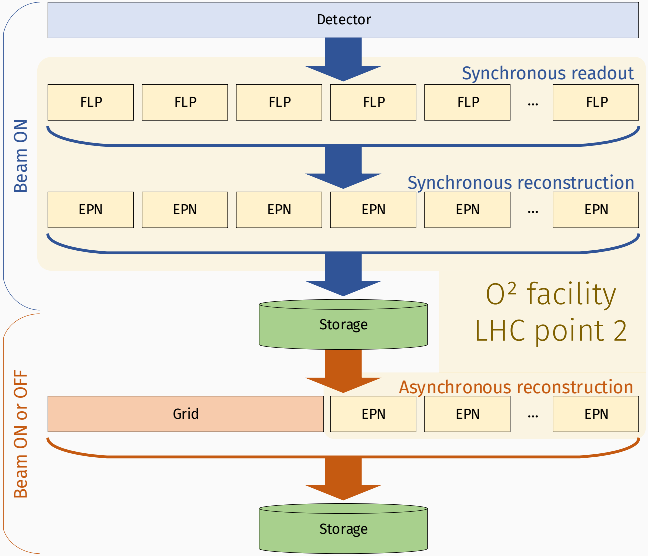
AliECS

A New Experiment Control System for the ALICE Experiment

Teo Mrnjavac
CERN EP-AID-DA
on behalf of the ALICE O²/FLP project

4 November 2019

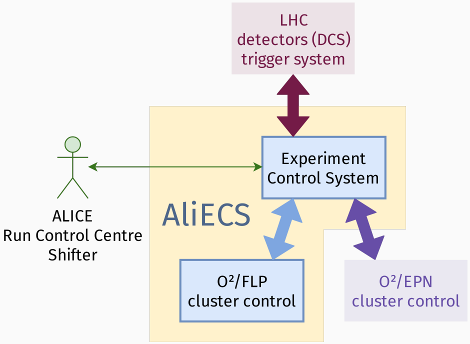
The ALICE Online-Offline computing system



- Multiprocess **data flow and processing** framework
- **100 000s of processes**, ~1000 machines
- **Synchronous and asynchronous** (grid-like) workflows
- One computing system, 2 types of node arranged in 2 clusters:

  FLP - First Level Processors
  EPN - Event Processing Nodes
- Operations will start in **2021**

ECS and O² cluster control



- Manage the lifetime of thousands of **stateful processes** in the O²/FLP cluster (control of O²/EPN delegated to a specialized O²/EPN cluster control)
- Minimize the waste of beam time by reusing processes and avoiding time-consuming process restart operations
- Interface with the LHC, the trigger system, the Detector Control System and other systems through common APIs

Managing a cluster with Apache Mesos

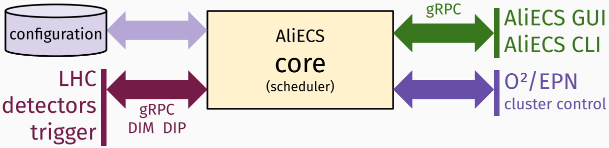
“Program against your datacenter like it’s a single pool of resources.”

Managing a cluster with Apache Mesos

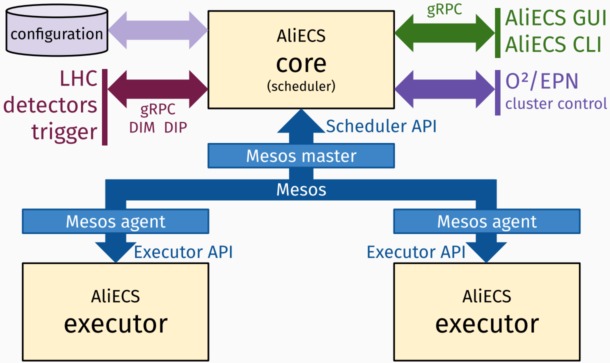
"Program against your datacenter like it's a single pool of resources."

- Mesos acts as a **distributed execution environment**
  which streamlines how AliECS manages its components, resources and tasks inside the O²/FLP farm.
- Benefits:
  - **knowledge** of what runs where,
  - **resource management** (ports, CPU, RAM, …),
  - **transport** for control messages,
  - task event **notification** (dead, failed to launch, …),
  - node attributes, high availability, checkpointing, …

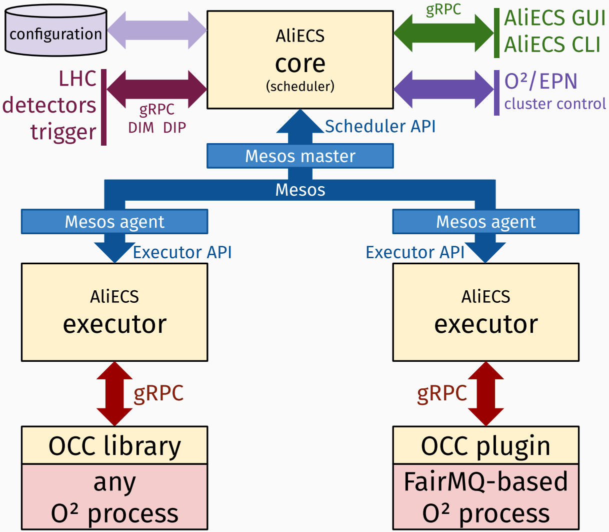
AliECS in a nutshell



AliECS in a nutshell



AliECS in a nutshell



AliECS in a nutshell

- Components:
  - AliECS core (incl. Apache Mesos scheduler) MESOS GO GRPC
  - AliECS executor MESOS GO GRPC
  - AliECS control and configuration utility (`coconut`) GO GRPC
  - Single process state machine debug utility (`peanut`) GO GRPC
  - O^2 control and configuration FairMQ plugin (`FairMQPlugin_OCC`) FAIR GRPC
  - O^2 control and configuration library (`libOCC`) GRPC
- Also available:
  - The web-based AliECS GUI node GRPC
  - AliECS deployment mechanism ANSIBLE

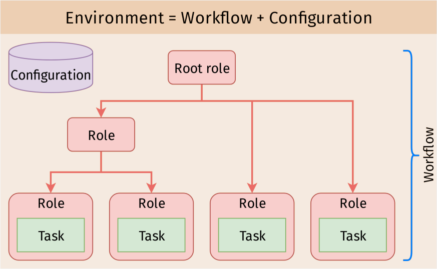
AliECS in a nutshell

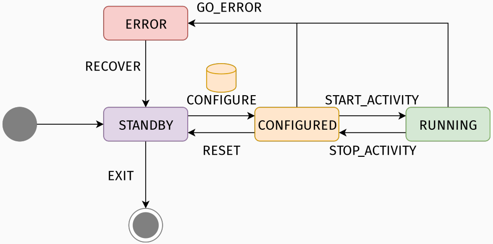
- Components:
  - AliECS core (incl. Apache Mesos scheduler) MESOS GO GRPC
  - AliECS executor MESOS GO GRPC
  - AliECS control and configuration utility (`coconut`) GO GRPC
  - Single process state machine debug utility (`peanut`) GO GRPC
  - O^2 control and configuration FairMQ plugin (`FairMQPlugin_OCC`) FAIR C++ GRPC
  - O^2 control and configuration library (`libOCC`) C++ GRPC
- Also available:
  - The web-based AliECS GUI node GRPC
  - AliECS deployment mechanism ANSIBLE

AliECS in a nutshell

- Components:
  - AliECS core (incl. Apache Mesos scheduler)
  - AliECS executor
  - AliECS control and configuration utility (`coconut`)
  - Single process state machine debug utility (`peanut`)
  - O^2 control and configuration FairMQ plugin (`FairMQPlugin_OCC`)
  - O^2 control and configuration library (`libOCC`)
- Also available:
  - The web-based AliECS GUI
  - AliECS deployment mechanism

AliECS concepts



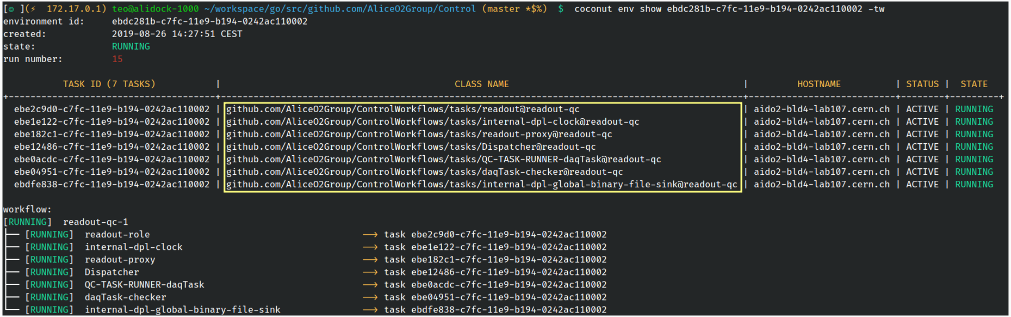


- AliECS schedules, configures and controls **tasks** (stateful processes)
- Each **role** represents either a task, or its own child roles
- A tree of roles is a **workflow**
- Tasks, roles and environments have their own **state machines**
- An environment in **RUNNING** state is granted a unique **run number** which remains valid until the **RUNNING** state exits

AliECS workflow and task configuration

- Based on **Git**, multiple repositories per AliECS instance
- Task descriptors and workflow templates are **YAML** (plus template system)
- Once loaded, every task type and workflow is **uniquely identified** by

  git repository + task/workflow file name + git revision



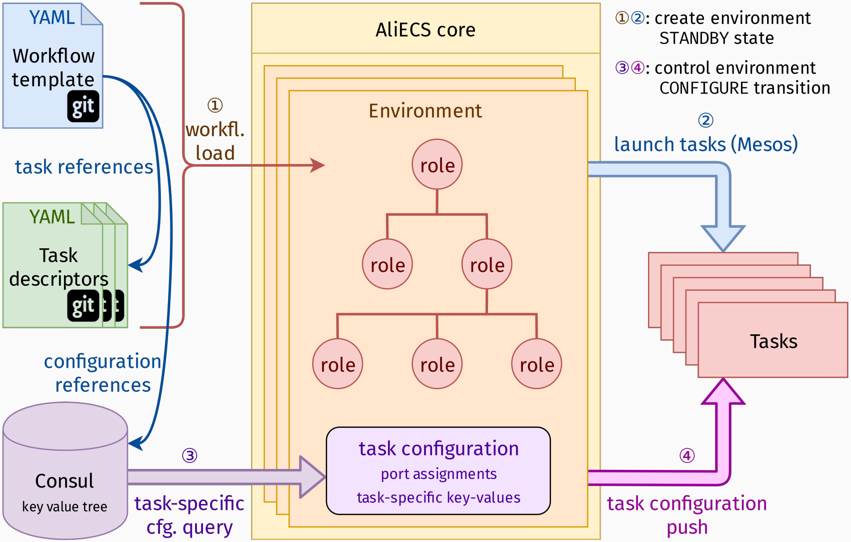
```
[☻ ](⚡  172.17.0.1) teo@alidock-1000 ~/workspace/go/src/github.com/AliceO2Group/Control (master *$%)  $  coconut env show ebdc281b-c7fc-11e9-b194-0242ac110002 -tw
environment id:     ebdc281b-c7fc-11e9-b194-0242ac110002
created:            2019-08-26 14:27:51 CEST
state:              RUNNING
run number:         15

           TASK ID (7 TASKS)           |                                           CLASS NAME                                           |        HOSTNAME            | STATUS |  STATE
+--------------------------------------+------------------------------------------------------------------------------------------------+---------------------------+--------+---------+
  ebe2c9d0-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/readout@readout-qc                              | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING
  ebe1e122-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/internal-dpl-clock@readout-qc                   | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING
  ebe182c1-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/readout-proxy@readout-qc                        | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING
  ebe12486-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/Dispatcher@readout-qc                           | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING
  ebe0acdc-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/QC-TASK-RUNNER-daqTask@readout-qc               | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING
  ebe04951-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/daqTask-checker@readout-qc                      | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING
  ebdfe838-c7fc-11e9-b194-0242ac110002 | github.com/AliceO2Group/ControlWorkflows/tasks/internal-dpl-global-binary-file-sink@readout-qc | aido2-bld4-lab107.cern.ch | ACTIVE | RUNNING

workflow:
[RUNNING]  readout-qc-1
├── [RUNNING]  readout-role                                       ⟶ task ebe2c9d0-c7fc-11e9-b194-0242ac110002
├── [RUNNING]  internal-dpl-clock                                 ⟶ task ebe1e122-c7fc-11e9-b194-0242ac110002
├── [RUNNING]  readout-proxy                                      ⟶ task ebe182c1-c7fc-11e9-b194-0242ac110002
├── [RUNNING]  Dispatcher                                         ⟶ task ebe12486-c7fc-11e9-b194-0242ac110002
├── [RUNNING]  QC-TASK-RUNNER-daqTask                             ⟶ task ebe0acdc-c7fc-11e9-b194-0242ac110002
├── [RUNNING]  daqTask-checker                                    ⟶ task ebe04951-c7fc-11e9-b194-0242ac110002
└── [RUNNING]  internal-dpl-global-binary-file-sink               ⟶ task ebdfe838-c7fc-11e9-b194-0242ac110002
```

Documentation: https://github.com/AliceO2Group/Control/blob/master/coconut/doc/coconut_repository.md

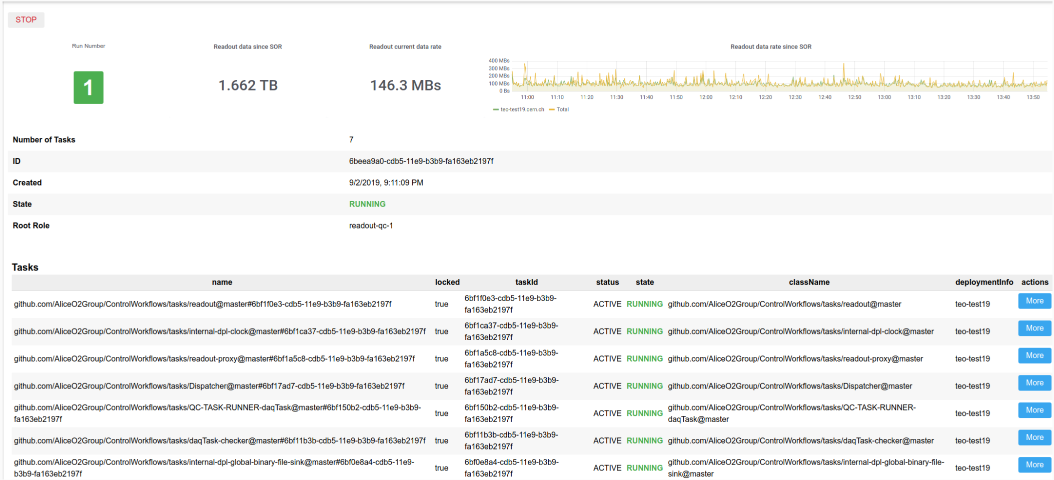
AliECS workflow and task configuration



①②: create environment
STANDBY state

③④: control environment
CONFIGURE transition

AliECS GUI



Run Number	Readout data since SOR	Readout current data rate
1	1.662 TB	146.3 MBs

Number of Tasks	7
ID	6beea9a0-cdb5-11e9-b3b9-fa163eb2197f
Created	9/2/2019, 9:11:09 PM
State	RUNNING
Root Role	readout-qc-1

Tasks

name	locked	taskId	status	state	className	deploymentInfo	actions
github.com/AliceO2Group/ControlWorkflows/tasks/readout@master#6bf1f0e3-cdb5-11e9-b3b9-fa163eb2197f	true	6bf1f0e3-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/readout@master	teo-test19	More
github.com/AliceO2Group/ControlWorkflows/tasks/internal-dpl-clock@master#6bf1ca37-cdb5-11e9-b3b9-fa163eb2197f	true	6bf1ca37-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/internal-dpl-clock@master	teo-test19	More
github.com/AliceO2Group/ControlWorkflows/tasks/readout-proxy@master#6bf1a5c8-cdb5-11e9-b3b9-fa163eb2197f	true	6bf1a5c8-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/readout-proxy@master	teo-test19	More
github.com/AliceO2Group/ControlWorkflows/tasks/Dispatcher@master#6bf17ad7-cdb5-11e9-b3b9-fa163eb2197f	true	6bf17ad7-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/Dispatcher@master	teo-test19	More
github.com/AliceO2Group/ControlWorkflows/tasks/QC-TASK-RUNNER-daqTask@master#6bf150b2-cdb5-11e9-b3b9-fa163eb2197f	true	6bf150b2-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/QC-TASK-RUNNER-daqTask@master	teo-test19	More
github.com/AliceO2Group/ControlWorkflows/tasks/daqTask-checker@master#6bf11b3b-cdb5-11e9-b3b9-fa163eb2197f	true	6bf11b3b-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/daqTask-checker@master	teo-test19	More
github.com/AliceO2Group/ControlWorkflows/tasks/internal-dpl-global-binary-file-sink@master#6bf0e8a4-cdb5-11e9-b3b9-fa163eb2197f	true	6bf0e8a4-cdb5-11e9-b3b9-fa163eb2197f	ACTIVE	RUNNING	github.com/AliceO2Group/ControlWorkflows/tasks/internal-dpl-global-binary-file-sink@master	teo-test19	More

AliECS GUI

- As few dependencies as possible to facilitate maintenance
- **Node.js** with Express.js as server framework
- `grpc/proto-loader` and `grpc` for communication with AliECS core
- UI and other components built from scratch and exported as `npm` module for look&feel consistency across O^2/FLP interfaces
- Puppeteer for integration tests
- Kafka-node for displaying browser notifications via common Notification Service

Conclusions

- The new ALICE O² computing system requires a **new control system**
- AliECS carries both **ECS** and **O²/FLP cluster control** duties
- Opportunity to leverage technologies such as **Mesos** and **Go** for a high performance, low latency ECS
  - Mesos gives us resource management, transport and much more
  - Minimize waste of beam time
  - Improved operational flexibility

AliECS on GitHub: github.com/AliceO2Group/Control
Configuration examples: github.com/AliceO2Group/ControlWorkflows

Backup slides

Target improvements

- Improved flexibility & latency:
  - **no workflow redeployment** when excluding/including a detector from data taking,
  - **recover** from process and server crashes,
  - **reconfigure** processes without restart,
  - **scale** workflows based on immediate needs.
- Next gen web-based GUIs with SSO & **revamped design**.
- Take advantage of modern developments in computing.

Why Go?

- Go is a statically typed general-purpose programming language in the tradition of C.
- 100% Free and open source.
- Prominent features include:
    - simple syntax and excellent readability,
    - garbage collection,
    - interface system and composition, but no inheritance,
    - lightweight processes (goroutines) and channels,
    - build system and remote package management included in compiler,
    - fast compilation,
    - statically linked native binaries.
- Go is already used in some components of the O² stack, including Consul, Docker and InfluxDB.

gRPC in AliECS

- **gRPC**, an RPC system based on Protobuf was chosen as the lingua franca of AliECS IPC:
  - backed by Google,
  - multi-language support,
  - already packaged for O²,
  - widely used in the microservices community.
- In AliECS, gRPC is used for
  - communication between the AliECS core and the GUI,
  - communication between the executor and the OCC plugin.
- Higher performance and better multi-language integration compared to REST (Swagger, etc.)
- Better interoperability and/or support/documentation compared to other RPC methods (JSON-RPC, MessagePack-RPC, `net/rpc`, Cap’n’Proto, etc.)

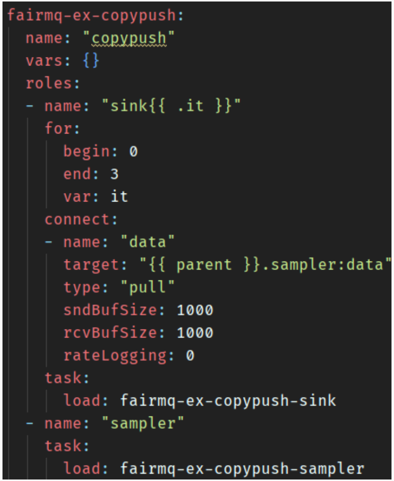
Why Apache Mesos? / Why not Kubernetes?

- Apache Mesos vs. Kubernetes is a false equivalence:
  - Apache Mesos is primarily a **cluster resource management system**
    - See also Marathon, Aurora, DC/OS...
  - Kubernetes is a **container orchestration platform**
    - “Opinionated software”: it enforces its own structure of Pods and Containers
- The benefits of Kubernetes + containerization are dubious at best in a heterogeneous environment such as the O²/FLP farm, which includes:
  - different configurations of FLP machines
  - custom PCIe hardware
  - physical point-to-point fiber links to detector front-end electronics.
- A **resource management system** with deployment functionality at the single process level such as **Apache Mesos** fits well with O²/FLP requirements.

A container orchestration platform wasn’t needed and still isn’t.

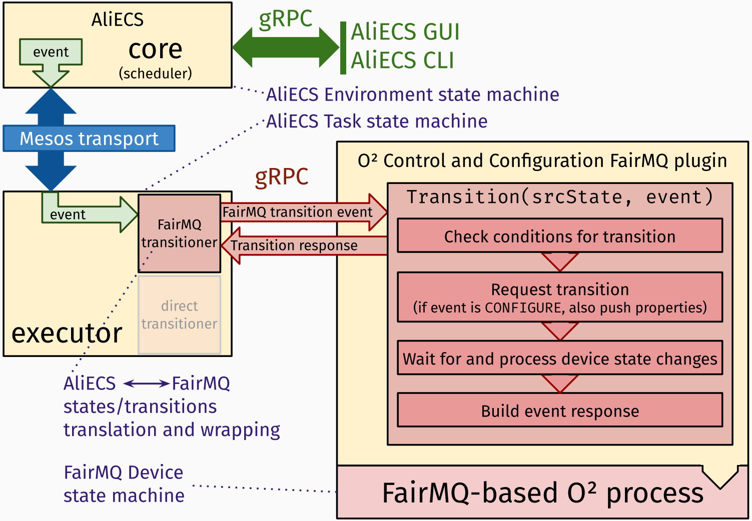
Workflows, roles and tasks

- Concepts:
  - **task** - the basic unit of control, generally 1 process
  - **role** - a node in the control tree, aggregates child roles and ultimately tasks
  - **workflow** - the in-memory control tree of an environment, made of roles which drive tasks
- Workflow templates generate workflows of tasks
  - Generated from DPL specs
  - Stored in O² configuration (YAML + Git)
  - Variables, iterators, internal references



```yaml
fairmq-ex-copypush:
  name: "copypush"
  vars: {}
  roles:
  - name: "sink{{ .it }}"
    for:
      begin: 0
      end: 3
      var: it
    connect:
    - name: "data"
      target: "{{ parent }}.sampler:data"
      type: "pull"
      sndBufSize: 1000
      rcvBufSize: 1000
      rateLogging: 0
    task:
      load: fairmq-ex-copypush-sink
  - name: "sampler"
    task:
      load: fairmq-ex-copypush-sampler
```

Control of FairMQ devices



AliECS core (scheduler)

event

gRPC

AliECS GUI

AliECS CLI

AliECS Environment state machine

AliECS Task state machine

Mesos transport

executor

event

FairMQ transitioner

direct transitioner

gRPC

FairMQ transition event

Transition response

O² Control and Configuration FairMQ plugin

```
Transition(srcState, event)
```

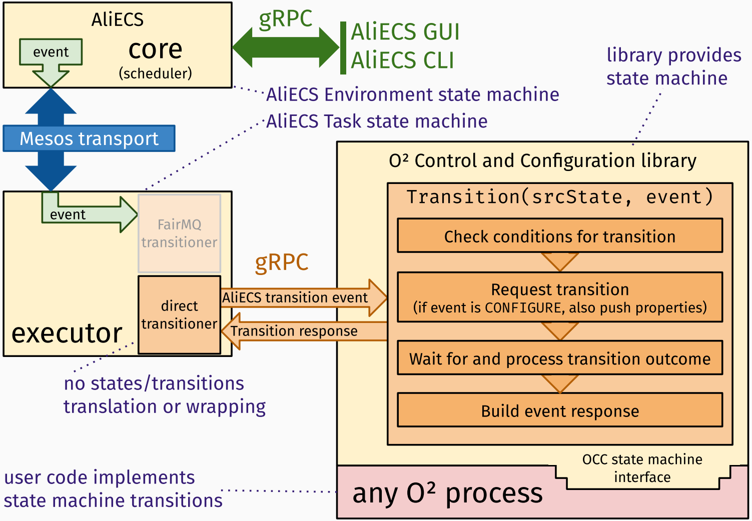
1. Check conditions for transition
2. Request transition (if event is `CONFIGURE`, also push properties)
3. Wait for and process device state changes
4. Build event response

FairMQ-based O² process

AliECS ⟷ FairMQ states/transitions translation and wrapping

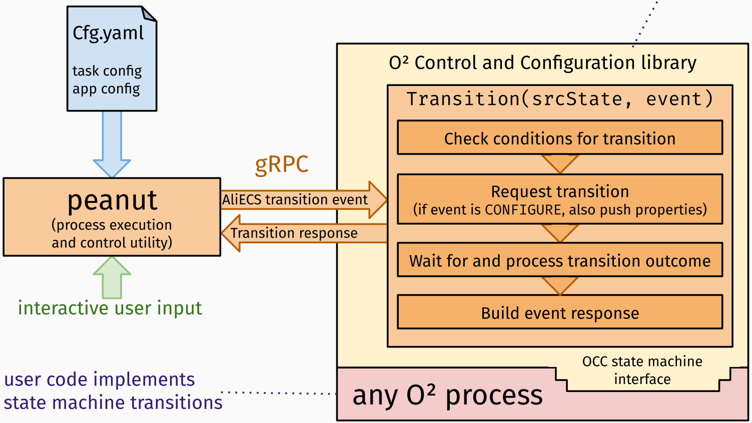
FairMQ Device state machine

Control of non-FairMQ O² processes (e.g. Readout)



Debug mode for non-FairMQ O² processes

Single process debug mode:
no AliECS core/executor
no Mesos



library provides
state machine

Cfg.yaml

task config
app config

peanut
(process execution
and control utility)

interactive user input

gRPC

AliECS transition event

Transition response

O² Control and Configuration library

`Transition(srcState, event)`

Check conditions for transition

Request transition
(if event is `CONFIGURE`, also push properties)

Wait for and process transition outcome

Build event response

OCC state machine
interface

any O² process

user code implements
state machine transitions