NTT Communications
Transform. Transcend.

FaaS is Not Only the Serverless

Stream Processing with Serverless

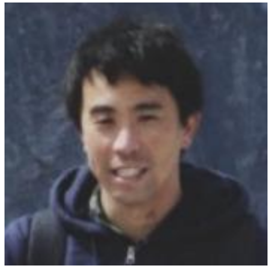
Jun Makishi, Kensaku Komatsu
NTT Communications

KubeCon | CloudNativeCon
Europe 2019

Kensaku Komatsu
Technical Manager, NTT Communications

@komasshu

https://github.com/kensakukomatsu



Jun Makishi
Senior Architect, NTT Communications

@JunMakishi

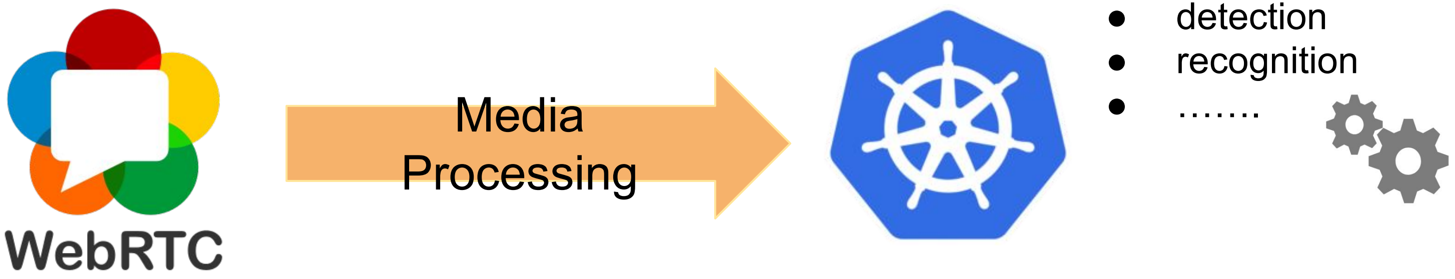
https://github.com/j-maxi

Main topic of this talk

Practical study and our experiment of

“Serverless Real-time Media Processing Platform for WebRTC interface”

built with **Kubernetes** and **open ecosystems**.



- recording
- detection
- recognition
- …….

Today, we'll talking about ...

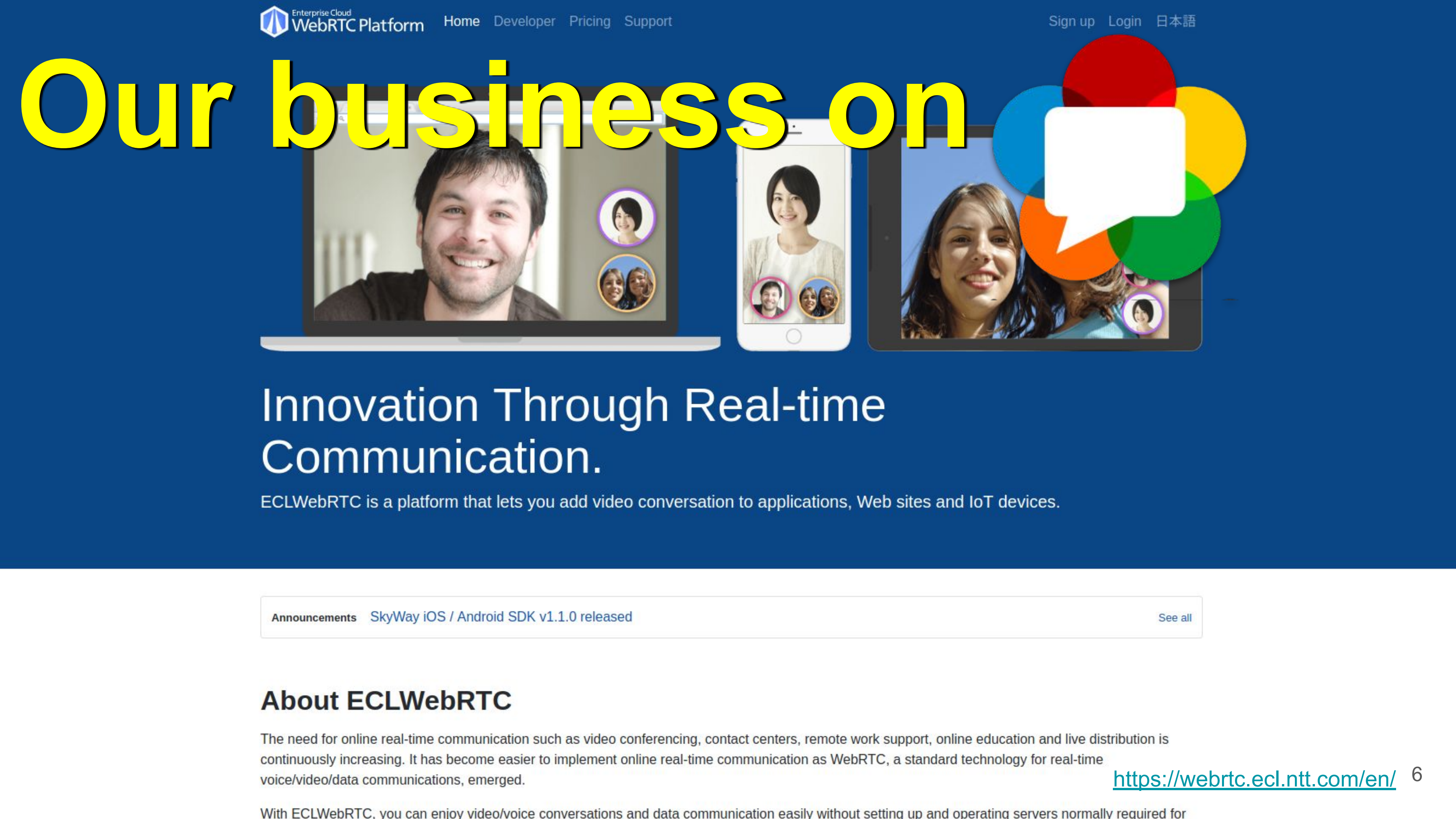
- New type of Serverless - Real-time Media Processing
- Kubernetes
- GRPC
- cloudevents

and more

Motivation

“Serverless Real-time Media Processing Platform for WebRTC interface”

Our business on



Innovation Through Real-time Communication.

ECLWebRTC is a platform that lets you add video conversation to applications, Web sites and IoT devices.

Announcements SkyWay iOS / Android SDK v1.1.0 released See all

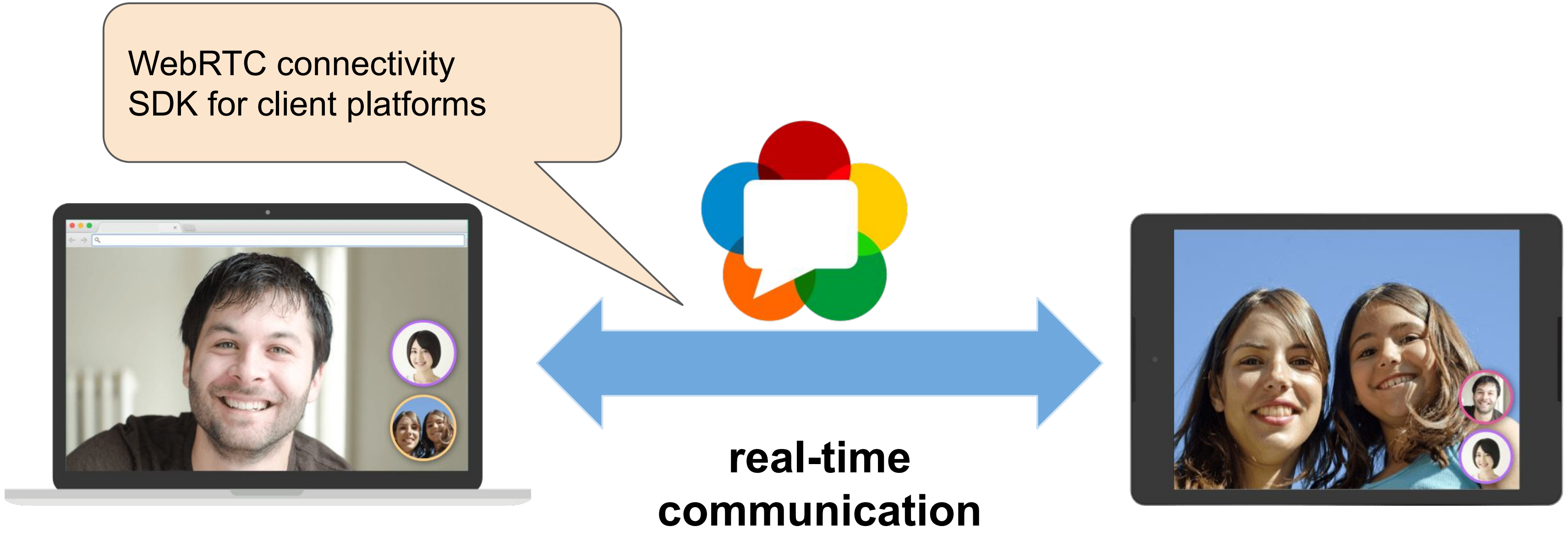
About ECLWebRTC

The need for online real-time communication such as video conferencing, contact centers, remote work support, online education and live distribution is continuously increasing. It has become easier to implement online real-time communication as WebRTC, a standard technology for real-time voice/video/data communications, emerged.

With ECLWebRTC, you can enjoy video/voice conversations and data communication easily without setting up and operating servers normally required for

https://webrtc.ecl.ntt.com/en/

Current model



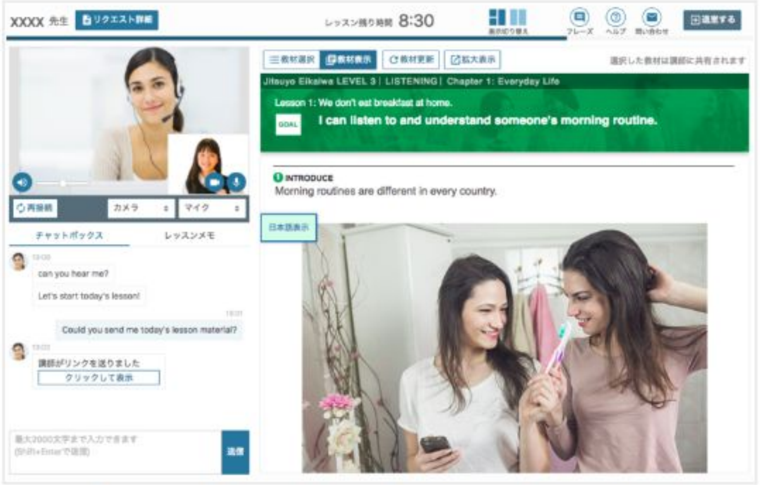
Use cases

- online education
- online healthcare
- video conference
- remote expert
- robot control
- ….

Enterprise Cloud WebRTC Platform

特徴2：一画面で映像やチャット、教材表示が完結するシームレスなレッスン体験

講師の映像やチャット、教材が一画面上に表示されることにより、Webブラウザと通信ソフトを行き来する複雑な操作が必要なく、集中してレッスンを進めることができます。「映像モード」と「教材モード」の表示切替機能により、教材を使ったレッスンや、講師の画面を大きくして口元を見ながら発音の練習も可能です。



Jitsuyo Eikaiwa LEVEL 3 | LISTENING | Chapter 1: Everyday Life

Lesson 1: We don't eat breakfast at home.

GOAL I can listen to and understand someone's morning routine.

INTRODUCE
Morning routines are different in every country.

can you hear me?

Let's start today's lesson!

Could you send me today's lesson material?

講師がリンクを送りました
クリックして表示

最大2000文字まで入力できます
(Shift+Enterで改行)

「教材モード」時のレッスンルームの画面イメージ（PCブラウザ版）

https://www.rarejob.co.jp/news/news_release/service/2018/0326/5126/

Voice from customers

- recording
- voice recognition
- object detection
- live splitting
- AR/MR
- ……

Enterprise Cloud WebRTC Platform

Need cloud computing power and full-managed platform for WebRTC !!

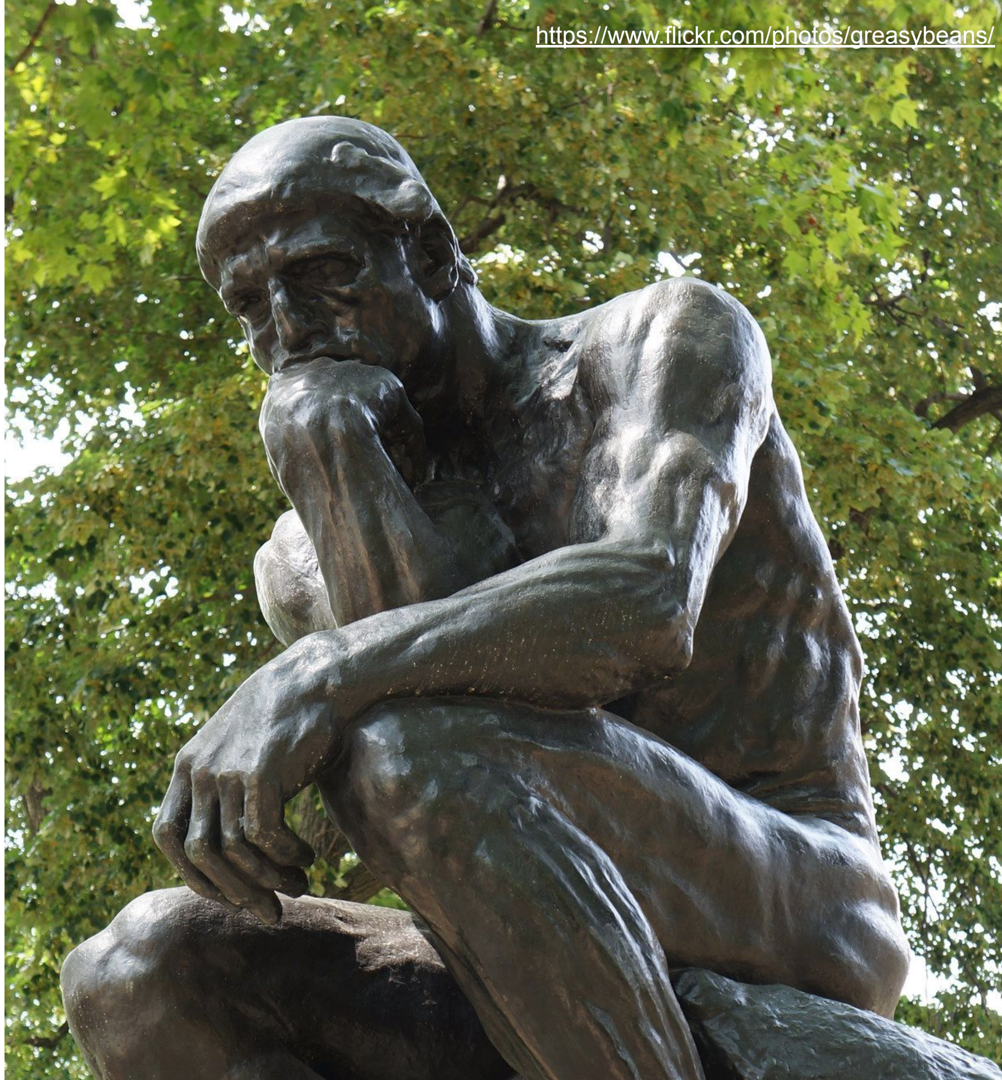
Enterprise Cloud WebRTC Platform

We thought ...

WebRTC IF PaaS ?

Serverless with
Media streaming ?

Long-term session
lifecycle ?



Our Descision

Media Pipeline Factory

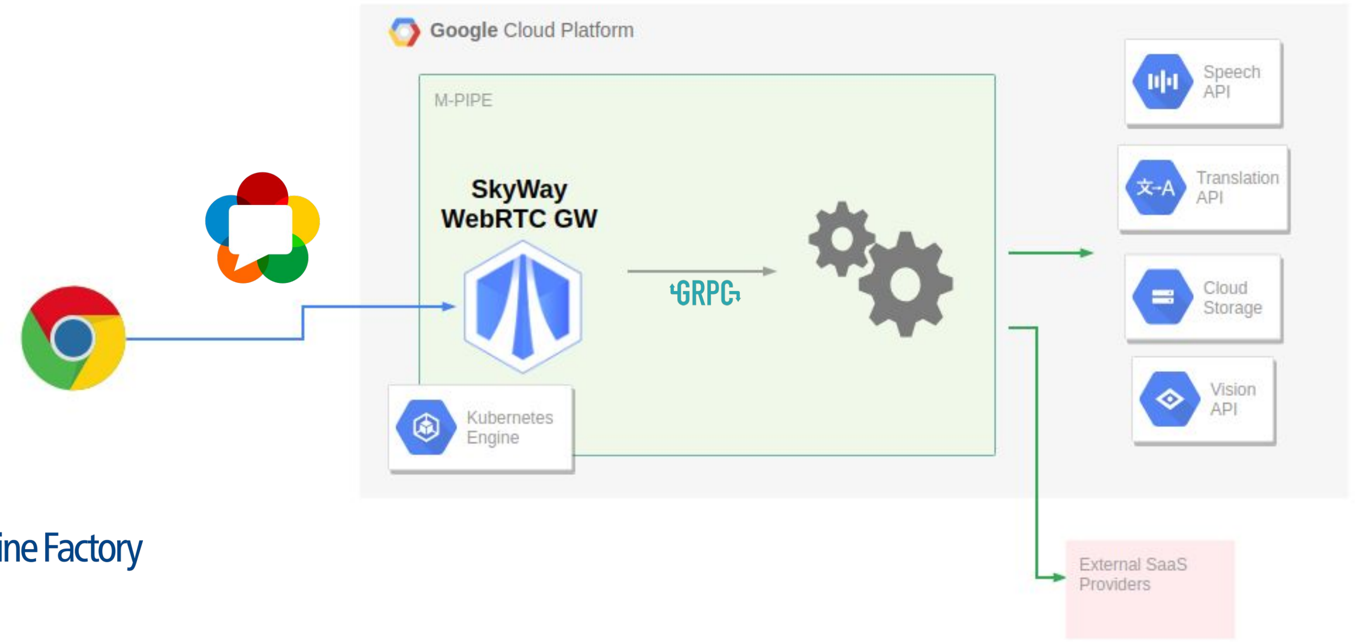
Evolve your business with real-time data enriched with Cloud APIs.

https://webrtc.ecl.ntt.com/m-pipe/en

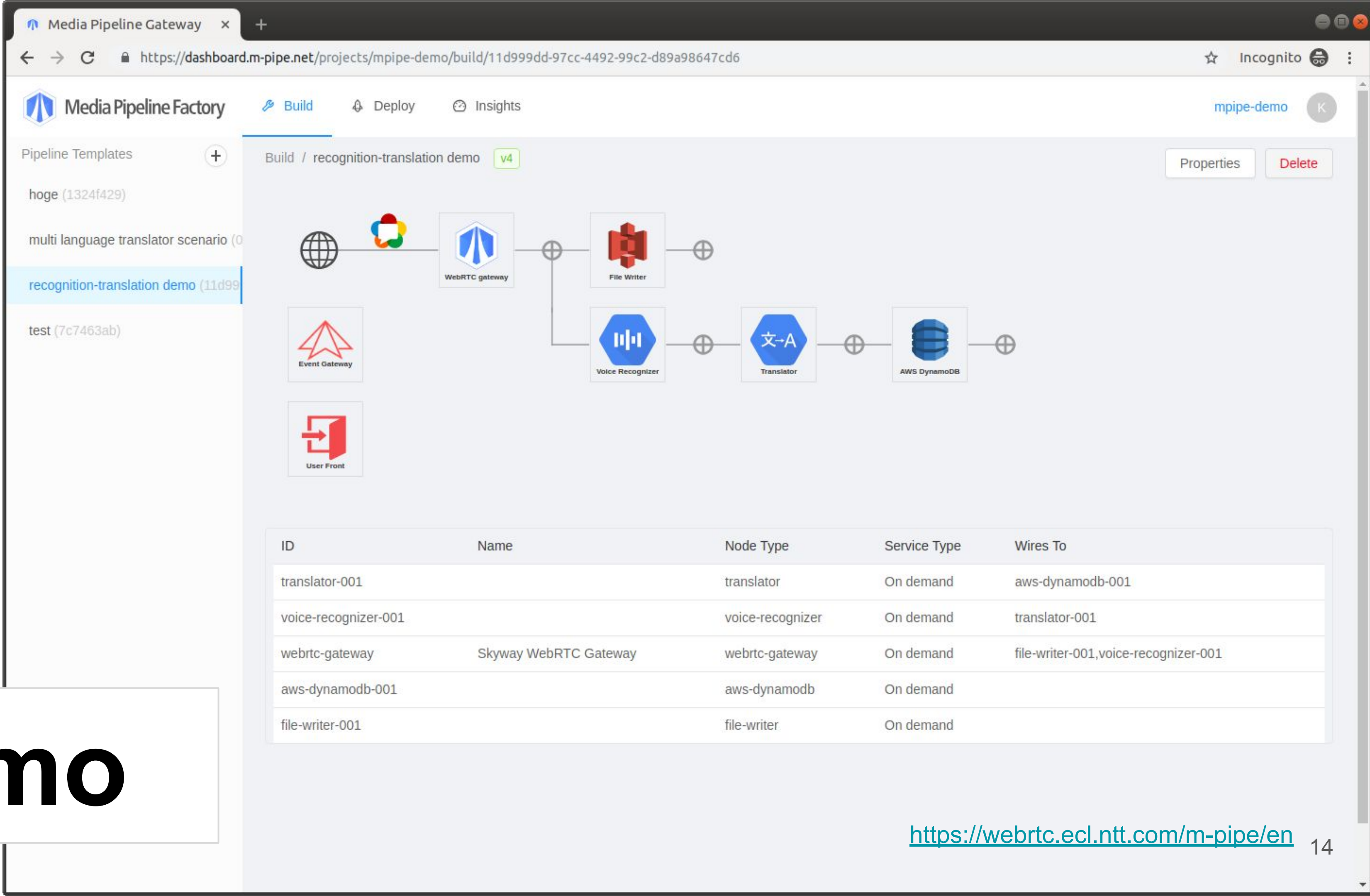
NTT Communications
Transform. Transcend.

Challenge

Built our own "Serverless Real-time media processing platform" using our WebRTC Gateway



Media Pipeline Factory

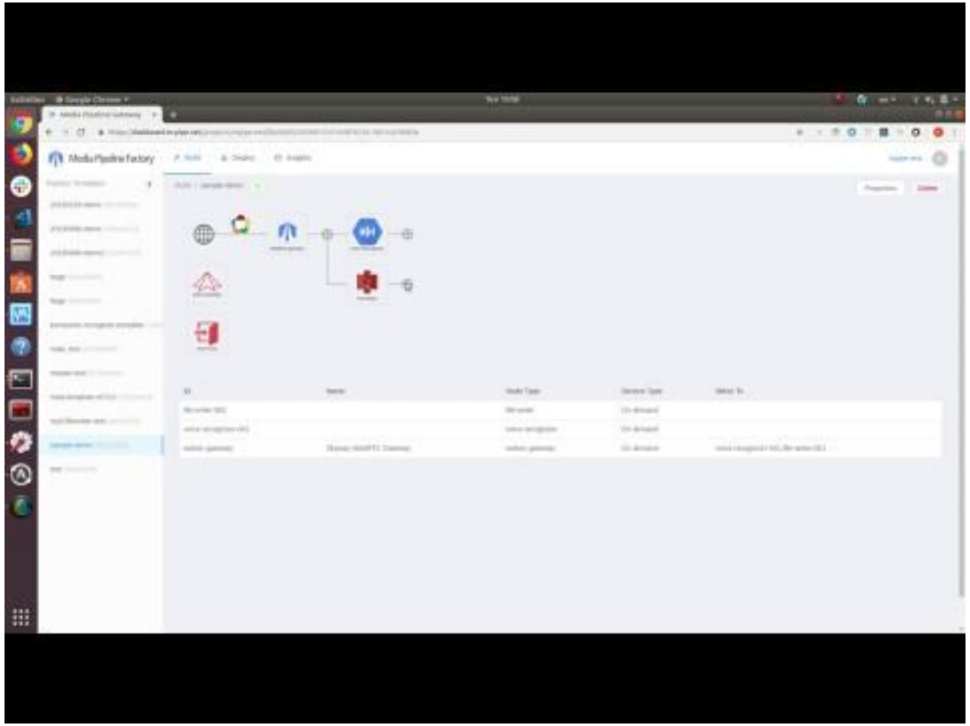


ID	Name	Node Type	Service Type	Wires To
translator-001		translator	On demand	aws-dynamodb-001
voice-recognizer-001		voice-recognizer	On demand	translator-001
webrtc-gateway	Skyway WebRTC Gateway	webrtc-gateway	On demand	file-writer-001,voice-recognizer-001
aws-dynamodb-001		aws-dynamodb	On demand	
file-writer-001		file-writer	On demand	

Demo

https://webrtc.ecl.ntt.com/m-pipe/en

Demo : dashboard



ID	Name	Node Type	Device Type	Wired To
file-writer-001		file-writer	On demand	
voice-recognition-001		voice-recognition	On demand	
webrtc-gateway	Skyway WebRTC Gateway	webrtc-gateway	On demand	voice-recognition-001, file-writer-001

Demo : sample-app

Code snipet

custom function

- Input Stream

```
const { InputStream } = require('skyway-m-pipe-sdk/connector');

const inputStream = new InputStream();

// you need to set hostname and port number of previous component
// please make sure that same token with previous as well
inputStream.start({ host: inHost, port: inPort, token });

inputStream.on( 'data', data => {
  // #=> data.type - arbitrary type data in string format
  //     data.meta - arbitrary meta data in string format
  //     data.payload - arbitrary payload data in binary format
})
```

- Output Stream

```
const { OutputStream } = require('skyway-m-pipe-sdk/connector');

const outputStream = new OutputStream();

outputStream.start({ port: outPort, token })

outputStream.write({
   type: 'test-stream',
   meta: JSON.stringify({ name: test, ts: Date.now() }),
   payload: Buffer.from( 'Hello world' )
})
```

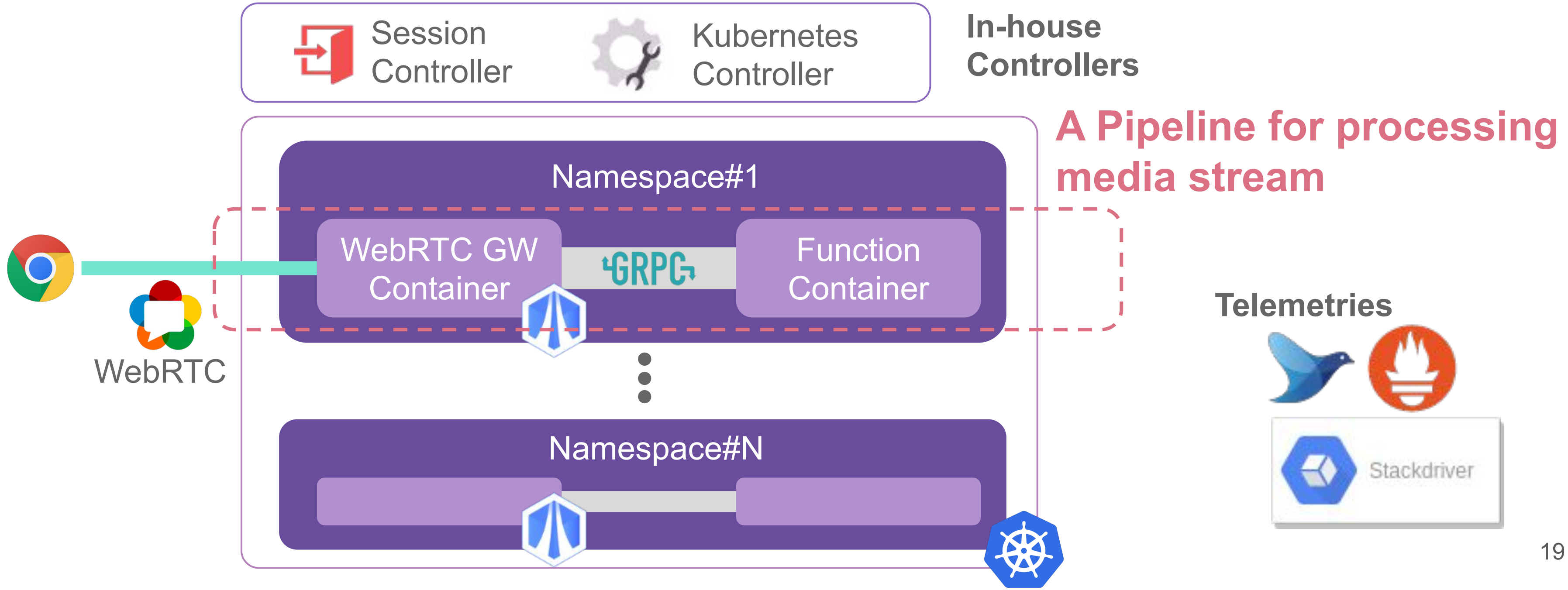
https://github.com/nttcom/skyway-m-pipe-sdk

Our solutions

“Serverless Real-time Media Processing Platform for WebRTC interface”

Platform Overview

- Media stream: latency and jitter sensitive, unbounded ordered data
- Run a chain of containers for media streaming with Kubernetes



Serverless

Run a function per event
(and consume resources only for it)

Event
= Media Streaming

Function
= Real-time Processing

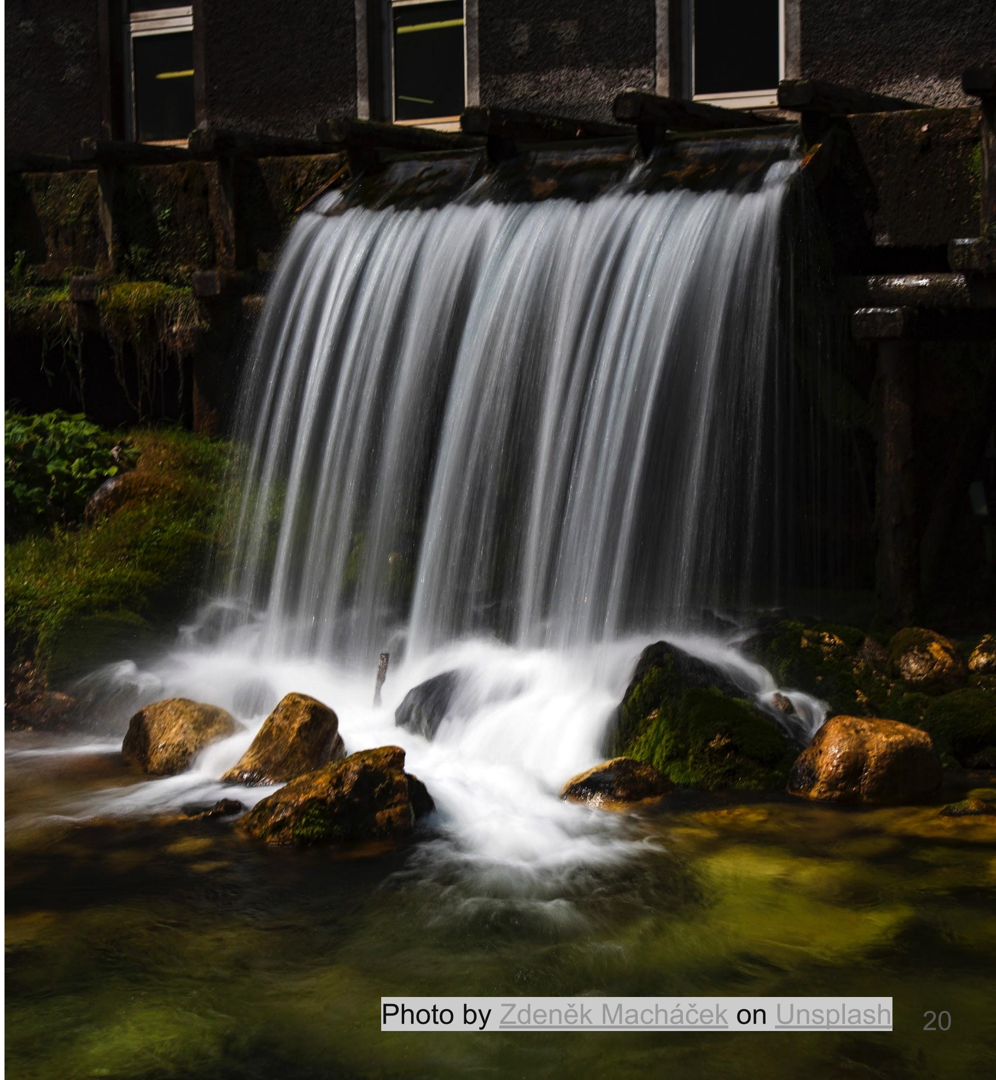
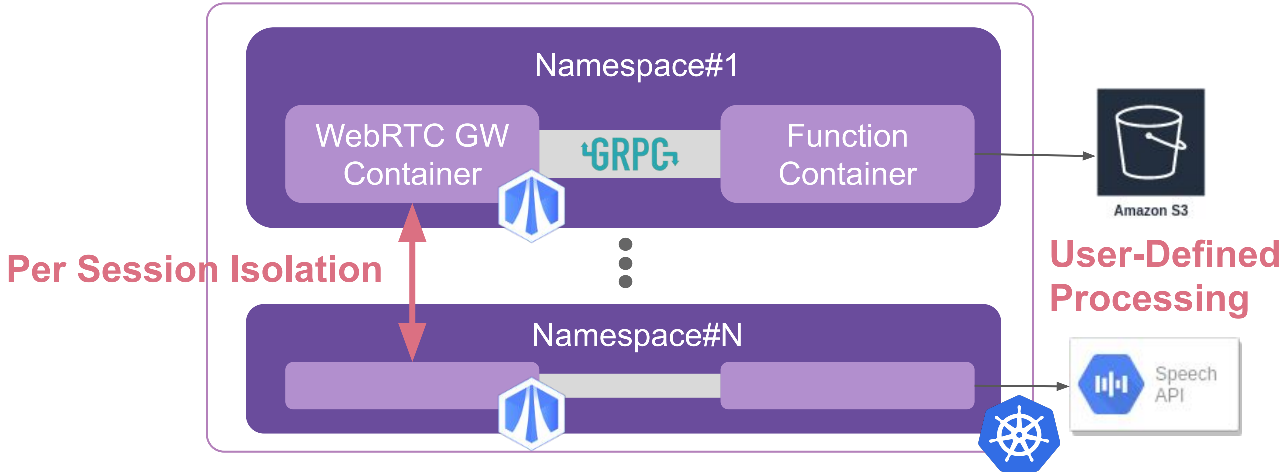


Photo by Zdeněk Macháček on Unsplash

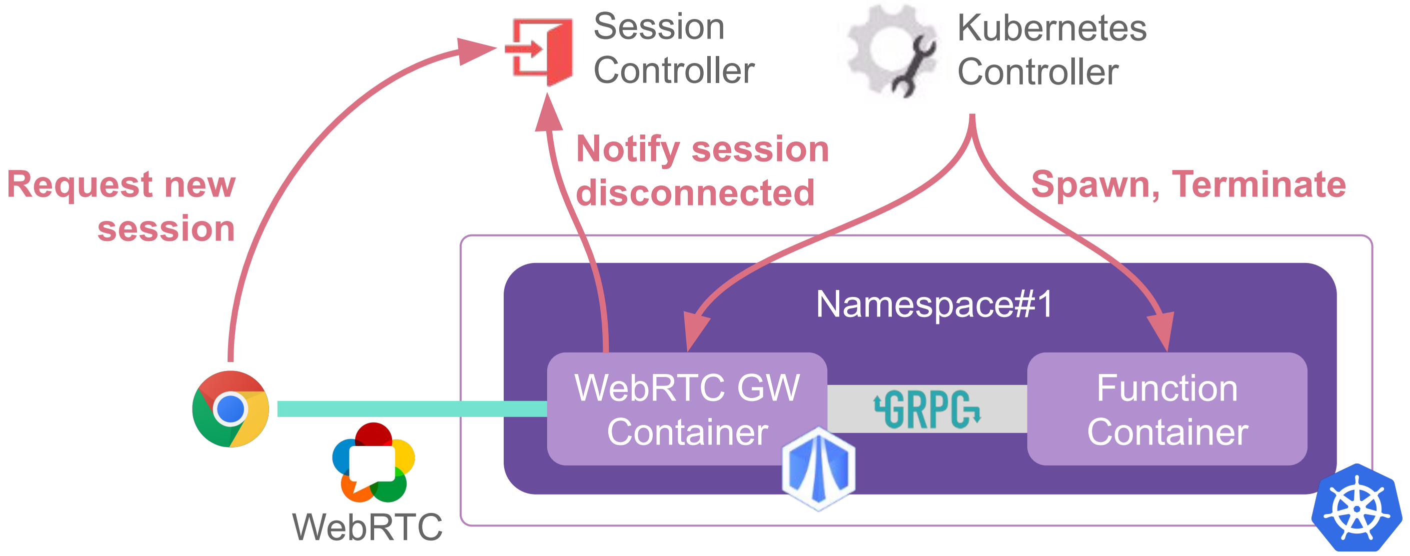
Serverless - Cascaded **Functions** per Session

- Allocate containers **per streaming session**.
- **Cascade** Gateway to user defined functions.
- Isolate sessions by container.
  - Horizontally scalable.
  - Split failure domain.



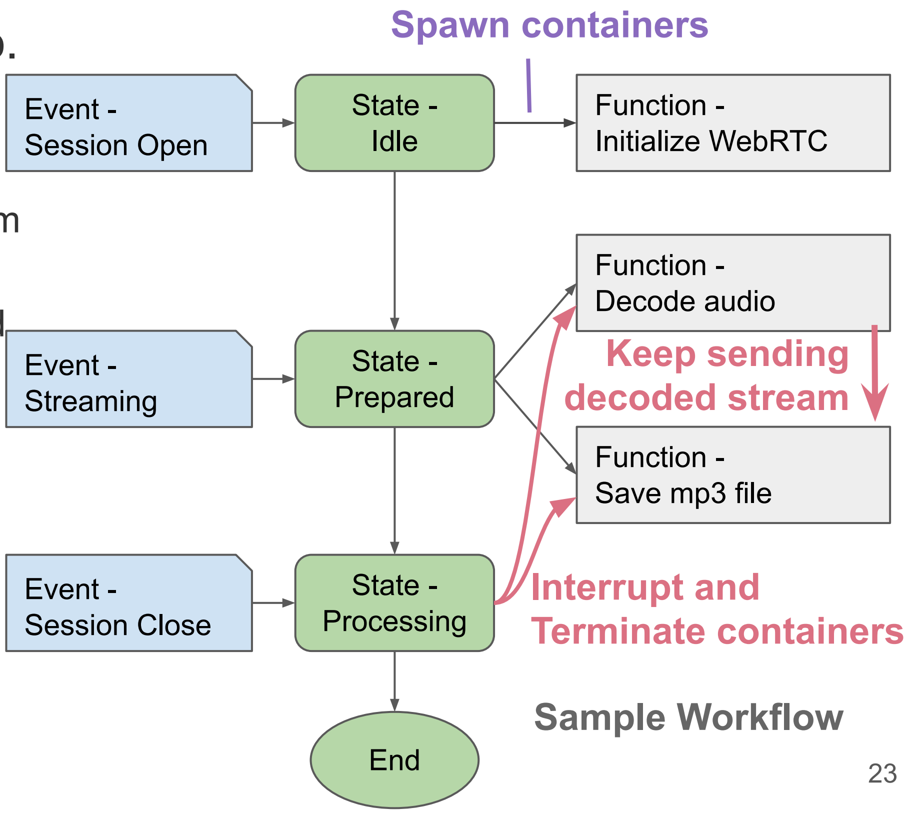
Serverless - Long-lived **Events**

- Run **long-lived containers** to follow streaming lifecycle.
  - Spawn containers for a new session.
  - Terminate containers on the session diconnected.



Serverless - Workflow for Long-lived Functions

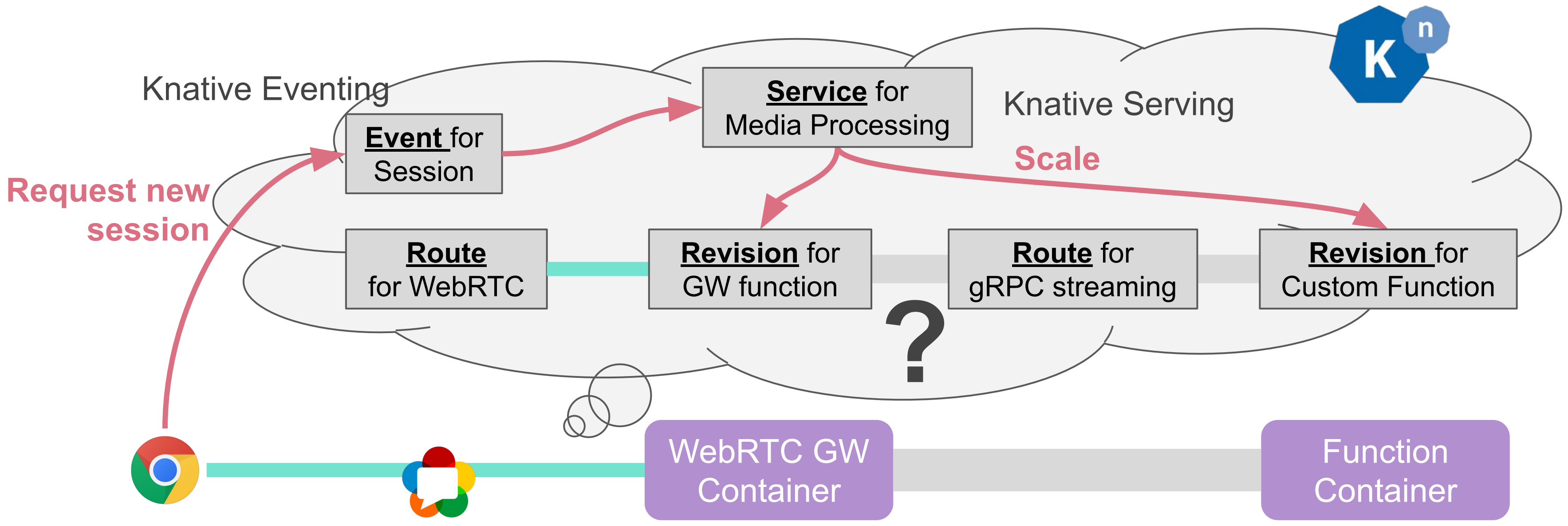
- Workflow for audio recording scenario.
  - Using CNCF Serverless WG spec.
- Long-lived function challenges.
  - **Function relationship** to propagate stream data while running the functions.
  - **Event to trigger "Interrupt"** the long-lived functions.



Sample Workflow

Serverless - Possible integration with Knative

- New event for WebRTC session lifecycle.
- Route function output to another function for streaming.



Internals - Kubernetes

Replicate and distribute Pods.

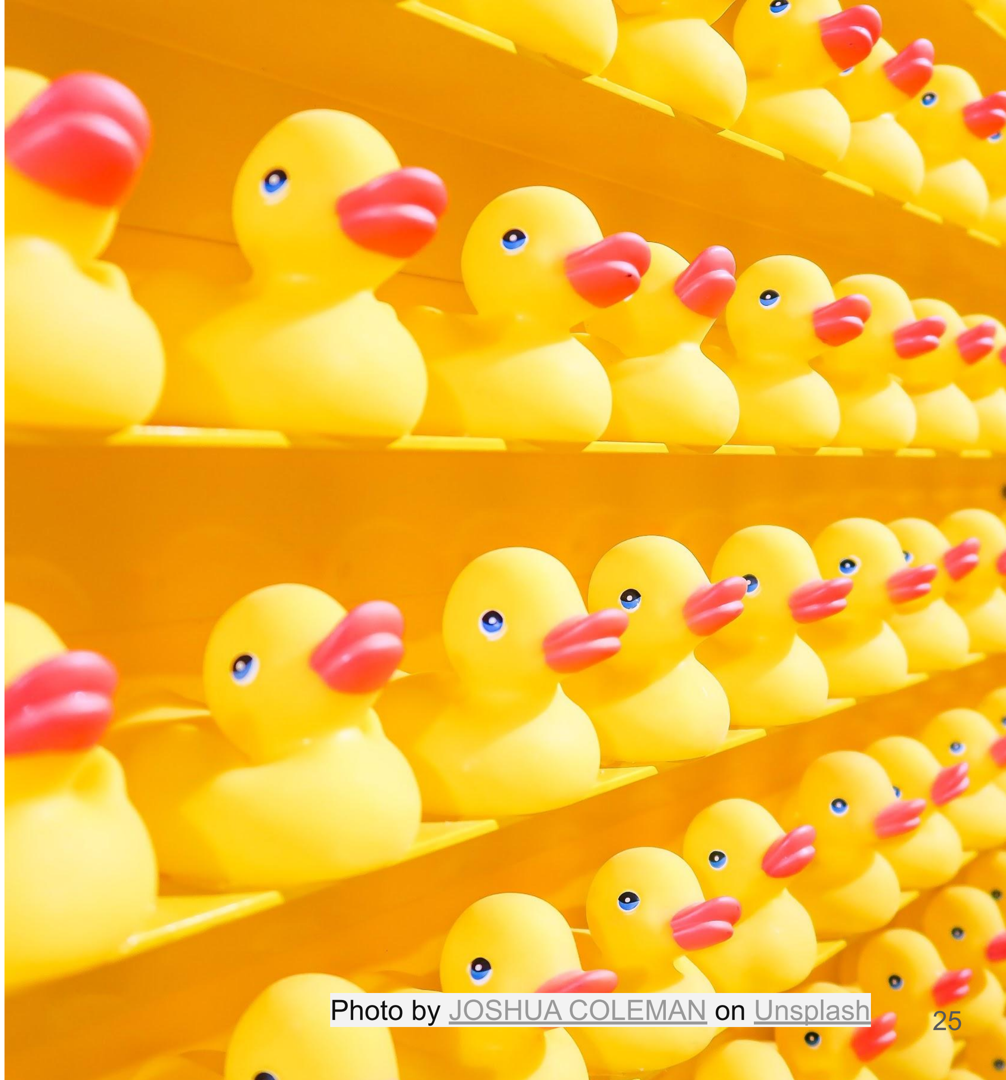
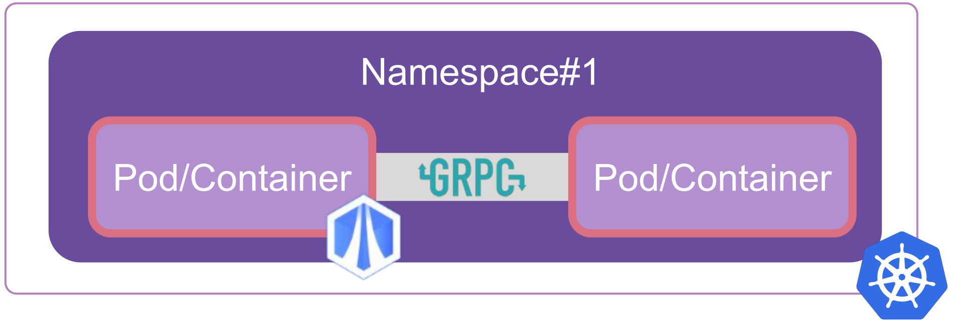
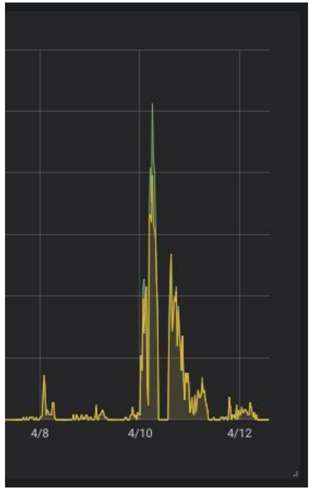


Photo by JOSHUA COLEMAN on Unsplash

Kubernetes - Pipeline with Pod

- Directly call Pod API to execute function.
- Challenges:
    - Sync multiple containers for a session.
    - API performance - Deal with **spike**. Spawn in FIFO.

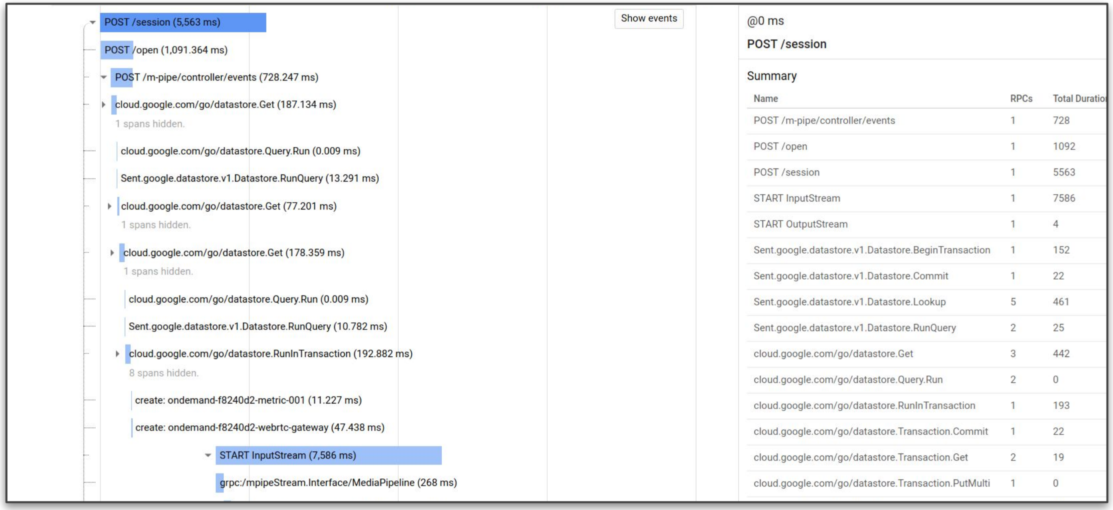




Pod API Spike

Kubernetes - Observability

- Distributed Tracing for container orchestration.
  - **Correlate** each function's start-up latency to an end-end workflow.
- Challenges:
  - Bind trace context to container lifecycle - propagate tracing context to container envvar.



```
POST /session (5,563 ms)
POST /open (1,091.364 ms)
POST /m-pipe/controller/events (728.247 ms)
cloud.google.com/go/datastore.Get (187.134 ms)
1 spans hidden.
cloud.google.com/go/datastore.Query.Run (0.009 ms)
Sent.google.datastore.v1.Datastore.RunQuery (13.291 ms)
cloud.google.com/go/datastore.Get (77.201 ms)
1 spans hidden.
cloud.google.com/go/datastore.Get (178.359 ms)
1 spans hidden.
cloud.google.com/go/datastore.Query.Run (0.009 ms)
Sent.google.datastore.v1.Datastore.RunQuery (10.782 ms)
cloud.google.com/go/datastore.RunInTransaction (192.882 ms)
8 spans hidden.
create: ondemand-f8240d2-metric-001 (11.227 ms)
create: ondemand-f8240d2-webrtc-gateway (47.438 ms)
START InputStream (7,586 ms)
grpc:/mpipeStream.Interface/MediaPipeline (268 ms)
```

Show events

@0 ms

POST /session

Summary

Name	RPCs	Total Duratio
POST /m-pipe/controller/events	1	728
POST /open	1	1092
POST /session	1	5563
START InputStream	1	7586
START OutputStream	1	4
Sent.google.datastore.v1.Datastore.BeginTransaction	1	152
Sent.google.datastore.v1.Datastore.Commit	1	22
Sent.google.datastore.v1.Datastore.Lookup	5	461
Sent.google.datastore.v1.Datastore.RunQuery	2	25
cloud.google.com/go/datastore.Get	3	442
cloud.google.com/go/datastore.Query.Run	2	0
cloud.google.com/go/datastore.RunInTransaction	1	193
cloud.google.com/go/datastore.Transaction.Commit	1	22
cloud.google.com/go/datastore.Transaction.Get	2	19
cloud.google.com/go/datastore.Transaction.PutMulti	1	0

Kubernetes - Multi Tenancy

- Isolate session and pipeline per customer.
  - Special inter-function validation mechanism.
- Challenges:
  - Credential Management, Security

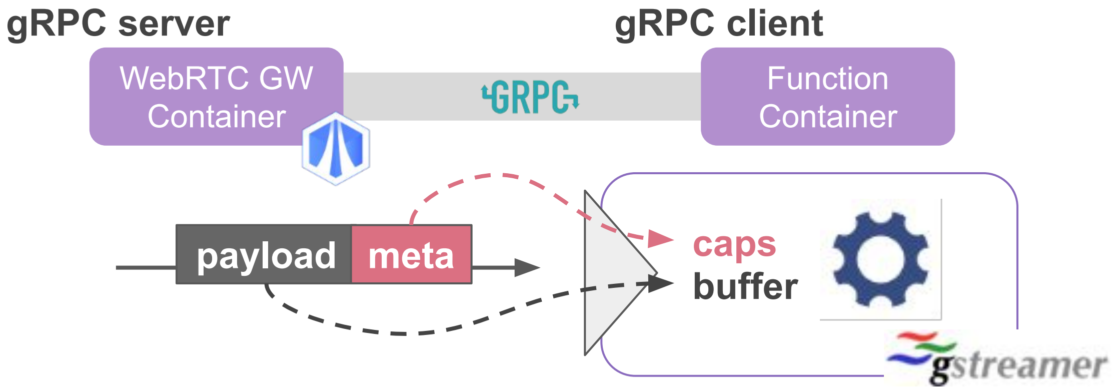
Internals - Open Ecosystem

Integrate building blocks of open ecosystem.

Photo by Ryan Fields on Unsplash

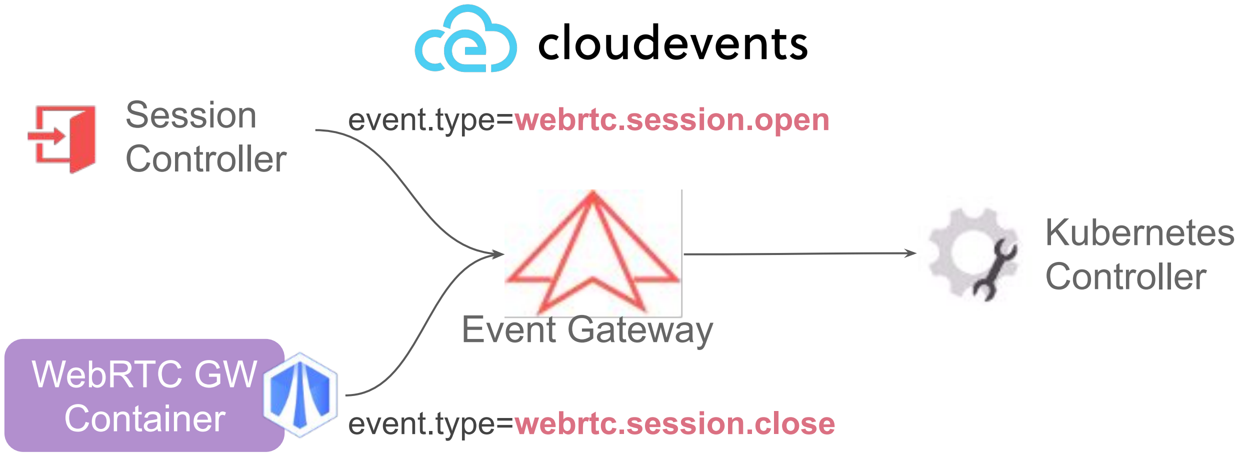
gRPC

- Server streaming RPC
- .proto message for media metadata and payload.
  - Inter-function operability, Gstreamer ready.
- Challenges:
  - Custom transport like UDP.



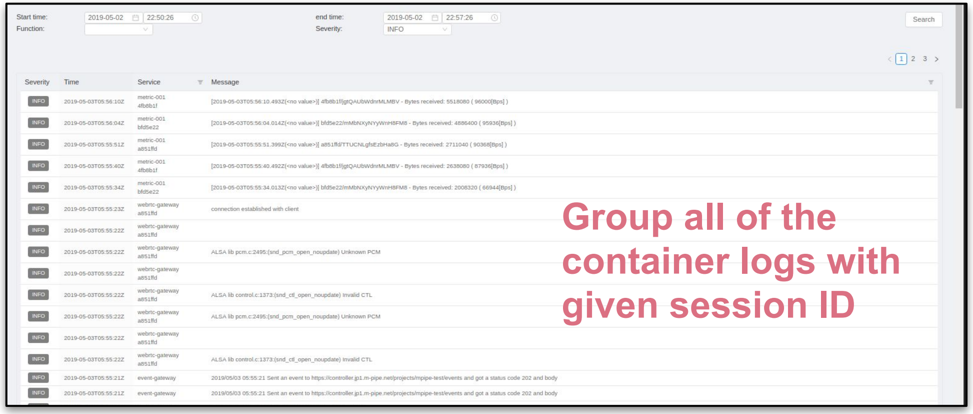
Cloudevents

- Defined streaming session events with Cloudevents v0.1.
  - **Loosely coupled** controllers and components.
- Challenges:
  - Event Tracing



Telemetry

- Group logs, metrics, and tracing with session ID.
  - Correlation based on Kubernetes metadata.
- Challenges:
  - Custom metadata correlation - Metadata Agent.
  - **Actionable metrics** - Drill down. Jump to other metrics with given metadata.



Start time:	2019-05-02 22:50:26	end time:	2019-05-02 22:57:26
Function:		Severity:	INFO

Severity	Time	Service	Message
INFO	2019-05-03T05:56:10Z	metric-001 4fb8b1f	[2019-05-03T05:56:10.493Z(<no value>)] 4fb8b1f/jgtQAUbWdnrMLMBV - Bytes received: 5518080 (96000[Bps])
INFO	2019-05-03T05:56:04Z	metric-001 bfd5e22	[2019-05-03T05:56:04.014Z(<no value>)] bfd5e22/mMbNXyNYyWnH8FM8 - Bytes received: 4886400 (95936[Bps])
INFO	2019-05-03T05:55:51Z	metric-001 a851ffd	[2019-05-03T05:55:51.399Z(<no value>)] a851ffd/TTUCNLgfsEzbHa8G - Bytes received: 2711040 (90368[Bps])
INFO	2019-05-03T05:55:40Z	metric-001 4fb8b1f	[2019-05-03T05:55:40.492Z(<no value>)] 4fb8b1f/jgtQAUbWdnrMLMBV - Bytes received: 2638080 (87936[Bps])
INFO	2019-05-03T05:55:34Z	metric-001 bfd5e22	[2019-05-03T05:55:34.013Z(<no value>)] bfd5e22/mMbNXyNYyWnH8FM8 - Bytes received: 2008320 (66944[Bps])
INFO	2019-05-03T05:55:23Z	webrtc-gateway a851ffd	connection established with client
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	ALSA lib pcm.c:2495:(snd_pcm_open_noupdate) Unknown PCM
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	ALSA lib control.c:1373:(snd_ctl_open_noupdate) Invalid CTL
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	ALSA lib pcm.c:2495:(snd_pcm_open_noupdate) Unknown PCM
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	
INFO	2019-05-03T05:55:22Z	webrtc-gateway a851ffd	ALSA lib control.c:1373:(snd_ctl_open_noupdate) Invalid CTL
INFO	2019-05-03T05:55:21Z	event-gateway	2019/05/03 05:55:21 Sent an event to https://controller.jp1.m-pipe.net/projects/mpipe-test/events and got a status code 202 and body
INFO	2019-05-03T05:55:21Z	event-gateway	2019/05/03 05:55:21 Sent an event to https://controller.jp1.m-pipe.net/projects/mpipe-test/events and got a status code 202 and body

Group all of the container logs with given session ID

Recap

Motivation:

Server-side (Cloud) real-time media processing for **WebRTC**

Solution:

Serverless Real-time Media Processing Platform

➔ Enpowered by Kubernetes, and other open ecosystem

Challenges:

- Integration for **the new serverless workflow and lifecycle**

Thank you

Media Pipeline Factory

Evolve your business with real-time data enriched with Cloud APIs.

https://webrtc.ecl.ntt.com/m-pipe/en

SDK of Media Pipeline Factory : https://github.com/nttcom/skyway-m-pipe-sdk
Sample codes of function container : https://github.com/nttcom/skyway-m-pipe-components
SkyWay WebRTC Gateway : https://github.com/skyway/skyway-webrtc-gateway