



# Redfish for Operating Systems, Applications, and Containers

DMTF Redfish Forum, Version 0.5

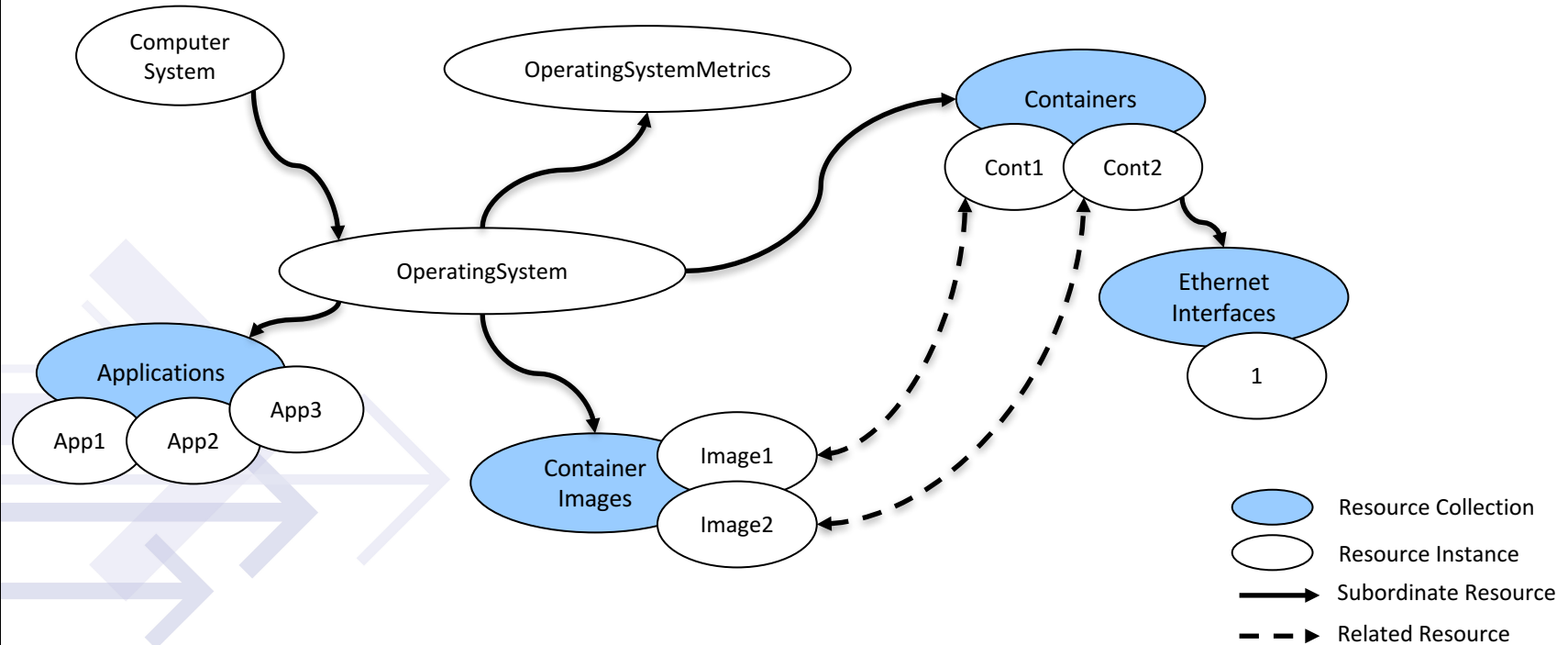
# Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the DMTF website: [www.dmtf.org](http://www.dmtf.org)

# Overview

- Manageability in Redfish has been tied to physical devices in an infrastructure
- The *ComputerSystem* resource shows the logical view of how resources are brought together to form a system, but does not show what is running on the system
- Users have expressed the desire to show software elements running on a system

# Proposed Model Diagram



# Proposed New Resources

- *OperatingSystem*: Contains information about the operating system running on the system, including virtualization engines
- *OperatingSystemMetrics*: Contains metrics for the operating system
- *Application*: Represents a software instance running on a system
- *ContainerImage*: Represents a container image available on a system
- *Container*: Represents a container instance running on a system
  - Contains its own *EthernetInterfaceCollection* to represent network connectivity of the container

# Virtual Machines

- *ComputerSystem* supports a *Virtual* value for *SystemType*
  - Virtual machines are intended to look just like physical systems
- This proposal recommends continuing the pattern where a virtual machine is modeled as a *ComputerSystem* resource
  - Allows for its own set of subordinate resources, such as *OperatingSystem*, *EthernetInterfaceCollections*, and others
  - Allows for existing *Reset* action to be used, with additional *ResetType* values of *Pause*, *Suspend*, and *Resume*
- Additional links required to show the system hosting the virtual machine
  - *HostingComputerSystem*: Shows the system is hosting the virtual machine
  - *VirtualMachines*: Shows the virtual machines hosted by the system

# Network Configuration

- Virtual machines and containers can have various configurations for exposing network interfaces
- Proposal includes the addition of a new *RoutingScope* property to cover the majority of use cases without needing to model virtual switches
- Proposed values
  - External: Interface acts as if it's physically on the external network of the host system
  - Bridge: Interface connects to a bridge interface on the host system
  - Limited: IP translation between the entity and external network is performed by the host system, such as with a NAT
  - **Feedback: Are others needed?**

# Hardware Assignments and Limits

- Virtual machines and containers have configurations for assigning hardware resources from the host system to the entity
  - Processor quantities and limits, memory size, storage, access to other devices
- Proposal includes properties for showing hardware assignments
  - *ComputerSystem* contains *VirtualMachineConfiguration*
  - *Container* contains *Limits*
  - Limited to CPU, memory, and storage control with general quantities
  - **Feedback: Is *VirtualMachineConfiguration* needed at this time? This property would be useful for clients creating virtual machines via Redfish, but most of the information can be gathered via existing properties.**

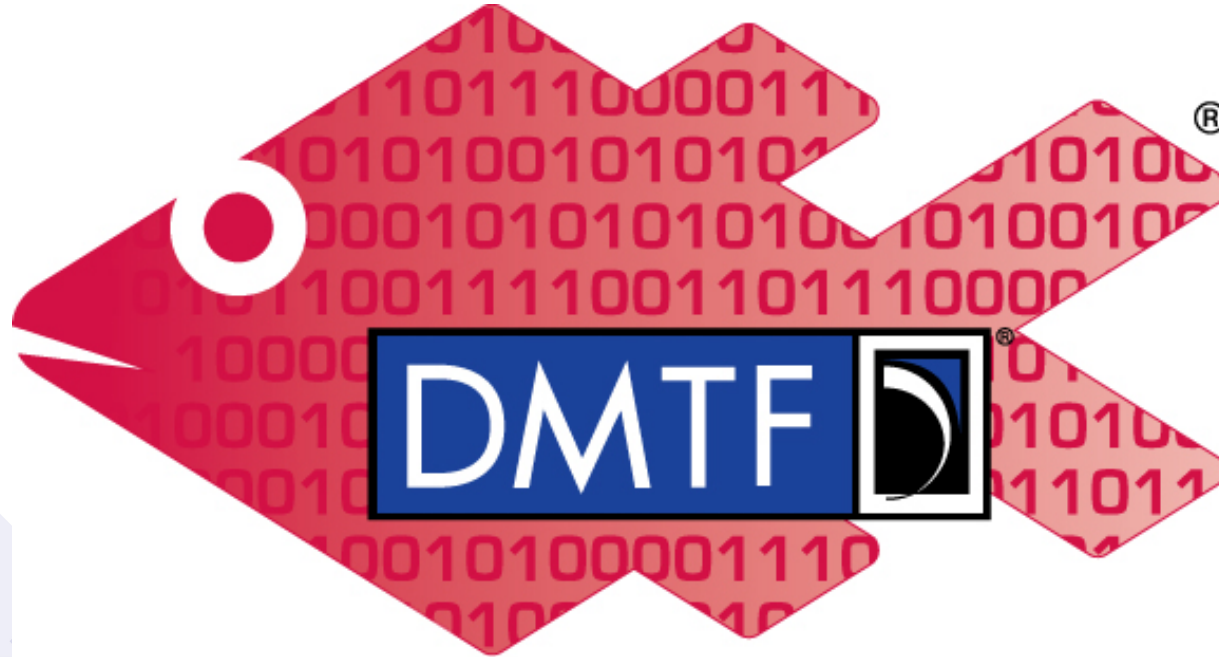


# Hardware Assignments and Limits (cont.)

- Gap: Representing processors in virtual machines and CPU pinning
  - Should the *ProcessorCollection* contain “virtual” CPUs for a virtual machine?
    - If the members reference the host system CPUs, this might change over time based on context switching
    - Showing “virtual” CPUs might be desirable to have the same manageability as a physical system
  - Should links to the host system CPUs in the configuration objects be used for representing affinity?
    - Requires the implementation to model all CPU cores with *SubProcessors*
    - Alternative would be to use identifiers for the CPU that are specific to the host OS
- Gap: Representing memory in virtual machines
  - DIMMs are not necessarily represented in a virtual machine
  - Is it reasonable to omit the *MemoryCollection* in favor of *MemorySummary*?
    - Consistency with physical system is lost with this pattern

# Future Areas to Consider

- NUMA configuration for virtual machines
- Bandwidth control for Ethernet interfaces
- CPU utilization limits
- Assignment of devices beyond CPUs, memory, and storage
  - Examples: GPUs, USB devices



# Redfish