EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Installing EESSI-ready software stacks on new clusters

What’s the issue?

- Many of us are getting new clusters soon
- EESSI is not production-ready yet
  - ...and you probably still want to have a local stack anyway
- What’s the best way to set up these stacks for your new cluster?

Goals

- Have a local stack available while EESSI isn’t ready yet
- Easy transition to EESSI when it’s production-ready
  - Reduce local stack
- Allow stacks to live next to each other
  - Present it as one large, merged stack
  - Allow users to easily switch between / use both stacks

How to set up the local stack?

EESSI

- Different scenarios
  - Ranging from completely ignoring EESSI to more or less duplicating the EESSI setup
  - Each has pros and cons
- Which one has the right balance between amount of work and the functionality it provides?

Scenario 1: ignore EESSI

- Do it in the same way as you have done before
  - Make sure you're prepared to support multiple architectures

- Pros
  - Quick and easy, no changes required
- Cons
  - Will it work nicely next to the EESSI stack?
  - How to transition to EESSI?

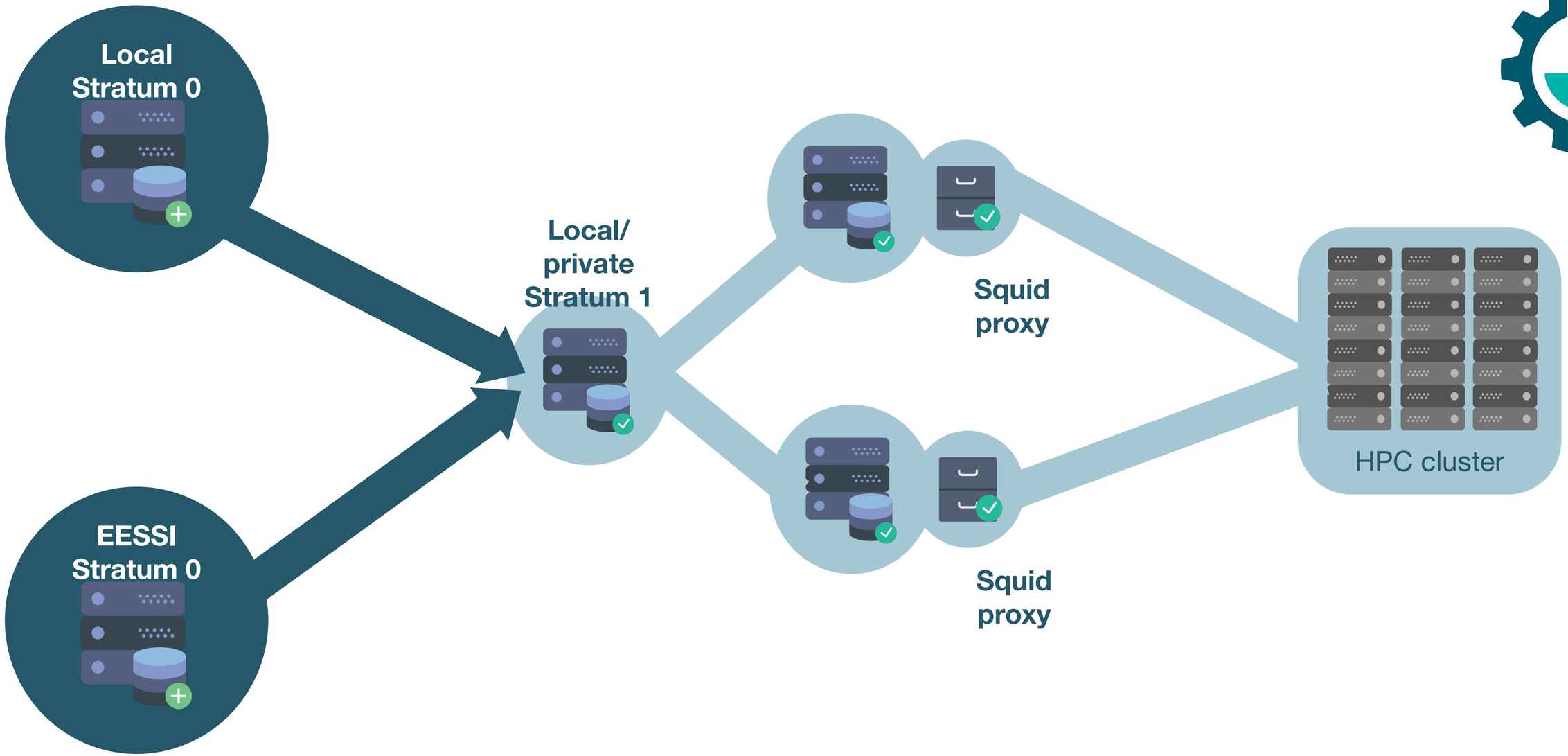
Scenario 2: the EESSI way

- Use a similar setup to EESSI
  - CVMFS + Gentoo Prefix + Easybuild + archspec + RPATH + same MNS
- Pros
  - Merging the stacks should be easy (?), e.g. by having one init script
  - Users can be allowed to mount/use the same merged stack on their PC
- Cons
  - Probably doesn’t make a lot of sense to do this
  - More work
  - Not really necessary to have Prefix; cluster OS is fixed, unless you have multiple clusters with different OSs
  - If you want to use your local stack for licensed apps, you don’t want to make it publicly available anyway

Find the right balance

- Use CVMFS (as you need some CVMFS infra for EESSI anyway…)
  - See next slide
- Use EasyBuild + RPATH support
  - Does that allow you to mix local + EESSI modules?
- What about the module naming scheme?
  - Ultimately, you could regenerate the module files of the EESSI stack for you naming scheme?
- Build your software in minimal containers to reduce chances of picking up host libraries

Possible CVMFS setup



Transition to EESSI

- Mount the repo and make it available to users:
  - by letting them source the init file, or even do that automatically?
  - by providing an EESSI module
  - by including (regenerating?) modulefiles of (some) EESSI apps in your stack
- How to deal with duplicates?
  - Deprecate and point users to same version in EESSI stack?
  - Symlink local modulefile to EESSI modulefile or regenerate the EESSI module files in your own stack?
- If you want to have the entire EESSI stack locally cached, consider setting up a private Stratum 1

What else?