RISER Project Cloud Workshop

Making the case for intent-driven orchestration

Thijs Metsch – Intel Labs

intel®

Ecosystem trends*.

Serverless & Event-Driven Architectures

Where user either does not or cannot influence server selection for the workload.

The State of Serverless Report [1]: "Over half of the organization in each Cloud have adopted Serverless."

Low-Code Programming

Integration of existing resources & services to enable rapid (higher-level) app-development.

The Business Value of Low-Code Application Development Platforms [2]: "The global low-code developer population will grow at a CAGR of 40% from 2021 to 2025".

Platform(s) Engineering

Combining platform layers that move "up the stack" as they add functionality while shifting to utility providers as much as possible.

Gartner report on Platform Engineering [3] "by 2026, 80% of software engineering organizations will establish platform engineering teams"

Conversational Programming

Moving away from the traditional IDE, In a conversational programming world you tell the system what you want.

Simon Wardley [4]: „Red Queen hypothesis: contrast the speed of one company with engineers building systems through conversational programming versus [...] company whose engineers are still wiring servers in racks."

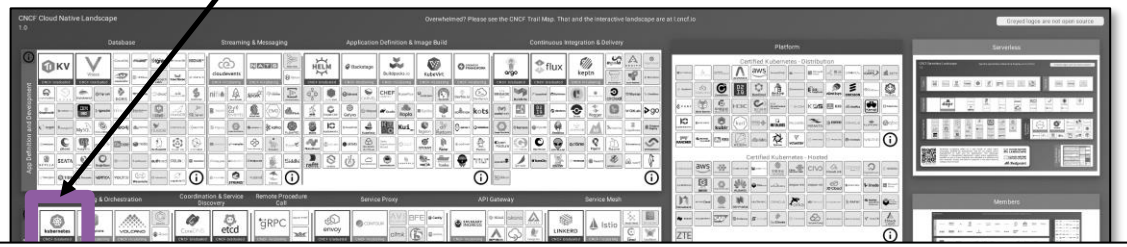
All these trends have 1 common theme: ***increazed abstraction from the underlying hardware.***

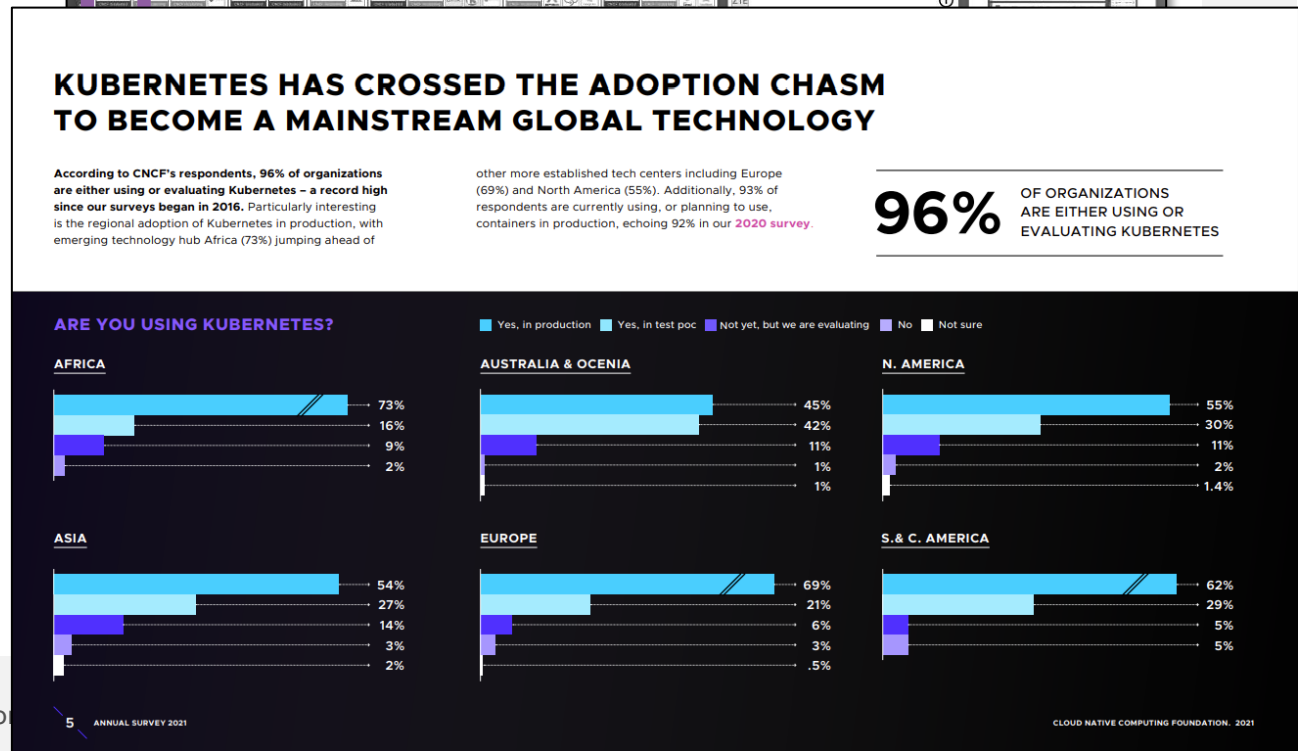
* At least for the frontrunners; typically Enterprise IT and Telcos are slower...

Ecosystem tooling.

- Cloud Native Computing foundation (CNCF)'s mission:
  - the open source, vendor-neutral hub of cloud native computing, hosting projects like Kubernetes and Prometheus to make cloud native universal and sustainable.
- Includes many tools for:
  - Observability
  - Service Orchestration,
  - Resource Orchestration,
  - Infrastructure Orchestration,
  - and many more.

Kubernetes is here…





KUBERNETES HAS CROSSED THE ADOPTION CHASM TO BECOME A MAINSTREAM GLOBAL TECHNOLOGY

According to CNCF's respondents, 96% of organizations are either using or evaluating Kubernetes – a record high since our surveys began in 2016. Particularly interesting is the regional adoption of Kubernetes in production, with emerging technology hub Africa (73%) jumping ahead of other more established tech centers including Europe (69%) and North America (55%). Additionally, 93% of respondents are currently using, or planning to use, containers in production, echoing 92% in our 2020 survey.

96% OF ORGANIZATIONS ARE EITHER USING OR EVALUATING KUBERNETES

ARE YOU USING KUBERNETES?

Yes, in production | Yes, in test poc | Not yet, but we are evaluating | No | Not sure

AFRICA: 73% 16% 9% 2%

AUSTRALIA & OCENIA: 45% 42% 11% 1% 1%

N. AMERICA: 55% 30% 11% 2% 1.4%

ASIA: 54% 27% 14% 3% 2%

EUROPE: 69% 21% 6% 3% .5%

S.& C. AMERICA: 62% 29% 5% 5%

5 ANNUAL SURVEY 2021

CLOUD NATIVE COMPUTING FOUNDATION. 2021

Figure 2: CNCF survey 2021 [1]

Resource Orchestration.

- Kubernetes is the leading platform for, cloud-native workloads – supporting various use cases.
- The behavior in terms of both the application and the infrastructure is a complex function, depending on e.g.:
  - POD specs, the IaaS used, and of course the physical compute, network, storage, and accelerator choice/configuration.
- While applications can be deployed unmodified in many differently configured Kubernetes environments, their KPIs will vary.

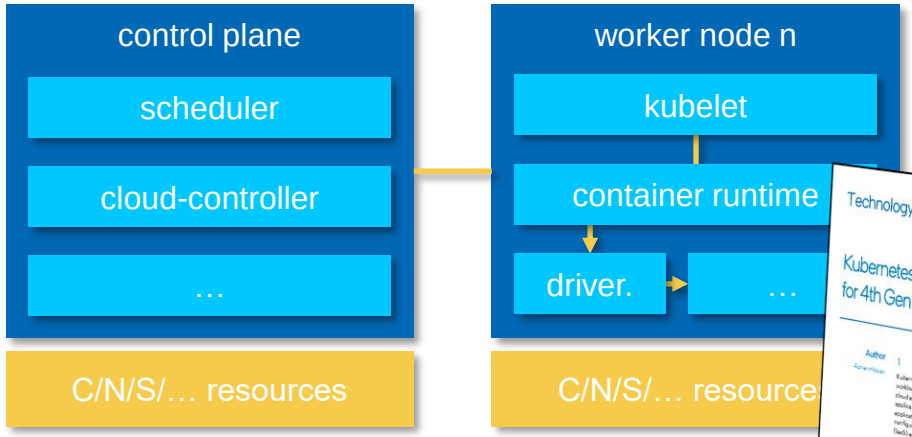


Figure 1: Kubernetes overall architecture.


Technology Guide

intel

Kubernetes - Resource Orchestration for 4th Gen Intel® Xeon® Scalable Processors

Figure 2: Kubernetes – Resource Orchestration for 4th Gen Intel® Xeon® Scalable Processors

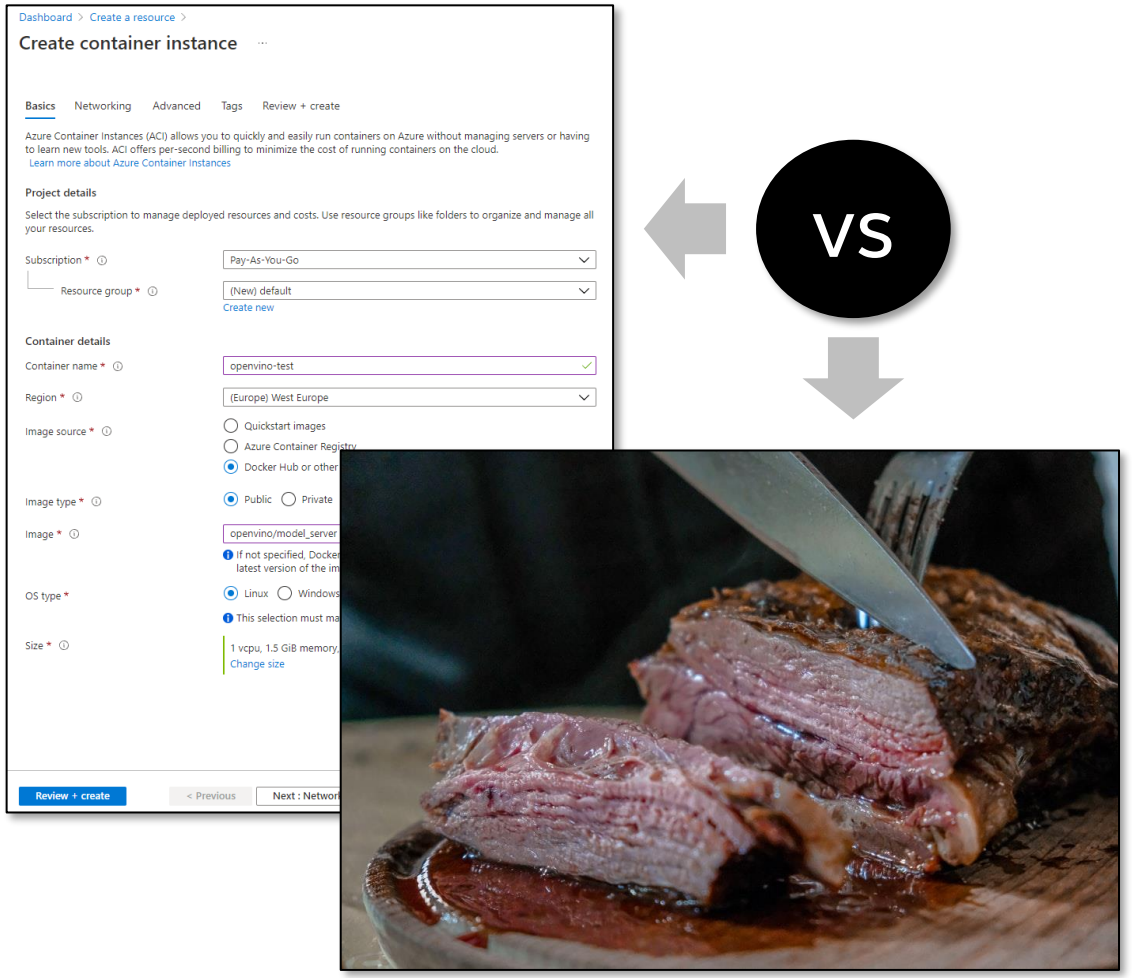
Intents.

- Intent as a „*a determination to act in a certain way*"…
- Intents address aspects of:
  - Enable Portability,
  - Are invariant,
  - Enable efficiency by understanding context.

Figure 1: Intent: Don't Tell Me What to Do! (Tell Me What You Want) [1]

Motivation; simplify adoption.

- *Goal*: enable platform features while embracing Cloud Native & Serverless methodologies & abstraction.
- *How*: Use intent-driven orchestration models to steer workloads while ***minimizing the user and cloud administrator's overhead.***
- *What*: Develop intent-driven orchestration capabilities; aka creating a Michelin star chef for K8s control planes.



Intent Driven Orchestration.

```
[…]
spec:
  replicas: 2
  template:
    spec:
      containers:
        - name: sample-function
          image: sample_function:0.1
          resources:
            requests:
              cpu: 0.75
              intel/cache: 4
              intel/cpu: 1
            […]
          securityContext:
            privileged: true
[…]
```

Figure 1: Kubernetes example manifest file – declarative state for resources.

```
apiVersion: "ido.intel.com/v1alpha1"
kind: Intent
metadata:
 name: my-intent
spec:
 targetRef:
   name: my-nginx-deployment
 priority: 1.0
 objectives:
   - name: p99-compliance
     value: 100
     measuredBy: default/p99latency
   - name: availability
     value: 0.99
     measuredBy: default/availability
   […]
```

Figure 2: Example Intent CRD.

- From declarative state to objective driven:
  - Enabled through planning (what/how) and scheduling (when/where) components.
- Enable IA feature differentiation while abstractions happening on user level (e.g. Serverless).
- Fundamental shift in how we do orchestration → towards automation & context awareness.

Why Planning?

- **Planning** is the process of thinking regarding the activities required to achieve a desired goal.
- Planning is a key component to achieve autonomic computing and address the self-* properties [3].
- Hence, it is essential we extend existing control-planes to support continious planning & decision making.

Space – aka „the exteme edge“

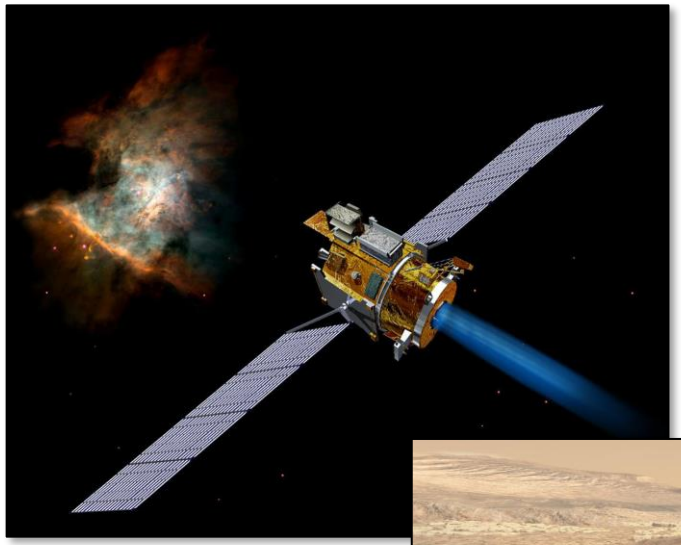
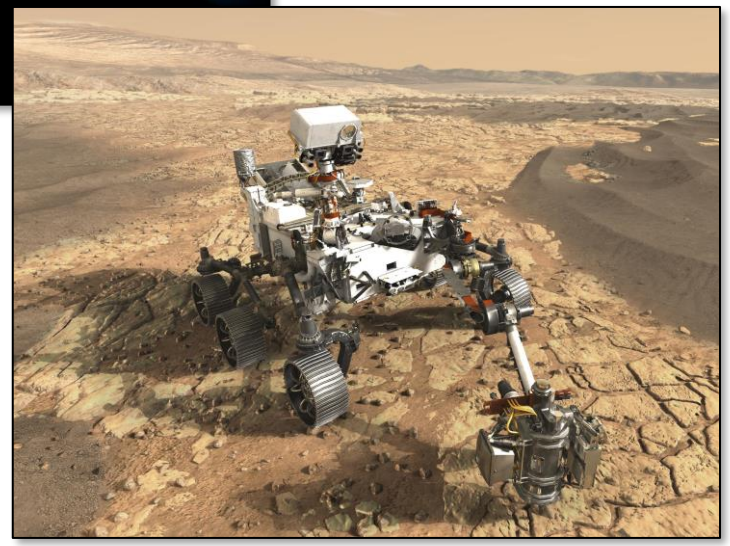
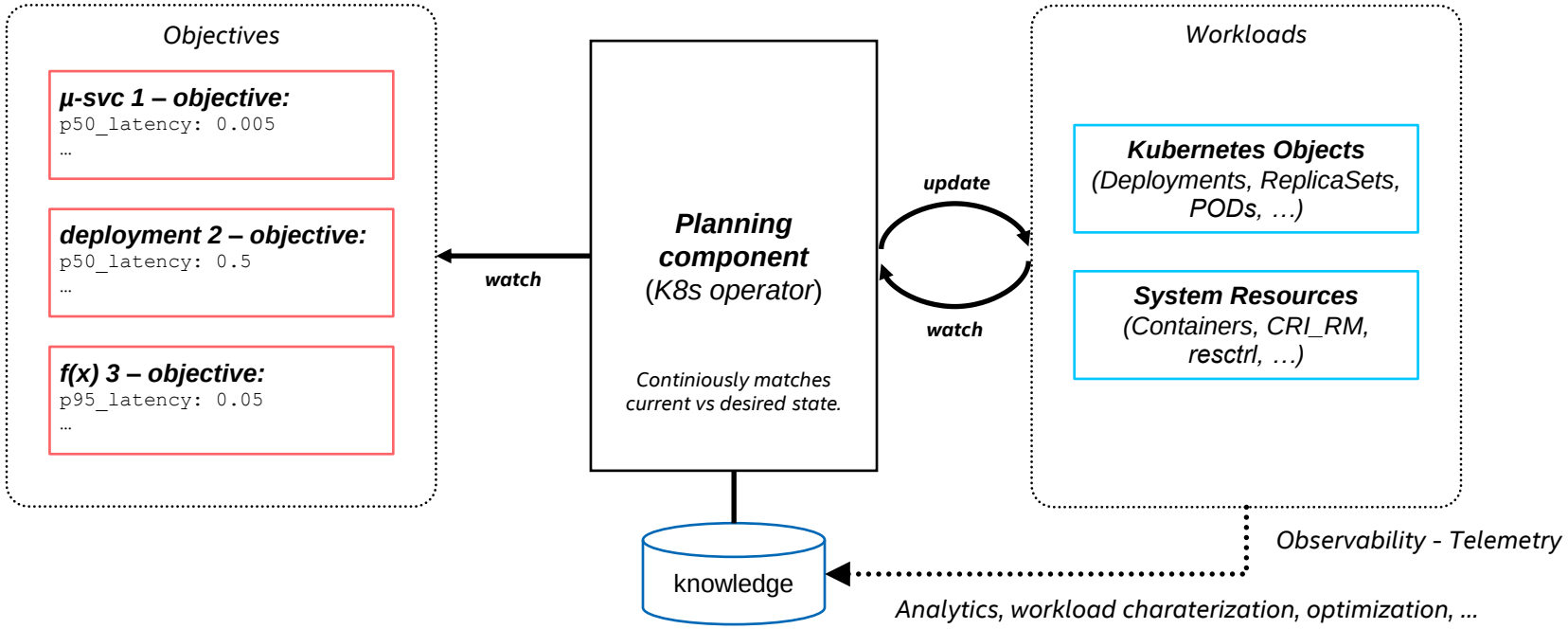


Figure 1: Deep Space 1 [1]

Figure 2: Mars 2020 Rover [2]

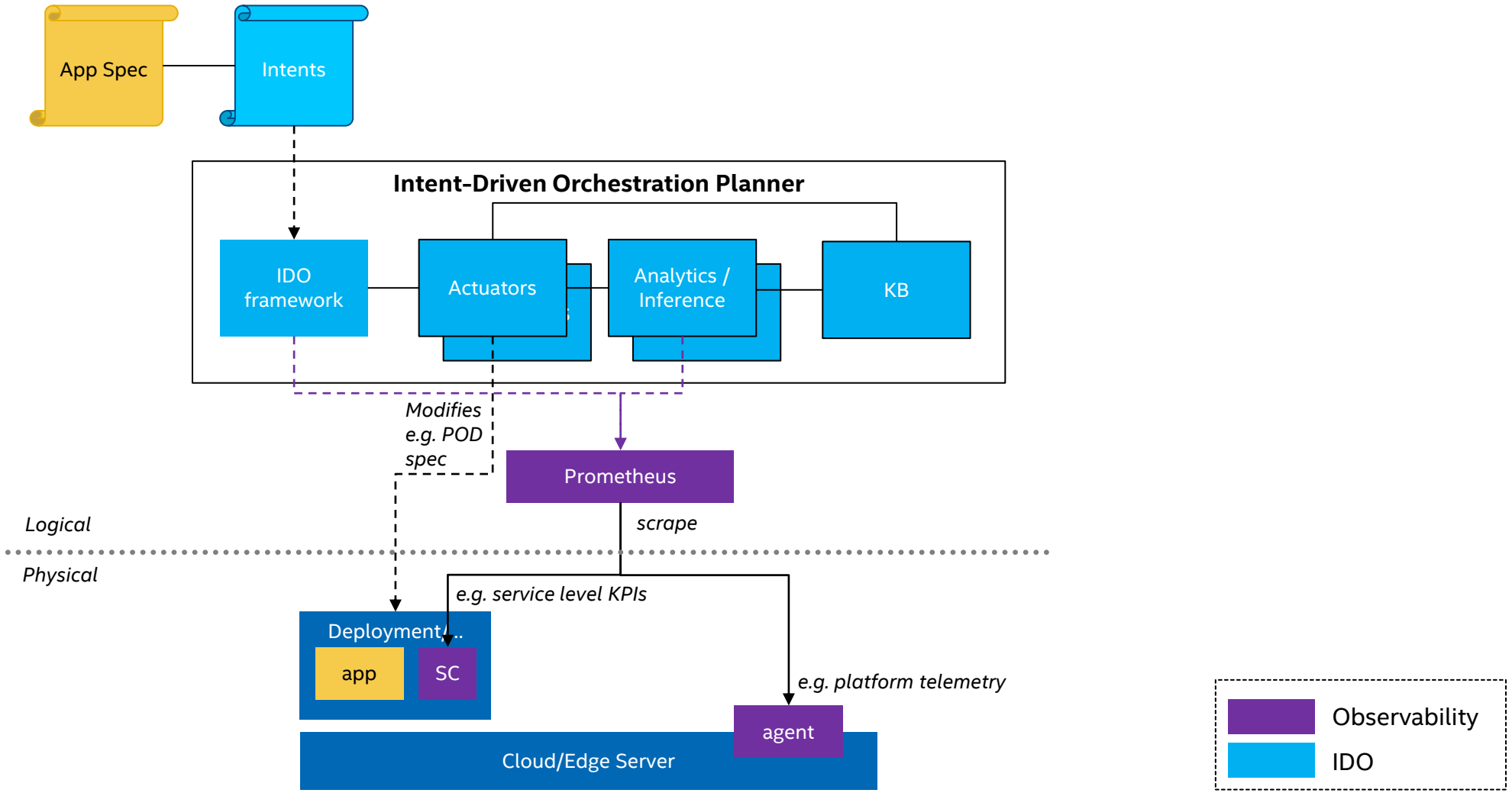


K8s* Operator & planning component.



* Or similar – same methdology could be applied to other orchestrators as well…

Intent-Driven Orchestration Planner.



Intent Declarations.

- An SLO is a service level objective: a target value or range of values for a service level that is measured by an KPI/SLI.
- *„Let the user what they truely care about – their objectives.“*
- *Note*: with intent-driven orchestration we do not strive to build an SLA management solution.

```
apiVersion: "ido.intel.com/v1alpha1"
kind: Intent
metadata:
  name: my-function-intent
spec:
  targetRef:
    kind: "Deployment"
    name: "default/function-deployment"
  objectives:
    - name: my-function-p95compliance
      value: 4
      measuredBy: default/p95latency
    - name: my-function-availability
      value: 0.99
      measuredBy: default/availability
[…]
```

Figure 1: Example Intent.

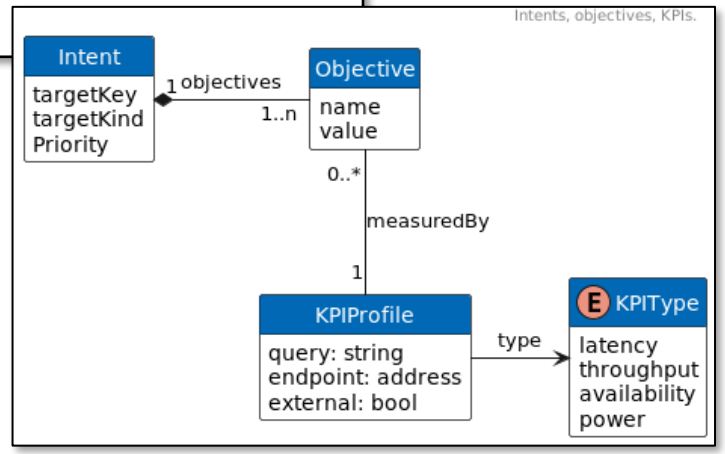
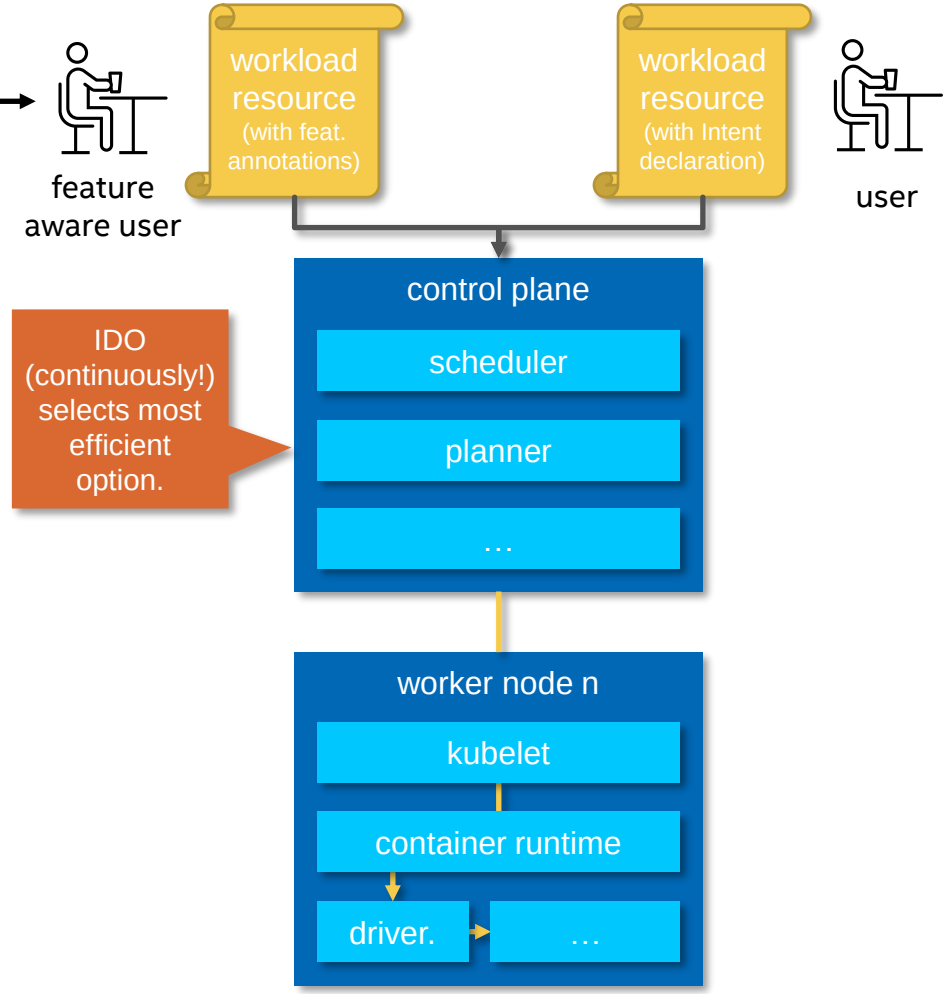


Figure 2: Interface definition.

Imperative intents vs declarative resource asks.

```
apiVersion: v1
kind: Pod
metadata:
  name: my-nginx-pod
spec:
  containers:
  - name: nginx
    image: nginx
    resources:
      requests:
        memory: "200Mi"
        cpu: "2"
        performance_profile: "2"
  nodeSelector:
    feature/cpu-cpuid.AESNI: "true"
```

Figure 1: workload manifest with explicit resource requests.




```
apiVersion: "ido.intel.com/v1alpha1"
kind: Intent
metadata:
 name: my-intent
spec:
 targetRef:
   name: my-nginx-deployment
 priority: 1.0
 objectives:
   - name: p99-compliance
     value: 100
     measuredBy: default/p99latency
   […]
```

Figure 2: intent declarations for a given workload.

Example – CPU rightsizing.

- Resource rightsizing is a key obstacle in K8s adoption.
  - Portable between environments – regardless if you are running on Atom to Xeon.
- Support for vertical rightsizing of resource requests & limits.
  - Initially targeting CPU rightsizing.

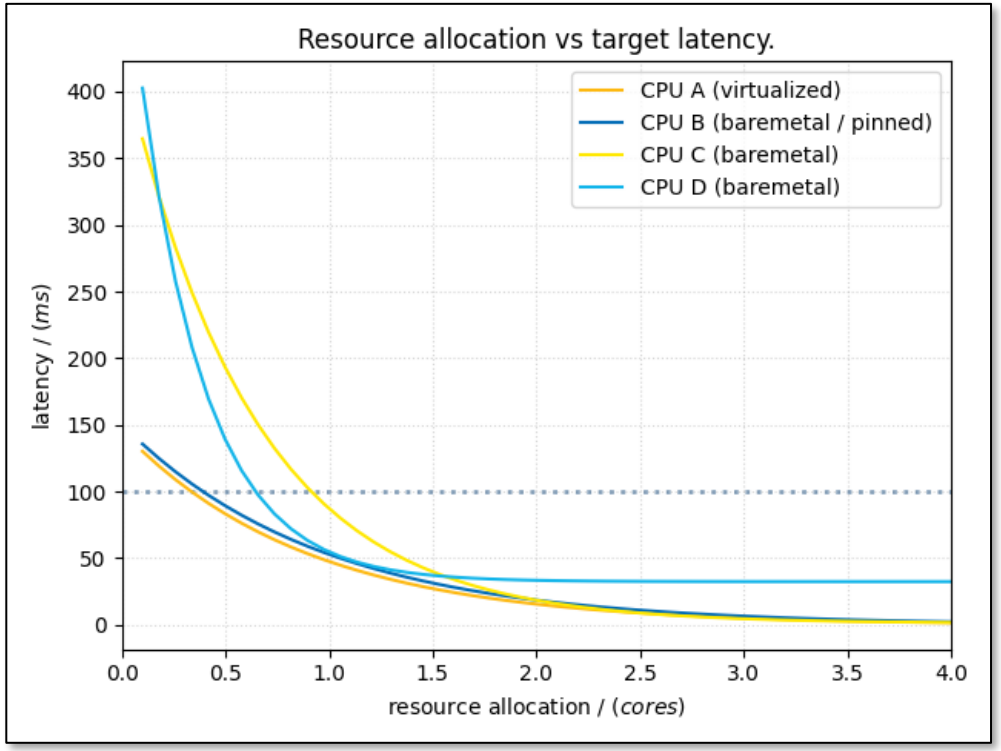


Figure 1: Visualization of the model underlying the CPU rightsizing actuator.

Example – CPU rightsizing (2).

- Support for Online/Offline AI/ML based analytics based on data coming from observability stack. Enables closed-loop automated
- Analytics is based on curve-fitting:
  - $latency = p_0 * e^{(-p_1 * cores)} + p_2$
- Actuation is done through injecting resource requirements in the POD spec.

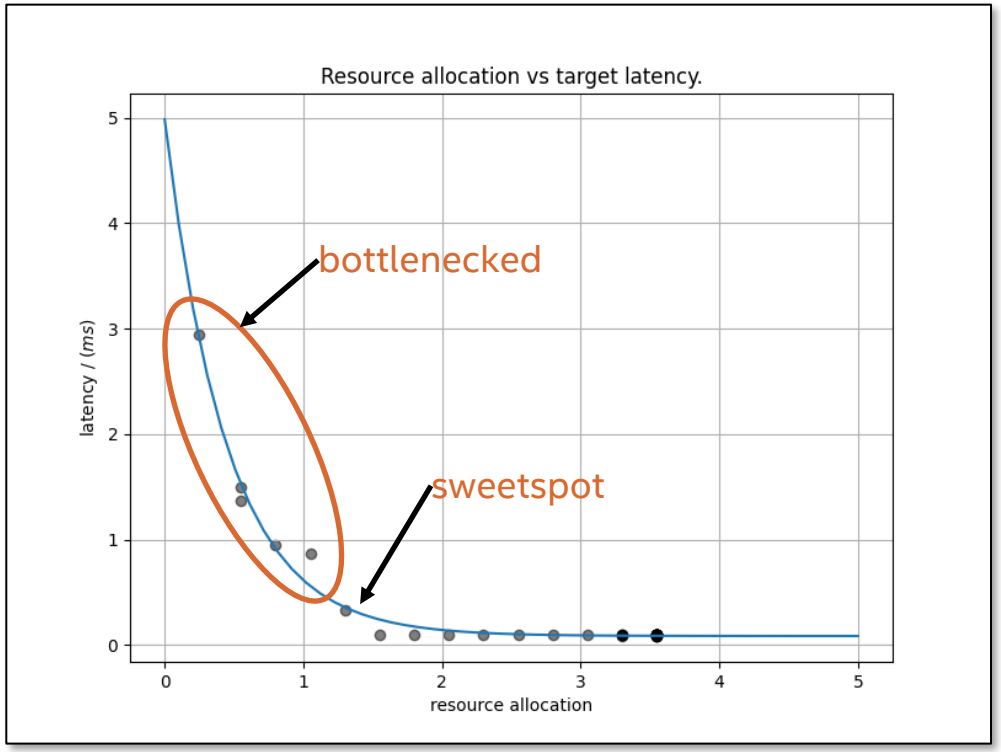
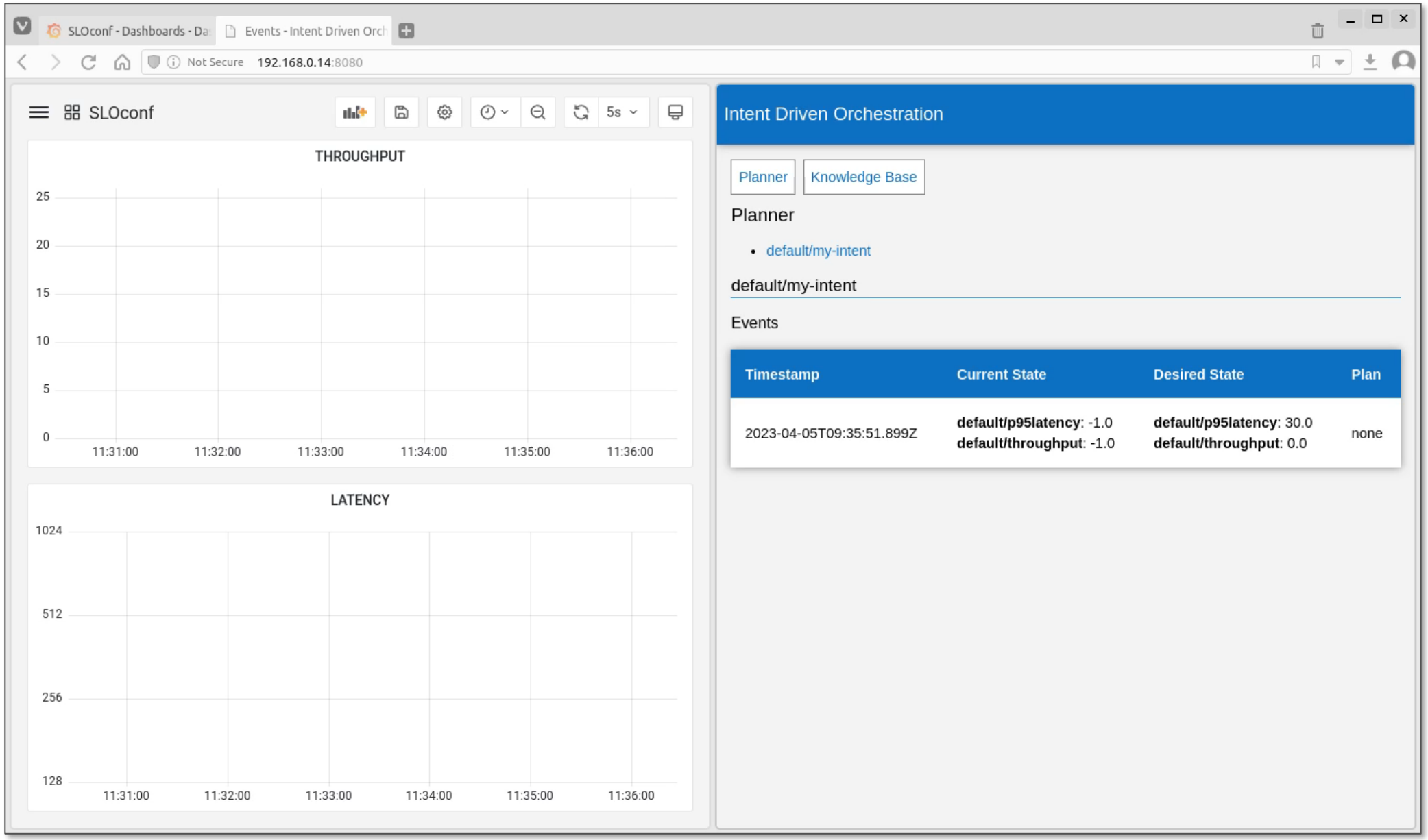


Figure 1: model describing effect of CPU rightsizing.


SLOconf - Dashboards - Da
Events - Intent Driven Orch
Not Secure 192.168.0.14:8080
SLOconf
5s
THROUGHPUT
25
20
15
10
5
0
11:31:00
11:32:00
11:33:00
11:34:00
11:35:00
11:36:00
LATENCY
1024
512
256
128
11:31:00
11:32:00
11:33:00
11:34:00
11:35:00
11:36:00
Intent Driven Orchestration
Planner
Knowledge Base
Planner
default/my-intent
default/my-intent
Events
Timestamp
Current State
Desired State
Plan
2023-04-05T09:35:51.899Z
default/p95latency: -1.0
default/throughput: -1.0
default/p95latency: 30.0
default/throughput: 0.0
none

Outlook.

- A framework to drive the shift from declarative state to declarative objectives.
- The the Intent Driven Orchestration Planning component for Kubernetes now.
- *Join us* in reshaping how resource orchestration is done today!

intel/**intent-driven-orchestration**

Intent Driven Orchestration enables management of applications through their Service Level Objectives, while minimizing developer and administrator overhead.

7 Contributors | 1 Issue | 33 Stars | 7 Forks

GitHub

https://github.com/intel/intent-driven-orchestration

intel®