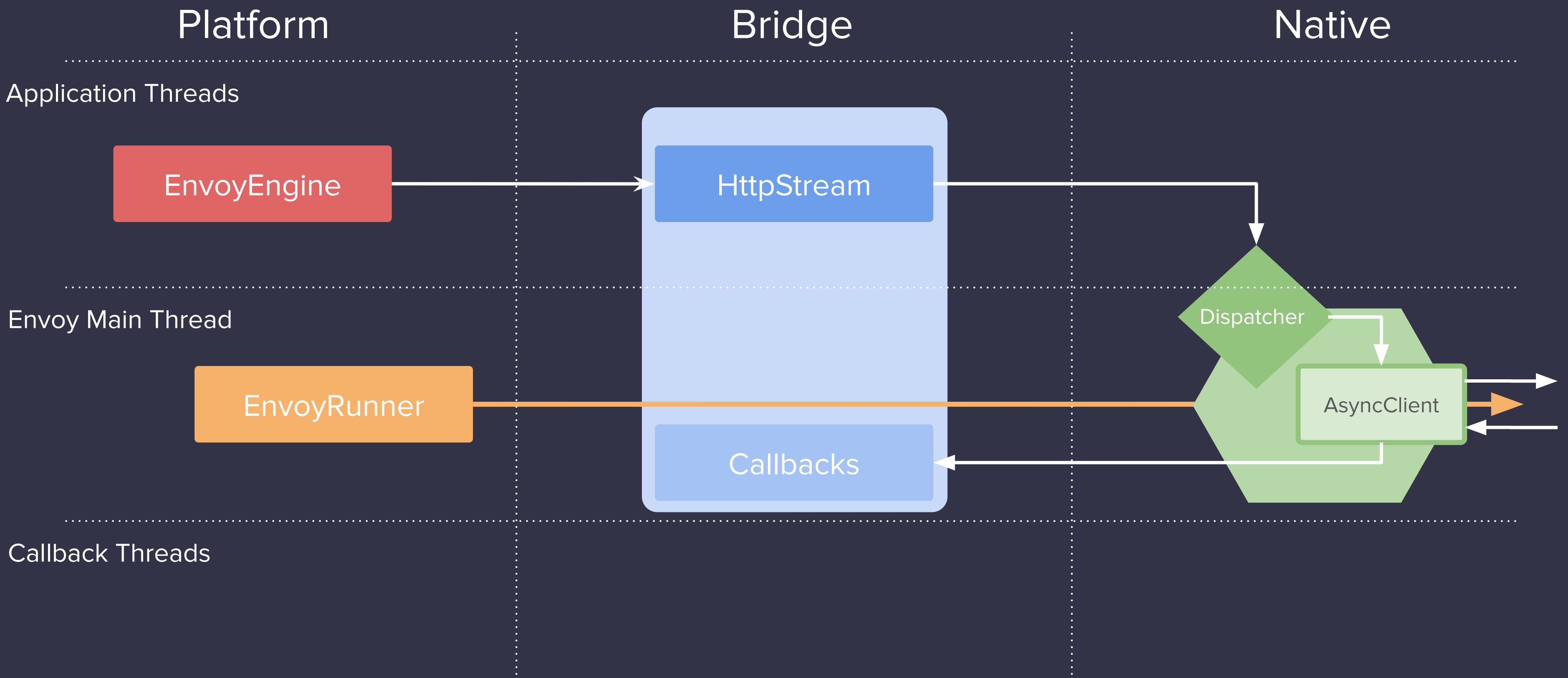
```
/**
 * Holds raw binary data as an array of bytes.
 */
typedef struct {
  size_t length;
  const uint8_t* bytes;
  envoy_release_f release;
  void* context;
} envoy_data;

/**
 * Callback indicating Envoy has drained the associated
buffer.
 */
typedef void (*envoy_release_f)(void* context);
```

Library Lifecycle - dispatching a stream



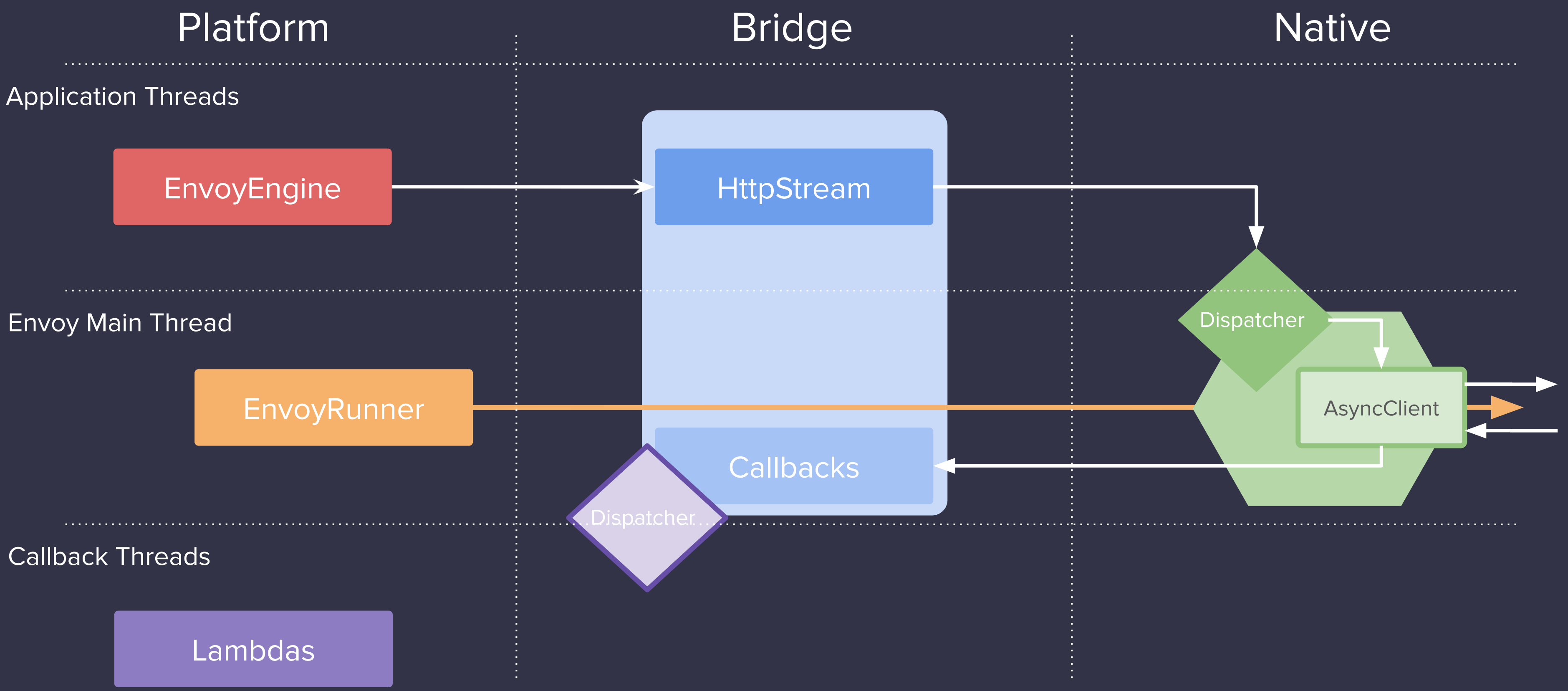
@goaway @junr03

Library Lifecycle - callbacks



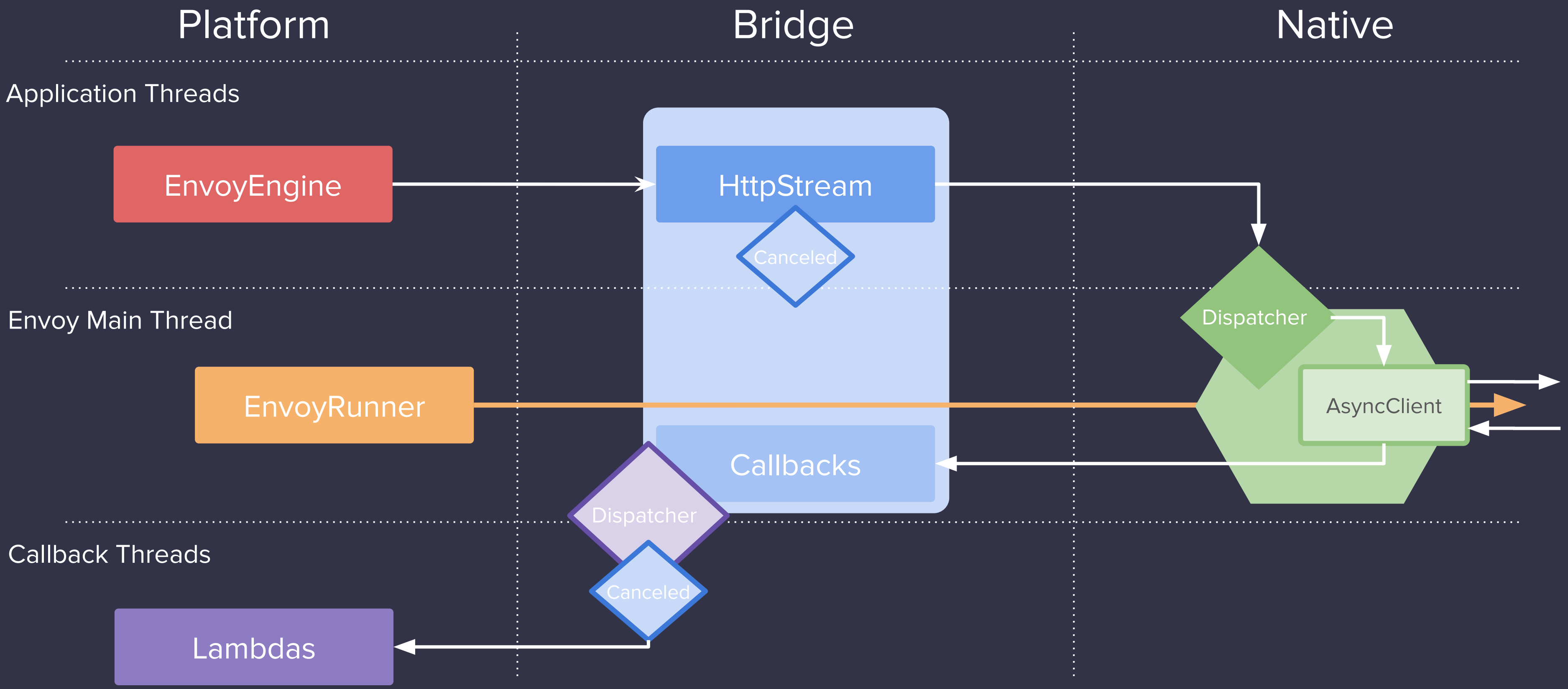
@goaway @junr03

Library Lifecycle - platform callbacks



@goaway @junr03

Library Lifecycle - cancellation

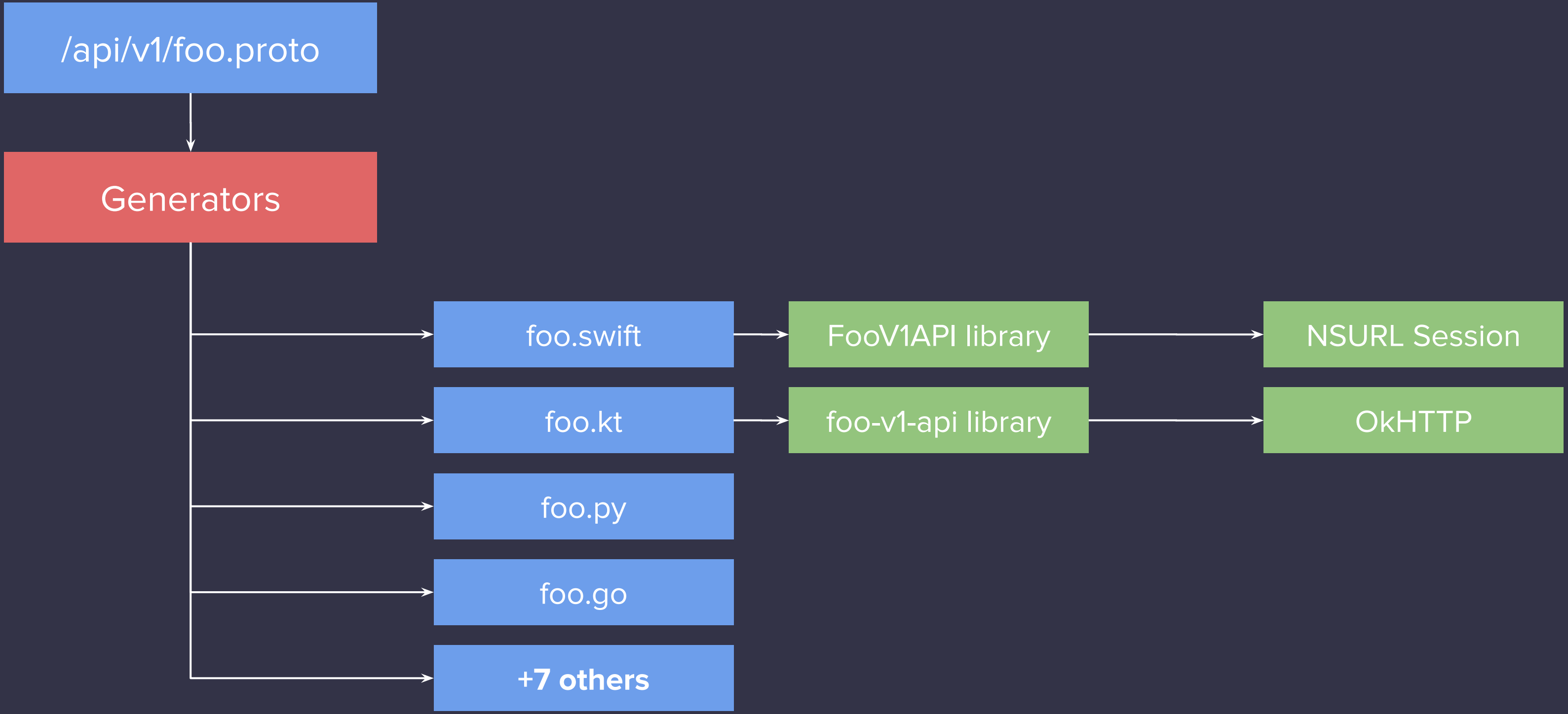


@goaway @junr03

Where we are Now

Alpha App at Lyft!

IDL pipeline



/api/v1/foo.proto

Generators

foo.swift

foo.kt

foo.py

foo.go

+7 others

FooV1API library

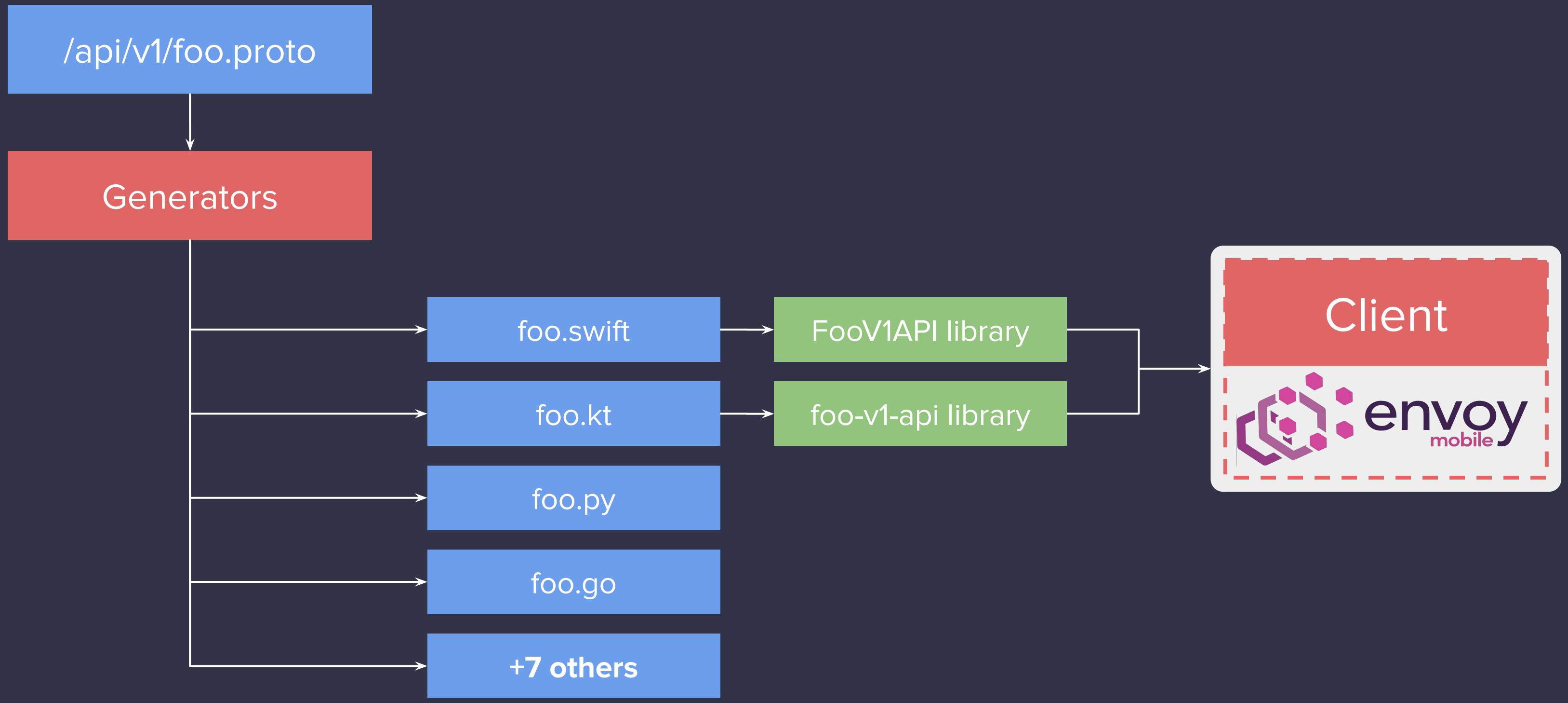
foo-v1-api library

NSURL Session

OkHTTP

@goaway @junr03

IDL pipeline



@goaway @junr03

Build an Engine

```
let envoy = try EnvoyClientBuilder(domain:
"api.envoyproxy.io")
  .addLogLevel(.warn)
  .addStatsFlushSeconds(60)
  .build()
```

Build a Request

```
let request = GRPCRequestBuilder(path:
"/pb.api.v1.Foo/GetBar")
  .addHeader(name: "x-custom-header", value: "foobar")
  .build()
```

Build a Response Handler

```
val handler = GRPCResponseHandler(Executor { })
  .onHeaders { headers, grpcStatus, _ ->
    ...
  }
  .onMessage { messageData ->
    // Deserialize message data here
  }
  ...
```

Make a request

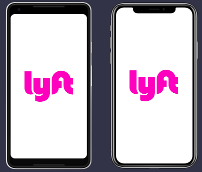
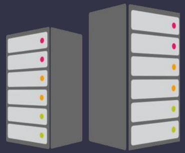
```
val emitter = envoy.send(request, responseHandler)
   .sendData(message)
   .sendData(message)

...
emitter.close()
```

Drop in Replacement

- Expose compatible bindings to classic network libraries: NSURL, OkHTTP
- Model-based APIs as a first class citizen

What are we solving for?

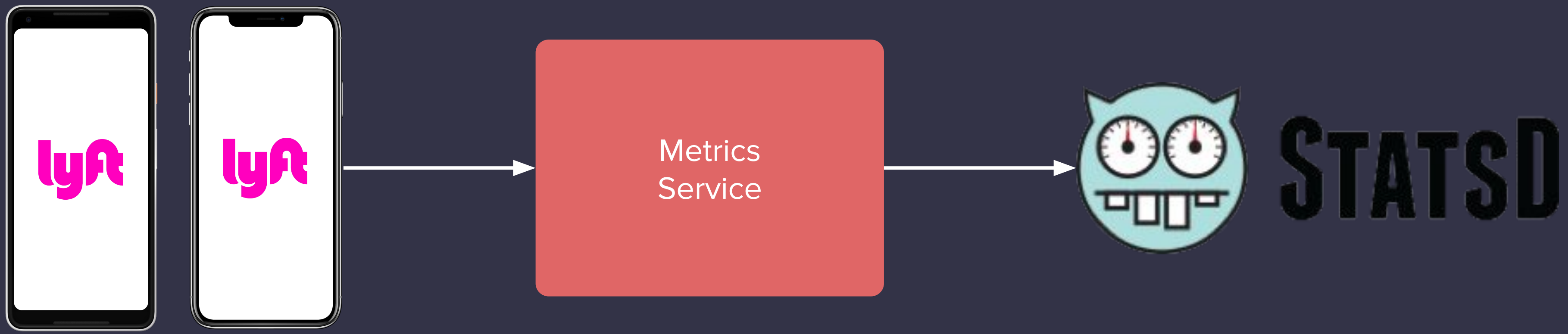
Three 9s at the server-side edge is meaningless if the user of a mobile application is only able to complete the desired product flows a fraction of the time.

		
Performance	✓	✓
Reliability	✓	✓
Extensibility	✓	✓
Observability	✓	✓
Configuration API	✓	✓

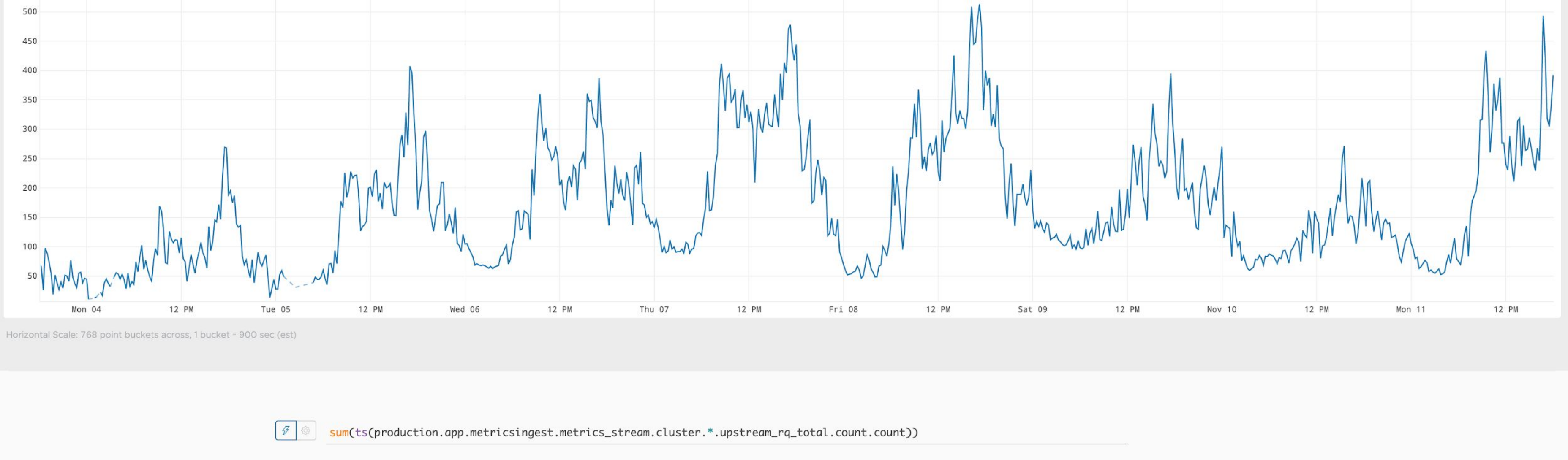
Observability

```
ts(envoy.cluster.*.upstream_rq.count)


ts(envoy_mobile.cluster.api.upstream_rq.count)
```

Time-series Metrics



Dashboards!



Horizontal Scale: 768 point buckets across, 1 bucket ~ 900 sec (est)

```
sum(ts(production.app.metricsingest.metrics_stream.cluster.*.upstream_rq_total.count.count))
```

```
ts(envoy_mobile.cluster.api.upstream_rq.count)
```

@goaway @junr03

Onwards!

Onwards!

- Protocol Experimentation
- API Listener - Filter stack
- Beyond Mobile phones!

Community

This is the beginning, join us!

Michael Schore
@goaway

Jose Nino
@junr03

envoy mobile

envoy-mobile.github.io