

Moving away from ETICS ... to Jenkins

or how I learned to stop worrying and replace Etics with a 300-line script

Dvořák, F.; Sitera, J.; Šustr, Z.

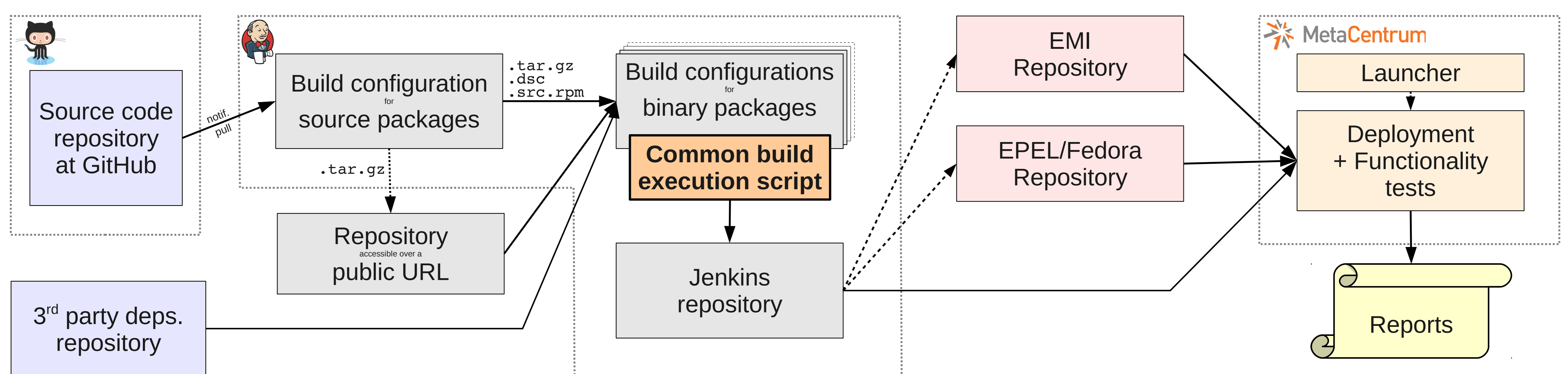
CESNET, a. i. e., Prague, Czech Republic

The middleware release process used within the EGEE series of projects and within EMI depended heavily on ETICS to provide centralized **build, integration, repository and archiving functions**. Although there was certain progress towards independence in the final era of EMI, in the end, each product team had to deal with the **de-commissioning of ETICS** separately.

The middleware development team at CESNET, the providers of four distinct grid products within the EMI stack, have adopted Jen-

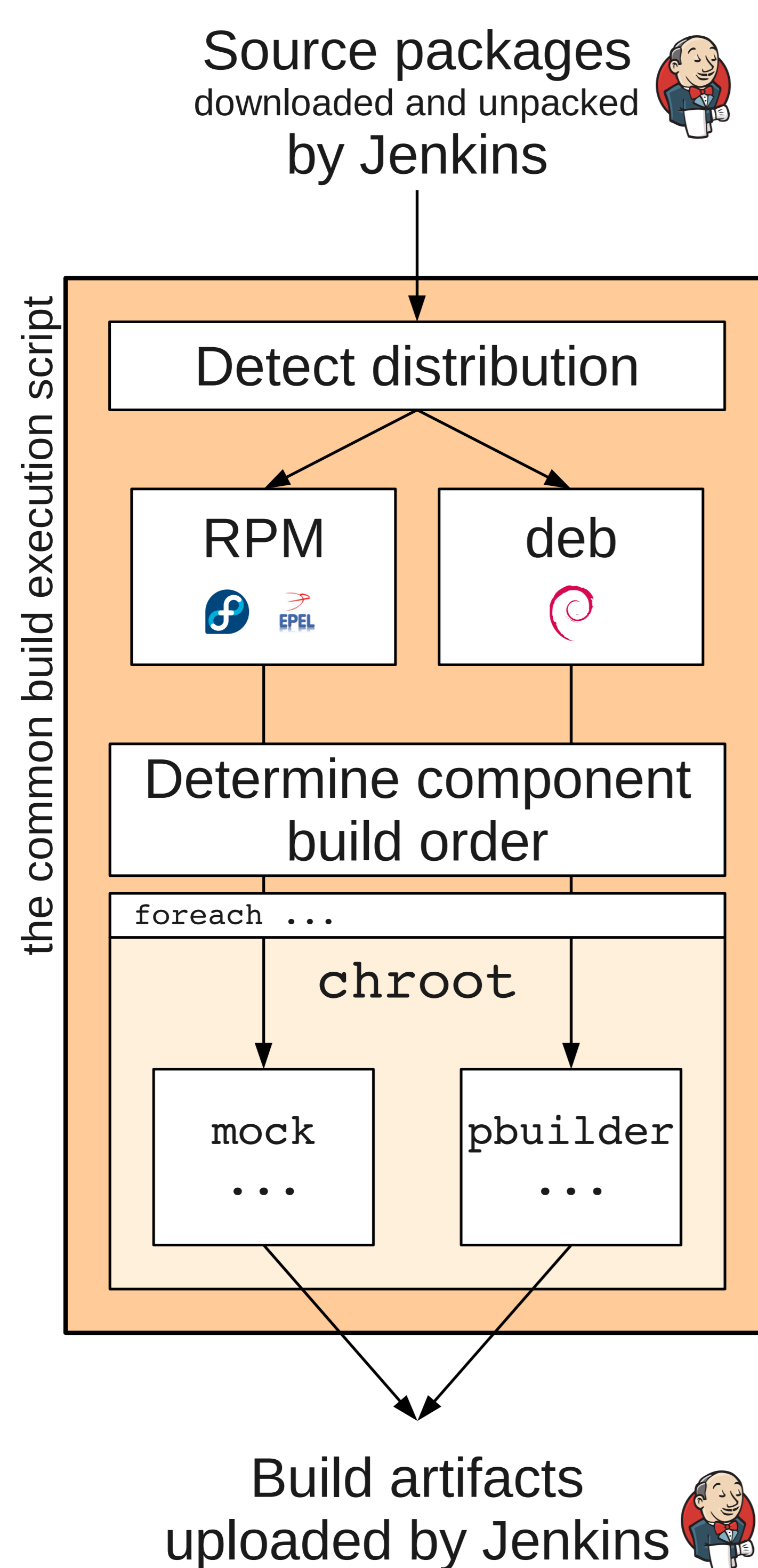
kins [1], [2] to provide integration functions missed after the end of EMI. This poster gives details of CESNET's solution, used to continuously build and integrate its four products, consisting, in total, of more than thirty packages. The continuous integration process takes care not only of regular or release builds, but also of continuous packaging and testing.

Within the few months in use, Jenkins was used successfully to produce at least one release for each relevant product.



Build Execution Script

The diagram below gives a high-level idea of the purpose and internal workings of the build execution script.



Comparison with the Etics-based solution

The solution based on Jenkins—illustrated in the diagram above—is less flexible but simpler, easier to maintain and adequately functional.

- Jenkins is used as a sophisticated scheduler for build operations with data manipulation and archival functions.
- + The **Makefile** targets for `mock` and `pbuilder` tools—already developed within the EMI project—are **reused** in the new environment.
- + The **missing functionality**—correctly ordered execution of the appropriate build tool in a pristine environment—is achieved through a simple build execution script.
- There is no automated detection of dependences between packages. The **build order** is hard-coded in the execution script. However, this is easily maintained for the handful of products provided by a single team, and seldom changes.
- + Completes full automation of the **continuous integration** chain for middleware products: GitHub → Jenkins → testing/certification cloud for CESNET's grid middleware products [3], whereby a commit to the development branch in the VCS triggers a build and then a deployment/functionality test of the change.

References

- [1] *Jenkins CI*, <http://jenkins-ci.org>
- [2] *CESNET Jenkins instance*, <https://emian.zcu.cz:8443/jenkins/>
- [3] *Mass Testing of EMI Products in Czech NGI's Virtualized Environment*, <http://egee.cesnet.cz/cvsweb/LB/CF12-mass-test.pdf>