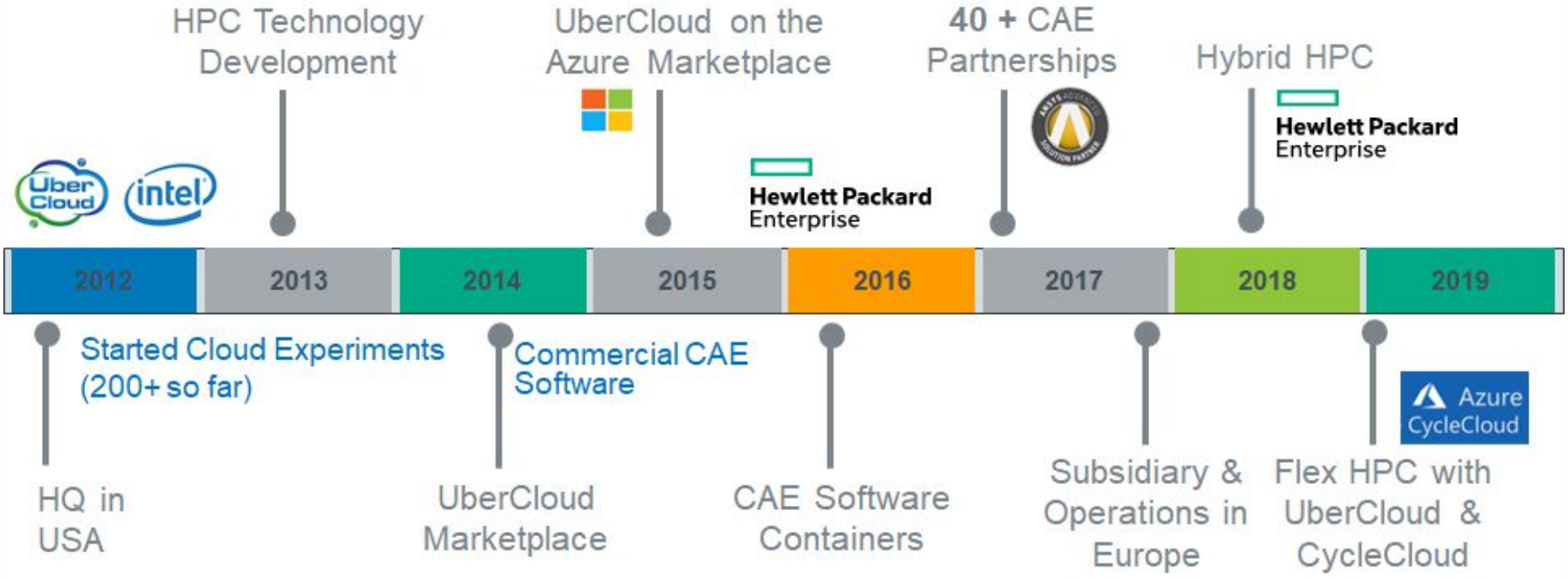
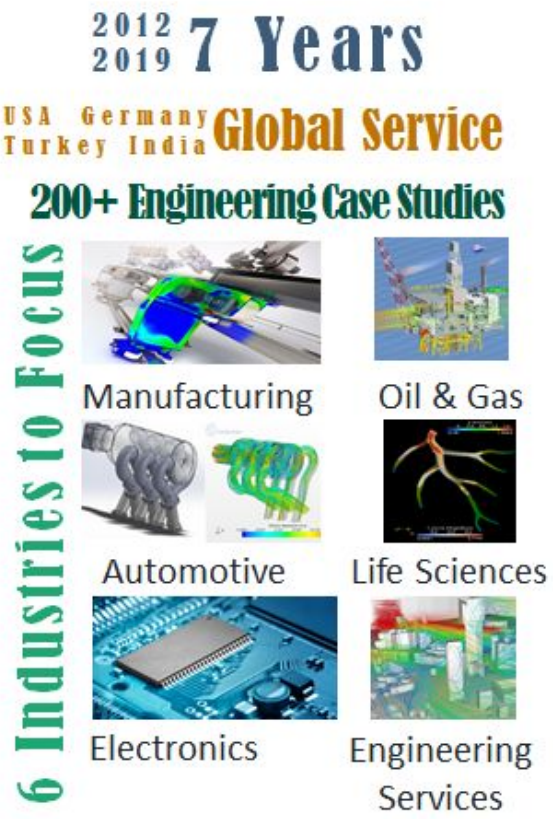

# UberCloud Containers on Kubernetes

# Brief History of UberCloud

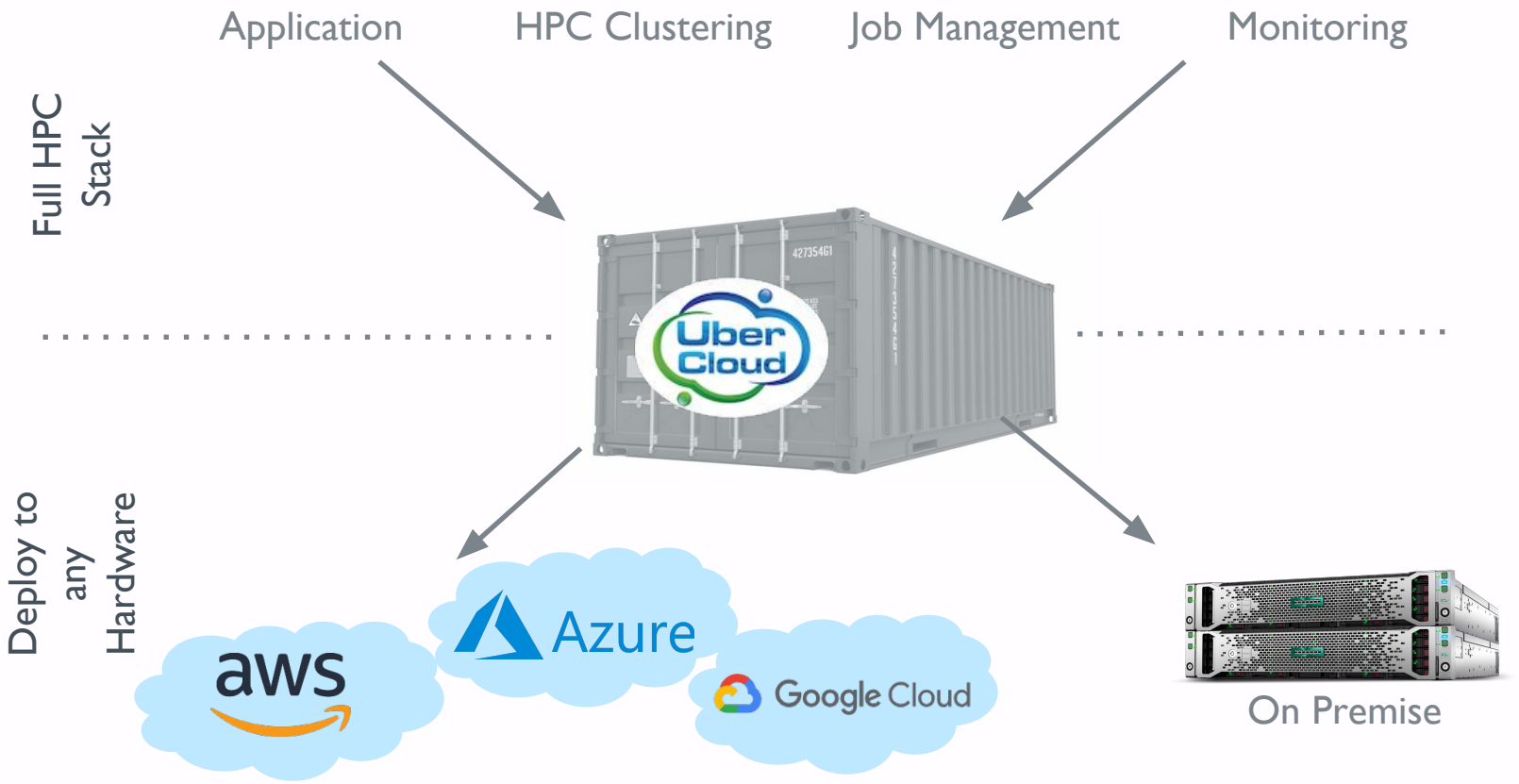


# References and Awards





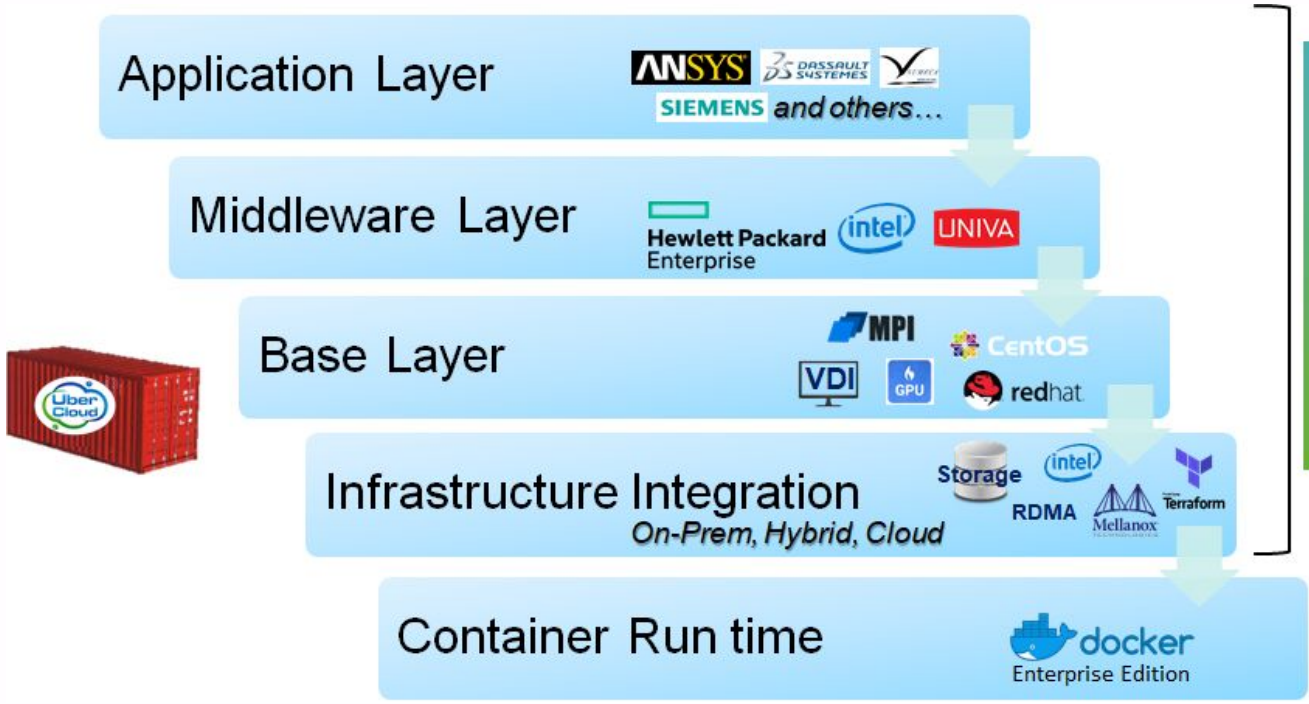

# Why / How we use UberCloud Containers



# UberCloud Containers provide:


- OCI compliant images containing all what an Engineer requires
- Ready to deploy in different Cloud Environments (Azure, AWS, GCP, Oracle Cloud) and on-premise environments without any changes
- Out of the box HPC layers with MPI, Devices, Storage, and Visualization capabilities
- Fully interactive GUI Nodes with engineering applications preconfigured
- GPU Support for visualization and computation
- Tight integration with batch scheduling system without manual configuration requirements
- Seamless Hybrid HPC and Bursting capabilities

# UberCloud HPC Containers



# Kubernetes: One way to orchestrate UberCloud

- The de-facto standard container orchestration platform which is supported by many companies (Google, Amazon, Microsoft, RedHat, Pivotal, …)
- Many organisations introduce Kubernetes in their IT environment either on-premises or in cloud. It is a known technology for Administrators and IT.
- UberCloud HPC containers make it simple for Engineers and administrators to run Applications like Comsol, Ansys, … in existing Kubernetes environments.

Alternatively: Orchestrating through Job Schedulers (like Univa Grid Engine), Atrio Composable Cloud, custom Infrastructure as Code implementations.

# Why UberCloud Containers on Kubernetes?

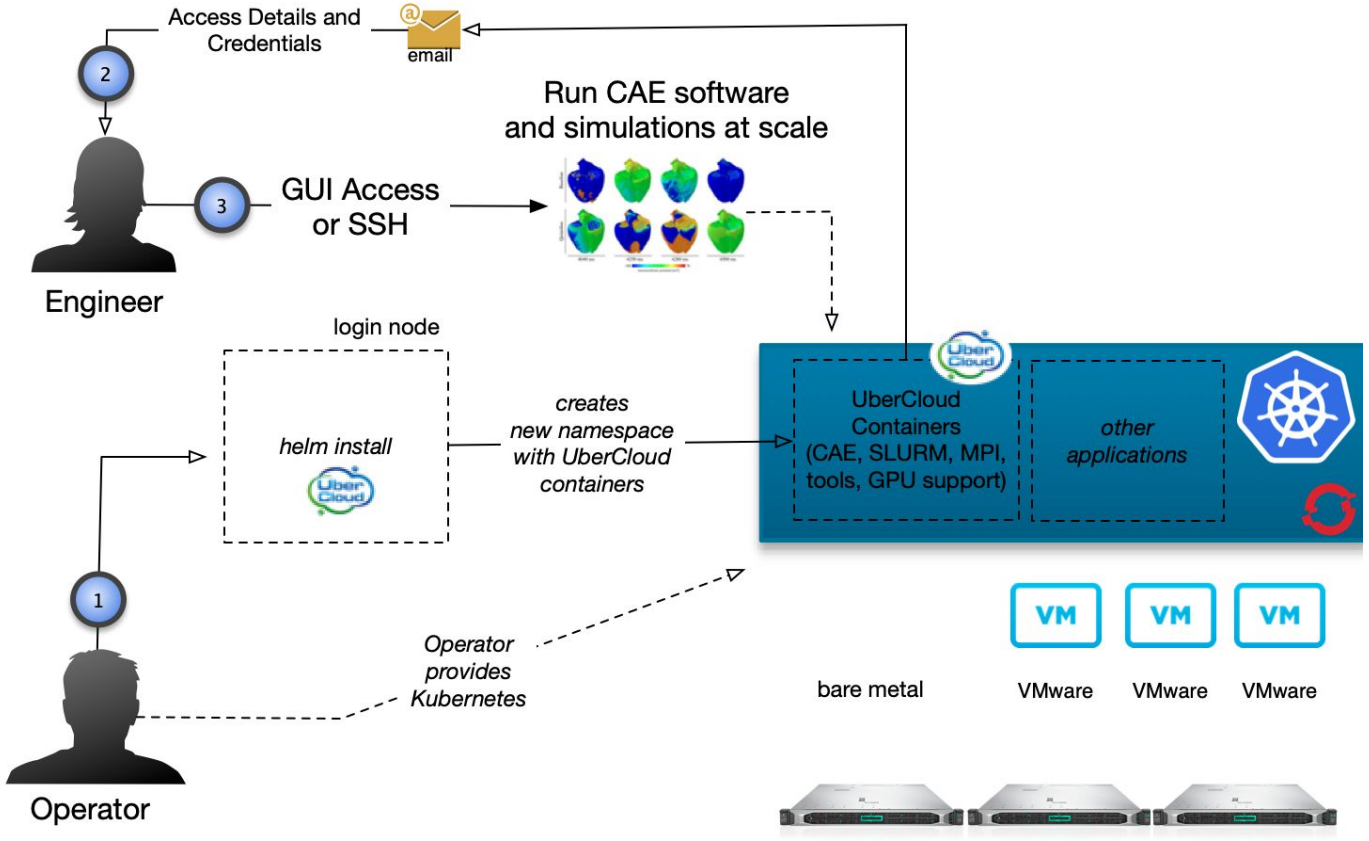
- ★ Speed up HPC adaptation
- ★ Maximize resource usage
- ★ Secure environment
- ★ No learning required - full GUI with virtual desktop
- ★ Wide availability of Kubernetes expertise in IT

- Enable Kubernetes to run HPC workloads
- Exploit under-utilized powerful resources in pre-existing Kubernetes environments
- Engineers can be 10-times faster executing simulations when using more powerful resources
- Always run the latest software with the newest security patches. No worries about updates.
- Disposable and dynamically requestable resources - get only charged for resources when you actually use it
- No complex setup required
- Accepted solution by IT - common understanding of the container technology

# UberCloud Containers address these gaps

- Kubernetes lacks HPC scheduler capability
- Kubernetes has no license orchestration
- Kubernetes does not provide native MPI support
- Kubernetes lacks devices and controllers needed for engineering software and HPC

# How?


Access Details and Credentials
email
2
Run CAE software and simulations at scale
3
GUI Access or SSH
Engineer
login node
helm install
creates new namespace with UberCloud containers
UberCloud Containers (CAE, SLURM, MPI, tools, GPU support)
other applications
1
Operator provides Kubernetes
Operator
VM
VM
VM
bare metal
VMware
VMware
VMware

# An HTC Case on OpenShift + UberCloud Containers

**Mixed OpenShift setup (virtualized and bare metal):**

1 ESXi host serving 3 VMs for the OpenShift control plane
*(HPE ProLiant DL360 Gen10)*

2 VMs + 1 bare metal host for running UberCloud containers in OpenShift
*(HPE Apollo 6500 Gen10)*

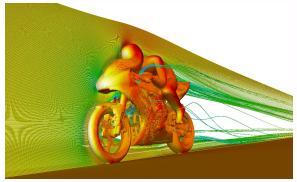
**Kubernetes:** OpenShift v3.10 / Kubernetes 1.10

**Workload:** MPI jobs (shared memory and distributed memory)

**Application:** OpenFoam with Motorbike model

**Workload Management:** UberCloud containers running containerized slurm workload manager

**Results:** Successfully run single node and multi-node jobs in a distributed UberCloud container setup running all workload (job submission + job execution).



# Thank you

burak@TheUberCloud.com