



UNIVERSITY OF LEEDS

Containers and HPC: Singularity at Leeds

Martin Callaghan

Advanced Research Computing, University of Leeds

www.arc.leeds.ac.uk



Introduction

I'm Martin Callaghan

Research Computing Consultant at the University of Leeds



This talk

- Introductions and ethos
- Our journey to containerisation
- Supporting users
- Our approach to building containers
- Case Study 1: Deep Learning and GPUs
- Case Study 2: WRF and MPI
- Case Study 3: R and 'strange' dependencies
- Conclusion and Questions



HPC and Research Computing at Leeds

Small team (just 4 of us)

5 clusters (~15000 cores, GPUs, Xeon Phis)

~700 active research users

Traditional HPC users

New users in Biological Sciences, Business & Economics, Humanities, Data Analytics

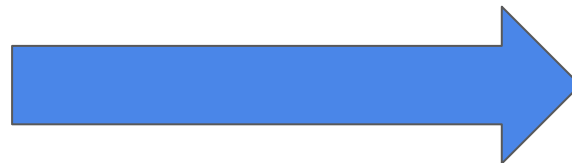
Need to remain agile and responsive to changing research needs



Our journey towards containers

Pressure from new (and existing) users:

- New codes & applications
- Interactivity
- Visualisation



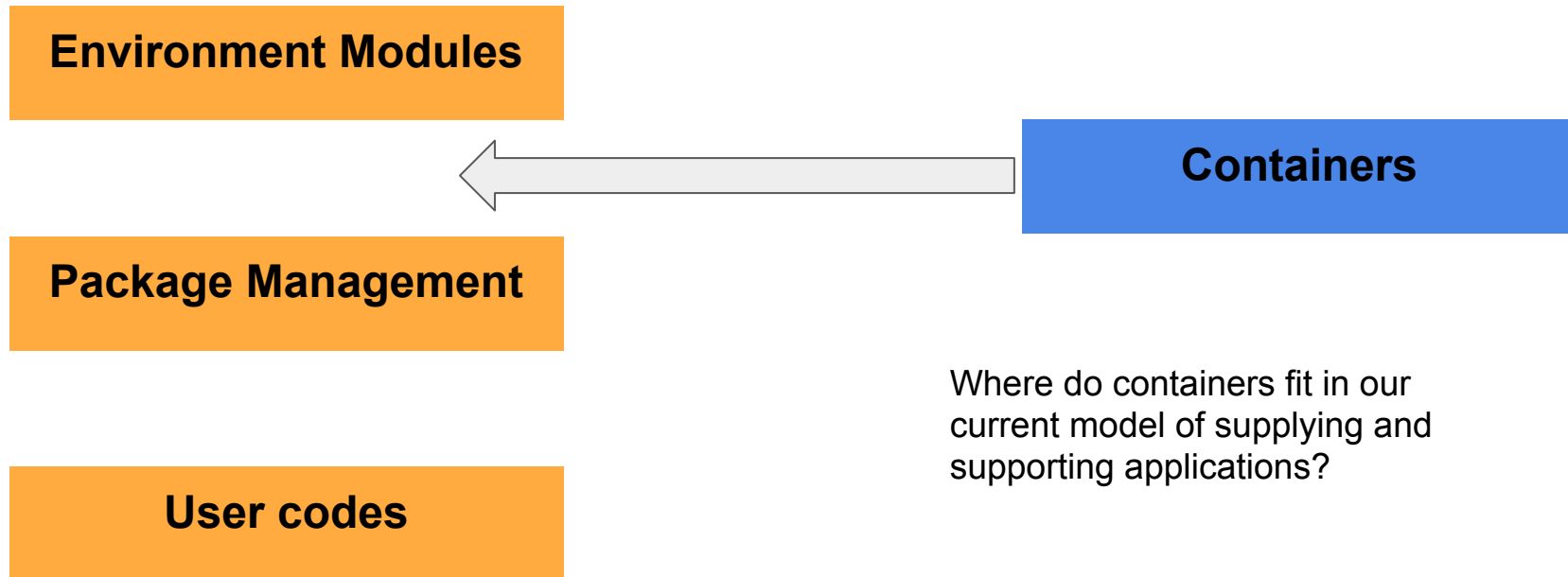
Pressure from technology:

- Cloud
- Accelerators
- New languages and paradigms





Managing applications



Where do containers fit in our current model of supplying and supporting applications?



User support and training

HPC and Research Computing isn't about doing things

FOR or **TO**

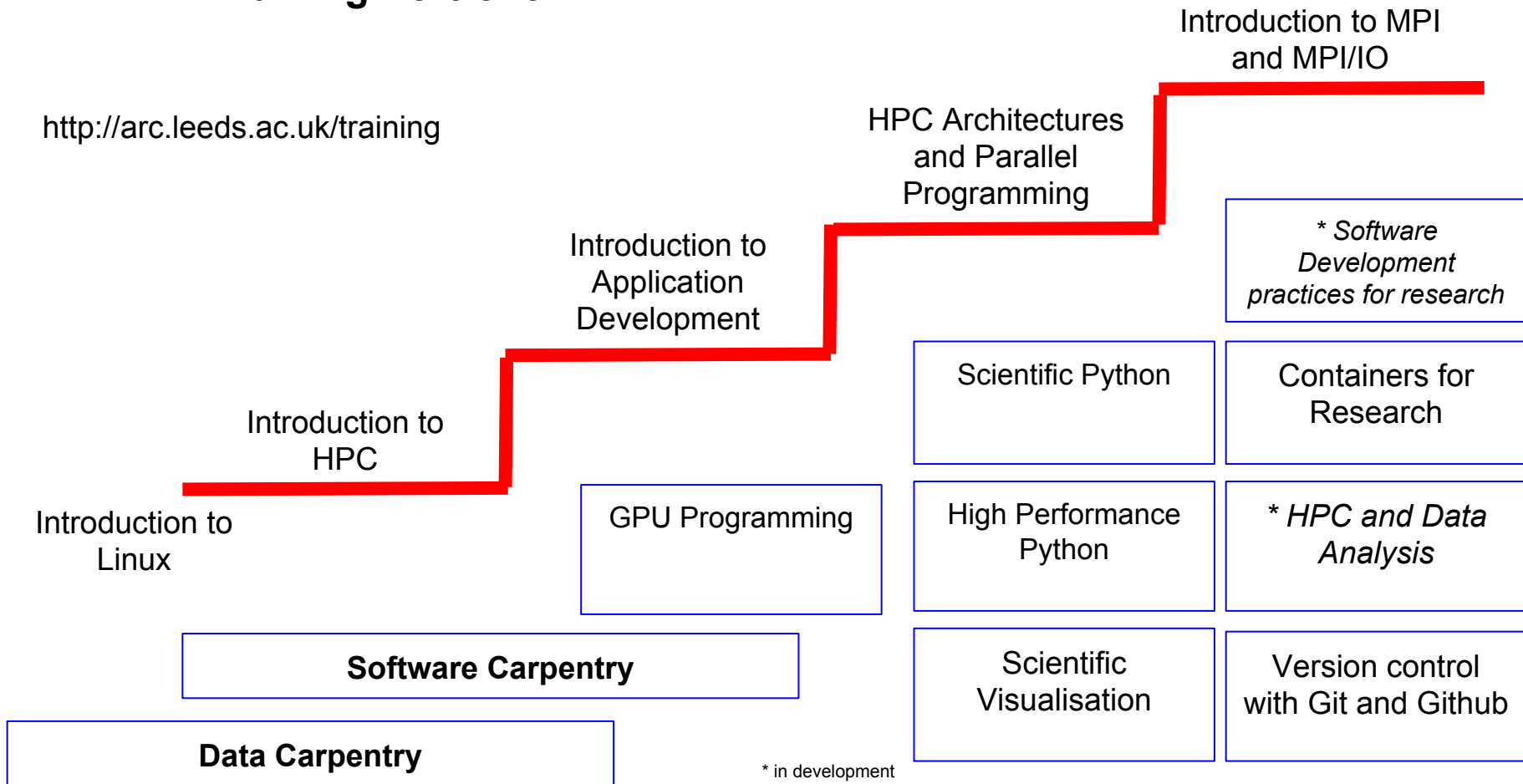
It's about doing things

WITH

Support, Training, Consultancy and Outreach go a long way towards promoting the ethos of **WITH** and developing trusting partnerships.

ARC/HPC Training Portfolio

<http://arc.leeds.ac.uk/training>





What's a container?

Containers allow us to:

- Wrap up a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries all on top of your favourite Linux distro.
- Almost a Lightweight VM, but containers share the kernel with other containers and processes, running as isolated processes in user space.
- A good way of providing a reproducible environment.



What problems can they help solve?

- Dependency Hell
- Imprecise Documentation
- Barriers to adoption and reuse
- Code rot: esp. outdated software dependencies
- Users who need a 'special' and quickly...
- Increasingly, software is available prebuilt in a container (OpenFoam, WRF)
- Sharing stacks

Could use...

VMs, Workflow tools, Package Managers

The case for containers in research

Container technologies can support **reproducible computational research** by allowing a complete research environment to be **scripted** and **shared**.



Our approach to building containers

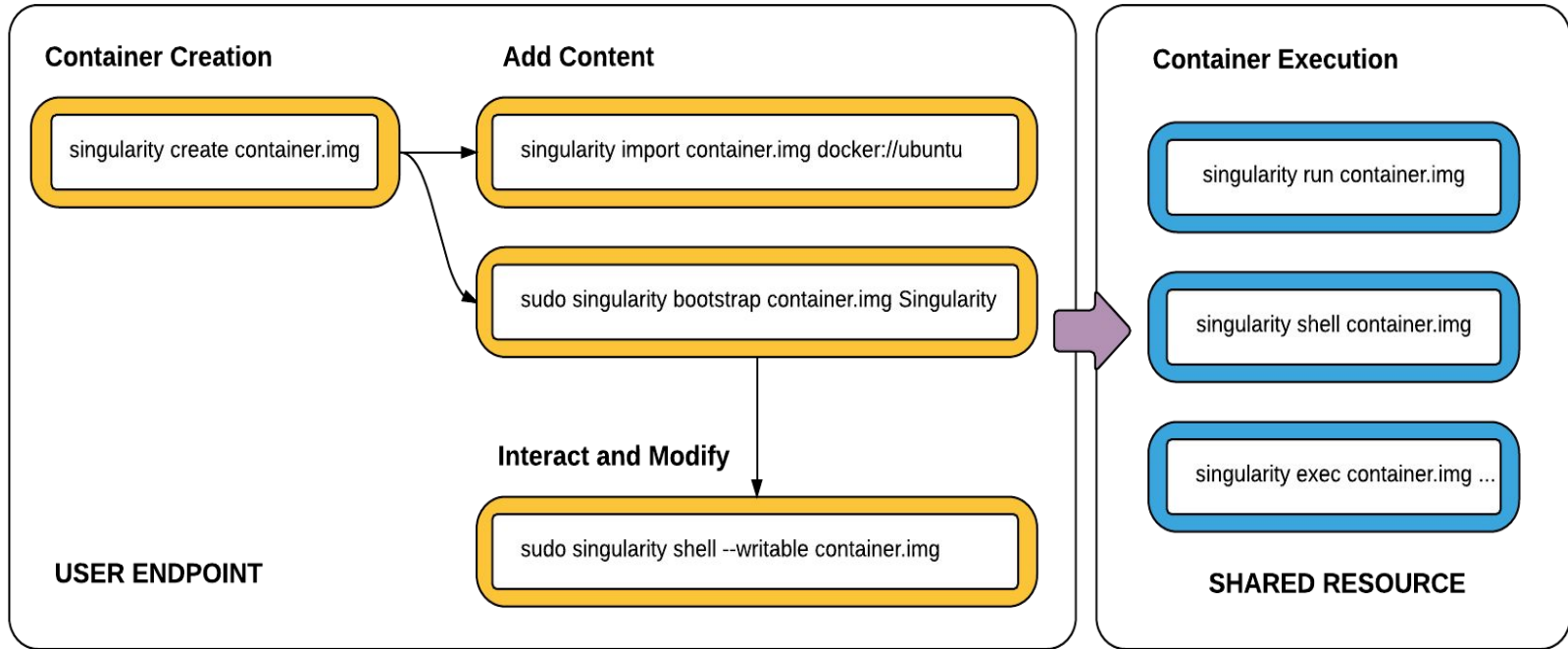
Layers and templates: Building Blocks

Script everything: no writable containers

Let users do as much as possible (we supply a VM)

No VM or root access?

Singularity workflow

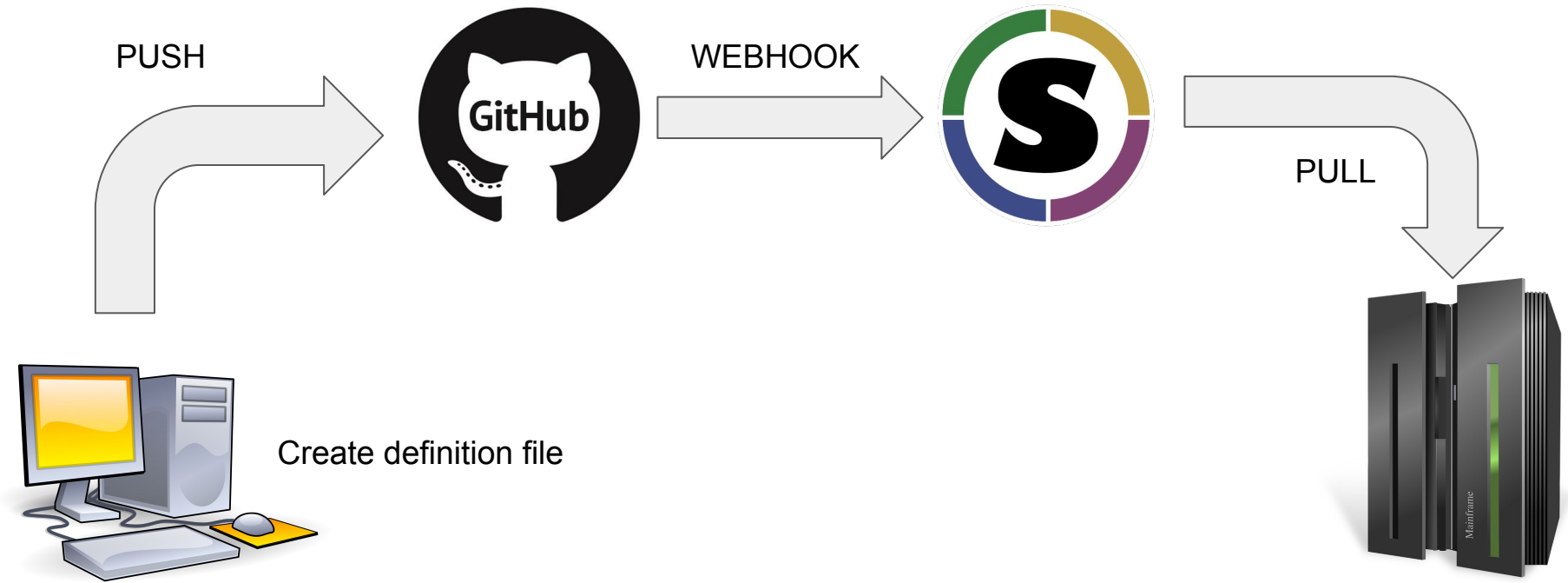




Leverage the available tools

Integrate Github and Singularity for a CI workflow:

[Singularity Hub:](#)
BUILD and TEST





Case Study 1: Deep Learning and GPUs

New researchers with new needs:

- Theano
- Tensorflow
- Keras
- Python
- Jupyter Notebooks
 - Interactive use
 - Batch use



Case Study 1: Solution

Single container with complete stack

Will run Jupyter Notebook

Some (clunky) SSH tunnelling and users can access a JN server running on a GPU compute node from their local browser.





Case Study 2: WRF

- WRF= Weather Research and Forecasting
 - A popular climate modelling application
 - Can be a little awkward to install- lots of dependencies
- This was a good opportunity to test out Singularity's MPI compatibility (we used OpenMPI).
- It works
- No appreciable overheads to using the containerised model over regular build



Case Study 3: R and strange dependencies

Fork of R

User familiarity with a single Linux distribution

Cascading and complex dependencies

R container runs as application under SGE as task array to calculate cycling routes.



Other interesting possibilities

Apache Spark

NoSQL databases

Windows applications running under WINE

Checkpointing containers with DMTCP

Just using containers for some applications, eg. OpenFOAM

Any questions?



UNIVERSITY OF LEEDS