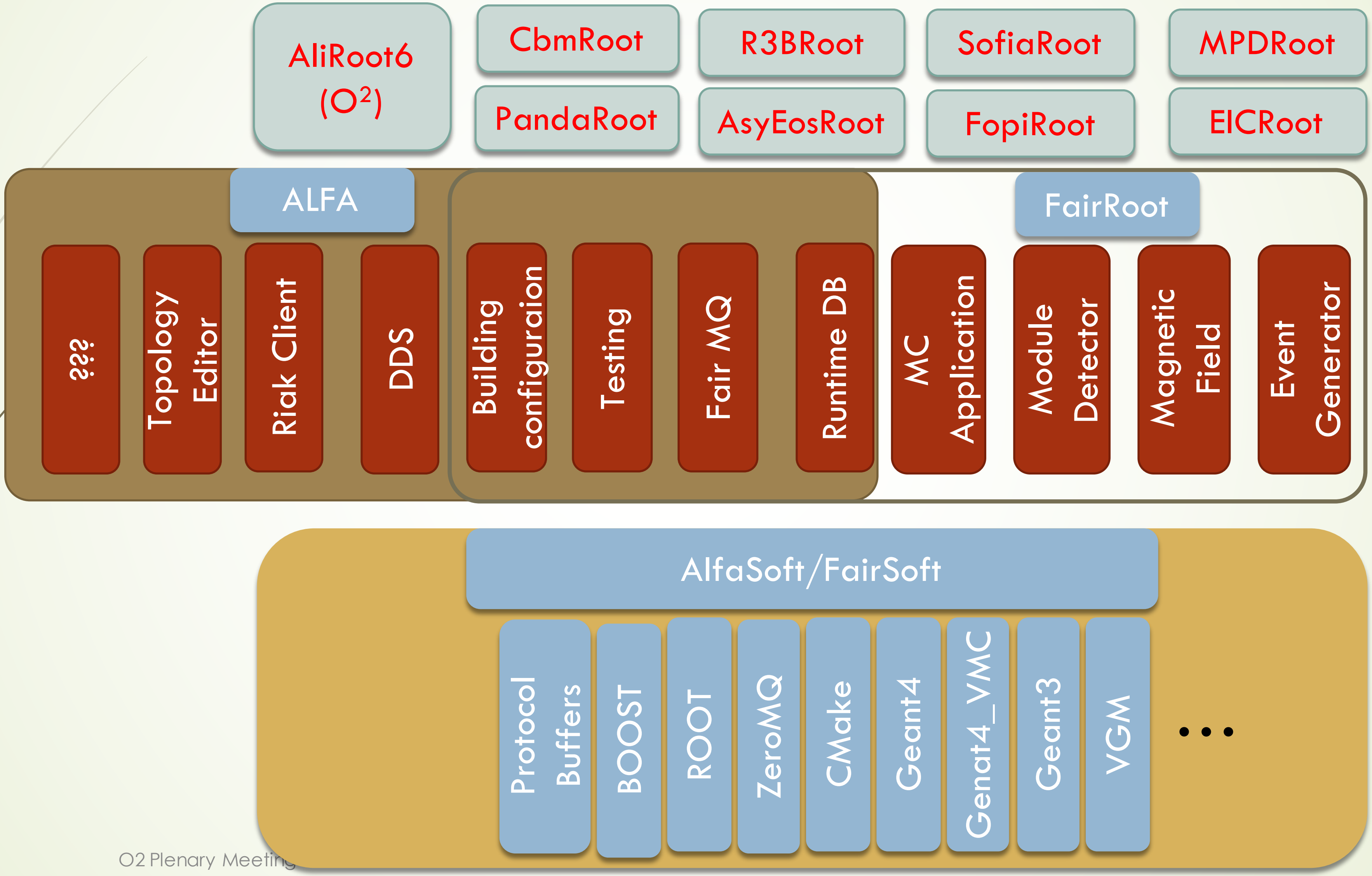
Status of ALFA

Mohammad Al-Turany

This report is a summary of the latest changes and plans of

- AlfaSoft/FairSoft
- FairMQ/FairRoot
- AliceO2



AliRoot6 (O^2)

CbmRoot

PandaRoot

R3BRoot

AsyEosRoot

SofiaRoot

FopiRoot

MPDRoot

EICRoot

ALFA

FairRoot

???

Topology Editor

Riak Client

DDS

Building configuraion

Testing

Fair MQ

Runtime DB

MC Application

Module Detector

Magnetic Field

Event Generator

AlfaSoft/FairSoft

Protocol Buffers

BOOST

ROOT

ZeroMQ

CMake

Geant4

Genat4_VMC

Geant3

VGM

…

AlfaSoft/FairSoft

Release “nov15” on Monday this week

https://github.com/FairRootGroup/FairSoft/releases/tag/nov15

Tested on:

Mac OSX: 10.9, 10.10, 10.11

OpenSuse: 13.1, 13.2, Leap 42.1

Fedora: 20, 21, 22, 23

Debian: 7, 8

Ubuntu: 14.04, 15.04, 15.10

Updated packages

New packages:
Sodium: 1.0.3

- CMake: 3.1.0 -> 3.3.2
- Boost: 1.58.0 -> 1.59.0
- Pythia8: 183 -> 212
- GEANT4: 10.1.0 -> 10.1.p02

- ROOT5: 5.34.32 -> 5.34.34
- ROOT6: 6.04.00 -> 6.04.02
- VGM: 4.2 -> 4.3
- GEANT4_VMC: 3-1-p1 -> 3.2
- Millipede: 4-01-01 -> 4-03-01
- ZeroMQ: 4.1.0 -> 4.1.3
- NanoMsg: 0.5-beta -> 0.6-beta

AlfaSoft/FairSoft

- Most packages are taken "as is"
- Changes are applied as patches when needed (Mainly to ROOT and Geant3)

Installation

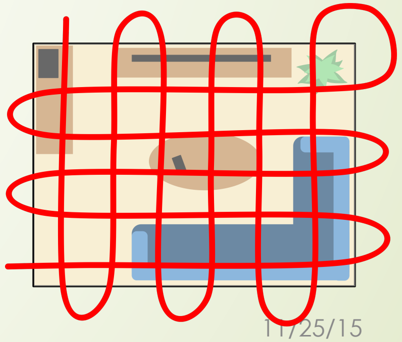
- Interactive (Menu)
- Installation with configuration file
  - Three example files (automatic.conf, grid.conf, and recoonly.conf) can be found in the main directory
- Make_clean.sh script

automatic.conf

```
compiler=
debug=yes
optimize=no
geant4_download_install_data_automatic=yes
geant4_install_data_from_dir=no
build_root6=no
build_python=no
install_sim=yes
SIMPATH_INSTALL=$PWD/installation
```

grid.conf

```
compiler=gcc
debug=no
optimize=no
geant4_download_install_data_automatic=no
geant4_install_data_from_dir=yes
build_root6=no
build_python=no
install_sim=yes
SIMPATH_INSTALL=$PWD/installation
build_for_grid=yes
```



Minimum configuration

- It works now (./configure recoonly.conf)
  - You need to set the compiler and installation directory
- It does not require Fortran
- It will not include simulation engines or event generators
- Instruction are on the main page of the repository

  https://github.com/AliceO2Group/AliceO2

./configure.sh recoonly.conf

- Will install:
  - BOOST
  - ROOT
  - ZeroMQ
  - nanoMSG
  - Protobuf
  - Xrootd
  - gtest
  - gsl

Need about 10 min to compile and install on alinsure machine

Need about 15 min to compile and install on a laptop

No Fortran is required

1 GByte

Cleaning packages

- This shell script will remove temporary files for the package specified by the first parameter. If the second parameter is present the script will also remove the installed files.
- To remove the temporary files of all packages the command is

```
./make_clean_install.sh all
```

- If the installed and temporary files of all packages should be removed the command is

```
./make_clean_install.sh all true
```

- If the installed and temprory files of a certain package need to be cleaned

```
./make_clean_install.sh   “package_name” true
```

This will remove installation files and make clean the package and all dependent packages

What is next?

- This is the last release of AlfaSoft/FairSoft with the installation scripts
- We are moving to the new AlfaBuild (Alibuild) system (see talk by Giulio Tomorrow)

Status of FairMQ/FairRoot

- More documentation and examples
- Huge effort on improving the modularity of the software
- Work is ongoing on integrating a key-Value data base to the parameter management

Re-organizing the examples and adding more documentation

- **MQ**
  - Set of simple FairMQ examples.
- **Simulation**
  - Set of simulation examples
- **Advanced**
  - Reconstruction examples, use of parameters, detector simulation and digitization event wise and timebased simulation, etc.

FairMQ Examples

Example 1: Sampler -> Sink

A simple topology of two devices - **Sampler** and **Sink**. **Sampler** sends data to **Sink** with the **PUSH-PULL** pattern.

Example 2: Sampler -> Processor -> Sink

A simple topology of three devices - **Sampler**, **Processor** and **Sink**. **Sampler** sends data to one or more **Processor**s, who modify the data and send it to one **Sink**. Transport with the **PUSH-PULL** pattern.

Example 3: DDS

This example demonstrates usage of the Dynamic Deployment System

Example 4: Copy & Push

A topology consisting of one **Sampler** and two **Sink**s. The **Sampler** uses the Copy method to send the same data to both sinks with the **PUSH_PULL** pattern. In contrary to the **PUB-PATTERN** pattern, this insures that all receivers are connected and no data is lost, but requires additional sockets.

Example 5: Request & Reply

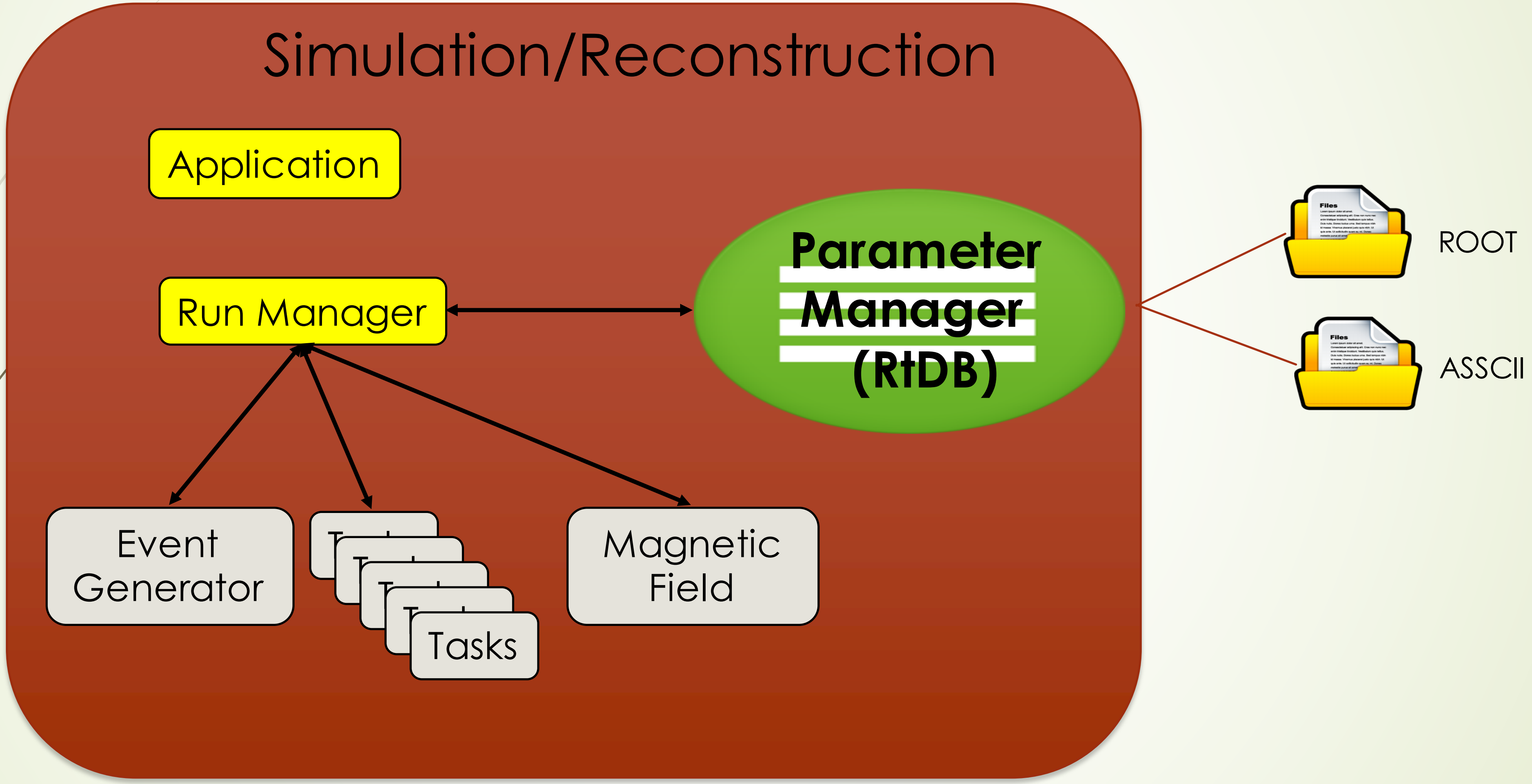
This topology contains two devices that communicate with each other via the **REQUEST-REPLY** pattern. Bidirectional communication via a single socket.

FairMQ Examples

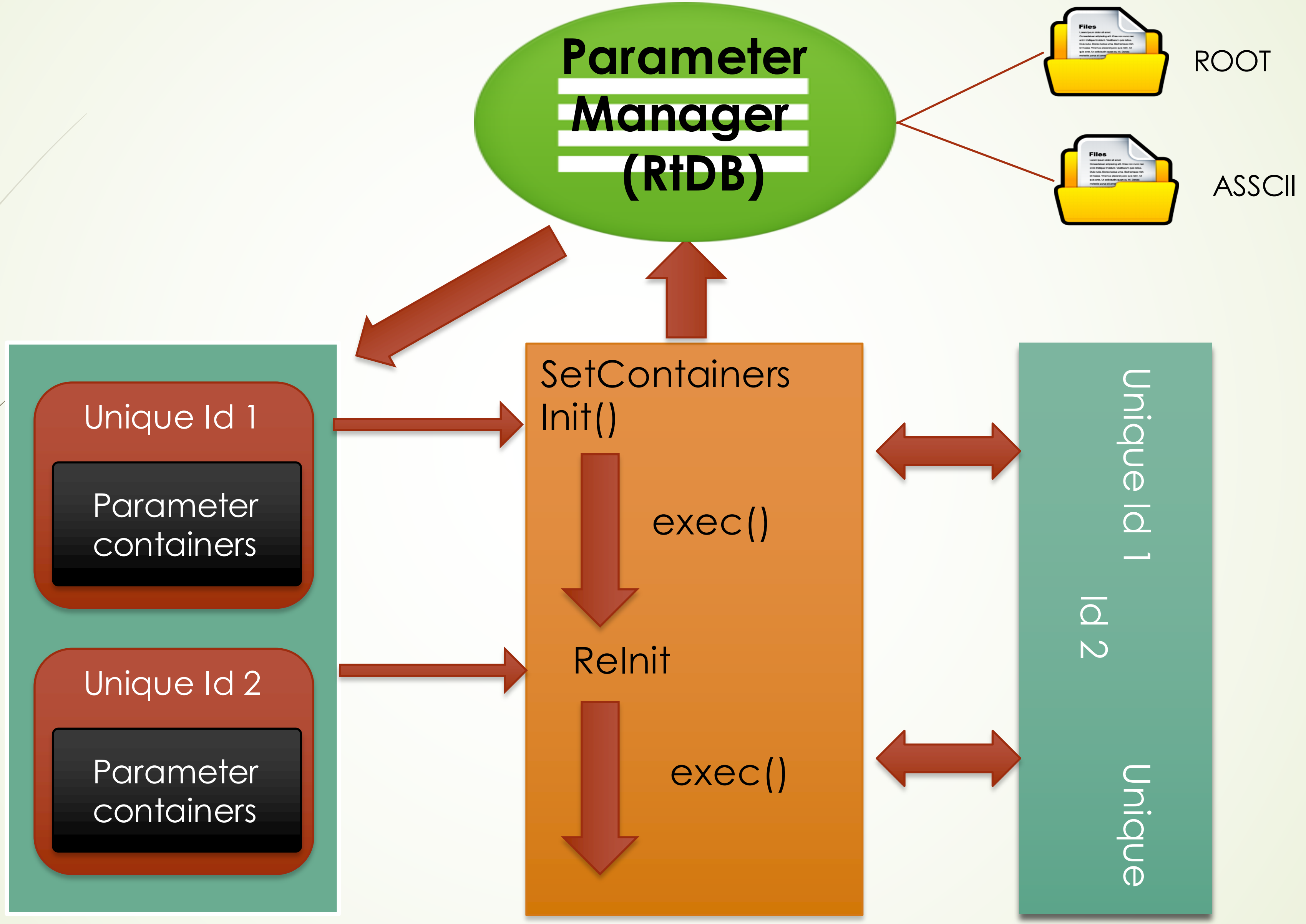
- **Example 6: Multiple Channels**
  - This example demonstrates how to work with multiple channels and multiplex between them.
- **Example 7: Communicating with ParameterMQServer**
  - This example shows how to communicate with the ParameterMQServer, that retrieves parameters from FairRuntimeDb.
- **Example 8: Sending Multipart messages**
  - This example shows how to send a multipart message from one device to the other. (two parts message parts - header and body).
- **GenericDevices: Generic devices example**
  - This example shows how to use the policy based design of the generic MQ-devices.
- **LmdSampler: (GSI List Mode Data format) Lmd Sampler**
  - In this tutorial a sampler read an lmd file (the one in /examples/advanced/tutorial8) and send the binary data to an unpacker device.

Parameter management in ALFA

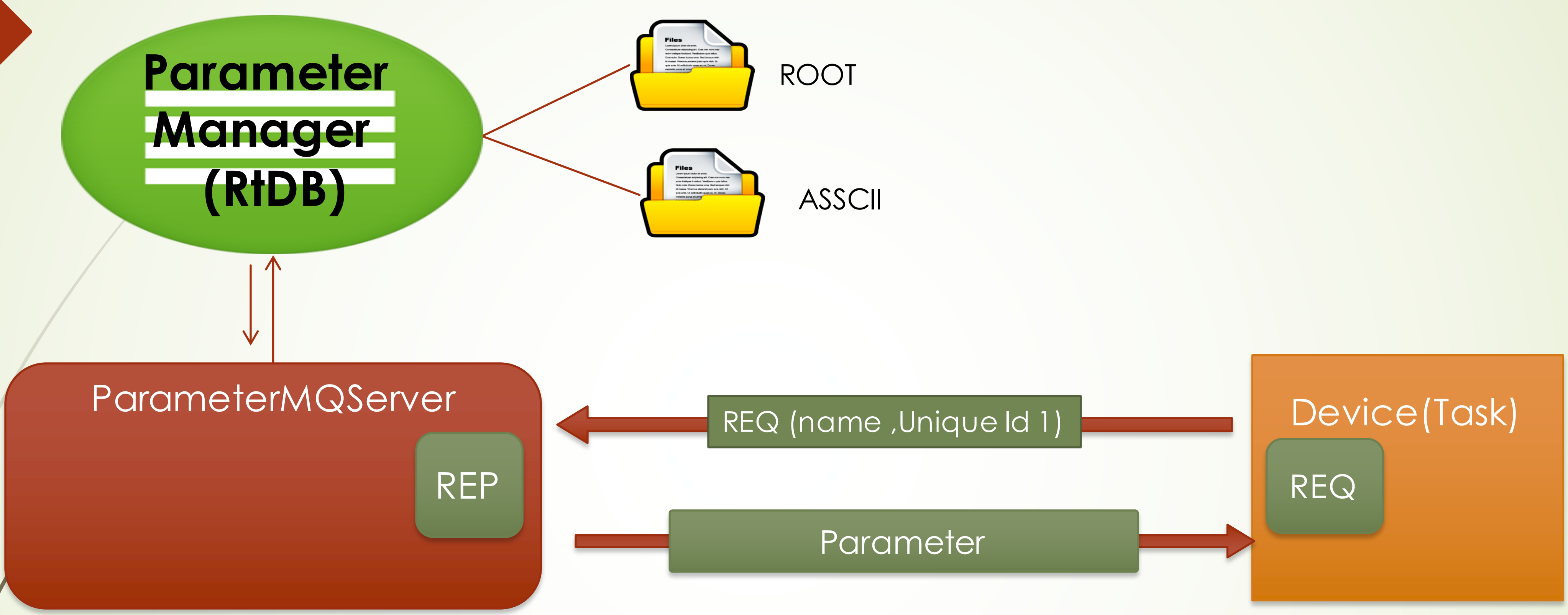
Parameter manager in FairRoot/AliceO2 (ROOT based single process)



Initialization scheme in FairRoot/AliceO2

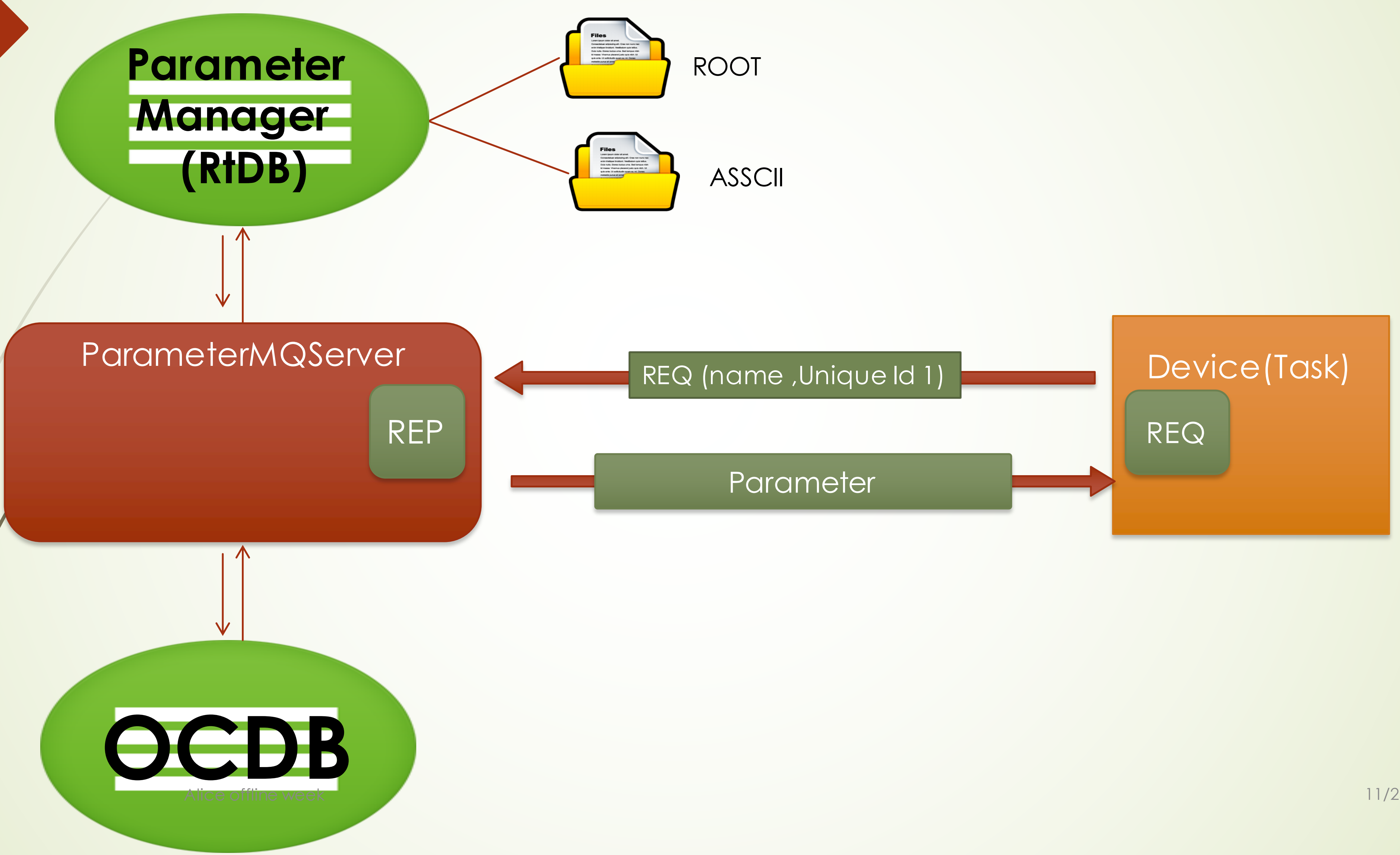


Parameter manager for ALFA



See example 7 in examples/MQ
https://github.com/FairRootGroup/FairRoot/tree/master/examples/MQ/7-parameters

Parameter manager for ALFA (ongoing work)



Parameter Manager (RtDB)

ROOT

ASSCII

ParameterMQServer

REP

REQ (name ,Unique Id 1)

Parameter

Device(Task)

REQ

OCDB

Parameter management

Distributed Model based on Riak

- high availability
- scalability
- fault tolerance
- configurable

Message-Queue based concept make it possible to use directly the native Java client of Riak

Two storage back-ends were tested:

- Bitcask
  - best latency
  - nodes out-of-memory
- LevelDb
  - similar performance
  - compressed storage

riak

Java

ØMQ

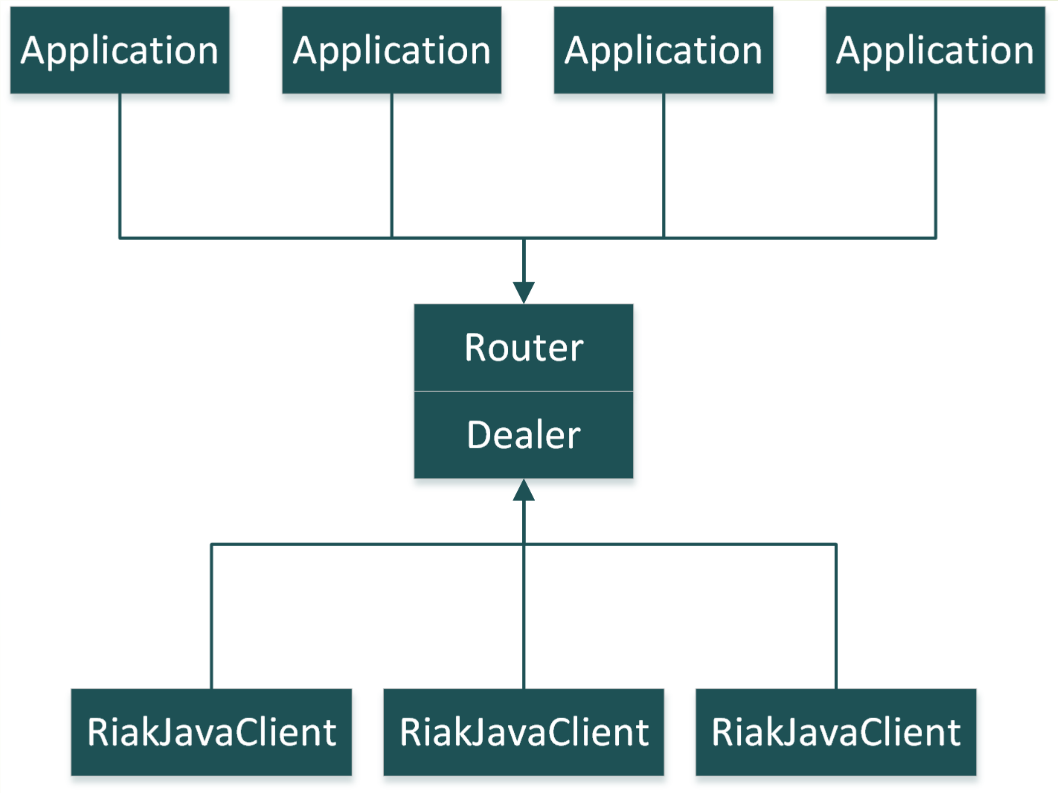
protobuf
Protocol Buffers

https://zenodo.org/record/31858#.VlU7WlSWGgx

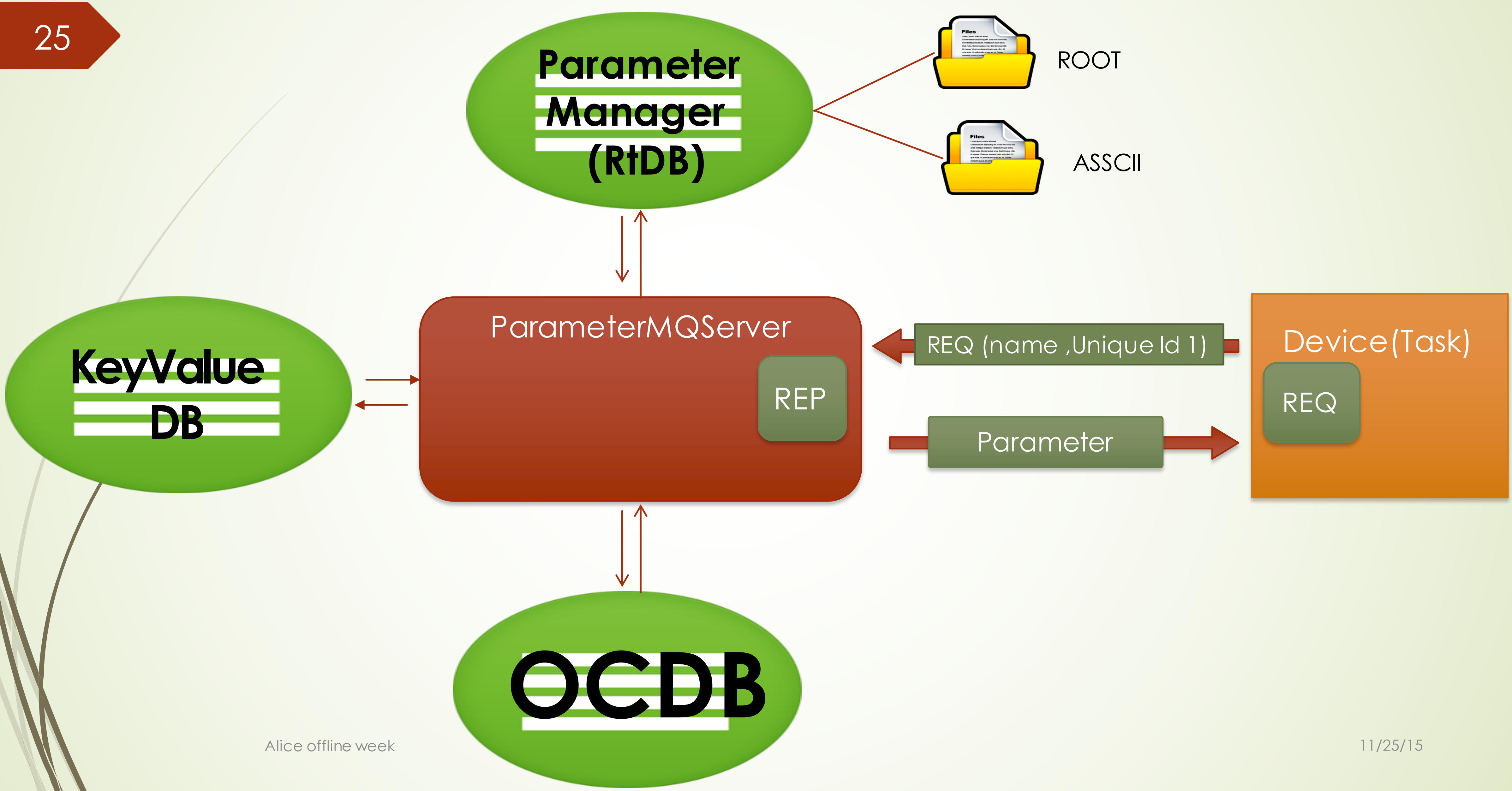
Tom Van Steenkiste

Load balancer pattern

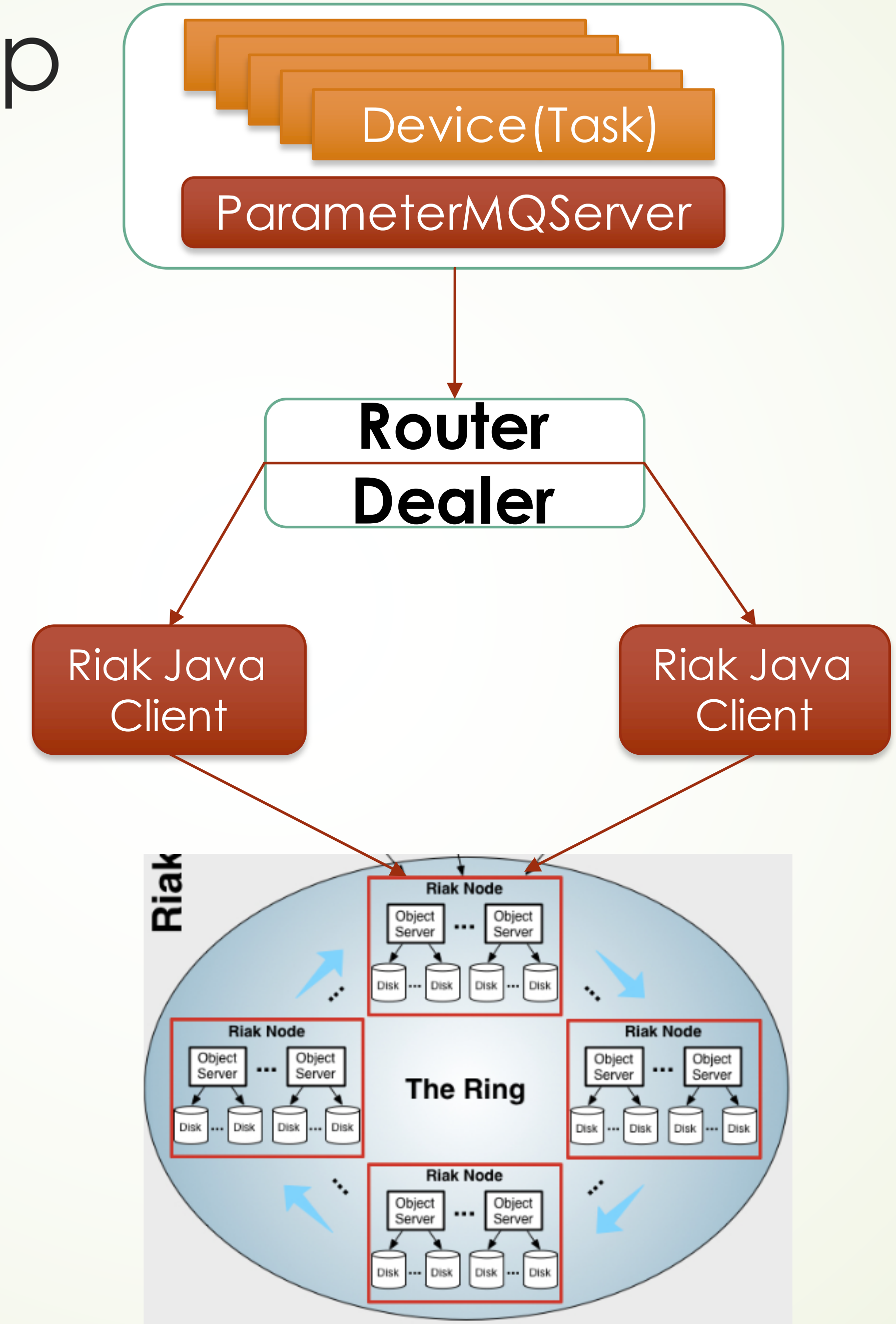
- Connecting applications only have to know how to reach the load balancer.
  - They don't have to be aware about the backend implementation of Riak and how many Java clients are online.
  - The Java clients themselves are also easily configured as they only need to connect to the Riak cluster and the load balancer.



Parameter manager for ALFA (ongoing work)



Dynamic setup



Device(Task)

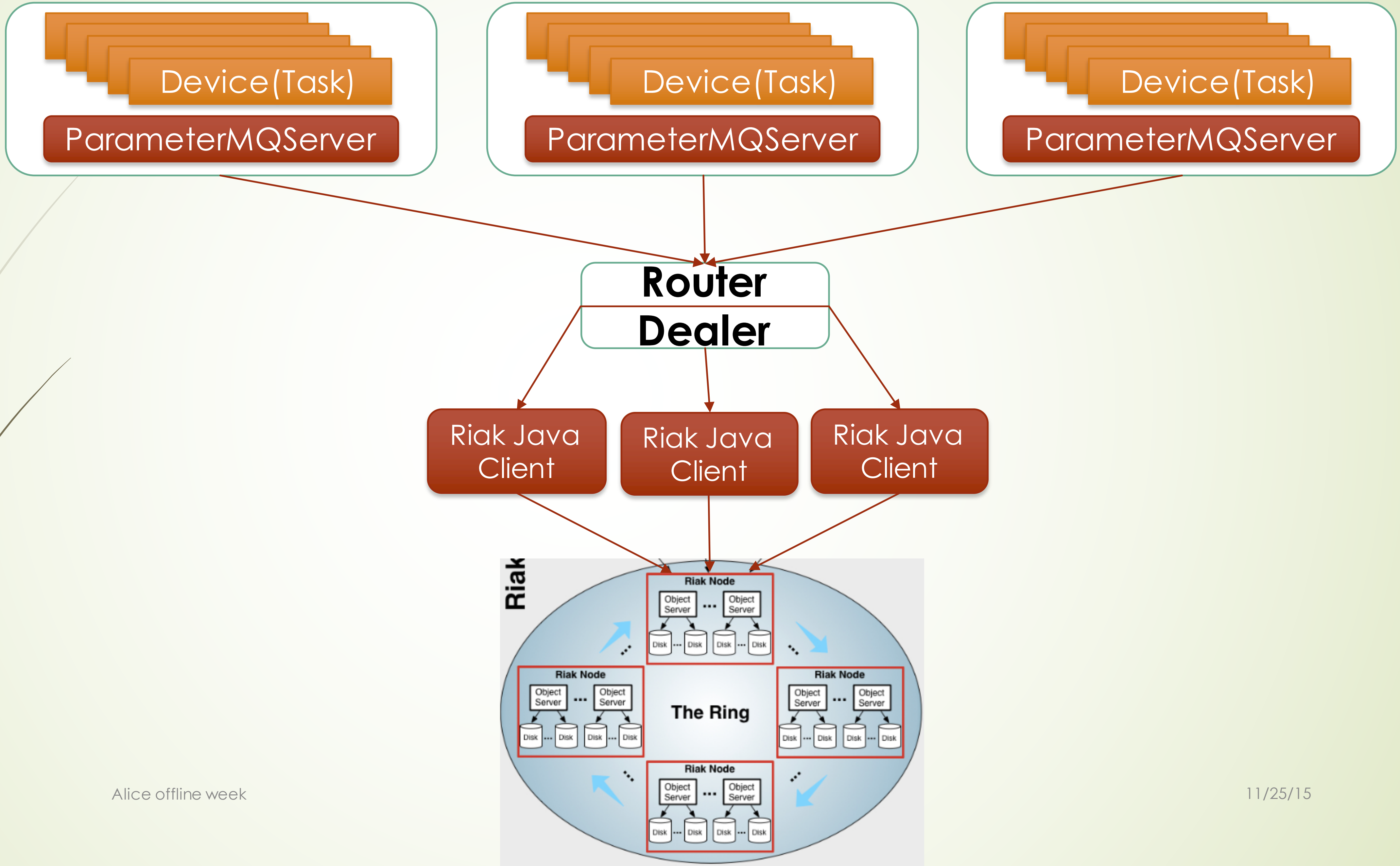
ParameterMQServer

Router

Dealer

Riak Java Client

Riak Java Client



Device(Task)

ParameterMQServer

Device(Task)

ParameterMQServer

Device(Task)

ParameterMQServer

Router

Dealer

Riak Java Client

Riak Java Client

Riak Java Client

AliceO2

- Introduce new tests and examples
- Reorganize FLP2EPN device configuration to use the new FairMQChannels. i.e:
  - Rename channels for FLP2EPN-distributed devices according to their roles.
  - Separate channel per role - data-in, hb-in (heartbeat),data-out, hb-out.

New Features:

- Check Headers
  - If the tool: include_what_you_use (llvm based tool) is installed on your system, the target checkHEADERS will be build
  - To use it: **make -k checkHEADERS**

Documentation

- Improved the documentation on GitHub
  - Documenting the full and minimum installations for the AlfaSoft/FairSoft

    https://github.com/AliceO2Group/AliceO2/tree/dev

- Improve configuration and update the Doxygen documentation

  http://aliceo2group.github.io/AliceO2