WHY ARE SMART BUILDINGS STILL DUMB?

A research agenda towards secure, resilient and evolvable smart buildings

Karolina Skrivankova, Mark Handley, Steve Hailes

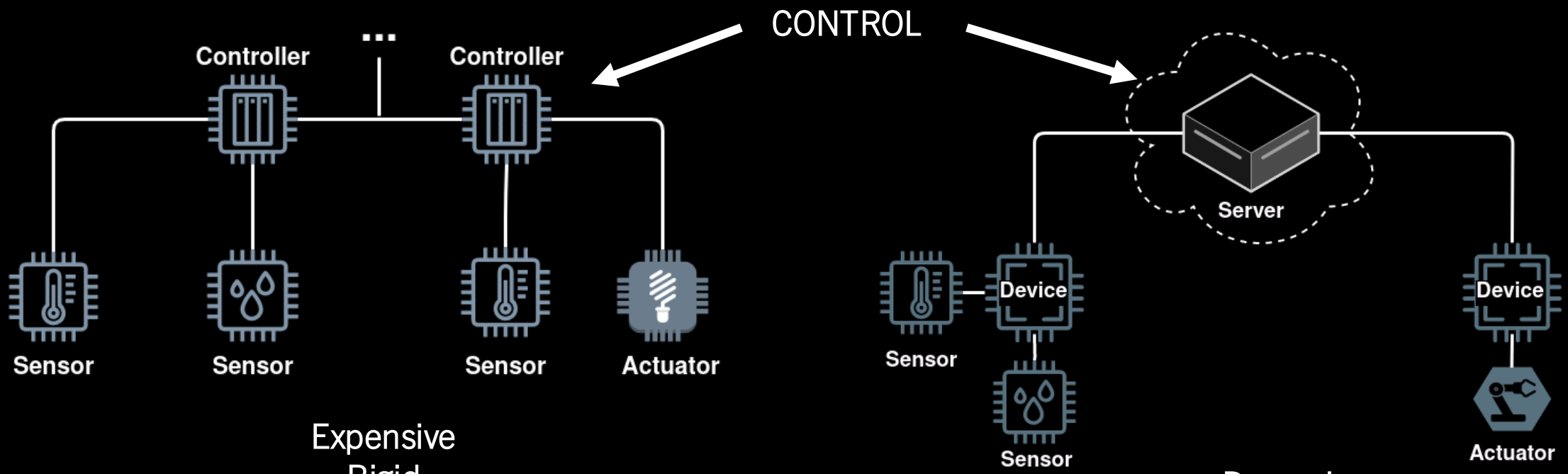
WHAT WOULD THE FUTURE SMART BUILDING SUPPORT?

- Frequent layout changes with integrated robotics
- Custom HVAC zones
- Low maintenance costs
- Energy efficiency

WHAT WOULD THE FUTURE SMART BUILDING SUPPORT?

- Frequent layout changes with integrated robotics
- Custom HVAC zones
- Low maintenance costs
- Energy efficiency
- Not turn the lights off when I've been sitting at my desk for 10 minutes
- Integrate CO_2 sensors during a pandemic

BUILDING AUTOMATION TODAY



Expensive
Rigid
High maintenance
Reliable

Dynamic
High maintenance
Unreliable

Rudimentary control policies

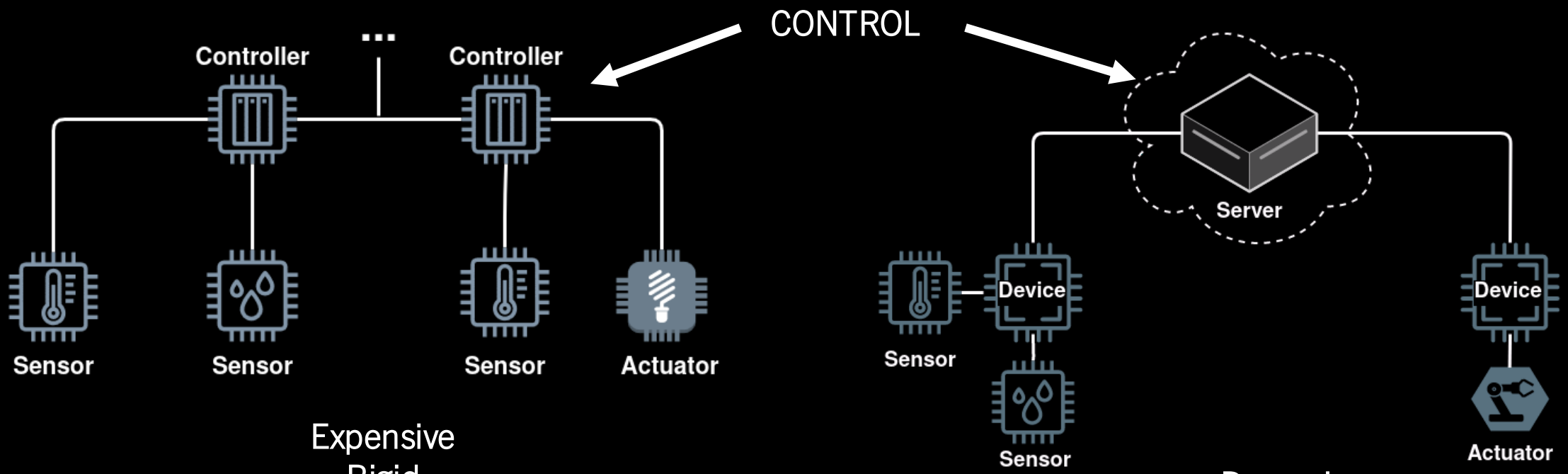
BUILDING AUTOMATION TODAY

DEPENDABILITY 

 SCALE

COST 

BUILDING AUTOMATION TODAY

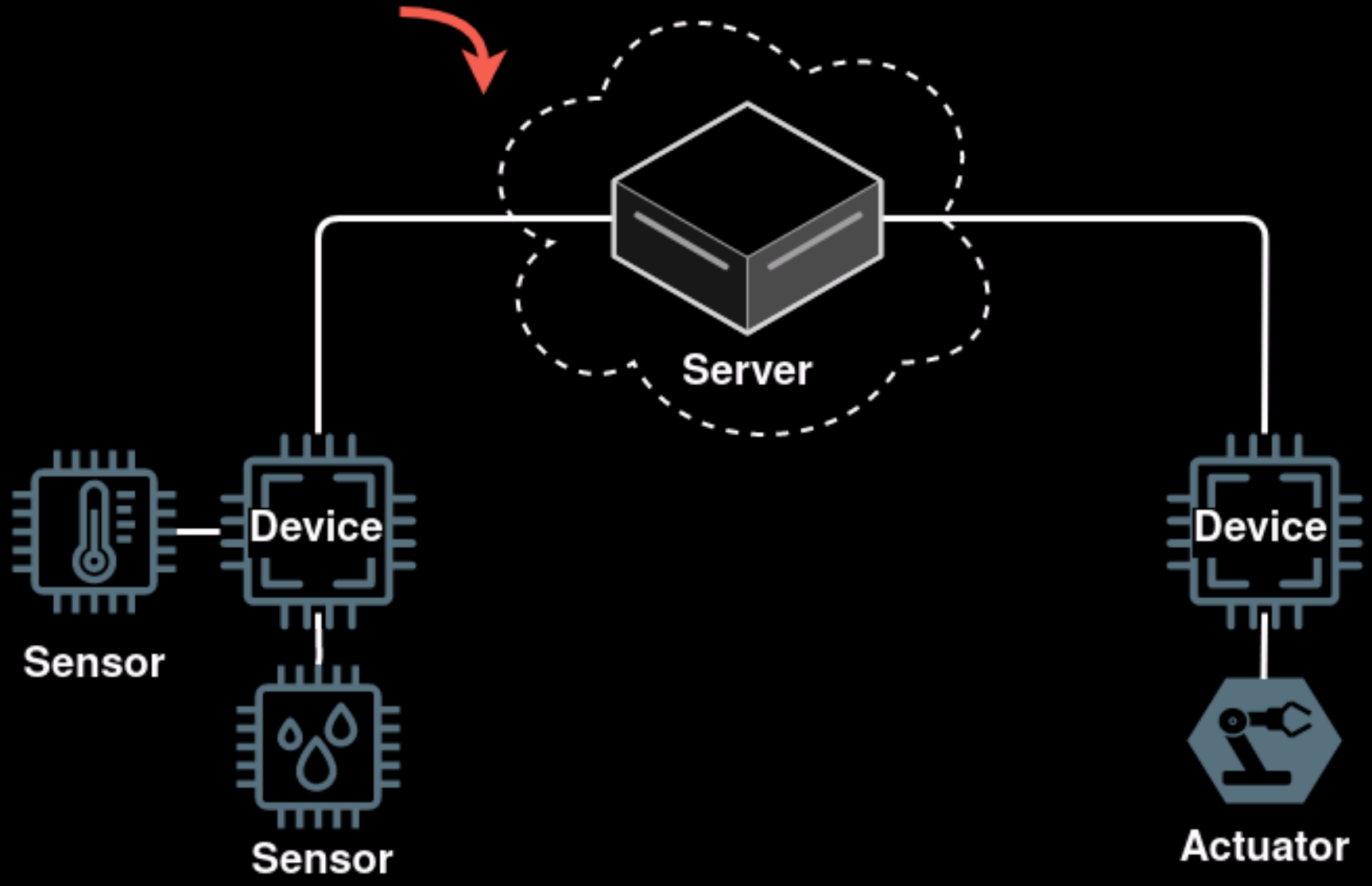


Expensive
Rigid
High maintenance
Reliable

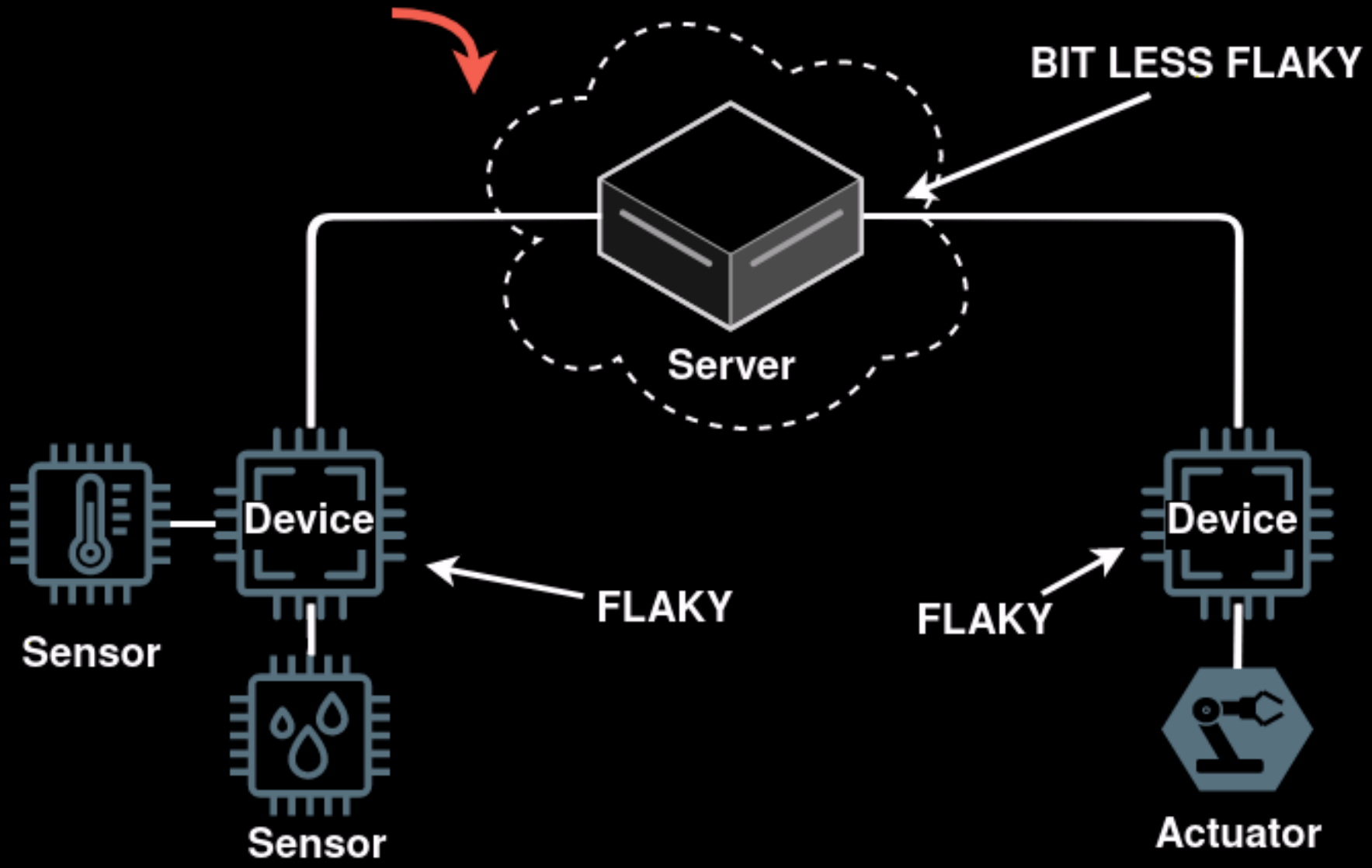
Dynamic
High maintenance
Unreliable

Rudimentary control policies

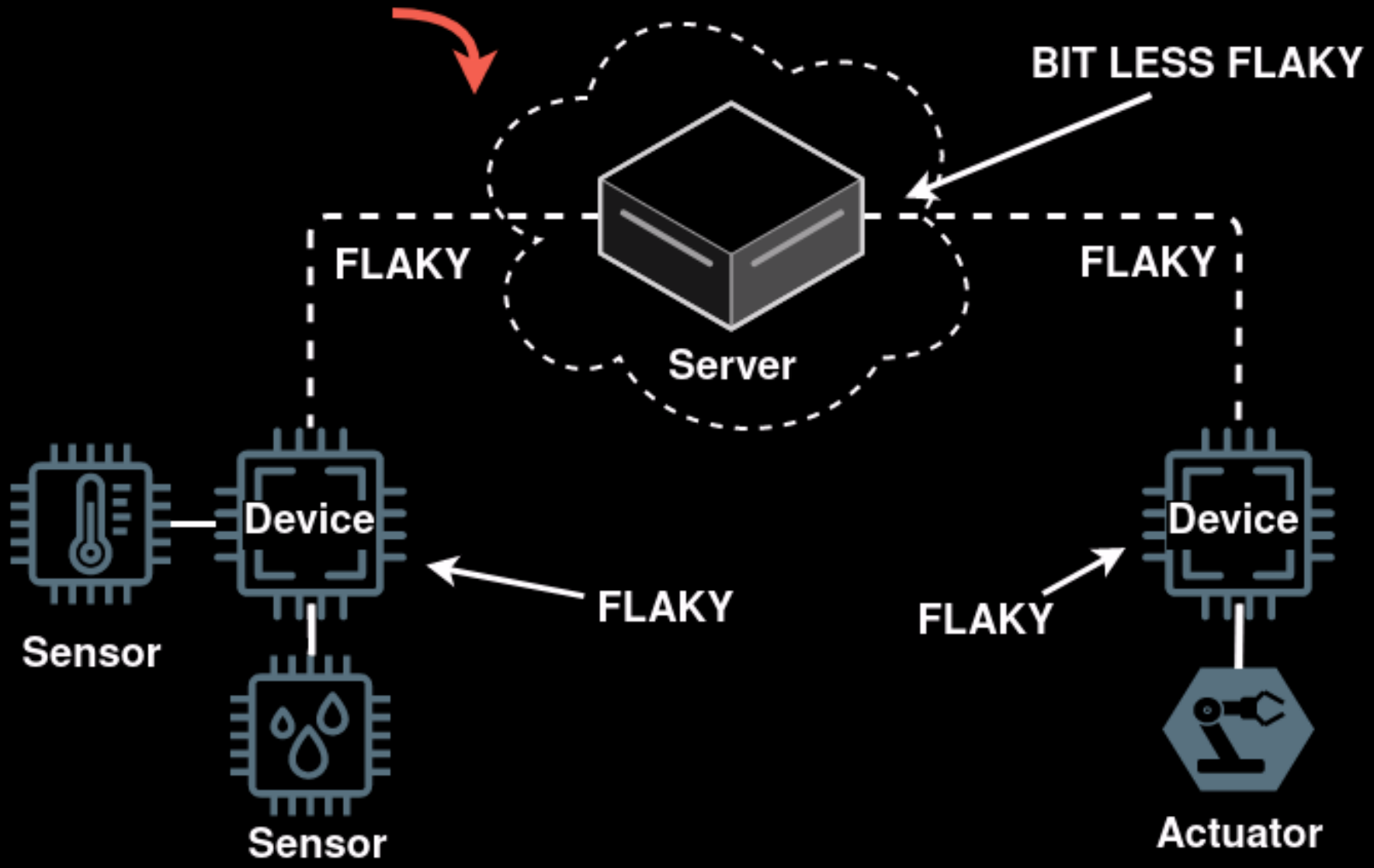
RUN A COMPLEX CONTROL POLICY FOR DECADES

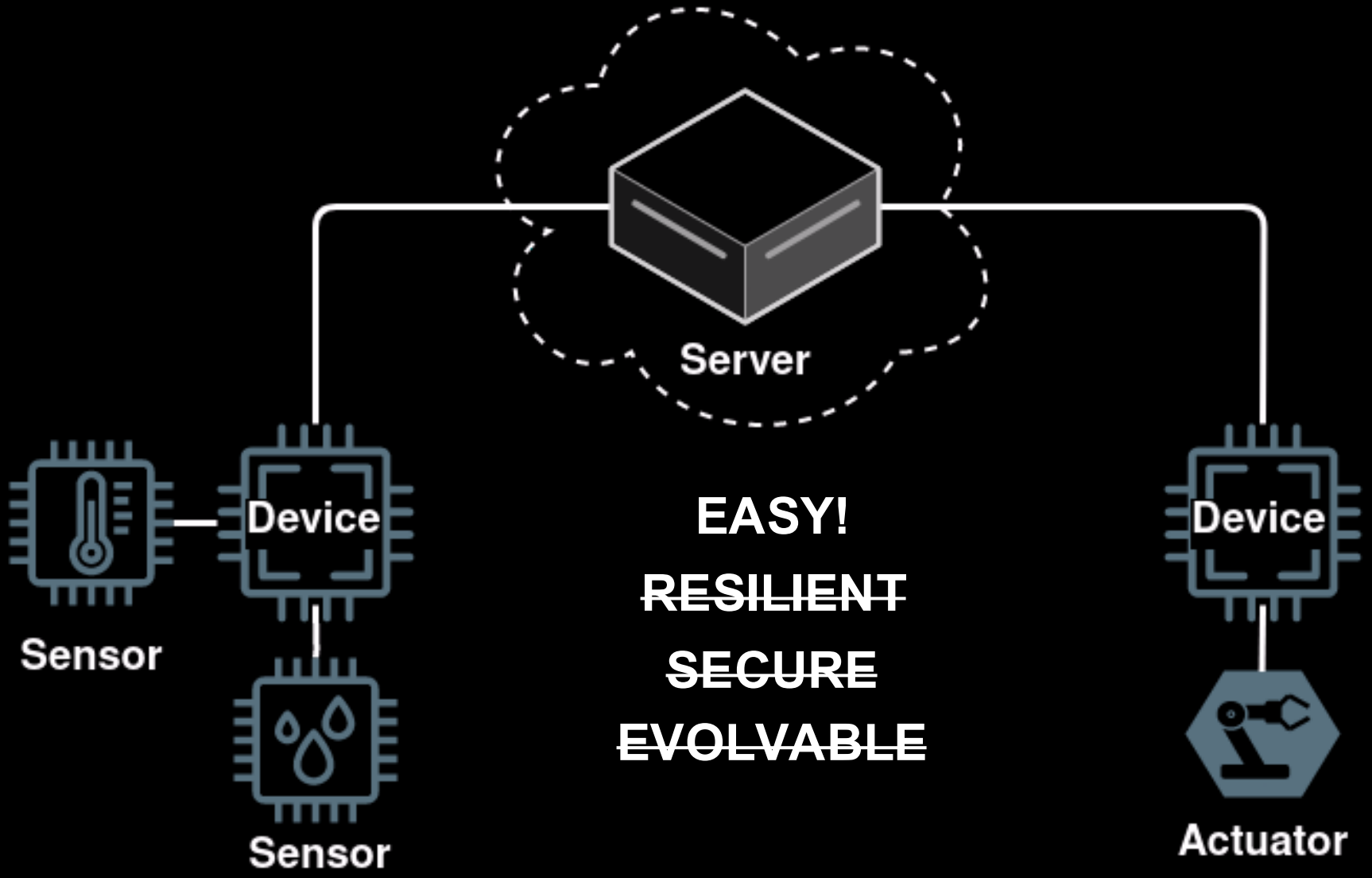


RUN A COMPLEX CONTROL POLICY FOR DECADES



RUN A COMPLEX CONTROL POLICY FOR DECADES



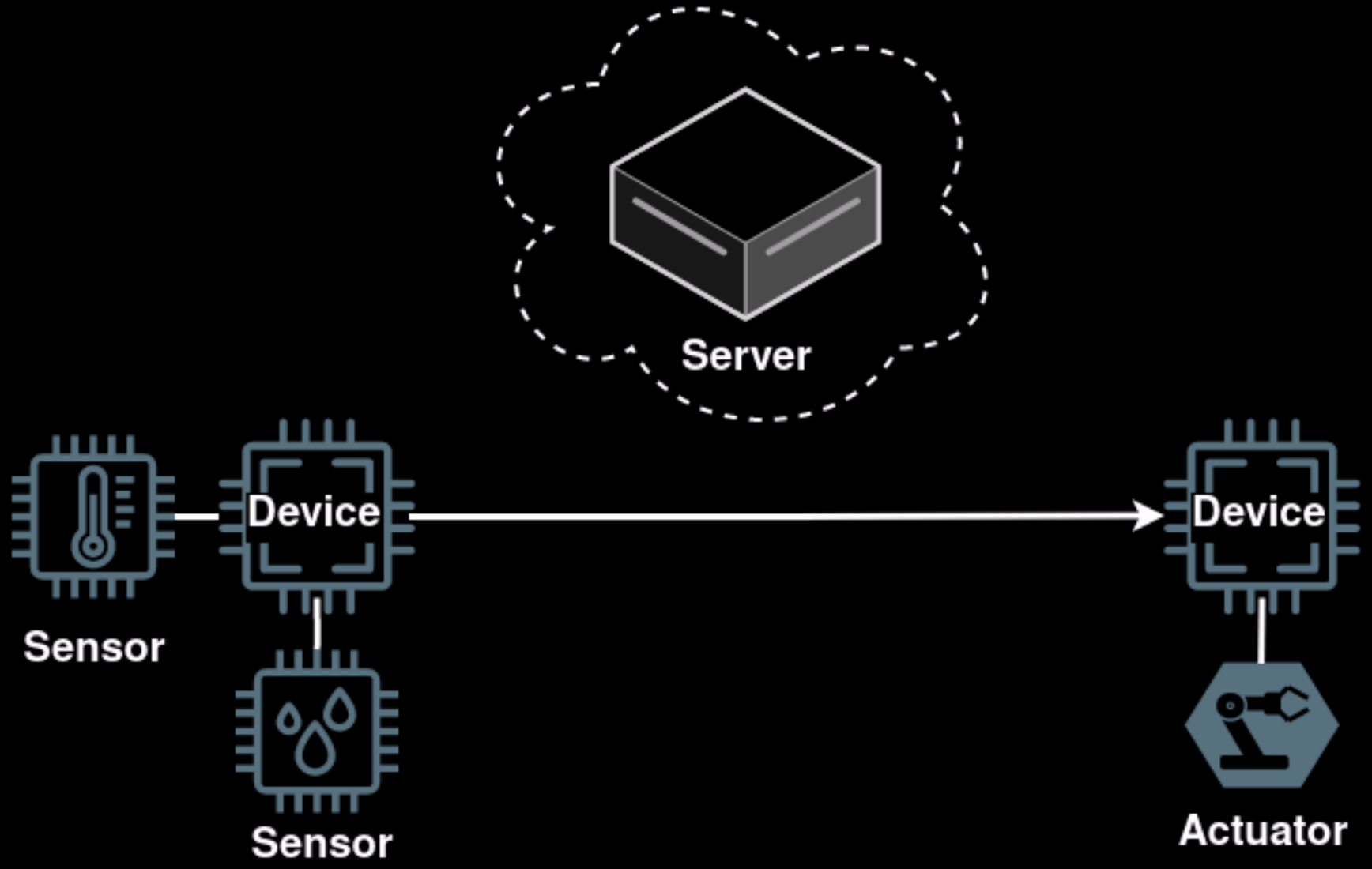


EASY!

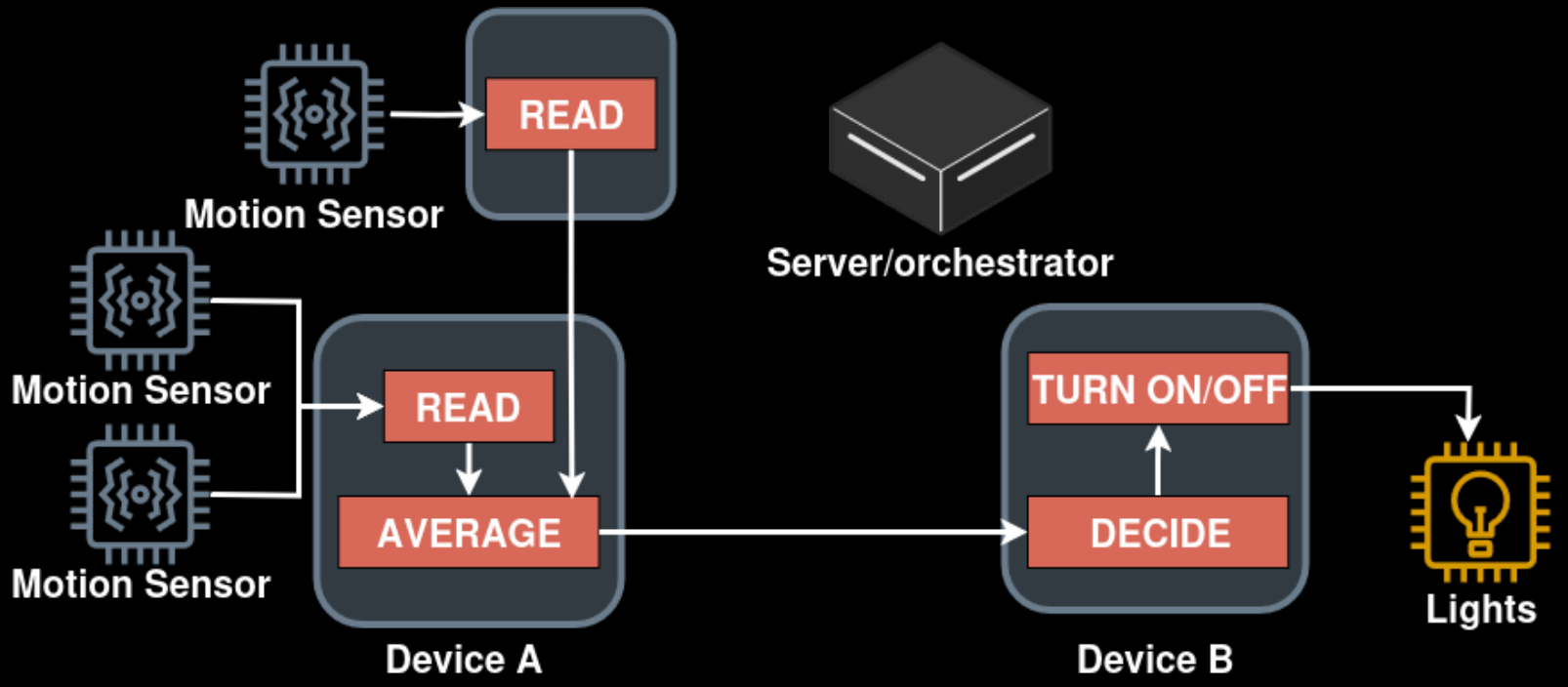
~~**RESILIENT**~~

~~**SECURE**~~

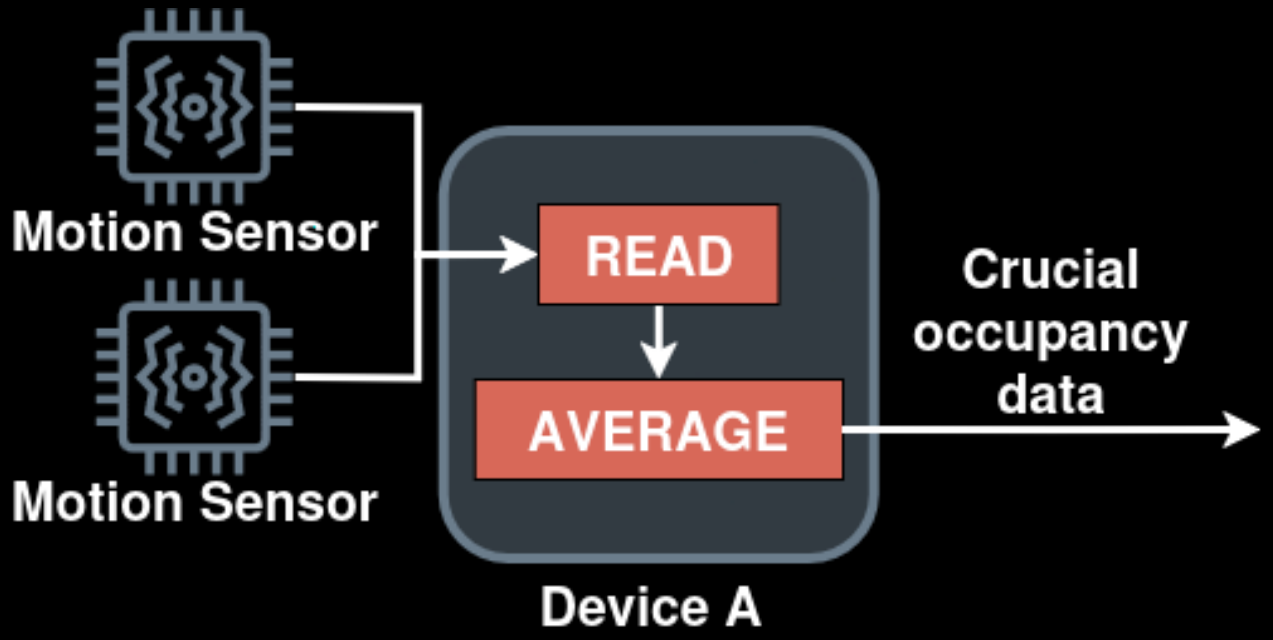
~~**EVOLVABLE**~~


Server
Device
Sensor
Sensor
Device
Actuator

EXAMPLE CONTROL SYSTEM

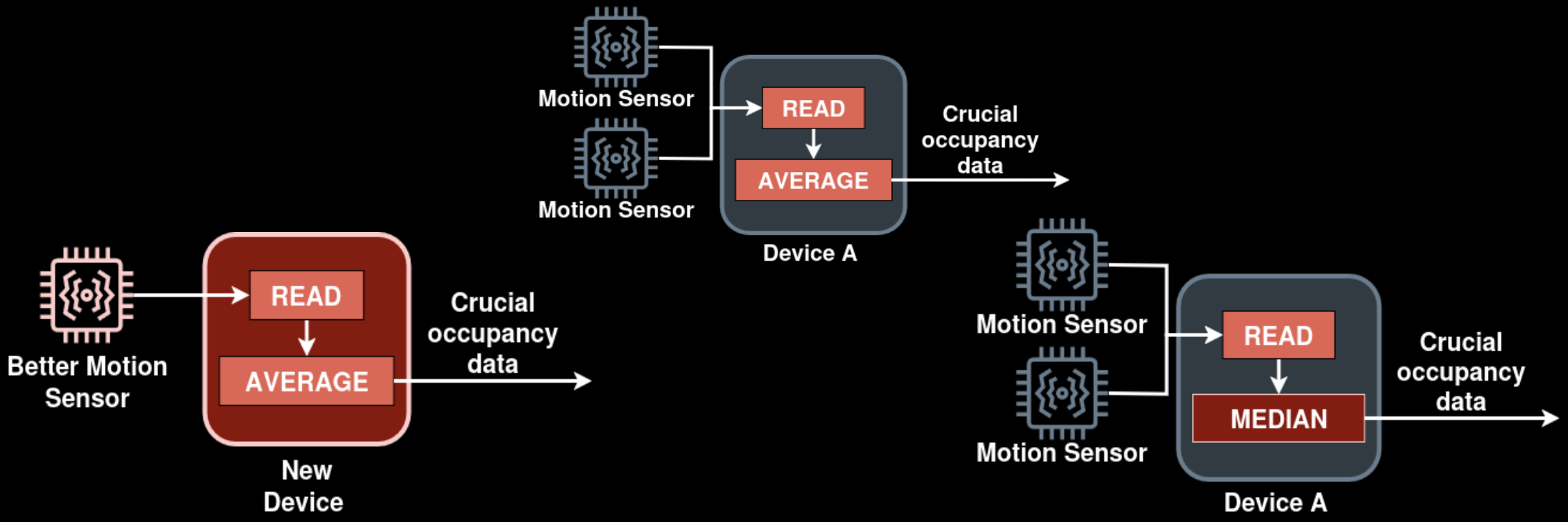


WHAT THE CONTROL SYSTEM GETS FROM A DEVICE



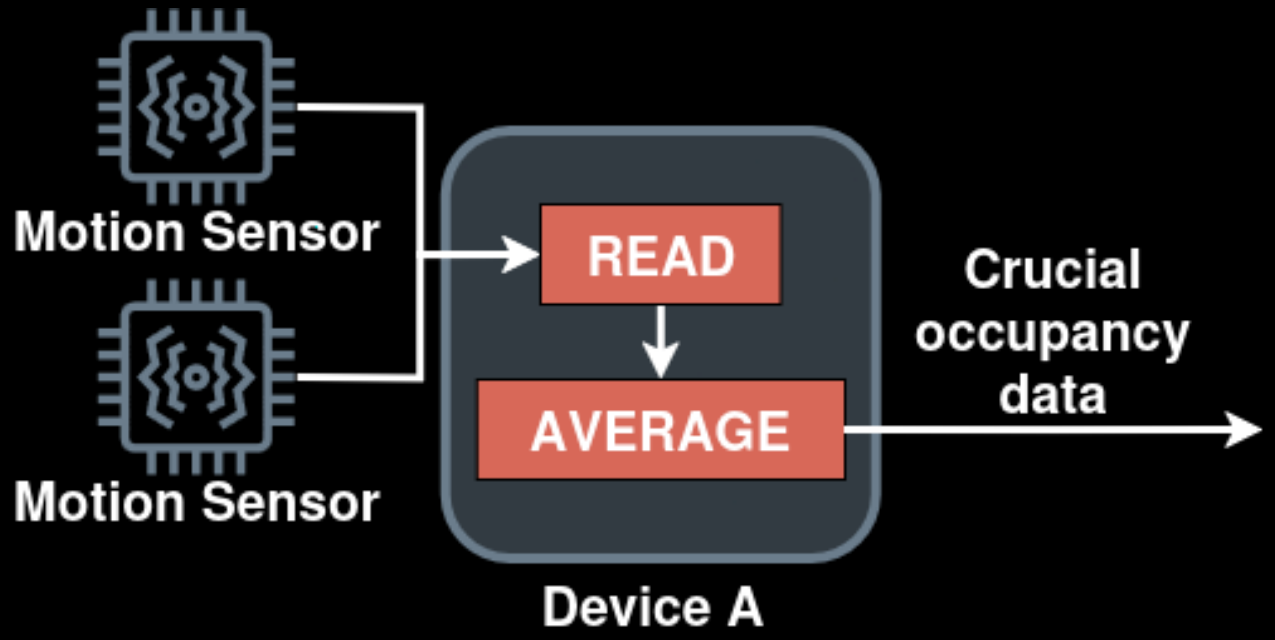
1. CONTROL COMPUTE
2. DATA SOURCES/SINKS

SUPPORTING SYSTEM CHANGES ON DEVICE LEVEL



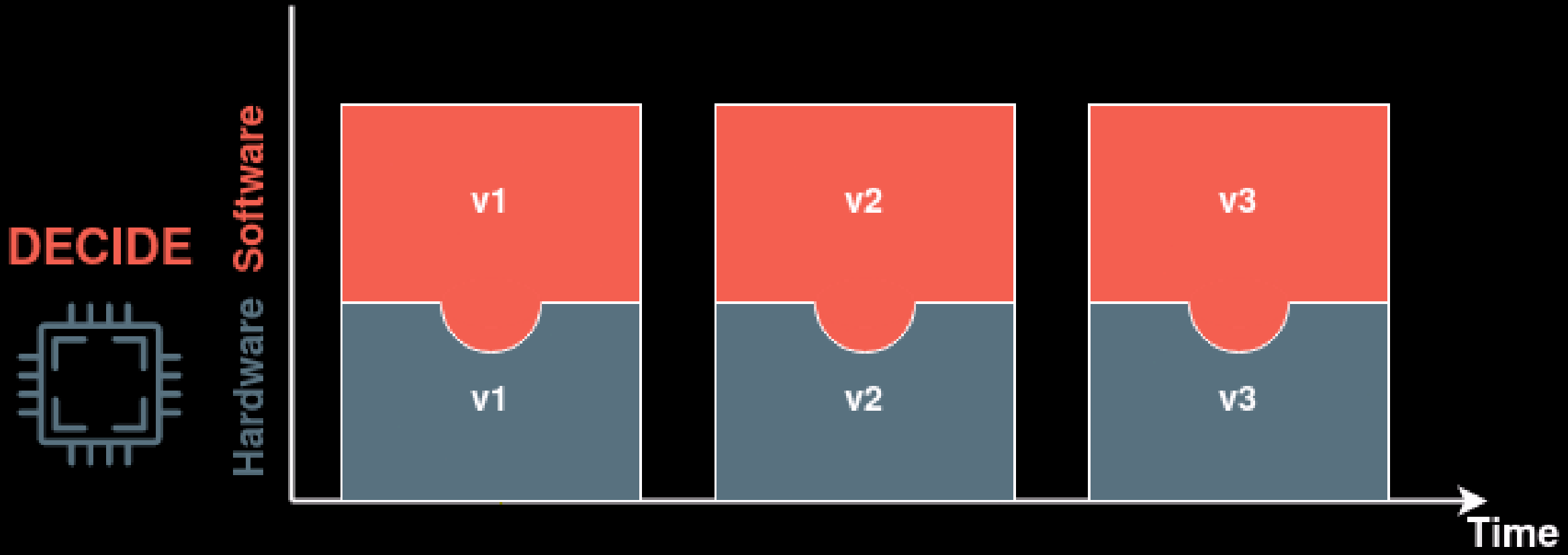
Old software on a new device

New software on an old device



Hardware 🔒 Software

Hardware ☠ Software



Hardware 🔒 Software

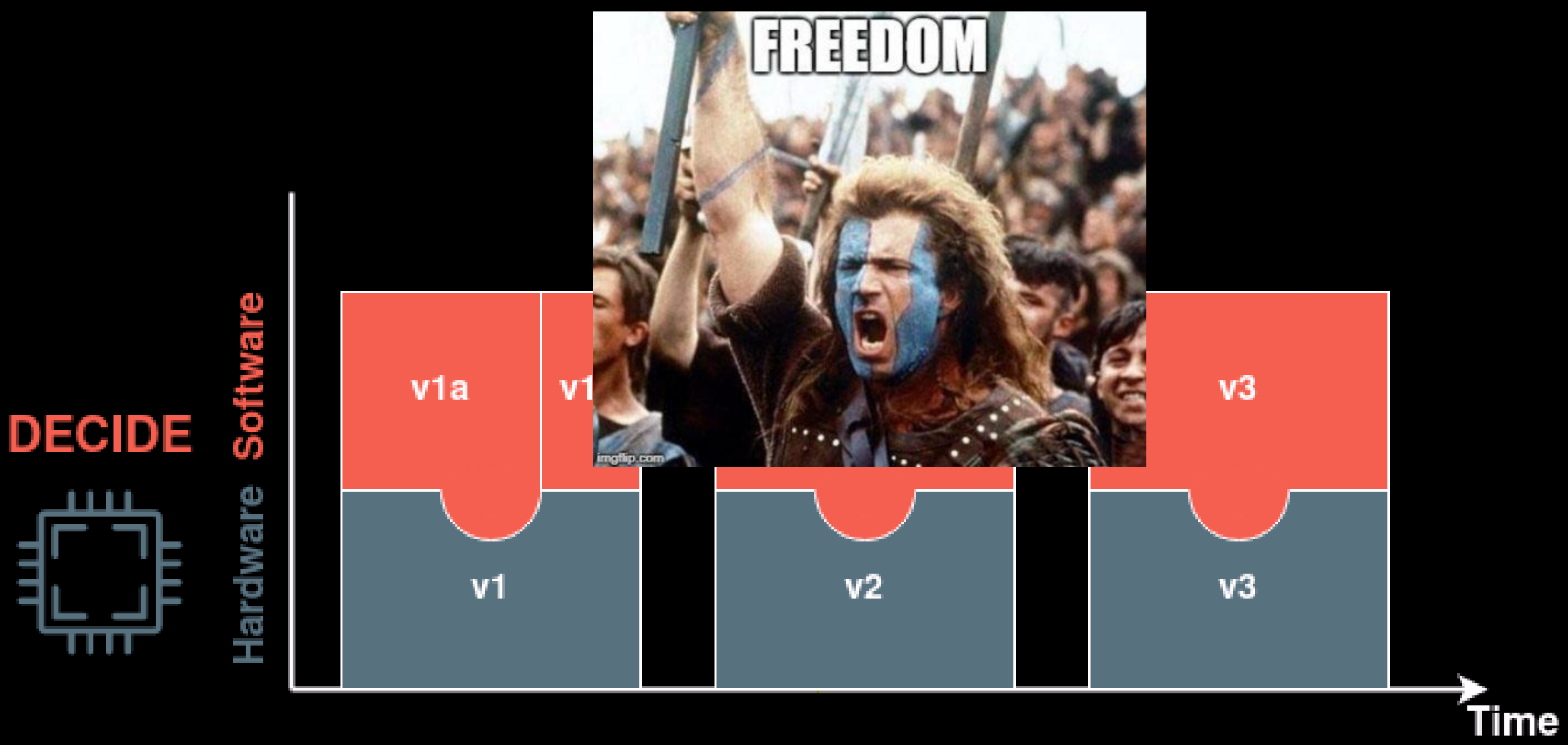
Hardware ☠ Software

DECIDE

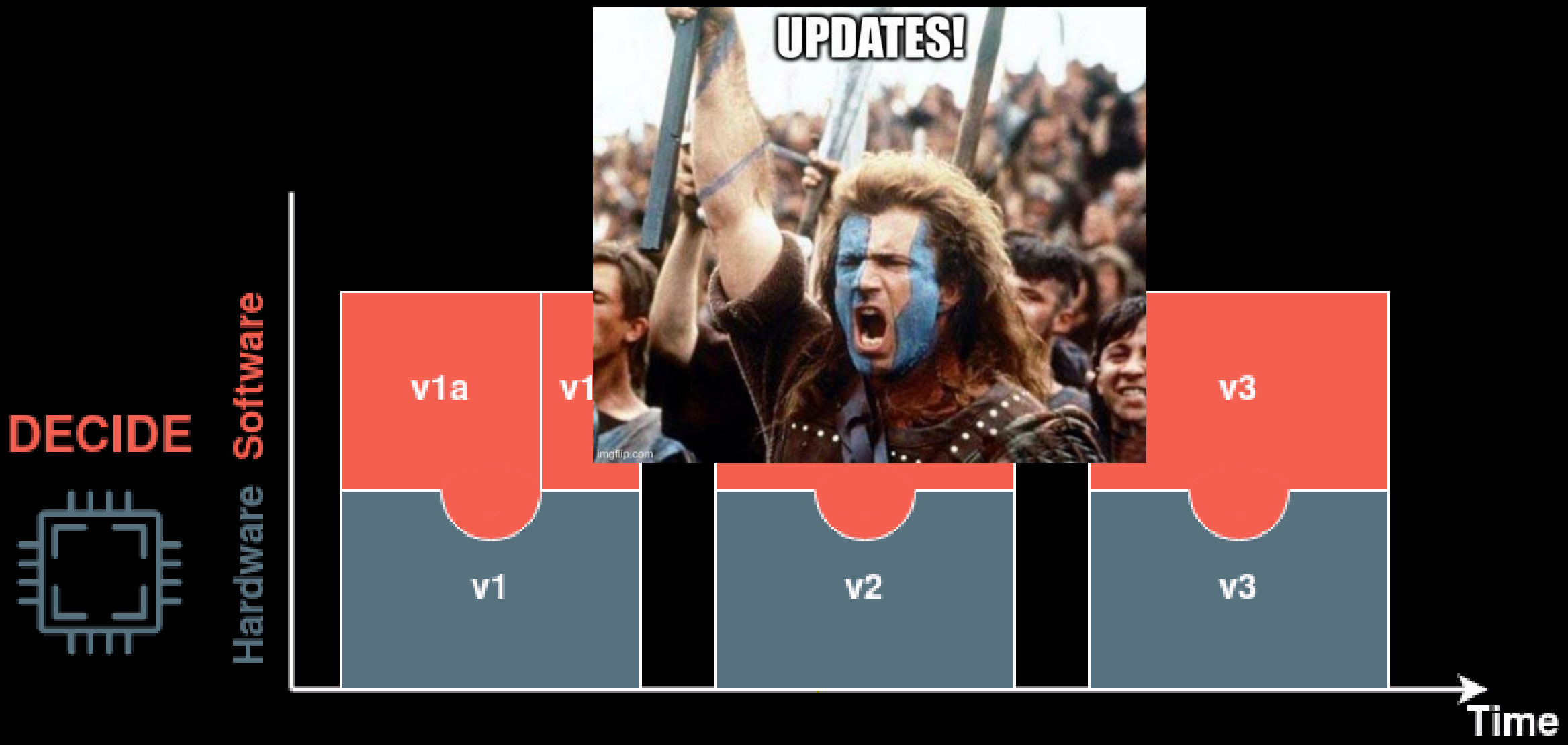
No specialization 🎯

Hardware 🔒 Software

Hardware ☠️ Software

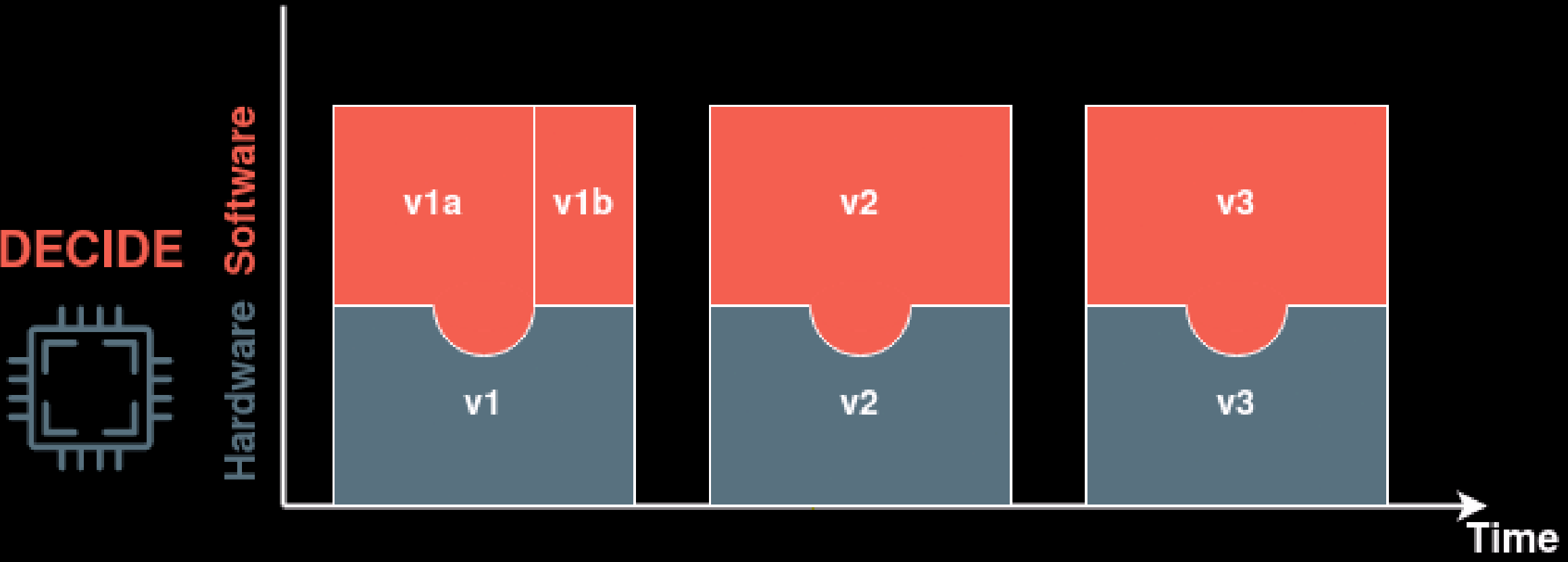


Software must become independent



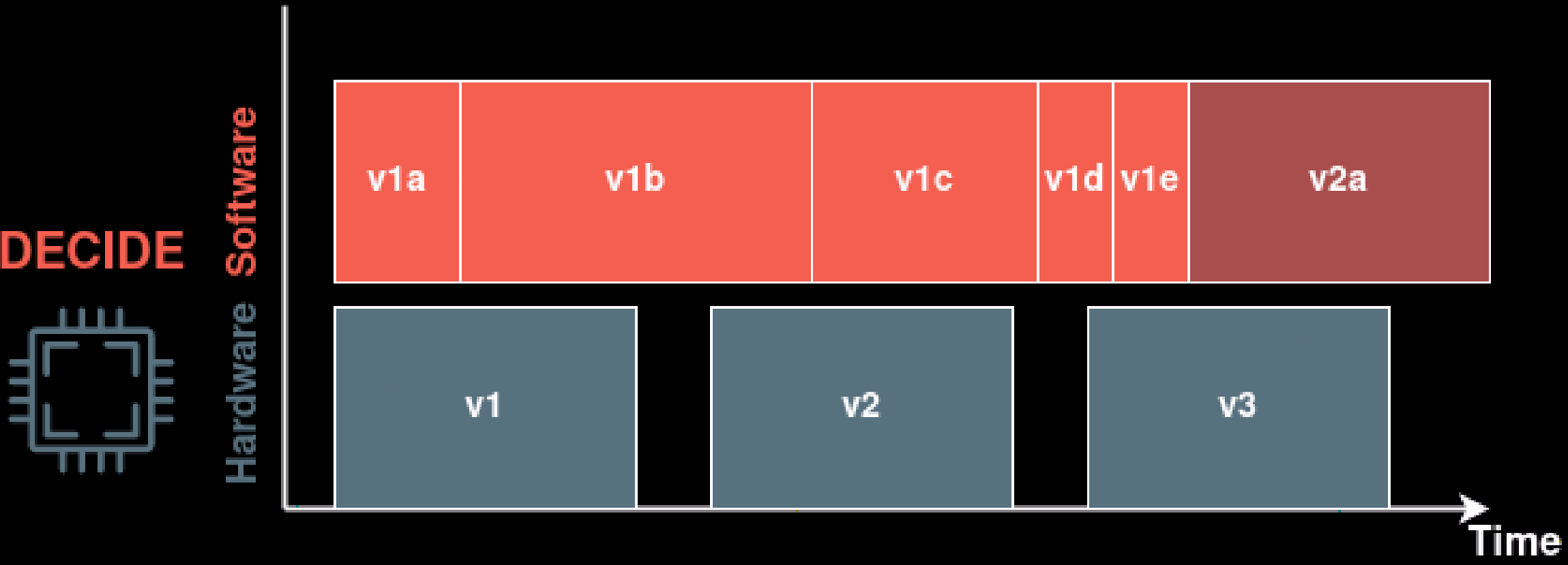
Software must become independent … and evolve

DECIDE

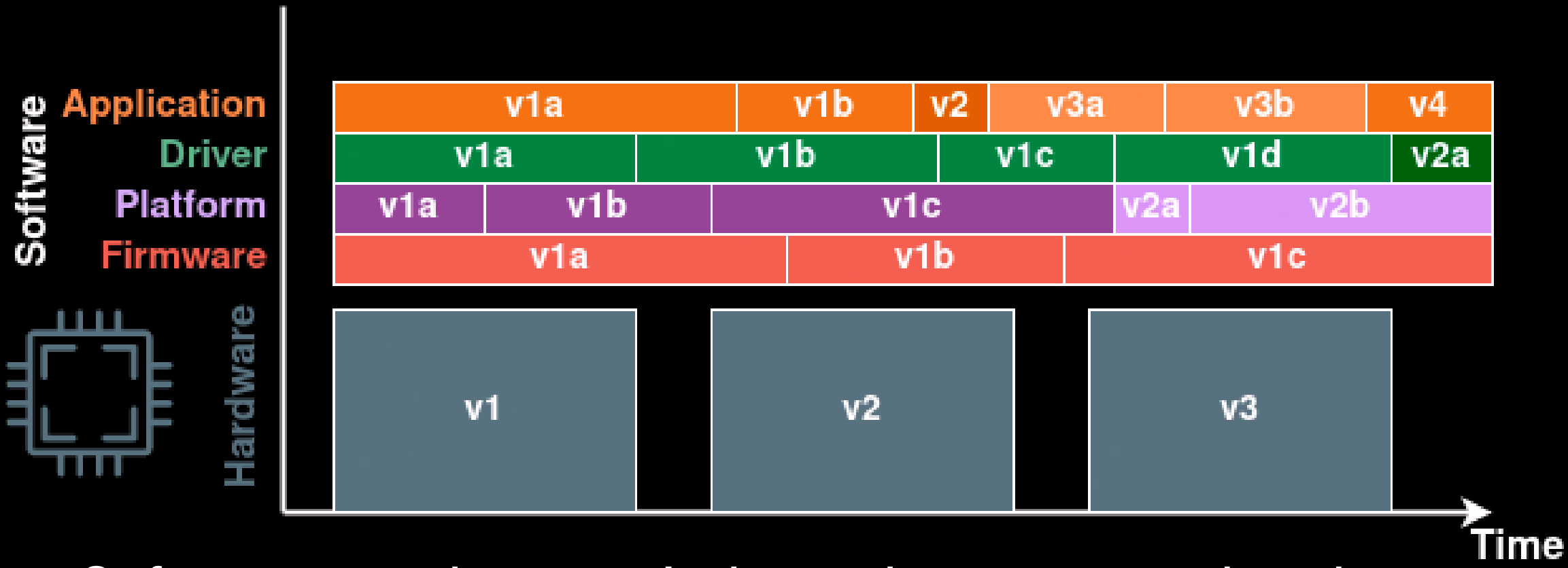


Software must become independent … and evolve

DECIDE



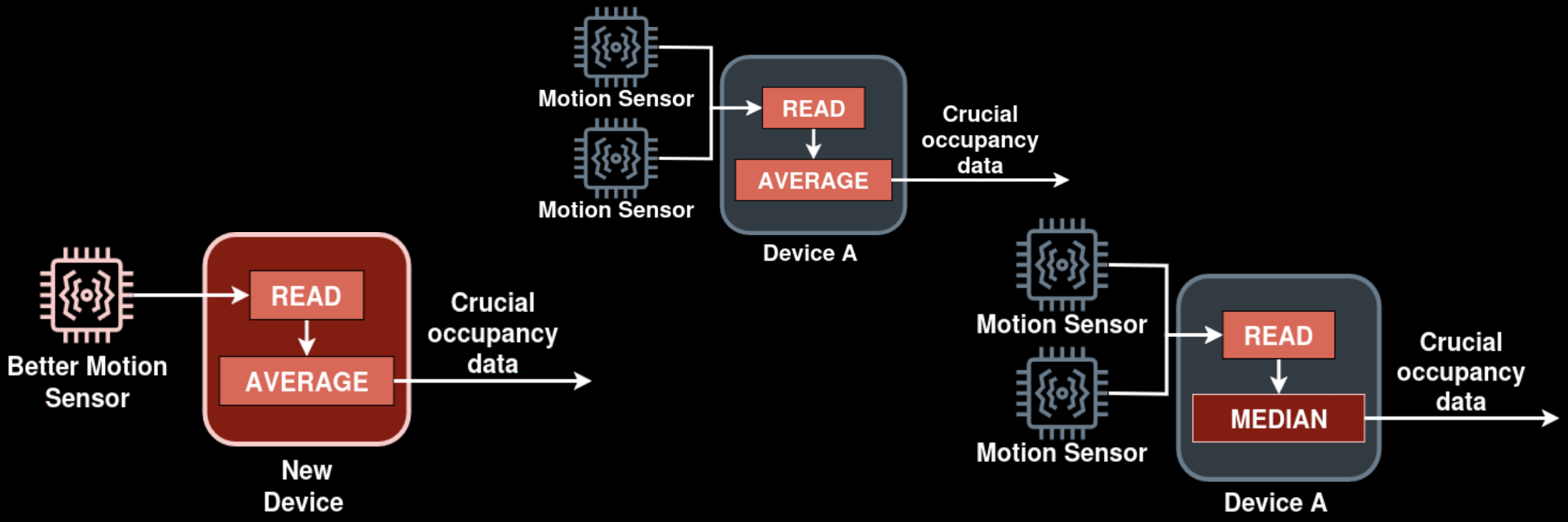
Software must become independent … and evolve



Software must become independent … and evolve

ECOSYSTEM that supports evolvability

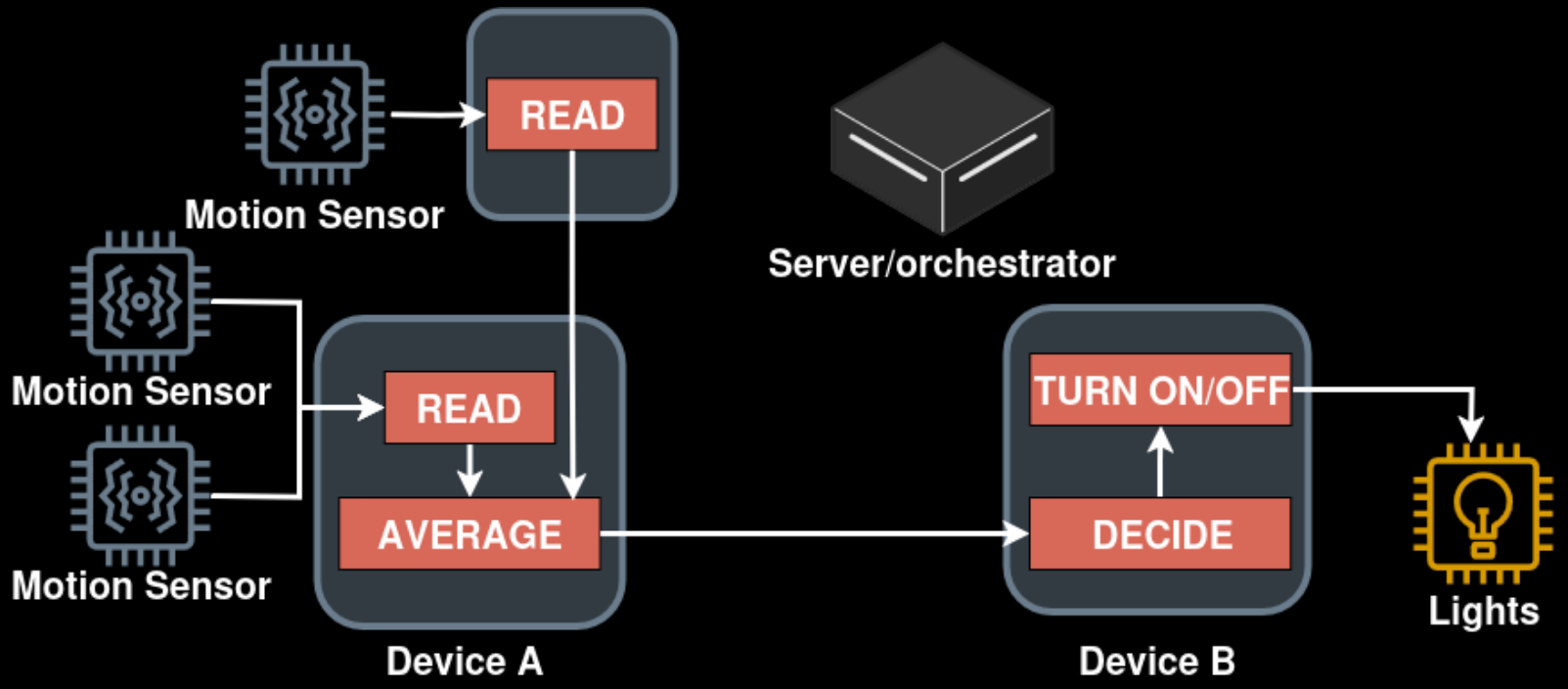
SUPPORTING SYSTEM CHANGES ON DEVICE LEVEL



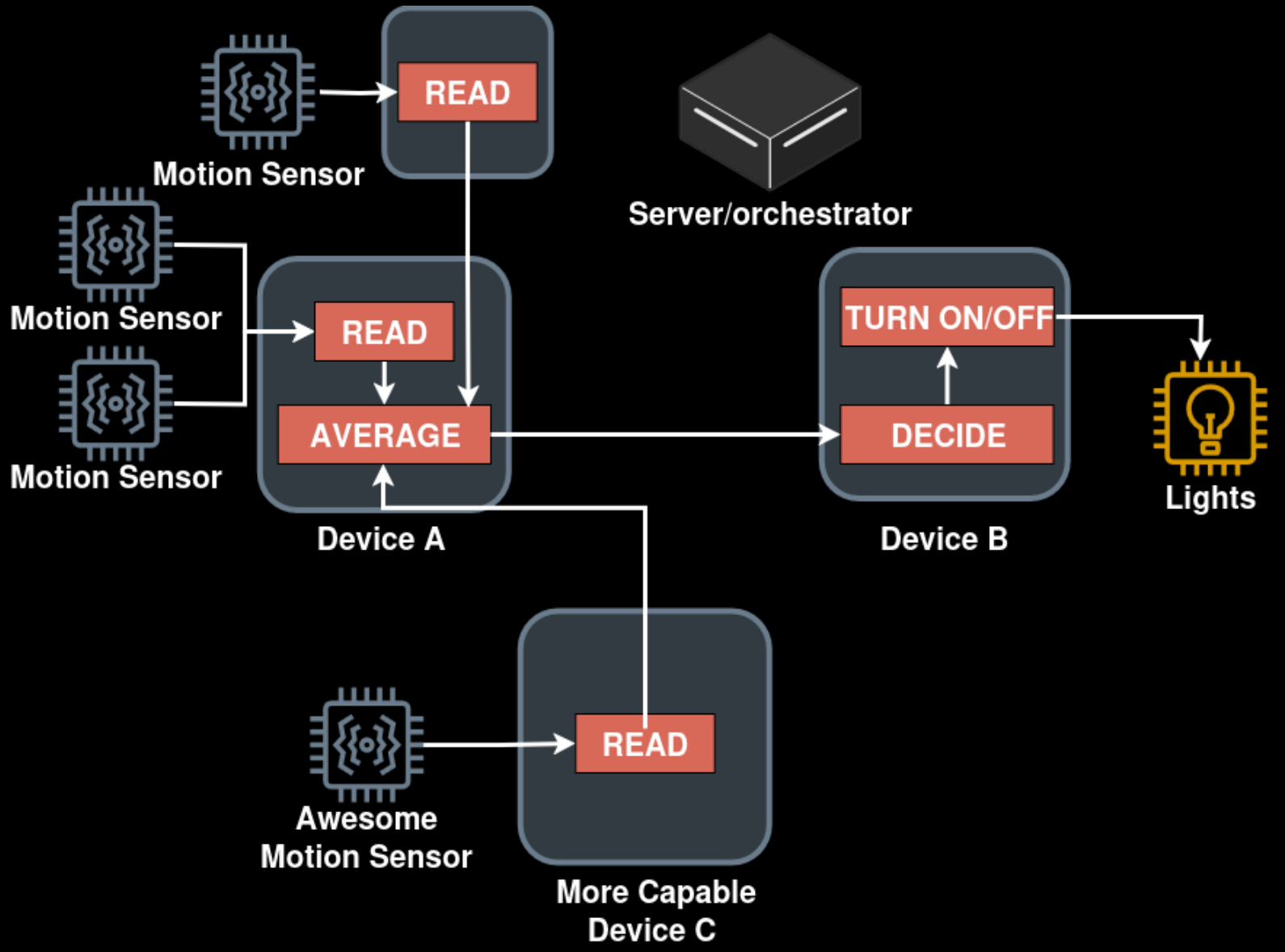
Old software on a new device

New software on an old device

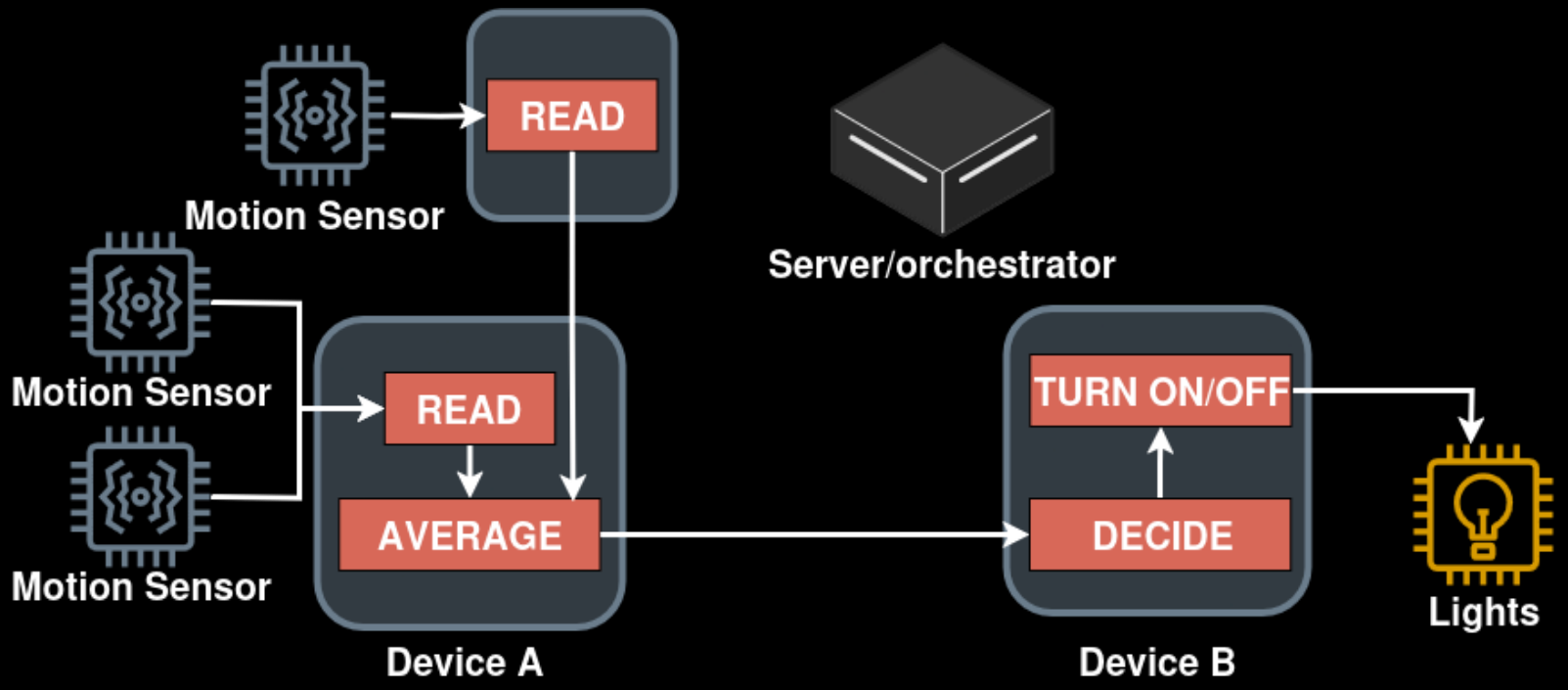
EXAMPLE CONTROL SYSTEM



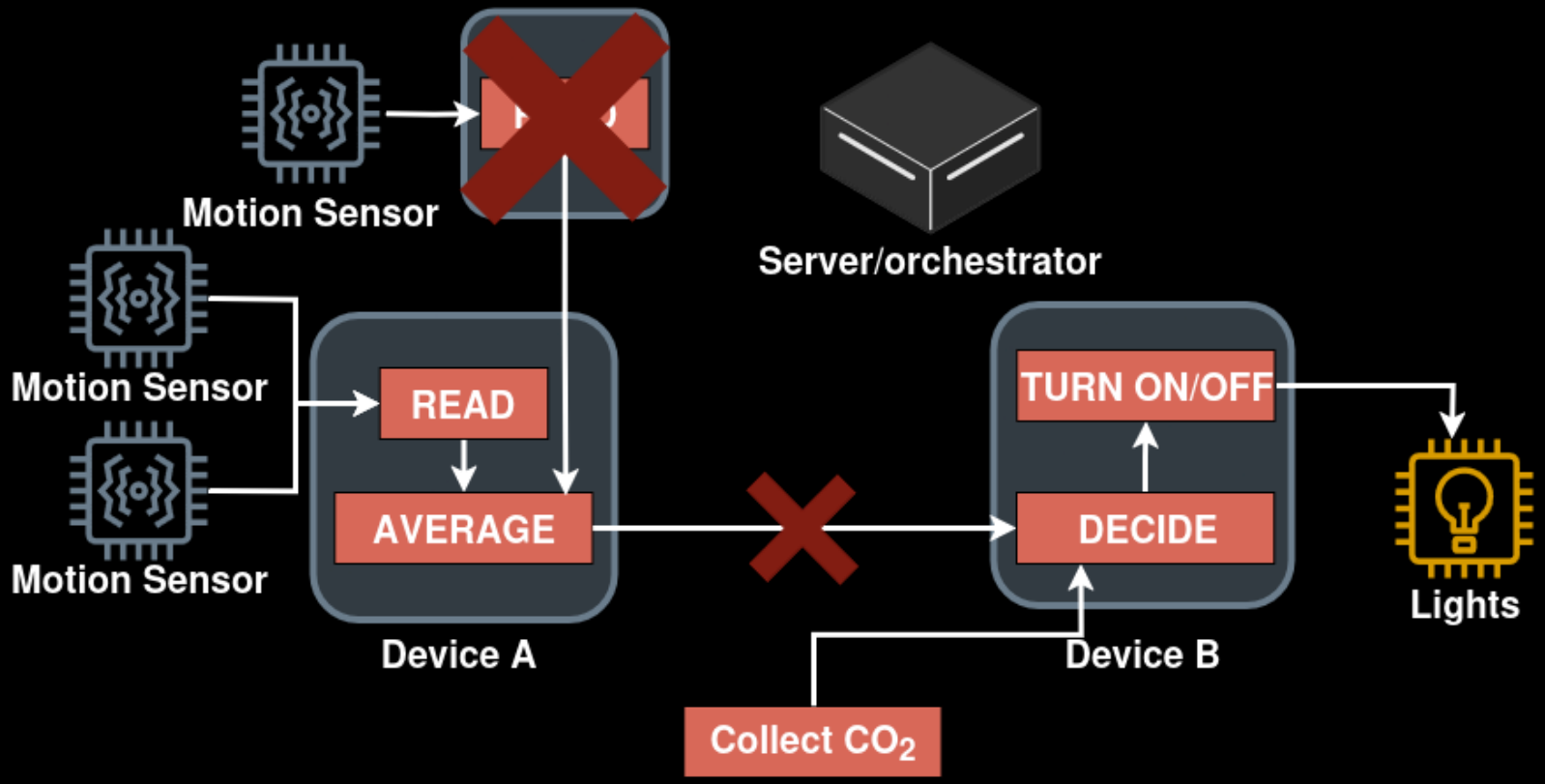
ADD FUNCTIONALITY AS DEVICES ARE ADDED



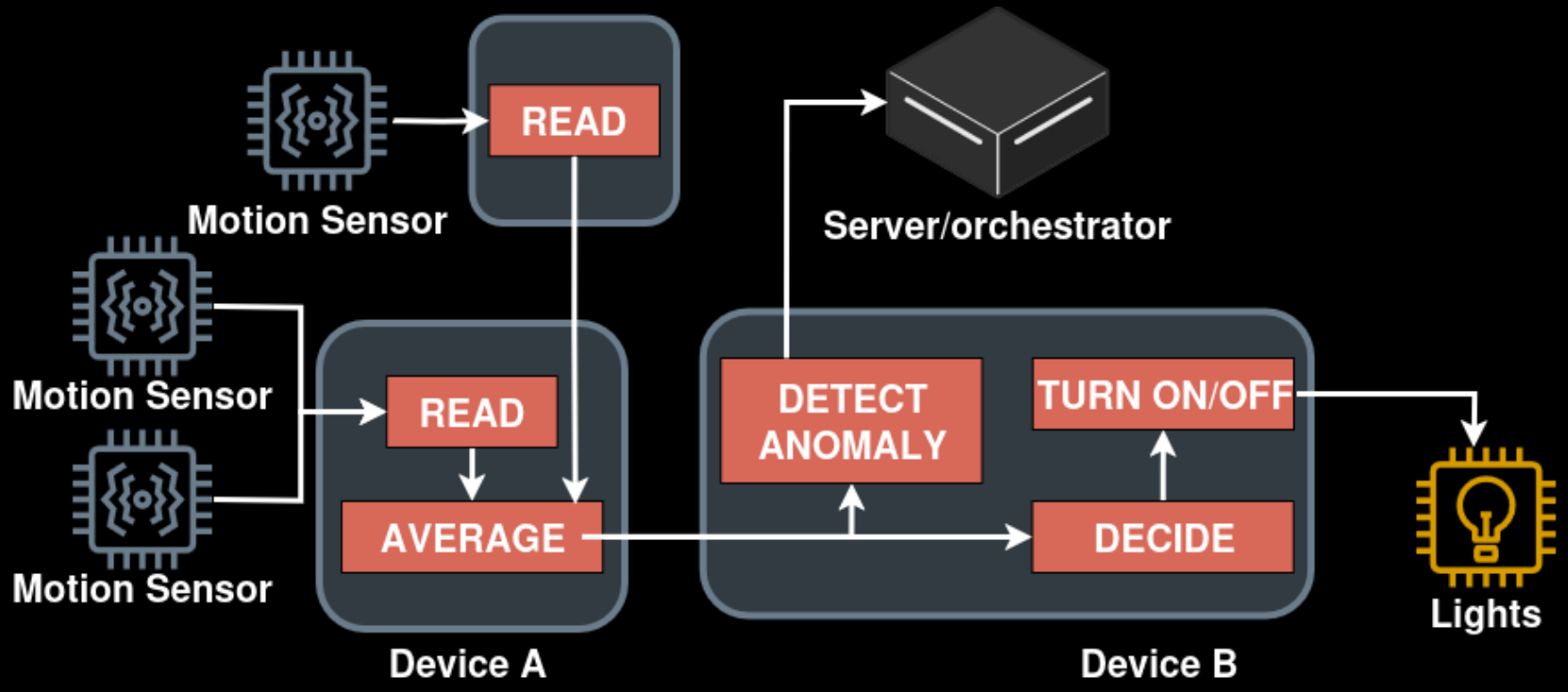
RETAIN FUNCTIONALITY AS DEVICES ARE LOST



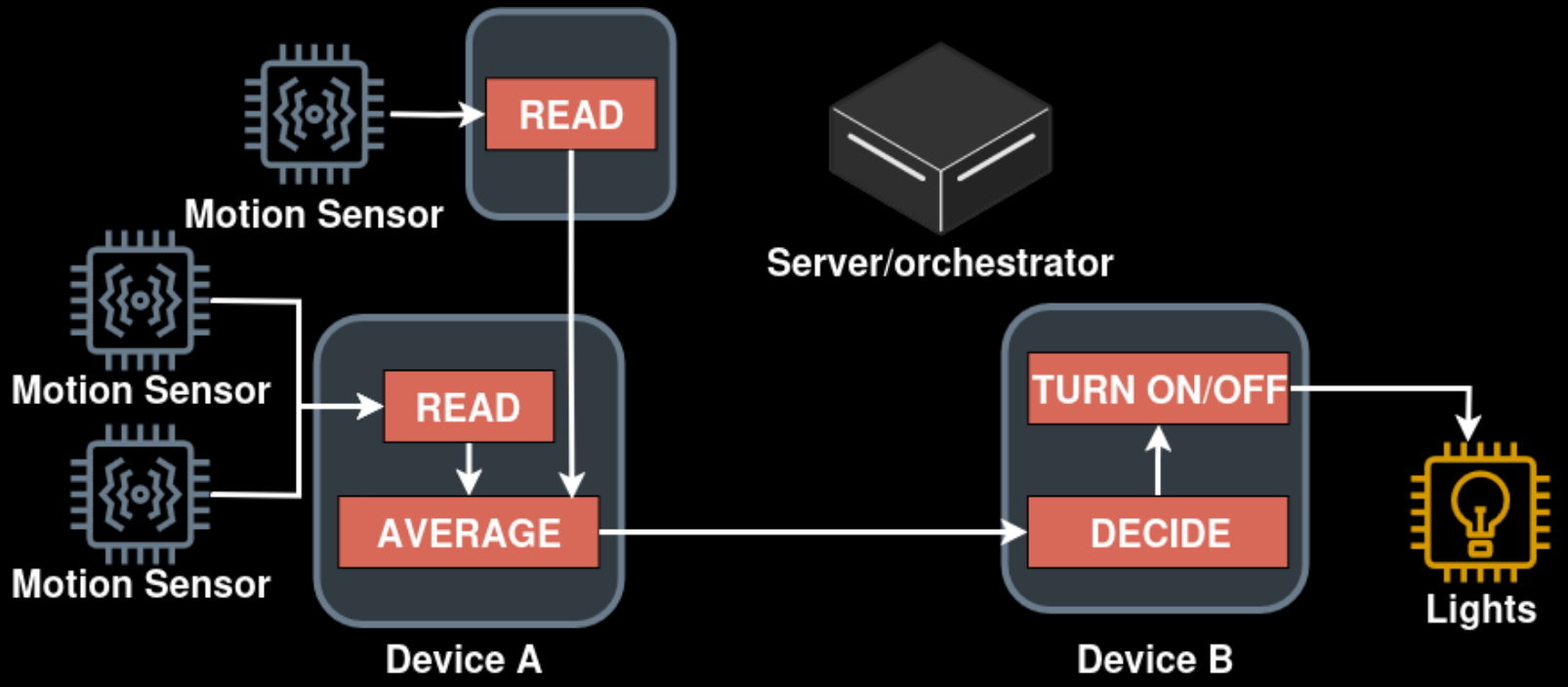
DEGRADE GRACEFULLY AS DEVICES ARE LOST



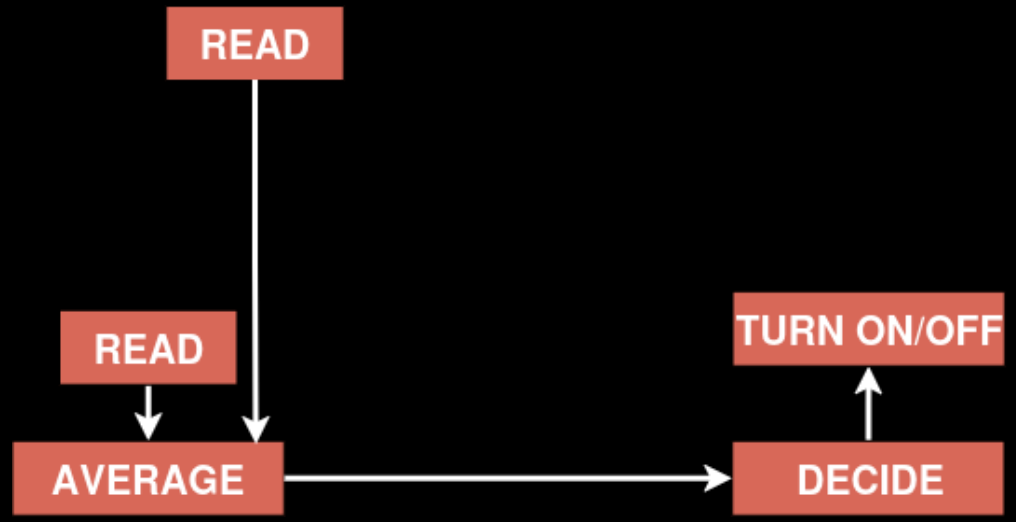
EXTEND FUNCTIONALITY WITH EXISTING DEVICES



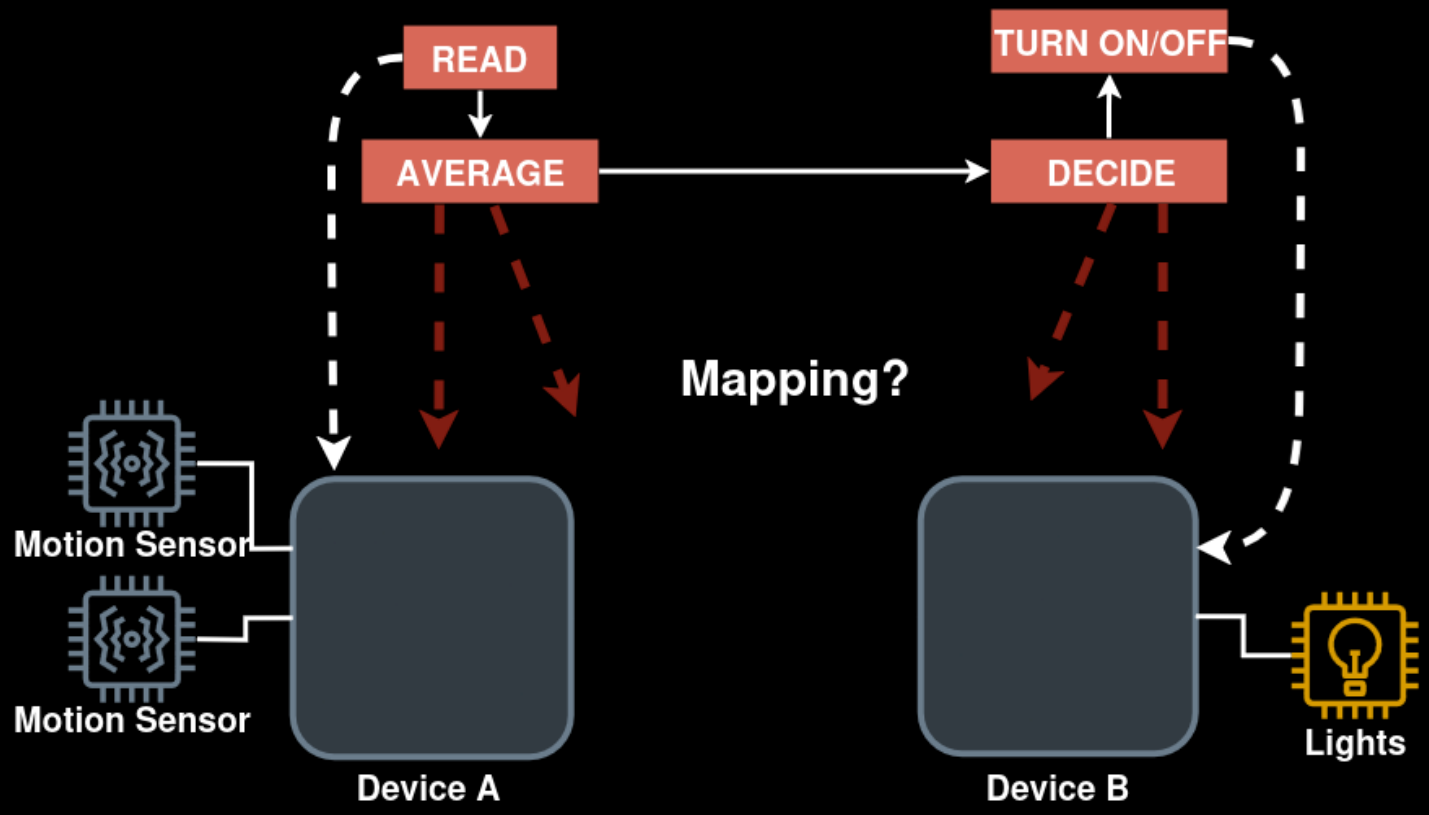
EXAMPLE CONTROL SYSTEM



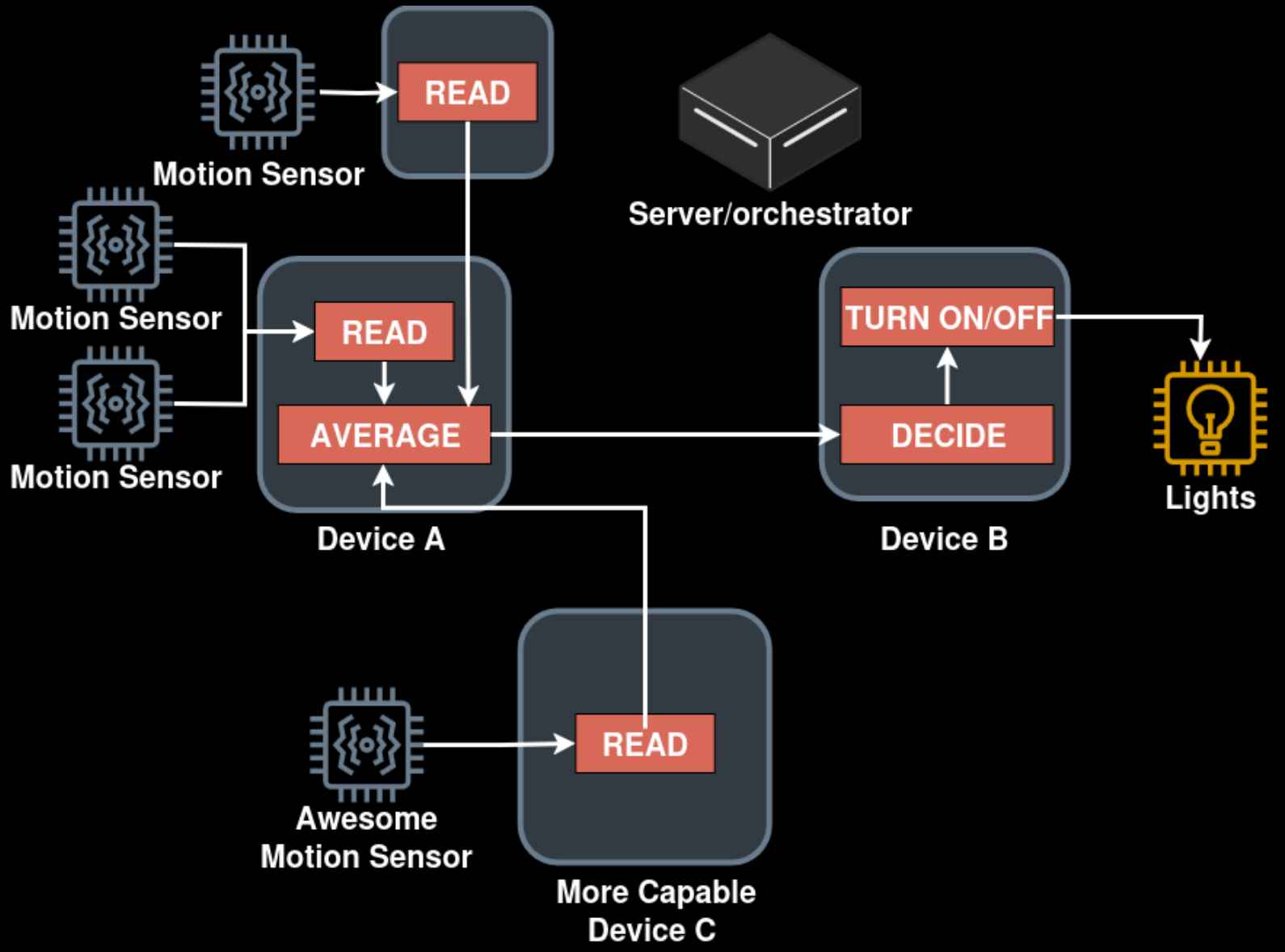
CONTROL TASK GRAPH



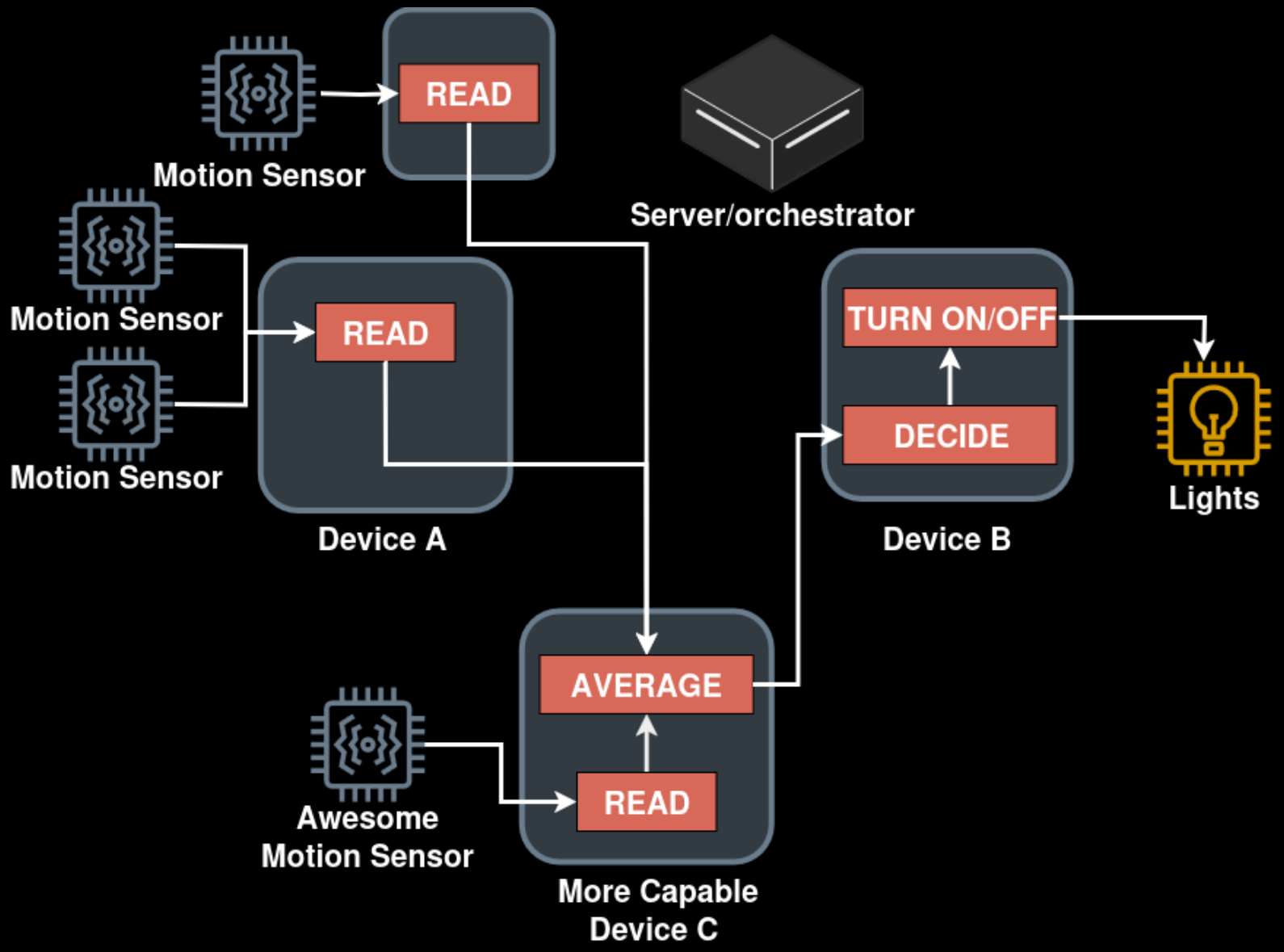
DEPLOYING THE CONTROL TASK GRAPH



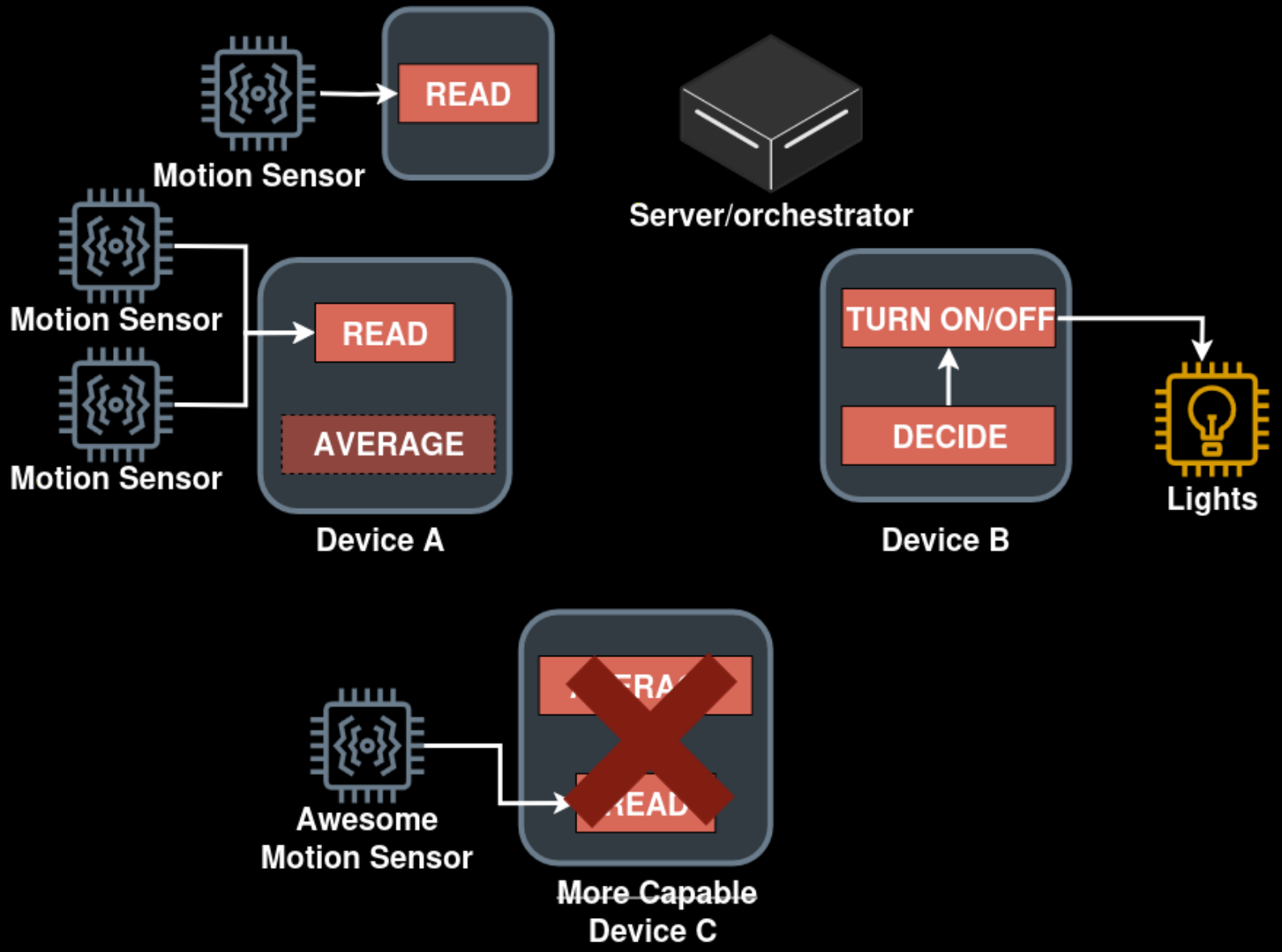
ADD FUNCTIONALITY AS DEVICES ARE ADDED



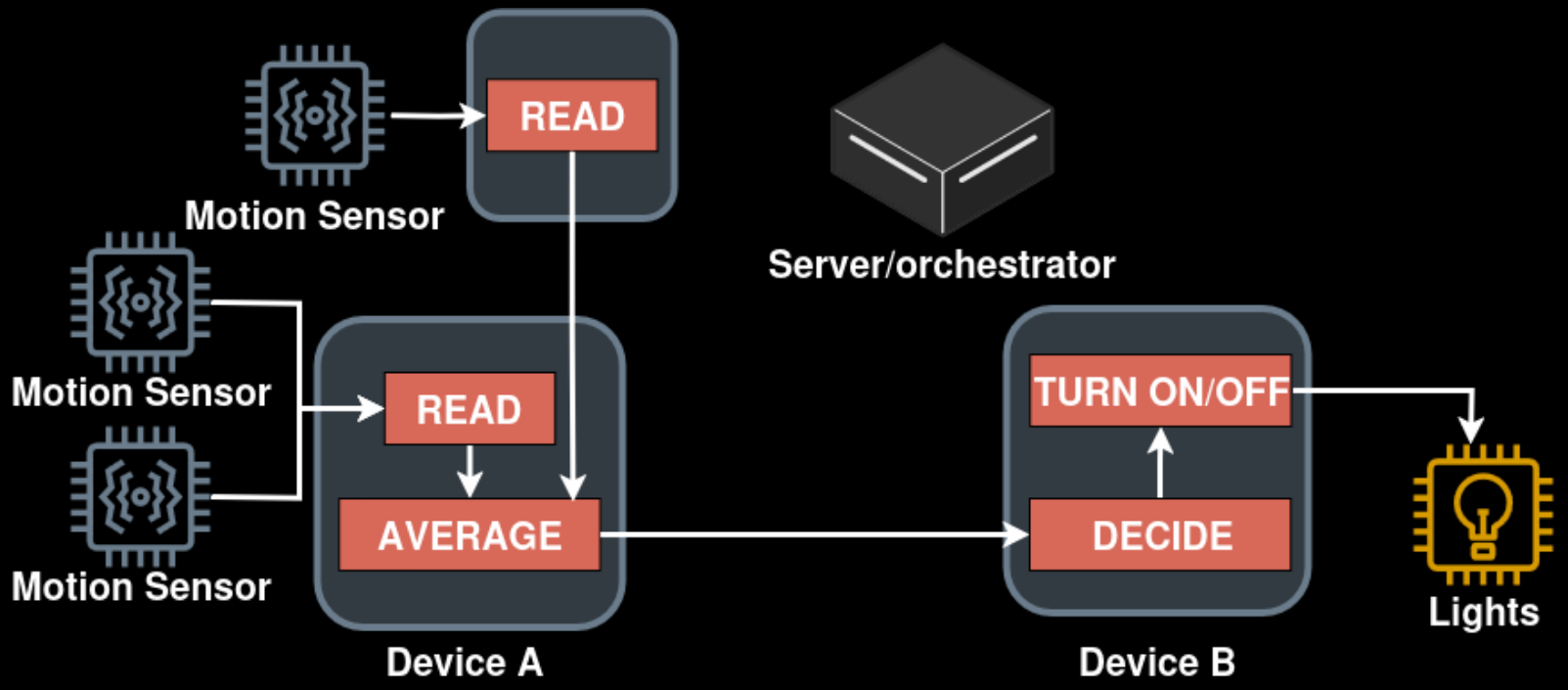
ADD FUNCTIONALITY AS DEVICES ARE ADDED

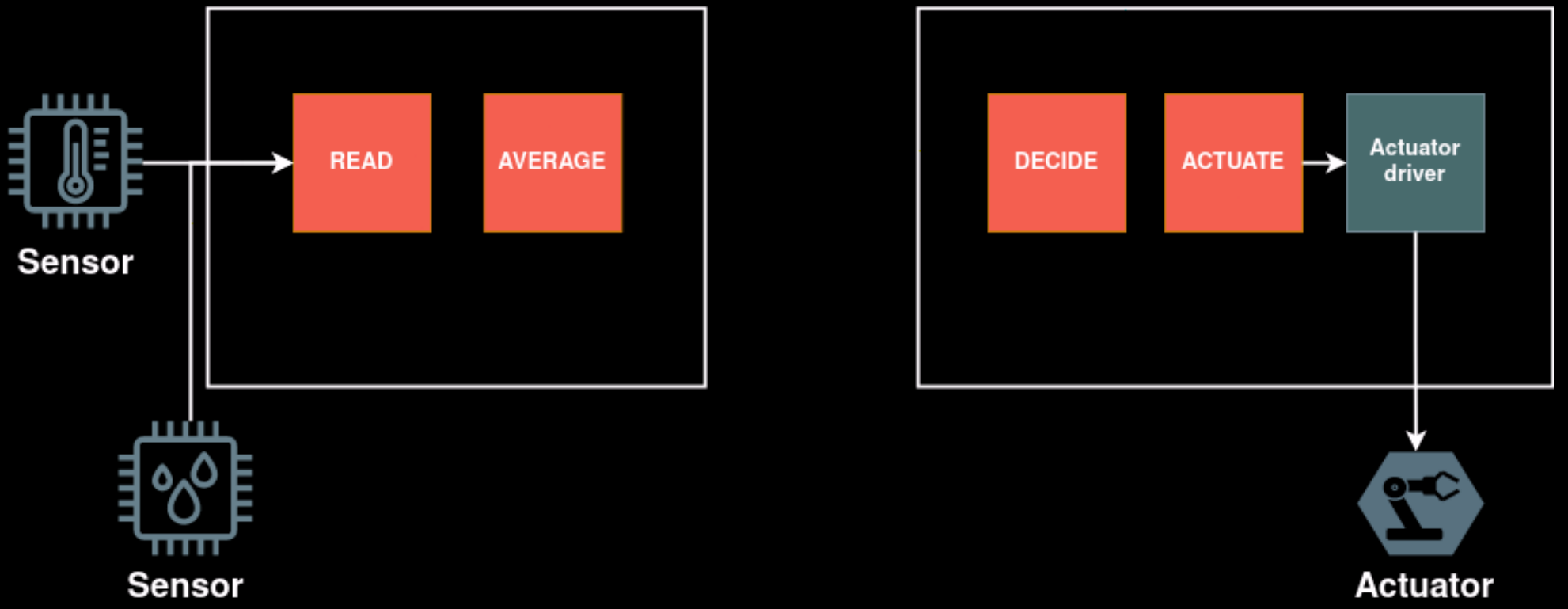


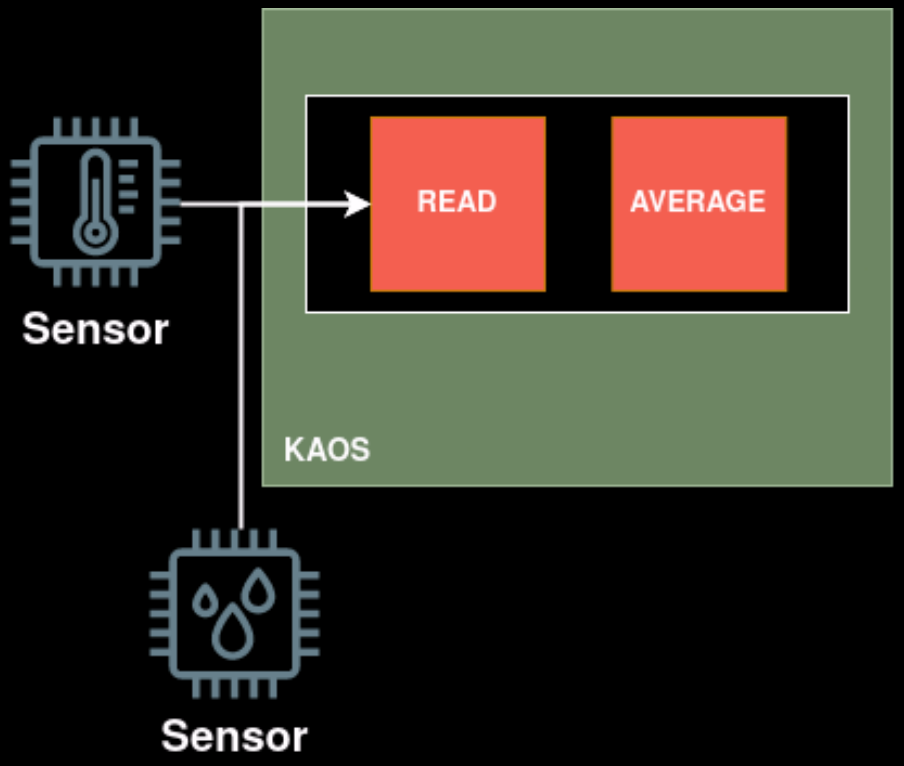
ADD FUNCTIONALITY AS DEVICES ARE ADDED

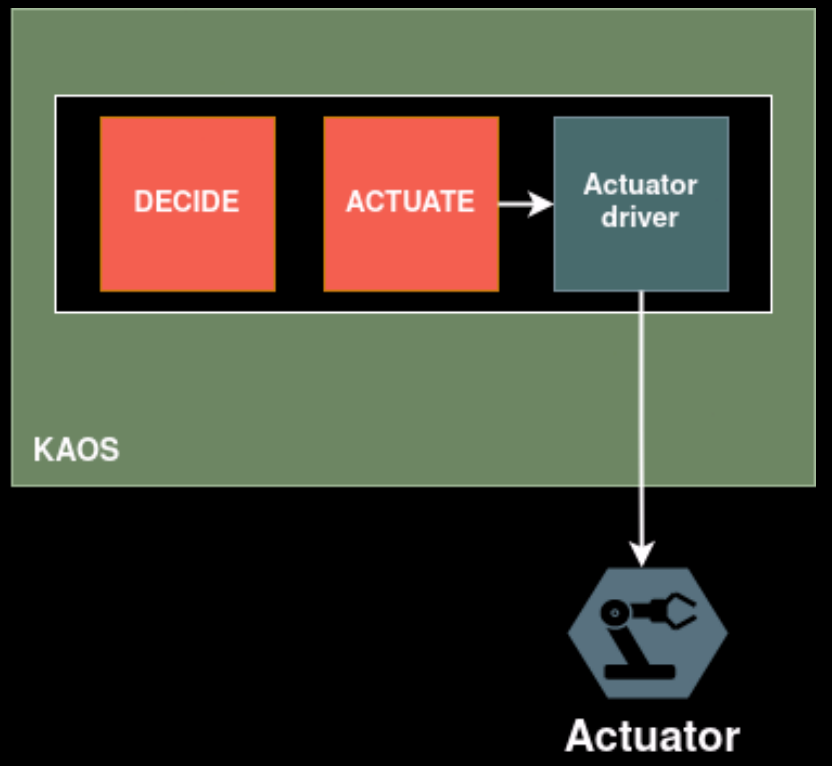


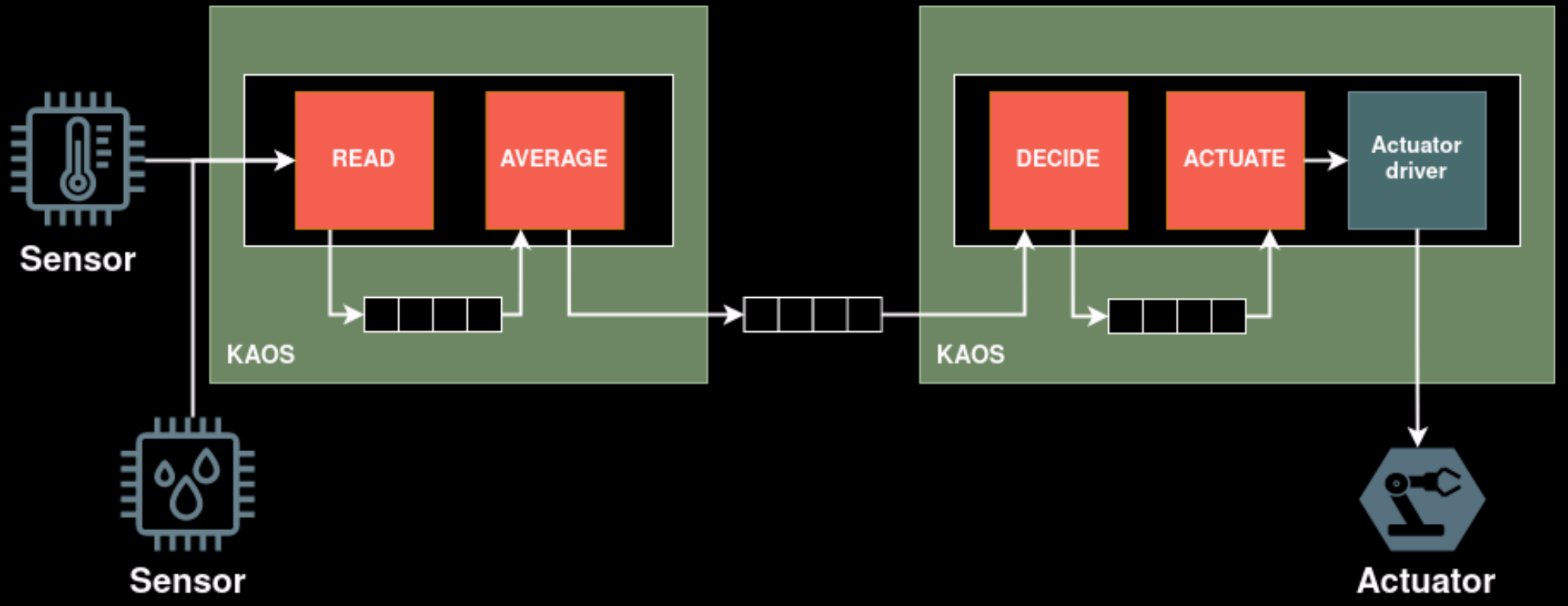
RETAIN FUNCTIONALITY AS DEVICES ARE LOST

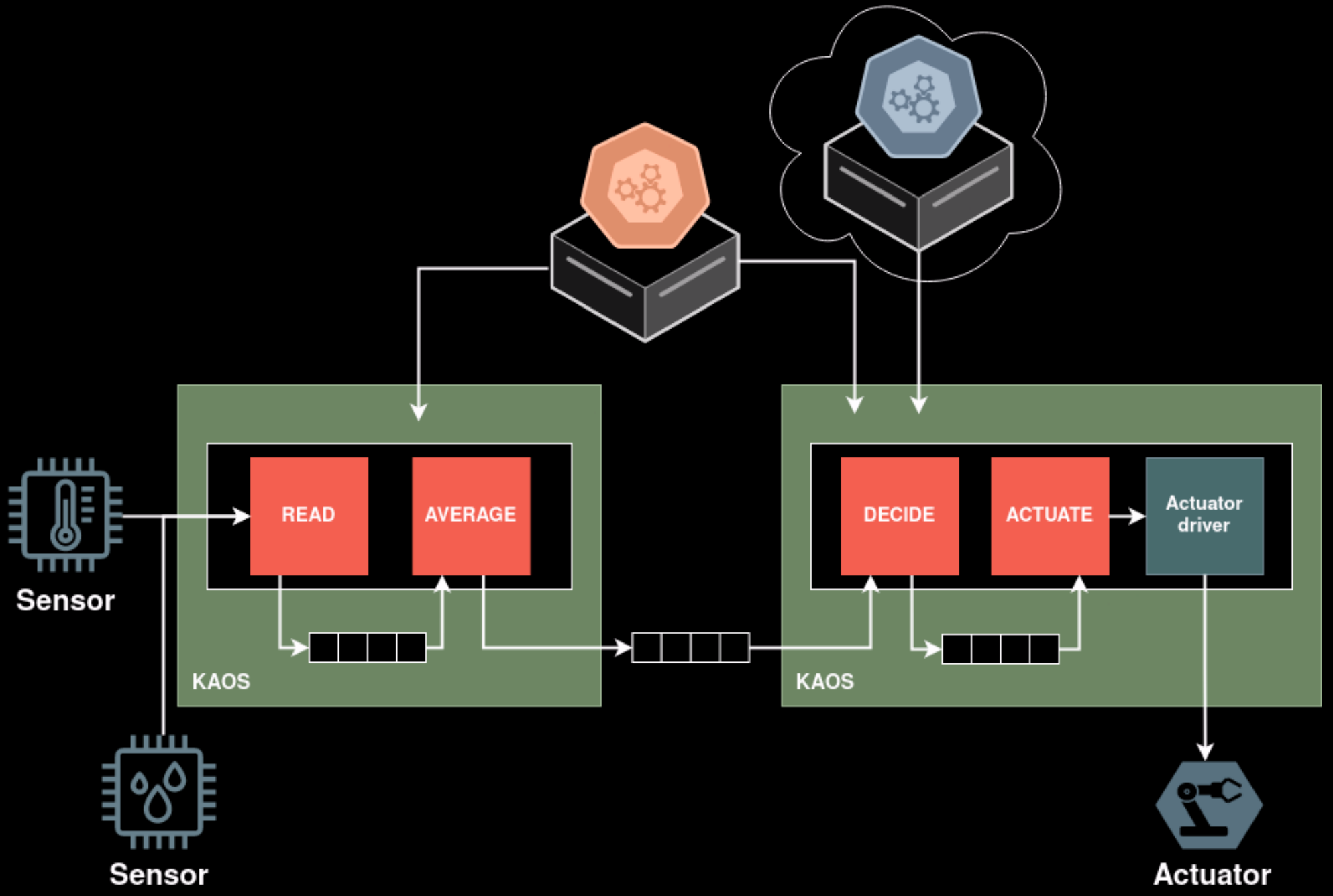



READ
AVERAGE
Sensor
Sensor
DECIDE
ACTUATE
Actuator driver
Actuator


READ
AVERAGE
Sensor
KAOS
Sensor


DECIDE
ACTUATE
Actuator driver
KAOS
Actuator


Sensor
READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Actuator


READ
AVERAGE
DECIDE
ACTUATE
Actuator driver
KAOS
KAOS
Sensor
Sensor
Actuator

ESP32

Small but mighty

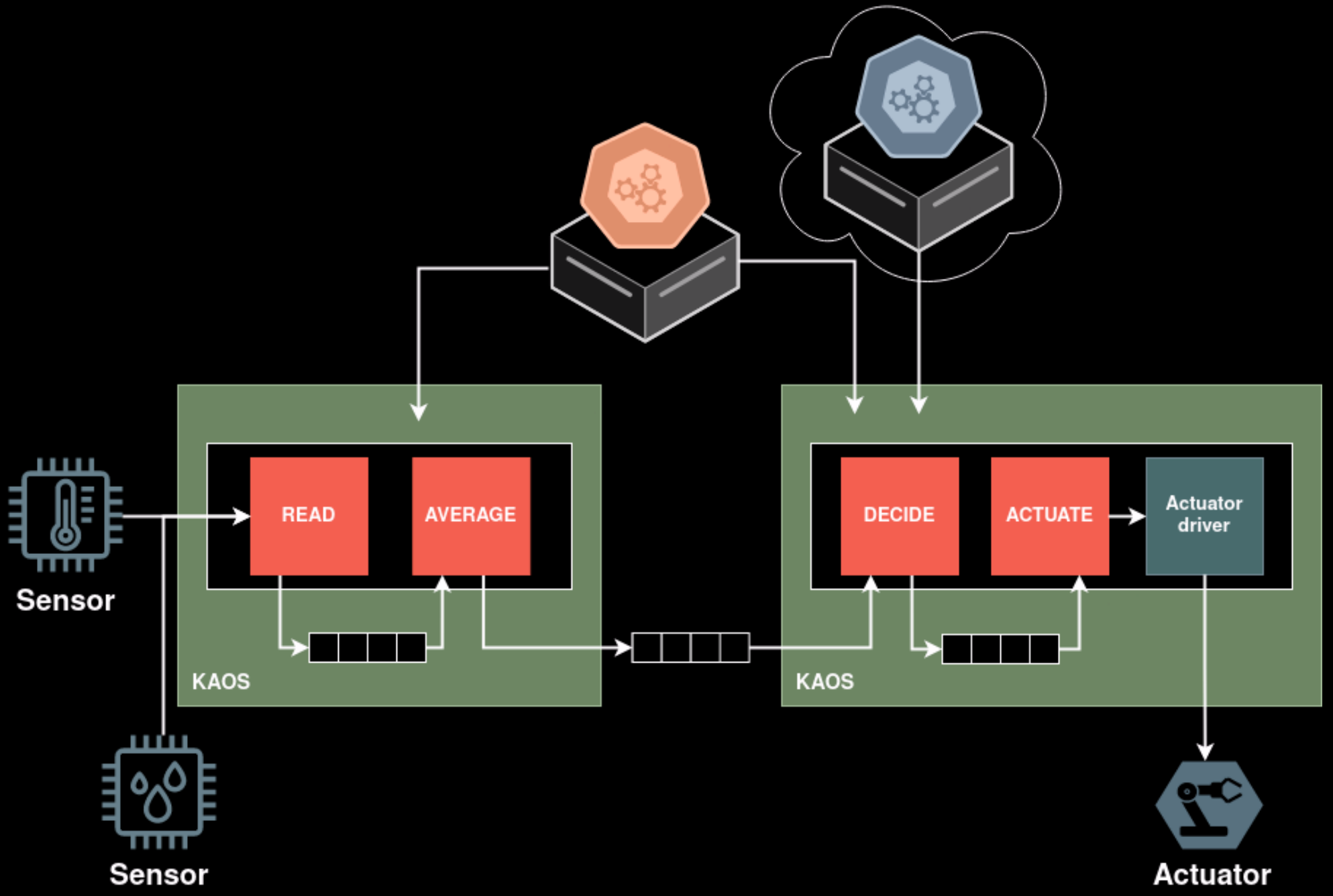
Cheap devices (~30p) with plenty of resources

Multi-core, co-processors, megabytes* of RAM

WiFi + Bluetooth

Very popular platform for smart devices

We can do stuff!


READ
AVERAGE
DECIDE
ACTUATE
Actuator driver
KAOS
KAOS
Sensor
Sensor
Actuator

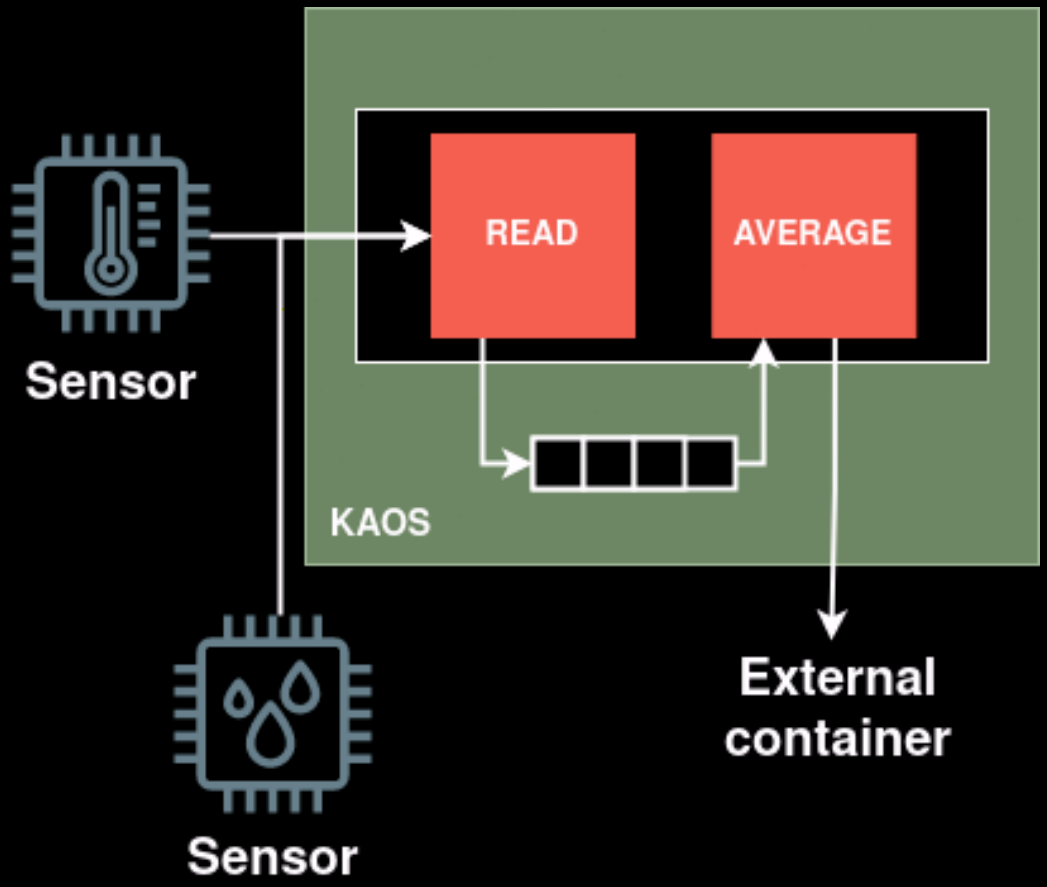
PORTABLE COMPONENTS

Decoupling hardware and software

Specialize software development

Accommodate multiple stakeholders

Run on various vendor platforms



PORTABLE COMPONENTS

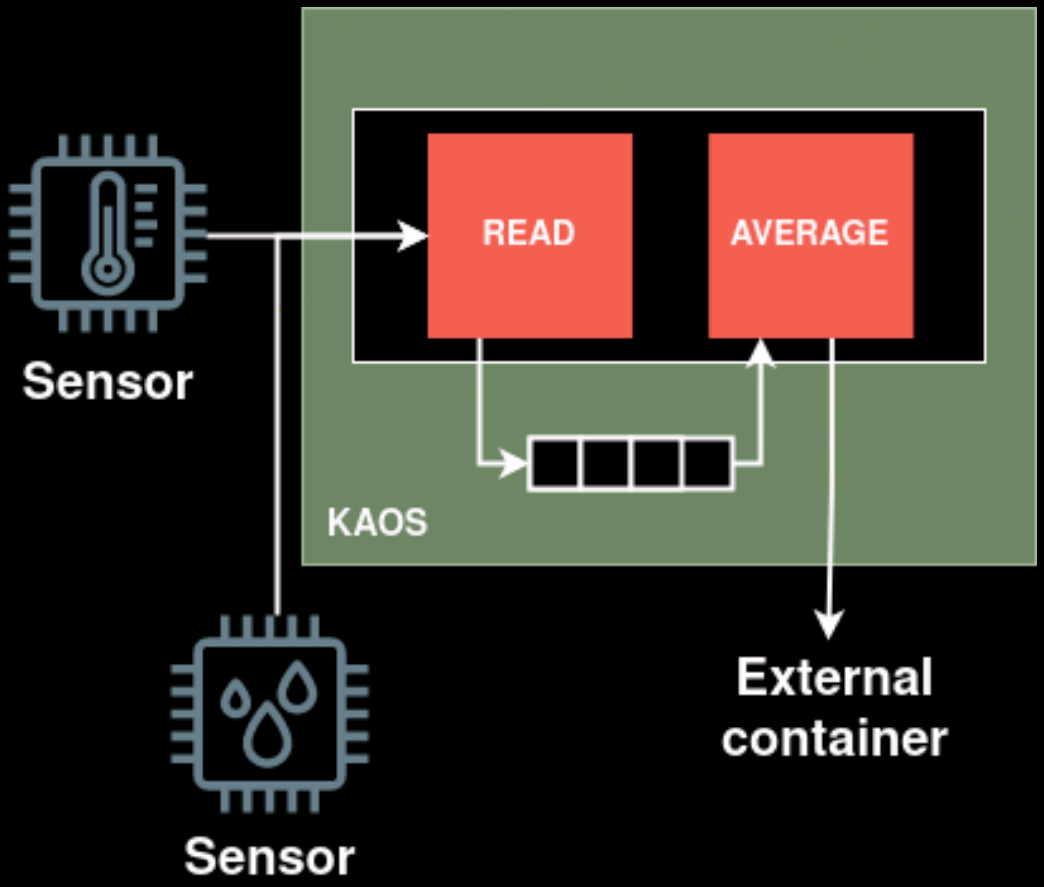
Decoupling hardware and software

Specialize software development

Accommodate multiple stakeholders

Run on various vendor platforms

$$$



PORTABLE COMPONENTS

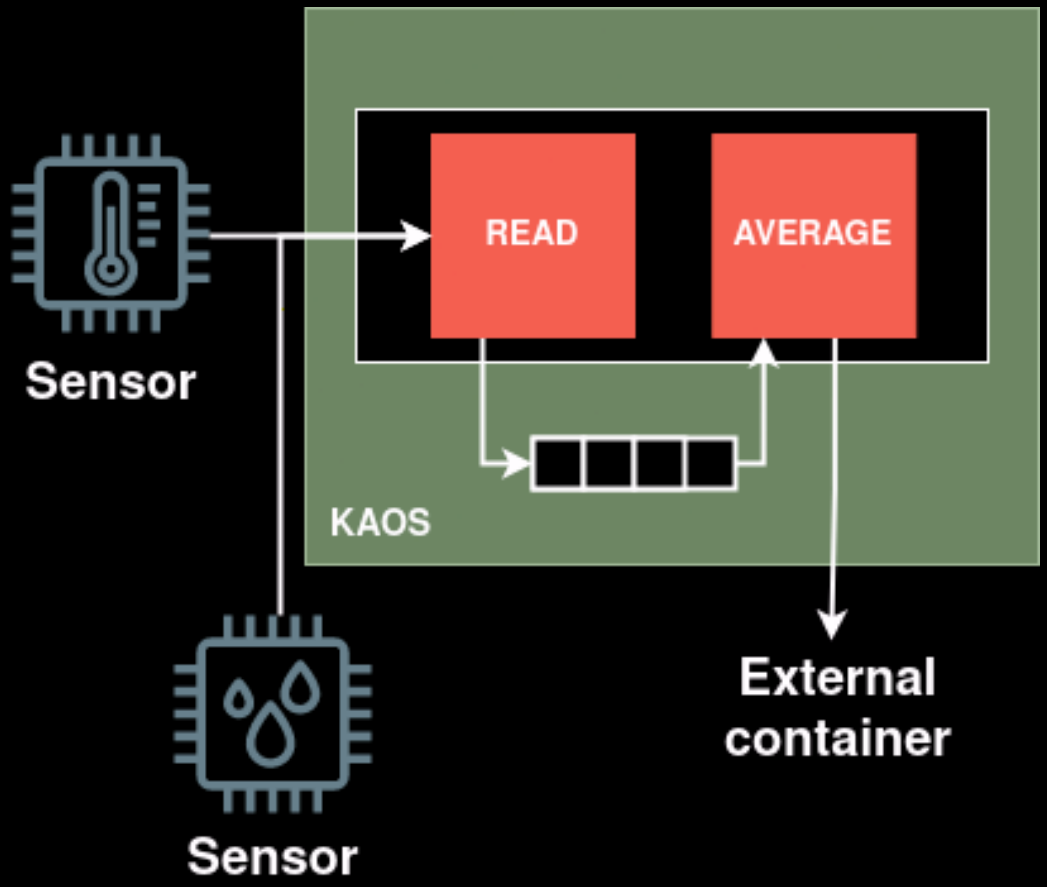
Decoupling hardware and software

Specialize software development

Accommodate multiple stakeholders

Run on various hardware platforms

→ **EVOLVABILITY**



PORTABLE COMPONENTS

Decoupling hardware and software

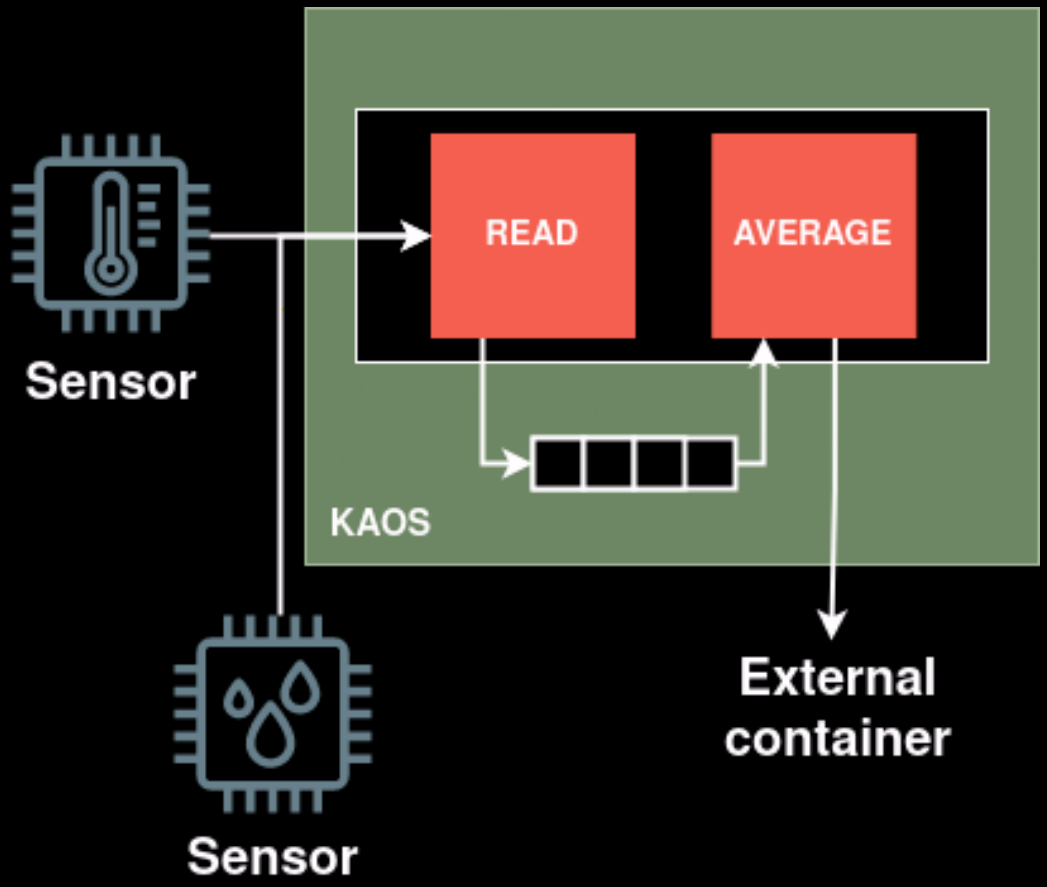
Specialize software development

Accommodate multiple stakeholders

Run on various hardware platforms

→ **EVOLVABILITY**

Restart computation and move it around



PORTABLE COMPONENTS

Decoupling hardware and software

Specialize software development

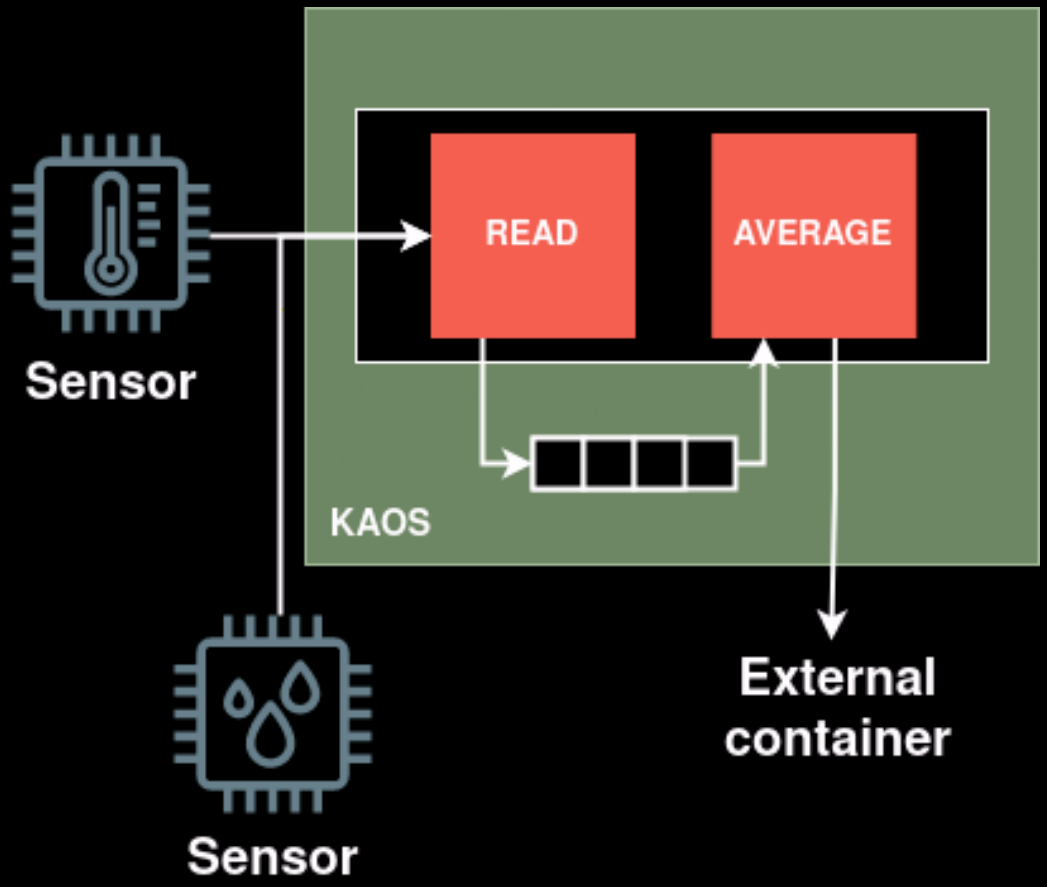
Accommodate multiple stakeholders

Run on various hardware platforms

→ **EVOLVABILITY**

Restart computation and move it around

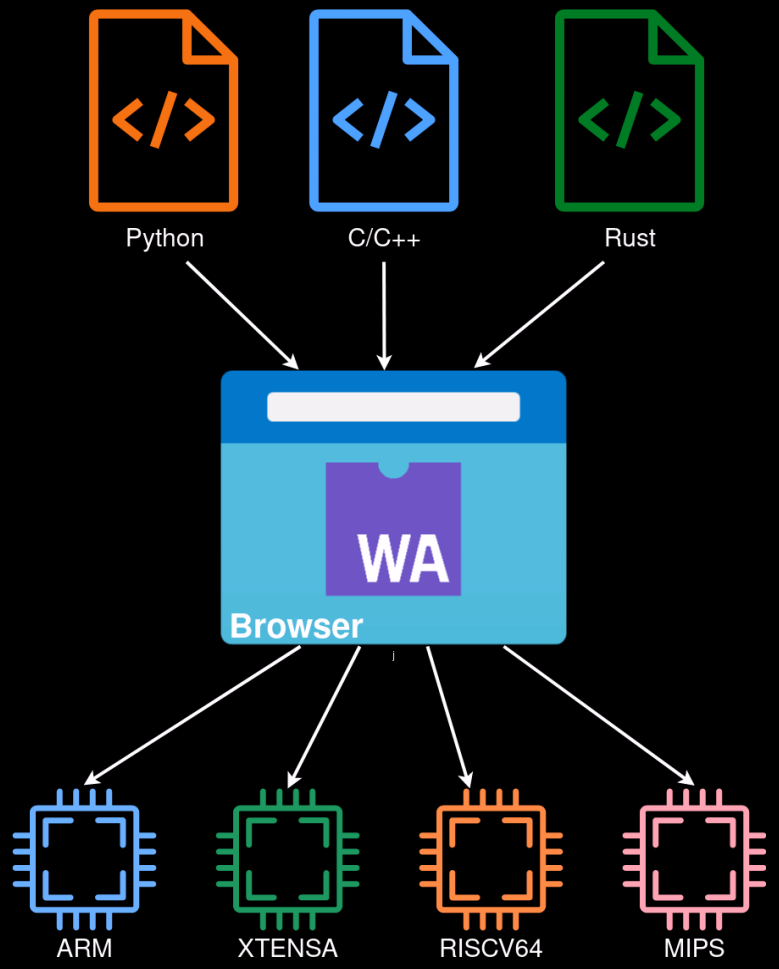
→ **RESILIENCE**



WEB ASSEMBLY ON ESP32?

Open portable binary standard

High performance computation in the browser

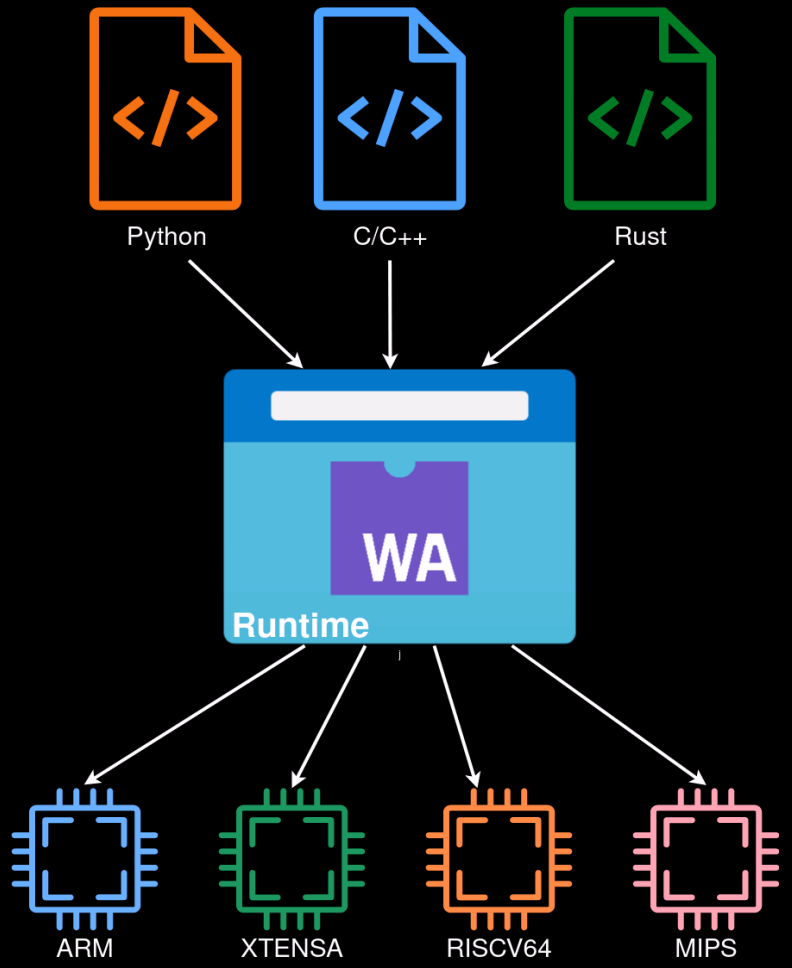


WEB ASSEMBLY ON ESP32?

Open portable binary standard

High performance computation ~~in the browser~~

Web Assembly Micro Runtime: small footprint



WEB ASSEMBLY ON ESP32?

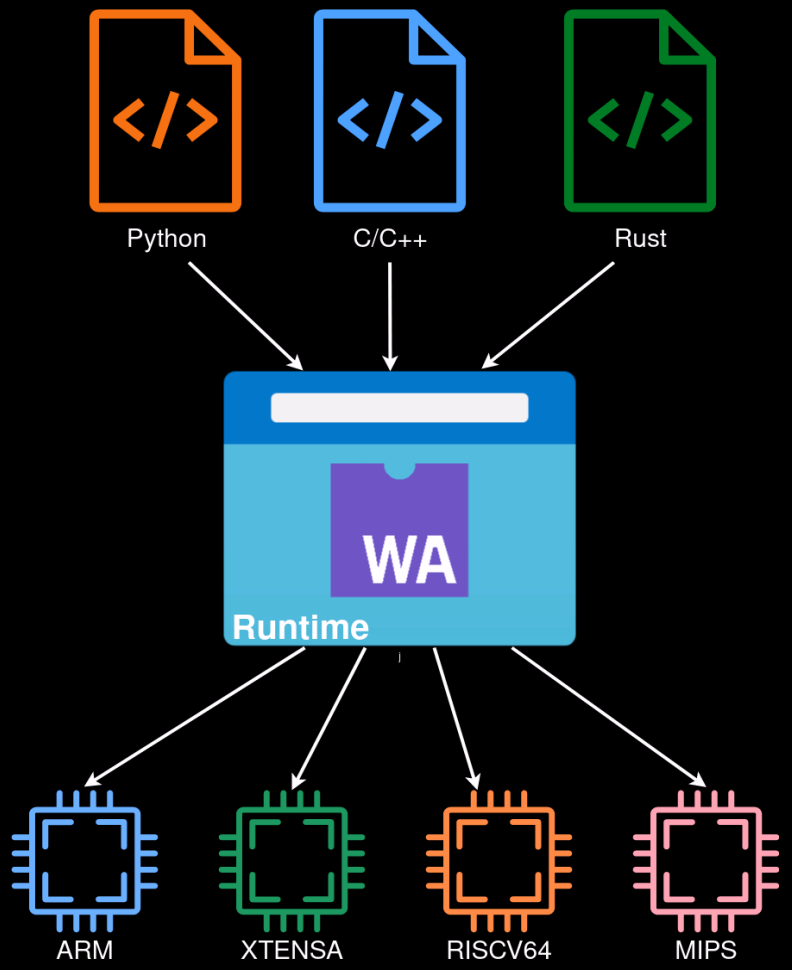
Open portable binary standard

High performance computation ~~in the browser~~

WAMR Micro Runtime: small footprint

Isolated sandbox → SECURITY

Independent portable containers



WEB ASSEMBLY ON ESP32?

Open portable binary standard

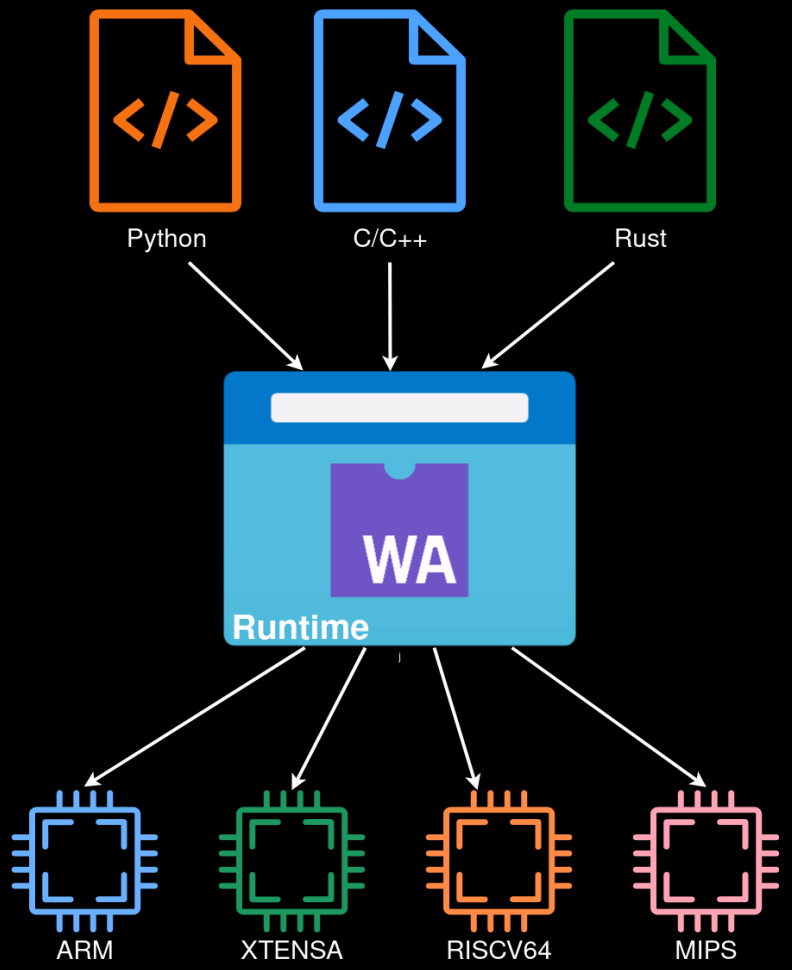
High performance computation ~~in the browser~~

WAMR Micro Runtime: small footprint

Isolated sandbox → SECURITY

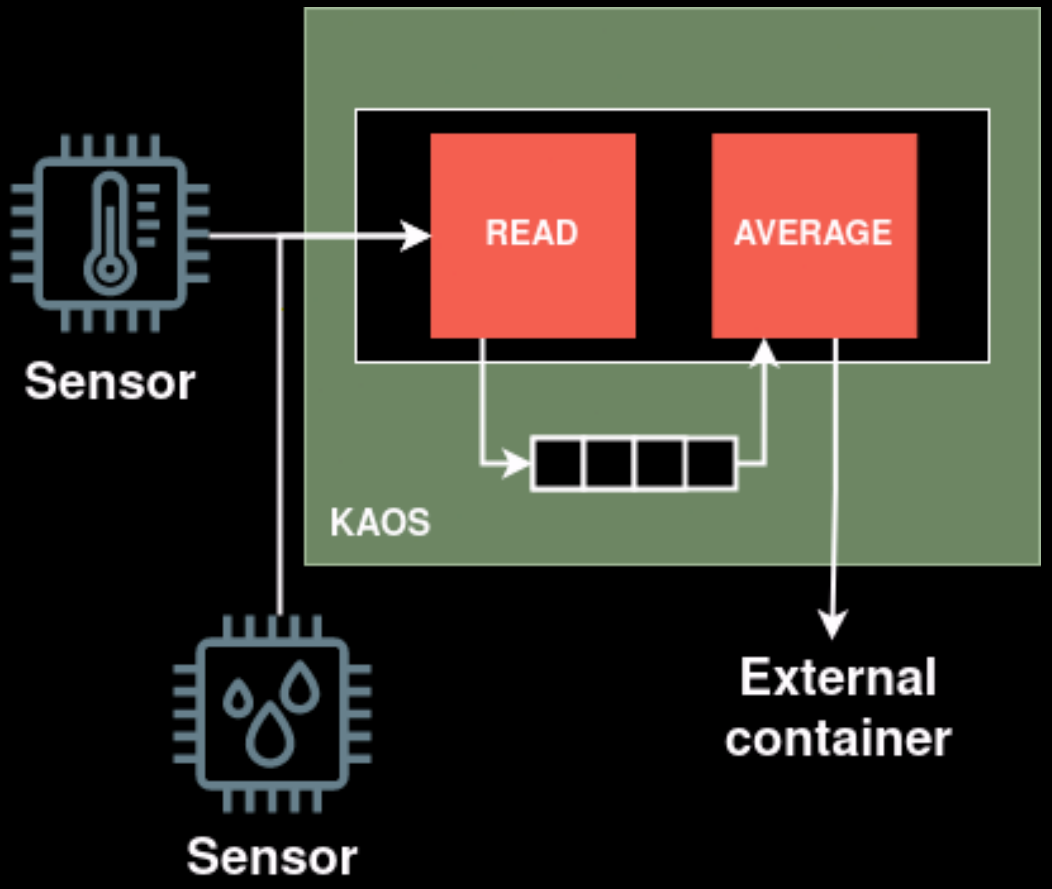
Independent portable containers

BUT limited interface – system calls



DEVICE PLATFORM

Containers need access to external resources



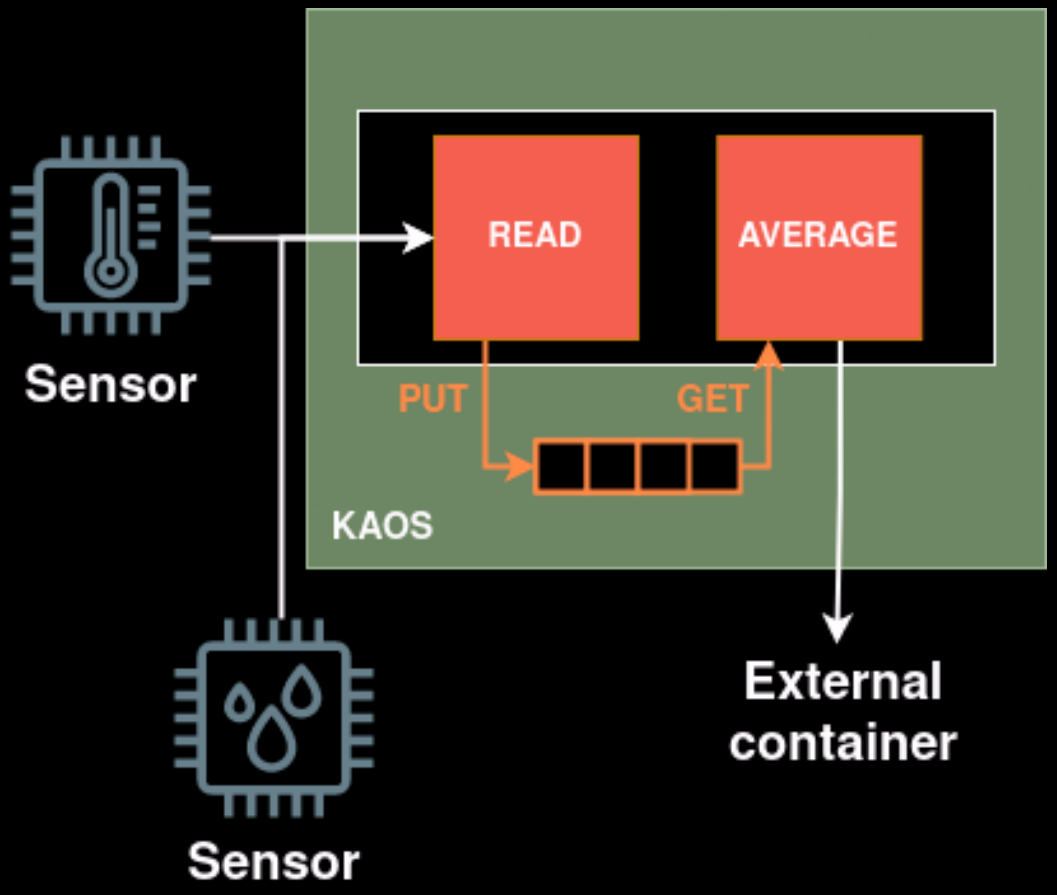
DEVICE PLATFORM

Container management

Containers need access to external resources

Message channels

Restricted API



COMMUNICATION CHANNELS

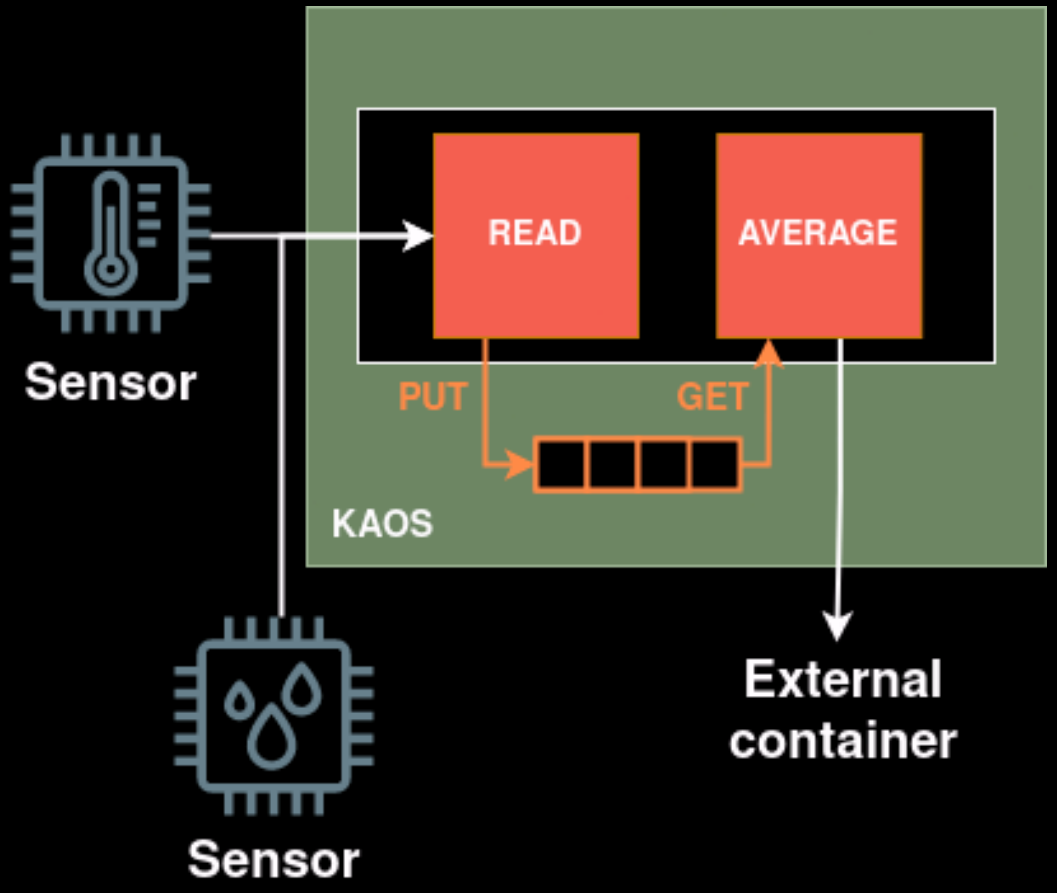
Interface: simple, future-proof

Expose communication services

Limit direct network access

Limit fate sharing - live when application dies

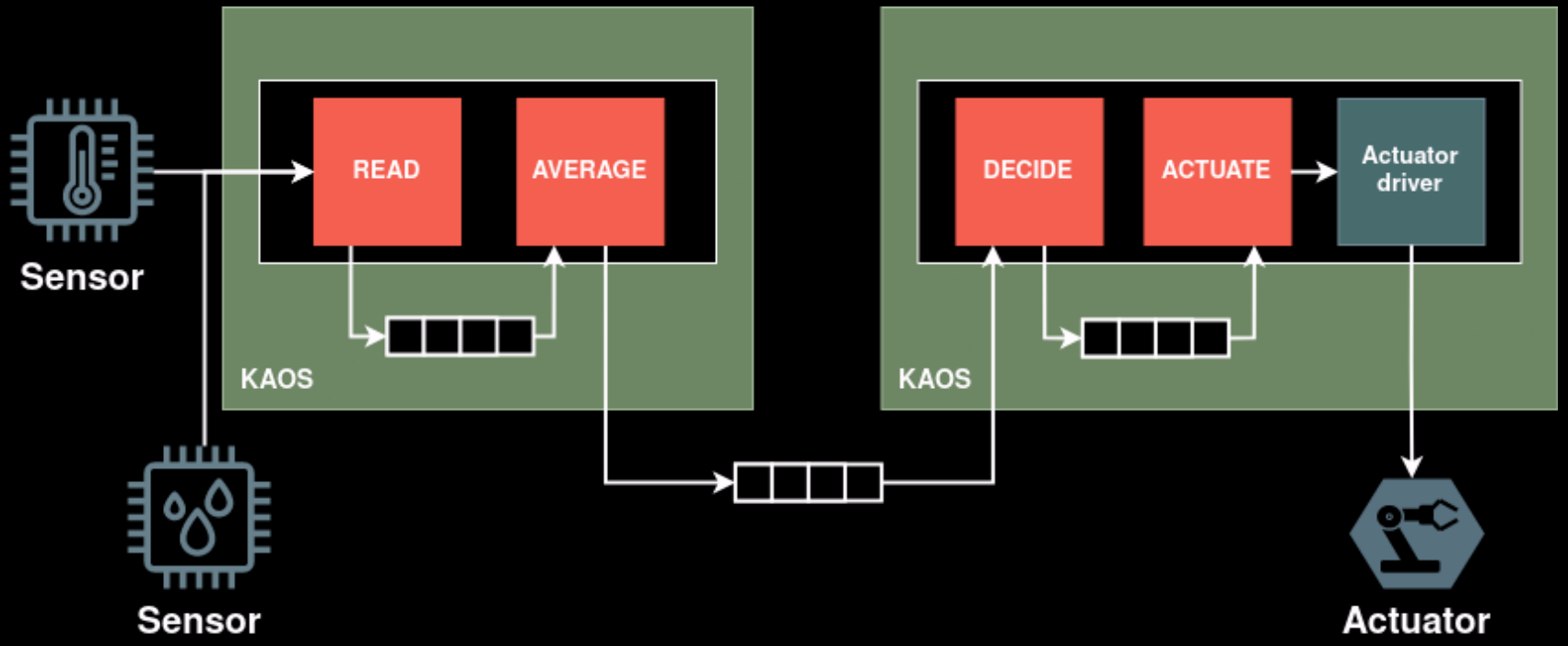
Limit disruption by events and restore service

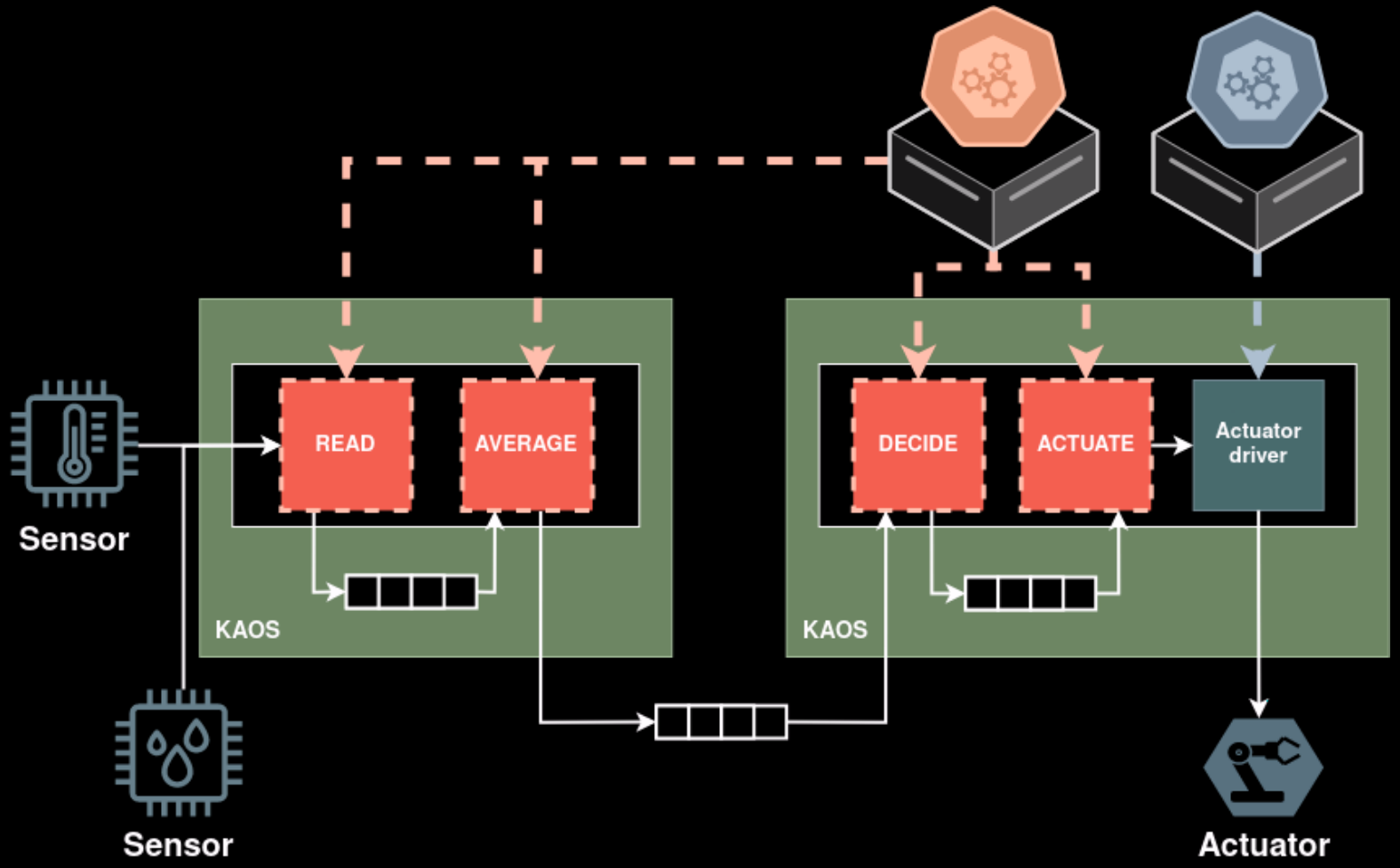


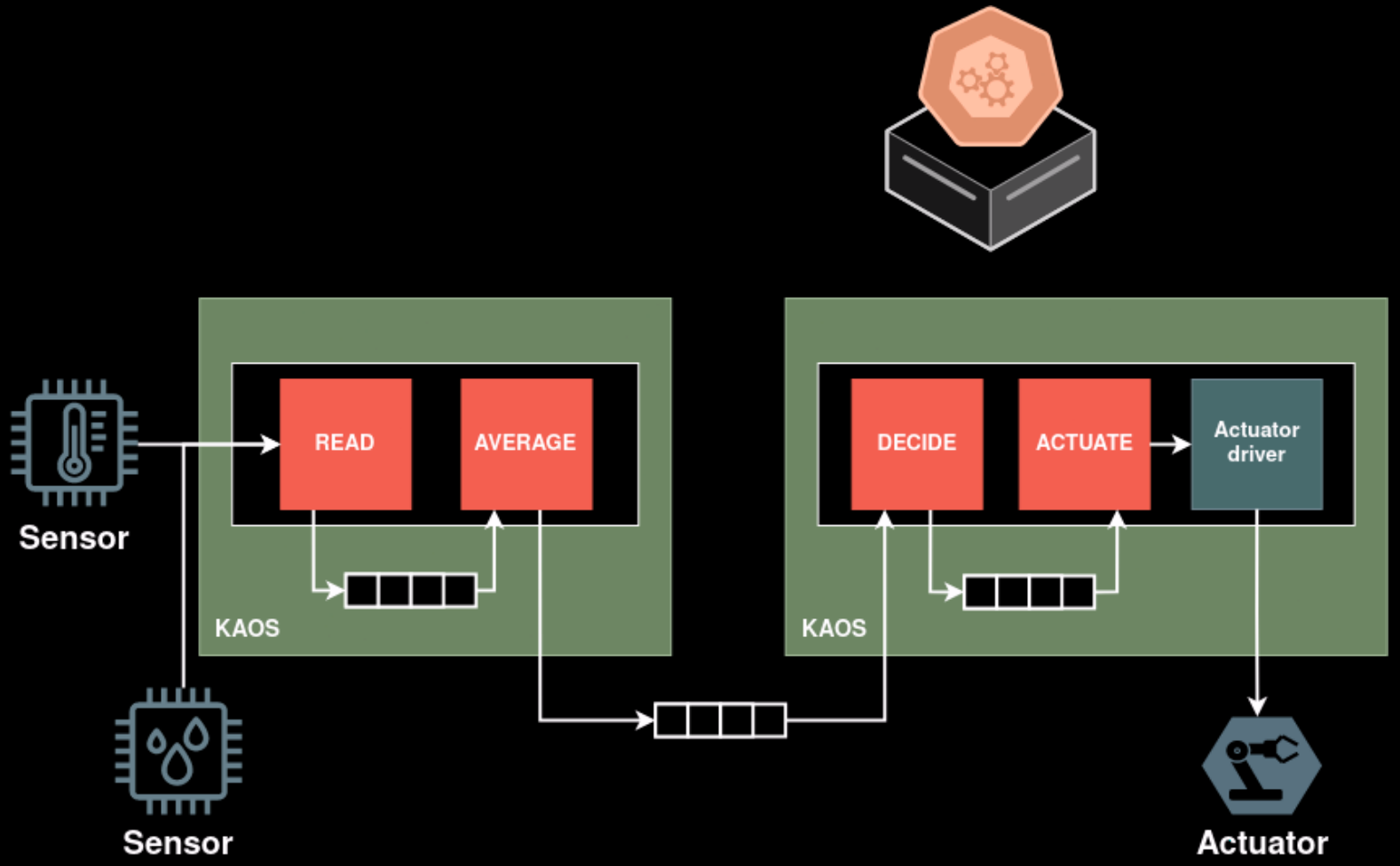
DYNAMIC COMPONENT PLACEMENT

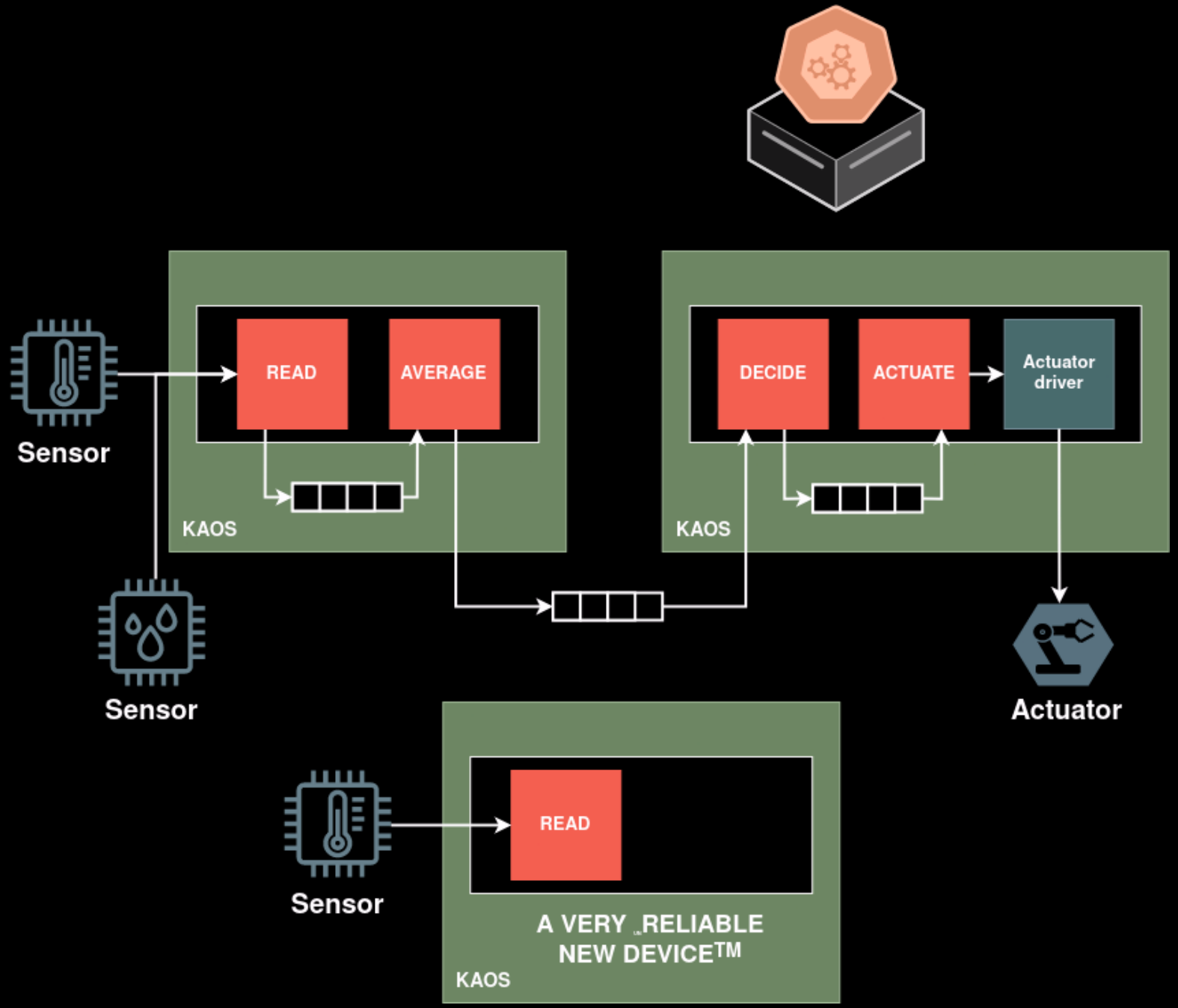
Respond to changes in the system

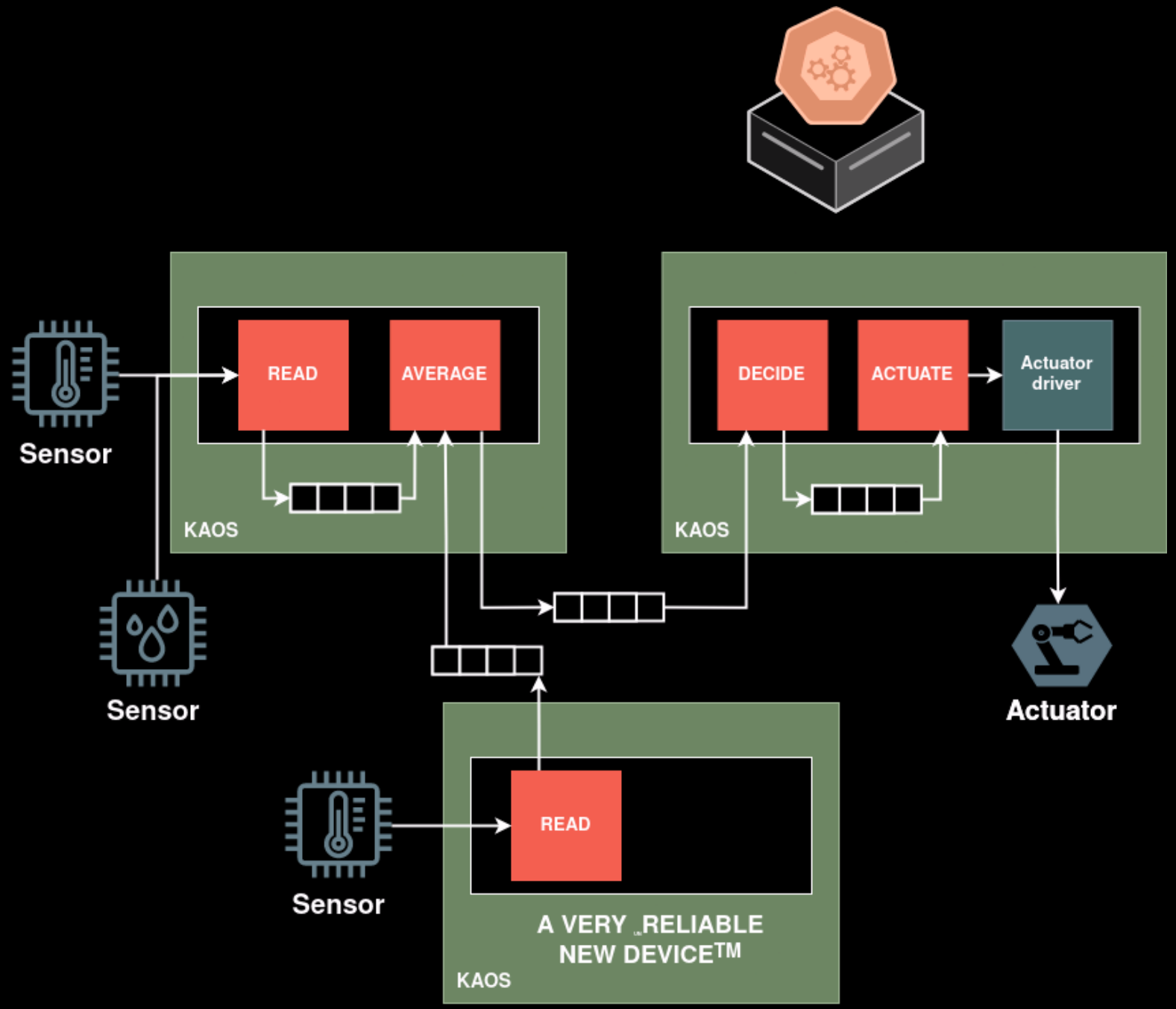
Tolerate unreliable hardware and links

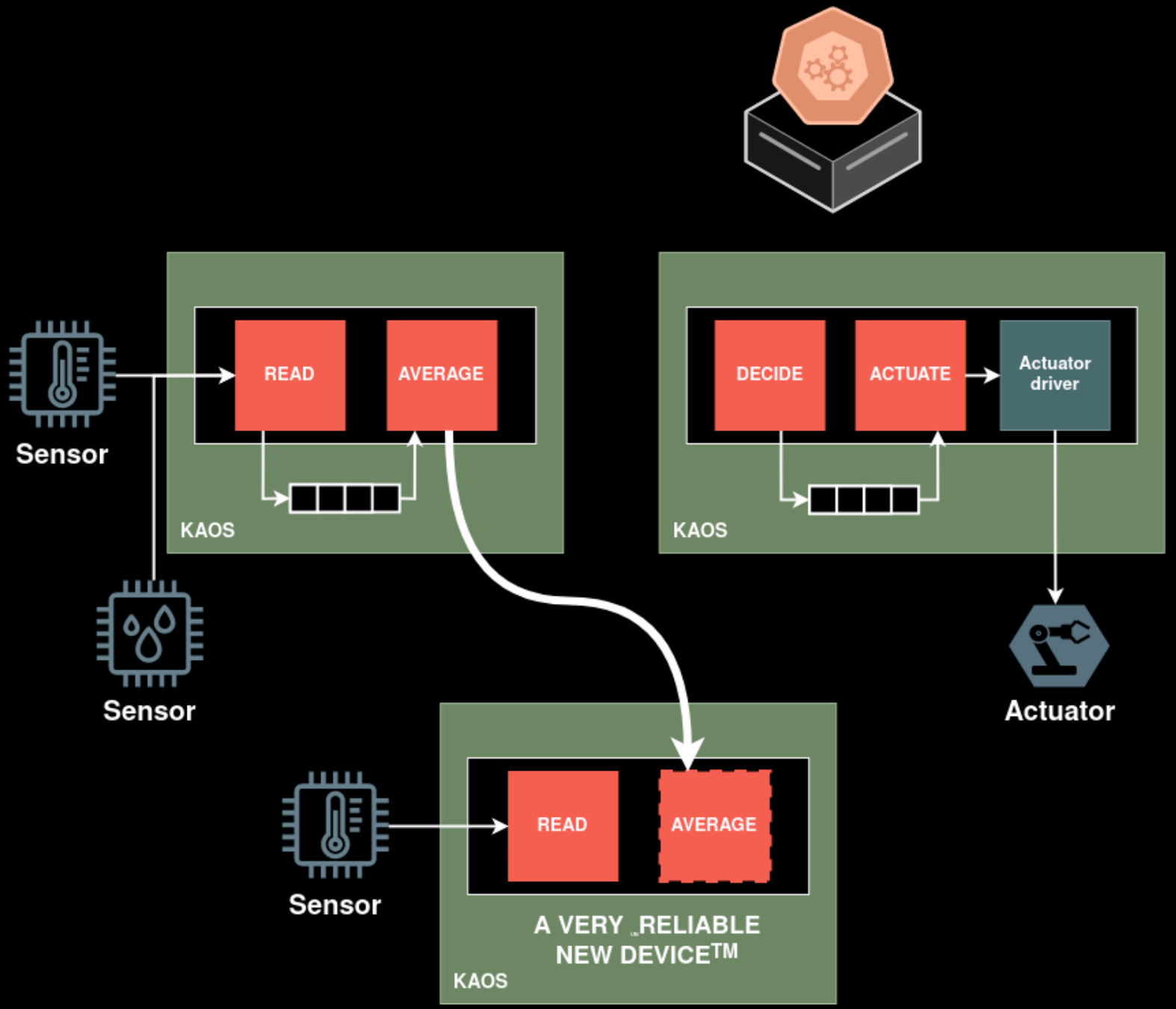


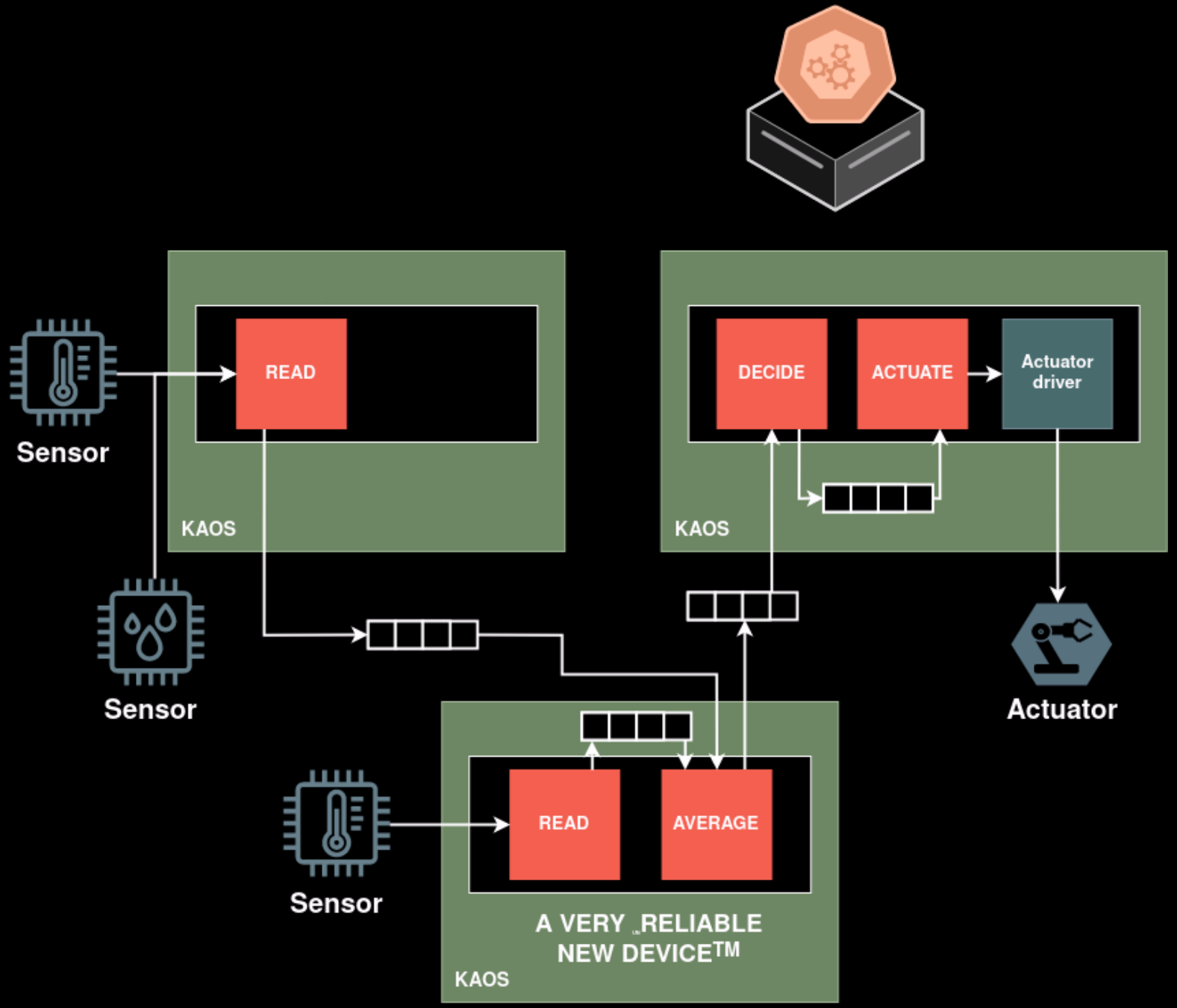

READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator

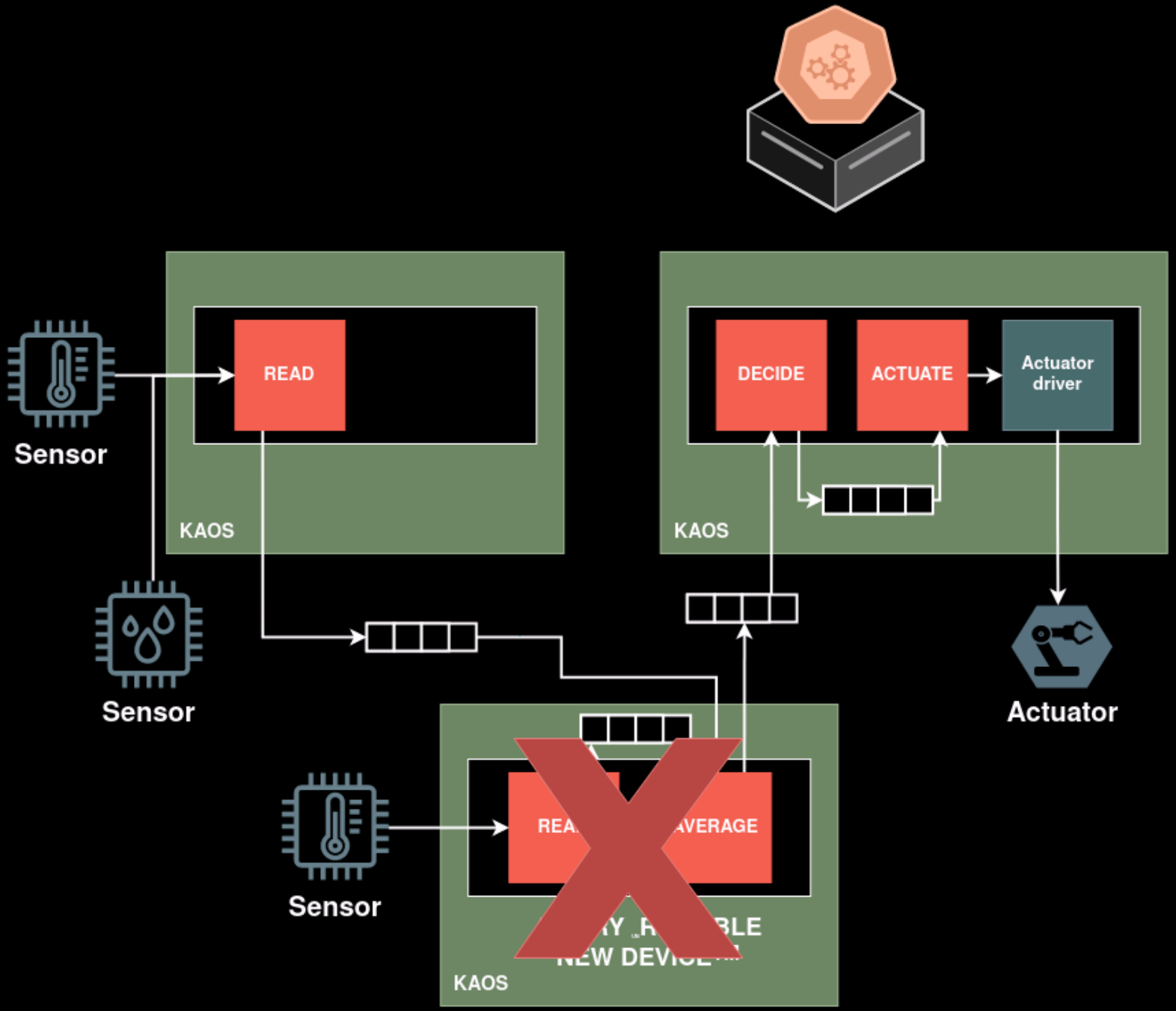

READ
AVERAGE
DECIDE
ACTUATE
Actuator driver
KAOS
KAOS
Sensor
Sensor
Actuator

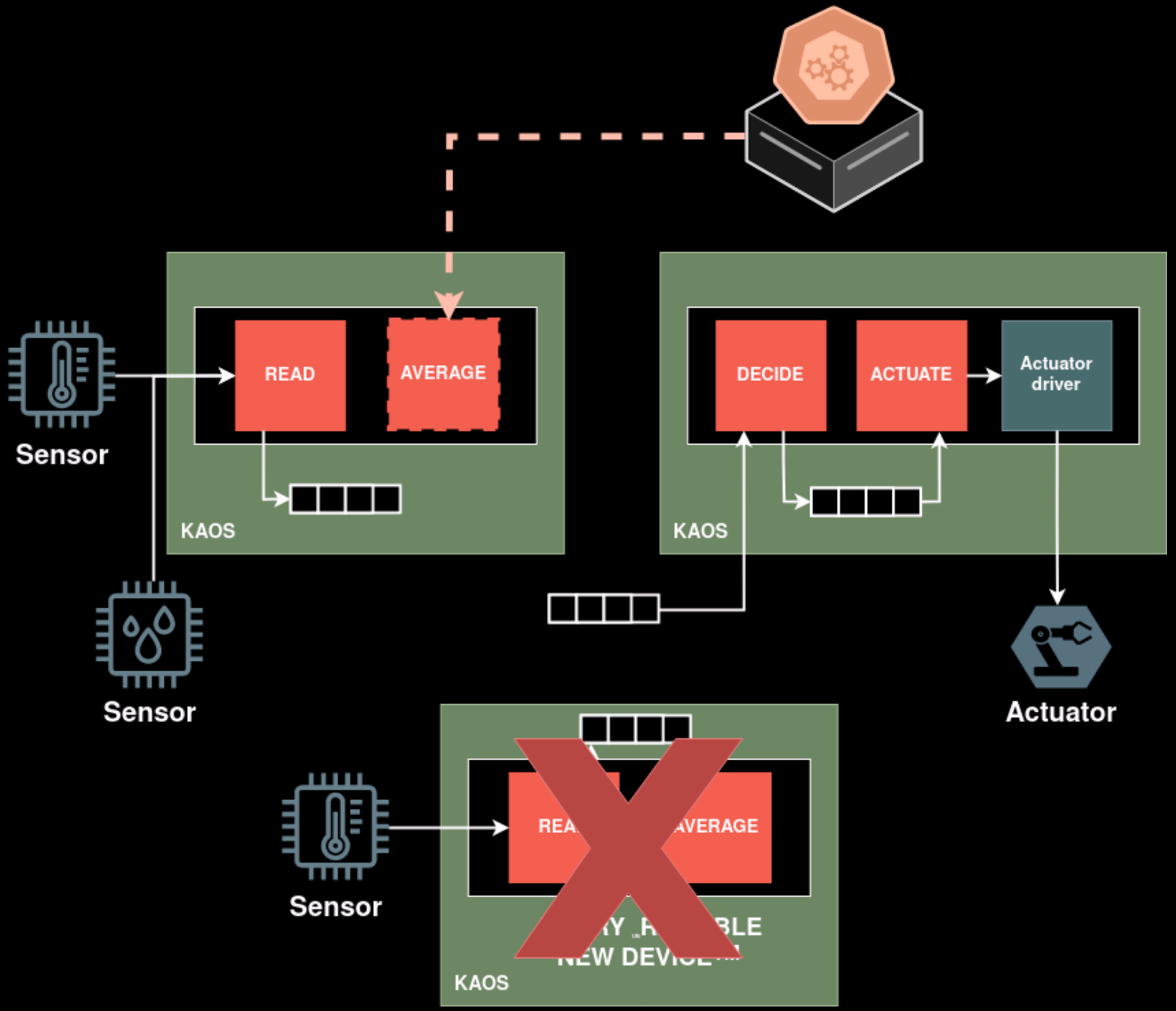

READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator
READ
A VERY RELIABLE NEW DEVICE™
KAOS
Sensor

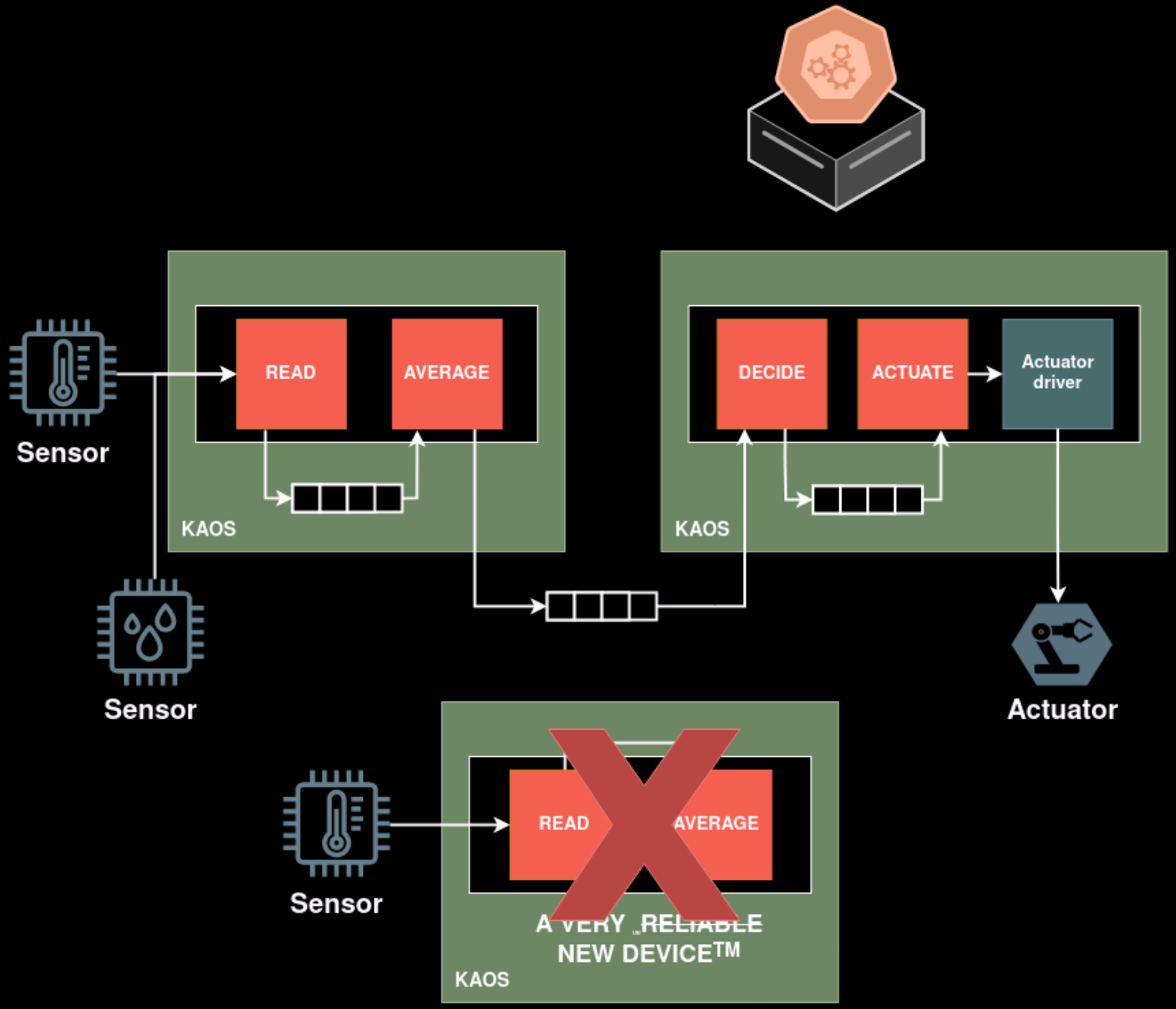

READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator
READ
Sensor
A VERY RELIABLE NEW DEVICE™
KAOS


READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator
READ
AVERAGE
Sensor
A VERY RELIABLE NEW DEVICE™
KAOS


READ
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator
READ
AVERAGE
Sensor
A VERY RELIABLE NEW DEVICE™
KAOS


READ
DECIDE
ACTUATE
Actuator driver
Sensor
Sensor
Sensor
Actuator
KAOS
KAOS
KAOS

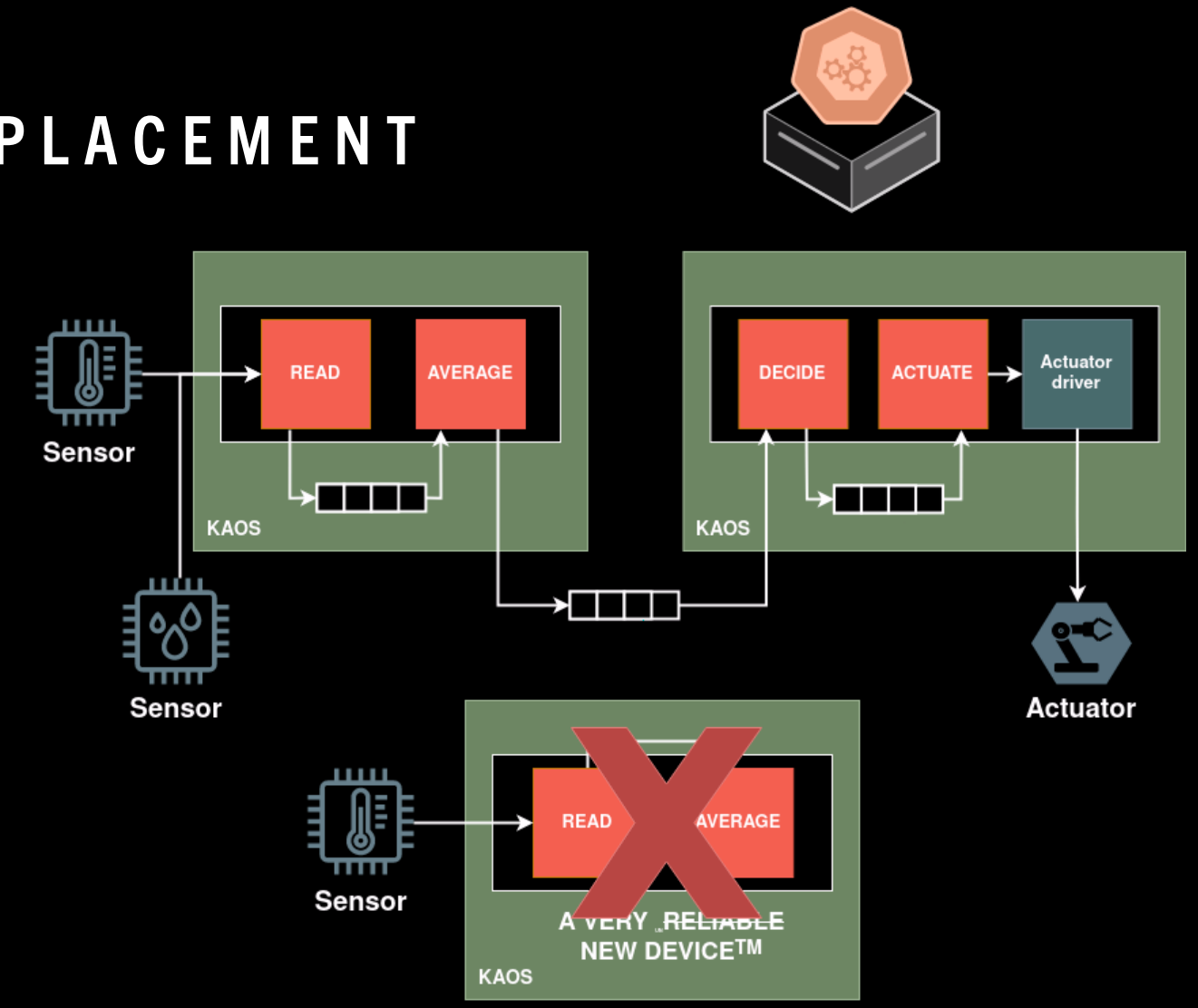

READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator
Sensor
KAOS


READ
AVERAGE
KAOS
DECIDE
ACTUATE
Actuator driver
KAOS
Sensor
Sensor
Actuator
READ
AVERAGE
A VERY RELIABLE
NEW DEVICE™
KAOS
Sensor

DYNAMIC COMPONENT PLACEMENT

Respond to changes in the system

Tolerate unreliable hardware and links



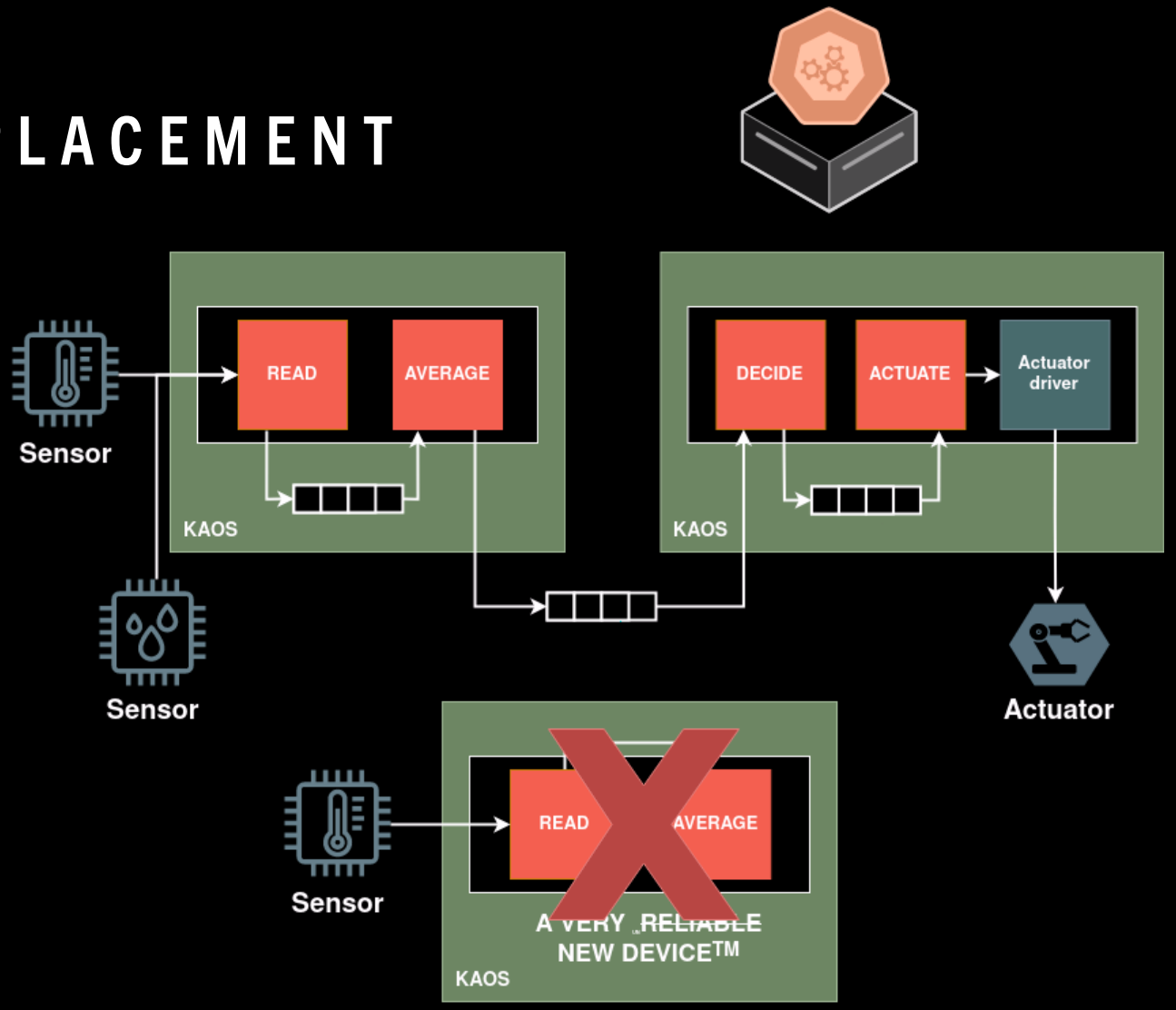
DYNAMIC COMPONENT PLACEMENT

Respond to changes in the system

Tolerate unreliable hardware and links

RESILIENCE

EVOLVABILITY



SECURE & RESILIENT IOT SOFTWARE NEEDS TO OUTLIVE THE HARDWARE

We need:

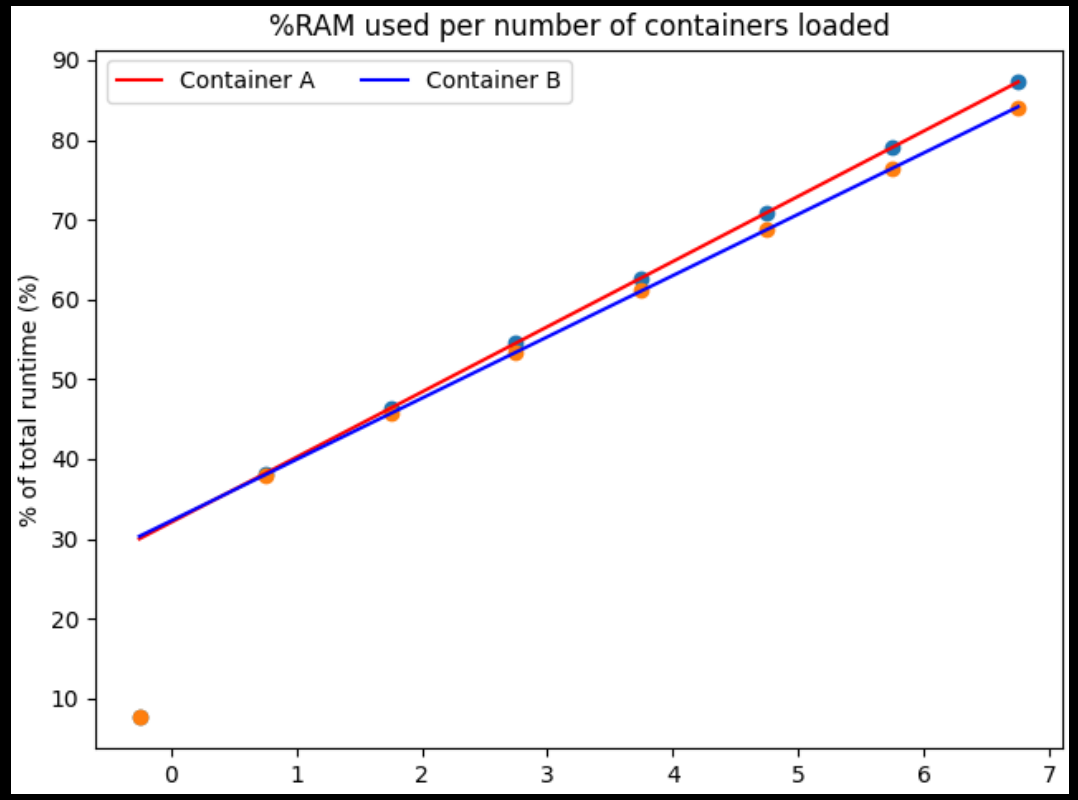
- **Componentization**
  - Independently evolving isolated control tasks and its mapping to hardware
  - Managed communication via message channels
- **Portability**
  - Migrating compiled code between different hardware platforms
- **Dynamic control graph adjustment**
  - Maintaining functionality in presence of failed or compromised hardware, flaky networks, requirement changes

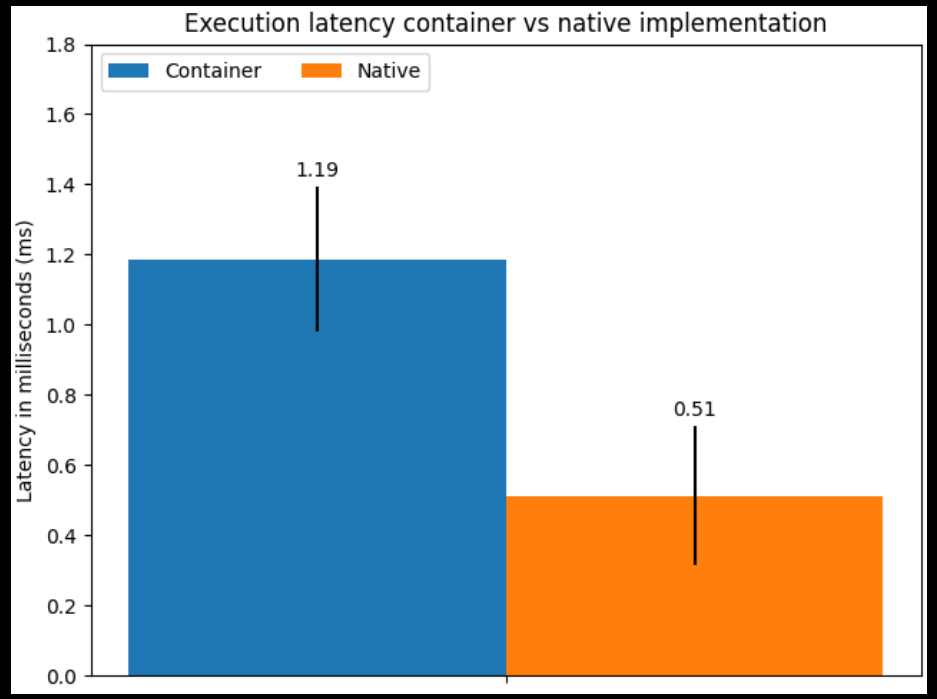
CALL TO ACTION: DEVIL IS IN THE DETAILS

INTERFACES!

- Managed communication: what services to support necessary interactions
- Control abstraction layer
- What should be the containerisation granularity
  - Supporting gradual deployment and legacy
- Orchestration, optimisation, mitigation strategies
- Development tools

FEASIBILITY TESTING





WHY WOULD BUSINESS-PERSON IN THEIR RIGHT MIND DO THIS?

- Existing demand-side interest
- Interest open-source solutions
- Aligning interests between parties
- Industrial building automation
  - Ubiquitous systems need to be long lasting
  - Serious vendor lock in issues

RUNNING THIS FOR DECADES?? I GET A NEW APPLE SMART KETTLE EACH YEAR

- But what about a boiler?
- Ease of maintenance
- Industrial building automation
- Ubiquitous
  - Becoming part of the building fabric

CONCERNS AND LIMITATIONS

- Sleeping devices
- Performance metrics
- Platform portability
  - Runtime
- API
  - Getting the balance right
  - Focusing the application to the right issues without constraining functionality and interoperability