Serverless Workflow

Specification

SIG App Delivery Sandbox Proposal

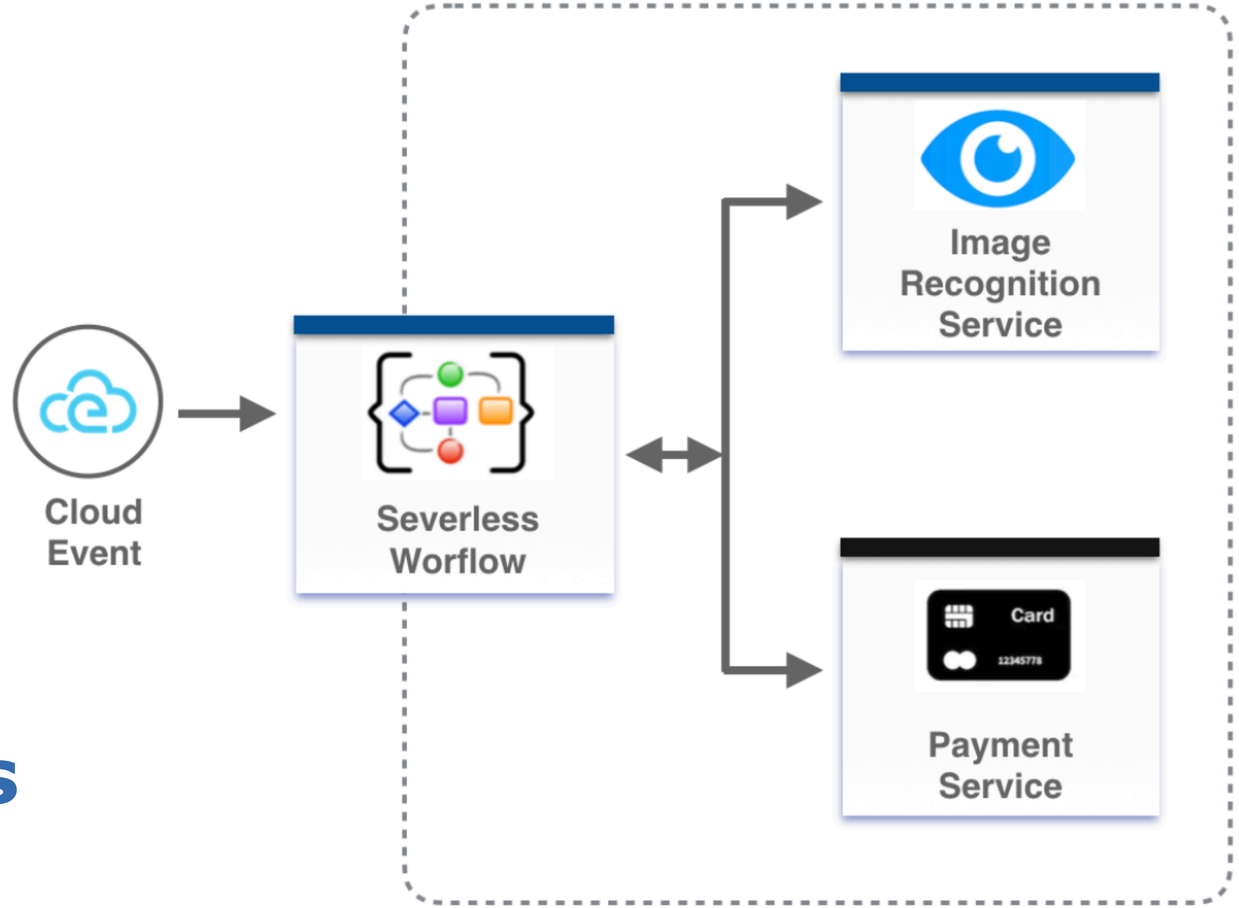
CLOUD NATIVE
COMPUTING FOUNDATION

Agenda

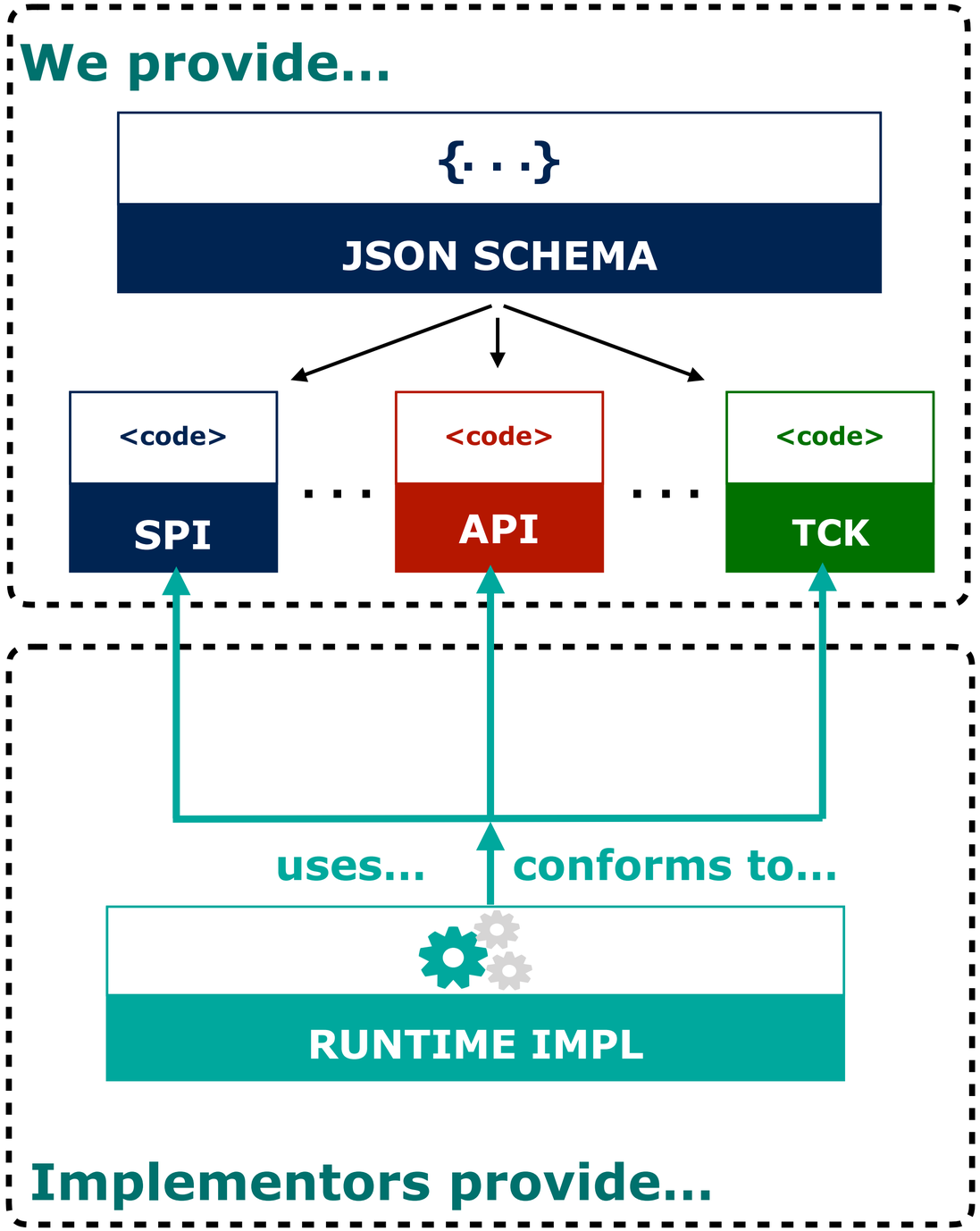
1. Introduction
2. Motivation
3. Key Features
4. Use Cases
5. Project Info
6. Demo

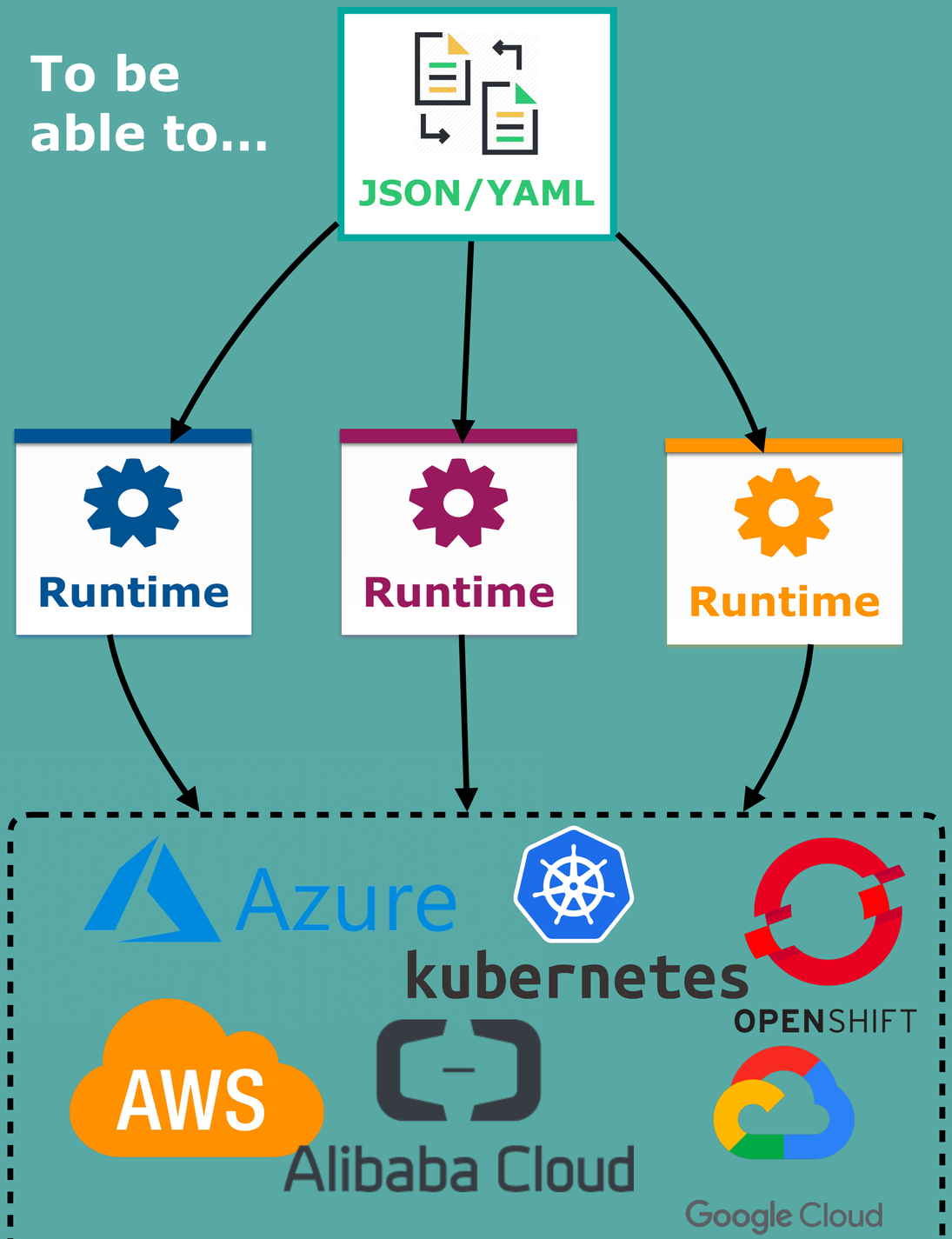
1 Introduction

Vendor-neutral Specification for defining the model of workflows responsible for orchestrating event-driven, serverless applications



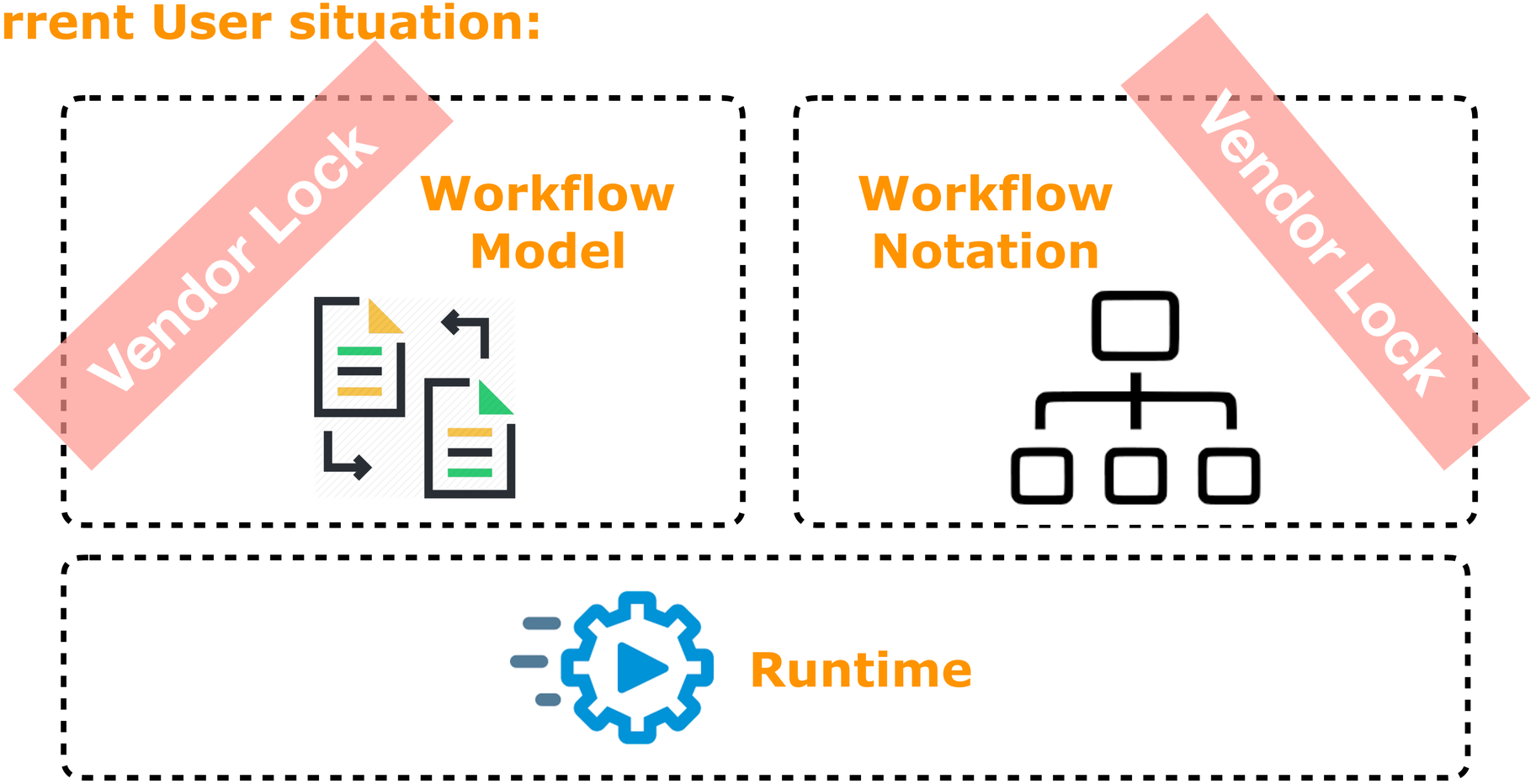
"Coordinate and manage Services and Events"


We provide...
{. . .}
JSON SCHEMA
<code>
SPI
<code>
API
<code>
TCK
uses...
conforms to...
RUNTIME IMPL
Implementors provide...


To be able to...
JSON/YAML
Runtime
Runtime
Runtime
Azure
kubernetes
OPENSHIFT
AWS
Alibaba Cloud
Google Cloud

Current Workflows implementations:

Current User situation:



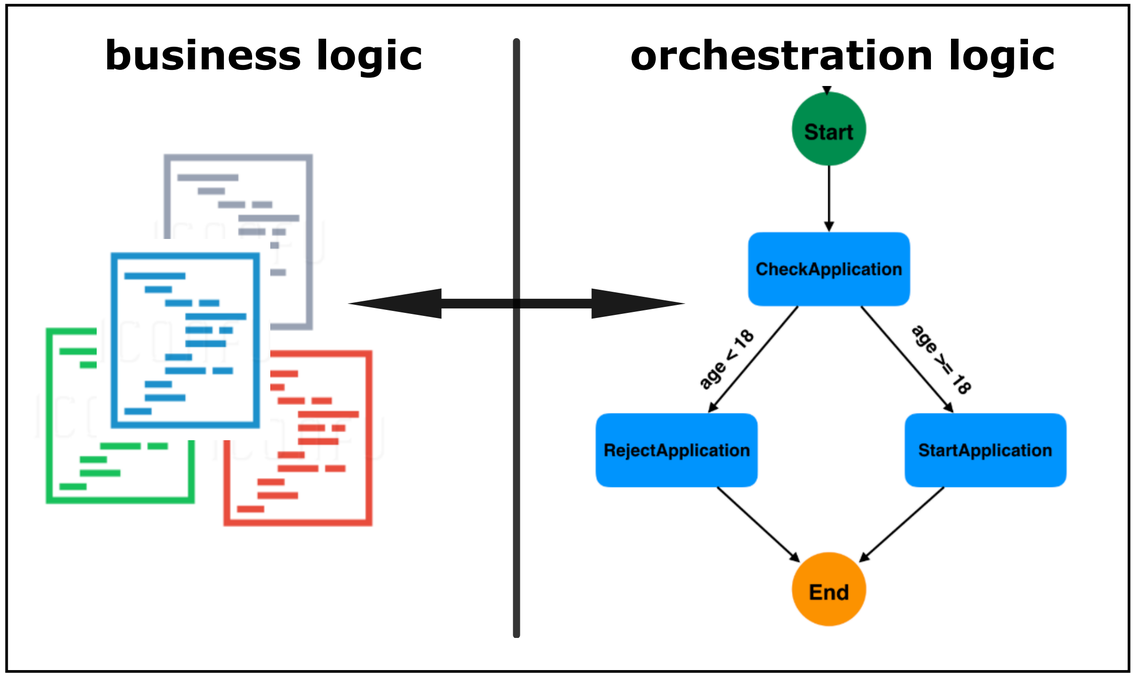
We need a portable and vendor-neutral specification!

3

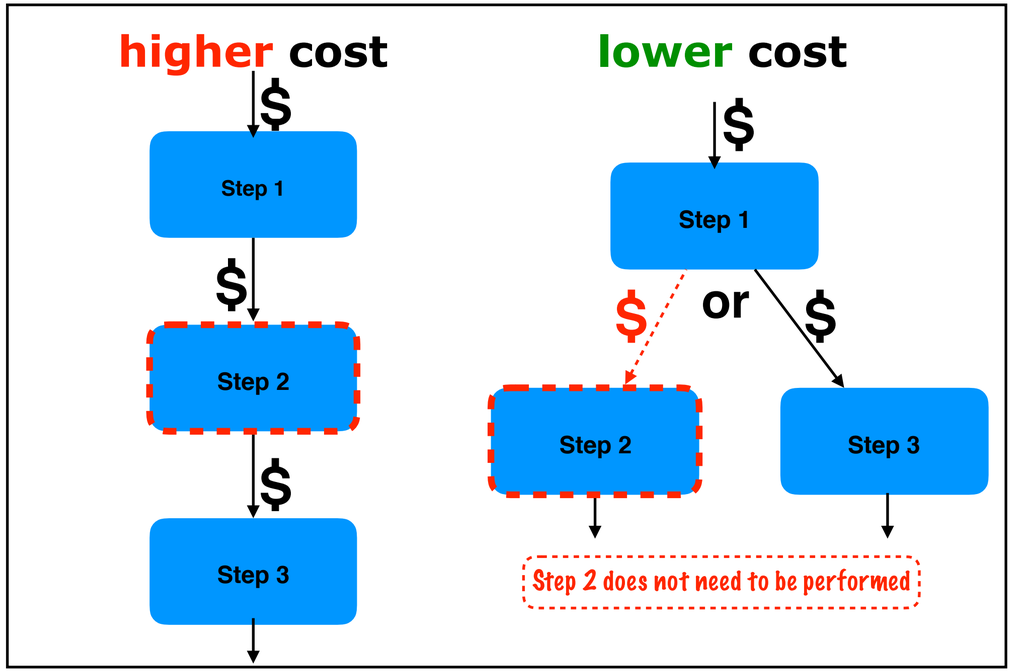
Key Features

Core Features categories:

Clear Separation of Concerns



Execution Cost



3

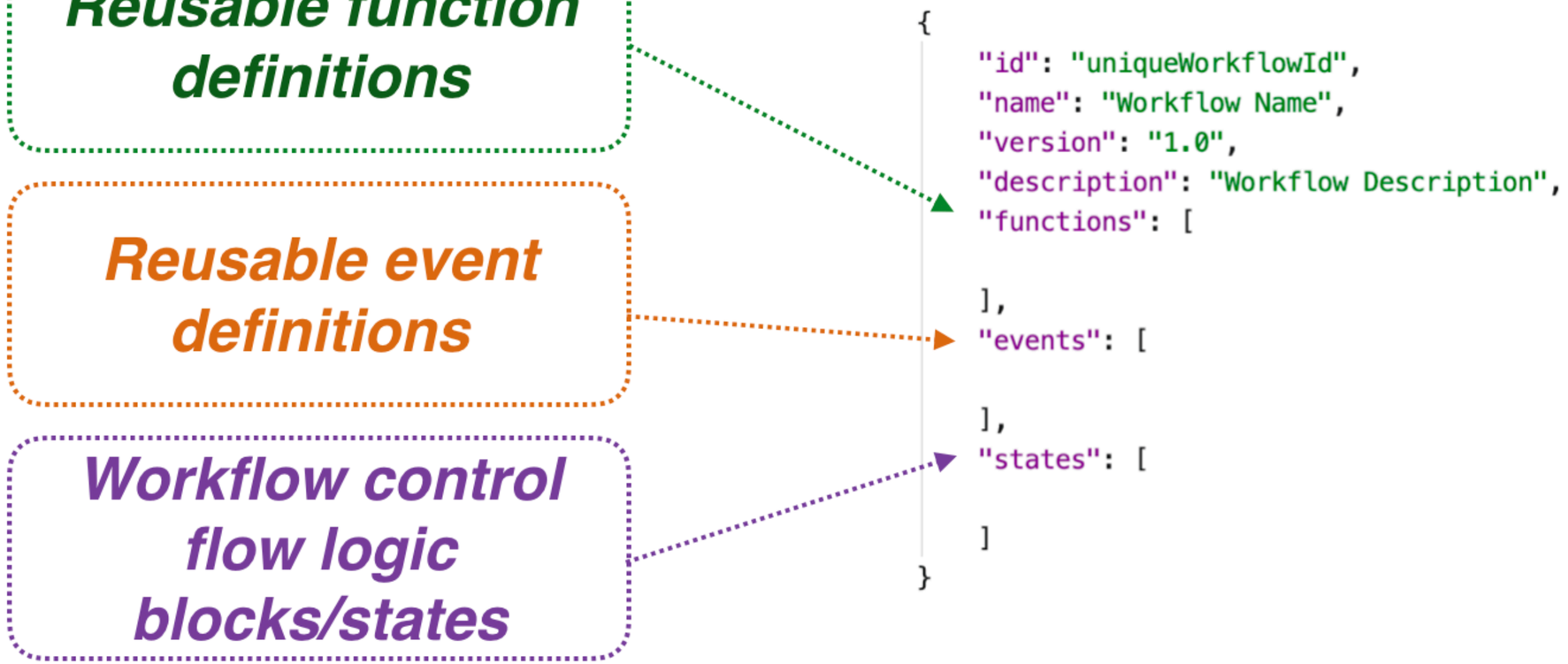
Key Features

Model Definition

Reusable function definitions

Reusable event definitions

Workflow control flow logic blocks/states



3

Key Features

Function Definitions

```
"functions": [
    {
        "name": "HelloWorldFunction",
        "resource": "arn:aws:lambda:us-east-1:123456789012:function:lambda-hello-world"
    },
    {
        "name": "HellowWorldFunction2",
        "resource": "myhellofunctionapp.azurewebsite.net/api/hellofunction",
        "type": "GET"
    },
    {
        "name": "HellowWorldFunction3",
        "resource": " https://openshift.redhat.com:8443/oapi/v1/sayhello",
        "type": "REST",
        "metadata": {
            "Authorization: Bearer": "$.token"
        }
    }
]
```

3

Key Features

Event Definitions

```
"events": [
  {
      "name": "ApplicationSubmitted",
      "type": "org.application.submitted",
      "source": "applicationsource",
      "correlationToken": "applicantId"
  },
  {
      "name": "SATScoresReceived",
      "type": "org.application.satscores",
      "source": "applicationsource",
      "correlationToken": "applicantId"
  },
  {
      "name": "RecommendationLetterReceived",
      "type": "org.application.recommendationLetter",
      "source": "applicationsource",
      "correlationToken": "applicantId"
  }
  ]
```

CloudEvent

```
{
    "specversion" : "1.0",
    "type" : "org.application.satscores",
    "source" : "applicationsource",
    "subject" : "Received SAT Score",
    "id" : "A234-1234-1234",
    "time" : "2020-01-05T17:31:00Z",
    "applicantId" : "STUDENT-12345",
    "data" : {
        "score": "1600"
    }
}
```

3

Key Features

State Definitions

```
{
    "name": "MyState",
    "type": "STATE_TYPE",
    "start": {
        "kind": "DEFAULT"
    }
    .. type specific params ..,

    "end": {
        "kind": "EVENT",
        "produceEvent": {
            "nameRef": "myEventName",
            "data": "$.person.name"
        }
    },
    ... or,
    "transition": {
        "nextState": "MyNextState"
    }
}
```

Unique Name

Defined Type

Can be start state

Can be end state

or

Can Transition

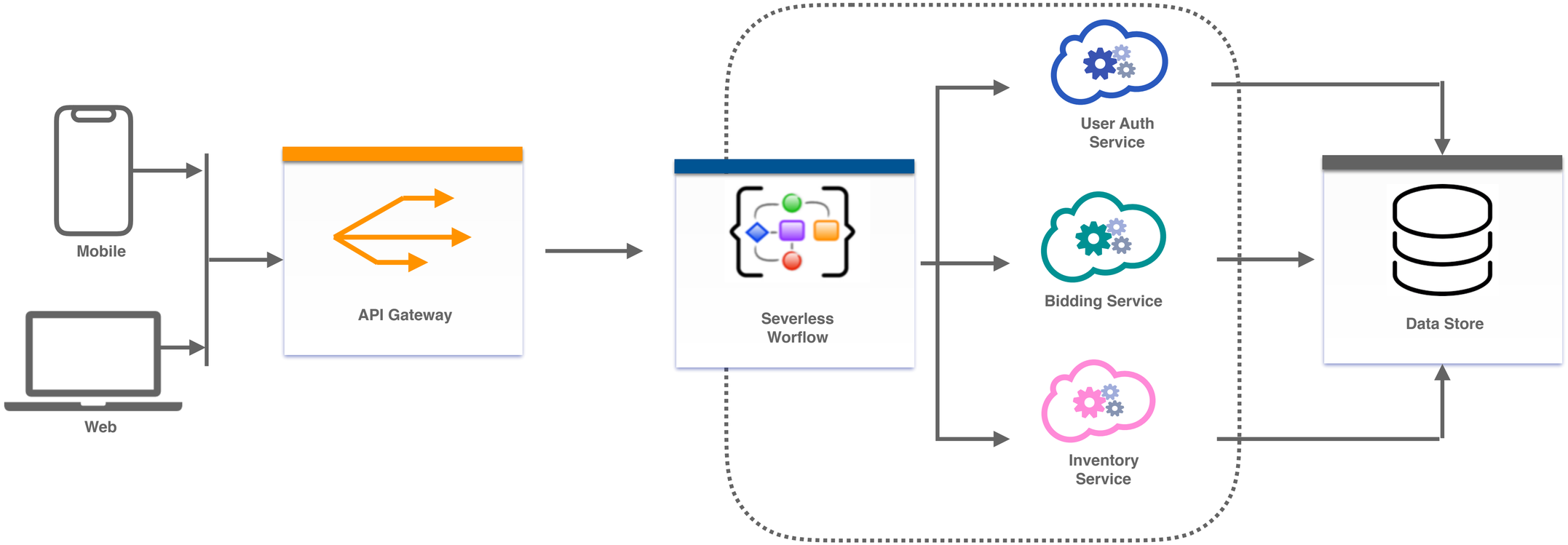
3

Key Features

State Types

Name	Description	Consumes events?	Produces events?	Executes actions?	Handles errors?	Allows parallel execution?	Makes data-based transitions?
Event	Define events that trigger action execution	yes	yes	yes	yes (includes retries)	yes	no
Operation	Execute one or more actions	no	yes	yes	yes (includes retries)	yes	no
Switch	Define data-based workflow transitions	no	yes	no	yes	no	yes
Delay	Delay workflow execution	no	yes	no	yes	no	no
Parallel	Causes parallel execution of branches (set of states)	no	yes	no	yes (includes retries)	yes	no
SubFlow	Represents the invocation of another workflow from within a workflow	no	yes	no	yes	no	no
Relay	Relay state data input to output	no	yes	no	yes	no	no
ForEach	Parallel execution of states for each element of a data array	no	yes	no	yes (includes retries)	yes	no
Callback	Manual decision step. Executes a function and waits for callback event that indicates completion of the manual decision	yes	yes	yes (including retries)	yes	no	no

Example: Online Vehicle Auction



You can use Serverless Workflows to coordinate all steps of an Online Vehicle Auction. These can include:

- Authentication of users making bids.
- Communication with Bidding and Inventory services
- Make decisions to start/end the auction under certain conditions

... many more, see

https://github.com/cncf/wg-serverless/blob/master/workflow/spec/usecases.md

5

Project Info

Group Info	CNCF Serverless WG - Workflow Sub-Group
Communication	Monthly zoom calls - first Monday every month @10-11 PT https://docs.google.com/document/d/1xwcsWQmMiRN24a7o7oy9MstzMroAup31oOkM5Dru1jQ/edit#heading=h.g2rizfze8av2
GitHub Info	https://github.com/cncf/wg-serverless/tree/master/workflow Version 0.1 release: https://github.com/cncf/wg-serverless/tree/v0.1/workflow/spec
Governance	Consensus and Community-driven Current Owners: Red Hat, Nokia, Camunda, Huawei
License	Apache v2.0
Community	Mailing list: https://lists.cncf.io/g/cncf-wg-serverless Slack: #serverless-workflow Slack channel: https://slack.cncf.io/ Blog: https://serverlessworkflow.blogspot.com/
TOC Sponsors	Brendan Burns, Liz Rice

6

Demo

Questions?