# Software Environment on JURECA

27 Jan 2016 | Alan O'Cais
             a.ocais@fz-juelich.de
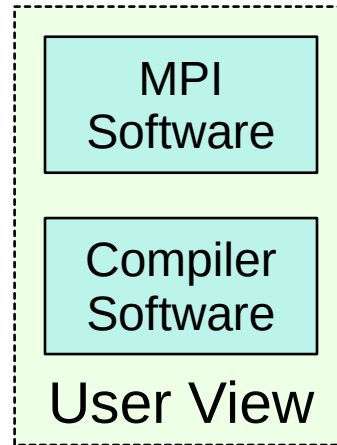
Why a hierarchy (using Lmod)?

- Compilers X MPI-Versions = module explosion

  – How to simplify naming and user view?

  – How to enforce consistency?

- Tcl/C modules not really developed any more and features very limited

  – Speed: lmod has a cache

  – Search: lmod has *spider*

  – Support for hidden modules

  – Additional features: Properties, Families,…

MPI
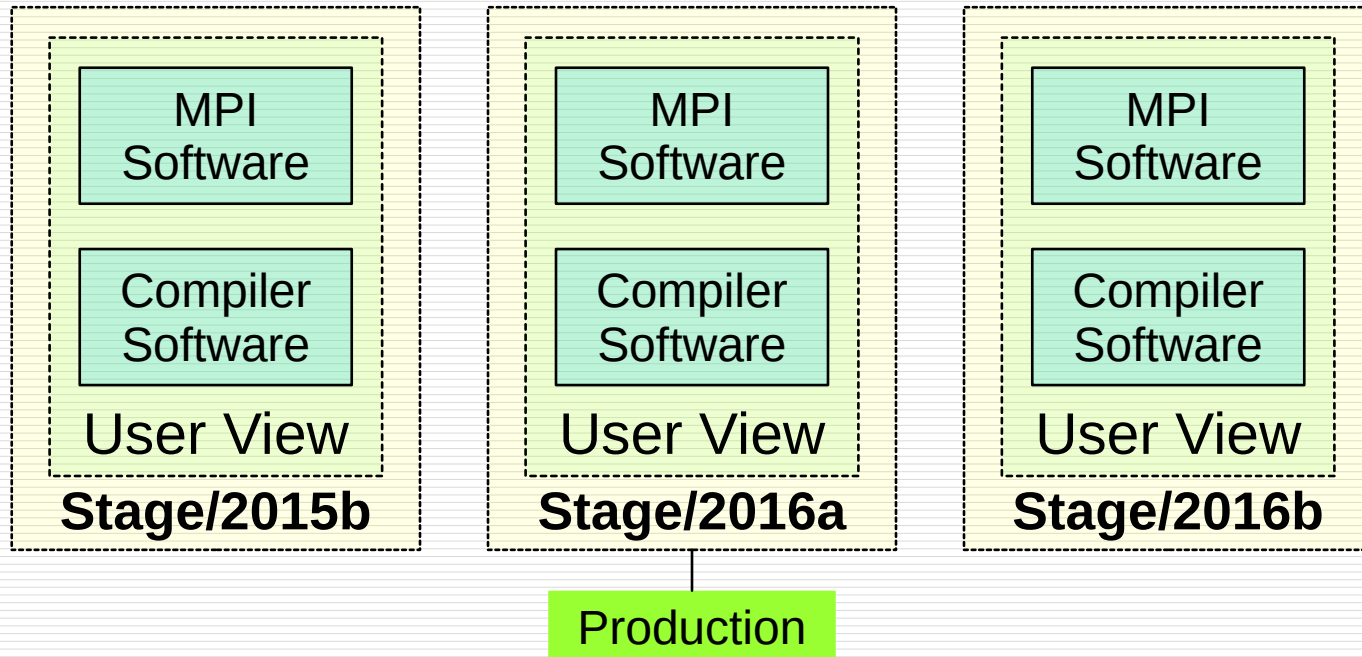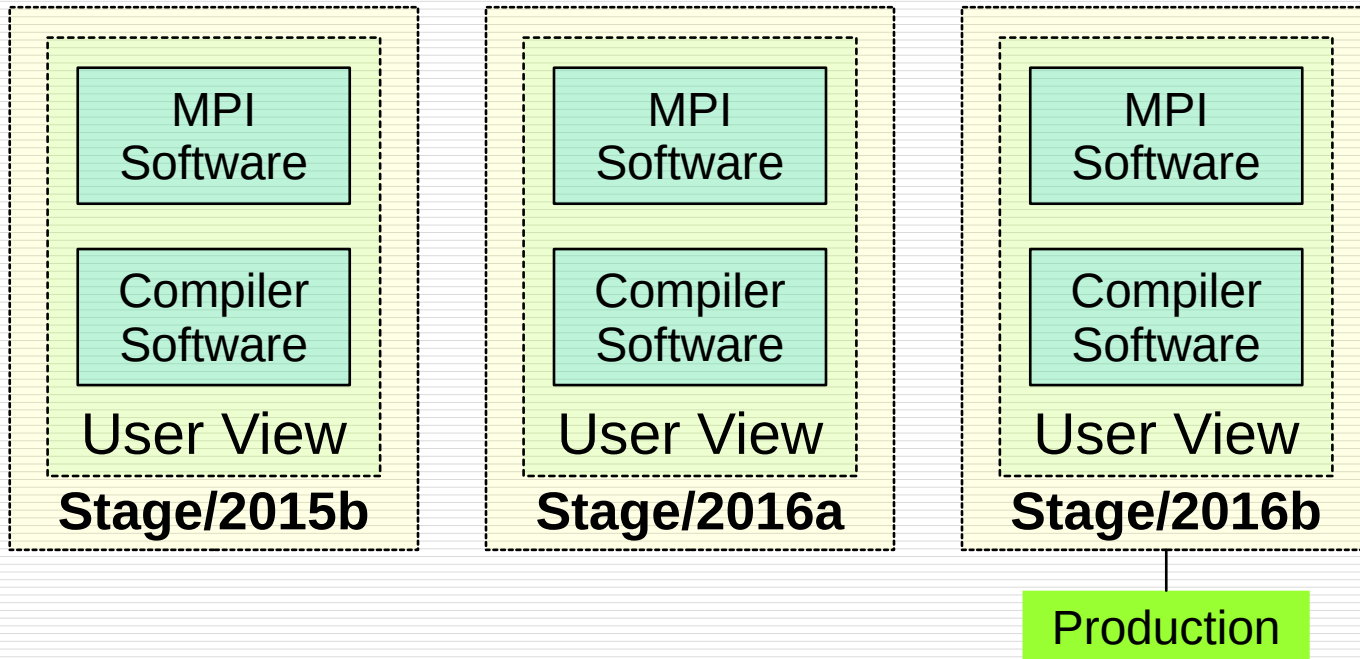Software

Compiler
Software
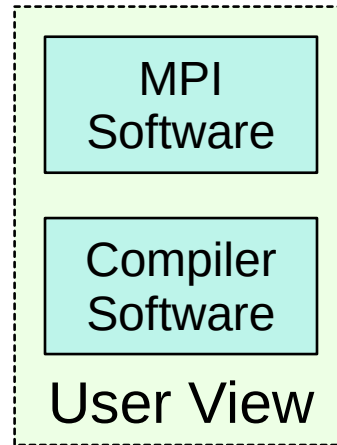
User View

Why `Stages`?

- Newest hardware means compiler upgrades can have large impact

- New toolchains and versions means *spider* output gives too many choices

- Use `Stages` to retire outdated software and manage the user view

- `Stages` allow trivial transition to entirely new software stacks
  - Including changes to EasyBuild and naming scheme

Extended View
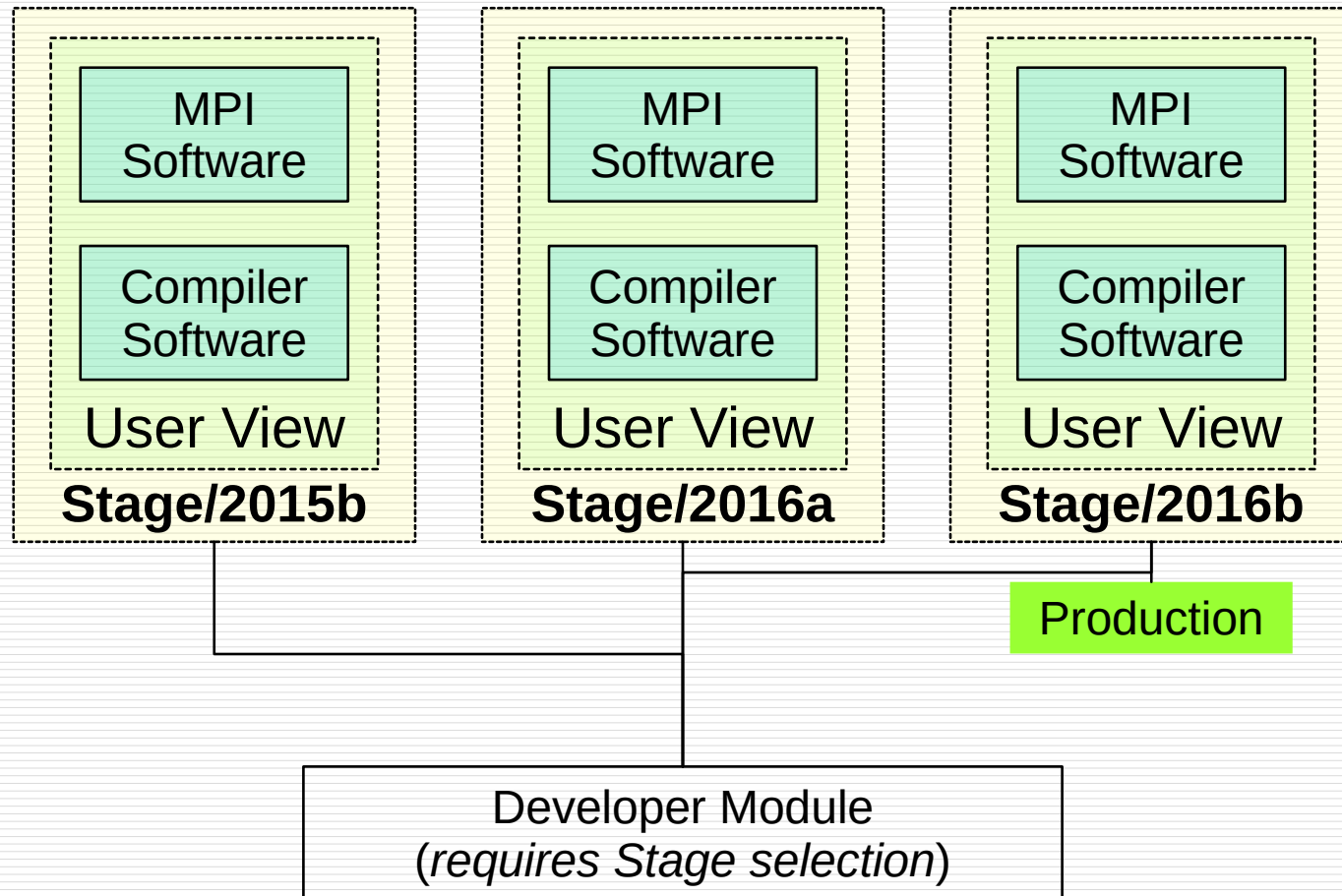
Stage/2015b	Stage/2016a	Stage/2016b
User View MPI Software Compiler Software	**User View** MPI Software Compiler Software	**User View** MPI Software Compiler Software

Production

Developer module

- Documentation and advice
- Permissions
  - EASYBUILD_SET_GID_BIT
  - EASYBUILD_STICKY_BIT
  - EASYBUILD_UMASK
  - EASYBUILD_**GROUP_WRITABLE**_INSTALLDIR
  - EASYBUILD_**READ_ONLY**_INSTALLDIR
- Software is built in RAM
  - EASYBUILD_BUILDPATH "/dev/shm/$USER"

Developer Module (2)

- Where are things installed?
  - EASYBUILD_INSTALLPATH (`Stage`)

- Resolving dependencies

  - EASYBUILD_ROBOT controls where to look for dependencies

- Submitting builds as jobs (using –job)

- All things module related

- Licences, restrictions,…

Typical `software` issues:

- EasyBuild is not *that* easy
  - Learning curve
  - Failed builds have to be restarted from scratch
  - XYZ software doesn't fit the model
- No place for static installations within Stages
- What about long term software support?
- Software responsibility, testing and maintenance?

Moving Forward

- Maintain our own *Golden* EasyBuild repo
  - Easyconfigs in this branch have priority
  - Development in `Stages/Devel` only
  - Code review by `swmanage`
    - Only easyconfigs in golden repo installed to Production
    - Keep a legacy repository of all previously installed easyconfigs from Production
  - Distinguish between supported software and dependencies
    - Actively search for updates to supported software
    - Hide dependency modules by default?

Standards in module files

- Relevant to supported software
- Maintainer and email contact should be listed
- Configuration of installation (installed modules, compiler options used...)
- Should provide usage information
- What else?
- EasyBuild will soon (?) support templates to enforce this (restrict enforcing to supported?)

What has JSC contributed?

- GCCcore concept

- Minimal toolchains

- MNS independent installation directories

- Configurable default optimisation

- Hierarchical module naming scheme

- Bug fixes and support

JÜLICH
FORSCHUNGSZENTRUM

What do we worry about?

- Environment explosion

- Dependencies creating clutter

- Version minimisation

- GPU-enabled software

- inodes!!

Comments and Questions

Mitglied der Helmholtz-Gemeinschaft