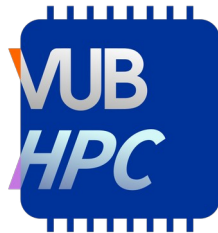


So you want to implement a bug fix or feature in EasyBuild framework...

Alex Domingo



VLAAMS
SUPERCOMPUTER
CENTRUM



Vlaanderen
is supercomputing

WHO AM I?

Hi! I'm Alex (github: [@lexming](#))

- ▶ Background
 - ▶ PhD in computational chemistry
 - ▶ User of Linux and FOSS in general since mid 2000's
- ▶ **Present time: HPC team of VUB** since 2019 ([hpc.vub.be](#))
 - ▶ Horizontal team: Linux sysadmin, software optimization, direct user support, hardware hammering
 - ▶ Maintainer of EasyBuild ([easybuild.io](#)): open source software build and installation framework for HPC
- ▶ Free time: eats chocolate and plays with Raspberry Pis

WHO AM I?

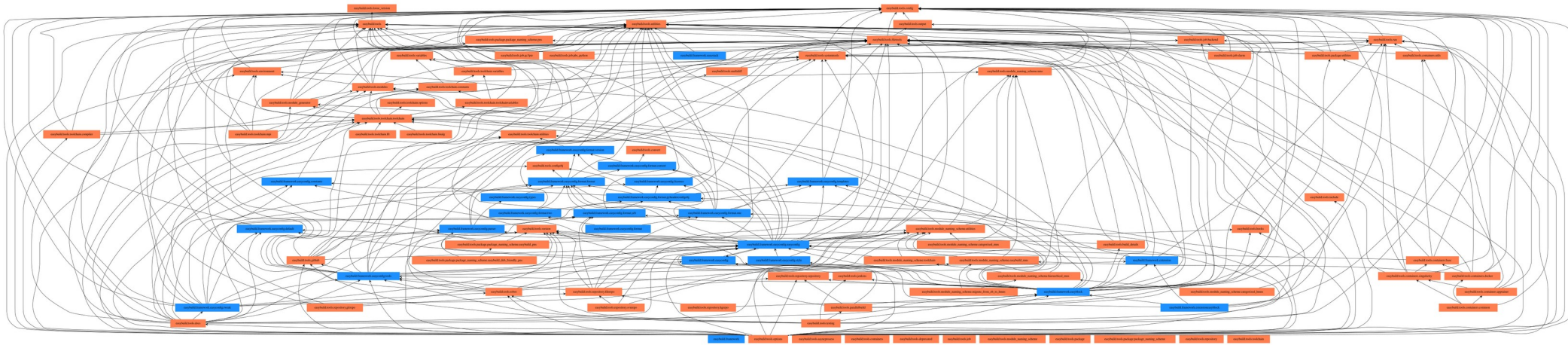
Hi! I'm Alex (github: [@lexming](#))

- ▶ I have absolutely no idea how EasyBuild framework works!

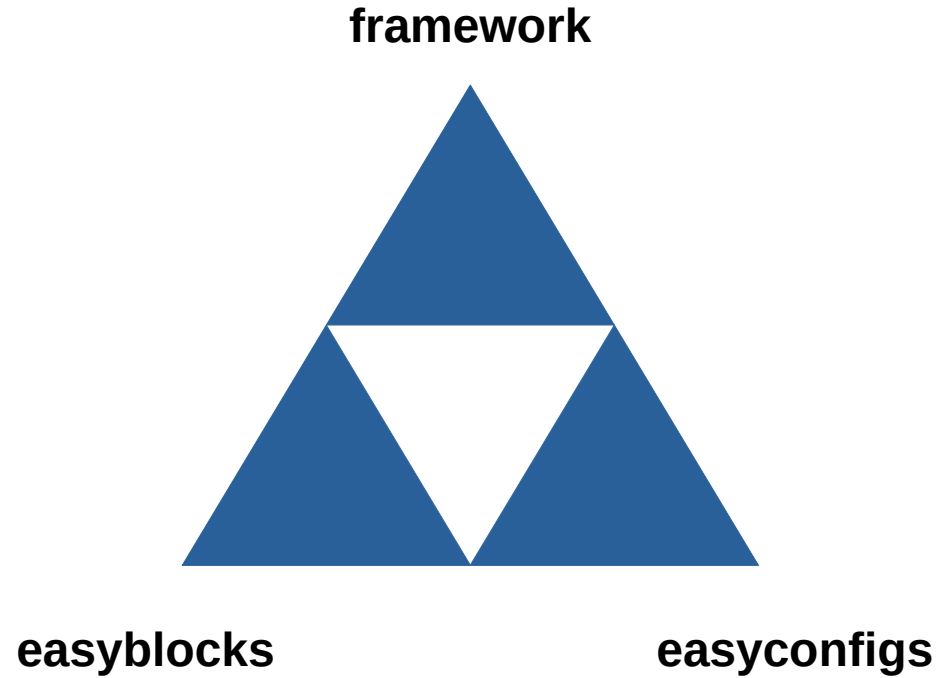
what am I doing here?
please send help...

EASYBUILD FRAMEWORK

Package diagram of **easybuild.framework** and **easybuild.tools**



! WARNING: It is known that firsts encounters with the maze of framework happen late at night! Be ready ☕ 😬



We come to EasyBuild for the easyconfigs...

```
$ eb PyTorch-2.0-foss-2023a-CUDA-12.1.0.eb
```

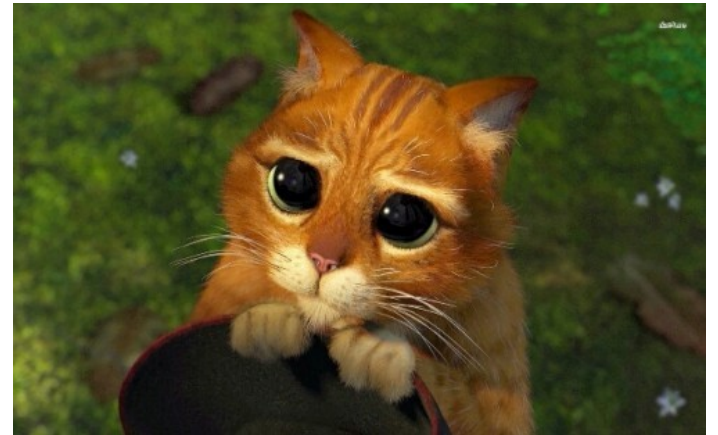
“EasyBuild supports 2995 different software packages”

<https://docs.easybuild.io/version-specific/supported-software>



... but users always manage to find software that is missing in the repo

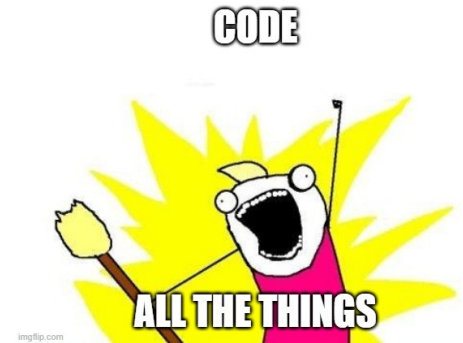
“Can you please install <insert link to github repo without releases or tags, a folder with a bunch of Perl scripts, last commit 5 years ago, instructions say something about conda>?”



Phase 1: easyconfigs

- ▶ Put on the developer shirt to make a new easyconfig
- ▶ Search in the repo for easyconfigs of other versions of software
- ▶ Search on the Internet for easyconfigs of target software
- ▶ Go through README of target software
- ▶ Figure out a procedure to install target software
- ▶ Look for easyconfigs that follow similar procedure
- ▶ Installation procedure cannot be done with available easyblocks! 🤪

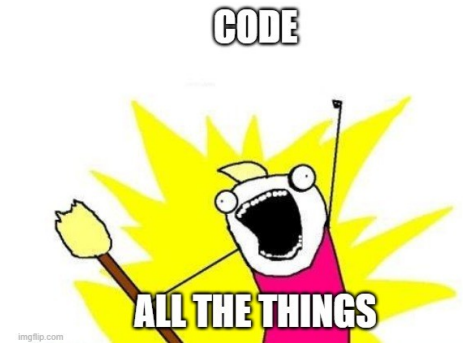
🤪 Some easyblocks such as **CmdCp** are very flexible and allow for heavy tinkering



Phase 2: easyblocks

 Check Easybuild docs:
docs.easybuild.io/implementing-easyblocks

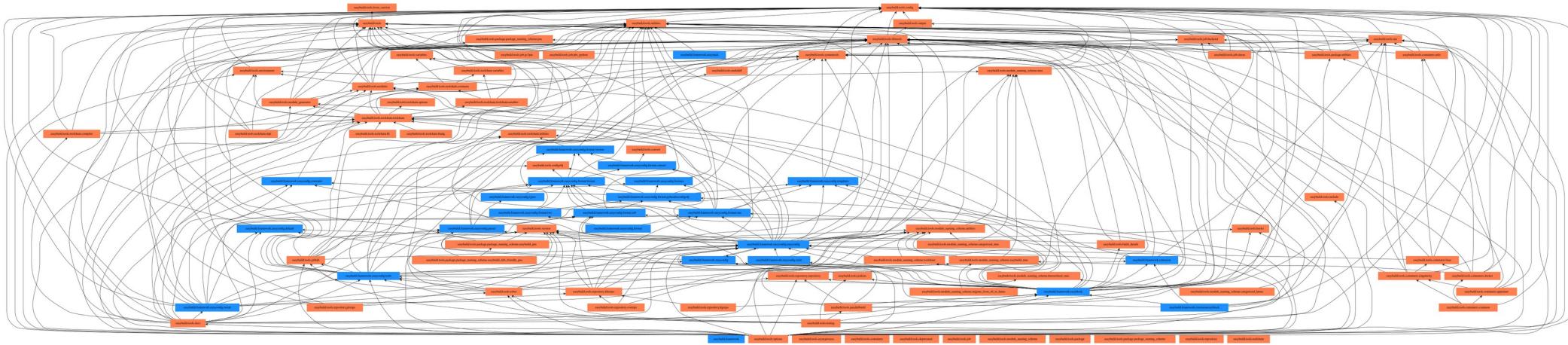
- ▶ Roll your sleeves up to make a new easyblock
- ▶ Start from scratch or on top of an existing easyblock
- ▶ Implement the **configure**, **build** and **install** steps
- ▶ Use the tooling in framework to run commands, access the file system, check dependencies...
- ▶ Some functionality is missing! You need to add functionality to EasyBuild... what now?



EASYBUILD DEVELOPER RABBIT HOLE

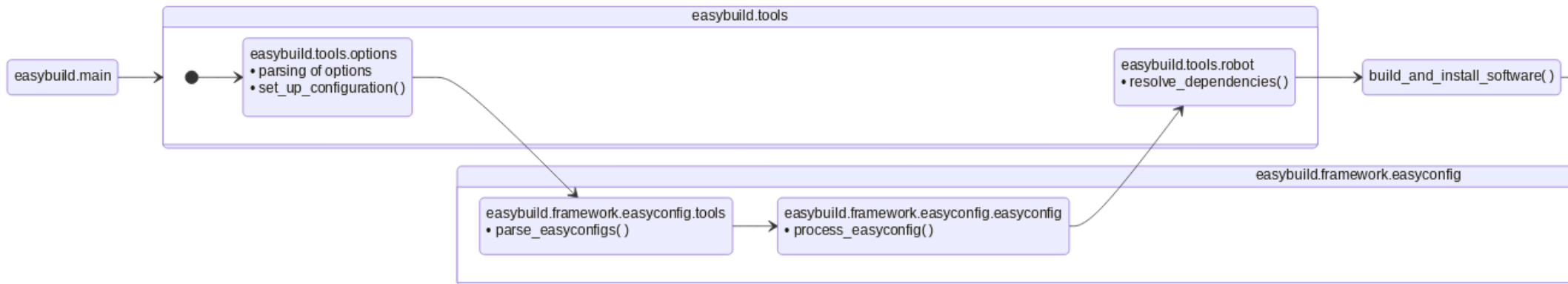
Phase 3: framework

- ▶ It's already 2:00 AM...



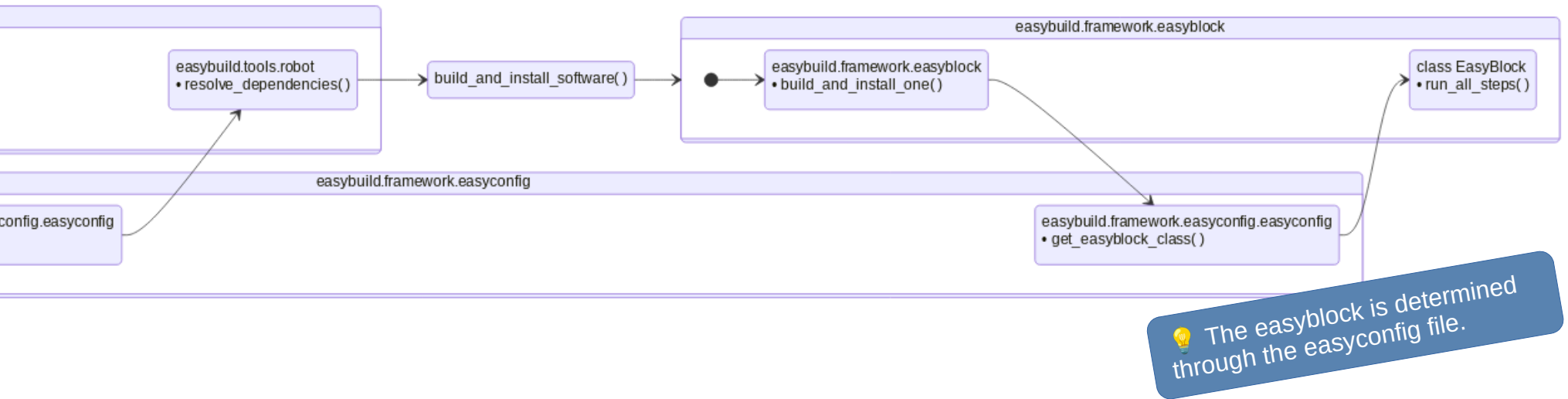
EASYBUILD WORKFLOW

What happens when you execute the eb command?

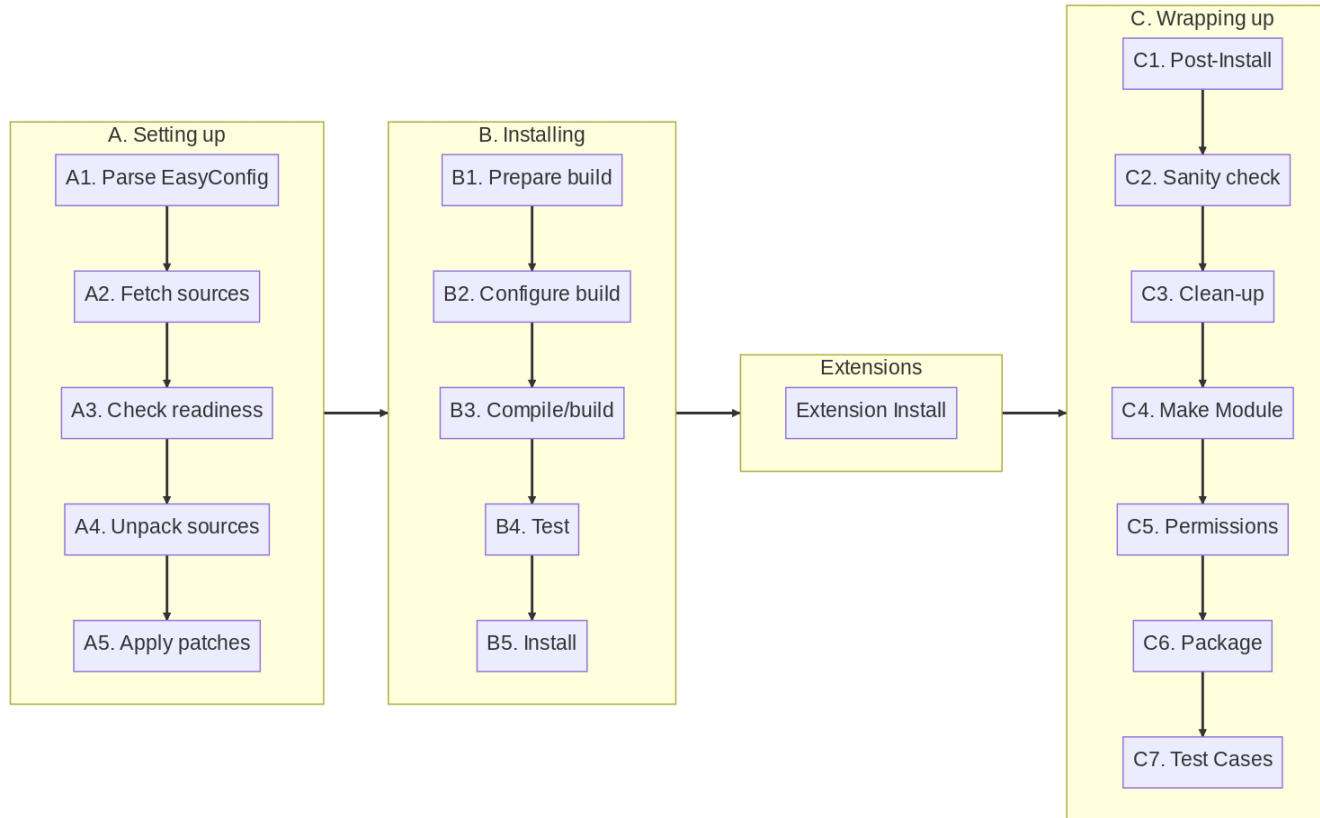


EASYBUILD WORKFLOW

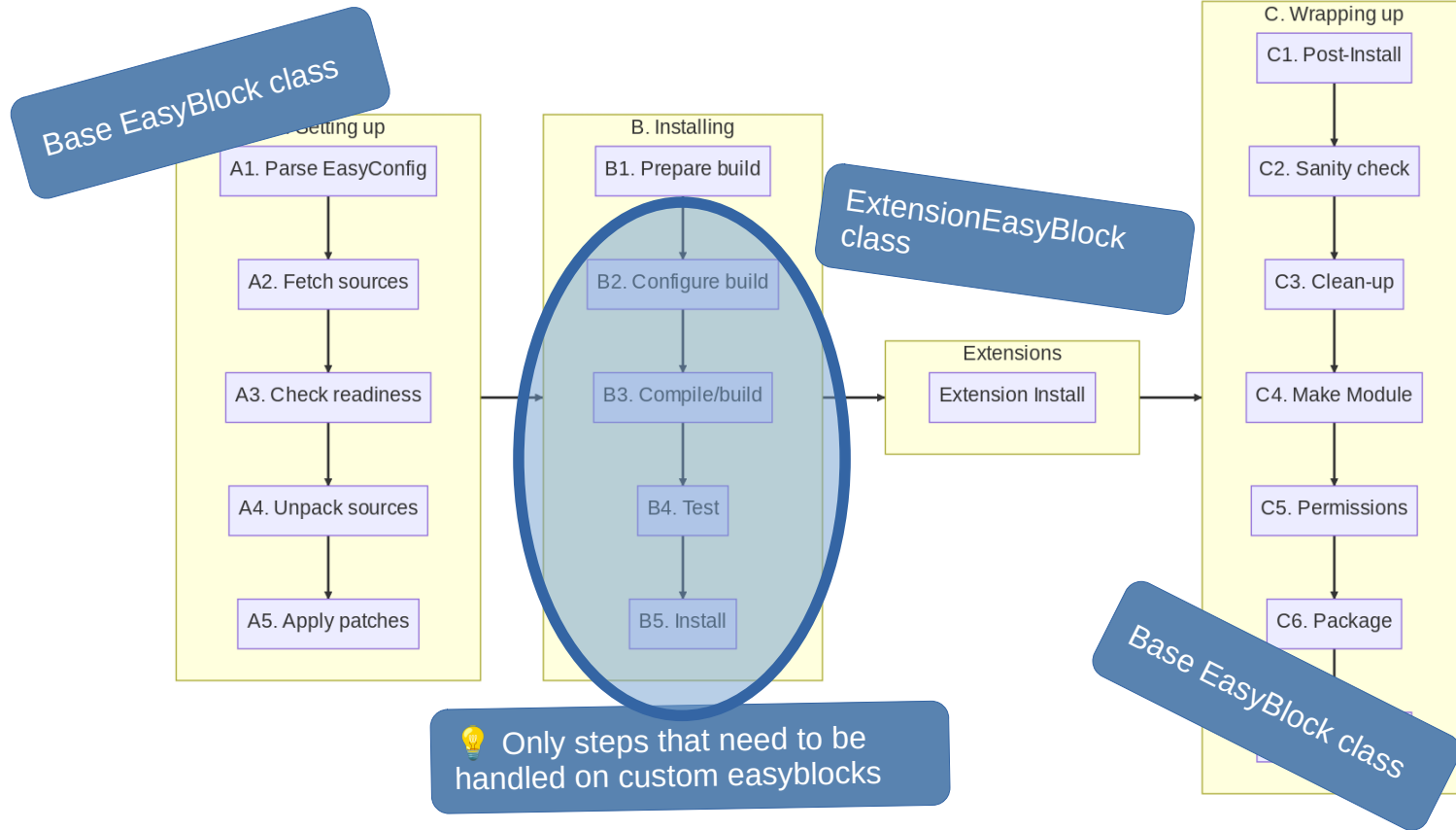
What happens when you execute the eb command?



EASYBUILD STEP-WISE INSTALLATION



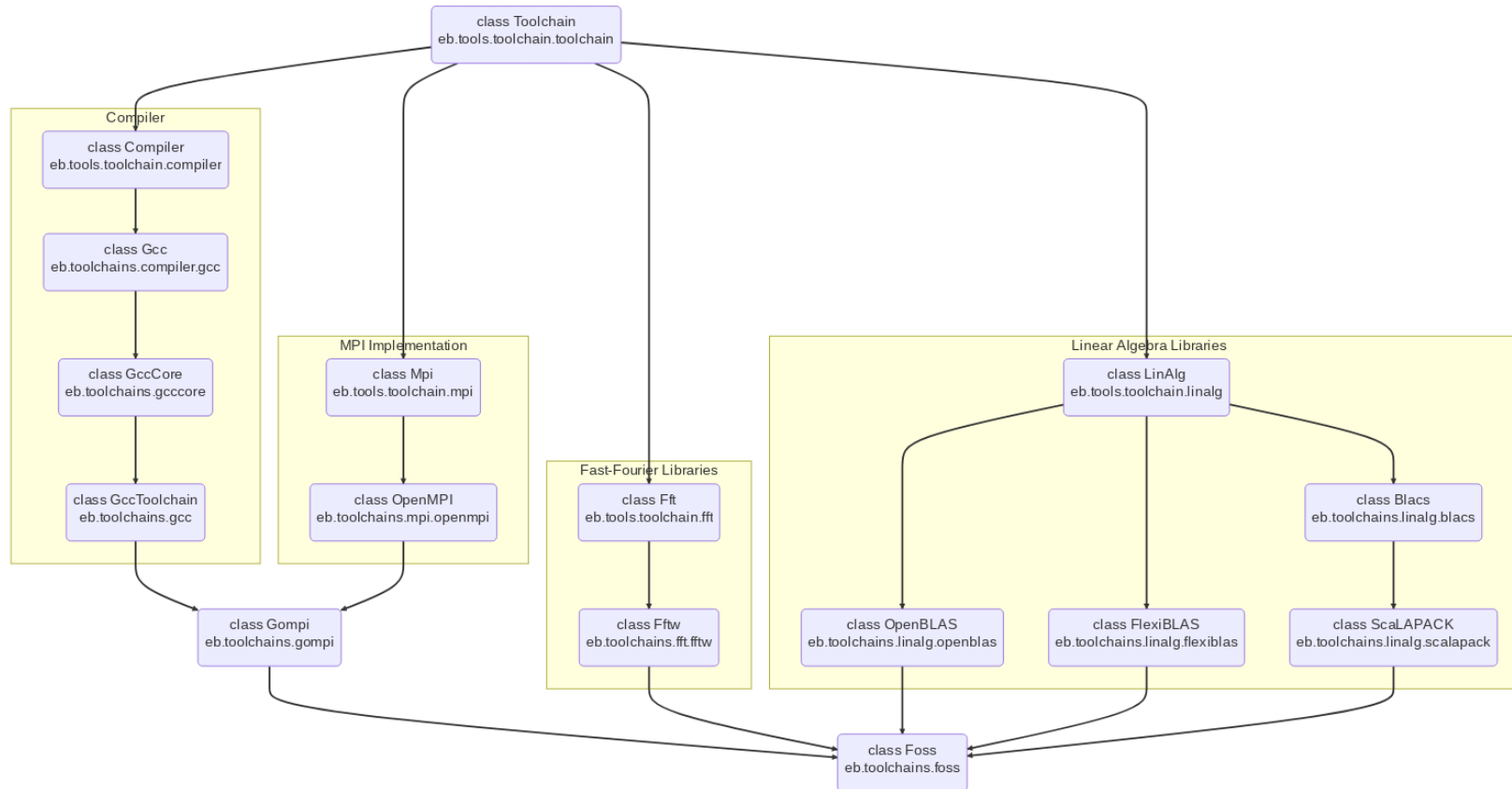
EASYBUILD STEP-WISE INSTALLATION



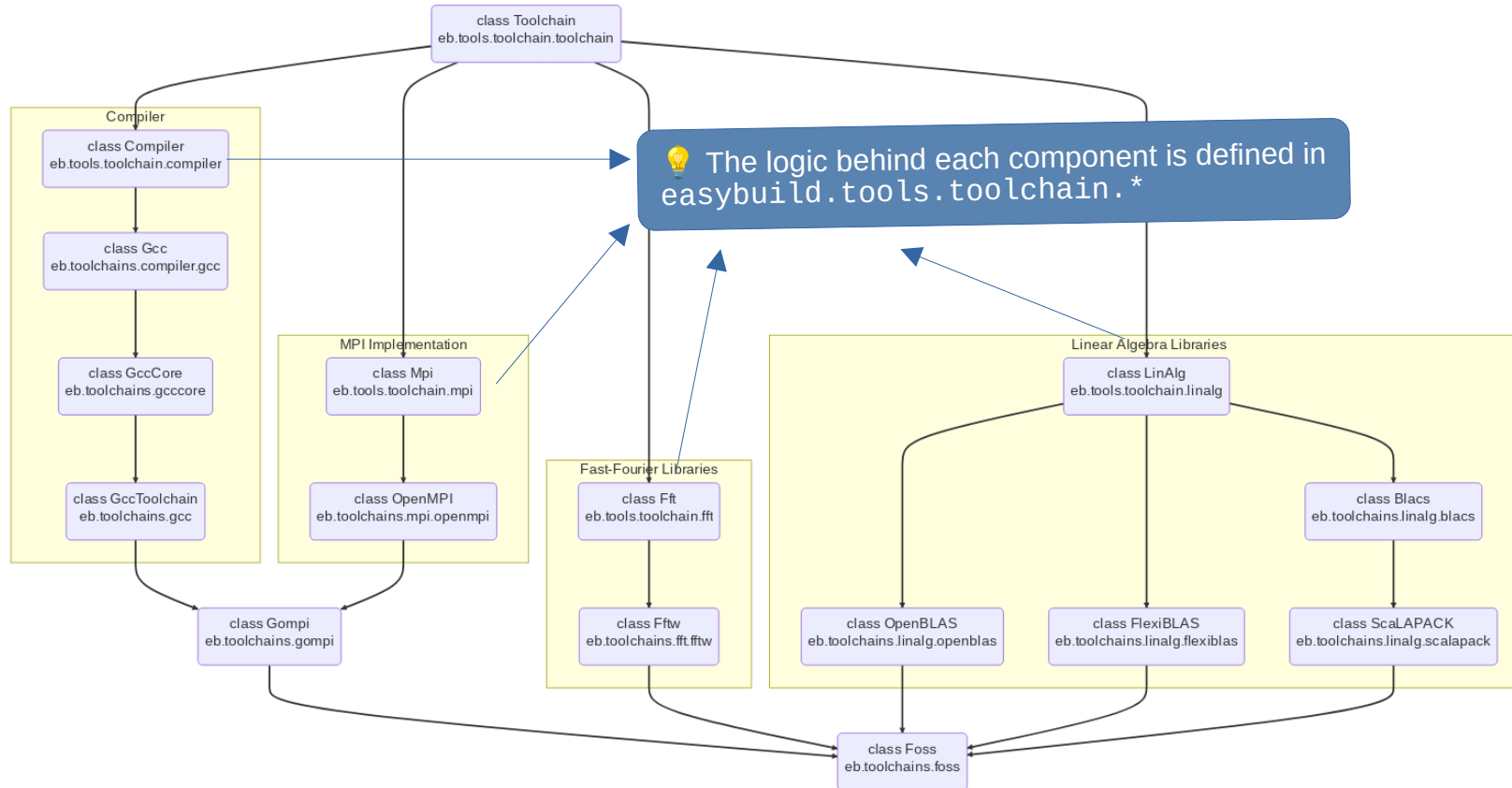
ExtensionEasyBlock class

- ▶ Extension installation: `ExtensionEasyBlock` uses the `Extension` class
- ▶ Standalone installation: `ExtensionEasyBlock` uses the `EasyBlock` class

EASYBUILD TOOLCHAINS



EASYBUILD TOOLCHAINS



Configuration options:

- ▶ `easybuild.tools.options` provides `EasyBuildOptions` with the definitions of all options
- ▶ `easybuild.tools.config` gathers all build options in `BuildOptions`, which can be easily accessed with `build_option()`

```
if build_option('ignore_test_failure'):
    print_warning("Test failure ignored")
else:
    raise EasyBuildError("Test failure")
```

Module tools:

- ▶ `easybuild.tools.modules` provides the interface to interact with the module in the host system
- ▶ `easybuild.tools.module_naming_scheme` contains the definitions of the supported MNS
- ▶ `easybuild.tools.module_generator` contains the engine to write module files used in the *Make Module* step of the installation

System information:


- ▶ `easybuild.tools.systemtools` provides information about the hardware and OS of the host system
- ▶ `easybuild.tools.environment` handles environment variables
- ▶ `easybuild.tools.filetools` provides interface to underlying file system

Others:

- ▶ `easybuild.tools.github` provides github integration
- ▶ `easybuild.tools.hooks` contains the engine running the hooks
- ▶ `easybuild.tools.containers` contains the interfaces to container systems
- ▶ `easybuild.tools.job` contains the interfaces to job schedulers


Reporting issues and bugs:

- ▶ 1. describe what you are trying to do
- ▶ 2. describe the steps to reproduce the problem
- ▶ 3. provide EasyBuild configuration and host system information
- ▶ 4. provide custom easyblock or easyconfig files
- ▶ 5. provide full error messages and tracebacks


 Might not be straightforward to determine if a bug belongs to framework

Pull requests:

 There is no integration in EasyBuild framework with GitHub

- ▶ **All unit tests must pass**
 - ▶ Bug fix: either that part of the codebase is uncovered by the tests or tests are also buggy
 - ▶ New feature: new unit tests have to be added covering the new feature
- ▶  Check EasyBuild docs on unit tests:
docs.easybuild.io/unit-tests/

CONCLUSIONS

- ▶ You can find your way around EasyBuild framework
 - ▶ Keep in mind the interplay between EasyBlock and EasyConfig
 - ▶ Keep in mind steps of the installation process and where they originate
 - ▶ Keep in mind features provided by `easybuild.tools`
- ▶ Contributing back code changes is a bit harder, but just a bit
- ▶ **All these information is now in EasyBuild docs** 
Check docs.easybuild.io/framework-overview

ACKNOWLEDGEMENTS

▶ **Ward Poelmans and Sam Moors (colleagues in VUB-HPC)**



▶ EasyBuild community

▶ VUB for hosting us

▶ VSC for financial support

