

Draft for a Deployment System for FESA Software

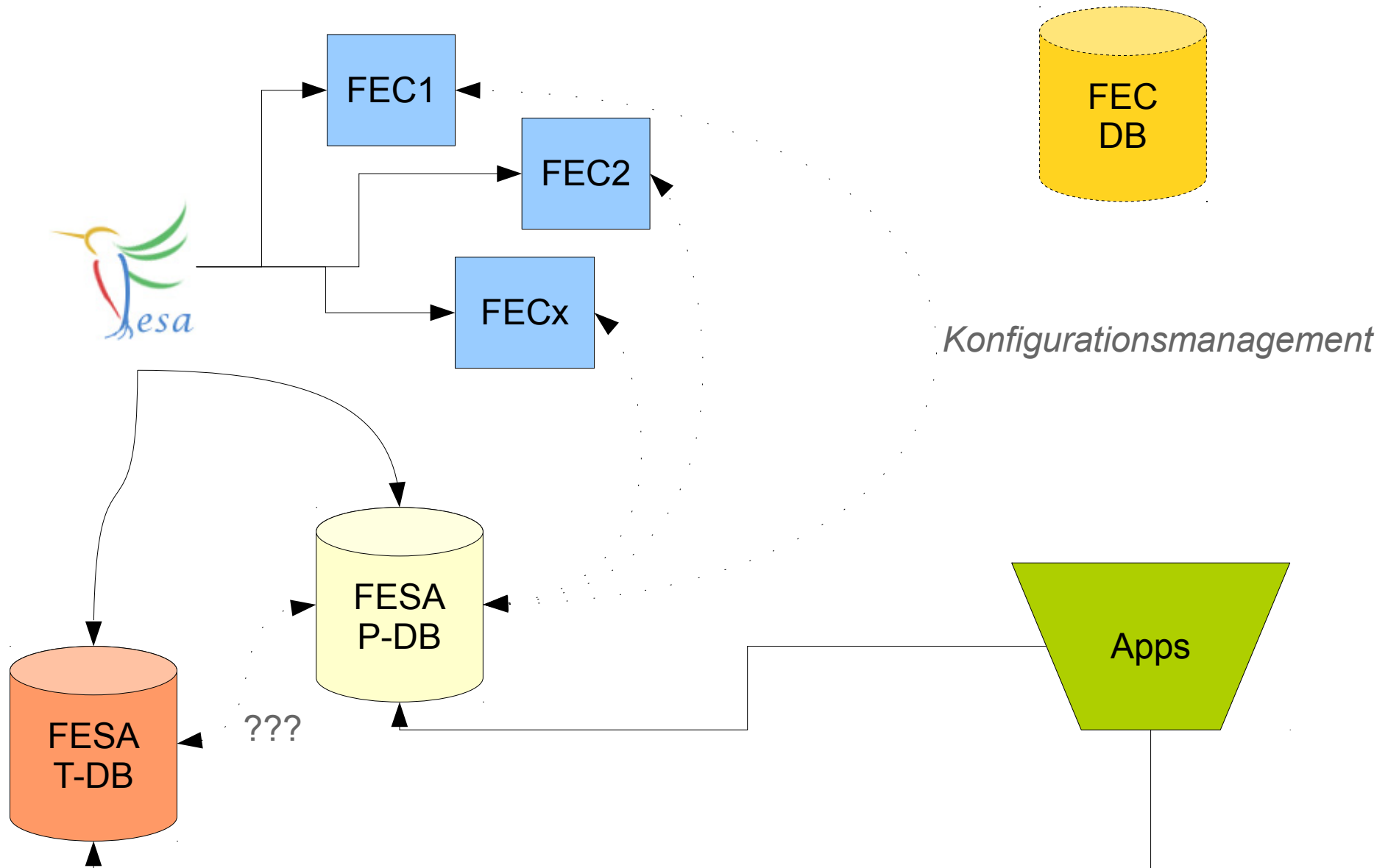
Release of FESA Software: Aspects

- „Physically“: binaries and configuration files*
- „Externally“: interface for accessibility by apps%
- „Internally“: improved overview for FEC maintainers

*: for FECs on asl cluster

=: designs and configuration in a database

Relations – existing and missing



Requirements

- 1) Collect FESA software in **binary repository**
- 2) **FEC setup** accompanied by a dedicated database (create, add, rename, delete, runtime environment configuration)
- 3) FESA software **configuration** for FECs
- 4) **Distribution** of FESA software for desired FECs (add, maintenance, update, rollback)
- 5) **Activation** of changes on FECs

1) Binary Repository

- FESA developers releases FESA software to binary repository once
 - Versioned storage
 - Independent of DEV / PRO environment
- Configuration files (cmw, fesa, messages) may be taken from FESA release - TBD
- **Dedicated database** is filled with information of released FESA deploy-unit in binary repository (Name, Version, timestamp, FESA version, required driver software, timing library version, ...)
- Sources can be tagged in software repository automatically

```
matthies@asl744:binary-repository>tr
.
|-- deploy-units
|  |-- BumperDU
|     |-- 1.0.0
|         |-- BumperDU
|         |-- some.cfg
|     |-- 1.0.1
|         |-- BumperDU
|         |-- some.cfg
|  |-- CryCupDU
|     |-- 2.0.2
|         |-- CryCupDU
|         |-- some.cfg
|     |-- 2.2.2
|         |-- CryCupDU
|         |-- some.cfg
|  |-- PowerSupplyDU
|     |-- 1.0.0
|         |-- PowerSupplyDU
|         |-- some.cfg
|     |-- 1.2.0
|         |-- PowerSupplyDU
|         |-- some.cfg
|     |-- 1.2.1
|         |-- PowerSupplyDU
|         |-- some.cfg
|-- drivers
|  |-- fesi
|     |-- 1.2.3
|  |-- slits
|     |-- 0.8.15
16 directories, 14 files
matthies@asl744:binary-repository>
```

2) FEC Setup and Maintenance with Database Support

- Operations:
 - Create a new FEC
 - Rename an existing FEC
 - Delete an existing FEC
 - Adapt runtime environment configuration (DEV or PRO)
 - Adapt FEC configuration
 - Select FESA software and drivers

GUI Draft: FEC Configuration

Configure FEC

Name

Callout: Ideally performs name availability checks against FEC DB

Callout: Select existing from FEC DB

Configuration

Callout: Select previous configurations from a list of stored configurations per FEC

User comfort

FESA Software

Drivers

Delete FEC

Callout: Select from DB which holds information about content of binary repository

Callout: FESA DB will be updated accordingly

GUI Draft: Deploy-Unit Selection

The screenshot shows a window titled "Select Deploy-Units" with a table of deploy units. The table has five columns: a checkbox, "DeployUnit", "DU Version", "FESA Version", and "Timestamp".

	DeployUnit	DU Version	FESA Version	Timestamp
<input type="checkbox"/>	BumperDU	1.2.3	4.0.0	20170314_1
<input checked="" type="checkbox"/>	BumperDU	1.2.5	4.2.0	20170407_1
<input type="checkbox"/>	CryCupDU	2.2.0	4.2.0	20161001_0
<input type="checkbox"/>	CryCupDU	2.2.1	4.2.0	20170731_1
<input type="checkbox"/>	PowerSupplyDU	1.5.4	4.2.0	20170518

Two callout boxes are present:

- A callout box pointing to the table data with the text: "Data retrieved from FEC DB / content of binary repository".
- A callout box pointing to the table area with the text: "Tool-Tip mit Liste enthaltener FESA Klassen anzeigen?".

At the bottom of the window are "Cancel" and "OK" buttons.

3) FESA Software Configuration for FECs: Instantiation

- Creation, manipulation, maintenance of FESA instantiation data
- Possibility to adjust FEC configuration in two ways:
 - manually in workspace, DB import into FESA DB possibility already available
 - directly in database via graphical front-end (web? / JAVA based?)
 - DB export into workspace required!

GUI Draft: FEC Instantiation

The screenshot shows a window titled "Instantiation" with a table of FEC entries. The table has the following columns: ID, Nomenclature, State, Accelerator, and Time Domain. The first two rows are selected (checked) and have a callout indicating they were retrieved from the FESA DB. The third row has a callout indicating it has a context menu with editing options. The bottom of the window has "Cancel" and "OK" buttons, with a callout indicating that the FESA DB will be updated accordingly.

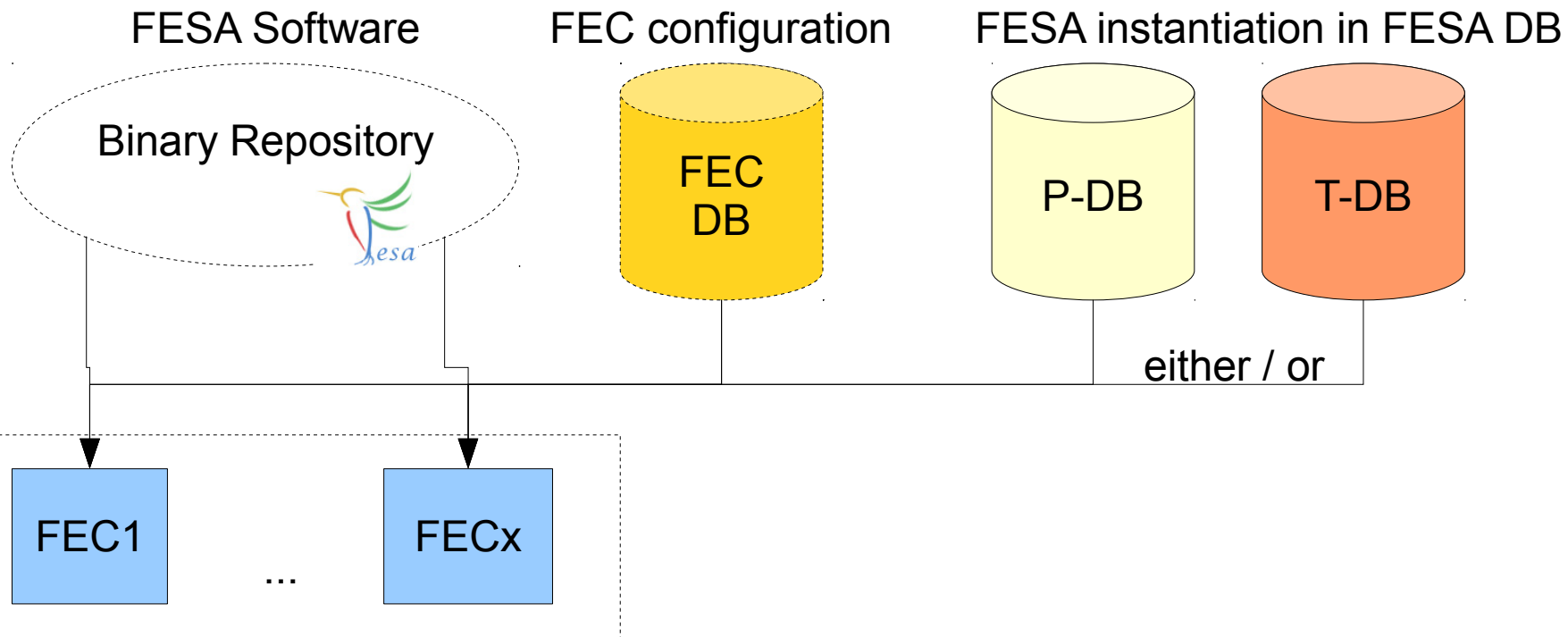
scuxl0075	Nomenclature	State	Accelerator	Time ...
<input checked="" type="checkbox"/>	YR01LB	operational	CRYRING	CRY ...
<input checked="" type="checkbox"/>	YR01LA	operational	CRYRING	CRY ...
scuxl0013	Nomenclature	State	Accelerator	Time Dom ...
<input type="checkbox"/>	GHTYKV3	development	CRYRING	HTB ...
<input type="checkbox"/>	YRT1KV2	development	CRYRING	YRT ...

Callouts:

- Data retrieved from FESA DB
- Context Menu with various editing possibilities: change, copy, copy to all, ...
- FESA DB will be updated accordingly

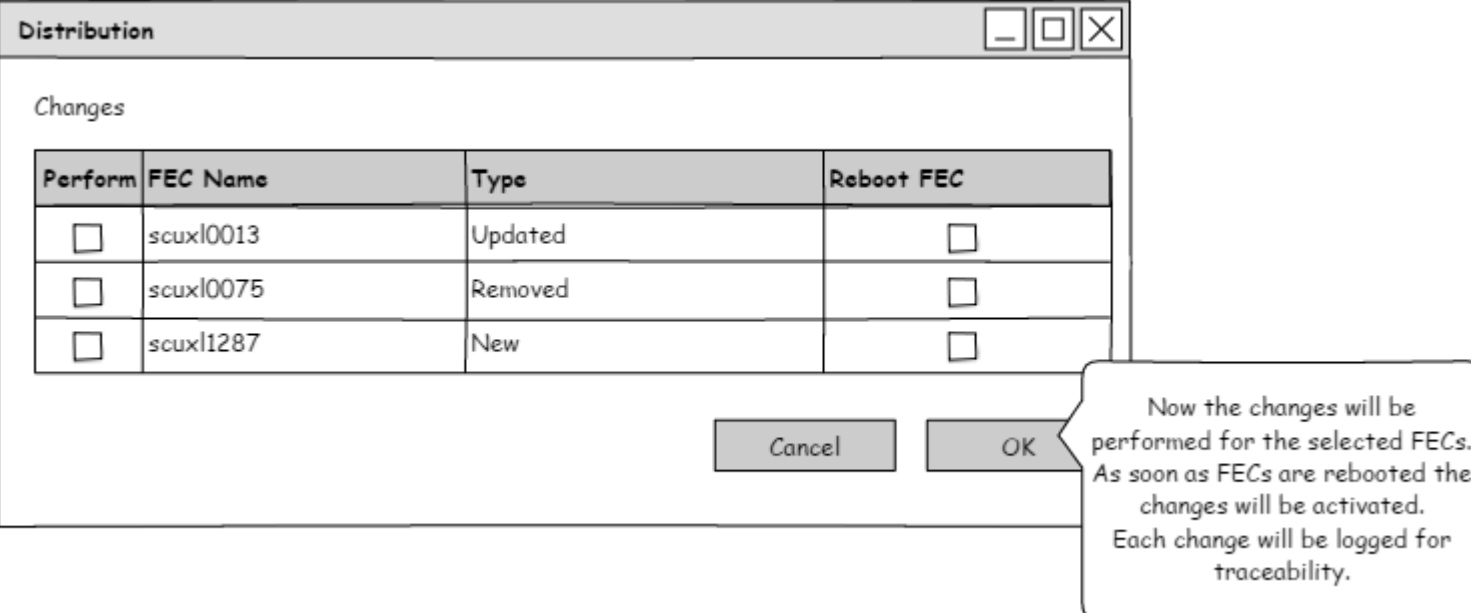
4) Distribution of FESA Software

- Add, update, rollback
- Distribution of FESA software for desired FECs from binary repository
- Extraction of required configuration from FESA database for the FEC



5) Activation of Changes

- Tool required to activate the changes on the affected FECs (e.g. by reboot)



The screenshot shows a dialog box titled "Distribution" with a "Changes" section. It contains a table with the following data:

Perform	FEC Name	Type	Reboot FEC
<input type="checkbox"/>	scuxl0013	Updated	<input type="checkbox"/>
<input type="checkbox"/>	scuxl0075	Removed	<input type="checkbox"/>
<input type="checkbox"/>	scuxl1287	New	<input type="checkbox"/>

Below the table are "Cancel" and "OK" buttons. A callout box points to the "OK" button with the following text:

Now the changes will be performed for the selected FECs. As soon as FECs are rebooted the changes will be activated. Each change will be logged for traceability.

Update to new FESA FWK Version

- Two possibilities:
 - Manual, recommended for larger changes
 - Automatically, recommended only for smaller changes in the FESA FWK that can be performed using migration tools
- Return to step 1)

Testing of FESA Software

- Several possibilities:
 - In development environment, e.g. using mock-ups
 - Designs and Instantiation in development database
 - In productive environment, using state „inUse“ to test on dedicated FECs, working preferably without binary repository
 - Designs and instantiation in productive database

To be discussed

- Security: How to avoid sneaking around the deployment system? (check timestamps?)
- Irritation: How to avoid ambiguities in different environments? → possible device name collisions in LSA?
- Precision: How strict about different versions in different environments? Additional checks between FESA databases directly possible? Additional checks during export to the FESA databases?

To be further discussed

- Switching between development and productive: allow moving of information in databases beside FEC configuration?
- Should productive FECs boot from the productive cluster?
- Traceability: who did what when? Log-in to deployment system with a user name, trace and log actions

Constraints

- Synchronization of data stored and software and configuration spread is vital
- System must be extendable for different driver software versions such as FESL
- System must be flexible enough to react on timing library changes (→ more clear versioning required for SAFTlib and corresponding timing firmware versions!)