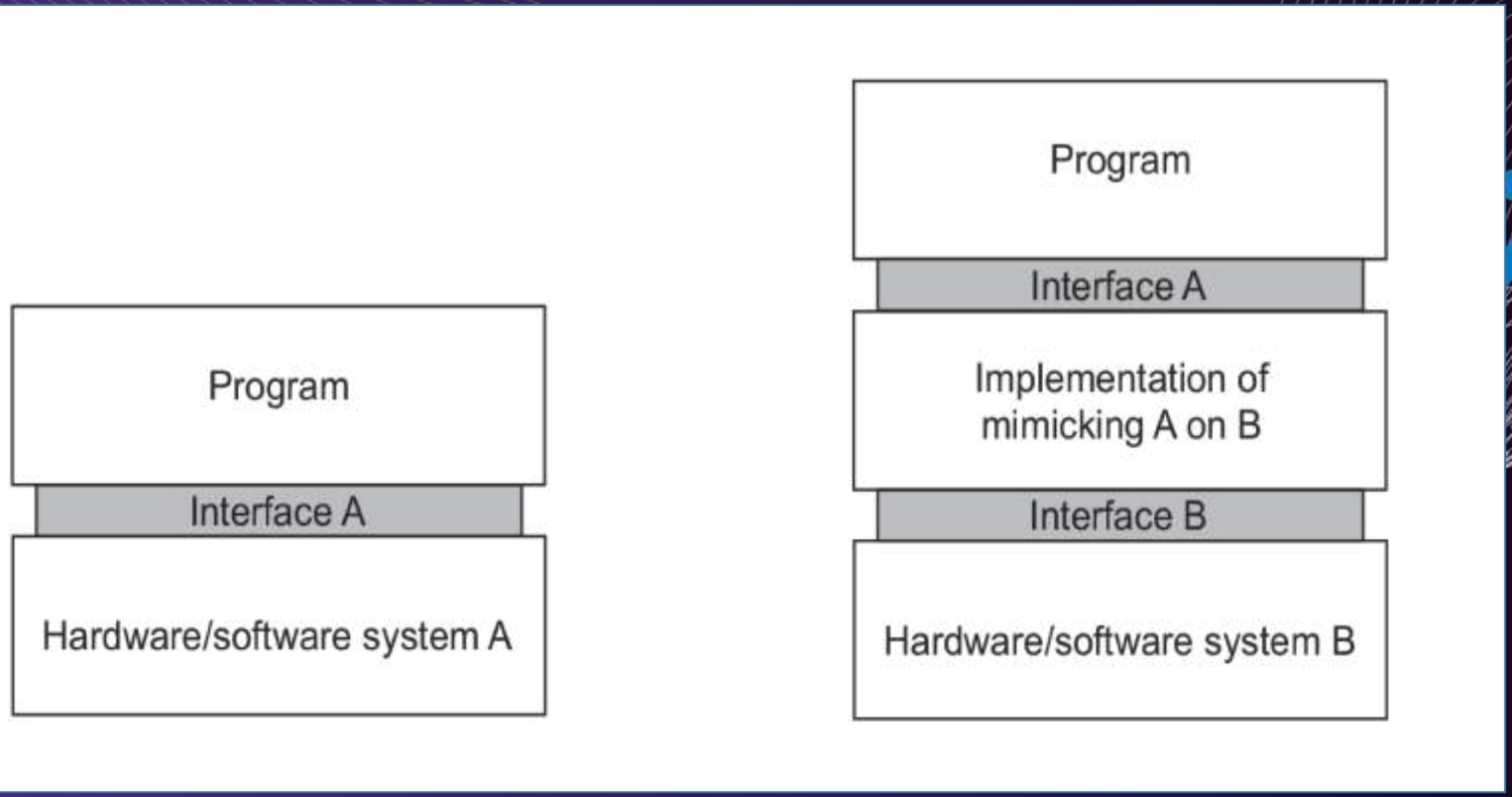
VIRTUALIZATION

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

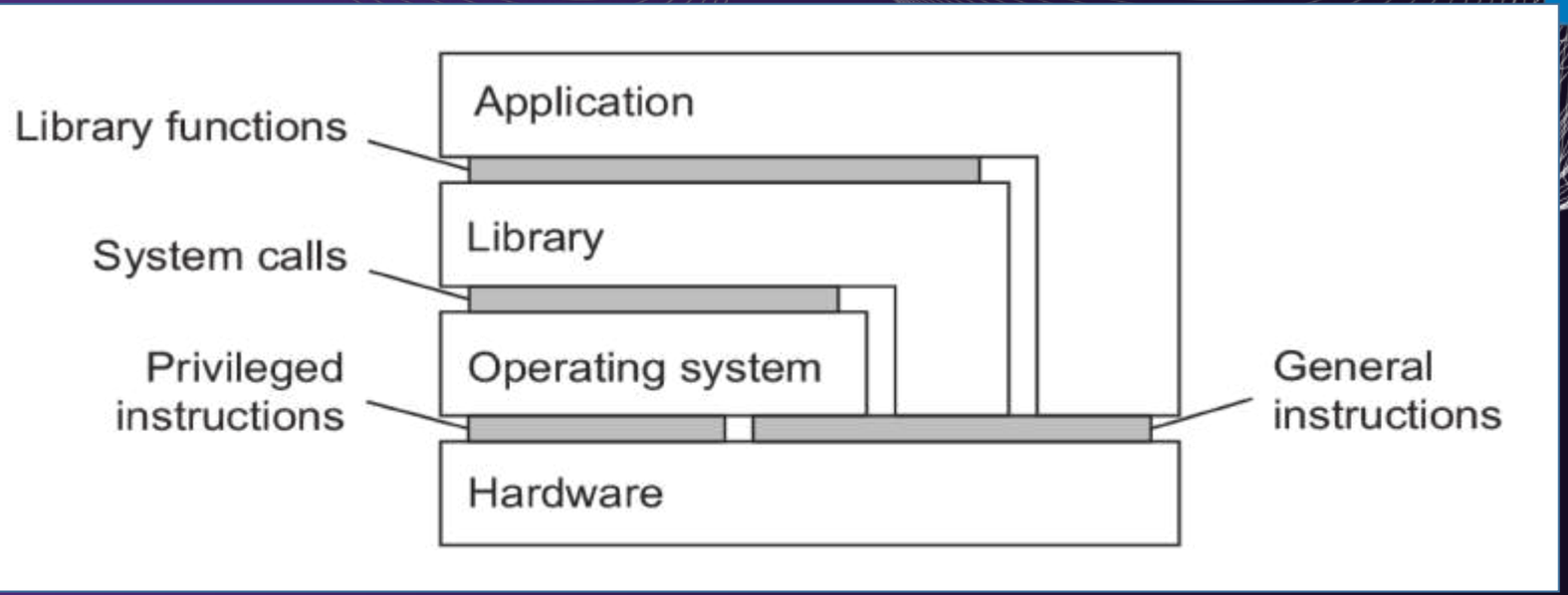
MIMICKING INTERFACES

- Hardware changes faster than software
- Ease of portability and migrating code
- Fault and attack isolation

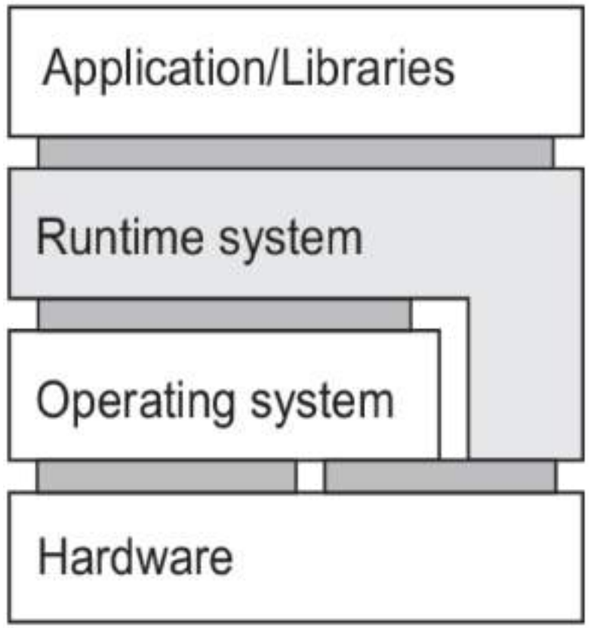


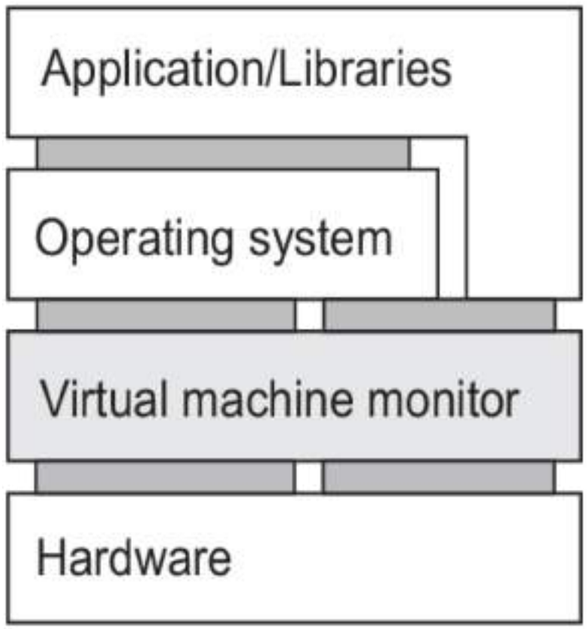
MIMICKING INTERFACES: TYPES OF INTERFACES

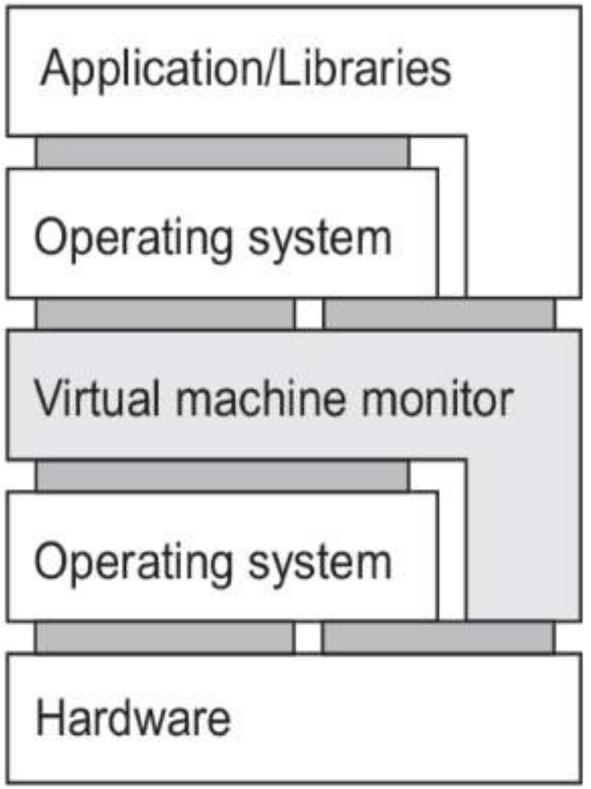
- Instruction set architecture:
  - Privileged instructions: allowed to be executed only by the operating system
  - General instructions: can be executed by any application
- System calls: what the operating system offers to applications
- Library calls: what specific libraries offer to applications



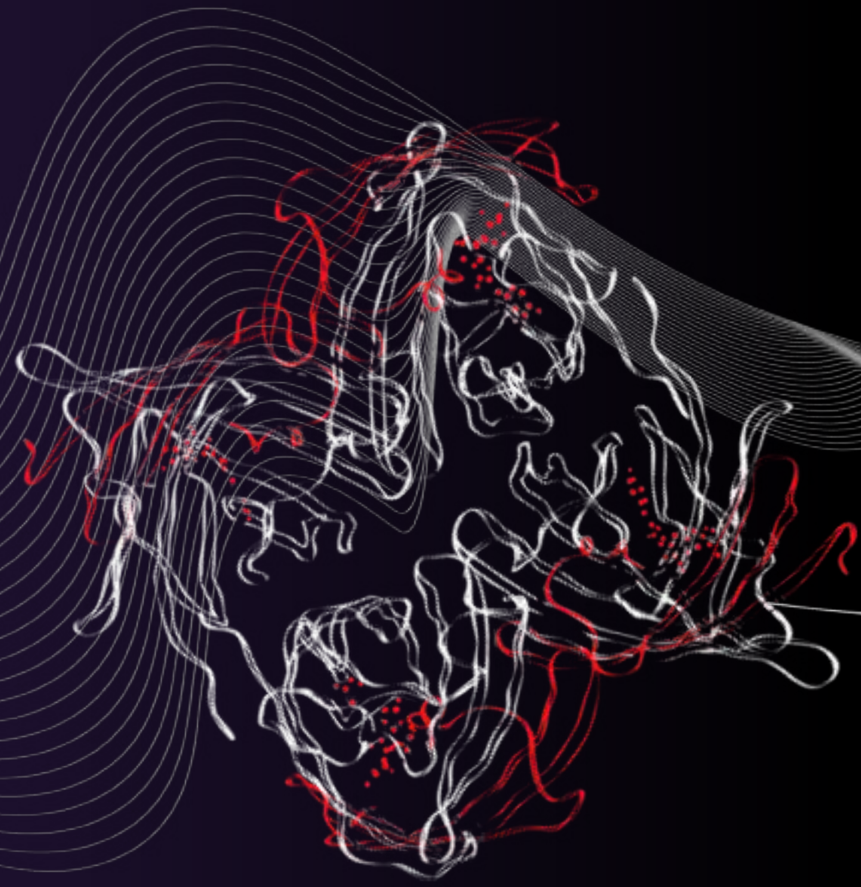
TYPES OF VIRTUALIZATION







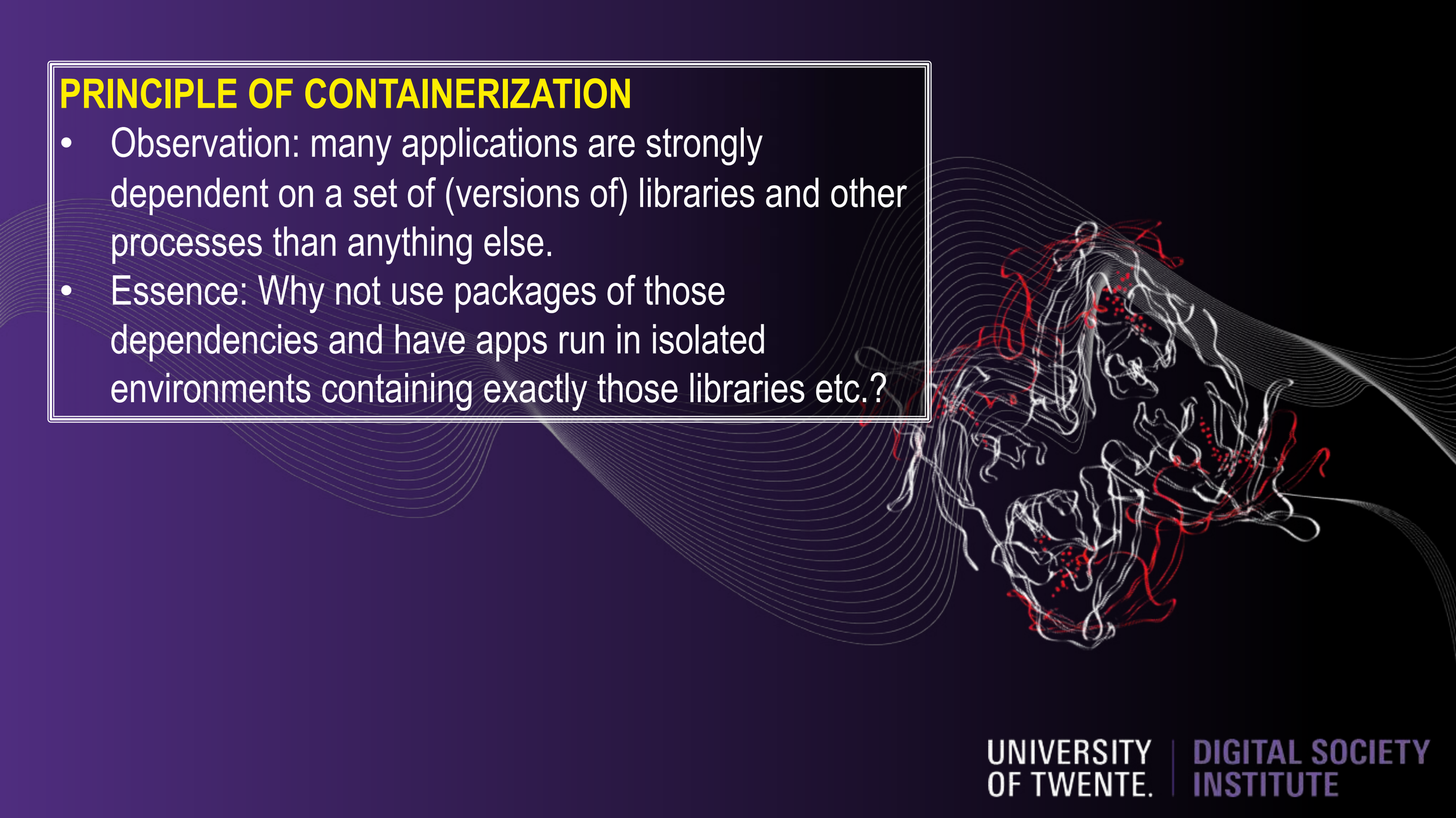
CONTAINERS



UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

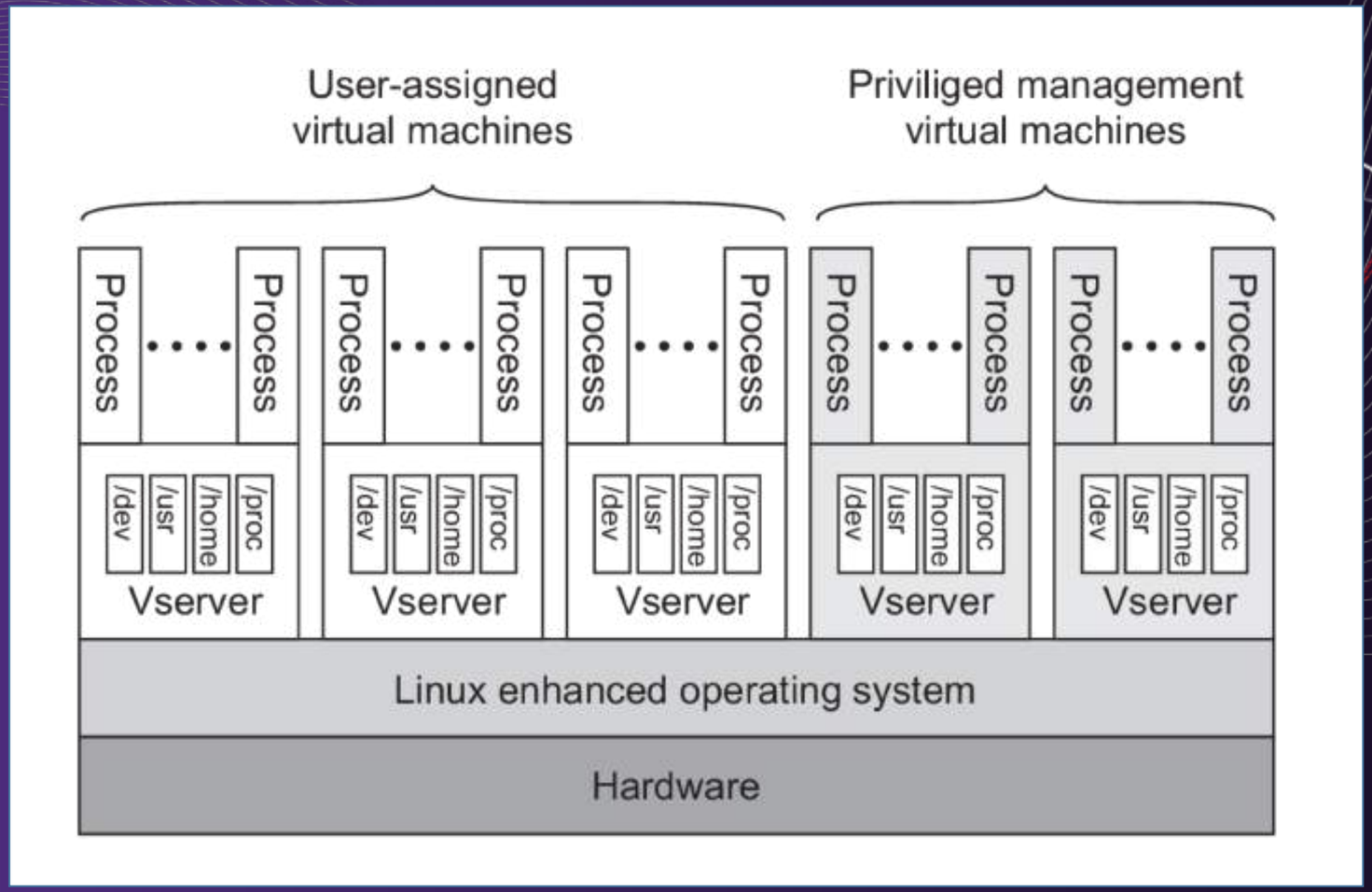
PRINCIPLE OF CONTAINERIZATION

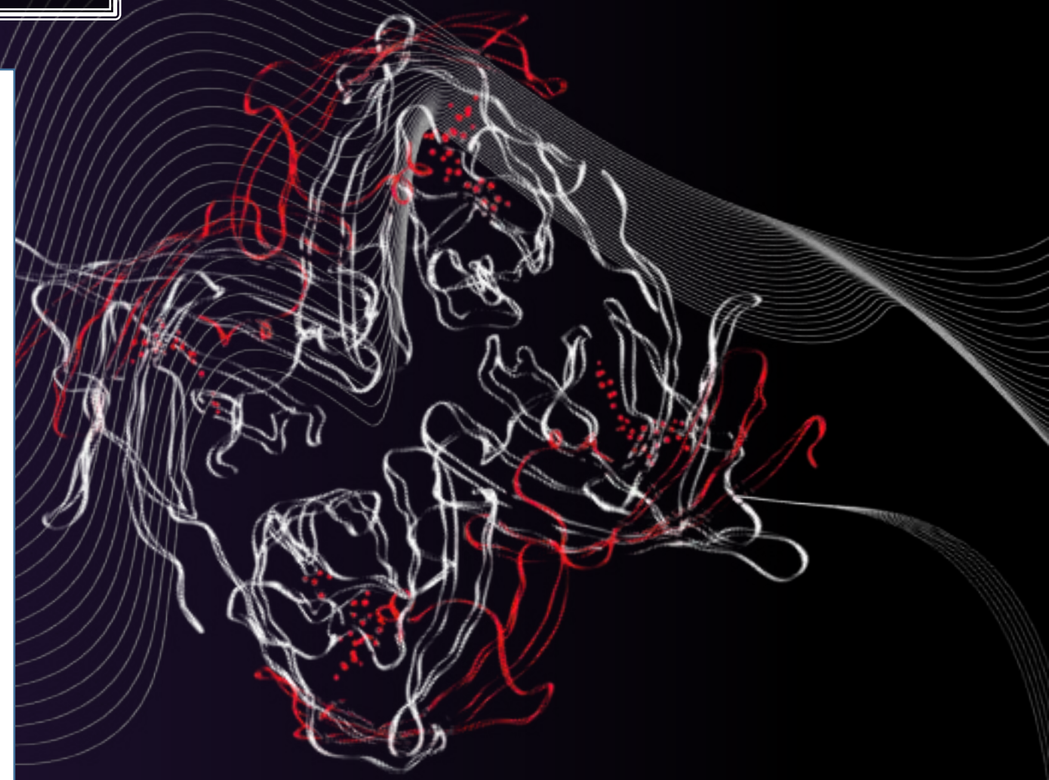
- Observation: many applications are strongly dependent on a set of (versions of) libraries and other processes than anything else.
- Essence: Why not use packages of those dependencies and have apps run in isolated environments containing exactly those libraries etc.?



PRINCIPLE OF CONTAINERIZATION

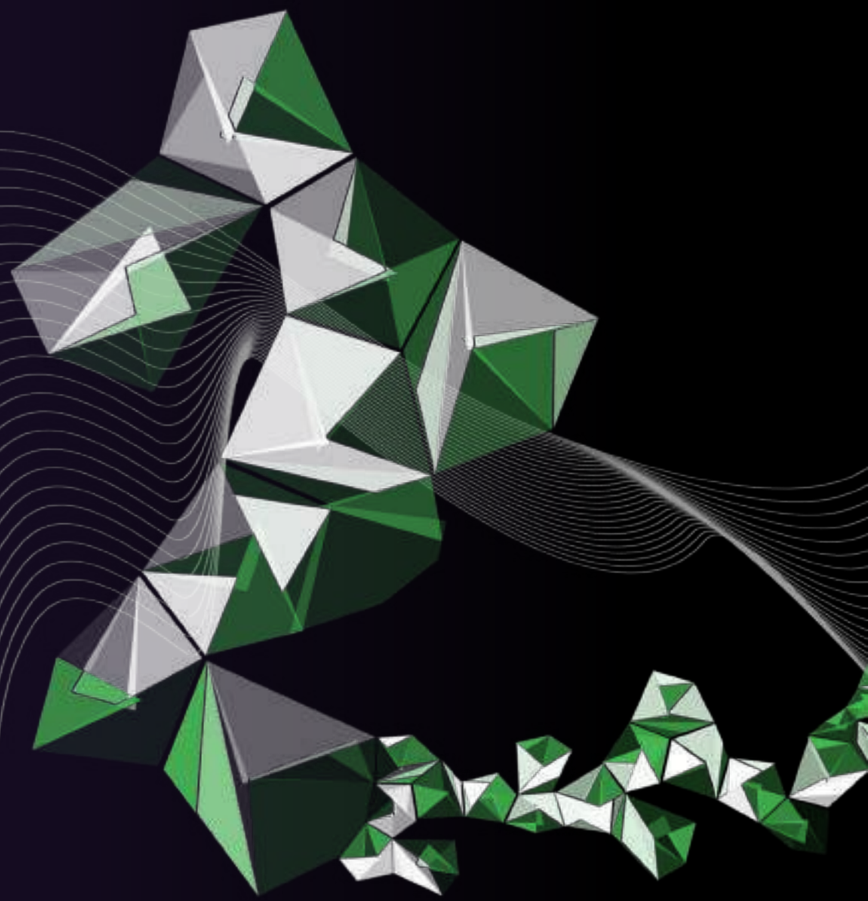
- Mimick the environment of an application
- Essence: make sure that namespaces are in order





UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

THE BIG DISCUSSION



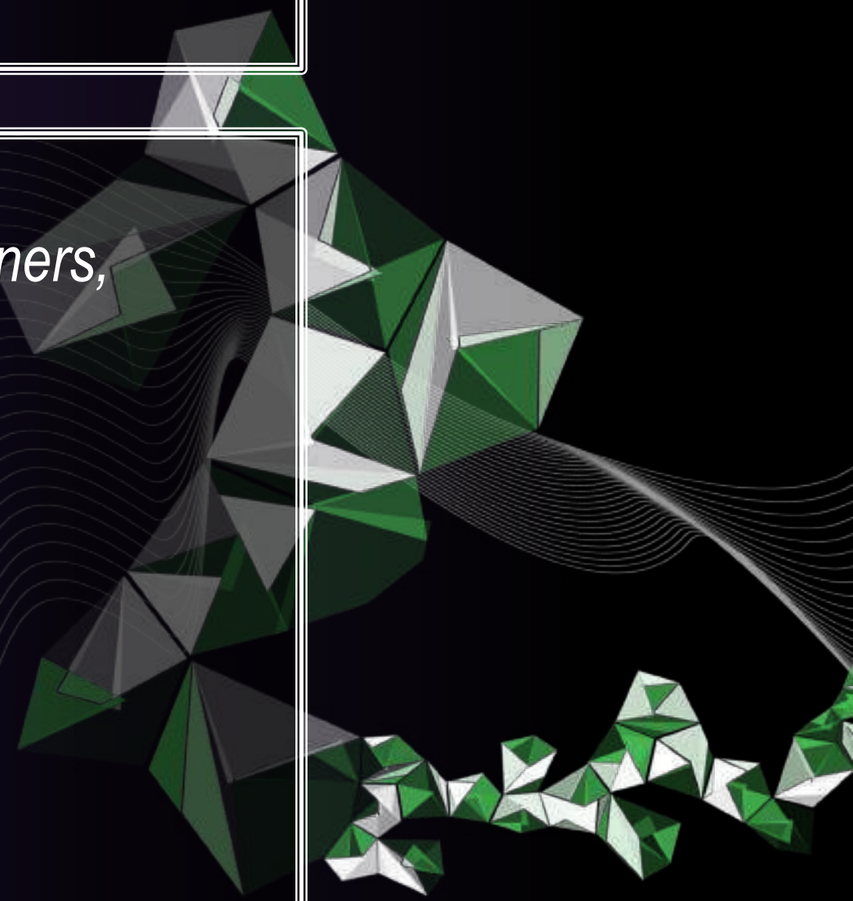
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

WHAT YOU CAN READ EVERYWHERE

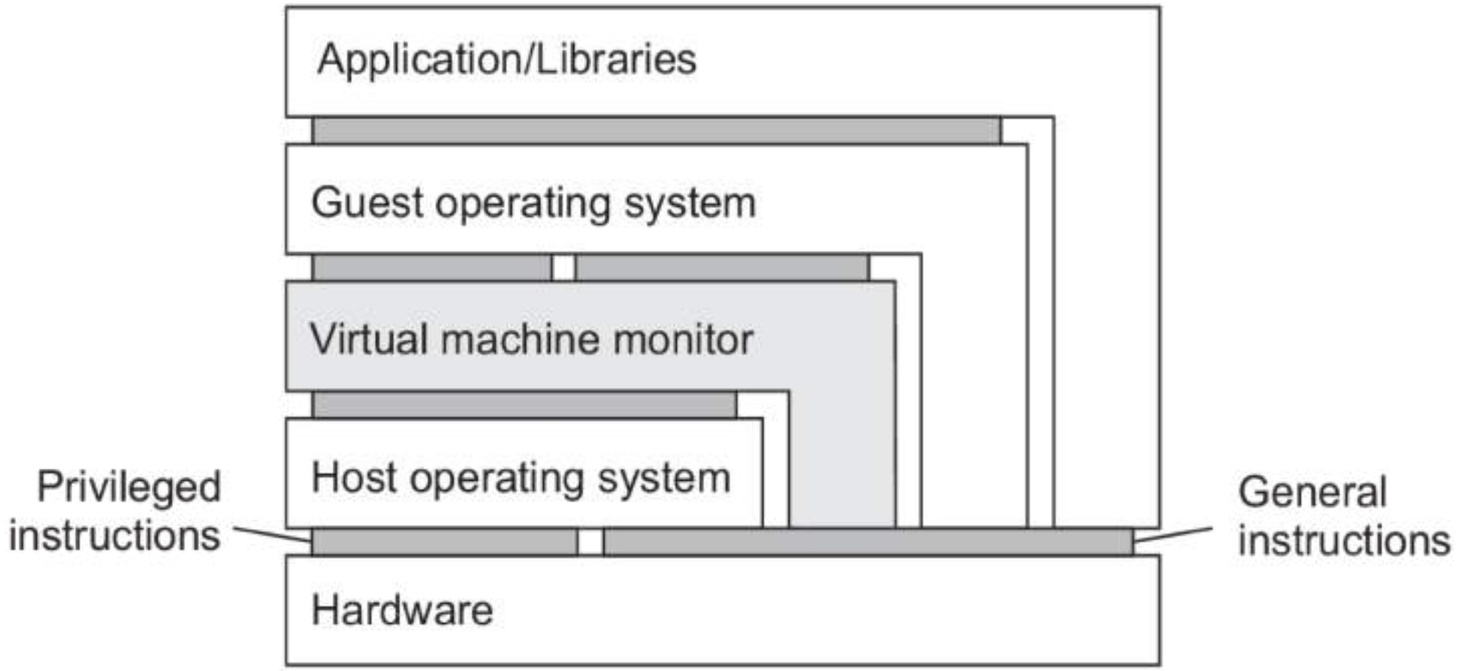
Docker containers are executable software package that includes all dependencies required to execute an application. With Docker containers, applications can work efficiently in different computer environments.

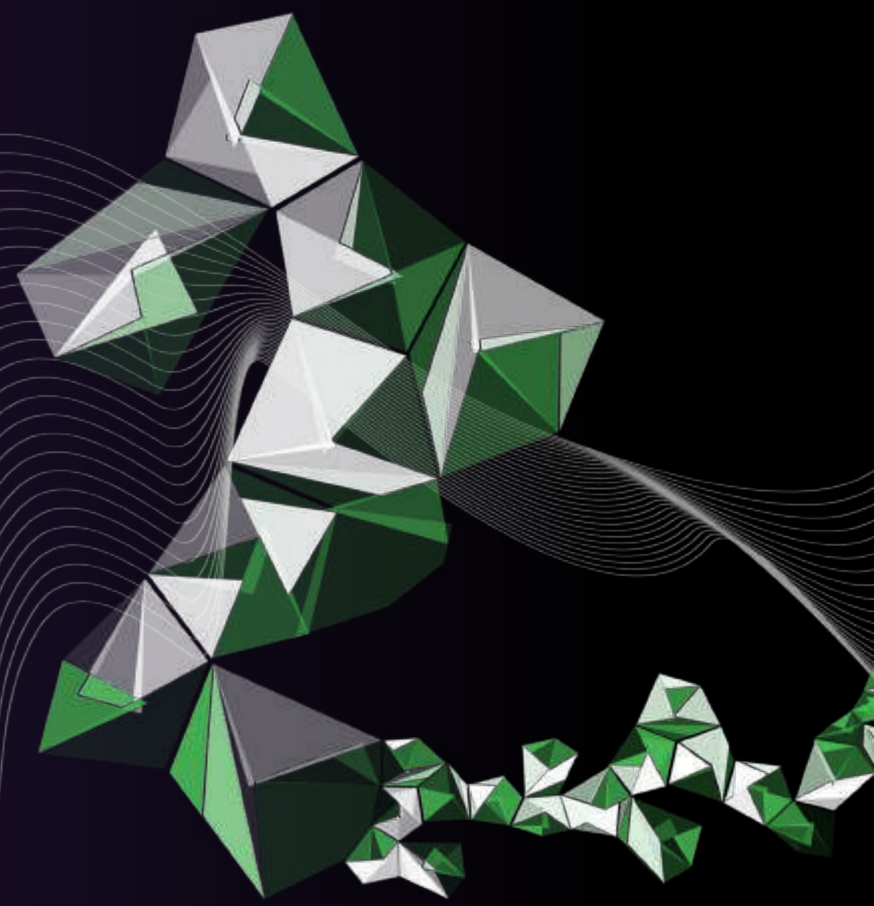
Below are the Docker Containers Features:

- *Lightweight*
- *Minimal overhead (CPU/IO/Network)*
- *Faster deployments*
- *Easily scalable*
- *Decrease storage consumption*
- *Portable, run it everywhere.*
- *Minimal base OS*
- *Application Isolation*



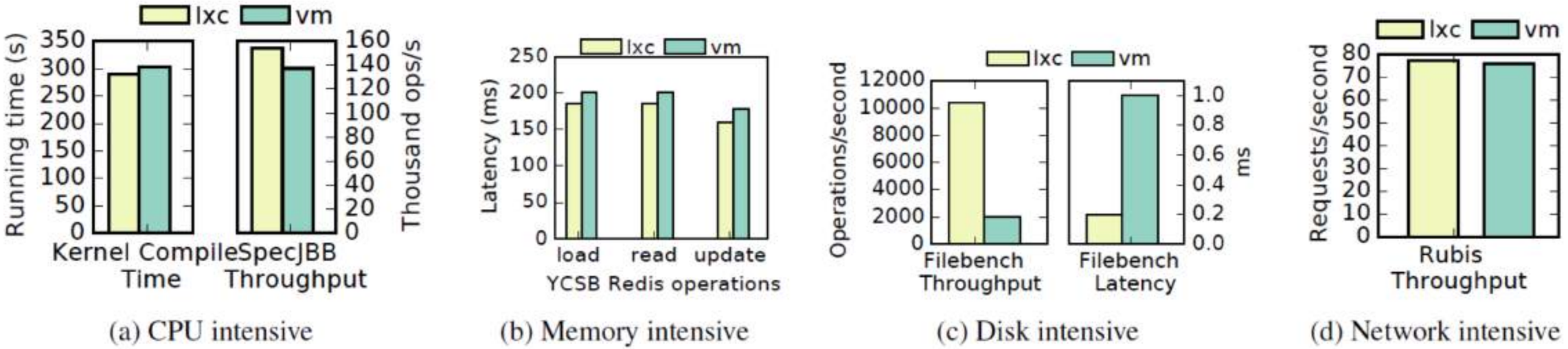
UNDERSTANDING PERFORMANCE





WHAT WOULD SEEM TO BE POTENTIAL PERFORMANCE BOTTLENCKS?

ON PERFORMANCE

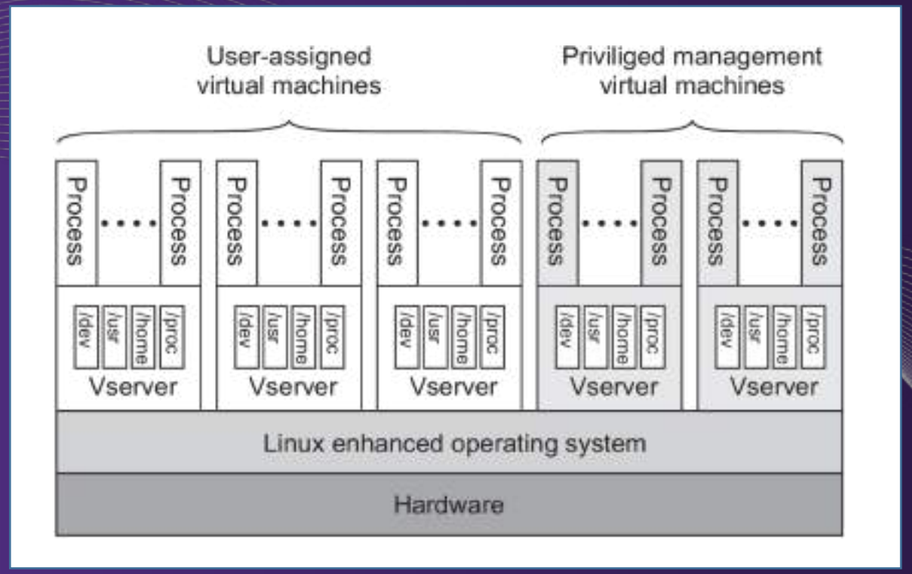


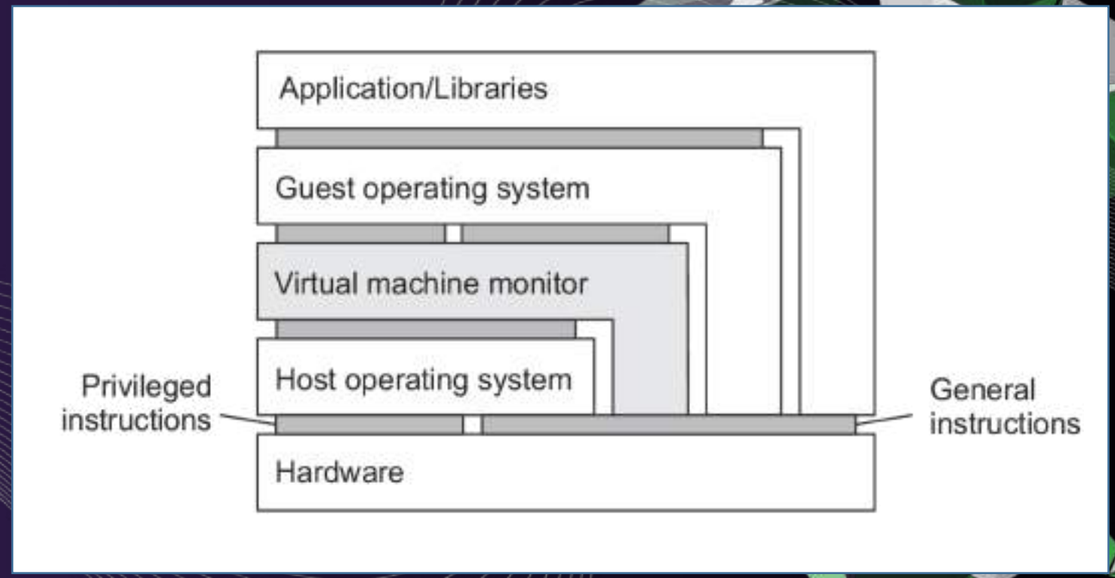
(a) CPU intensive (b) Memory intensive (c) Disk intensive (d) Network intensive

TAKEN FROM

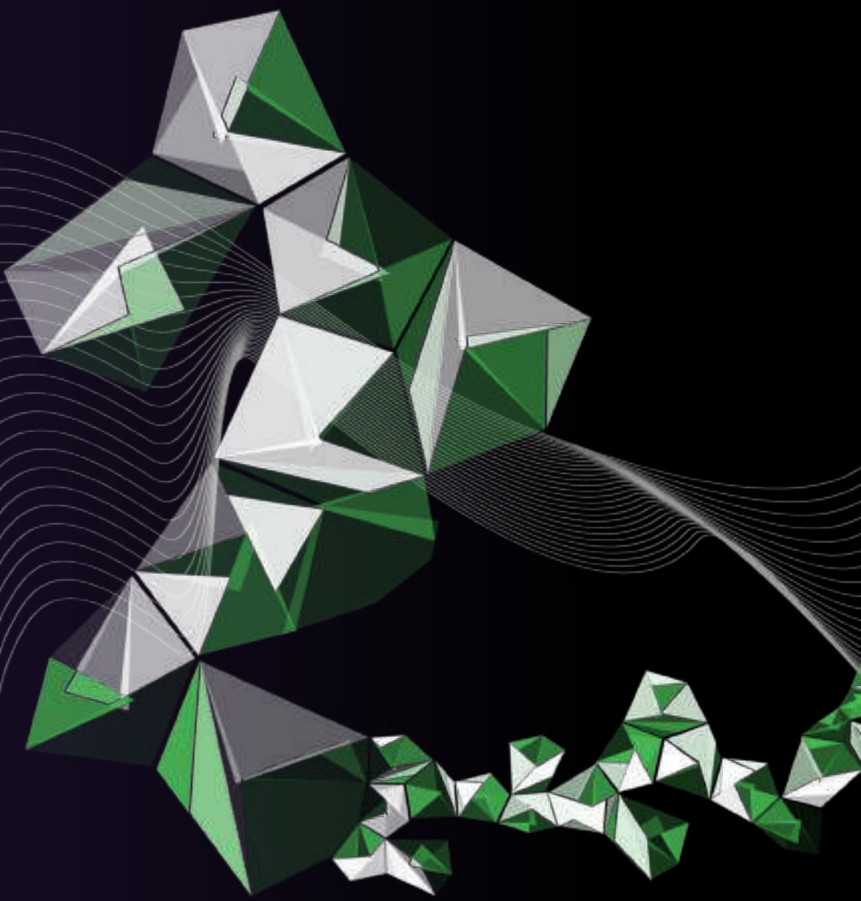
- Containers and Virtual Machines at Scale: A Comparative Study
  P. Sharma, L. Chaufournier, P. Shenoy, Y.C. Tay.
  Proc. 17th International Middleware Conference, December 2016, ACM.

ON PORTABILITY & MIGRATION





WORK TO DO



UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

STARTING POINTS & SUGGESTIONS

- The Ideal Versus the Real: Revisiting the History of Virtual Machines and Containers.
  A. Randal.
  ACM Computing Surveys, vol. 53(1), February 2020.

- Containers and Virtual Machines at Scale: A Comparative Study
  P. Sharma, L. Chaufournier, P. Shenoy, Y.C. Tay.
  Proc. 17th International Middleware Conference, December 2016, ACM.

- A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues
  IEEE Communications & Tutorials , vol. 20 (2), 2018.

- Distributed Systems book
  H1, H3.1, H3.2, H3.4, H3.5

- Key issue: develop your own opinion on virtual machines versus containers and be sure that you make clear to understand both technologies.