A BRIEF HISTORY OF ADOBE I/O

- Adobe I/O APIs
- Adobe I/O Console
- Adobe I/O Runtime
ADOBE I/O RUNTIME – WHY?

➤ Adobe platform for Adobe customers
➤ Custom environments
ADOBE I/O RUNTIME

- Apache OpenWhisk
  - Community
  - API
  - Complete system
- GA June 2019 🎉
- Used by
  - Adobe Customers (third party)
    - Yamaha
    - Microsoft
  - Adobe Teams (first party)
    - Adobe Experience Manager (CMS)
    - Adobe Target
ADOBE I/O RUNTIME USE CASES

➤ Which use cases? All the use cases.

➤ Extremes:
  ➤ 50GB mem + 30min timeout + 100GB disk + 1 concurrency
    ➤ Image processing
  ➤ 128MB mem + 10ms timeout + 500 concurrency
    ➤ User facing browser applications
    ➤ Functions at the edge

➤ Everything else:
  ➤ Microservice replacement
SERVERLESS EVOLVES

➤ Cloudflare
➤ Knative
➤ OpenWhisk
SERVERLESS EVOLVES

➤ Cloudflare
  ➤ V8 isolates - multiple functions, single process
  ➤ Function scaling in process + multi-process
➤ Knative (Serving)
  ➤ General purpose
  ➤ Function per image
  ➤ Function scaling as pods per service via HPA/KPA
➤ OpenWhisk
  ➤ Sequences/Conductor Actions/Compositions
Gateways Evolve

➤ Envoy

➤ Features for serverless

➤ …or things I’d rather not do in OpenWhisk
  ➤ Authorization
  ➤ Rate limiting
UNLOCKING THE RIGHT TOOL FOR THE USE CASE

➤ Use case: latency-sensitive services
  ➤ V8 Isolate

➤ Use case: general purpose microservice scaling
  ➤ Knative

➤ Use case: complex function workflows
  ➤ OpenWhisk

➤ Use case: consistent handling of ingress features
  ➤ Envoy
OPENWHISK BACKGROUND

➤ Components
   ➤ Controller
   ➤ Invoker

➤ Controller
   ➤ Data plane and control plane ingress

➤ Invoker
   ➤ Container management
KEYS

➤ 🐦 Separate planes
➤ 🏨 Delegate ingress features
➤ 🎓 Delegate execution
➤ 🤝 Normalize request/response
SEPARATE THE PLANES

Clients

OW Gateway

OW Controller

CRUD

Invoke

Execution

NEW!

Clients

OW Gateway

OW Controller

CRUD

Invoke

Execution

Control Plane

Data Plane

OW Router

Invoke

NEW!
SEPARATE THE PLANES

➤ Control plane
  ➤ Provided by: OpenWhisk (already exists)

➤ Data plane
  ➤ Provided by: OpenWhisk Router (new)
    ➤ A data plane controller

➤ TODO:
  ➤ A new service to build!
  ➤ Implement execution providers for OpenWhisk, Knative, V8 isolate service
  ➤ Allow resource allocation steps
DELEGATE INGRESS FEATURES

Clients --> OW Gateway

OW Controller
- Authentication
- Rate Limit

Clients --> OW Gateway (Envoy)
- Authentication
- Rate Limit

OW Controller
OW Router
DELEGATE INGRESS FEATURES

➤ Envoy

➤ Authorization
  ➤ External authorization filter
  ➤ A dedicated service
  ➤ Applies to control plane and data plane

➤ Rate limiting
  ➤ Rate limit filter
  ➤ Adaptive concurrency filter (experimental)
  ➤ Applies to control plane and data plane
DELEGATE EXECUTION

Clients

OW Gateway

OW Controller

CRUD

Invoke

Clients

OW Gateway

Control Plane

OW Controller

CRUD

Data Plane

OW Router
DELEGATE EXECUTION EXAMPLE: KNATIVE

➤ Service allocation (stateful use of services)

➤ General purpose - use Stemcells!
  ➤ Before traffic
    ➤ Create general knative services (aka Stemcells)
  ➤ When traffic begins
    ➤ Allocate a specific service to a specific function
    ➤ Use the function
      ➤ Install the function?
  ➤ When traffic increases
    ➤ Scale the service (install the function each time it scales to a new pod)
DELEGATE EXECUTION EXAMPLE: KNATIVE

1. Handle Function1
2. Allocate Service2
3. Route to Service2

0. Launch Stemcells

OpenWhisk Router

Knative

Service1
Service2
Service3

0. Install Function1?
1. Execute Function1!
NORMALIZE REQUEST/RESPONSE

OpenWhisk Router

Transform Request

Knative

V8 Isolates

OpenWhisk

Transform Response

Logging/Metrics
OPENWHISK ROUTER SUMMARY

➤ Not a proxy, exactly

➤ Requirements
  ➤ Provide a data plane API for function execution
  ➤ Resolve functions to execution environments
  ➤ Request/Response normalization - provide consistent metadata per invocation
  ➤ Provide state management cache
    ➤ To optimize resource (re)usage
    ➤ Function resource allocation
    ➤ Function resource GC

➤ Negative requirements
  ➤ Does not use Container primitives, just optional resource management for execution environments
  ➤ Does not use static routes

➤ Potential
  ➤ Streaming
  ➤ Outsource processing (AWS/Azure/Google)
STATE OF OPENWHISK ROUTER

➤ Prototype coming to OpenWhisk core repo

➤ Help us extend OpenWhisk support for more cases!
  ➤ https://openwhisk.apache.org
  ➤ Github https://github.com/apache/openwhisk
  ➤ Slack https://openwhisk-team.slack.com/
  ➤ Mailing list: dev-subscribe@openwhisk.apache.org
THANK YOU!

- Tyson Norris
  - tysonnorris@gmail.com
  - @noizenheimer