Hewlett Packard Enterprise

Hewlett Packard Labs

Deploying Containers on Secure HPC Systems

CANOPIE-HPC: 6th International Workshop

David Brayford November, 17, 2024

What do we want to do?

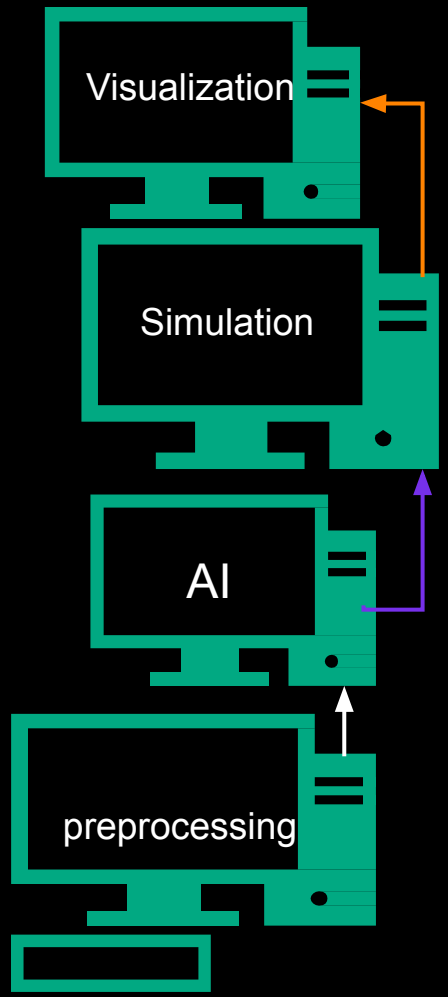
- Provide containerized software to the community
- Provide a mechanism for users to modify the containers
- Provide a simplified mechanism to easily extend the containers
- Provide verified container recipes to the community
- Provide a simplified container build process by working with the developers of the container technologies
- Provide a **standardized** recipes for all group to use across an organization

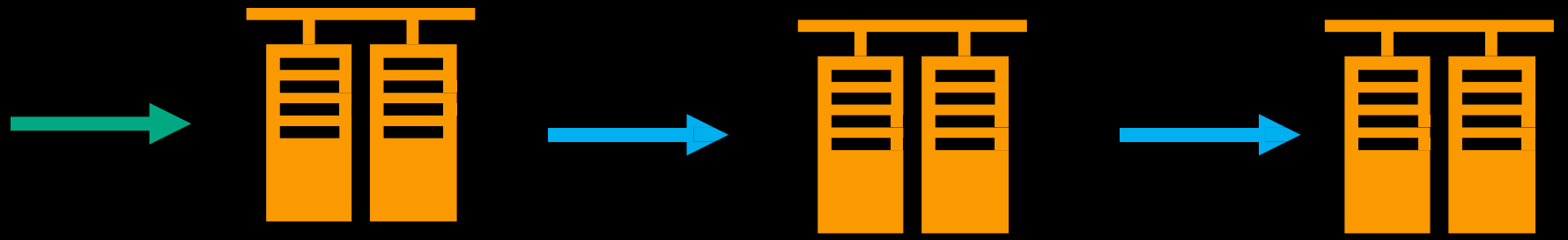
Develop a How-To guide as a community resource to help others.

The guide will evolve over time

Extensible HPC Containers

The ability to combine and transition multiple workflows from **different groups to HPC systems in an extensible way**





“Containers need to be <u>standardized</u> across all groups”
“Standardized containers need to be <u>extensible</u>”
”Containers need to be <u>flexible</u>”

“HPC CONTAINERS ARE NOT *<u>PORTABLE</u>*”
“HPC CONTAINERS ARE NOT *<u>REPRODUCIBLE</u>*”

Key Challenges

- Hardware configuration
- Data storage and data movement
- Security & center/system policies
- System software
- Execution patterns
- Dependencies
- Reproducibility
- Performance & stability
- Container size
- Licenses
- Host software
- **Flexibility, Extensibility, Standardization**

Key Challenges Security

- Known Unkowns (system specific)
- Security policies will be system and possibly partition specific
- Data movement & data staging can take several weeks
- A specialized build service is in operation
  - Applications and software needs to be built by center staff
- Building the software can take a long time if code must be verified before compiling
- Lack of flexibility
- Security policies will change
  - Security policies generally become more restrictive over time

Do's (Datacenter & Sysadmin) "*Suggested*"

- Understand the hardware, software and policies of the different HPC systems in the datacenter
- Provide a HPC container build environment
- Produce workflow diagrams for how the container will be deployed on the specific HPC systems and/or partitions
- Provide template recipes for every HPC system supported, which configure the environment for GPUs/accelerators, MPI & communication infrastructure and directory mount etc
- Provide container recipes and container images for "standard" workflows used on the system
- Provide detailed and clear documentation, How-Tos and examples

Don’ts (Datacenter & Sysadmin) “*Suggested*”

- Don’t provide a service you don’t understand
  - Unexpected issue frequently occur
  - Spend a lot more time supporting users than expected
  - User complaints and poor quality of service
- Don’t reinvent the wheel
- A single method, policy won’t be suitable for all the systems you expect containers to be deployed on

Do's (Users) "*Suggested*"

- Build containers from recipes (Dockerfiles)
- Easier to replicate, modify and verify
- Easier to produce performant workflow based on the specific hardware & software setup of the system
- Recipes/Dockerfiles are easier to inspect and transfer between systems
- Minimal space required for storing recipes (Dockerfiles)
- Use verified templates/recipes provided to create new workflows
- Easier and quicker to debug
- Understand how the HPC container runtime works
- Default environment configurations, overrides
- Understand the policies and operational concepts of the system you want to use

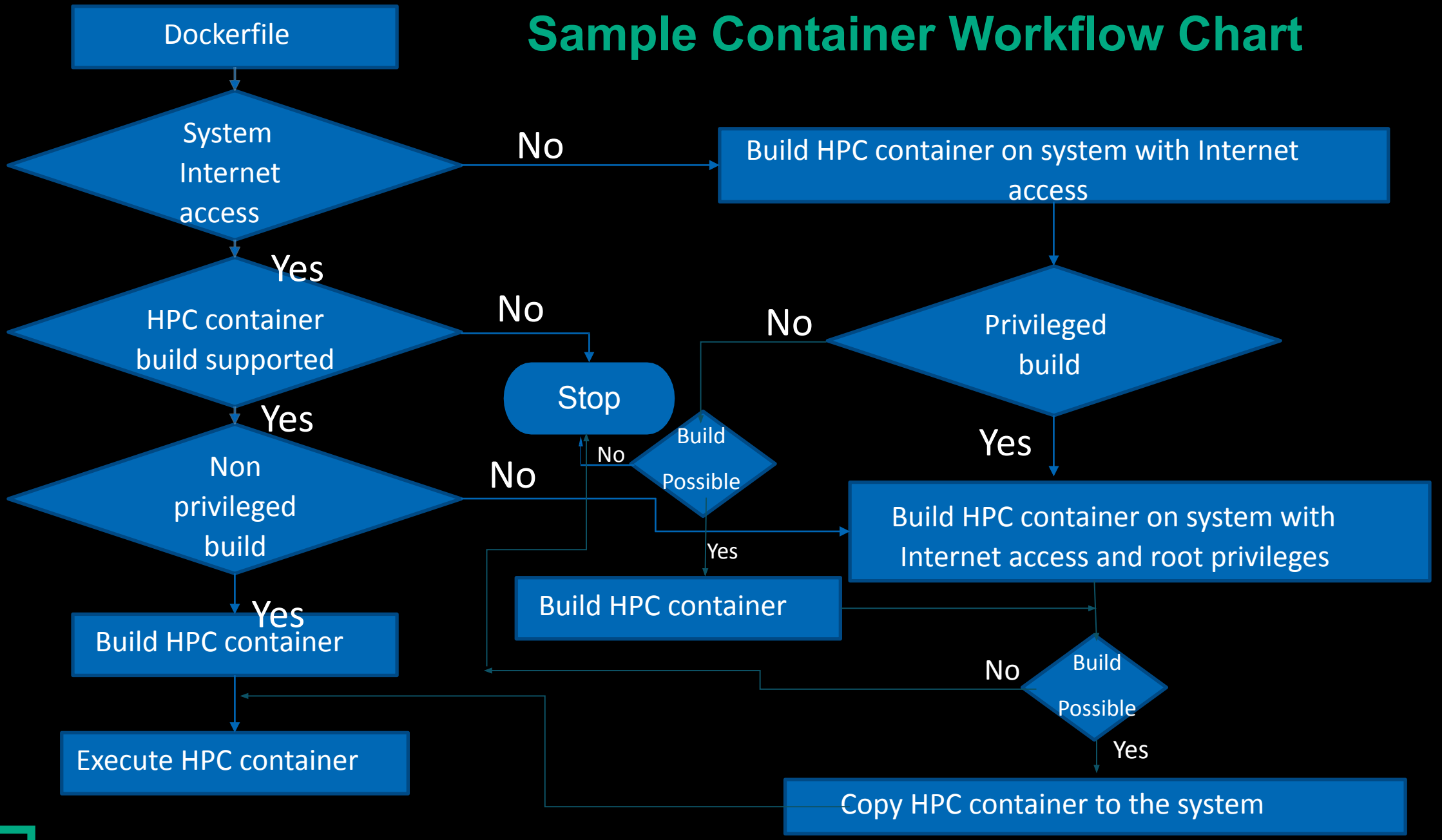
Don’ts (Users) “*Suggested*”

- Don’t bring a container image from another system or repository
  - HPC containers are “***NOT***” portable
  - Potentially difficult to get working, probably not performant, hard to debug
- Don’t store data in the container, keep the container and data separate
  - Data transfer & storage, stability and performance issues with TB size containers
- Don’t install everything you might ever need in the container image
  - We want minimal size container images
- Don’t require additional components to be installed by the data center on the system
  - Changes to a production system in operation is often not possible

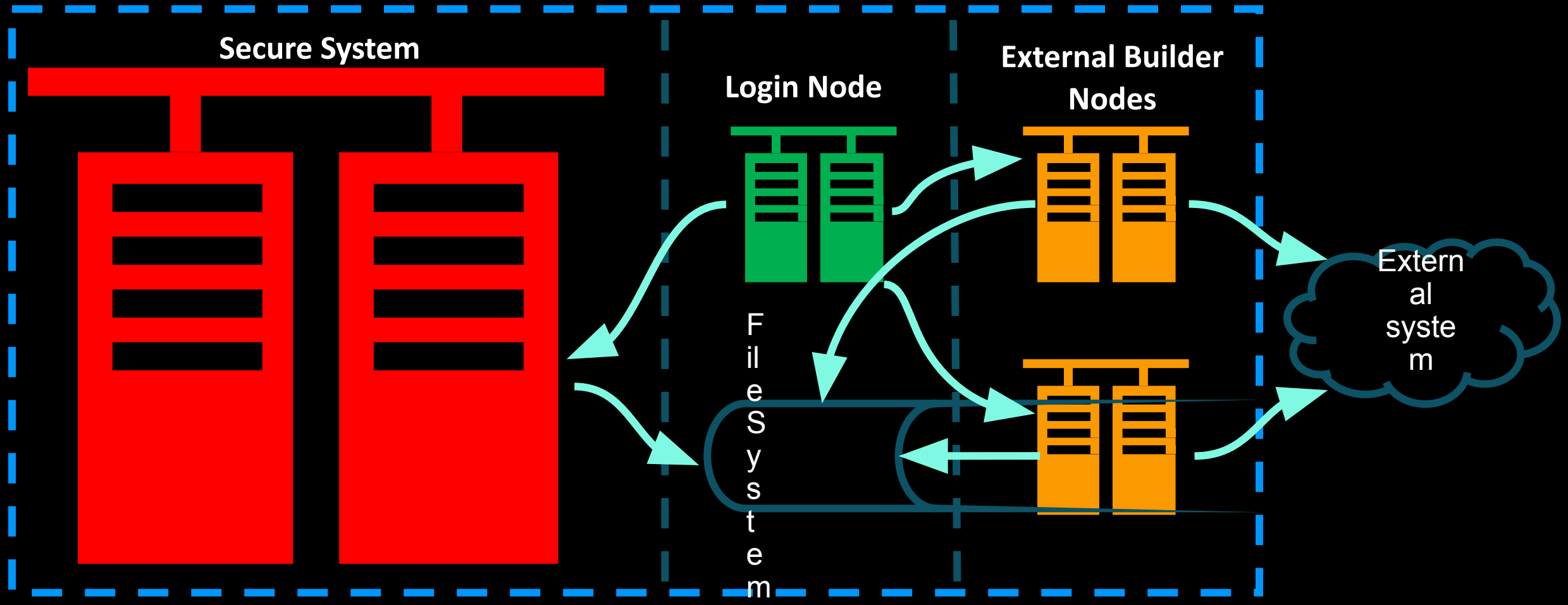
Don’ts (Users)

- Don’t perform an operation inside a container that would crash the system
  - If an operation crashes the system when running directly on it, expect the same to happen if run inside a container
- Don’t expect the container to behave the same if the underlying system has changed. HPC containers are “***NOT***” reproducible
  - The containers are dependent on the underlying system
- Don’t install licenced software that can be executed outside of its licence a greement
- Don’t install anything inside container directories which are mounted from the host at runtime
  - Some HPC container technologies mount the hosts $HOME inside the co ntainer at runtime.

Sample Container Workflow Chart



Building containers on secure systems



Key Takeaways

- Bring together the key stakeholders to formulate a "**Standard**" recipe format for your organization and systems
- Have a consistent method for the generation of recipes across all groups in the organization
- Ensure that the recipes are consistent with different container technologie s
- Simplify the process of creating Dockerfiles and container images
- Work with HPC container developers to provide a simplified build and deployment methods

Think of building a structure out of Lego blocks, they need to fit together

Key Takeaways

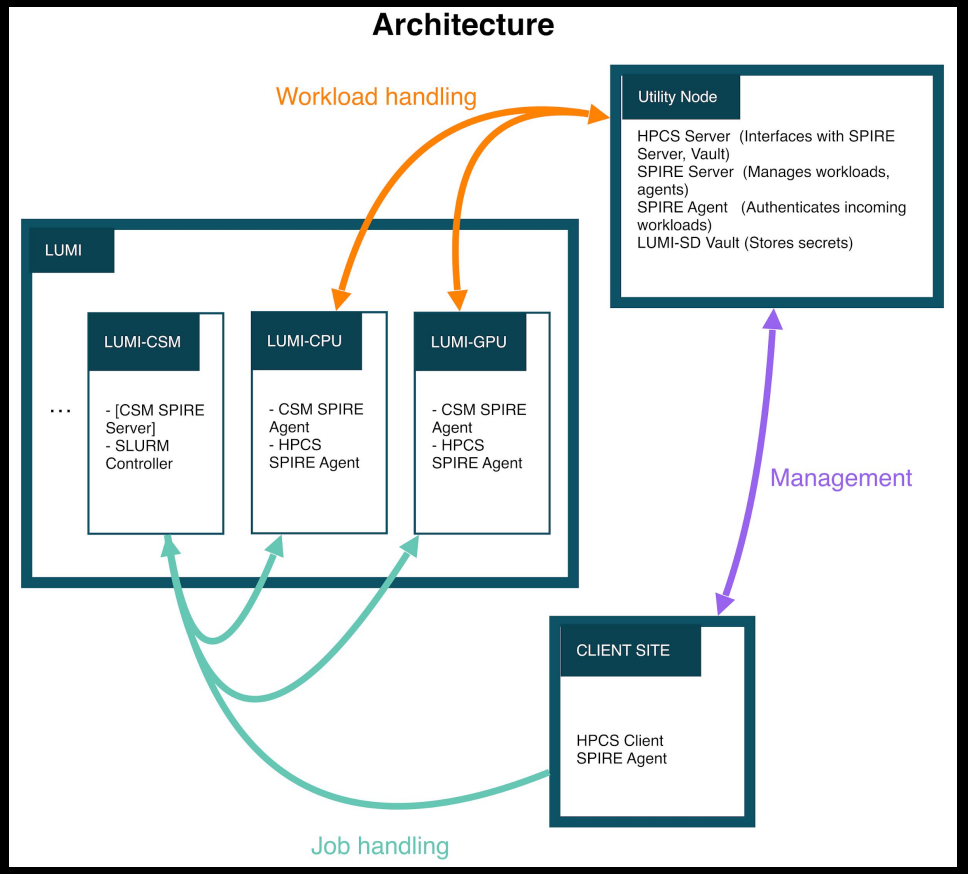
- Build containers from recipes
  - Easier to replicate, modify and verify
  - Easier to produce workflow and system specific containers
  - Generate containers with smaller memory footprints
  - The ability to generate performance portable containers based on the specific hardware & software setup of the system
  - Minimal space required for storing recipes
  - Use existing verified recipes to create new recipes and container images

Think about cooking or baking, it's very difficult to remove an ingredient from cake or meal

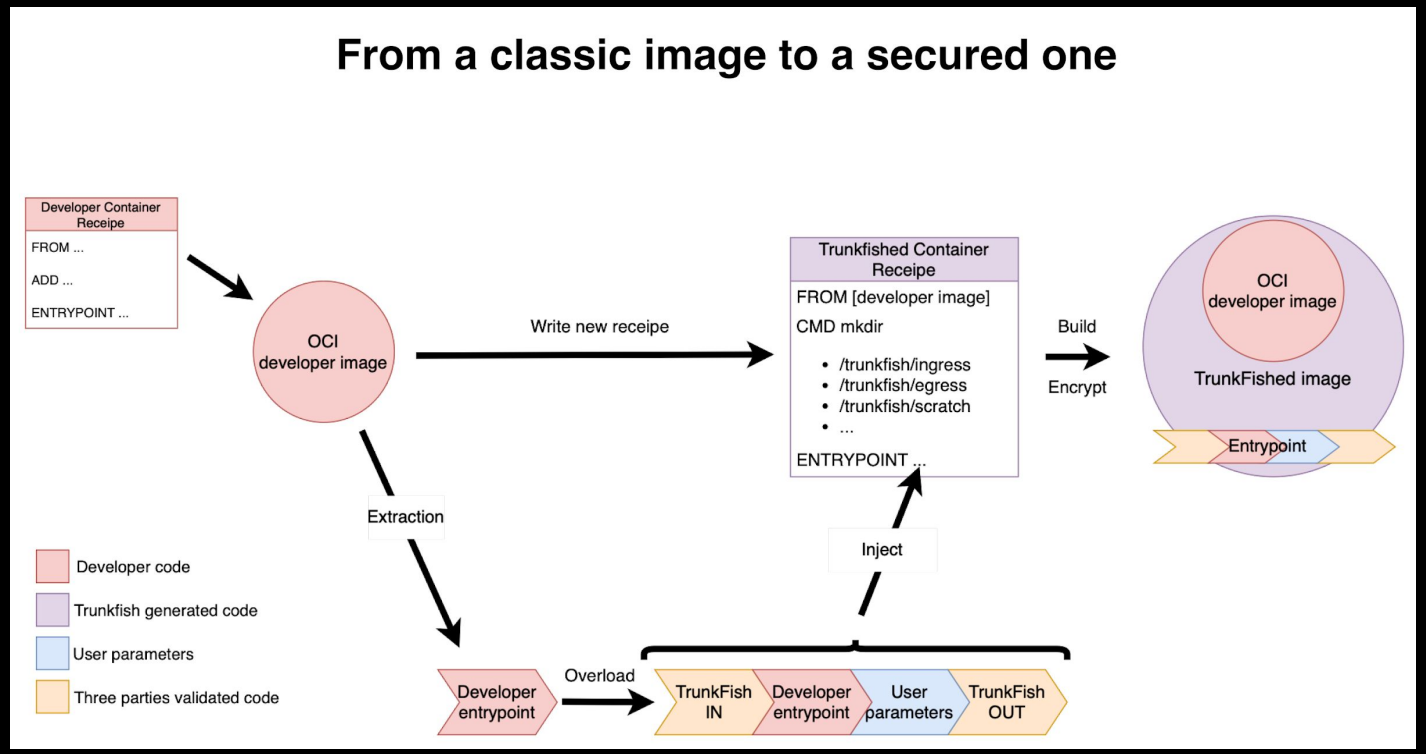
Secure Workloads & Confidential Computing

- We want to ensure that nobody with access to the HPC system can examine the unencrypted data
- Becoming more important with the medical community who want to use HPC resources to process data faster
- Becoming more important for industry and workloads that run in the cloud
- The data must remain encrypted and only decrypted within the secure encrypted enclave (container runtime)
- The containerized secure workflows are often difficult to configure and impact performance negatively

High Performance Computing Secure (HPCS)

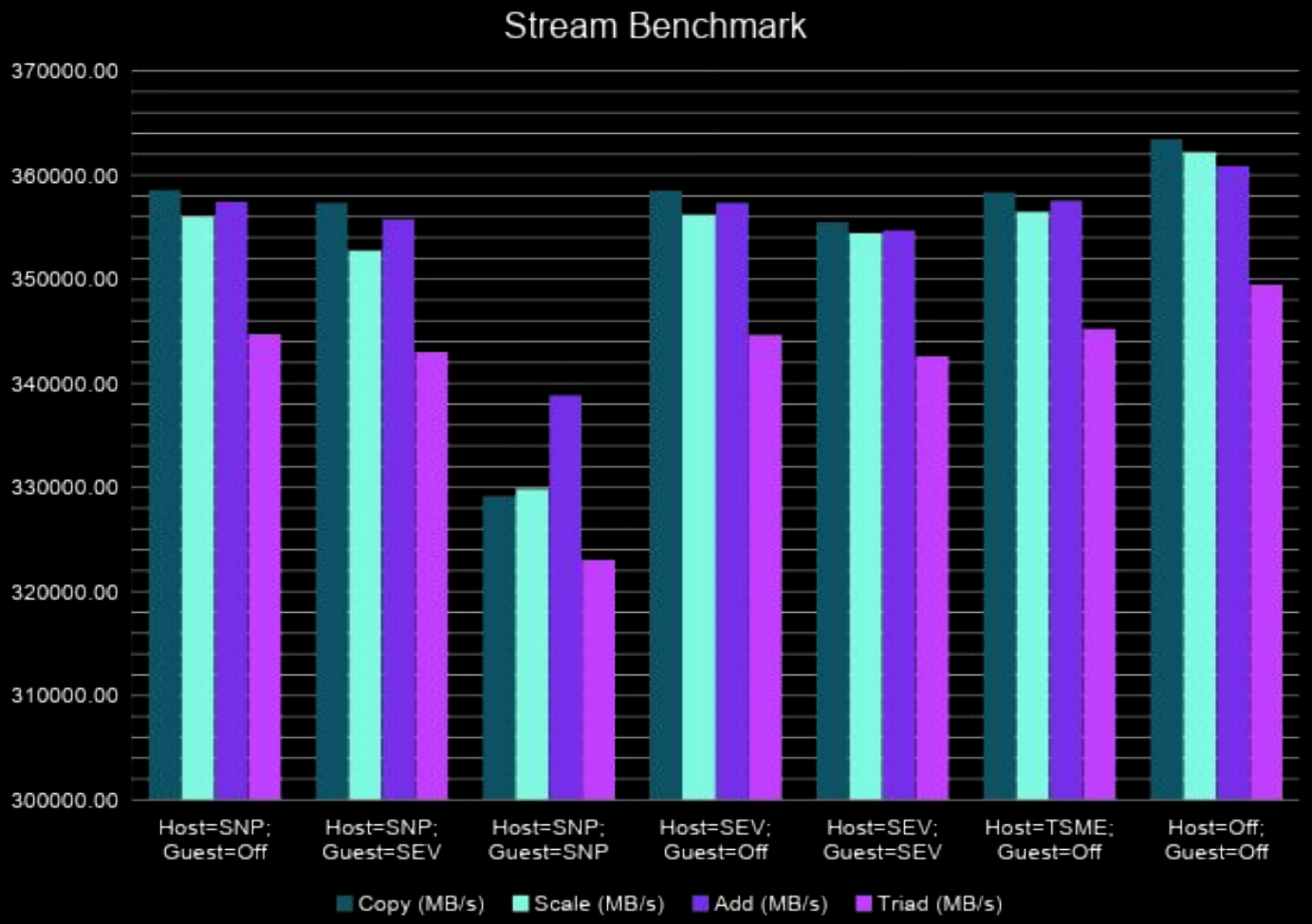


Architecture for secure HPC
https://github.com/cscfi/hpcs



- Encrypted containers without root access
- Workload attestation & identity management (SPIRE)
- End-to-end encryption for ALL data
- Zero-trust model
- K8 based for an easy and modulable deployment

Performance impact for memory encryption in confidential computing



VM configuration

- 12 core
- 256 GB allocated RAM
- Rocky Linux 9.4
- Kernel 6.10.0-snp-guest

For Stream 5.10 copy benchmark:

export OMP_PLACES=0:12:1
export OMP_NUM_THREADS=12

Performance impact:

- TSME on host: **1.4%**
- Non-encypted VM, with SEV enabled on host: **1%**
- SEV-ES: **2%**
- SEV-SNP: **10.4%**

AutoTuning HPC applications

- Provide system specific build environments and tools within a container
- Enable building and executing of application on nodes with a minimal OS, which don’t support source code compilation (missing header files)
- Submit a job that builds and tests multiple different compilation configurations for the application to find the best configuration for CPU’s & GPU’s

Automatic Tuning of HPC Applications: The Periscope Tuning Framework

Distributed AI Workflows

- Provide containerized AI frameworks with MPI support
- Enable the AI workload to be executed across 100's or 1000's of compute nodes
- Containers provides a mechanism to run different AI workflows and software versions on a system without worrying about compatibility issues

"Deploying Scientific AI Networks at Petaflop Scale on Secure Large Scale HPC Production Systems with Containers"

"Deploying AI frameworks on secure HPC systems with containers"

Quantum Computing

QuantEx project, PRACE Horizon 2020

- Provide a mechanism to deploy the QuantEx distributed quantum simulator on various production and experimental HPC systems
- Executed on Intel x86, AMD x86, Arm ThunderX2, Arm A64FX, IBM Power9 and Nvidia V100 GPU's
- Developed in Julia with MPI and profiled using LIKWID
- Executed on up to 320 nodes of Kay, ICHEC

david.kenneth.brayford@hpe.com