Basic introduction to software containers

Application in scientific practice

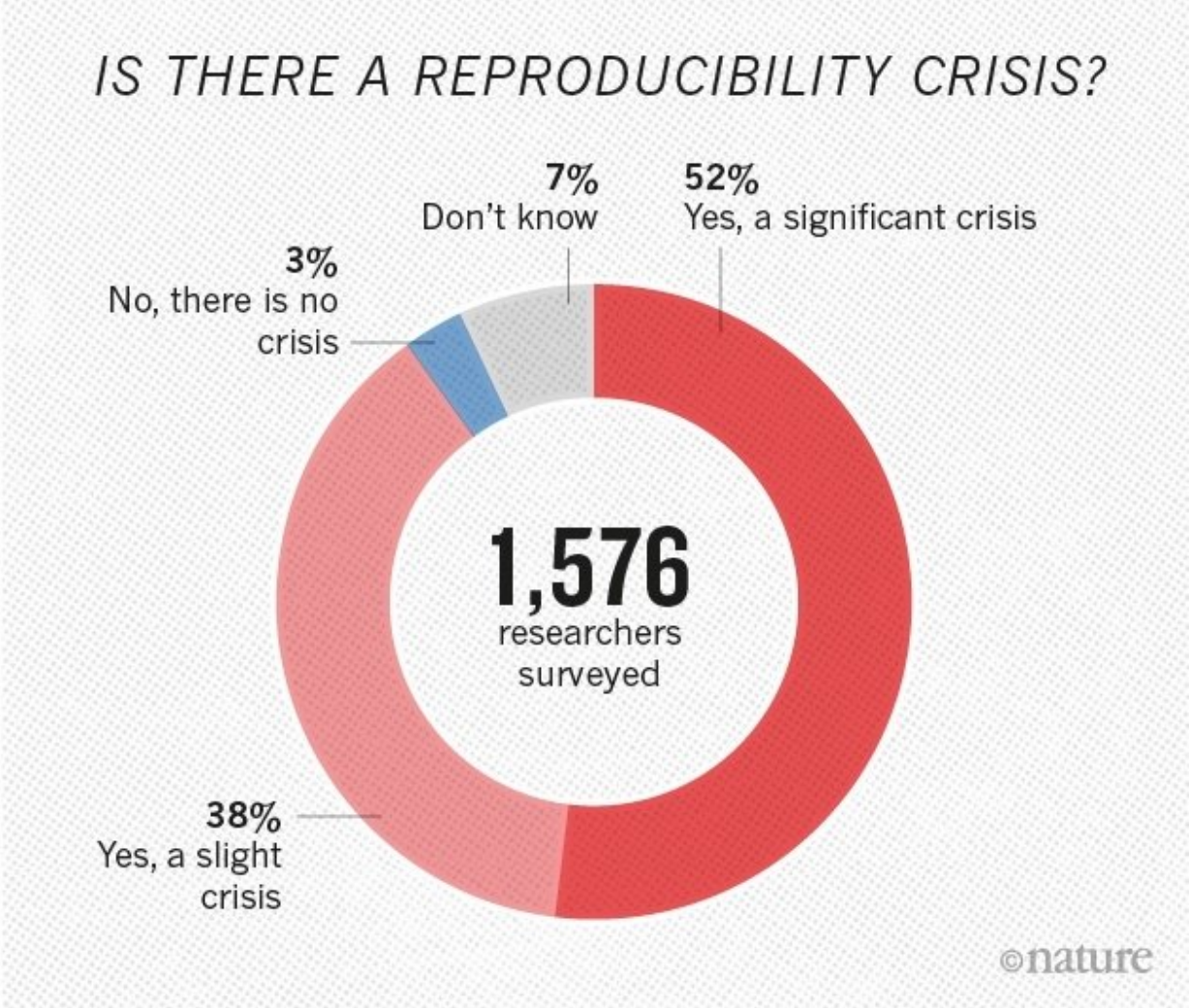
Docker and Singularity

Toni Hermoso Pulido

Bioinformatics Unit

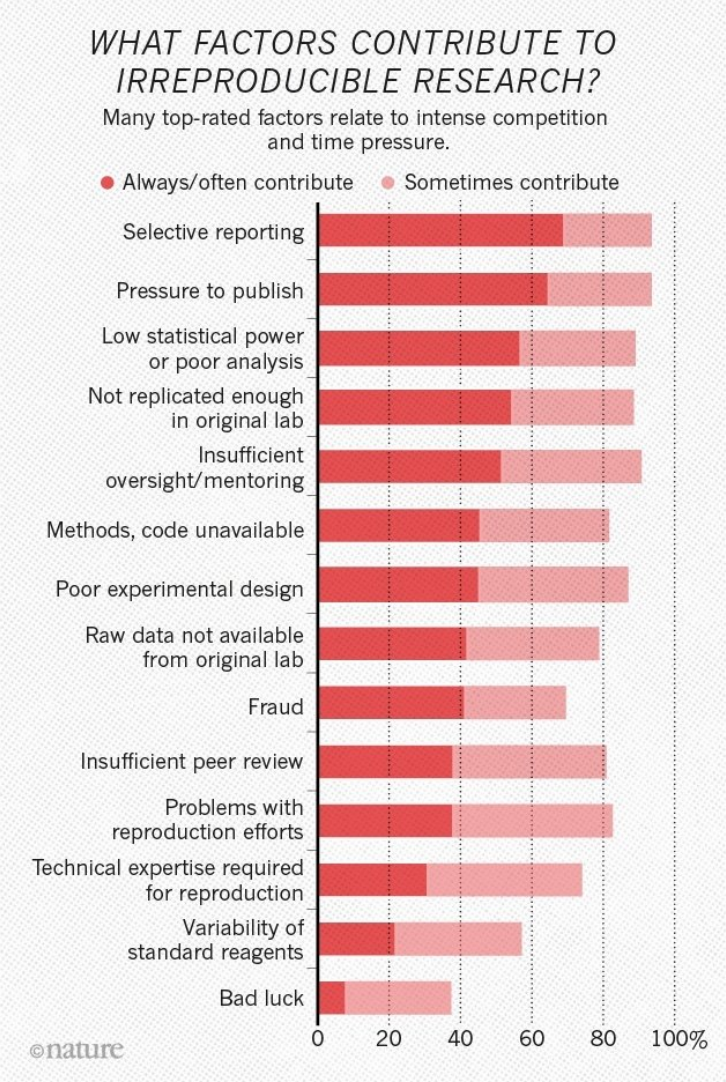
CRG, Barcelona

Reproducibility Crisis



According to a 2016 poll of 1,500 scientists reported in the journal Nature, 70% of them had failed to reproduce at least one other scientist's experiment (50% had failed to reproduce one of their own experiments). Ref

Reproducibility Crisis



According to a 2016 poll of 1,500 scientists reported in the journal Nature, 70% of them had failed to reproduce at least one other scientist's experiment (50% had failed to reproduce one of their own experiments). Ref

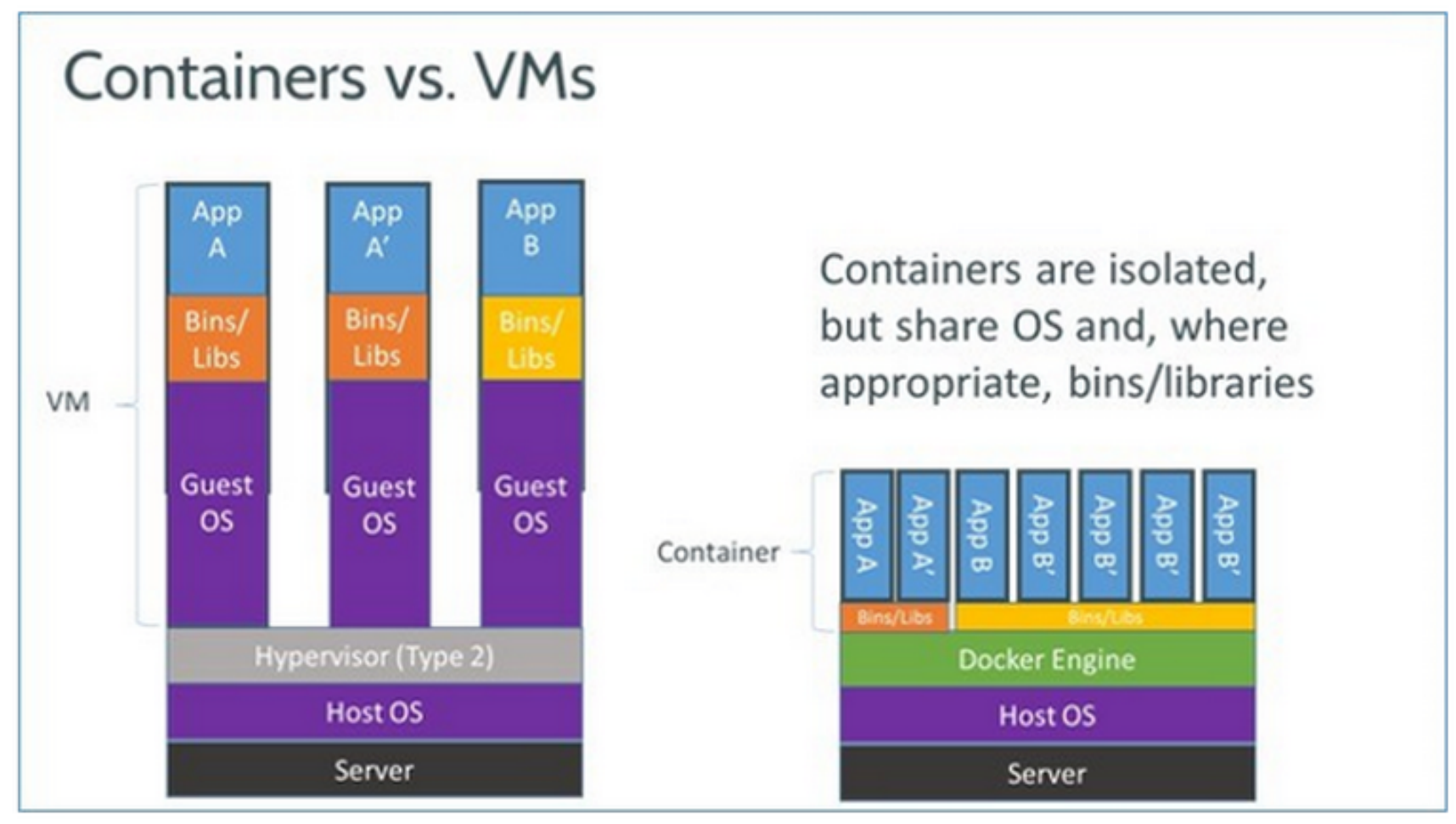
Containers

Containers in New Jersey

Containers in science

- Mantainability
- Portability
- Reproducibility

Virtual machines vs containers



http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

Virtualisation

Pros and Cons

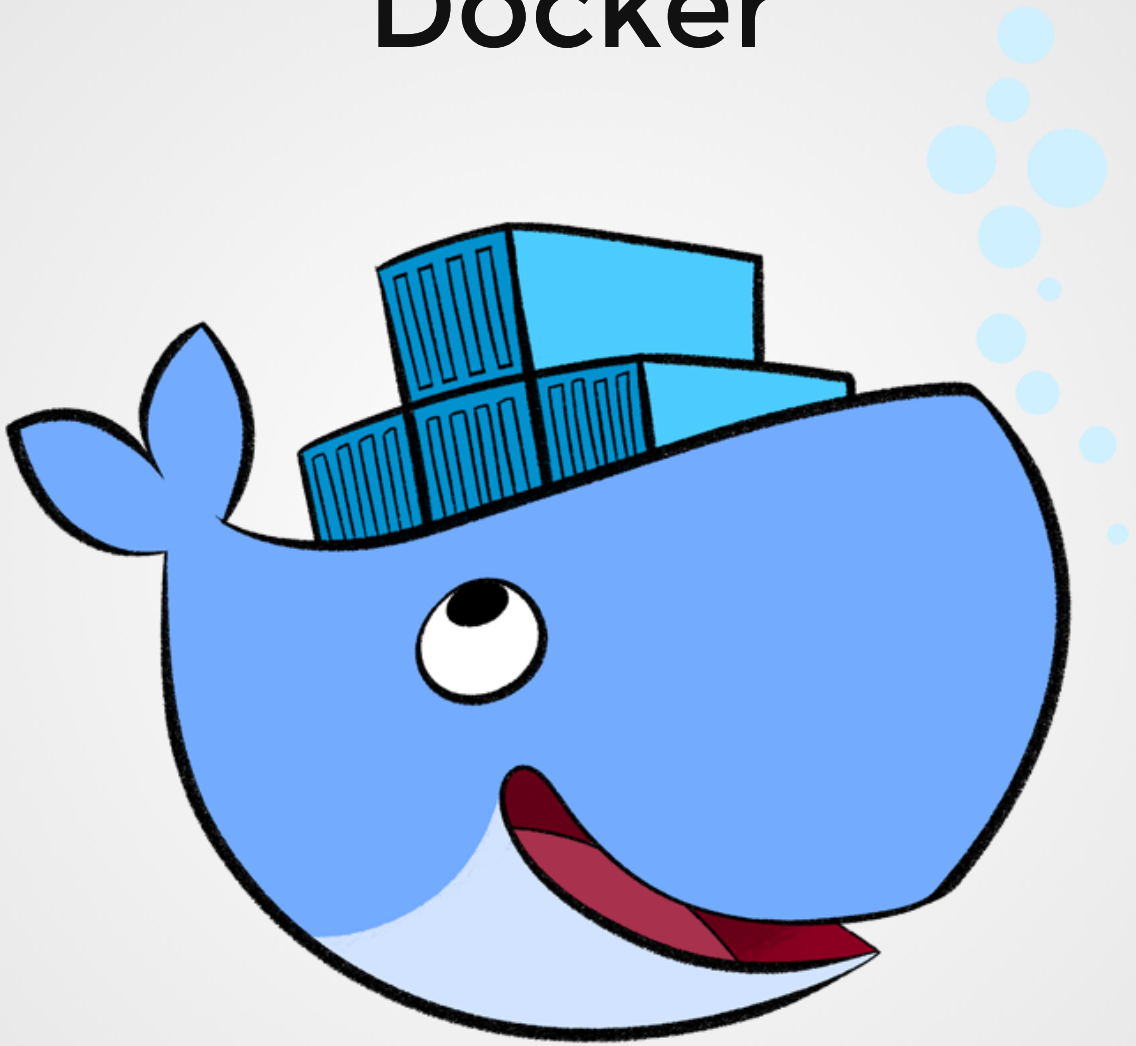
- **PRO**: Very similar to a full OS
- **PRO**: With current solutions, high OS diversity
- **CON**: Need of more space and resources
- **CON**: Slower than containers
- **CON**: Not as good automating

Containerisation

Pros and Cons

- **PRO**: Faster
- **PRO**: No need of full OS installation. Less space.
- **PRO**: Current solutions allow easier distribution of recipes. More portability
- **PRO**: Easier automation
- **CON**: Some cases might not be exactly the same as a full OS
- **CON**: With current solutions, still less OS diversity

Docker



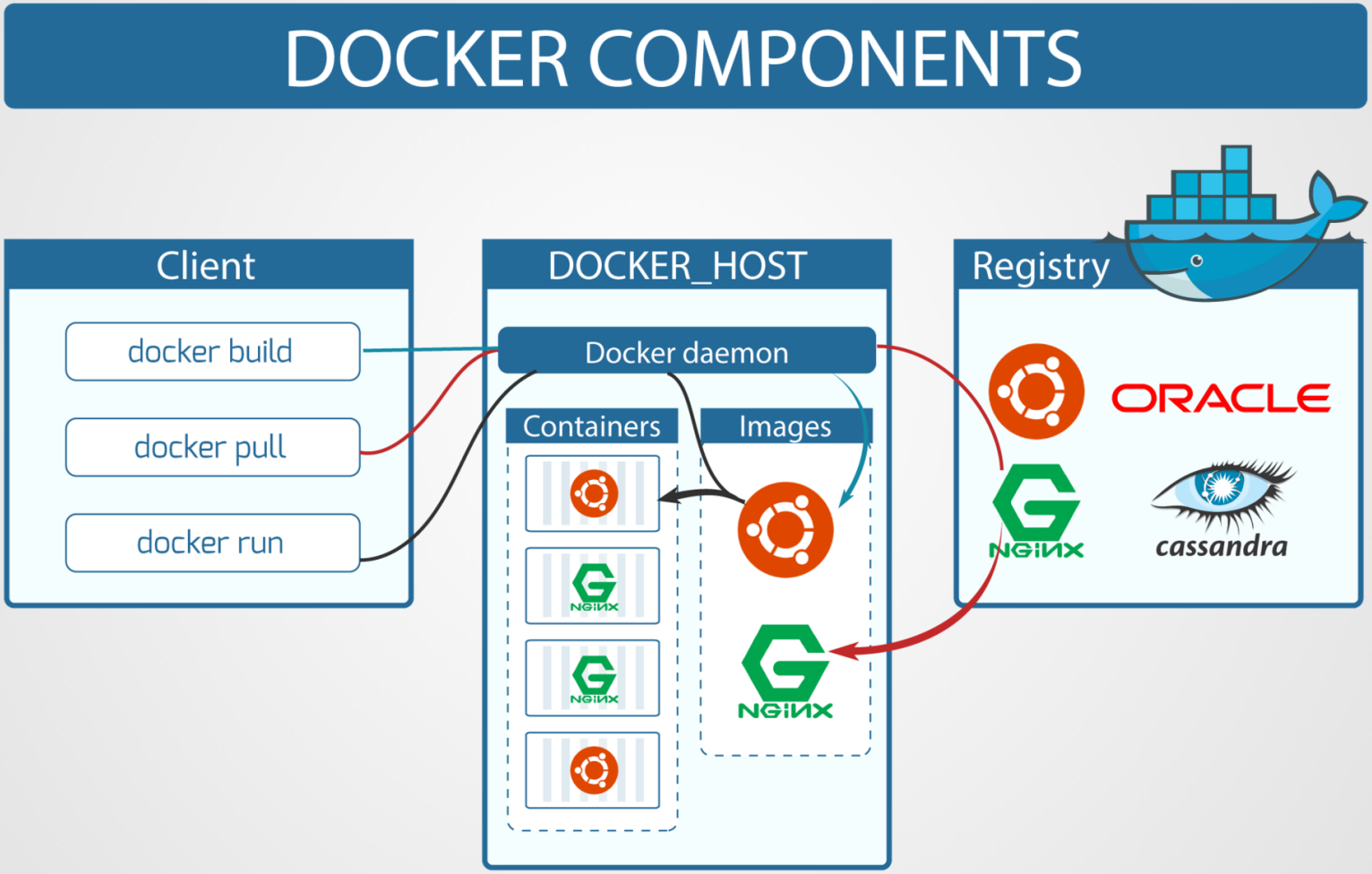
Docker

- Platform for developing, shipping, and running applications
- Infrastructure as application/code
- Established Open Container Initiative

As a software:

- Docker Community Edition
- Docker Enterprise Edition

Docker architecture



Docker image

- Read-only templates.
- Containers are run from them
- Images are not run
- Images have several layers

Docker image - Instructions

- Recipe file:
  - `Dockerfile Reference`
- Instructions
  - *Every instruction generates an image layer*
  - FROM: use a base image (notice tag)
  - ADD, COPY: add files to image filesystem
  - RUN: execute command in image
  - ENV, ARG: Run and build environment variables
  - CMD, ENTRYPOINT: Command to execute when generated container starts

Docker container

- Generated from an image (template)
- Image: read-only
- Container: read-write
- Can be converted into image
  - `docker commit`
- 1 imatge -> n diverse containers
  - Diversity:
    - Volumes / Mounting points
      - Different data or configs
    - Different exposed ports

Run container

```
$ docker run biocorecrg/c4lwg-2018 /bin/echo "Hello world!"
```

Docker registry and Docker hub

- Images are stored locally
- They can also be shared in a registry
- Main Public one: Docker hub

Examples:

https://hub.docker.com/u/biocorecrg/

https://github.com/CRG-CNAG/docker-debian-perlbrew

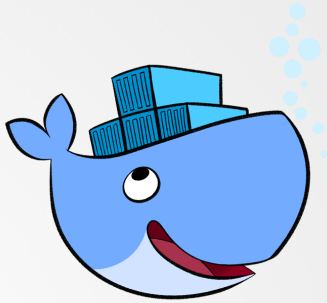
Singularity

containers for HPC

https://singularity.lbl.gov/

https://doi.org/10.1371/JOURNAL.PONE.0177459

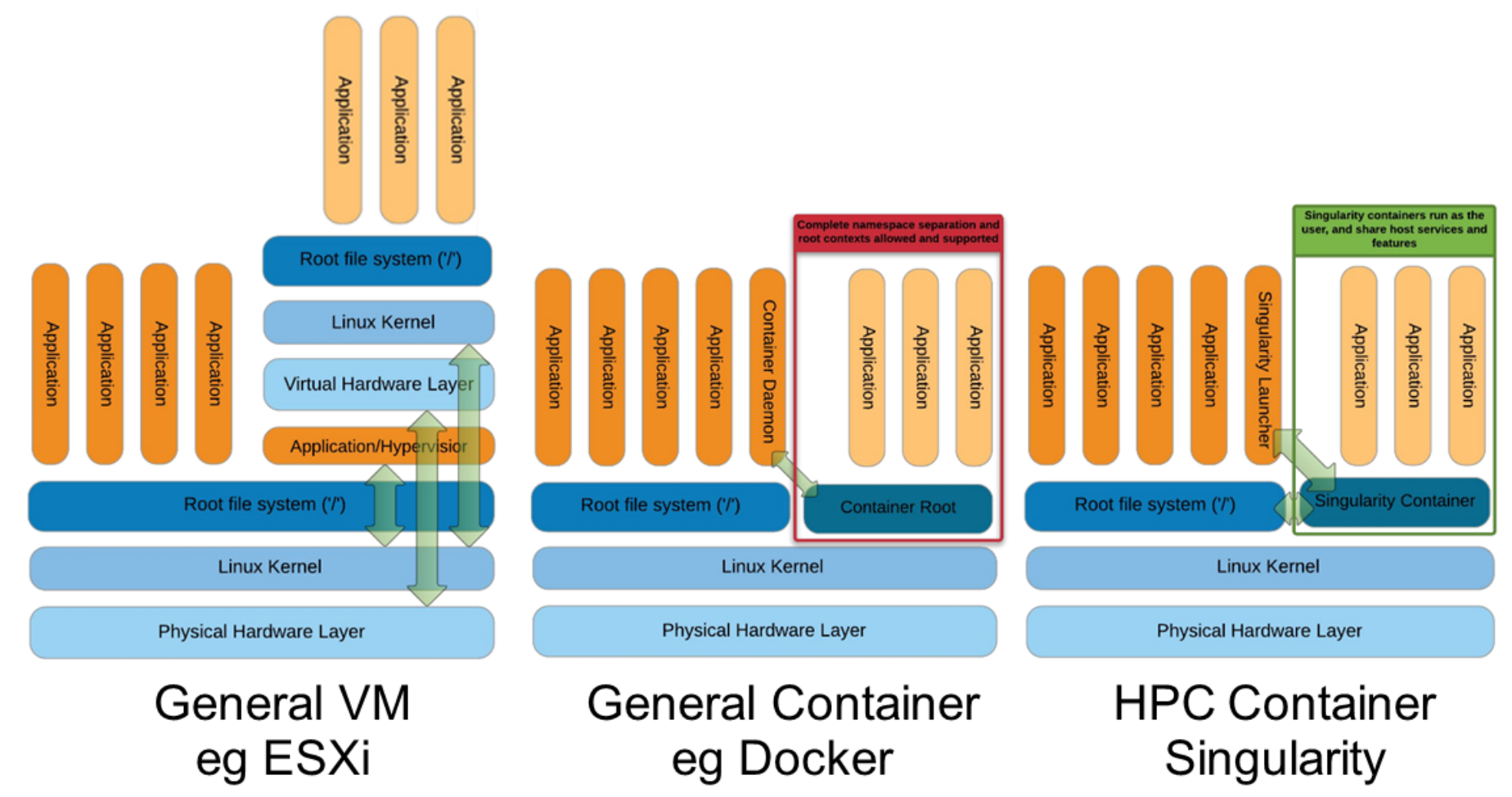
Singularity vs Docker



Summarising

- Docker -> Microservices
- Singularity -> HPC

Singularity architecture



http://geekyap.blogspot.com.es/2016/11/docker-vs-singularity-vs-shifter-in-hpc.html

Singularity - Strenghts

- No dependency of a daemon
- Can be run as a simple user
- Image/container is a file (or directory)
  - More easily portable
- Two type of images
  - Read-only (production)
  - Writable (development)

Singularity - Weaknesses

- At the time of writing only good support in Linux
  - *Not a big deal in HPC environments, though*
- For some uses you need root account (or sudo)
- Still young project compared to other solutions

Singularity - run

Execute a command

```
$ singularity exec c4lwg-2018.simg /bin/echo 'Hello world'
```

Execute a command (with clean environment)

```
$ singularity exec -e c4lwg-2018.simg /bin/echo 'Hello world'
```

Execute a shell

```
$ singularity shell c4lwg-2018.simg
```

Execute defined runscript (parameters can be used)

```
$ singularity run c4lwg-2018.simg
```

Scientific containers good pratices

- Put data and configuration files outside of images
  - Mount them if necessary
- Choose specific software/distribution versions
  - Not latest tags
- Save container recipes
- Save also binary container/images if possible

Further reading

- The impact of Docker containers on the performance of genomic pipelines.
- Performance Evaluation of Container-based Virtualization for High Performance Computing Environments