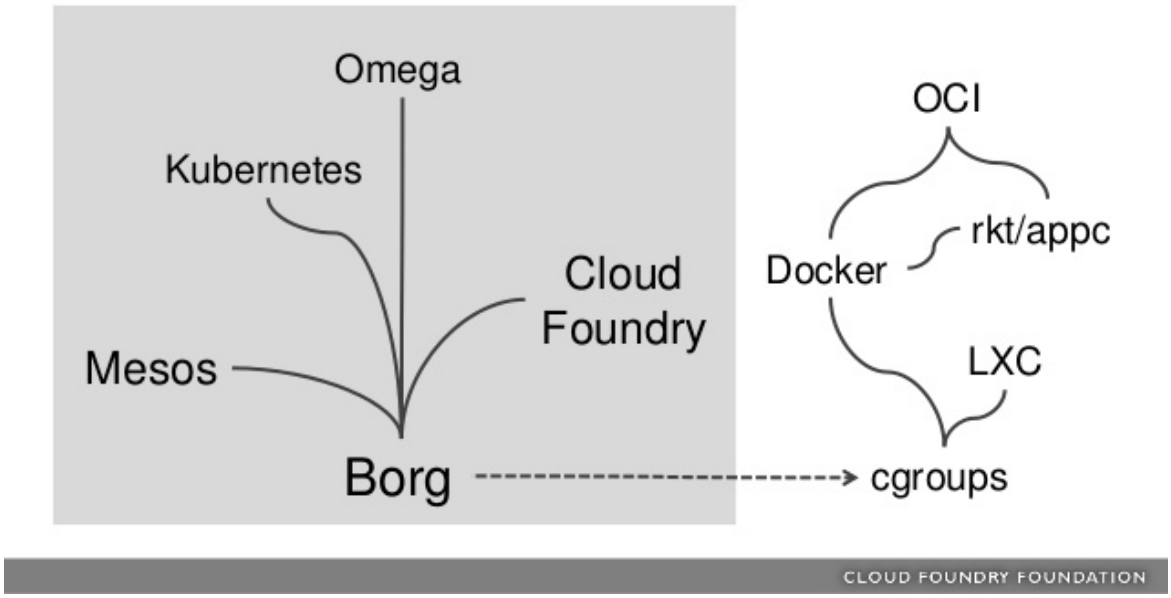
Why Kubernetes?

Why Kubernetes? Borg Heritage

What primarily distinguishes Kubernetes from other systems is its heritage. Kubernetes is inspired by Borg - the internal system used by Google to manage its applications (e.g., Apps, GCE, Gmail).

Kubernetes is a beneficiary of over 15 years of lessons learned by Google in writing and operating Borg. Kubernetes becomes a safe choice for a container management due to its heritage. The current growth in Kubernetes is making it easier to implement and handle workloads not found in a Google data center.

Large-scale cluster management at Google with Borg is an excellent source to learn more about the ideas behind Kubernetes.



The Kubernetes Lineage *Chip Childers, Cloud Foundry Foundation*
Source: *The Platform for Forking Cloud Native Applications* presentation

Borg is the inspiration in underlying technologies used in container runtime today and behind the data center systems**.** In 2007 Google contributed **cgroups** to the Linux*. cgroups limits the resources used by collection of processes*. cgroups and Linux namespaces are at the heart of containers today, including Docker.

While Borg was still a secret at Google, Mesos was inspired by discussions with Google. Mesos constructs a multi-level scheduler to more efficient use of a data center cluster.

The Cloud Foundry Foundation embraces the 12 factor application principles that provide guidance to build scalable, cloud ready and automated web applications deployment. Borg and Kubernetes offer these principles as well.

Other Solutions than Kubernetes

Docker Swarm is the solution provided by Docker Inc. It has been re-architected recently and is based on SwarmKit. It is embedded with the Docker Engine.

Apache Mesos is a data center scheduler, which can run containers through the use of *frameworks*. Marathon is the framework that lets you orchestrate containers.

Nomad from HashiCorp, the makers of Vagrant and Consul, is another solution for managing containerized applications. Nomad schedules tasks defined in *Jobs*. It has a Docker driver which lets you define a running container as a task.

Rancher is a container orchestrator-agnostic system, which provides a single pane of glass interface to manage applications. It supports Mesos, Swarm, Kubernetes.