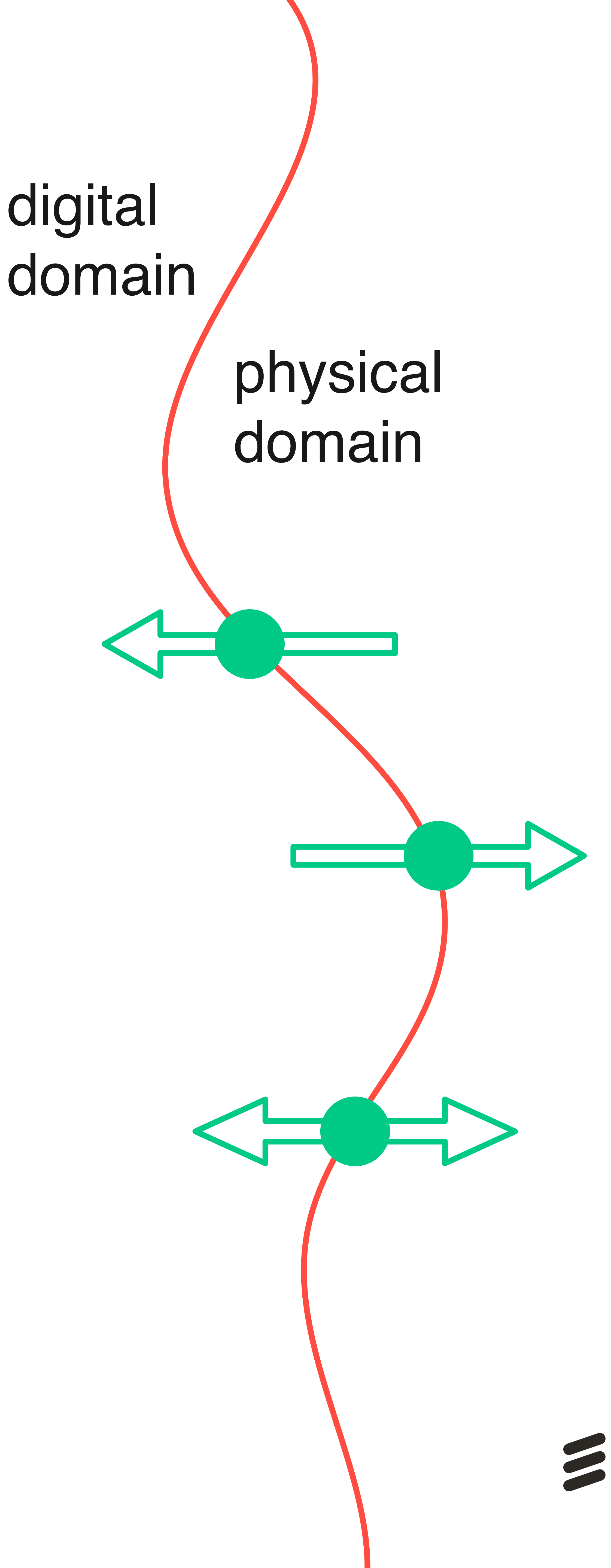
Kappa–Serverless IoT

Per Persson
Ericsson Research
Lund, Sweden

Serverless IoT

Does it even make sense?

- IoT is about
  - capturing, digitally representing, and manipulating the physical world
  - devices forming the interface between the domains
  - long-running, stateful applications
- Things IoT should adopt from cloud and serverless
  - simplicity
  - resource sharing, multi-tenancy
  - fine-grained metrics



digital domain

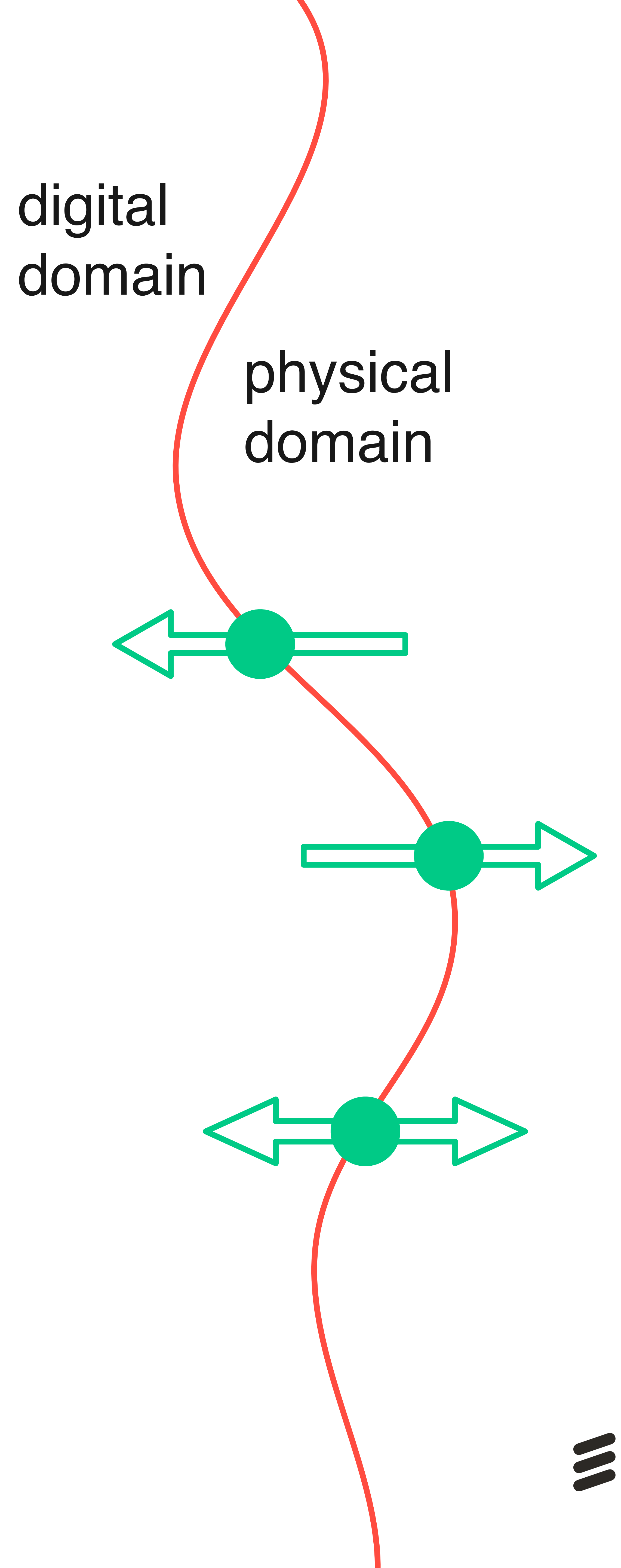
physical domain

Serverless IoT

Does it even make sense?

- IoT is about
  - capturing, digitally representing, and manipulating the physical world
  - devices forming the interface between the domains
  - long-running, stateful applications
- Things IoT should adopt from cloud and serverless
  - simplicity
  - resource sharing, multi-tenancy
  - fine-grained metrics

Serverless computing	Serverless IoT
transparent provisioning	semantics
stateless	local state
short lived	long lived, but mostly sleeping
use your favourite language/paradigm	restricted programming model



IoT-programming with Calvin

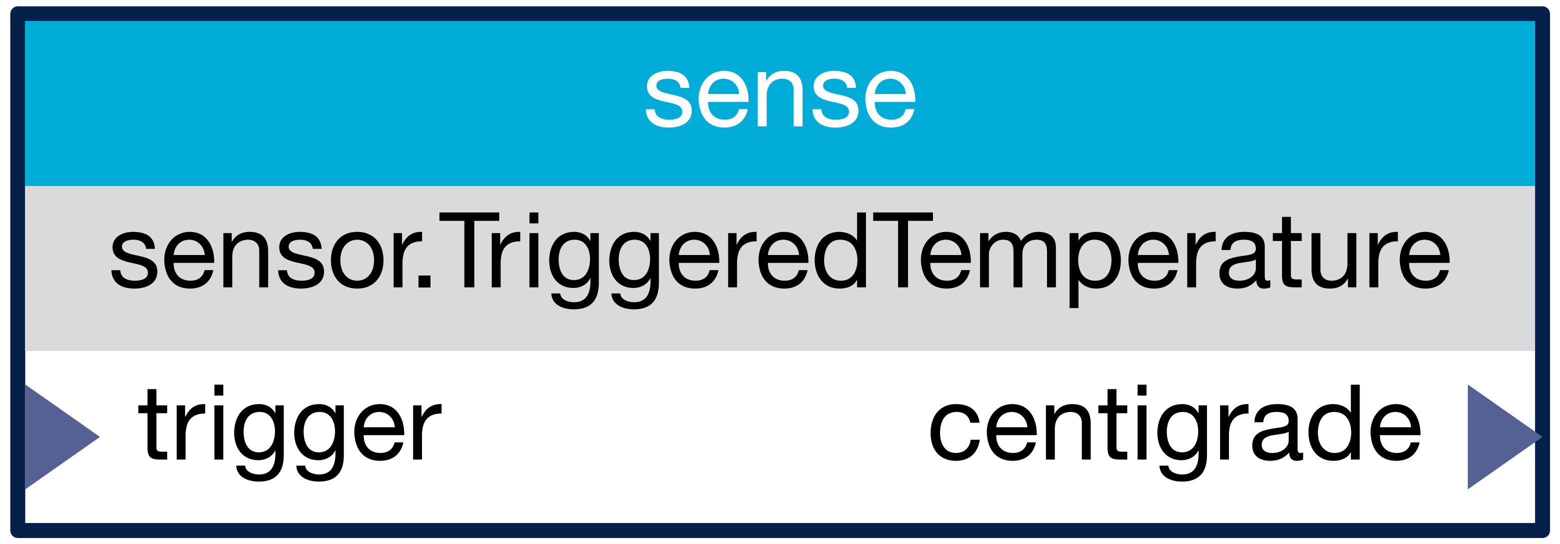
Separating the what from the where and how

Runtimes and Actors

capabilities:
sense.temp
math.fft

attributes:
location="kitchen"
owner="me"

Runtime

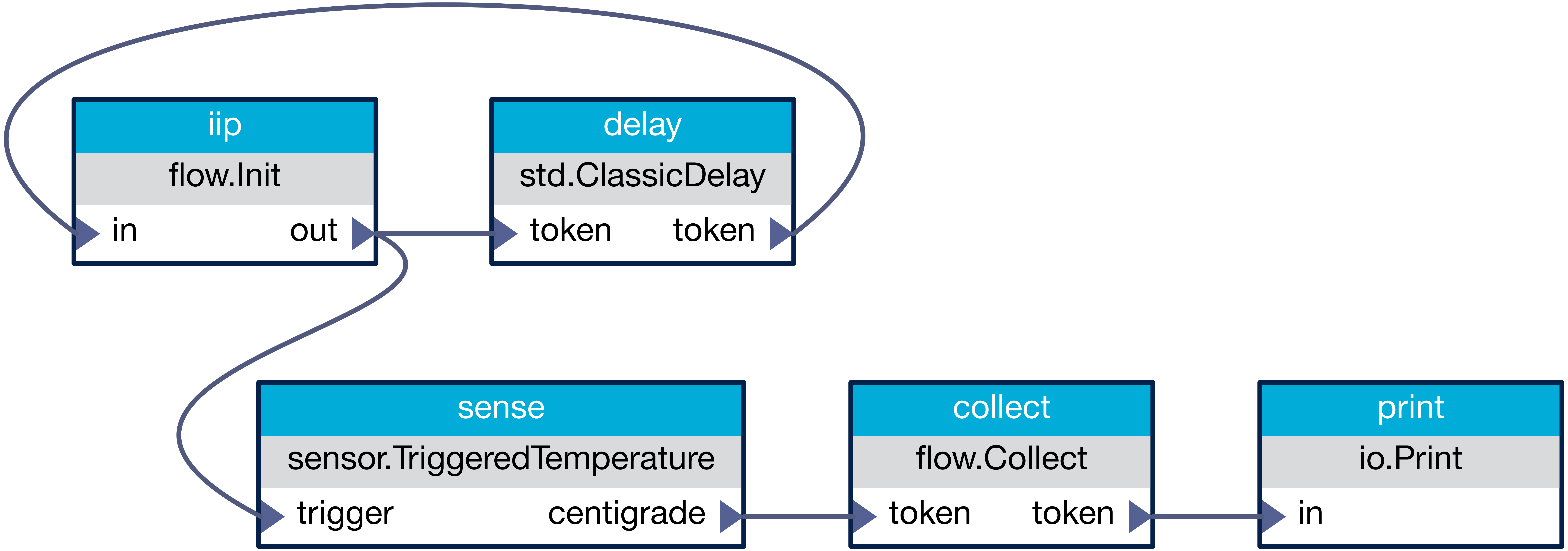


Actor

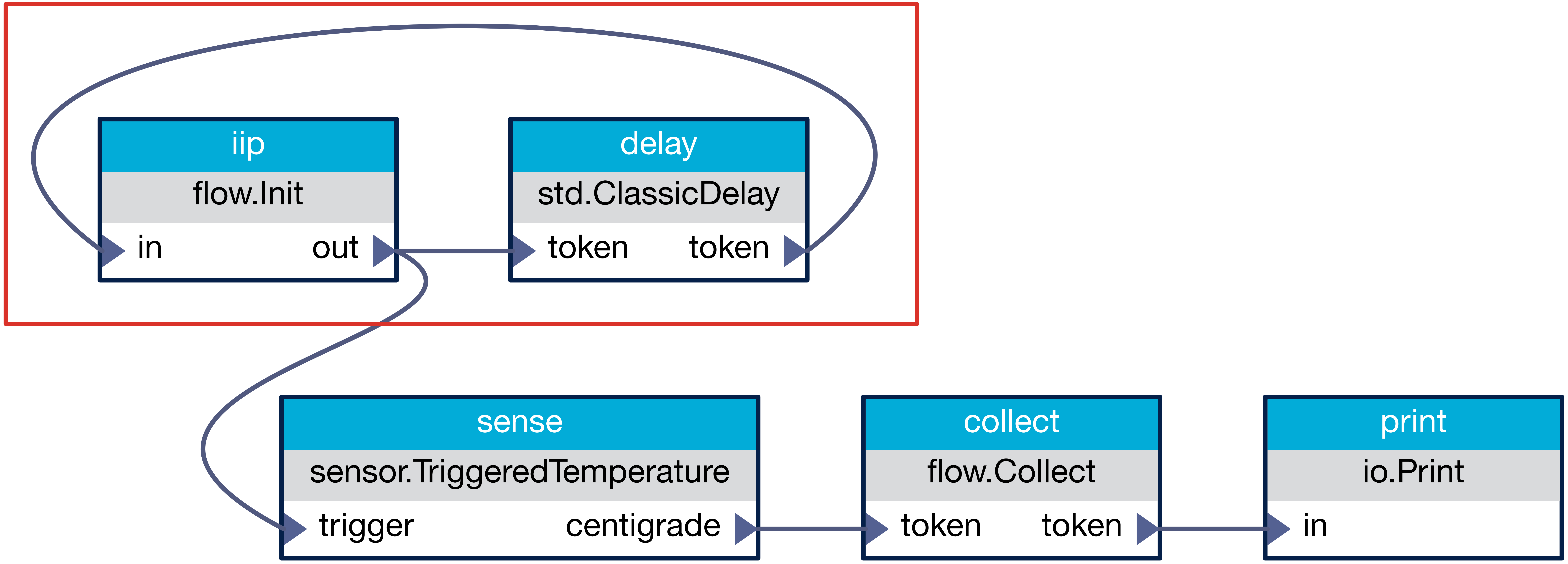
- Portable platform abstraction layer
  - ARM CortexM (constrained)
  - ARM CortexA and up to DC (base)
- Tracks capabilities and attributes
- Handles message passing and security
- Mesh of runtimes create **single machine illusion**

- Unit of isolation and execution
- Responds to events and incoming messages
- Atomic operations
- Local state only
- Moves to runtime to access its resources
- Common code across all runtimes

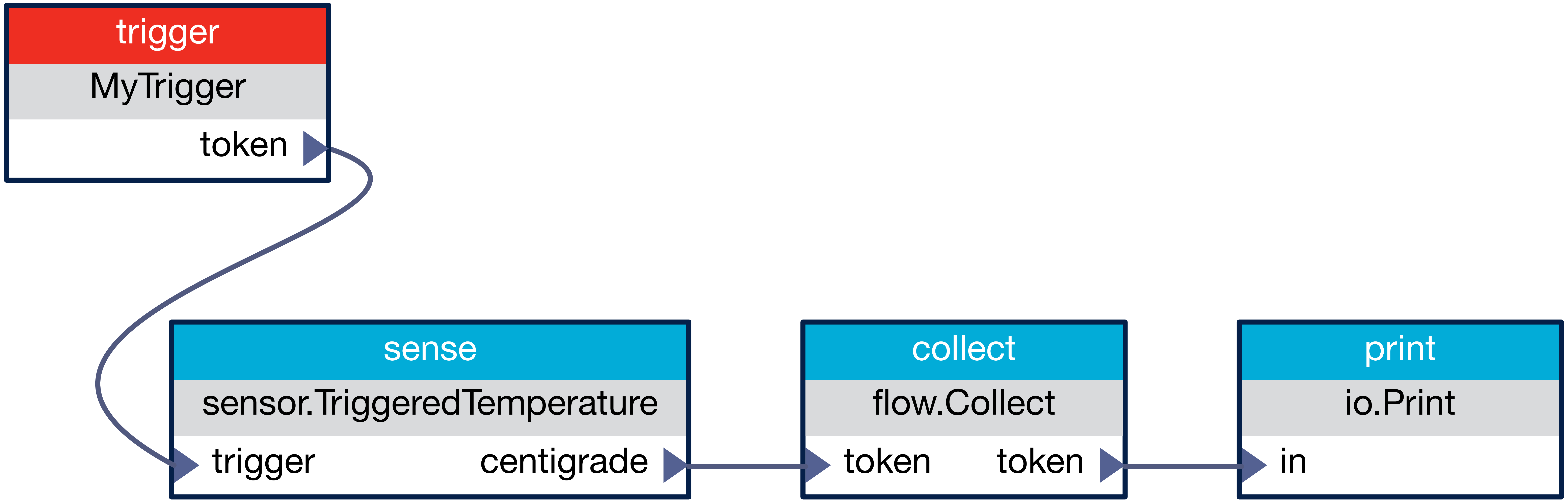
Applications



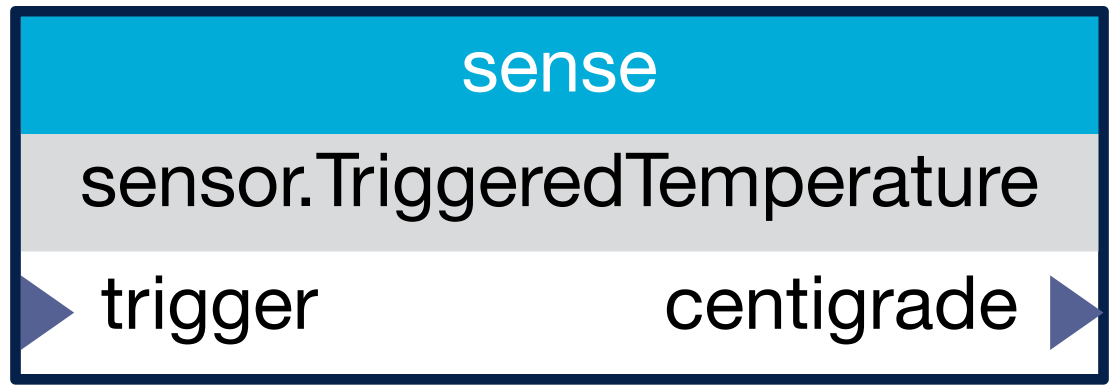
Applications



Applications



Finding Resources



sense

sensor.TriggeredTemperature

trigger

centigrade

capabilities:
sense.temp

attributes:
country="us"

capabilities:
sense.temp

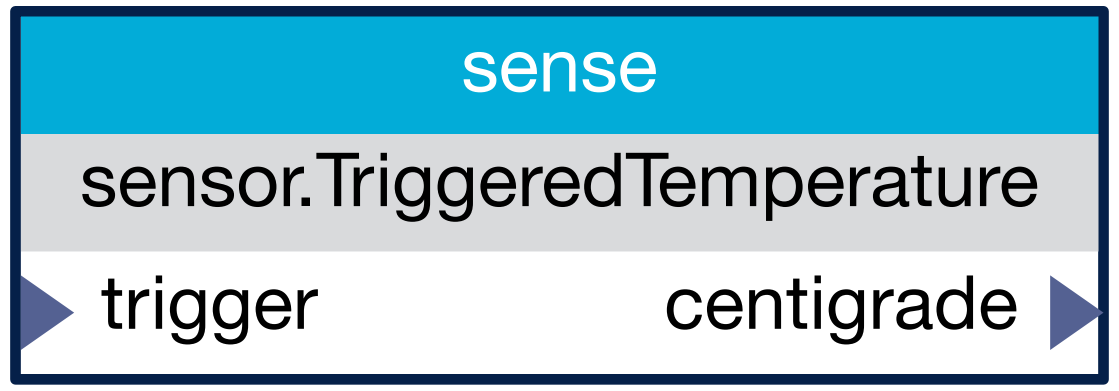
attributes:
country="sweden"

capabilities:
sense.humidity

attributes:
name="us"

Finding Resources

```
apply sense : attr_match(country="us")
```



sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
country="us"

capabilities:
sense.temp

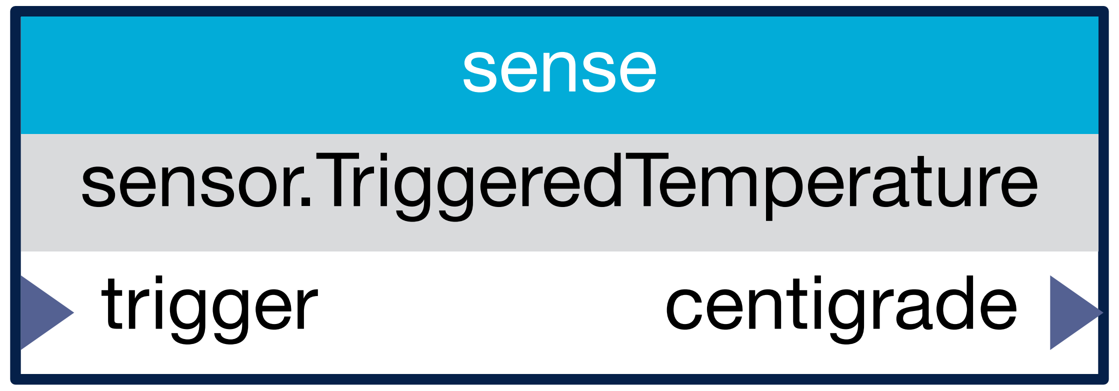
attributes:
country="sweden"

capabilities:
sense.humidity

attributes:
name="us"

Finding Resources

```
apply sense : attr_match(country="us")
```



capabilities:
sense.temp

attributes:
country="us"

capabilities:
sense.temp

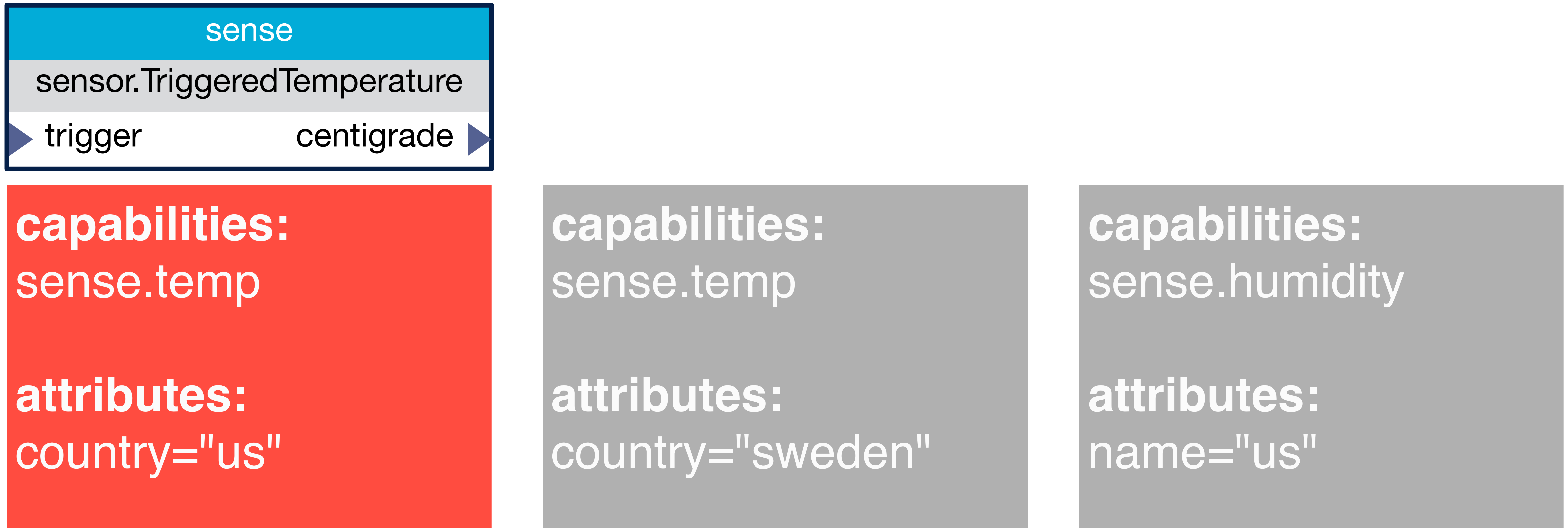
attributes:
country="sweden"

capabilities:
sense.humidity

attributes:
name="us"

Finding Resources

```
apply sense : attr_match(country="us")
```



sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
country="us"

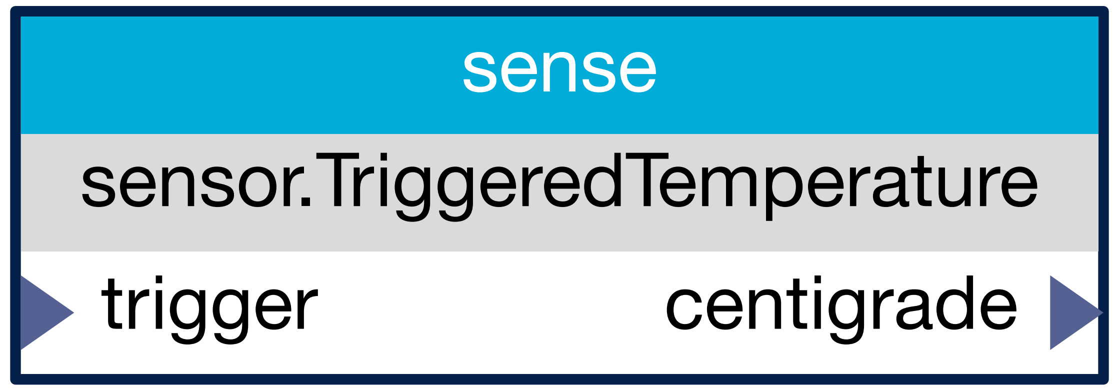
capabilities:
sense.temp

attributes:
country="sweden"

capabilities:
sense.humidity

attributes:
name="us"

Replication and Scaling



capabilities:
sense.temp

attributes:
name="s1"
owner="me"

capabilities:
sense.temp

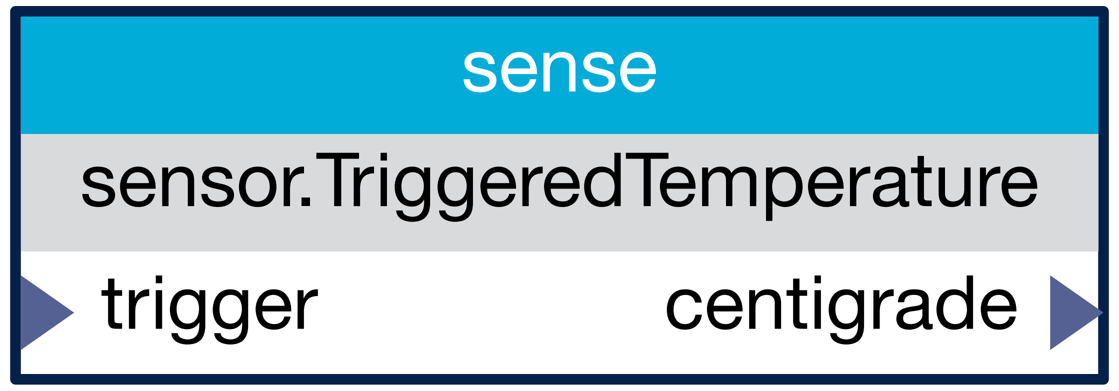
attributes:
name="s2"
owner="me"

capabilities:
sense.temp

attributes:
name="s3"
owner="me"

Replication and Scaling

```
apply sense : device_scaling() & attr_match(owner="me")
```



sense

sensor.TriggeredTemperature

trigger

centigrade

capabilities:
sense.temp

attributes:
name="s1"
owner="me"

capabilities:
sense.temp

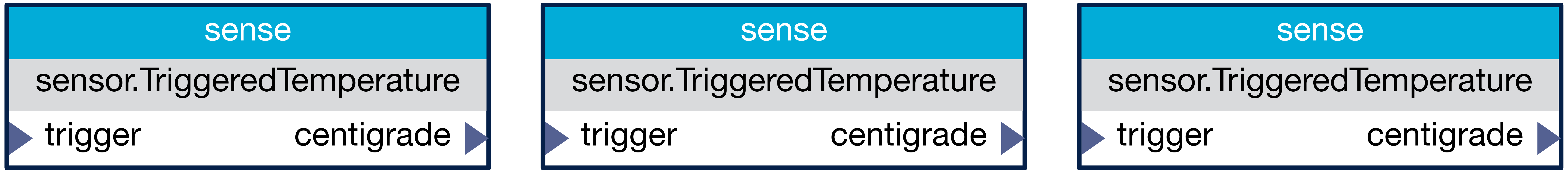
attributes:
name="s2"
owner="me"

capabilities:
sense.temp

attributes:
name="s3"
owner="me"

Replication and Scaling

```
apply sense : device_scaling() & attr_match(owner="me")
```



sense

sensor.TriggeredTemperature

trigger centigrade

sense

sensor.TriggeredTemperature

trigger centigrade

sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s1"
owner="me"

capabilities:
sense.temp

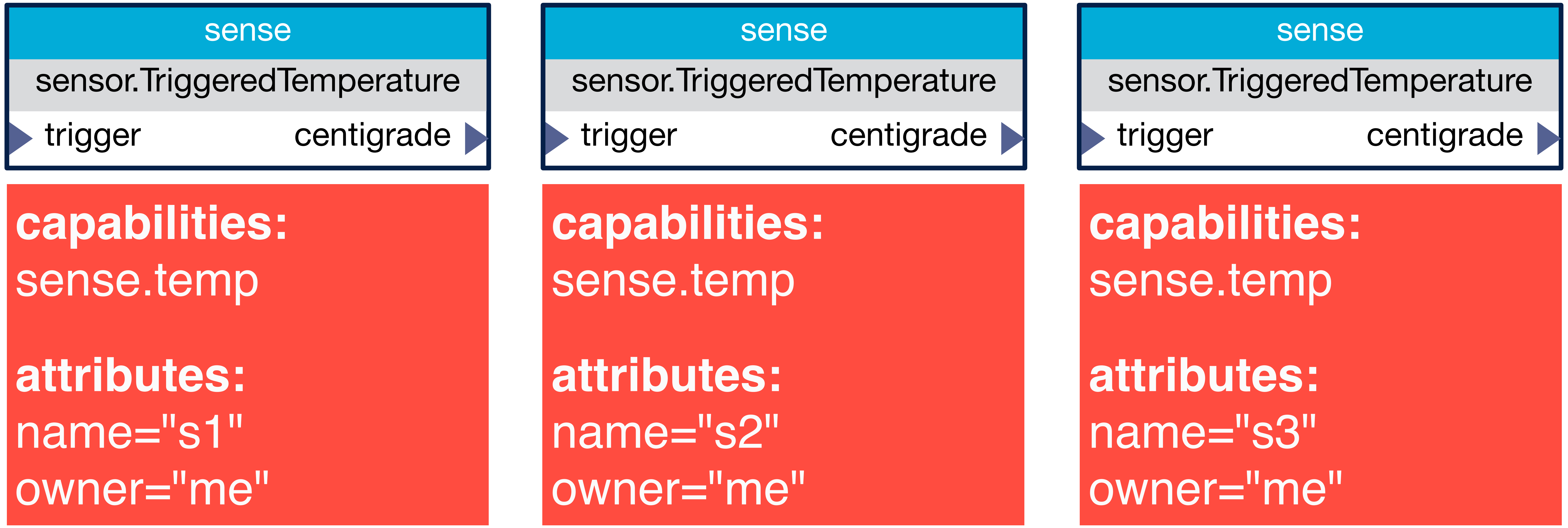
attributes:
name="s2"
owner="me"

capabilities:
sense.temp

attributes:
name="s3"
owner="me"

Replication and Scaling

```
apply sense : device_scaling() & attr_match(owner="me")
```



sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s1"
owner="me"

sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s2"
owner="me"

sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s3"
owner="me"

Replication and Scaling

```
apply sense : device_scaling() & attr_match(owner="me")
```

sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s1"
owner="me"

sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s2"
owner="me"

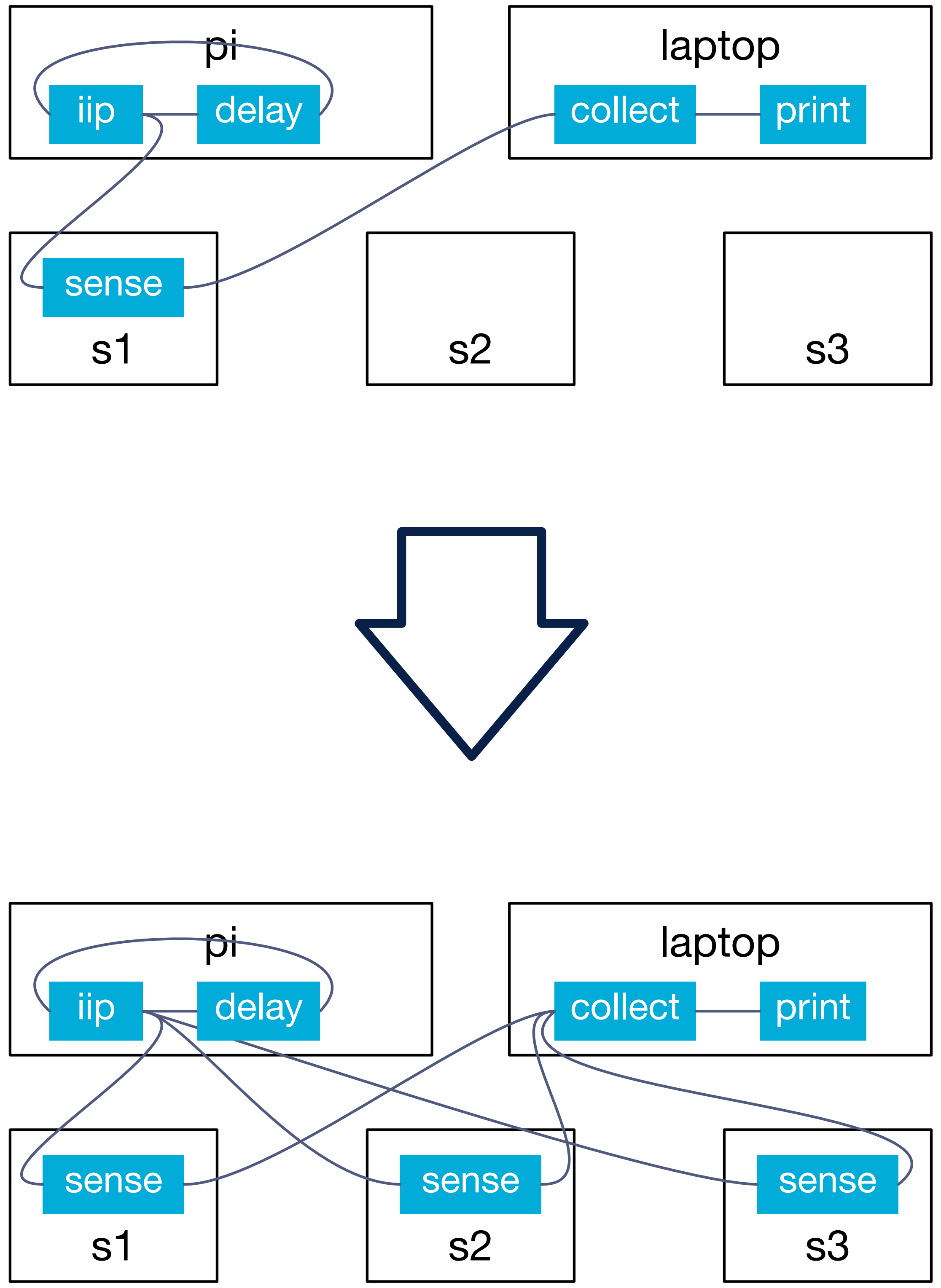
sense

sensor.TriggeredTemperature

trigger centigrade

capabilities:
sense.temp

attributes:
name="s3"
owner="me"



Replication and Scaling

```
apply sense : device_scaling() & attr_match(owner="me")
apply sense : performance_scaling() & attr_match(owner="me")
```

sense
sensor.TriggeredTemperature
trigger centigrade

capabilities:
sense.temp

attributes:
name="s1"
owner="me"

sense
sensor.TriggeredTemperature
trigger centigrade

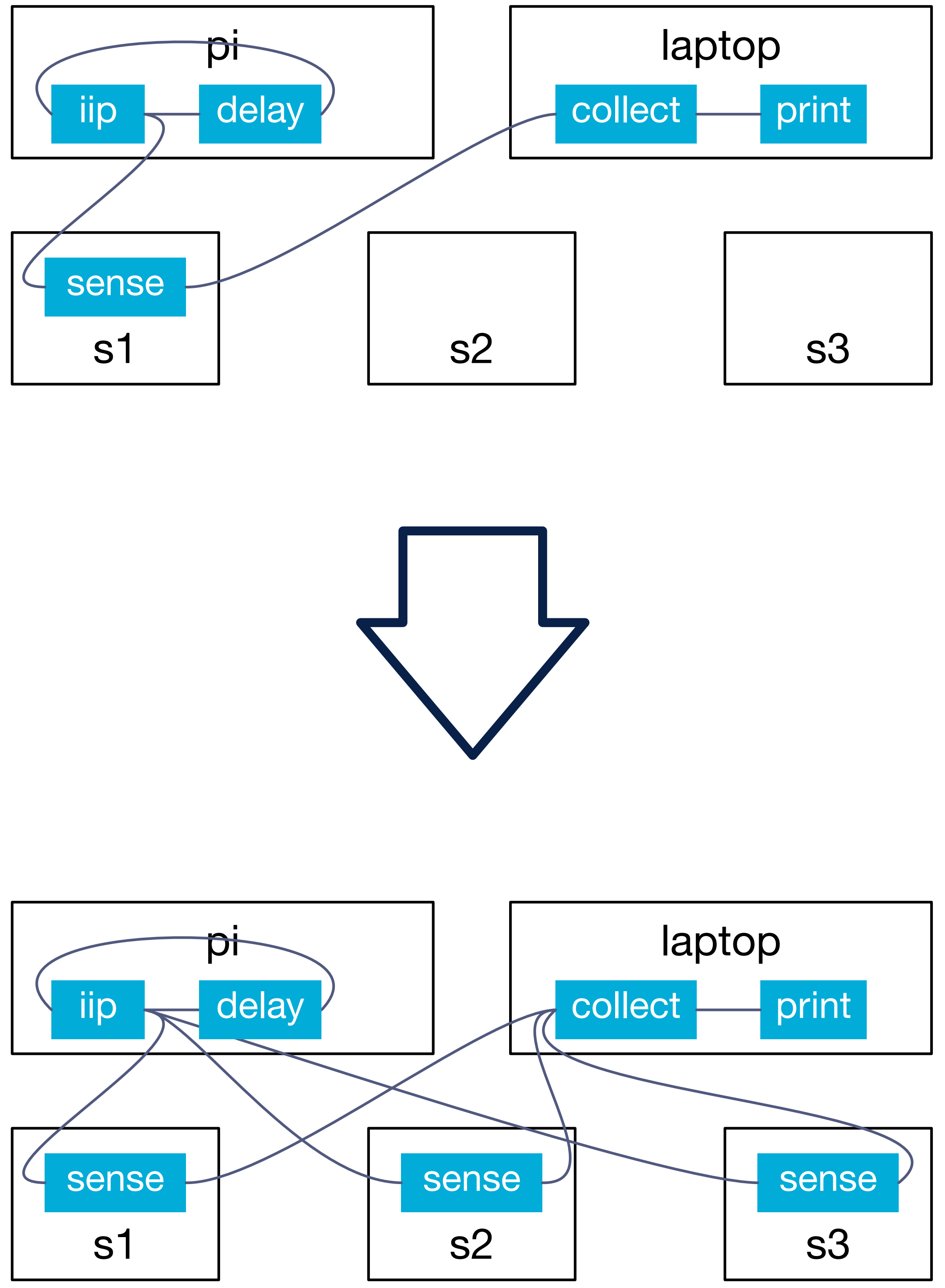
capabilities:
sense.temp

attributes:
name="s2"
owner="me"

sense
sensor.TriggeredTemperature
trigger centigrade

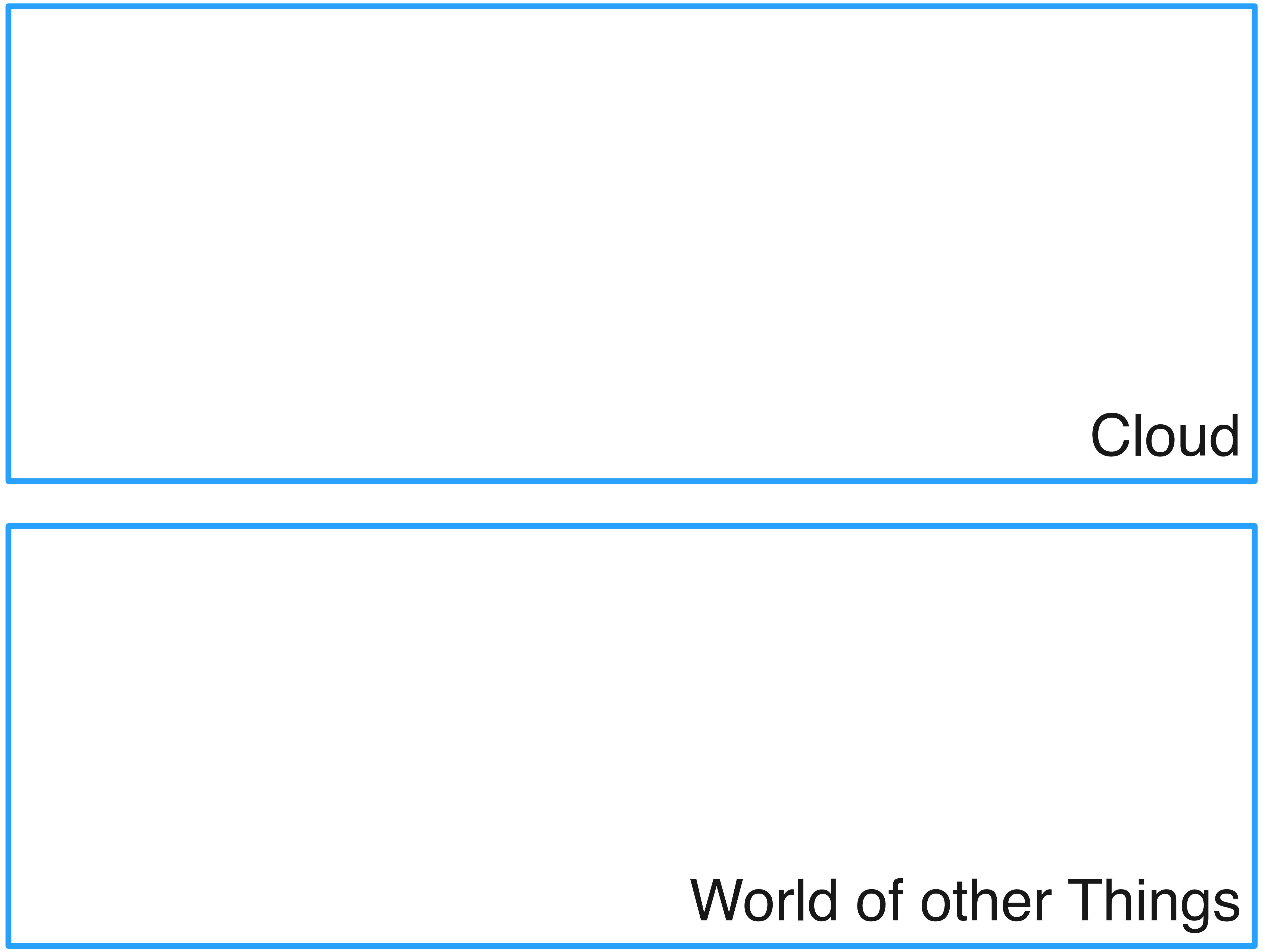
capabilities:
sense.temp

attributes:
name="s3"
owner="me"



Kappa–Serverless IoT

Architecture



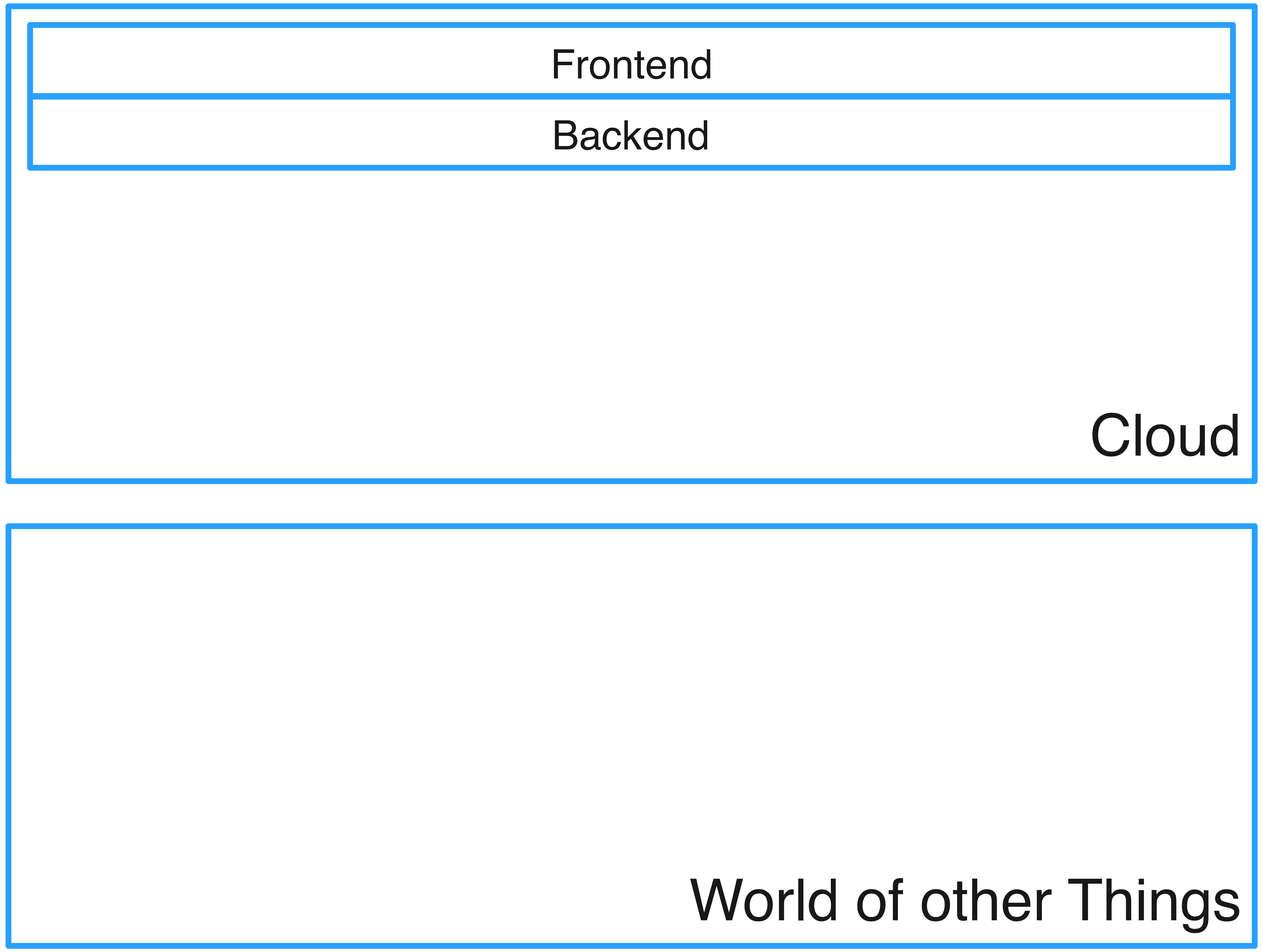
Cloud

World of other Things

Architecture

- Web service with REST API

Command	Meaning
PUT /kappa	Deploy script in body, return reference
POST /kappa/ref	Send JSON data to kappa inport
GET /kappa/ref	Read JSON data from kappa outport
DELETE /kappa/ref	Delete a kappa
GET /kappa	List all kappas



Frontend

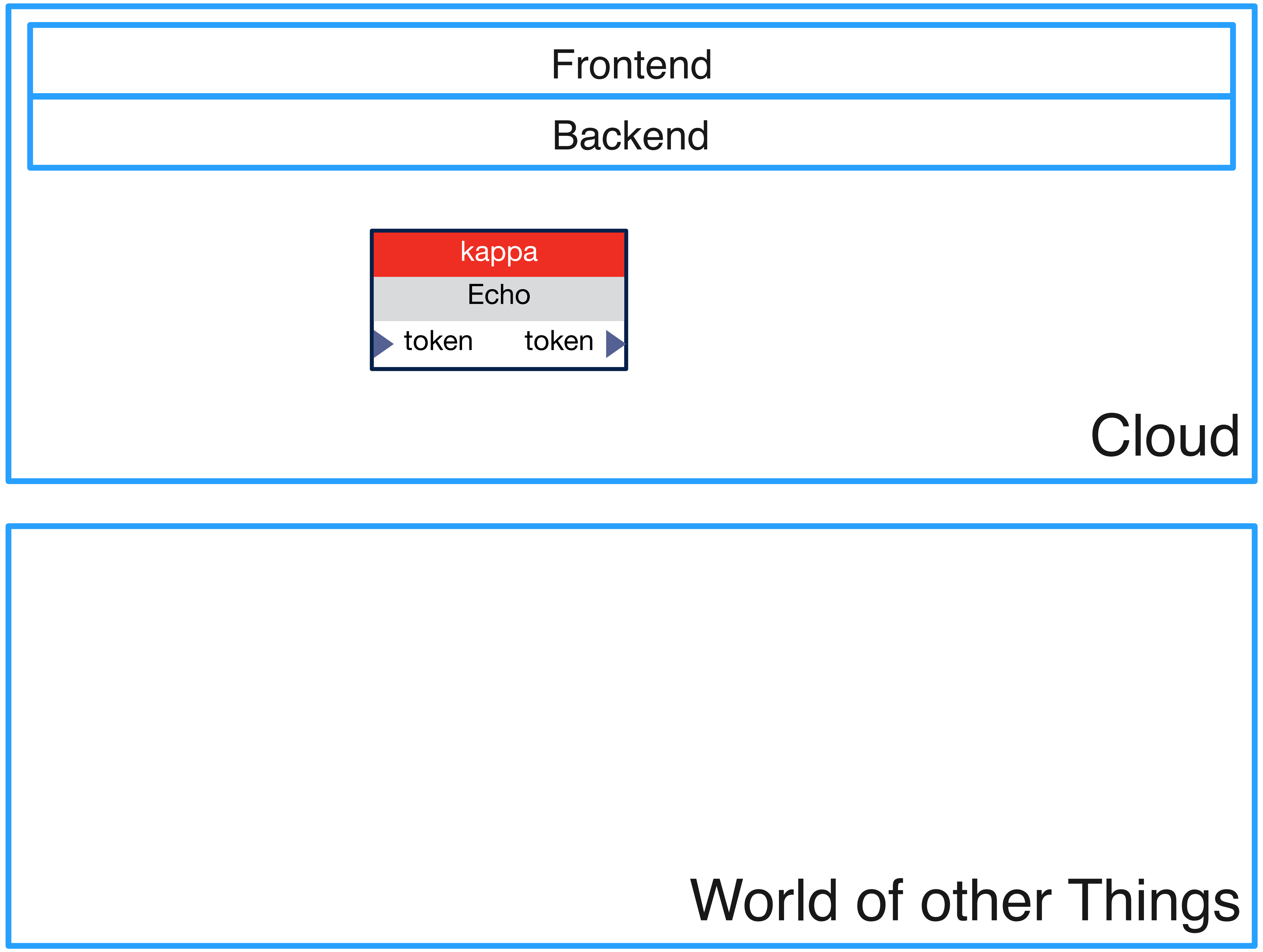
Backend

Cloud

World of other Things

Architecture

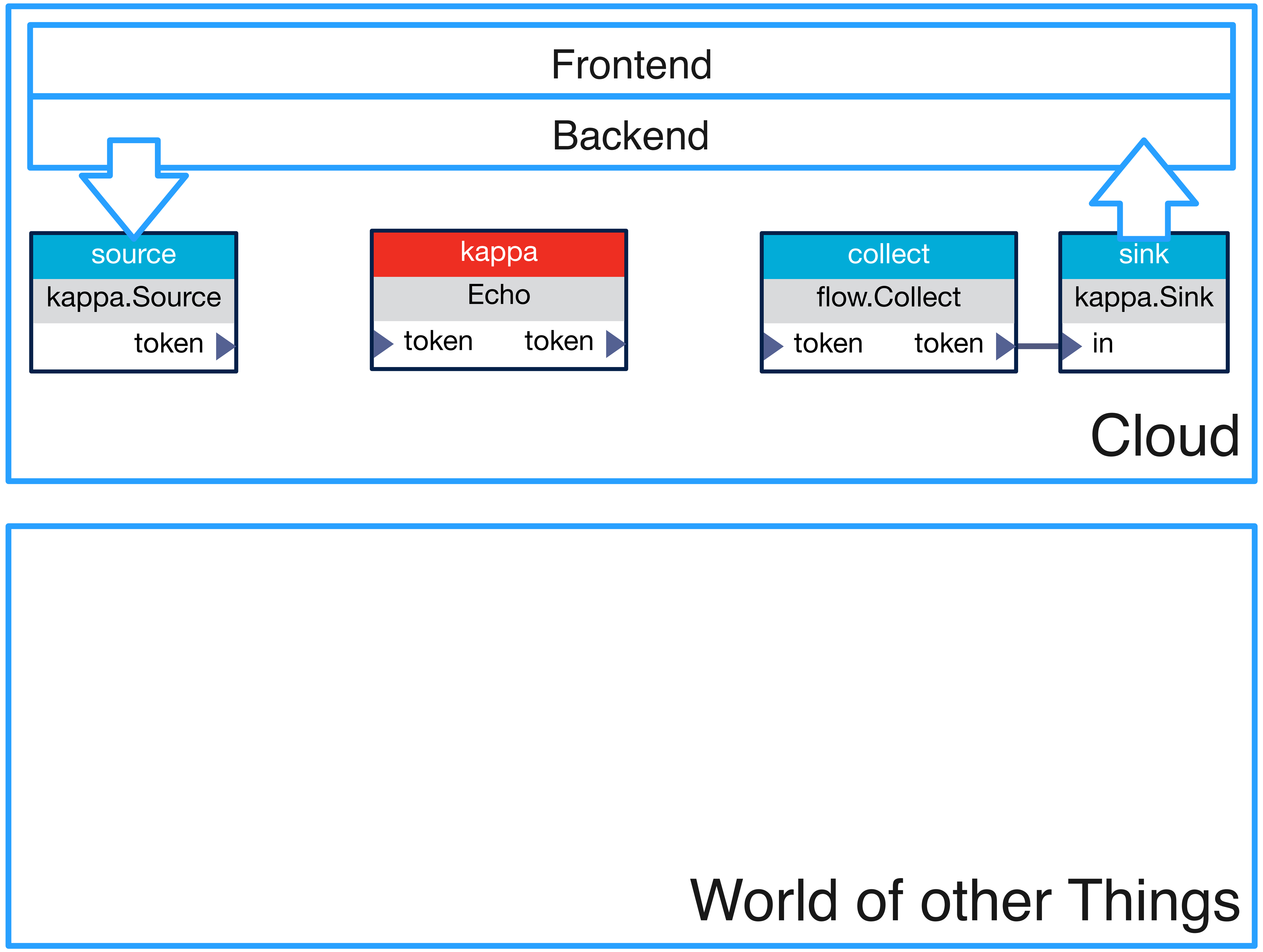
- Web service with REST API
- FaaS script defined as a component



Command	Meaning
PUT /kappa	Deploy script in body, return reference
POST /kappa/ref	Send JSON data to kappa inport
GET /kappa/ref	Read JSON data from kappa outport
DELETE /kappa/ref	Delete a kappa
GET /kappa	List all kappas

Architecture

- Web service with REST API
- FaaS script defined as a component
- Runtime with two special actors as bridge to API
  - kappa.Source
  - kappa.Sink

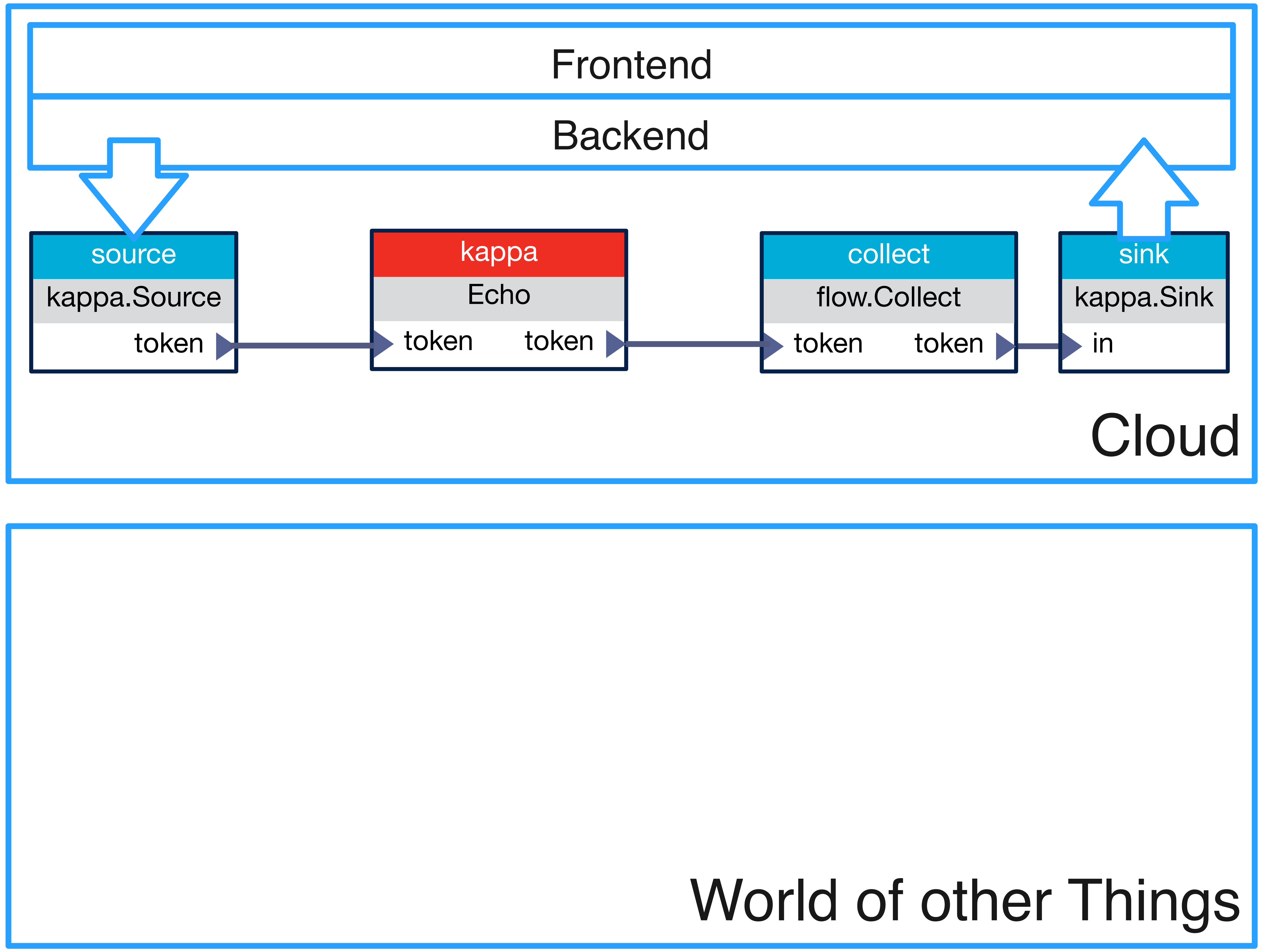


Command	Meaning
PUT /kappa	Deploy script in body, return reference
POST /kappa/ref	Send JSON data to kappa inport
GET /kappa/ref	Read JSON data from kappa outport
DELETE /kappa/ref	Delete a kappa
GET /kappa	List all kappas

Architecture

- Web service with REST API
- FaaS script defined as a component
- Runtime with two special actors as bridge to API
  - kappa.Source
  - kappa.Sink
- Wrap component in source, collect, and sink

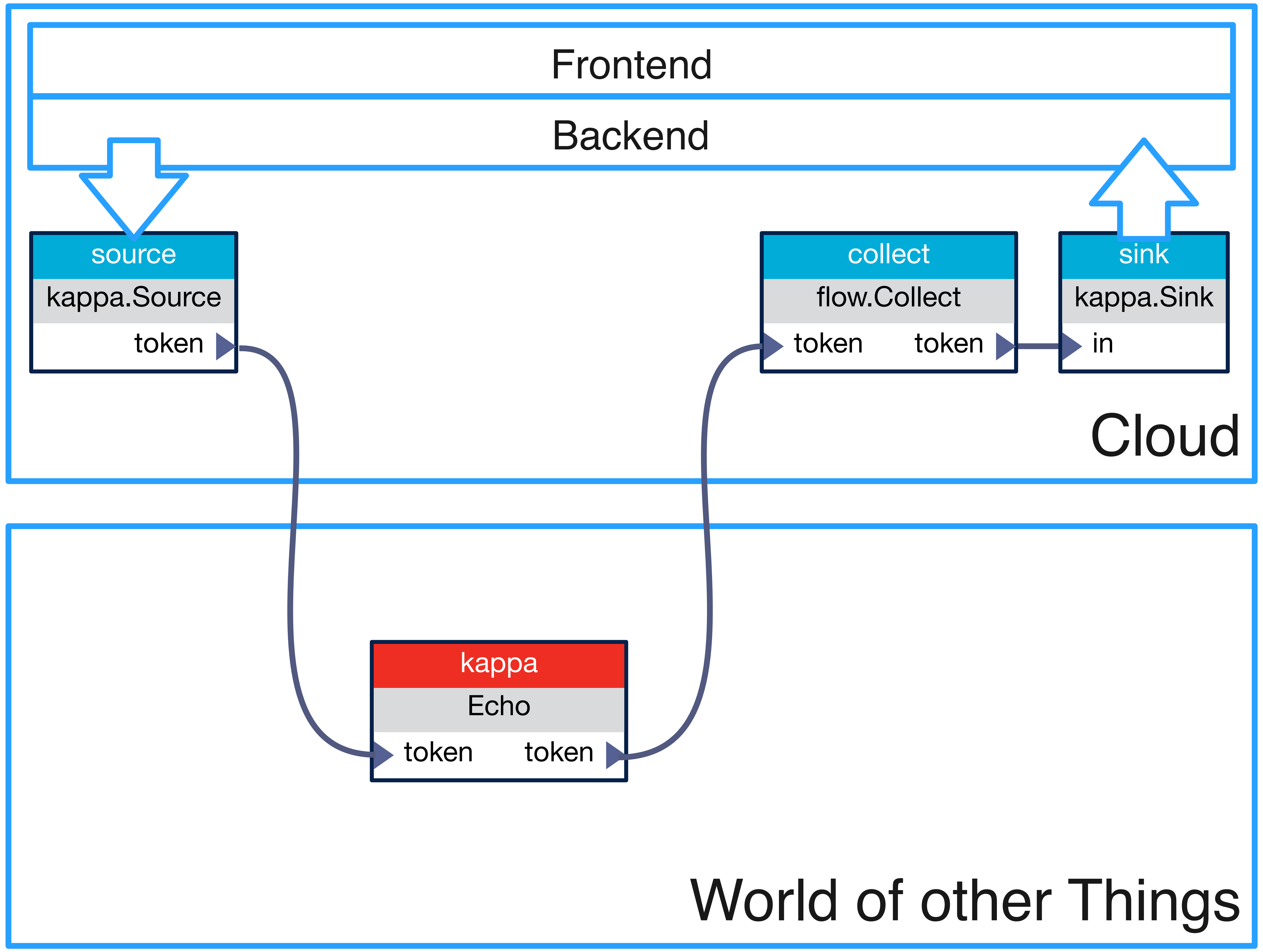
Command	Meaning
PUT /kappa	Deploy script in body, return reference
POST /kappa/ref	Send JSON data to kappa inport
GET /kappa/ref	Read JSON data from kappa outport
DELETE /kappa/ref	Delete a kappa
GET /kappa	List all kappas



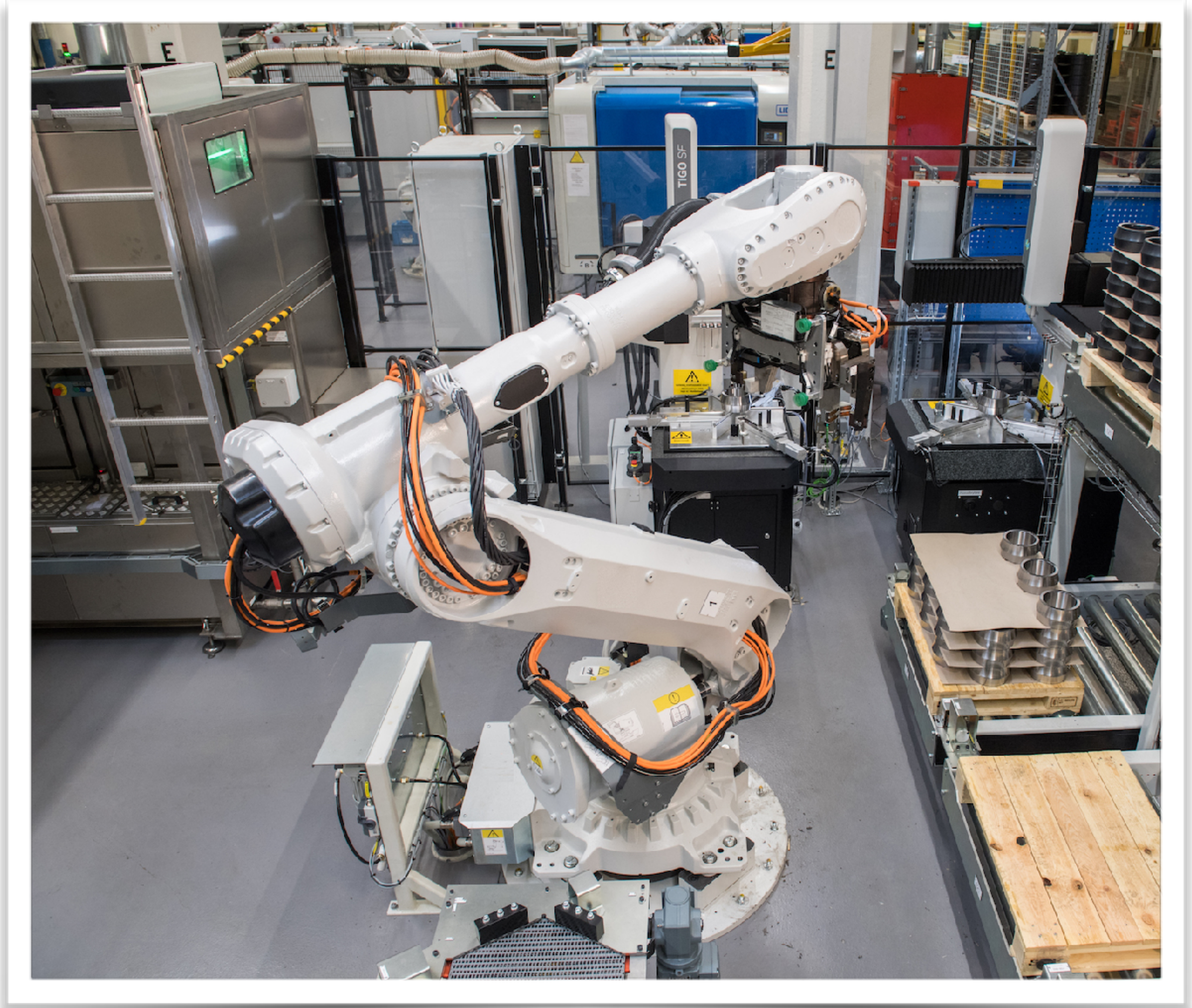
Architecture

- Web service with REST API
- FaaS script defined as a component
- Runtime with two special actors as bridge to API
  - kappa.Source
  - kappa.Sink
- Wrap component in source, collect, and sink
- Deploy, communicate with, and destroy kappa

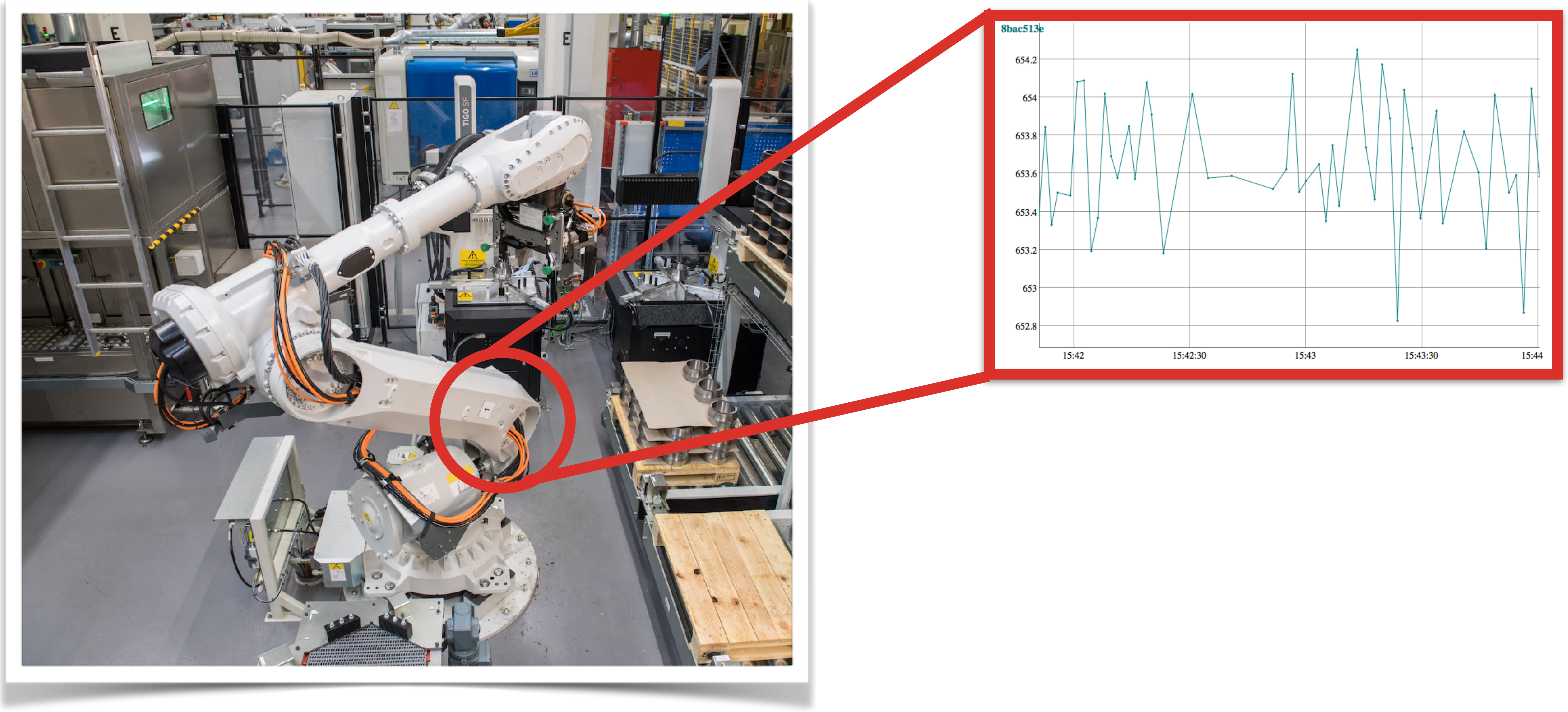
Command	Meaning
PUT /kappa	Deploy script in body, return reference
POST /kappa/ref	Send JSON data to kappa inport
GET /kappa/ref	Read JSON data from kappa outport
DELETE /kappa/ref	Delete a kappa
GET /kappa	List all kappas



Example Use Cases

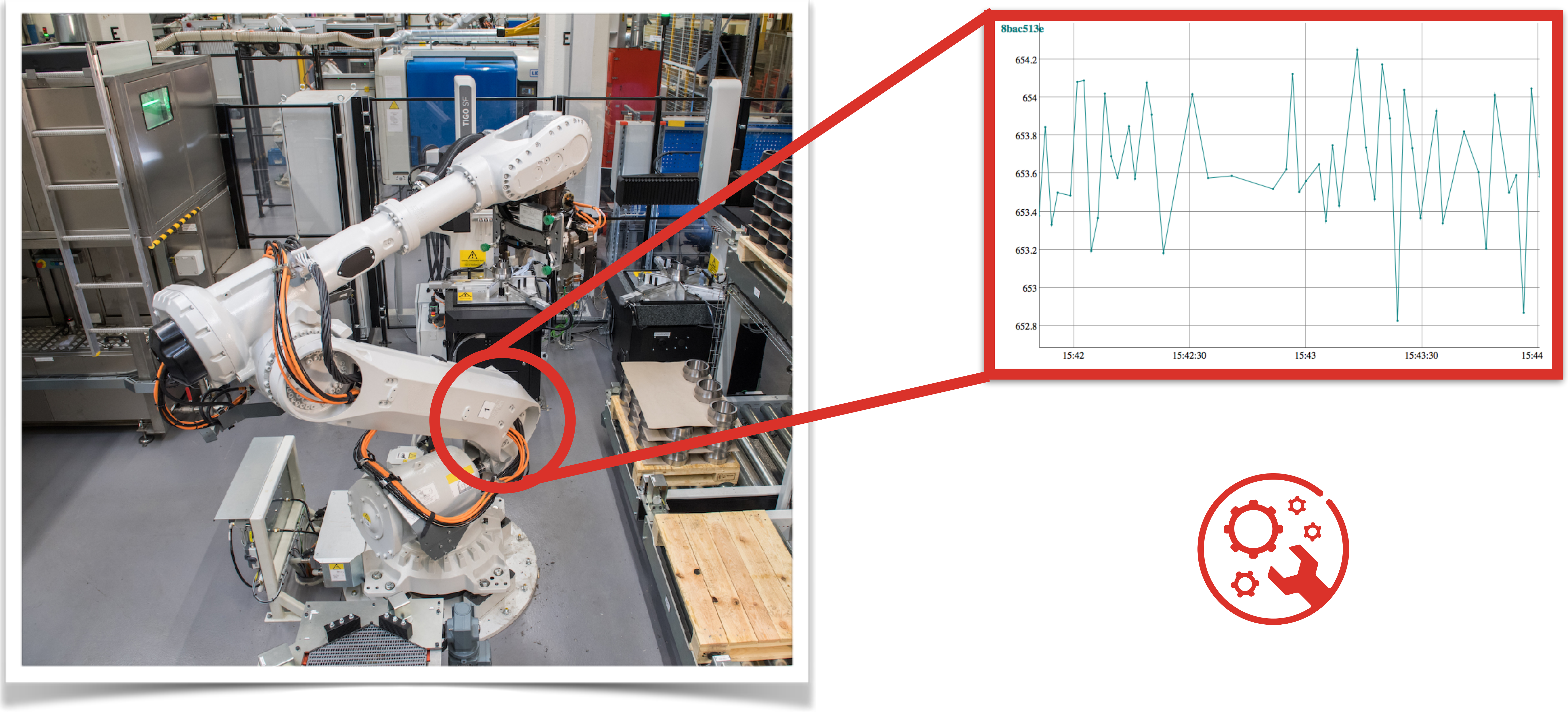


Example Use Cases



Introspection
Visualize running system, log-on-condition, etc.

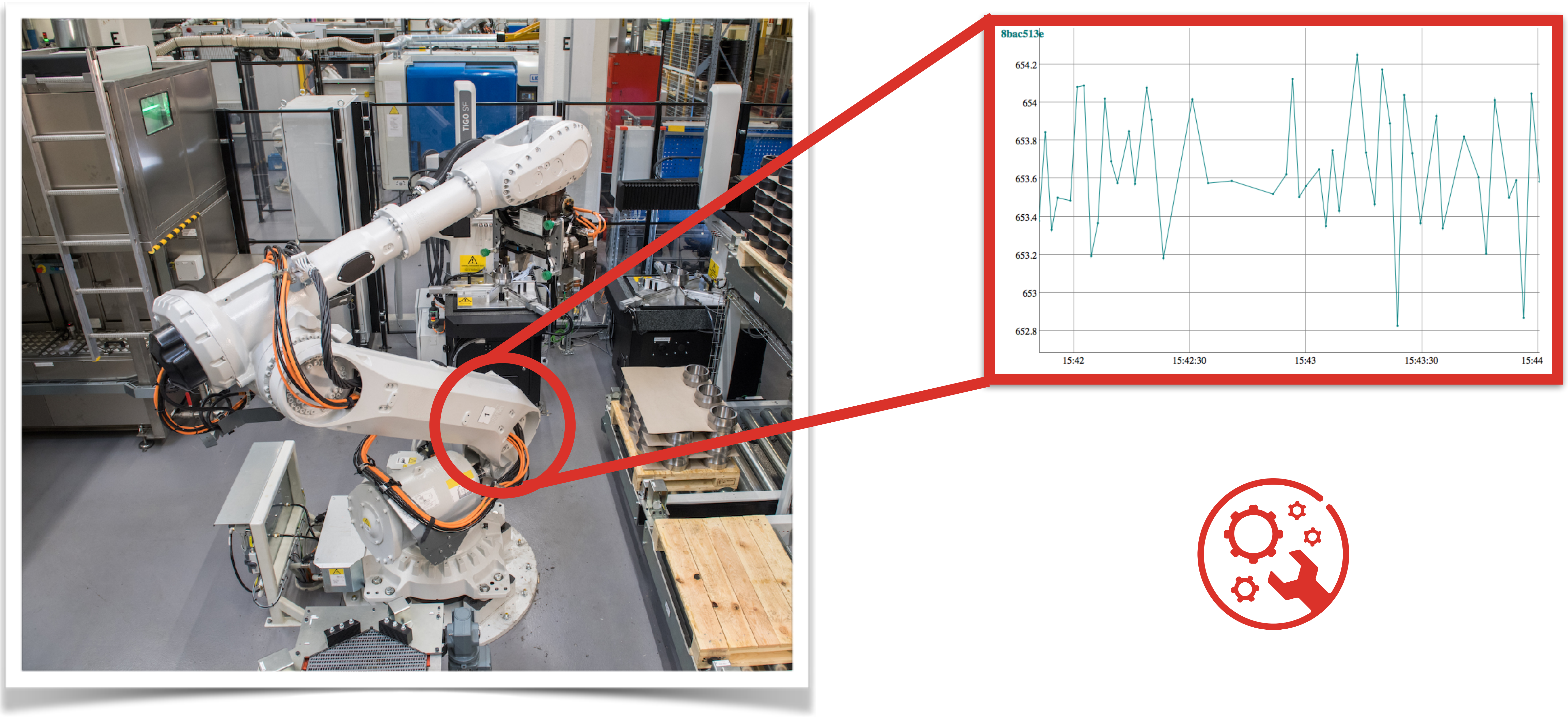
Example Use Cases



Introspection
Visualize running system, log-on-condition, etc.

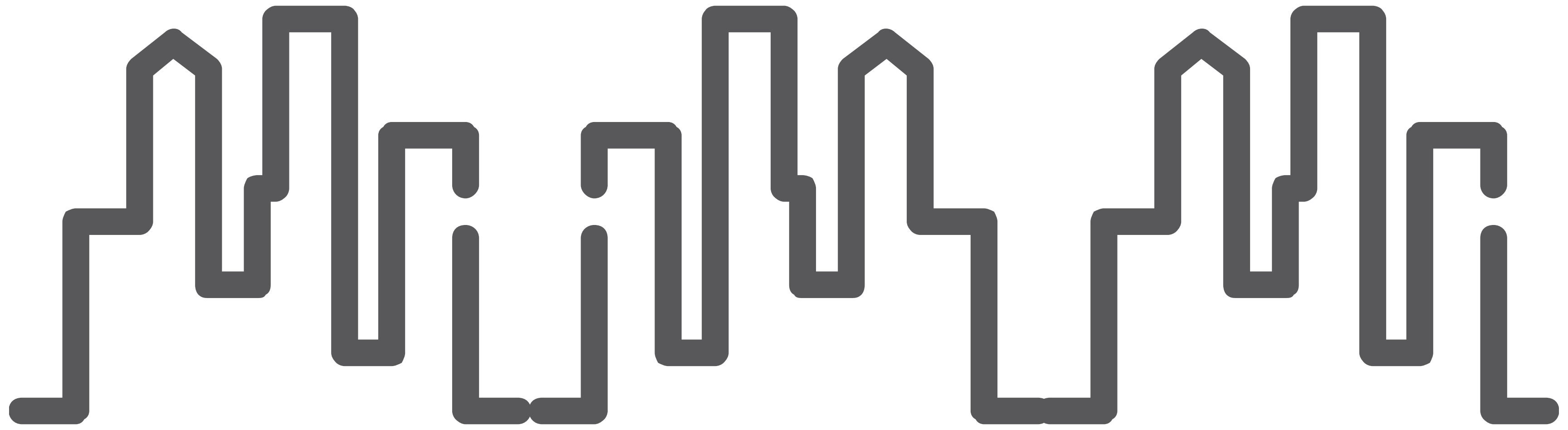
µService creation
Deploy local control loops, use API to set desired state

Example Use Cases



Introspection
Visualize running system, log-on-condition, etc.

μService creation
Deploy local control loops, use API to set desired state



Large area sampling
Statistically sound sampling of data without violating privacy (e.g. indoor temp)

Serverless IoT Revisited

If you can't be a first class cloud citizen,
at least be a first class neighbour