KubeCon
CloudNativeCon
North America 2019

# Serverless WG
# Tell me where it hurts

*Mark Peek, VMware*
*Ruben Romero, RedHat & Mauricio Salatino, Camunda*
*Doug Davis, IBM*

# Agenda



- Overview of Serverless WG
  - Summary and status of the Serverless WG
  - Developments in Serverless Technology
  - CloudEvents
- Workflow Specification
- BoF Community Discussion
  - Community adoption and concerns
  - Where should Serverless WG go next?

# Let's look back to mid-2017



## Serverless: Why a CNCF WG

- Serverless is a design pattern being used in cloud native architectures
- Numerous open source offerings
  - https://github.com/JustServerless/awesome-serverless
- Early stages
  - Don't want to constrain innovation
  - But at the same time want to ensure interoperability for users
- Need for:
  - Function portability
  - Service interoperability (for example, common context and eventing)
  - ??? (perhaps more but again, early stages)

# And, what was the CNCF WG proposal?



Create CNCF WG to discuss whether there are meaningful documents to be driven at this time for serverless.

From CNCF Serverless/FaaS Notes

**What should a CNCF WG do in this space?**

- Provide clarity for ourselves and consumers w.r.t. what this topic is all about:
  - Define common terminology
  - Define the scope of the space as it exists today - over time this may change
  - Identify common use cases and patterns between existing implementations
- Identify potential next steps for the community (may or may not happen within CNCF):
  - Identifying areas where we'd like to see some harmonization or interop work
- Create a white/position-paper on Serverless in relation to Cloud Native.

# How did we do?



- Formed the serverless wg in mid-2017 and kicked off weekly meetings
- Initial deliverables – released 2018
  - Whitepaper
    - Overview of technology
    - State of ecosystem
    - Recommendations for possible CNCF next steps
  - Landscape
- Initiative: CloudEvents
  - CloudEvents project started Dec 2017
  - CNCF Sandbox project approved May 2018
  - Version 1.0 with SDKs and Incubator status approved in Oct. 2019
- Initiative: Workflow
  - Update right after this…

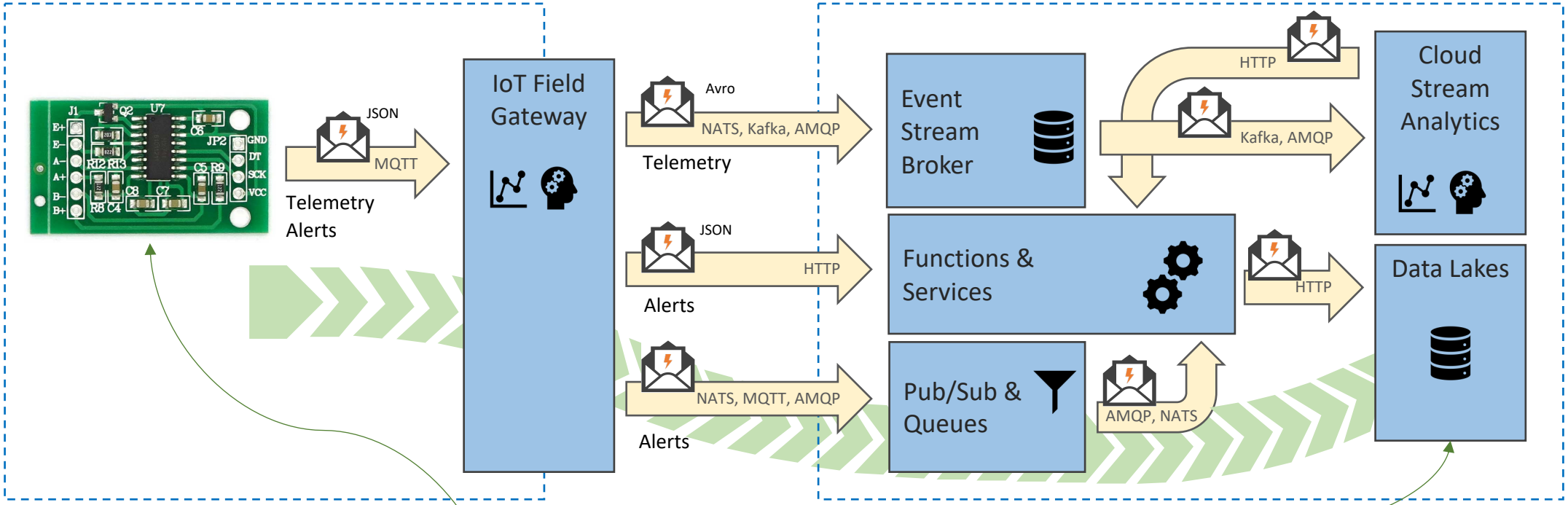
# Developments in Serverless



- Industry shift & acceptance of loosely coupled, event driven architecture
  - Driven by public cloud FaaS – AWS, Azure, GCP, IBM, Oracle
- Serverless/FaaS Projects
  - Knative Serving
  - OpenFaaS
  - OpenWhisk
  - Riff
  - And others…nuclio, fission, fn
- Event Driven Projects
  - KEDA - Kubernetes-based Event Driven Autoscaling
  - Dapr – Distributed Application Runtime
  - Knative Eventing
- Divide between FaaS/Serverless and CaaS, PaaS is narrowing

# Why CloudEvents?






- Event data is often routed via multiple hops and often using different protocols
- How is what gets sent here easily routed to and stored here in hybrid edge/cloud and multi-cloud systems?

# CNCF CloudEvents



- Event Protocol Suite developed in CNCF Serverless WG
  - Common metadata attributes for events
  - Flexibility to innovate on event semantics
  - Simple abstract type system mappable to different encodings
- Transport options
  - HTTP(S) 1.1 Webhooks, also HTTP/2
  - MQTT 3.1.1 and 5.0
  - AMQP 1.0
- Encoding options
  - JSON (required for all implementations)
  - Extensible for binary encodings: Avro, AMQP, etc.

cloudevents

# What's next?



- Proposed work streams
  - **Packaging contract**
  - **Function signatures**
  - **Subscription API**
  - **Discovery**
  - Observability, tracing
  - Schema registry (CloudEvents)
  - E2E Security
  - Additional education and rationale for platform design decisions
  - Runtime config/policy of function

# Workflow

*Ruben Romero, Red Hat*
*Mauricio Salatino (@salaboy), Camunda*

# Workflow - Overview



- Workflow Orchestration Language
  - Concise, portable and vendor neutral
  - Can be used for:
    - Serverless function orchestration
    - Flow coordination
    - Manage Data flow
- Integrates with event-based architectures
    - CloudEvents are first-class citizen
    - Provides visibility and state handling

# Workflow - Model



- Workflow Model:
  - Definition and Validation using JSON Schema
  - Compact and human-readable
  - Minimal complexity (compared to other workflow impls)
  - Support for modularity and logical grouping (SubFlow)
  - Some features under review:
    - Support for custom extensions (added via PR: http://tiny.cc/66tgfz)
    - Support for Metadata (adding via PR: http://tiny.cc/yat7fz)

# Workflow - Example



{} emergency-scenario.json > ...

```
{
    "name": "Emergency Handling",
    "version": "1.0",
    "startsAt": "emergencyReported",
    "triggerDefs": [{   "name": "emergencyReported",
                        "source": "emergency-reported-uri",
                        "type": "pkg/emergencyReported"
                    }],
    "states": [{
                    "name": "emergencyReported",
                    "type": "EVENT",
                    "events": [{ "eventExpression": "name eq 'emergencyReported",
                                 "actions": [{ "function": {
                                                    "name": "auditEmergencyReported",
                                                    "type": "HTTP",
                                                    "parameters": {
                                                        "emergency": "$emergency"
                                                    }}
                                             }],
                               "nextState": "coordinateEmergency"}]
                },
                {
                    "name": "coordinateEmergency",
                    "type": "OPERATION",
                    "actions": [{ "function": {
                                    "name":"initiateProcedure",
                                    "type": "HTTP"
                                }},
                                {"function": {
                                    "name":"auditProcedureInitiated",
                                    "type": "HTTP"
                                }}],
                    "nextState": "followUpEmergency"
                },
                {
                    "name": "followUpEmergency",
                    "type": "OPERATION",
                    "actions": [{ "function": {
                                    "name": "validateEmergencyFinished"
                                    }}],
                    "end": true
                }
    ]
}
```

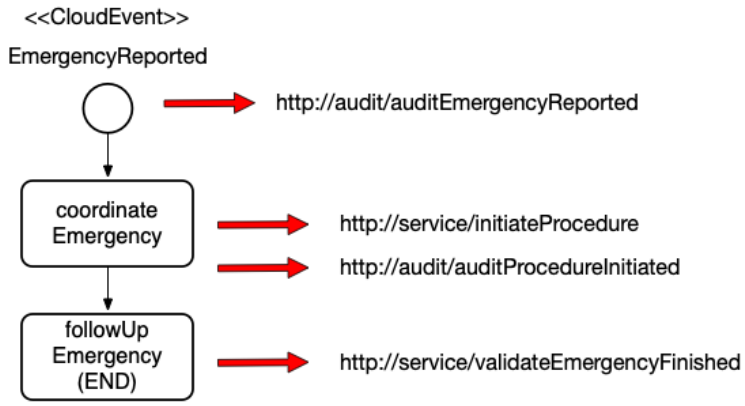
CloudEvents Mapping

States/Nodes

Context/Data

Actions

Function Reference



# Workflow – Community Implementations



- http://tiny.cc/n4yffz - Java API/SPI (open-source)
- http://tiny.cc/37yffz - Java Impl (open-source)
- http://tiny.cc/3gs8fz - Java CLI Linter/Validator (open-source)
- http://tiny.cc/shs8fz - VSCode Extension (JSON Schema support) (open-source)

# Workflow – Community & Status



## Current Status

- Active project
- Continuous enhancements on spec doc: http://tiny.cc/ke0ffz
- Involvement by multiple companies (contribution, impl etc)
- Growing community support

## What is needed to move forward

- Strong leadership from involved parties
- Better organizational structure
- Interest from more companies and the community
  - GitHub: https://github.com/cncf/wg-serverless
  - Email: cncf-wg-serverless@lists.cncf.io
  - Subscription: https://lists.cncf.io/g/cncf-wg-serverless
  - Slack: https://slack.cncf.io/ #serverless-workflow

KubeCon | CloudNativeCon

North America 2019

# BoF

*Community*

# BoF - We want to hear from you!



- Questions for the community:
  - Who is using Serverless in today?
    - Who is using it in production?
    - Using it for tooling, e.g. CI/CD?
    - Who is using it in public vs private?
    - Main usecases / workloads?
    - What's holding people back from using it?
  - What are the pain points you are experiencing w.r.t. Serverless?
    - Interop? Portability? Tooling? Debugging?
  - Are there certain workloads that aren't well supported today but you'd like to see?
  - Is the split between PaaS, CaaS and FaaS/Serverless confusing? Meaningful?

# BoF Community Discussion



- What's next?

- Proposed work streams
  - **Packaging contract**
  - **Function signatures**
  - **Subscription API**
  - **Discovery**
  - Observability, tracing
  - Schema registry (CloudEvents)
  - E2E Security
  - Additional education and rationale for platform design decisions
  - Runtime config/policy of function

# Thank You!



- Serverless WG : https://github.com/cncf/wg-serverless
  - Workflow: https://github.com/cncf/wg-serverless/tree/master/workflow/spec
- CloudEvents : https://cloudevents.io/
  - Org : https://github.com/cloudevents
  - Spec repo : https://github.com/cloudevents/spec
  - SDKs : https://github.com/cloudevents/sdk-...
- Weekly calls : Thursdays at 12pm ET
- Questions?