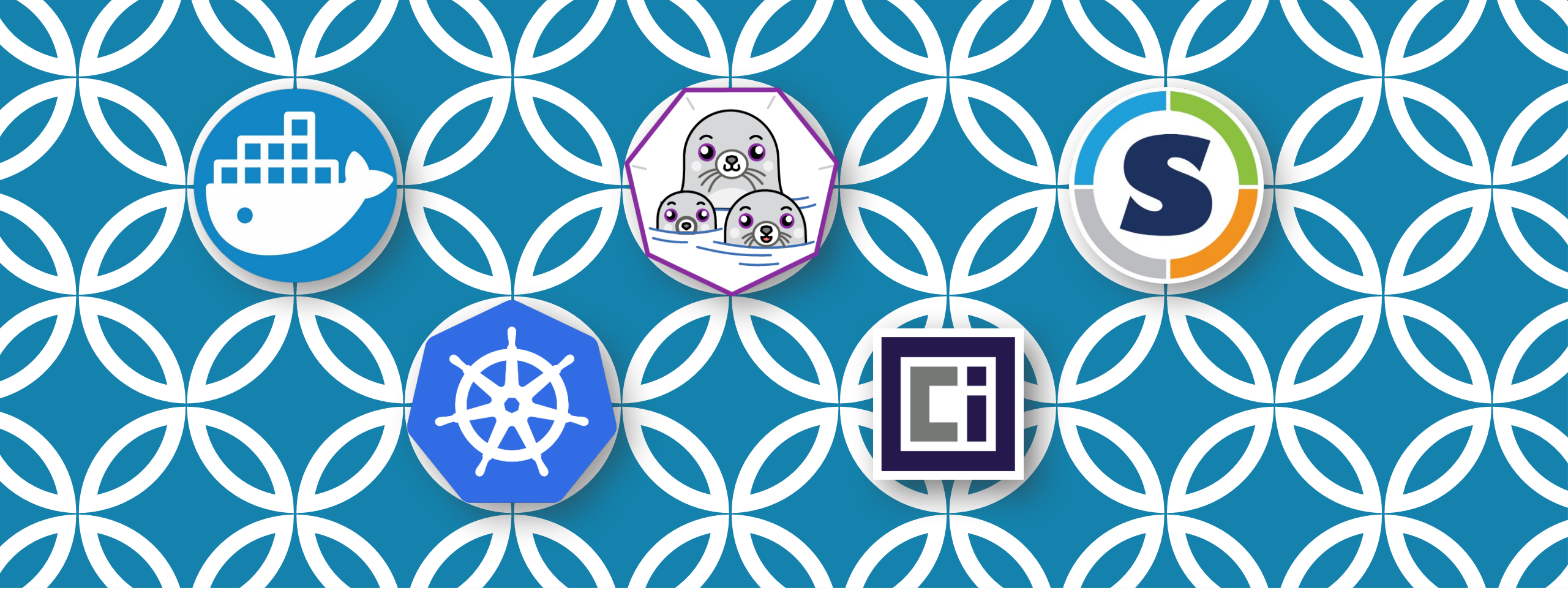


CONCEPTS DE CONTENEURISATION

BioInfoDiag – Séminaire 2025

QUI SUIS-JE

Rémi THEVENOUX

Développeur
Cellule Bioinformatique
(2023)

PLAN

Concepts généraux

- Problématique

Docker & Singularity

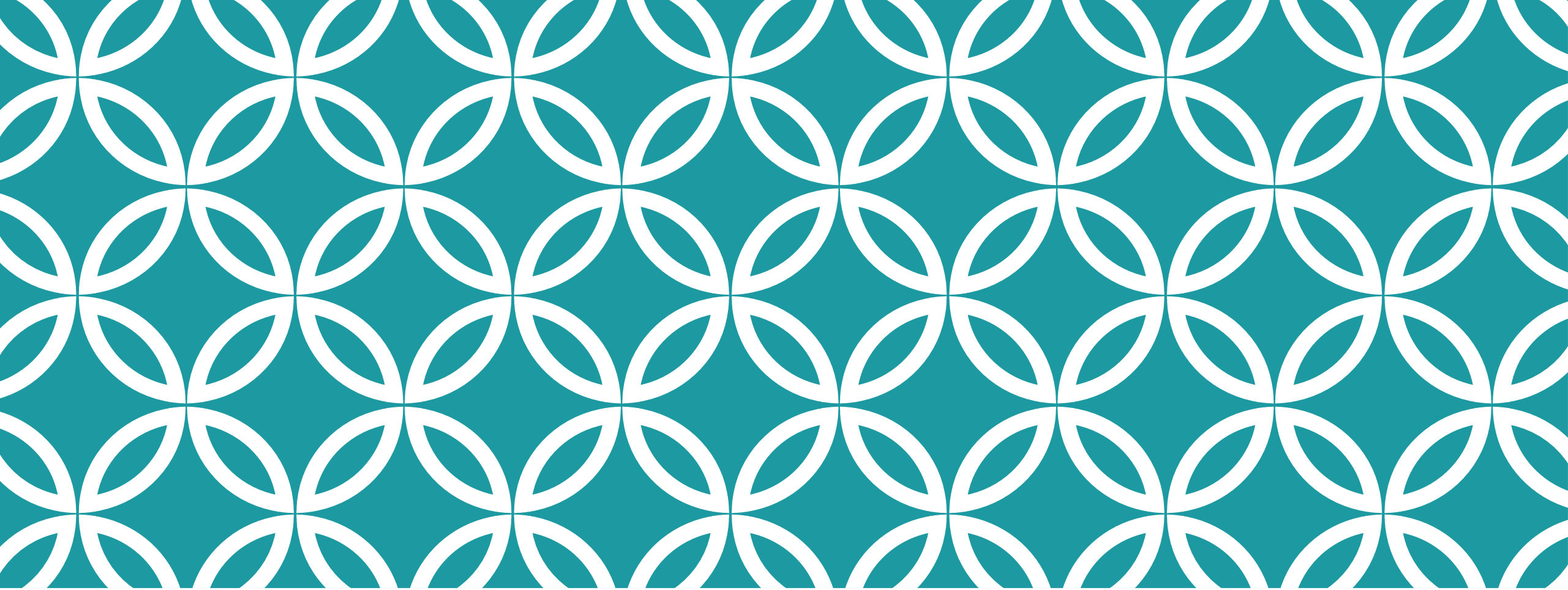
- Fonctionnement

Écosysteme & alternatives

- OCI, Podman, Kubernetes

Discussion

- Utilisation au CHU de Toulouse



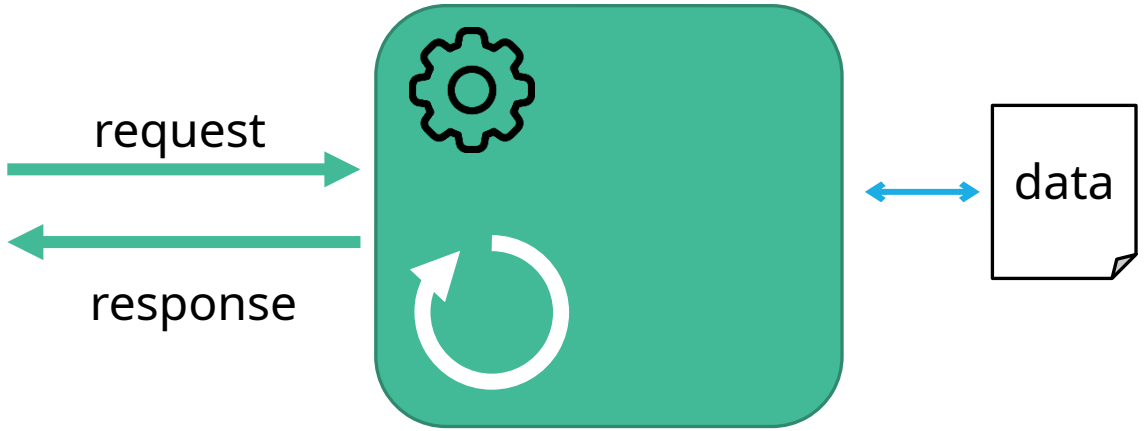
CONCEPTS GÉNÉRAUX

JOB VS SERVICE

Job/Task

ex: Workflow d’analyse

Service

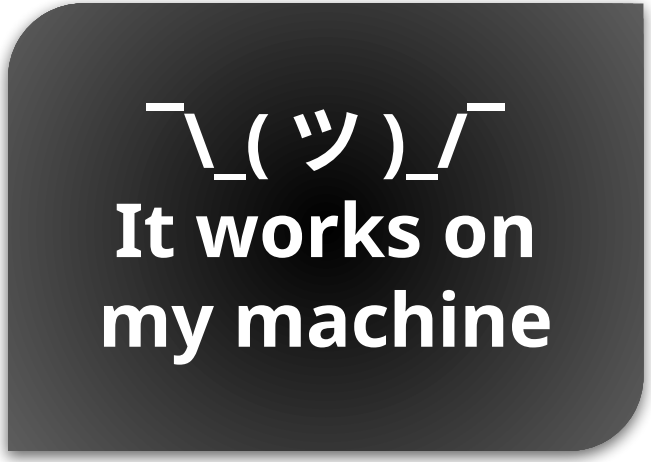


ex: Server web

PROBLÉMATIQUE

Reproductibilité / Portabilité

Conflits Dépendances



Autres aspects

- Isolation
- Uniformisation déploiements
- Scalabilité
- …

CONFLIT DE DÉPENDANCE

Exemple Python

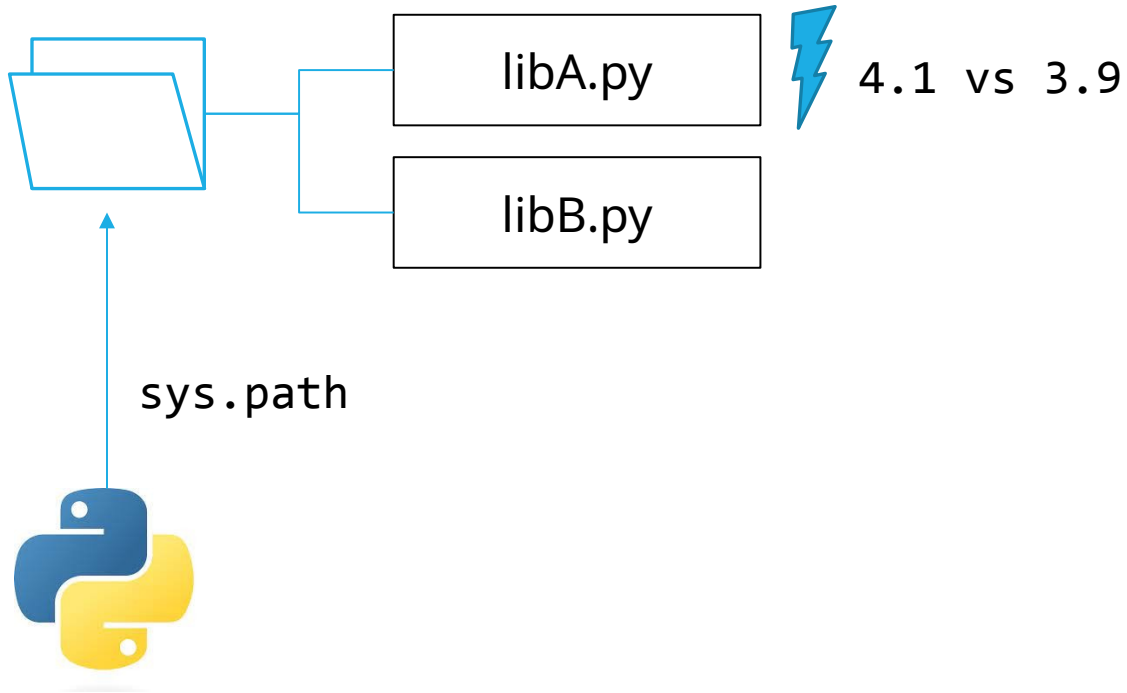
requirement.txt

app1.py

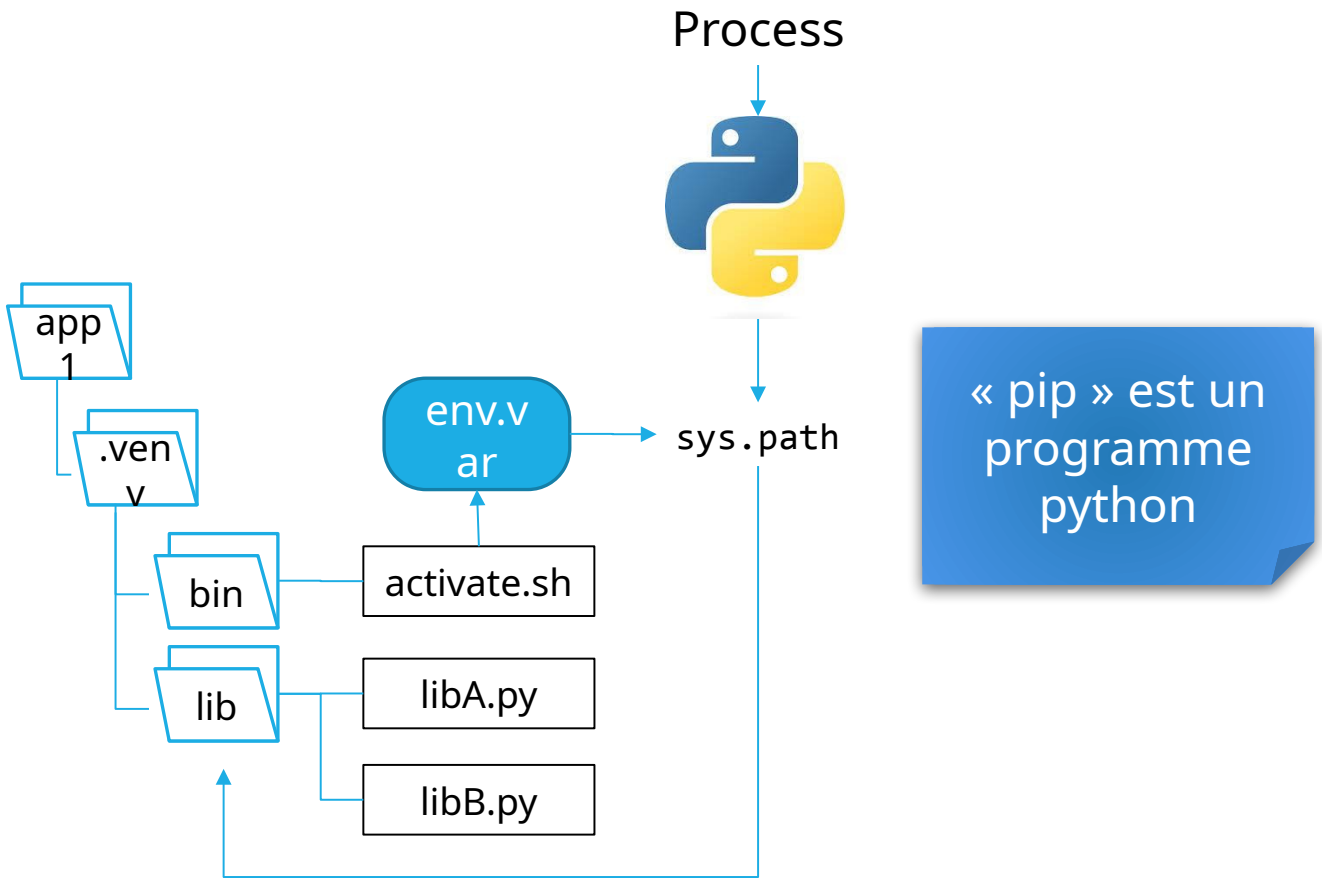
```
libA : 3.9
libB : 2.1
```

app2.py

```
libA : 4.1
```



PYTHON VENV + PIP



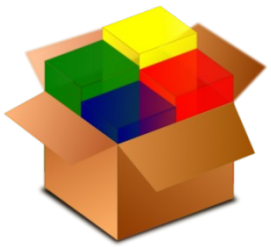
Limites

- Dépendances Python uniquement
- Ne permet pas de gérer la version de Python

CONDA

Gestionnaire environnement

create → créé un dossier
activate → modifie env.var. (notamment PATH)



Gestionnaire package

similaire à apt, yum,

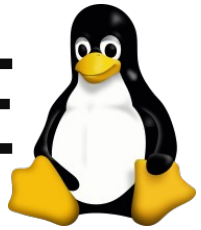
… Binaire
Python
…

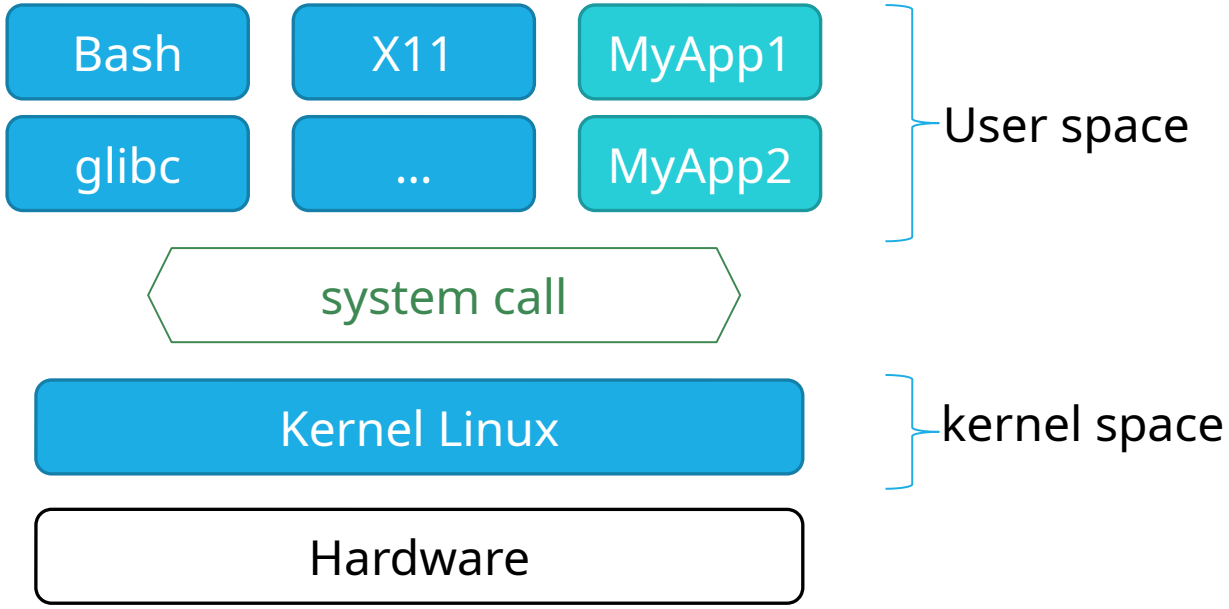
« Relocalisable »

Limites

- Pas de contrôle sur l'OS
  - glibc

USER SPACE / KERNEL SPACE

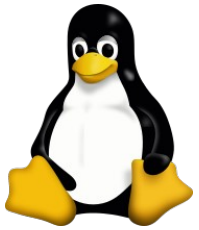


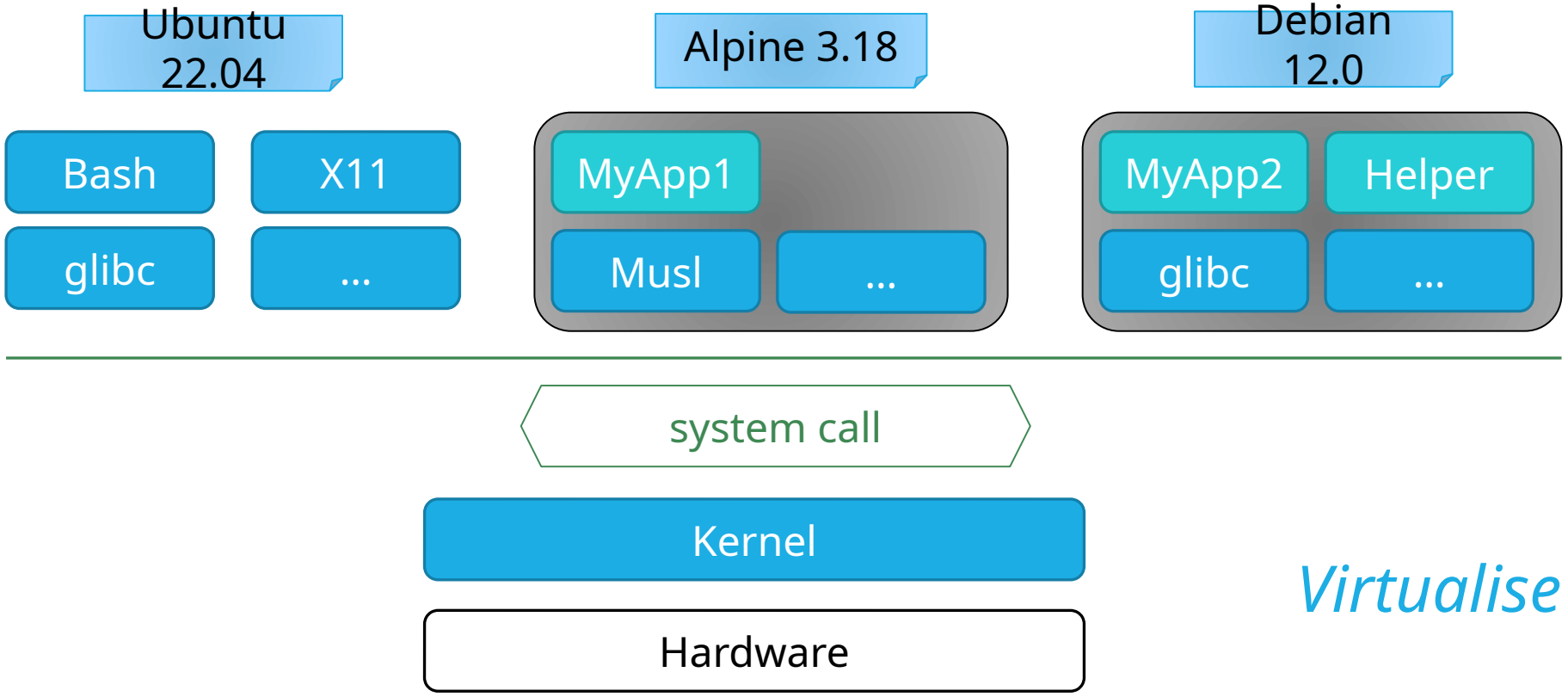


Linux distrubition

User install

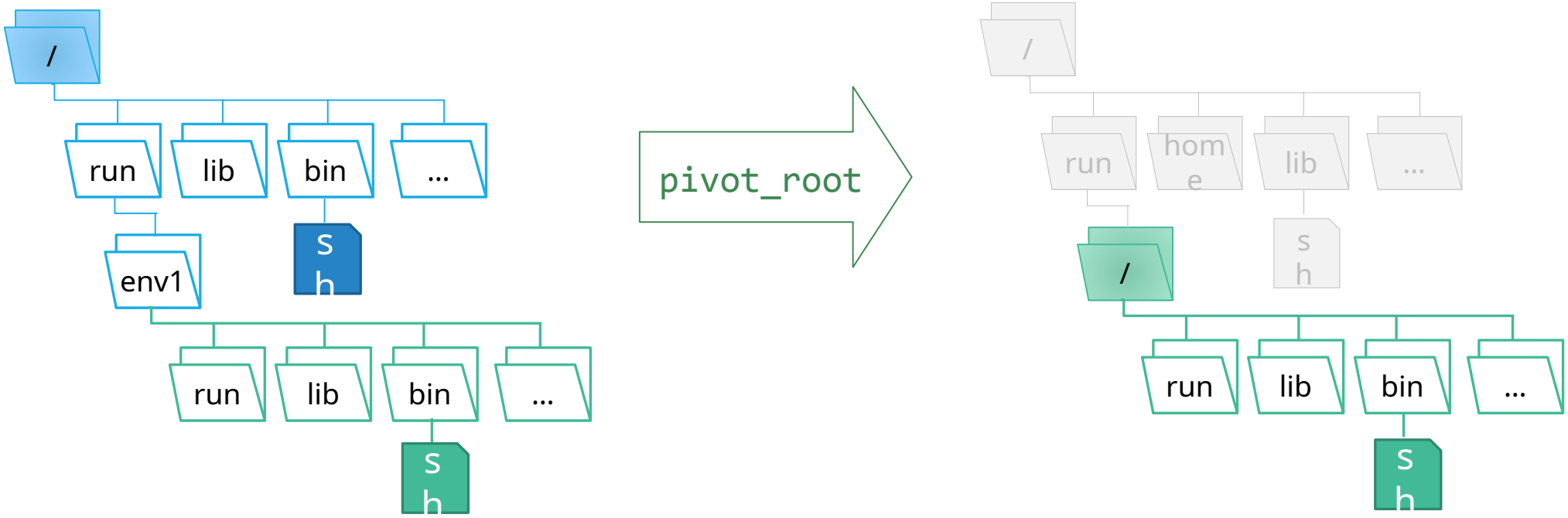
CONTAINER





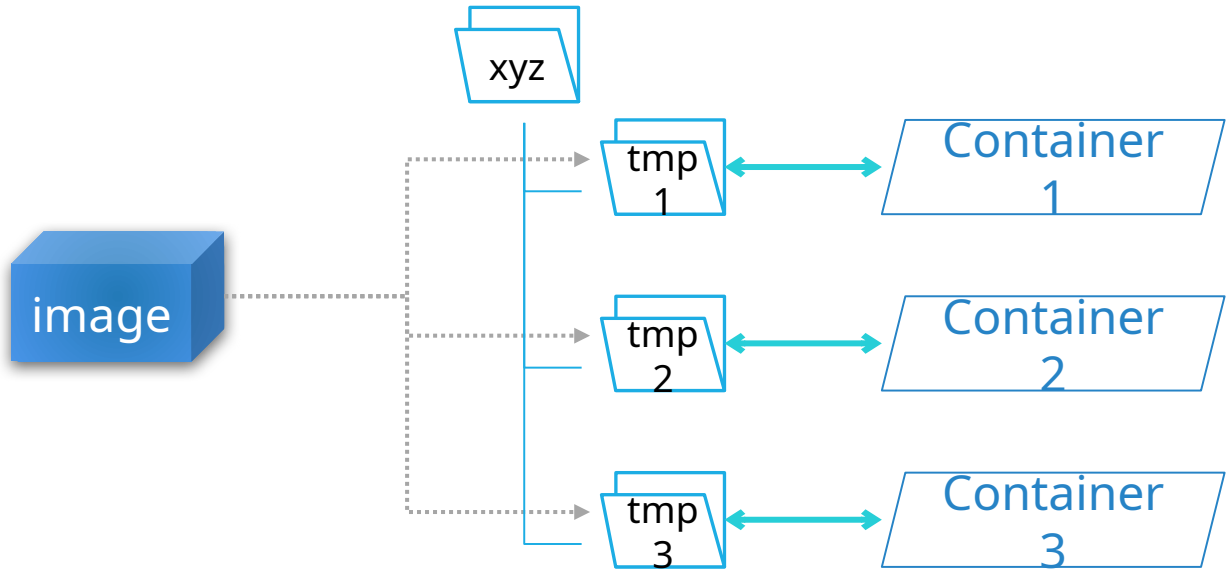
Virtualise le userspace

CHANGE ROOT



chroot unsecure https://tbhaxor.com/pivot-root-vs-chroot-for-containers/

CONTAINER

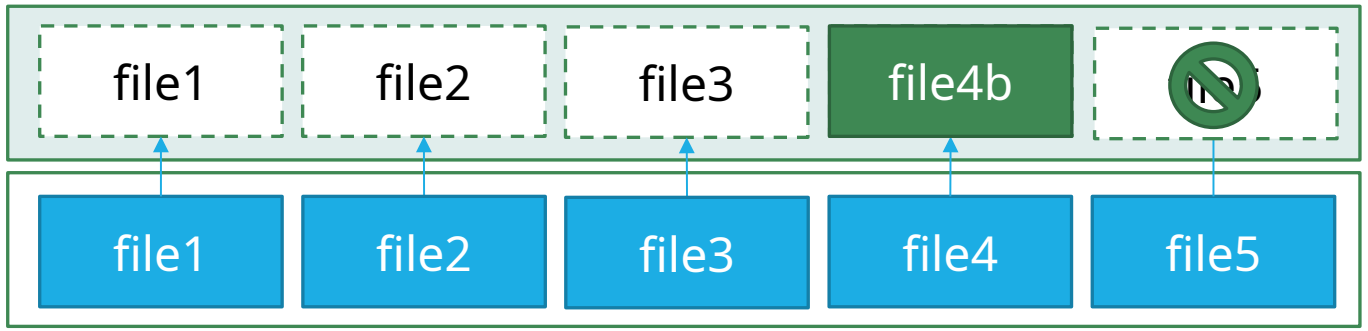


Cycle de vie

1. Création d'un répertoire
2. Change « root »
3. Exécute tâche
4. Supprime répertoire

→ *Éphémère*

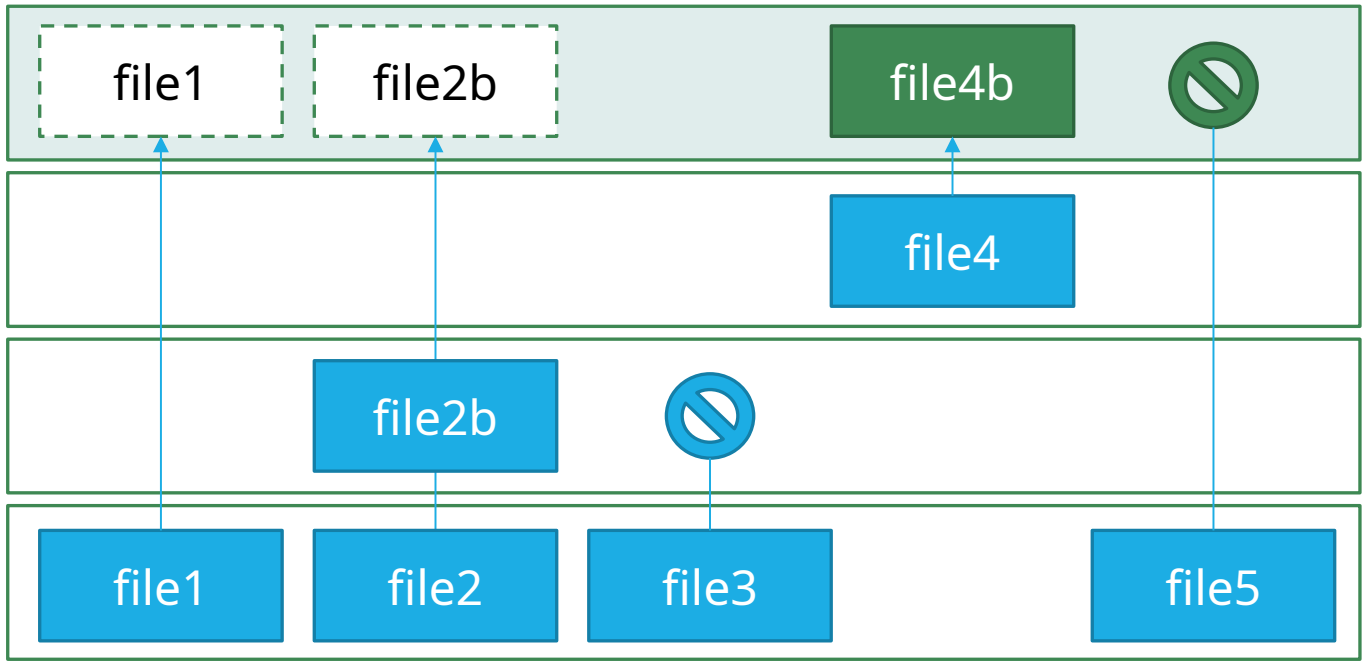
UNION MOUNT



Container layer – copy on write

Image layer - Read only

UNION MOUNT overlayfs2



Container layer – copy on write

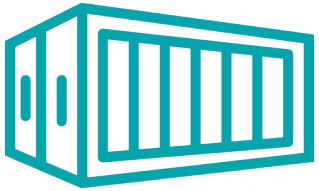
Layer 3

Layer 2

Layer 1

Image
Read
only

CONTAINER



Autres isolations

Namespace : *virtualise aspect 'kernel'*

- User
- PID
- Net
- ...

Cgroup : limitation des ressources

- CPU
- RAM

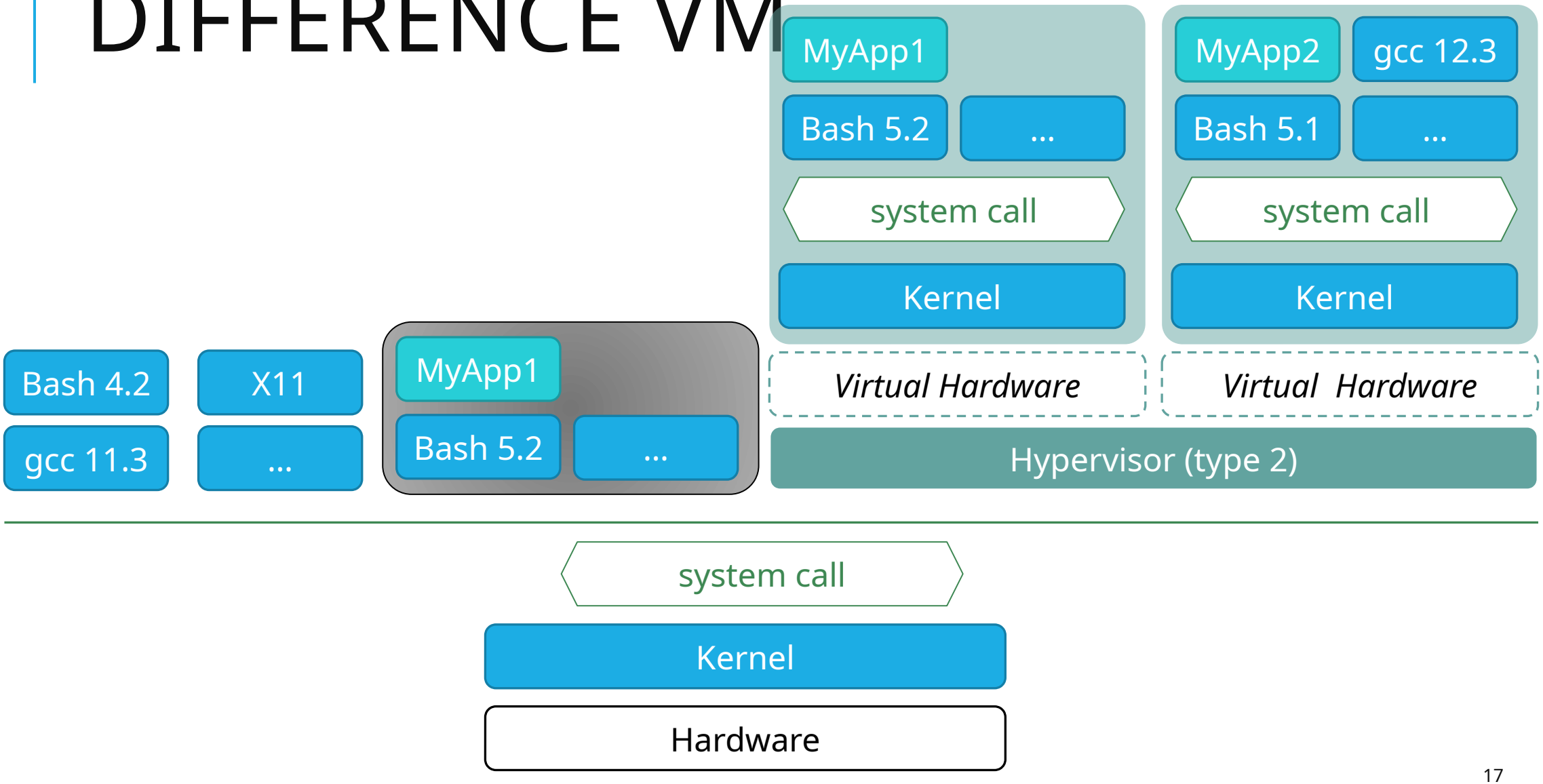
Différentes définitions selon techno

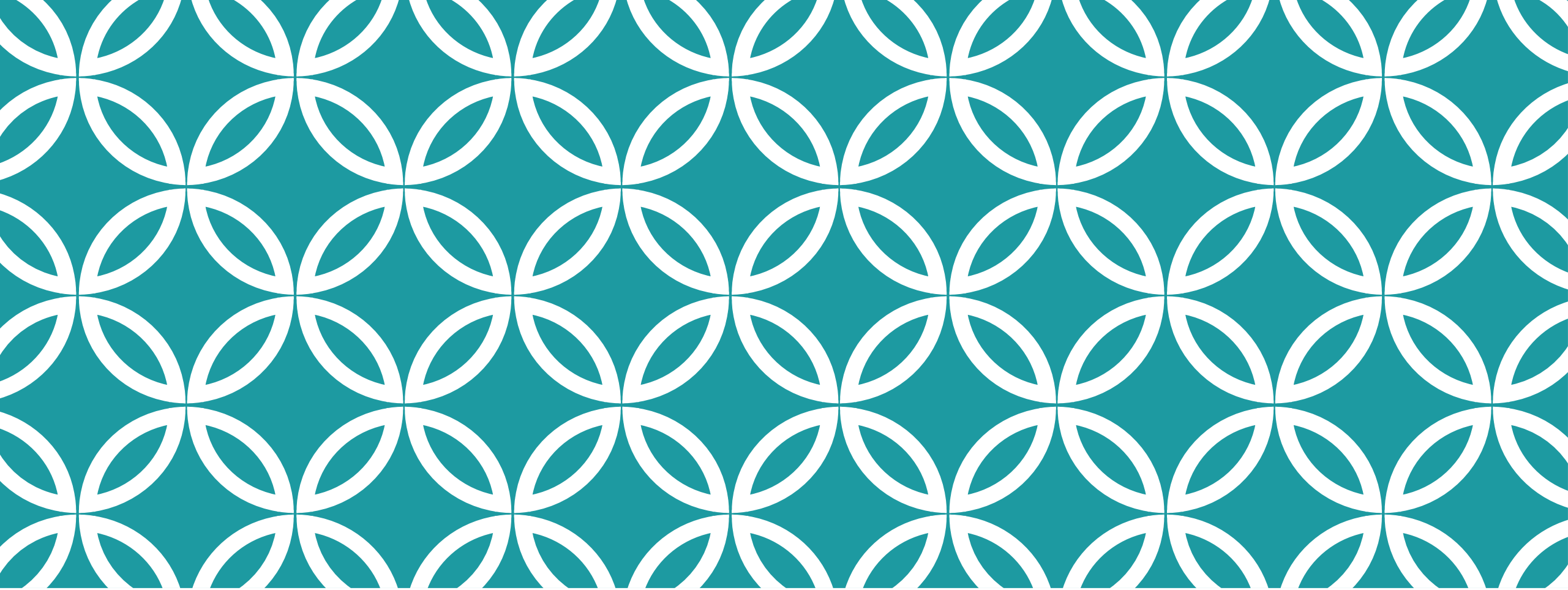
	Docker	LXC	Singularity
Ephémère	Oui	Non	Oui
Isolation user	Oui	Oui	Non

Etc.

https://www.linuxembedded.fr/2020/11/namespaces-la-brique-de-base-des-conteneurs

DIFFÉRENCE VM

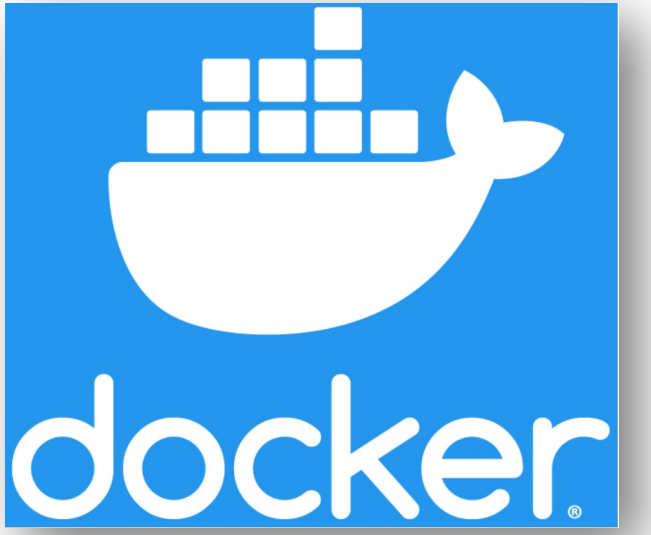




DOCKER & SINGULARITY

DOCKER

Première solution « clé en main » de conteneur (2013)



Ne pas confondre :

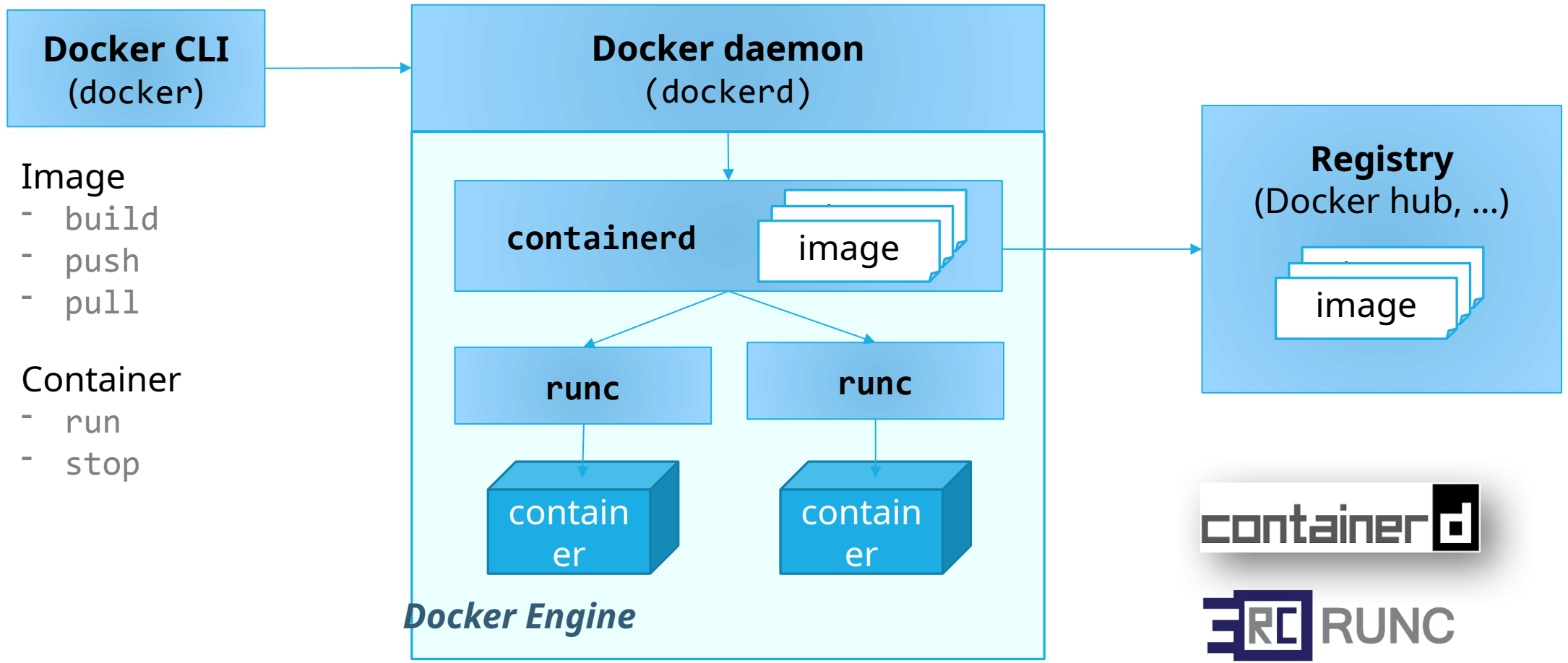
- La compagnie Docker Inc.
- L'outil `docker` (gratuit)
- Docker Desktop (payant si >250 employés)
- Le format d'image
- Le dépôt d'image DockerHub

DOCKER ARCHITECTURE

Principales commandes

Image
- `build`
- `push`
- `pull`

Container
- `run`
- `stop`



DOCKERFILE & LAYERS

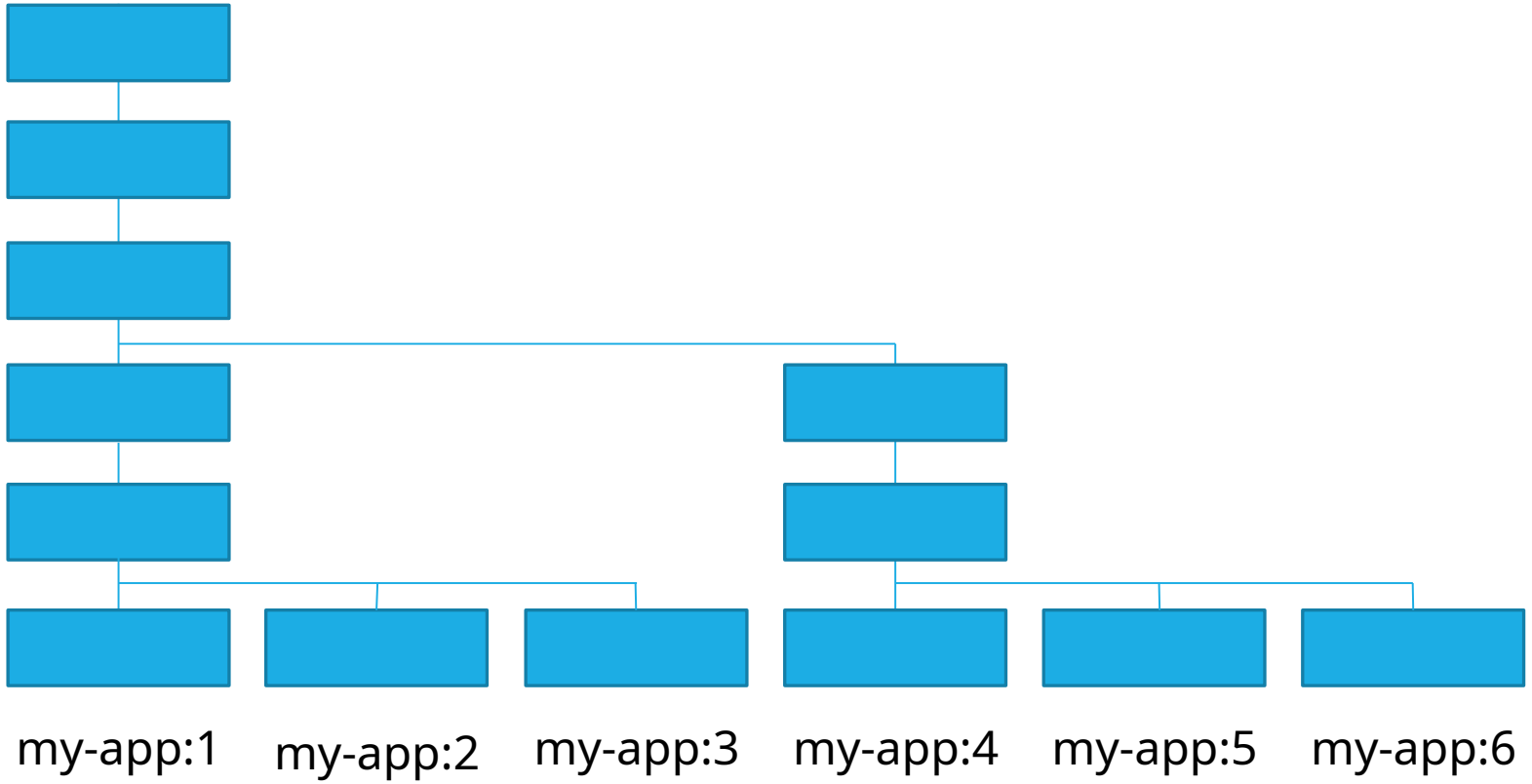
DockerFile

```
FROM python:3.10

COPY requirement.txt

RUN pip install

COPY *.py
```



ÉVOLUTION

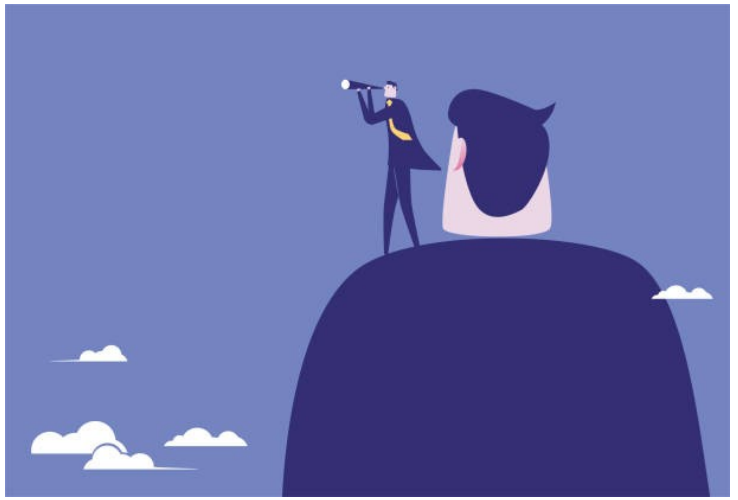
chroot

BSD jail, Solaris Zone, Linux vServer, OpenVZ

LXC

Docker

Singularity, Podman, …



Écosystème foisonnant

SINGULARITY

Technologie de container (2015)

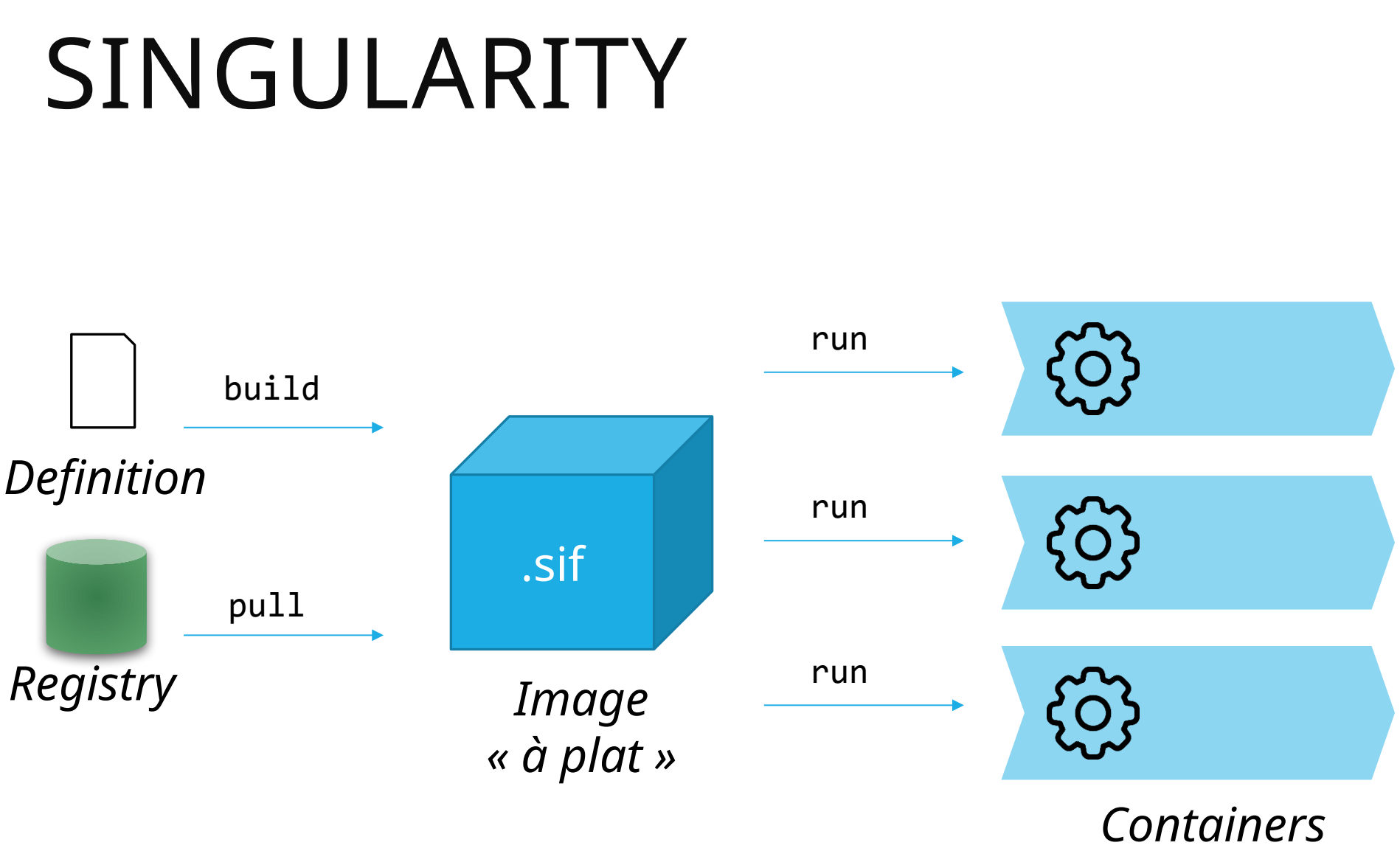
Développé pour les HPC

Support commercia Sylabs

Daemonless

Rootless

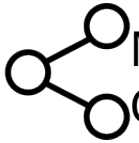
SINGULARITY



IMAGES

Docker

« Repo » local
Système de tag intégré



Mutualisation des couches communes
Cache lors du build

Docker: Accès daemon ↔️ accès toutes images
Podman: Par utilisateur

Fall-back : `load/save` or `image scp`

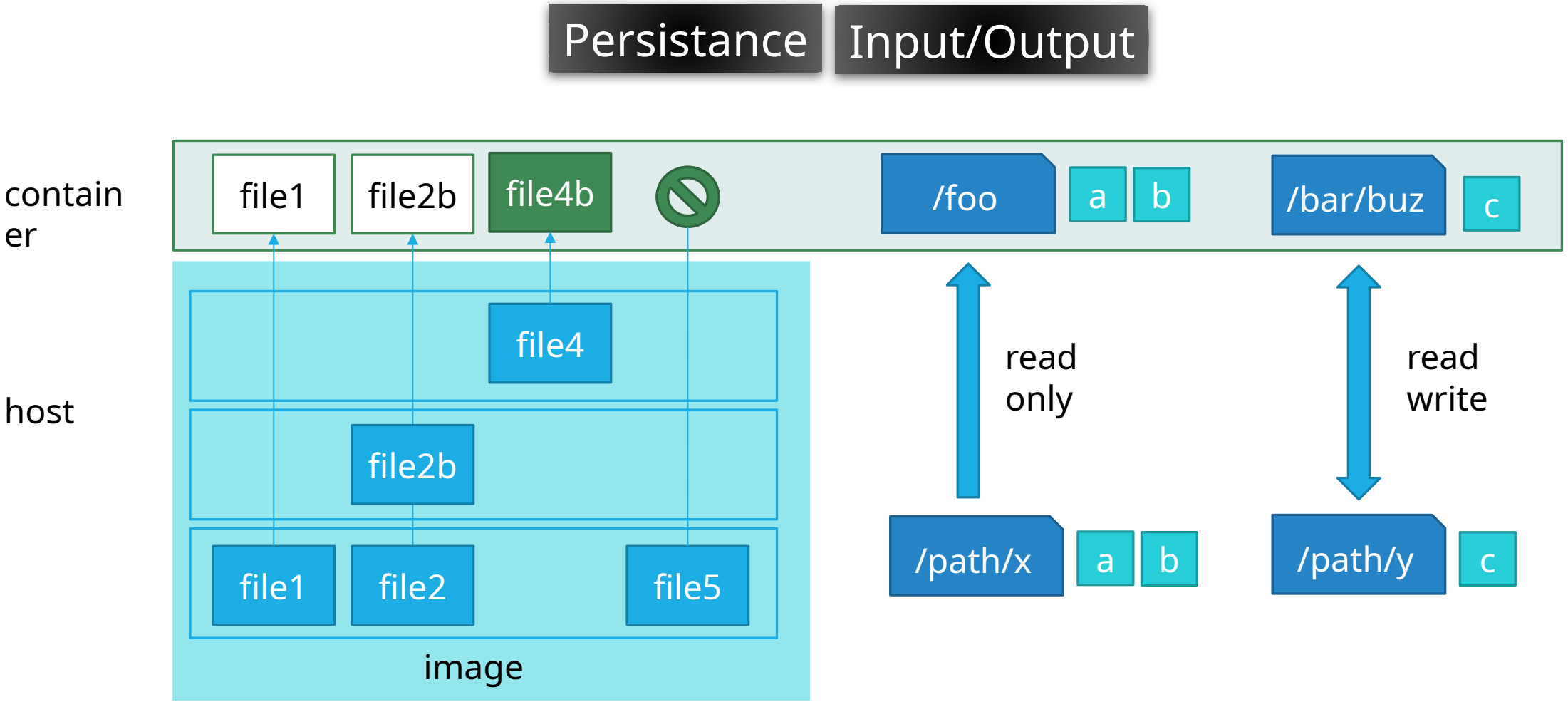
Singularity

Simple fichier : transfert via `cp`

Accès à l'image ↔️ permission sur le fichier

Cryptage

VOLUME


Persistance
Input/Output
container
host
file1
file2b
file4b
/foo
a
b
/bar/buz
c
file4
file2b
file1
file2
file5
image
read only
read write
/path/x
a
b
/path/y
c

VOLUME

Singularity

Default mount

`$HOME`	`$PWD`
`/tmp`	`/proc`
`/sys`	`/dev`

```
--bind /host/path:/container/path
```

Docker

Bind mount

- Host ↔️ Container

```
-v /host/path:/container/path
```

Volume

- Persistance sans interaction avec host

```
-v db-data:/container/db/data
```

USER

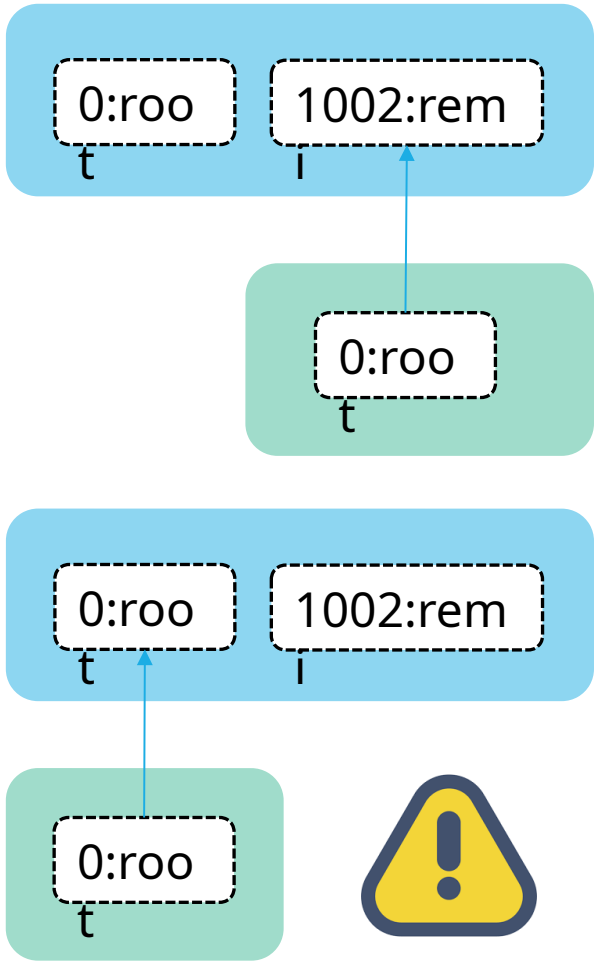
Singularity

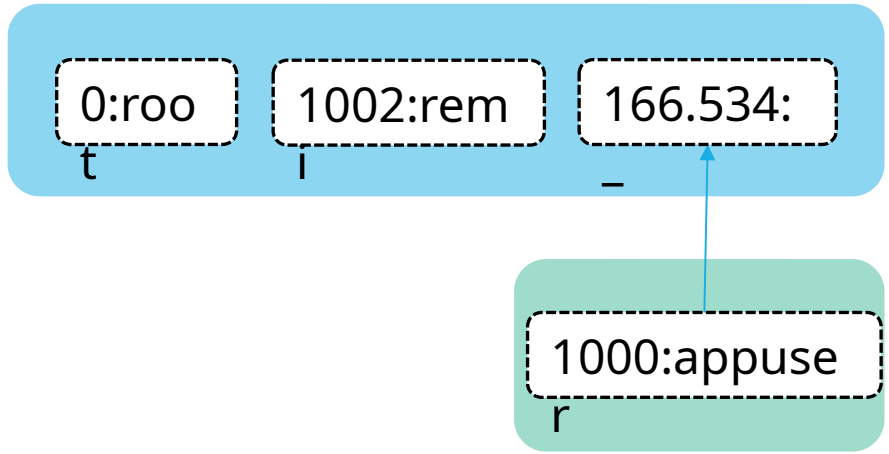
Host User

=

Container User

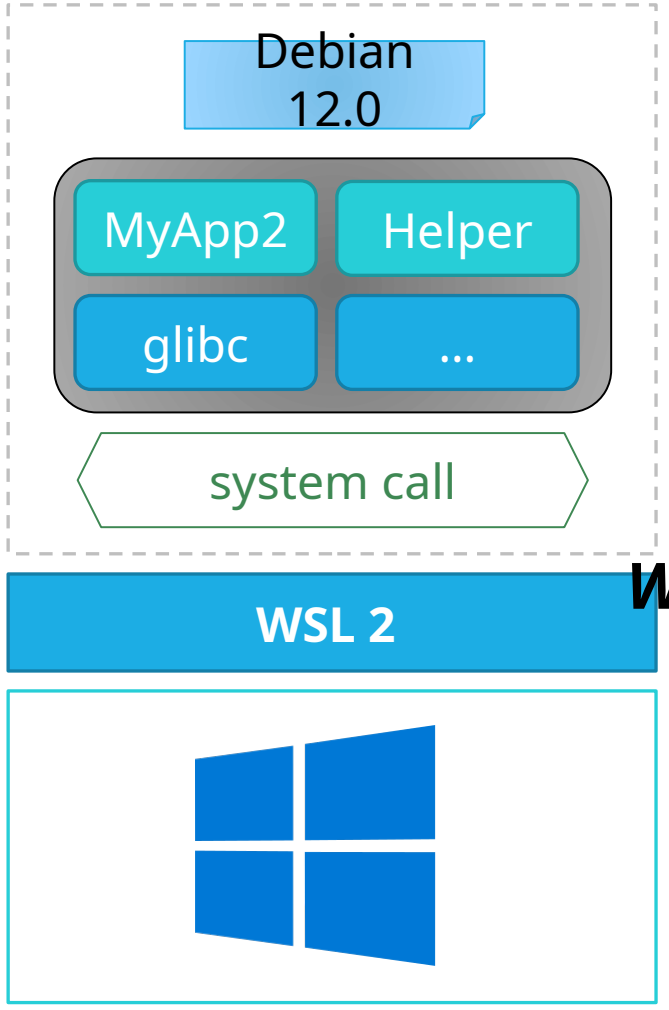
Docker « It’s complicated »





Daemon rootless ?
Configuration daemon
Option --user, --userns
Image instruction USER

WINDOWS / LINUX



Docker

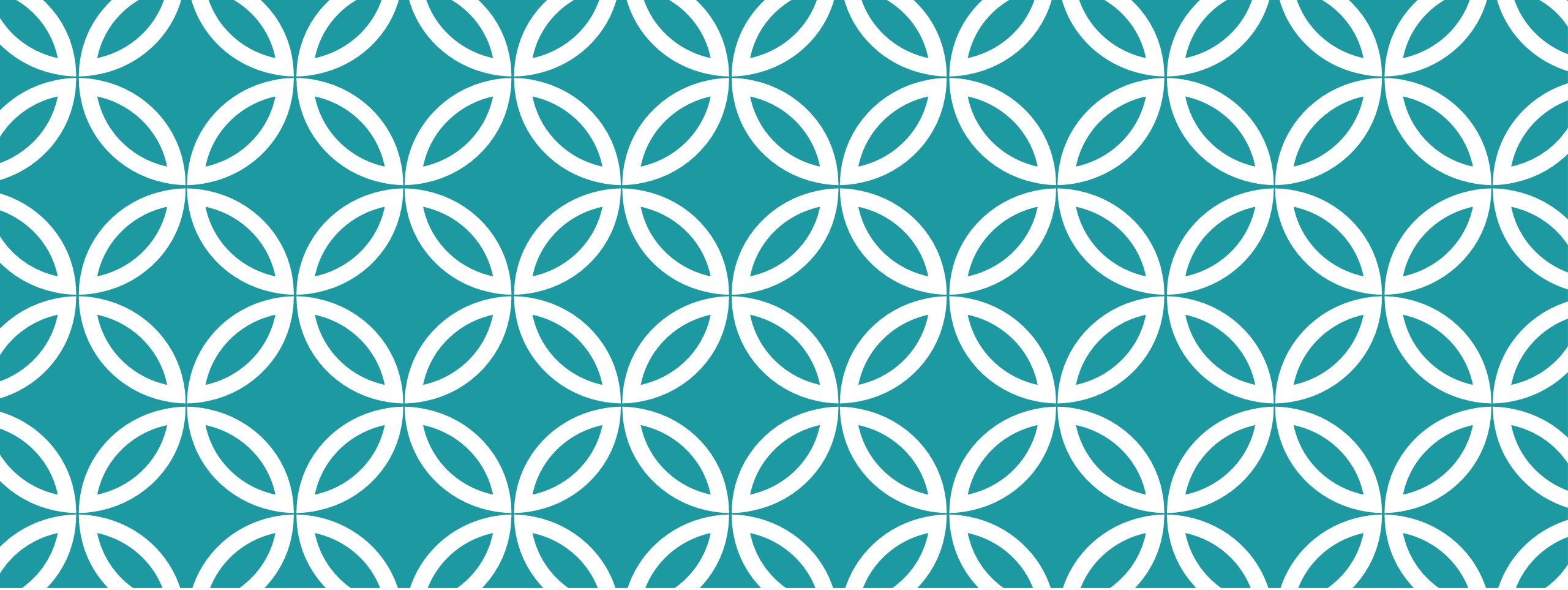
Compatible

Singularity

Non compatible
→ Utilisation d'une VM

Windows Subsystem for Linux

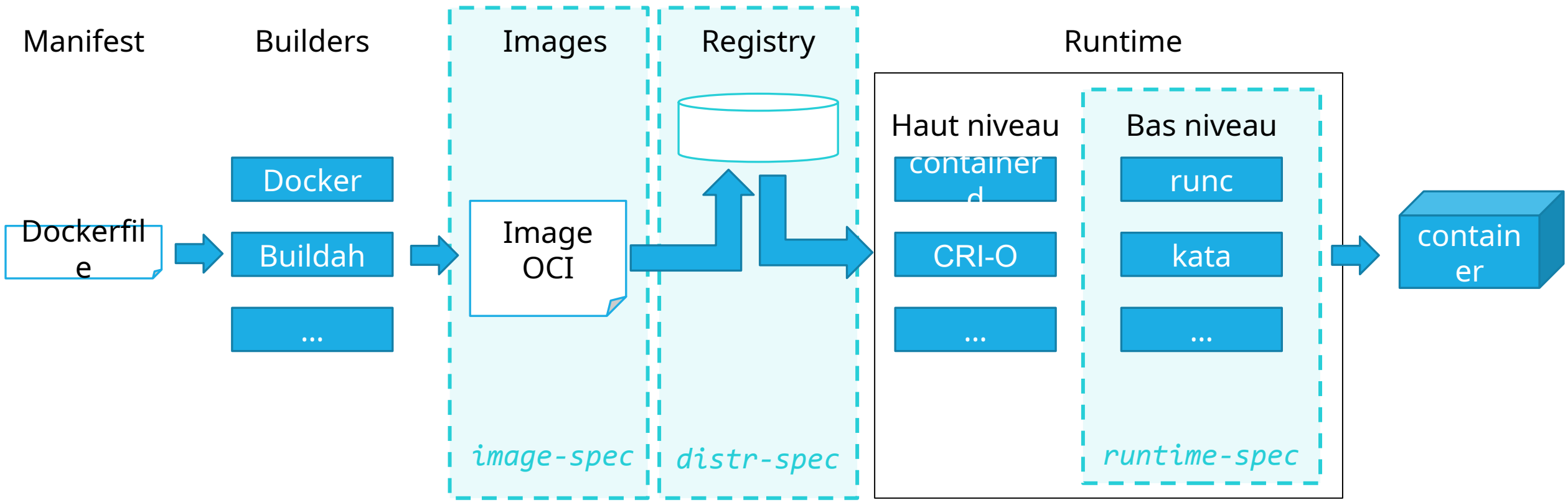
Linux on Windows



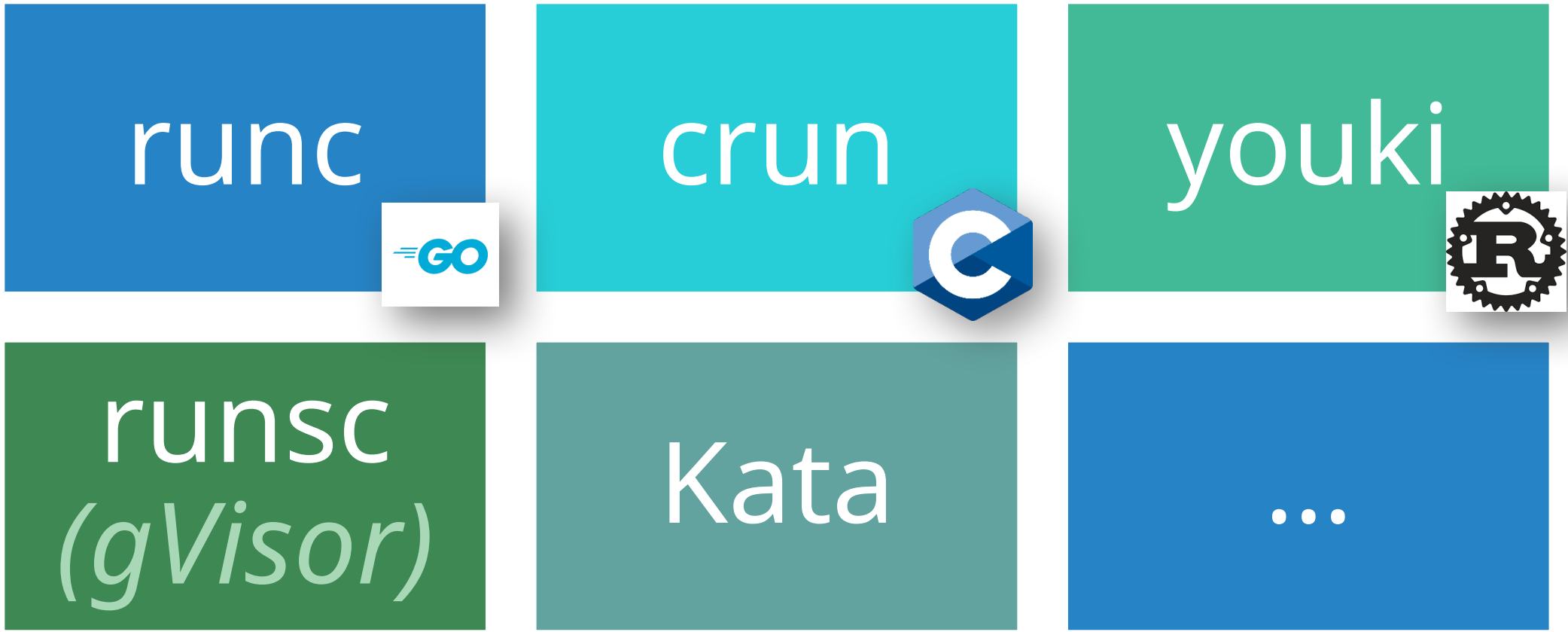
ÉCOSYSTEME & ALTERNATIVES

FORMAT STANDARD

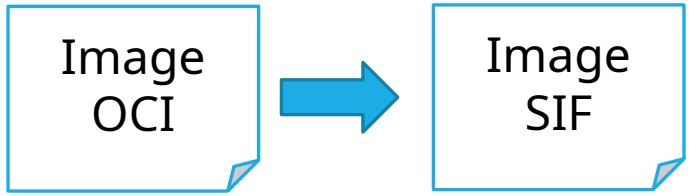
OPEN CONTAINER INITIATIVE



ALTERNATIVE RUNTIME



SINGULARITY : HPC



Alternatives

Apptainer — Fork open-source Singularity

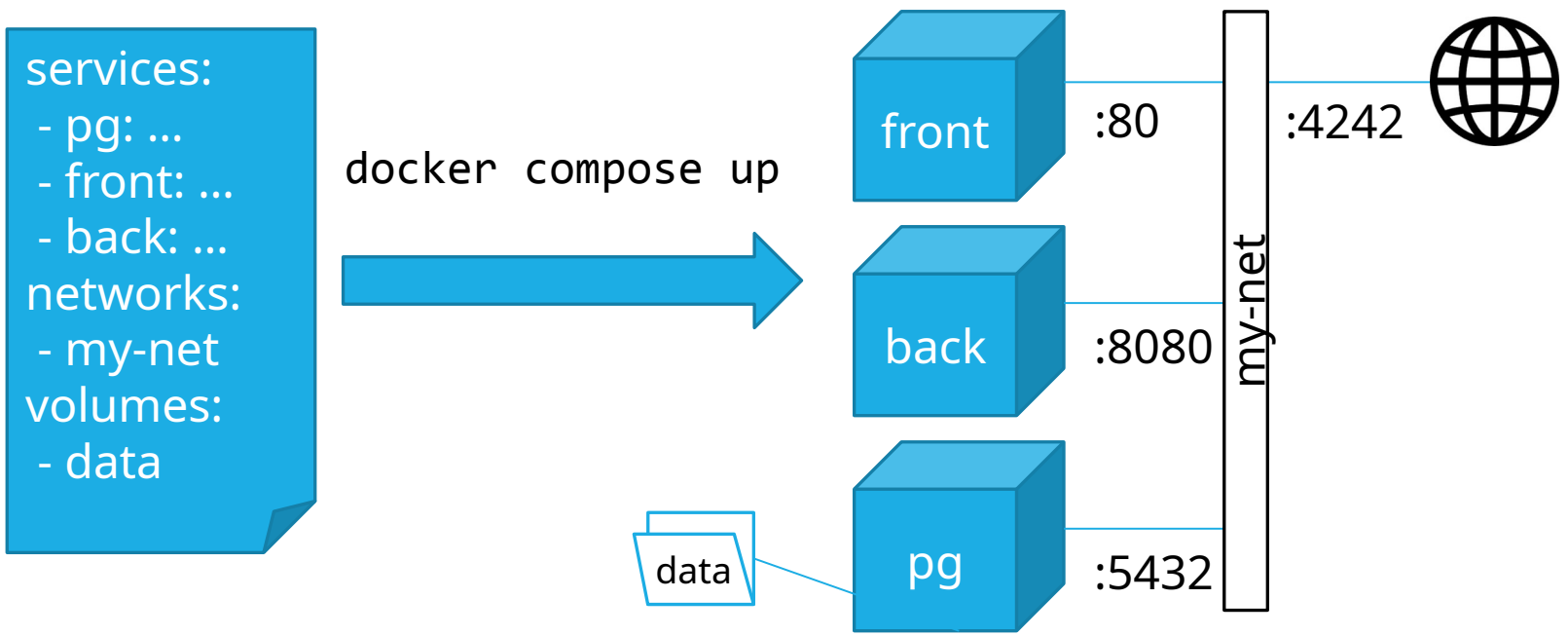
Charliecloud — Los Alamos National Laboratory

Sarus — CSCS - Centre suisse de calcul scientifique

Shifter — last commit: 3 years

Enroot — NVIDIA

DOCKER COMPOSE



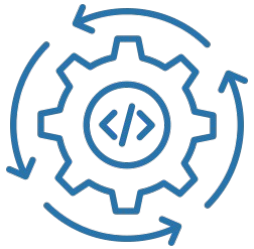
- Coordonner plusieurs
- Up/down 1 commande
- Plus explicites que des lignes commande

docker-compose.yml

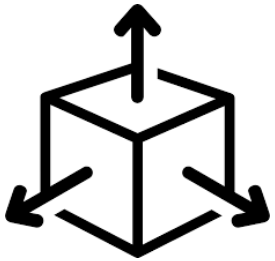
Singularity Compose : projet peu actif

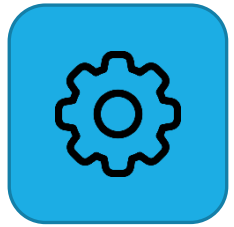
CLUSTER & ORCHESTRATION

CI/CD, Agile

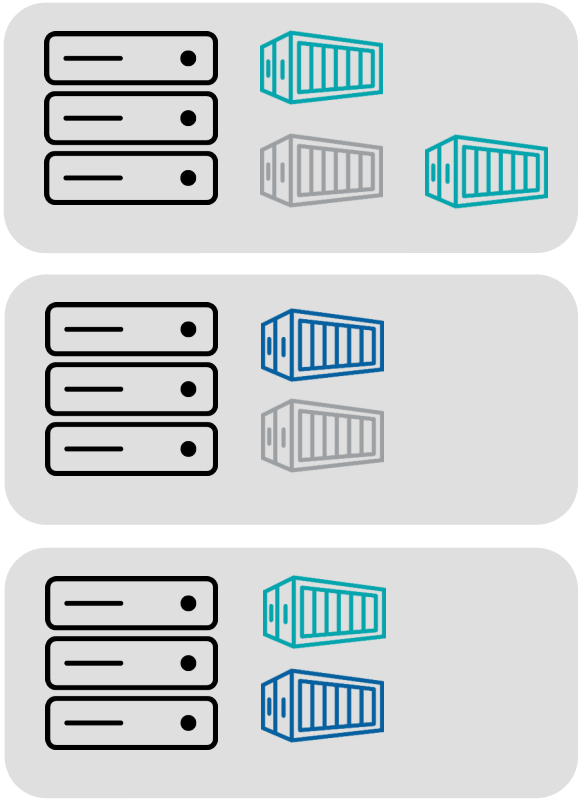


Microservice, scalabilité

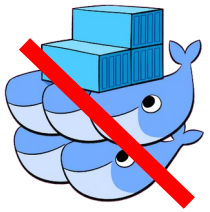




Provisioning
Monitoring
Scalling

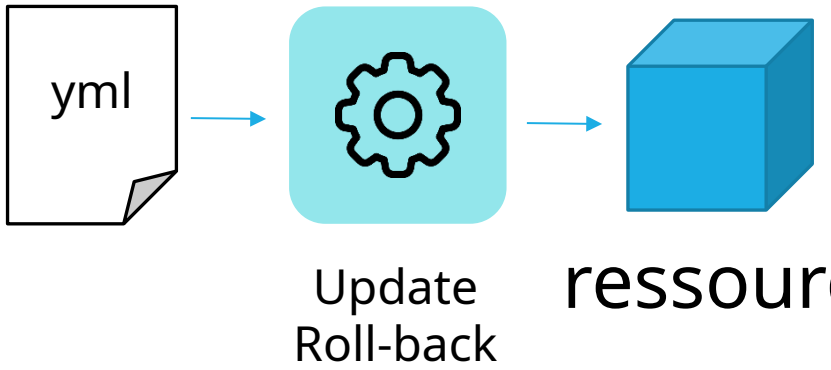


kubernetes



Docker Swarm

KUBERNETES



Container
Base de donnée
Certificats TLS
Ingress/route
…

Écosystème très riche

Single-node

minikube

K3S

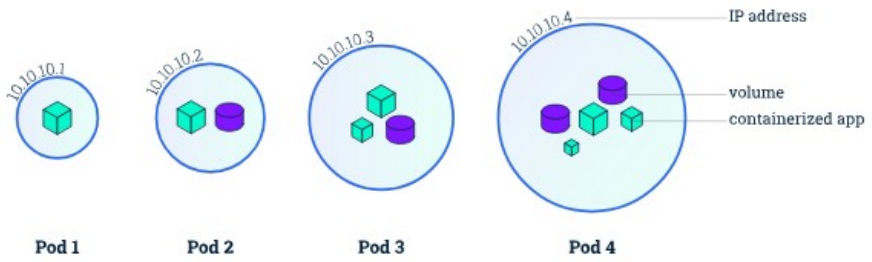
MicroK8s

PODMAN / KUBERNETES

kubernetes

RED HAT® OPENSHIFT

Brique de base de : Pod



podman

Daemonless

Rootless

- Tester pod
- Alternative docker-compose
- Mature pour production

COMPATIBILITÉ AVEC DOCK

```
docker
build .
docker ps –a
docker run
```

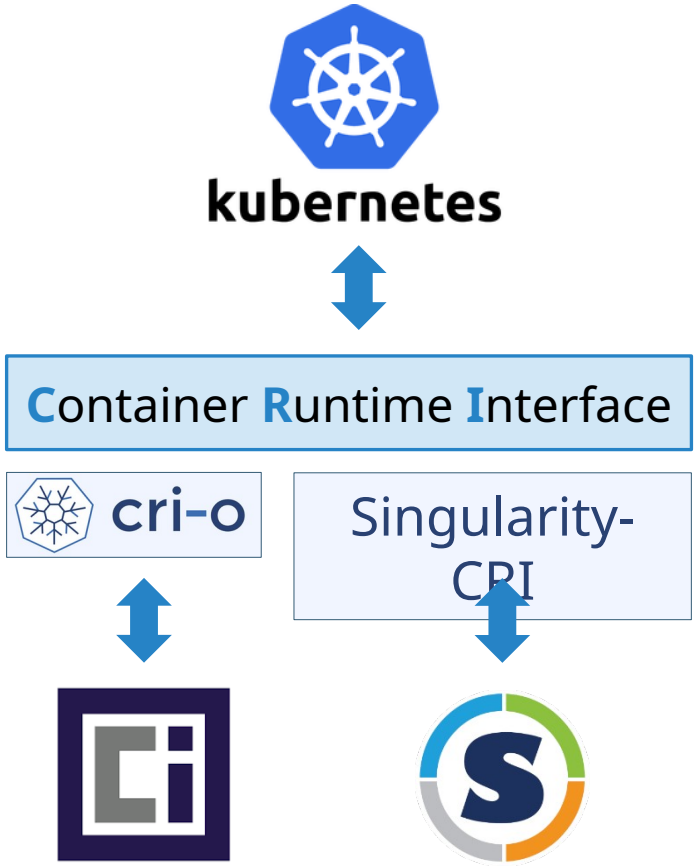
```
podman
build .
podman ps –a
podman run
```

Certaines options incompatibles
Notamment : *network*

Pas de compatibilité avec *Docker Compose*

Le projet à part « `podman-compose` » permet une compatibilité

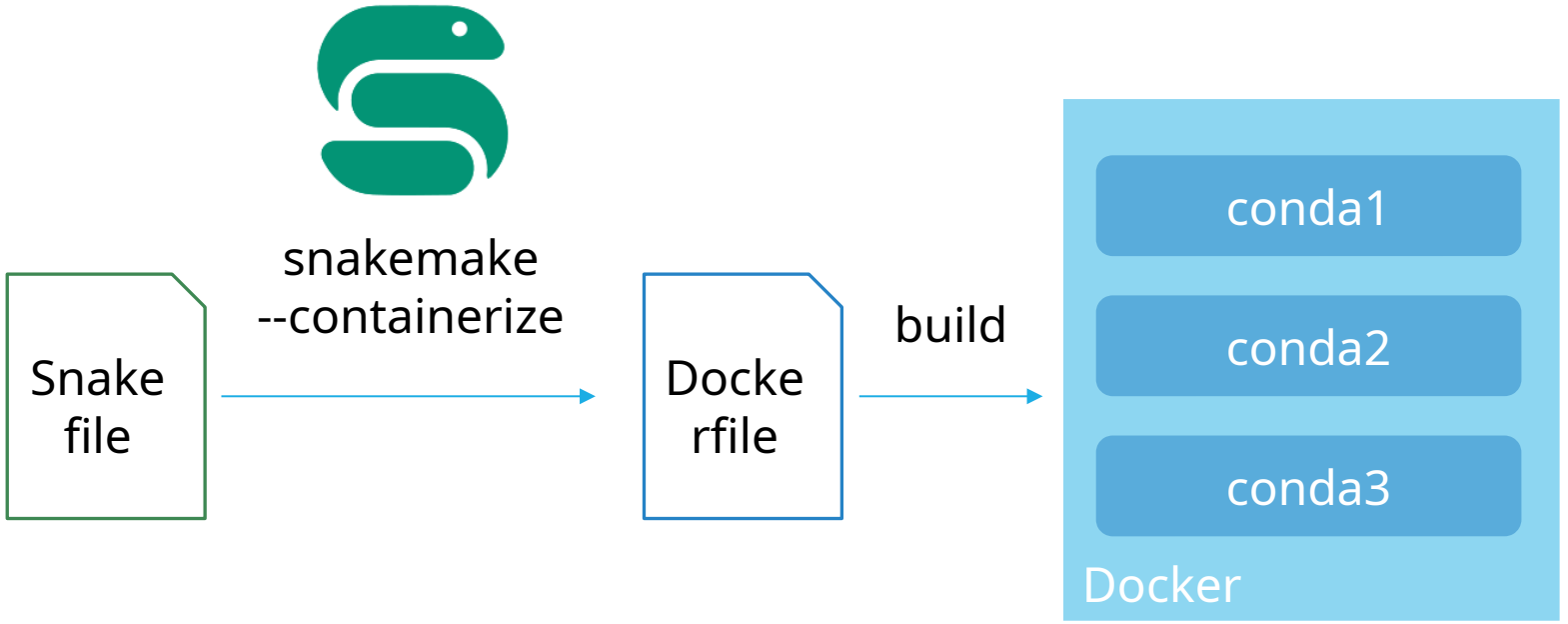
CLOUD NATIVE COMPUTING FONDATION



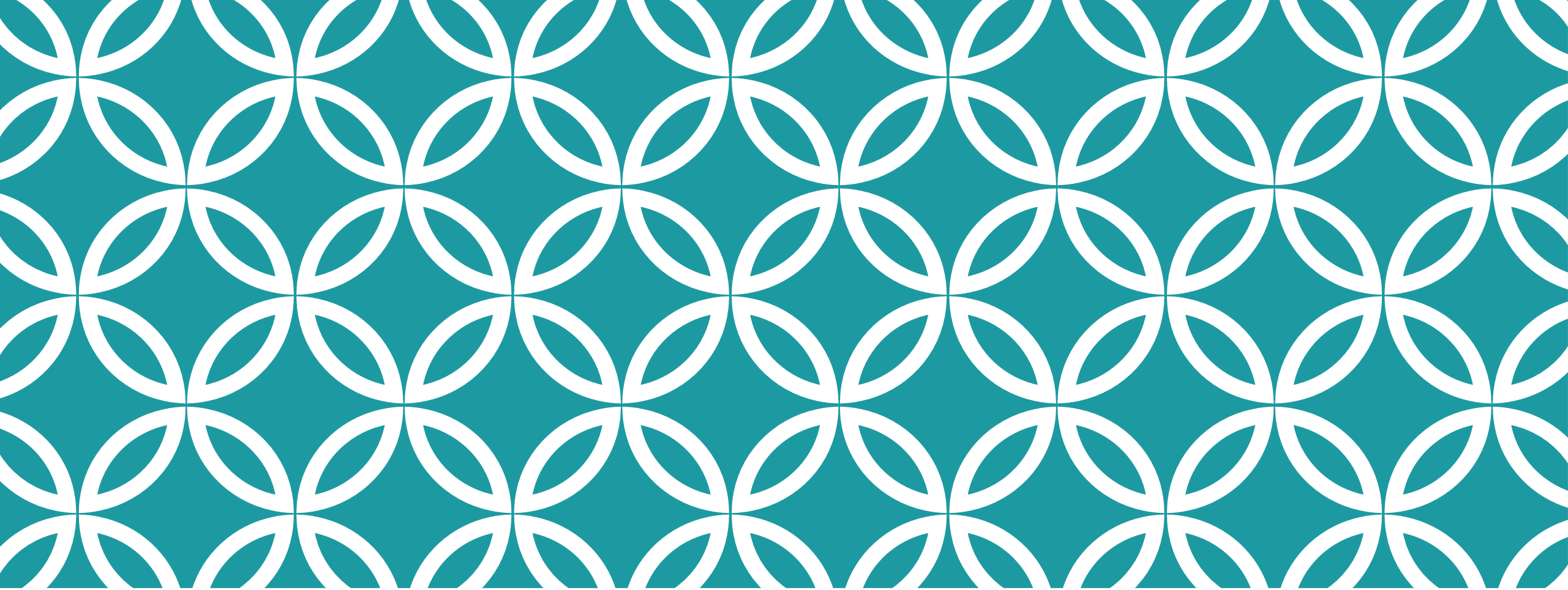
CLOUD NATIVE
COMPUTING FOUNDATION

https://landscape.cncf.io

DOCKER + CONDA



https://snakemake.readthedocs.io/en/stable/snakefiles/deployment.html#containerization-of-conda-based-workflows



DISCUSSION

AU CHU DE TOULOUSE ?

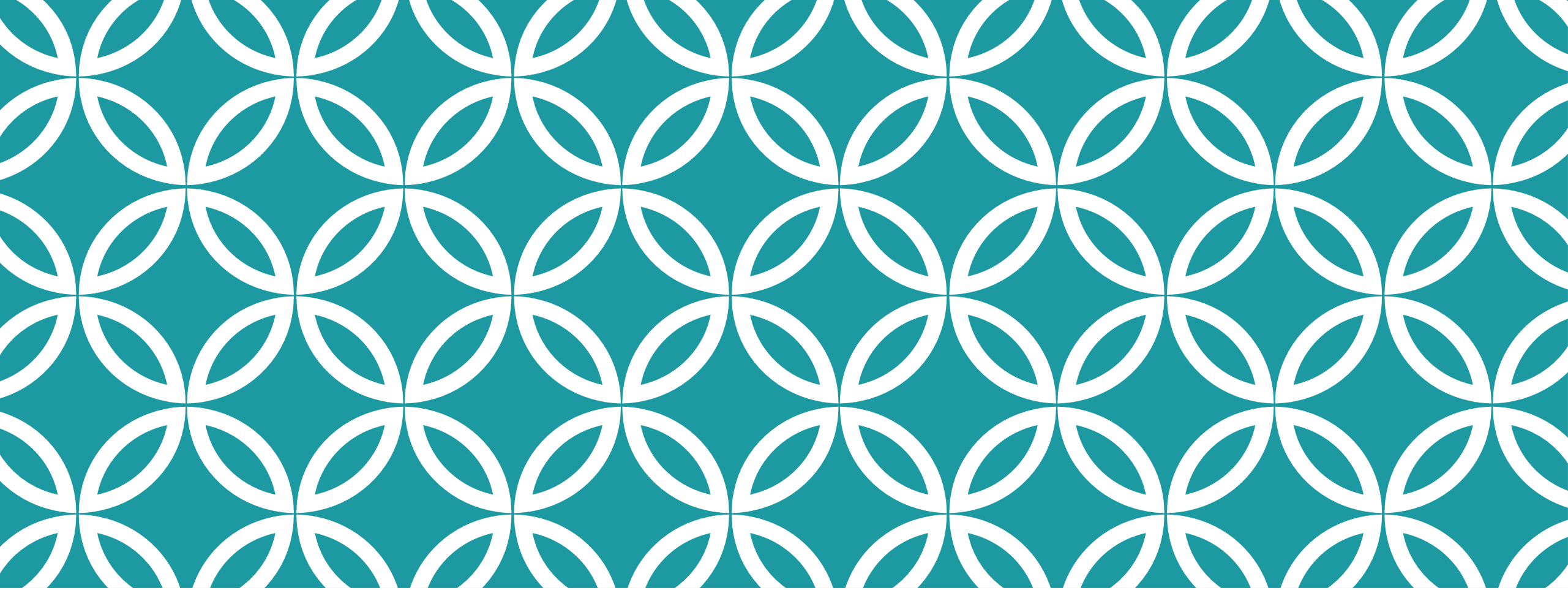
Worflow
- « Packaging »

ABiWoL *(Fréderic Escudié)*
- Automatisation lancement workflow, archivage, …

Vidjil *(Inria)*
- Analyse répertoires lymphocytaires

SaMa *(Fréderic Escudié & Rémi Thevenoux)*
- Sample-sheet Manager

LIS *(Fréderic Escudié & Rémi Thevenoux)*
- Stockage, visualisation et curation de résultats NGS



MERCI