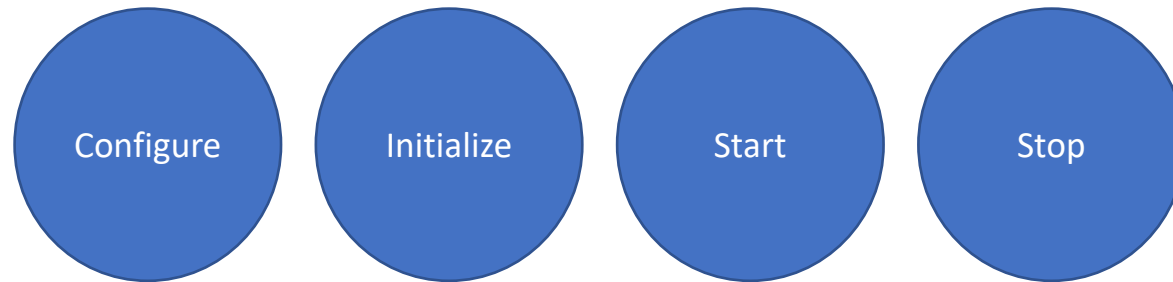


Processing farm control (EPN control)

M. Al-Turany, SDE Group, GSI/IT

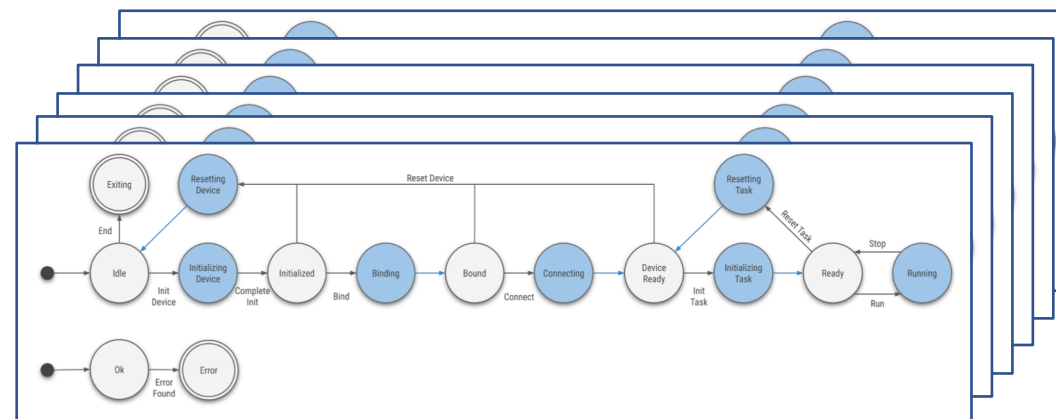
Controlling the processing farm with FairMQ devices:

- One has to make the **entire** cluster state available for the experiment control system and **not single process one**



Exported cluster state machine

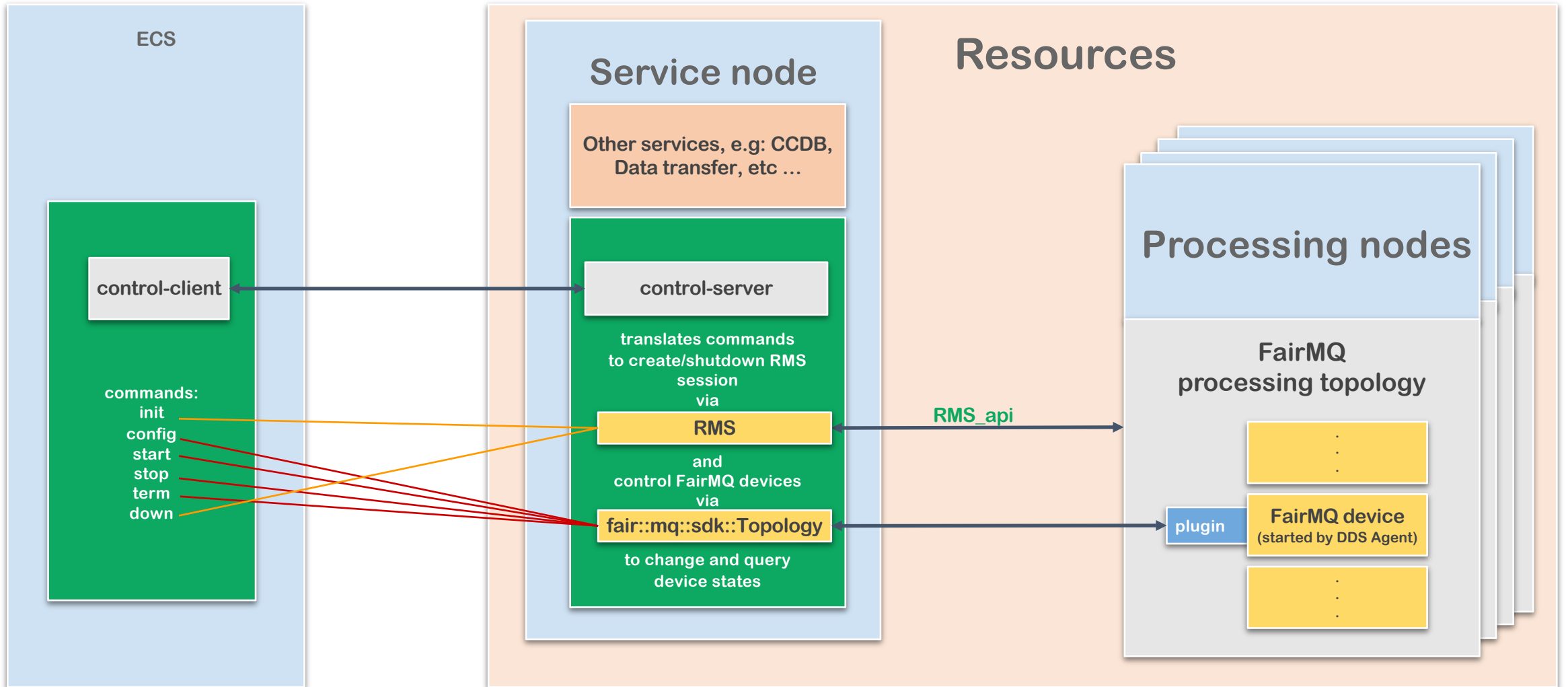
EPNs internal state machine (FairMQ)



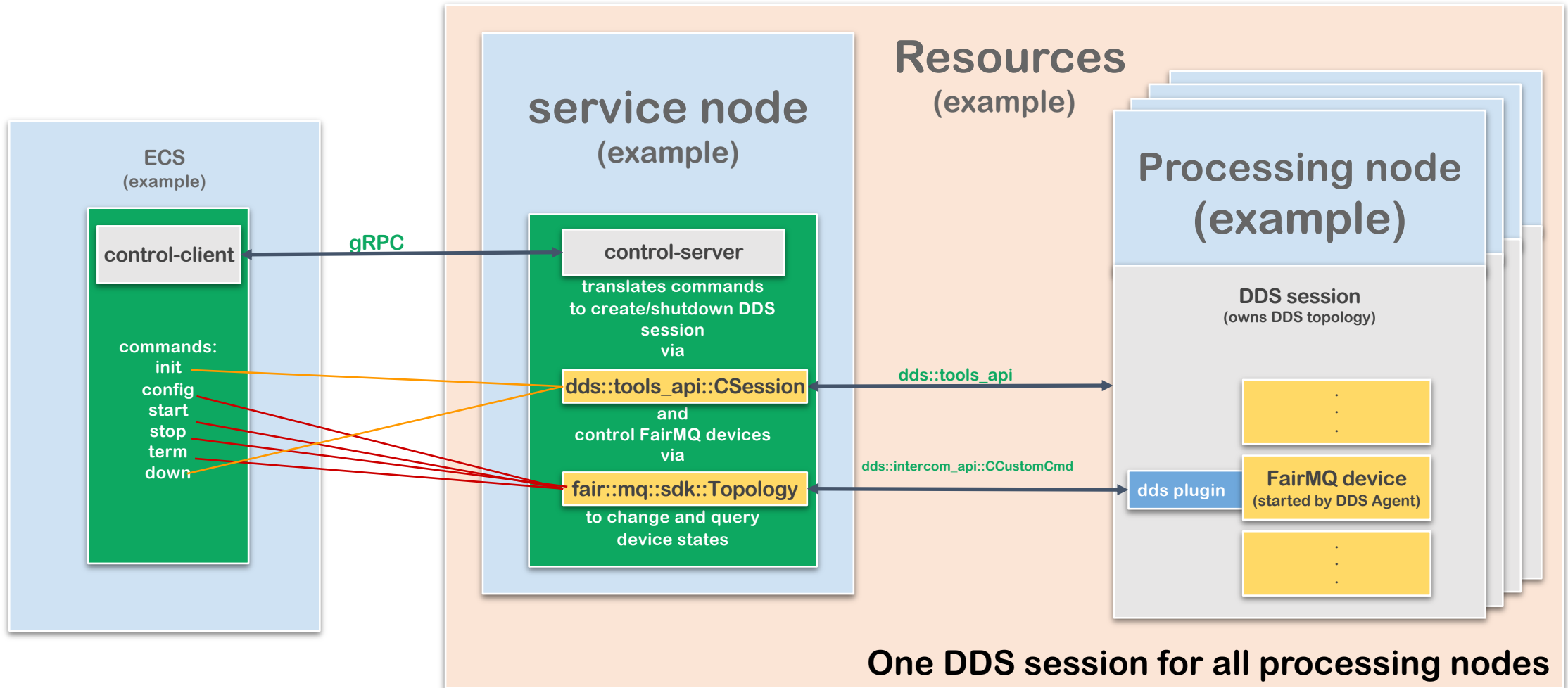
The process controller, should:

- Have the knowledge about the full topology of connected FairMQ devices.
- Launch/setup the run-time environment and the FairMQ devices
- Drive the device state machines in lock-step across the full topology
- Push the device configuration,
- Monitor (some aspects of the application's) operation and handling/reporting (some) error cases.

Controller Design



Controller Example (DDS based)



DDS-control

An example of how to control/communicate with a system backed by DDS and FairMQ.

FairRootGroup / DDS-control

Unwatch 3 Star 0 Fork 1

<> Code Issues 1 Pull requests 0 Projects 0 Wiki Security Insights

No description, website, or topics provided.

28 commits 1 branch 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

File/Folder	Description	Last Commit
AndreyLebedev	Create fairmq sdk topology only once	Latest commit f5108ca yesterday
cmake	Update project skeleton	last month
dds-control-server	Create fairmq sdk topology only once	yesterday
proto	Fix for google/protobuf/port_def.inc not found on macOS	last month
sample-client	Add RMS plugin and config file CLI arguments for dds-control-server	3 days ago
utils	Correct launchctl config	last month
.clang-format	Add git core files	2 months ago
.gitignore	Add git core files	2 months ago
CMakeLists.txt	Correct launchctl config	last month
README.md	Update README.md	3 days ago

README.md

DDS-control

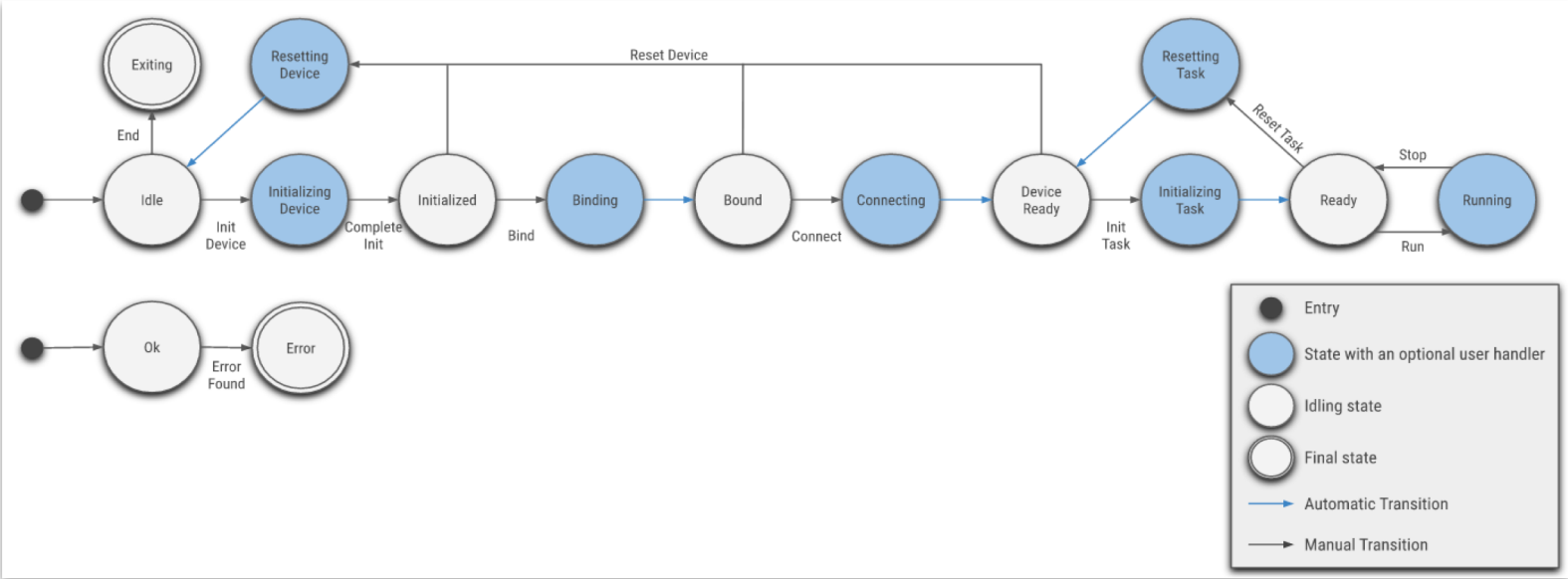
Introduction

DDS-control project is an example of how to control/communicate with a system backed by [DDS](#) and [FairMQ](#).

<https://github.com/FairRootGroup/DDS-control>

FairMQ State Machine & Example ECS Command Mapping

ECS command	DDS/FairMQ actions
init	DDS: Create session, submit agents, activate topology -> devices go in Idle state
configure	Devices: InitDevice->CompleteInit->Bind->Connect->InitTask
start	Devices: Run
stop	Devices: Stop
term	Devices: ResetTask->ResetDevice->End
down	DDS: Shutdown session



```
-- DEPENDENCY FOUND    VERSION                PREFIX
-- Boost               1.69 (>= 1.67)        /usr
-- DDS                 2.5.5.g3aa26d5 (>= 2.4) /home/dklein/projects/DDS/build/install
-- FairLogger         1.5.0 (>= 1.2.0)      /home/dklein/projects/FairLogger/build/install
--
-- COMPONENT          BUILT?  INFO
-- fairmq             NO      (enable with -DBUILD_FAIRMQ=ON)
-- tests              NO      (enable with -DBUILD_TESTING=ON)
-- nanomsg_transport NO      (default, enable with -DBUILD_NANOMSG_TRANSPORT=ON)
-- ofi_transport      NO      EXPERIMENTAL (requires C++14) (default, enable with -DBUILD_OFI_TRANSPORT=ON)
-- dds_plugin         NO      (default, enable with -DBUILD_DDS_PLUGIN=ON)
-- pmix_plugin        NO      (default, enable with -DBUILD_PMI_PLUGIN=ON)
-- examples           NO      (enable with -DBUILD_EXAMPLES=ON)
-- docs               NO      (default, enable with -DBUILD_DOCS=ON)
-- sdk                YES     (disable with -DBUILD_SDK=OFF)
```

libFairMQ_SDK.so

find_package(FairMQ COMPONENTS sdk)

**Then link against
FairMQ::SDK**

#include <fairmq/SDK.h>

Improvement in the DDS implementation compared to September

	Device per collection	Collections	Devices	INIT	CONFIG	START	STOP	TERM	DOWN
Sept 2019	12	300	3600	--	38.1	5.9	5.4	24.0	--
Oct 2019	12	300	3600	--	6.0	0.9	0.8	3.7	--
Nov 2019	12	300	3600	26.6	4.6	0.9	0.8	3.3	1.7

Sep 19: One agent per Task

Oct 19: One agent per node

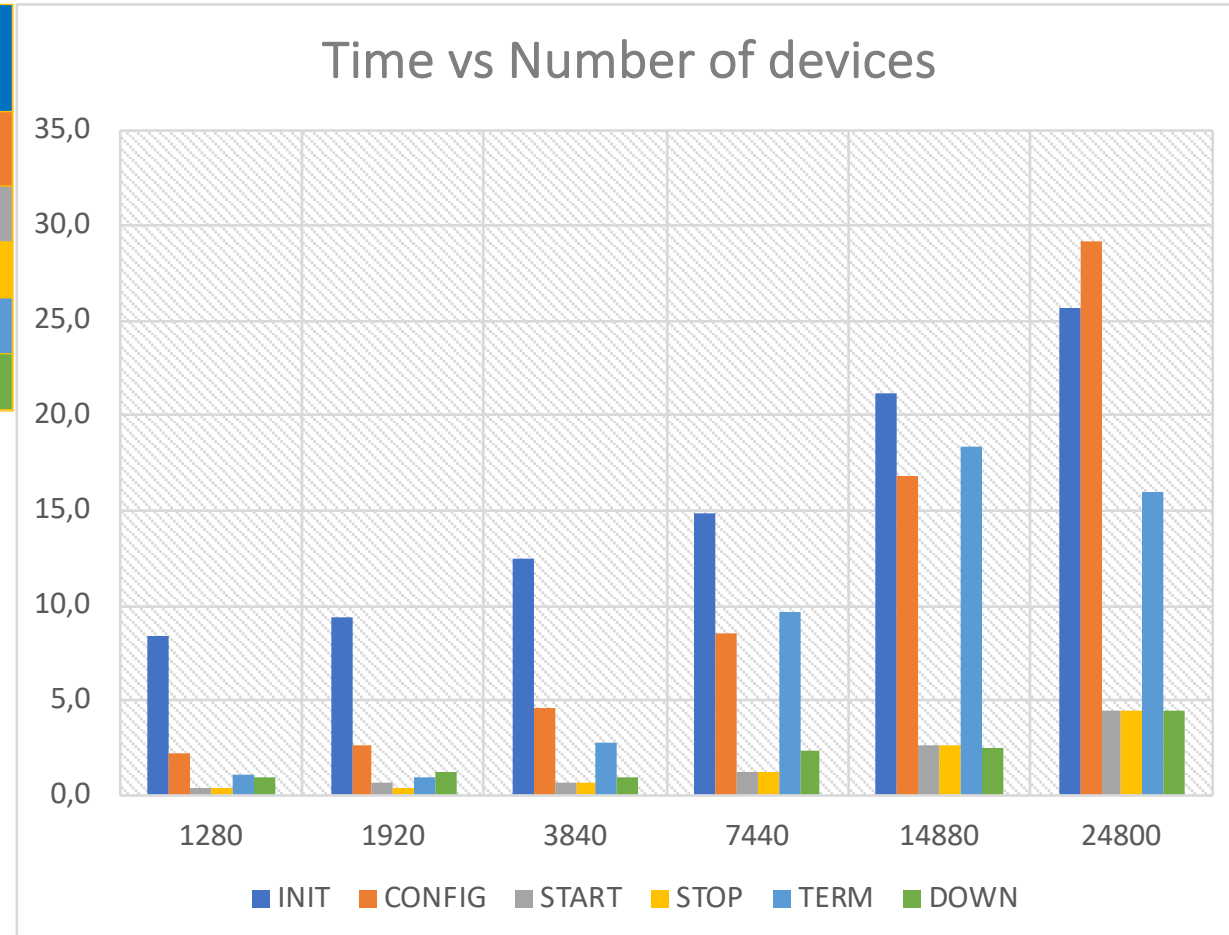
Nov19: Multiple agent per node (one agent per collection in this test)



DDS Based controller

INIT	Start DDS session → Submit agents using SLURM plugin → Activate topology
CONFIG	InitDevice → CompleteInit → Bind → Connect → InitTask
START	Run
STOP	Stop
TERM	ResetTask → ResetDevice → End
DOWN	Shutdown DDS session

- 60 nodes, 64 physical cores, 256 GB
- Slurm as RMS
- DDS-Slurm plugin
- gRPC `1.23.0`
- DDS `2.5-87`
- Boost `1.70.0`
- FairMQ `1.4.9` + `fmq_sdk_dds.patch`



What is new!



PMIx 10¹⁸



PMix started 2017

Start Someplace!



- **Resolve launch scaling**
 - Pre-load information known to RM/scheduler
 - Pre-assign communication endpoints
 - Eliminate data exchange during init
 - Orchestrate launch procedure

PMIx (Process Management Interface for Exascale)

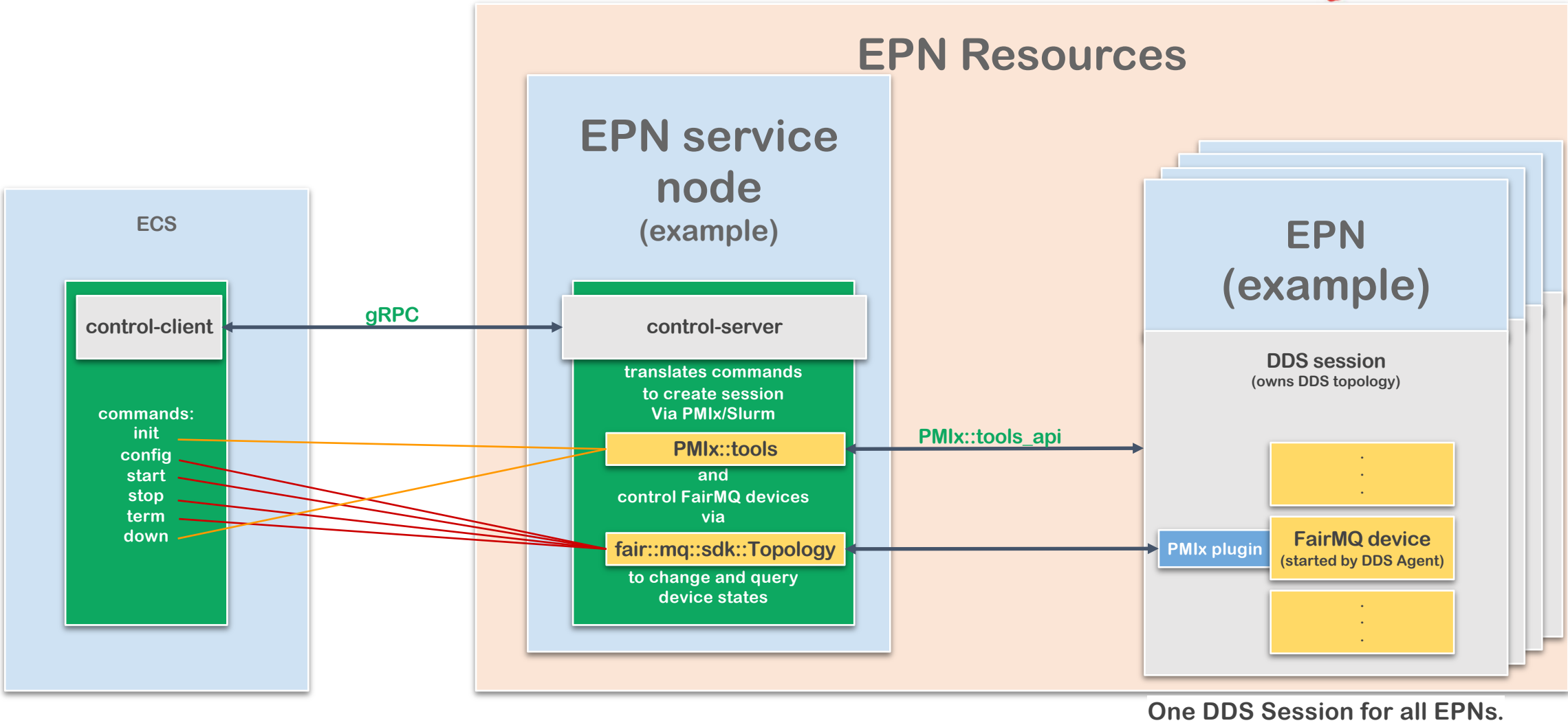
- Originally developed and distributed as part of MPICH, has historically been used as a means of exchanging wireup information needed for interprocess communication and deployment of processes
 - Distributed key/value store for data exchange
 - Asynchronous events for coordination
 - Enable interactions with the resource manager
- PMIx also covers: Resource allocation, process/job mgmt (creation/deletion/monitoring), system information, error notifications
- PMIx provides server, tool, and client APIs

<https://github.com/pmix/pmix>

<https://github.com/pmix/pmix-standard>

Controller Example (PMIx based)

PROTOTYPE



One DDS Session for all EPNs.

PMIx based controller, to do:

- Implement a PMIx based command transport (PMIx_Notify_event()...) in fair::mq::sdk::Topology
 - Implement *dds-session*, *dds-submit*, *dds-topology* counterparts based on PMIx_Tool_*(), PMIx_Allocation_request(), PMIx_Job_control(), PMIx_Spawn(), PMIx_Query()
- A production-ready PMIx support would need at least another 2-3 man-months

Comparison

Preliminary

Run with 3840 Devices

Controller	configure (s)	start (s)
PMIx based	1.5	0.03
DDS Based	4.6	0.7

What need to be defined to implement the EPN controller for ALICE:

- A common dictionary (objects, states and methods).
- A coherent error handling across the different boundaries has to be agreed and implemented.
- Interfaces to CCDB, calibrations, storage ...etc.

