Document Identifier: DSP2063

Date: 2021-12-02

Version: 1.0.0

Redfish SmartNIC White Paper

Supersedes: None

Document Class: Informational

Document Status: Published

Document Language: en-US

CONTENTS

Foreword

The Redfish SmartNIC White Paper was prepared by the Redfish Forum of the DMTF.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about the DMTF, see http://www.dmtf.org.

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

- Jeff Hilland - Hewlett Packard Enterprise
- Michael Raineri - Dell Inc.

1 Introduction

SmartNICs have reached the point of prevalence in the industry where consistent manageability via Redfish is necessary. Sometimes known by other names and classifications, such as DPU, XPU, or IPU, SmartNICs typically represent the fusion of some form of networking functionality with other computational capacity, such as programmable offload processing capability, fixed function offload capacity, artificial intelligence, machine learning, and digital signal processing. These processing functions can be collocated with amounts of storage, either ephemeral or persistent, giving SmartNICs equivalent capability of a traditional computing system. However, they are different from a traditional system in that they augment functionality to another computer system or server, extending the capability and capacity of the system especially in targeted scenarios. Because of this similarity to a traditional computer system, it should be as no surprise that representing them in Redfish is similar to that of a traditional computer system. There are, however, differences and those differences depend on the type of SmartNIC.

For the purposes of Redfish, SmartNICs are broken down into three basic classifications:

- ASIC-based solutions offloading only networking capacity of fixed functions.
- FPGA-based implementations offloading capacity of programmable functions.
- SOC-based offerings offloading processing, (such as a GPU), storage (often NVMe), or other capacity.

These classes of SmartNICs have several things in common. They all have networking functions. Consequently, the networking functions are represented in Redfish using the Advanced Communication Device (ACD) model. They also have some form of physical representation. And the physical aspects are represented using the Redfish chassis model.

- Networking functions are represented in Redfish using the Advanced Communication Device (ACD) model as described in the ACD model section. All SmartNICs must have their networking functions represented.
- Physical aspects are represented using the Redfish chassis model. These requirements can be found in the Chassis model section. All SmartNICs must have their chassis functions represented.
- The fabric model can be used to show additional networking capabilities of the device beyond the ACD model. This includes automatic assignment of IP address values, any IP address values that have been assigned, eBGP controls, and configurable switch properties such as routing. This is described in the Fabric model and SmartNICs section.

From this point, the three classifications of SmartNICs differ in their Redfish representation:

- ASIC-based solutions have no additional requirements since their functionality can be represented using the ACD and chassis models.
- FPGA-based implementations need to show their processing capacity. This is done using the `Processors` collection allowed on the `NetworkAdapter` resource, which was added to the model for this purpose.
- SOC-based solutions, as well as implementations with capacity beyond that described above, use a `ComputerSystem` resource for resource representations as described in the Computer system model section.

2 Chassis model

The `Chassis` resource is used to represent a physical container. While this is typically an enclosure within a rack, the definition is flexible and also represents smaller modules within an enclosure or entire racks that contain sets of enclosures. The `ChassisType` property is used to categorize the type of container the resource represents. Within the `Chassis` resource, the `Contains` and `ContainedBy` properties found inside `Links` are used to express the physical containment relationships between `Chassis` resource instances.

SmartNICs are complex devices with their own sets of components, sensors, power, and thermal information. A `Chassis` resource is used to model the SmartNIC in order to represent the physical boundaries of the device and encapsulate the underlying components that support the SmartNIC's functionality.

The following properties in a `Chassis` resource that represents a SmartNIC exhibit the following patterns:

- `ChassisType` should contain the value `Card`. Depending on the underlying architecture of the SmartNIC, other values, such as `Module` and `Zone` might apply.
- `NetworkAdapters` contains a link to the collection of network adapters found in the SmartNIC. This is described further in the ACD model section.
- `ContainedBy` within `Links` contains a reference to the `Chassis` resource that represents the container of the SmartNIC. Likewise, the `Contains` property within `Links` of the referenced `Chassis` resource will reference the `Chassis` resource for the SmartNIC.
- `ComputerSystems` within `Links` contains a reference to the `ComputerSystem` resources that represent the systems that perform offload computation for the SmartNIC. Depending on the type of SmartNIC, this property might not be present. This is described further in the Computer system model section.

Other properties in a `Chassis` resource that represents a SmartNIC might be implemented depending on the capabilities of the SmartNIC.

Figure 1 shows the relationship diagram for the chassis model. The `Chassis` resource `Card1` represents the SmartNIC and the `Chassis` resource `1U` represents the container of the SmartNIC.

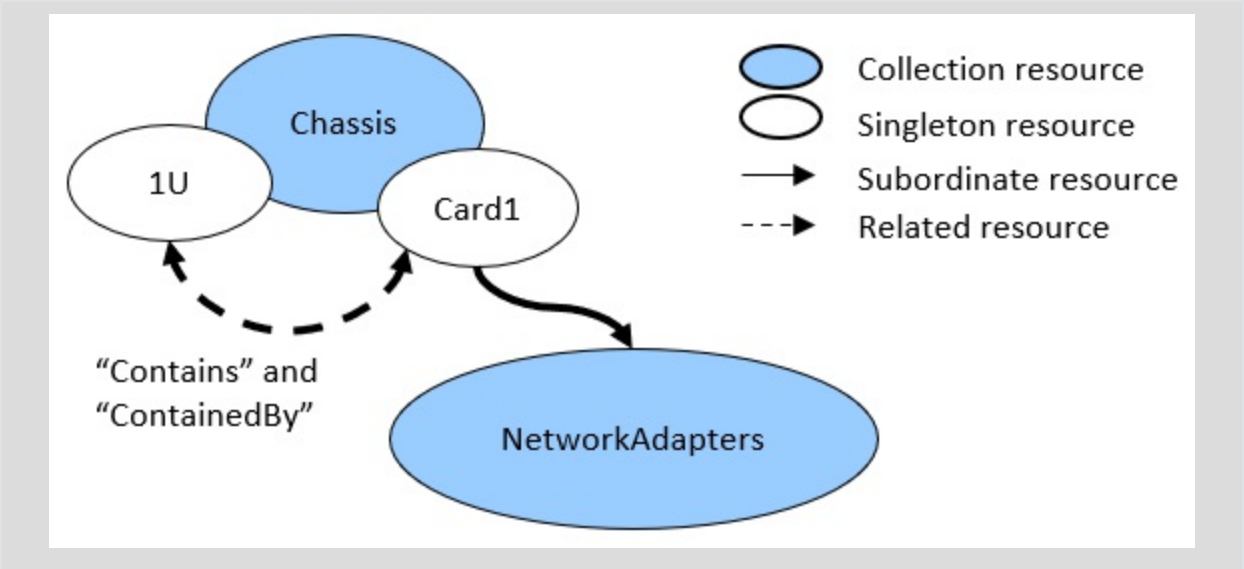

Figure 1

Example `Chassis` resource that represents a SmartNIC:


```
{
    "@odata.id": "/redfish/v1/Chassis/Card1",
    "@odata.type": "#Chassis.v1_17_0.Chassis",
    "Id": "Card1",
    "Name": "Card Chassis",
    "ChassisType": "Card",
    "AssetTag": "Chicago-45Z-2467",
    "Manufacturer": "Contoso",
    "Model": "9900",
    "SKU": "9541236",
    "SerialNumber": "637XR7568R4",
    "PartNumber": "334587-S34",
    "PowerState": "On",
    "IndicatorLED": "Off",
    "Location": {
        "PartLocation": {
            "ServiceLabel": "Card Slot 1",
            "LocationType": "Slot",
            "LocationOrdinalValue": 0,
            "Reference": "Rear",
            "Orientation": "LeftToRight"
        }
    },
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
```

```
    "ThermalSubsystem": {
        "@odata.id": "/redfish/v1/Chassis/Card1/ThermalSubsystem"
    },
    "EnvironmentMetrics": {
        "@odata.id": "/redfish/v1/Chassis/Card1/EnvironmentMetrics"
    },
    "NetworkAdapters": {
        "@odata.id": "/redfish/v1/Chassis/Card1/NetworkAdapters"
    },
    "Sensors": {
        "@odata.id": "/redfish/v1/Chassis/Card1/Sensors"
    },
    "Drives": {
        "@odata.id": "/redfish/v1/Chassis/Card1/Drives"
    },
    "Links": {
        "ComputerSystems": [
            {
                "@odata.id": "/redfish/v1/Systems/Card1-SoC"
            }
        ],
        "ManagedBy": [
            {
                "@odata.id": "/redfish/v1/Managers/BMC"
            }
        ],
        "ContainedBy": {
            "@odata.id": "/redfish/v1/Chassis/1U"
        }
    }
}
```

3 ACD model

The Advanced Communication Device (ACD) model is comprised of several resources contained within a `Chassis` resource.

- `NetworkAdapter` - Represents the container of components, such as controllers, functions, and ports, for a network device.
- `NetworkDeviceFunction` - A single network function part of an overall network device.
- `Port` - A physical port on a network device.
- `Processor` - A dedicated offload processor for a network device, such as an FPGA or GPU. This type of component is not applicable to all network devices.
- `AllowDeny` - Access policies for a network function. Not all network devices support direct configuration of access policies in their network functions.
- `NetworkInterface` - Provides the relationships from the system consuming network resources to the resources that represent the network functions.

The `NetworkAdapter`, `NetworkDeviceFunction`, `Port`, and `NetworkInterface` resources are used for all SmartNICs. The `AllowDeny` resource can also be used for all SmartNICs if the underlying device supports configuring access policies.

SoC-based SmartNICs contain additional linkage from the ACD model to the `ComputerSystem` resources that represent the systems used for offload computation. This link is provided in the `NetworkDeviceFunction` resource with the `OffloadSystem` property found inside of `Links`. The usage of a system for this type of SmartNIC is described further in the Computer system model section. Figure 2 shows the relationship diagram for the ACD model for an SoC-based SmartNIC.

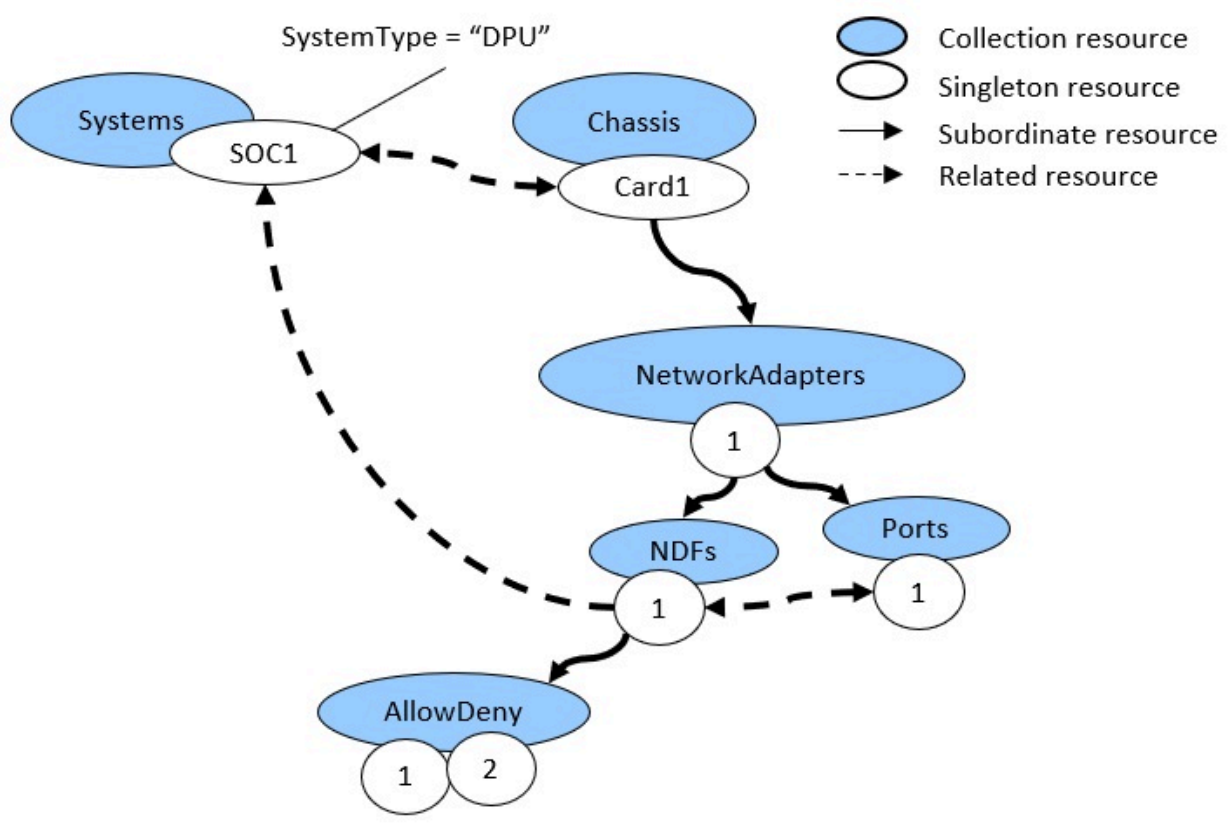

Figure 2

FPGA-based SmartNICs use the `Processor` resources to represent the offload FPGAs for the SmartNIC. The `NetworkAdapter` resource contains a `Processors` property for representing the set of offload FPGAs, or other types of processors. The `NetworkDeviceFunction` resource contains the `OffloadProcessors` property within `Links` to so which processors are performing offload computation for the network function. Conversely, the `Processor` resource contains the `NetworkDeviceFunctions` property within `Links` to show which network functions to which the processor is performing offload computation. Figure 3 shows the relationship diagram for the ACD model for an FPGA-based SmartNIC.

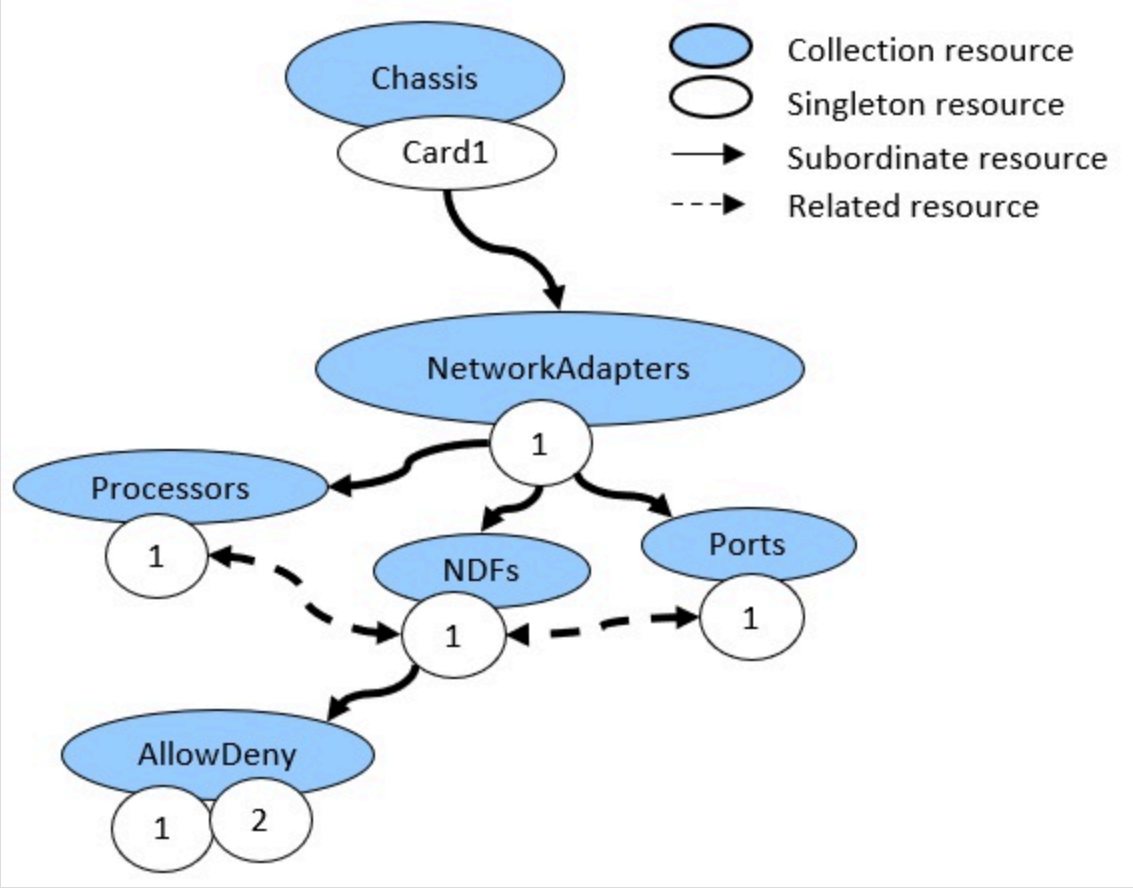
Chassis
Card1
Collection resource
Singleton resource
Subordinate resource
Related resource
NetworkAdapters
1
Processors
1
NDFs
1
Ports
1
AllowDeny
1
2

Figure 3

4 Fabric model and SmartNICs

The fabric model in Redfish is used to represent several aspects of SmartNICs. Figure 4 shows the relationships to the Redfish fabrics model with the relevant resources in the ACD model.

SmartNICs often have properties similar to switches, such as when it has multiple internal virtual functions mapped to a single egress port. In this case, there is the functionality of a switch inside of the SmartNIC. This often means that switch can be configured with eBGP settings to represent underlay and overlay networks. However, it isn't necessary to model the switch in Redfish with the `Switch` resource. Instead, the functionality can be modeled with the `AddressPool` resource, which contains the relevant eBGP properties, such as CIDR and neighbor related functions.

SmartNICs will also often be assigned an IP address range and perhaps other protocol related settings related to DNS, DHCP, gateway, NTP and similar network functions. These properties and settings are also represented in the `AddressPool` resource. By associating the virtual functions, represented as `NetworkDeviceFunction` resources, of a SmartNIC with the `Endpoint` resources, which are used to represent the IP settings for the virtual functions, and by associating those `Endpoint` resources with an `AddressPool` resource, clients can determine the IP addresses for each virtual function. The implementation should use unique IP addresses and other settings from the address pool and assign the values in each endpoint.

Redfish `Zone` resources can be used to represent subnets as well as which endpoints have allowed routes between them. Since endpoints belonging in the same zone are assumed to have routable fabric between any other endpoint in that zone, clients can use multiple `Zone` resources to represent allowable traffic patterns. Note that the `AllowDeny` resources found in the `NetworkDeviceFunction` resources may affect what is allowed to be sent or received, but that is separate and distinct from whether the traffic routing exists.

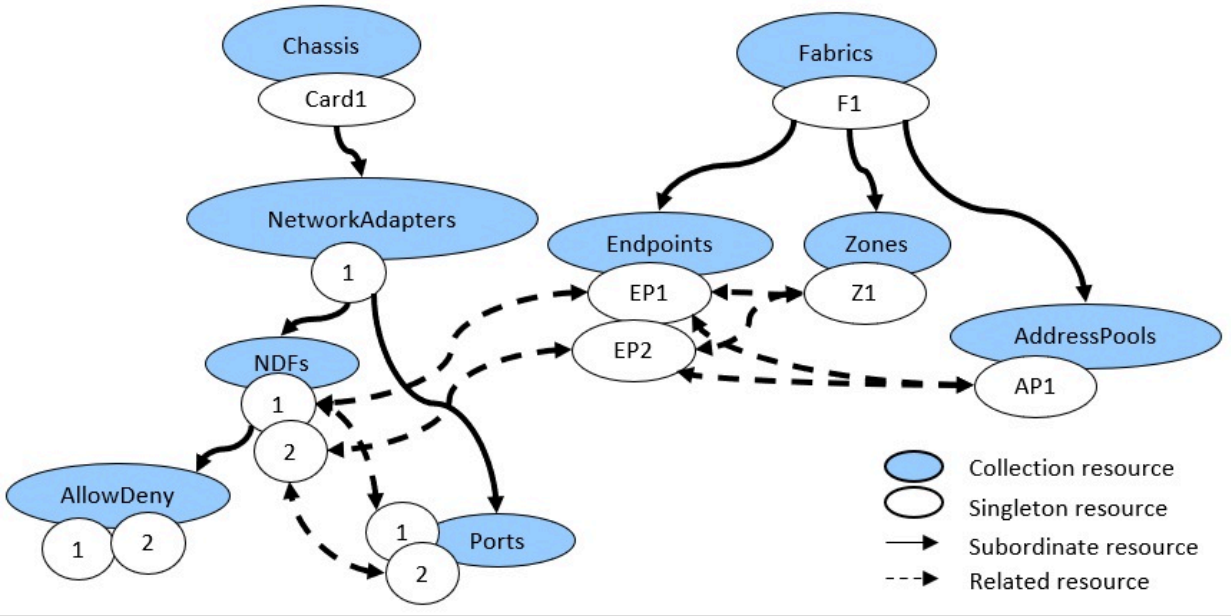
Chassis
Card1
NetworkAdapters
1
NDFs
1
2
AllowDeny
1
2
1
2
Ports
Fabrics
F1
Endpoints
EP1
EP2
Zones
Z1
AddressPools
AP1
Collection resource
Singleton resource
Subordinate resource
Related resource

Figure 4

5 Computer system model

If the SmartNIC to be represented contains resources beyond those described previously, such as a GPU, memory, or storage, those resources are represented using the `ComputerSystem` resource, where the `SytemType` property contains the value `DPU`. Since the `ComputerSystem` resource already contains references to collections of processors, memory, storage, and other components, implementations can leverage these definitions to represent the additional capabilities of the SmartNIC. Figure 5 shows an example SmartNIC with its system representation.

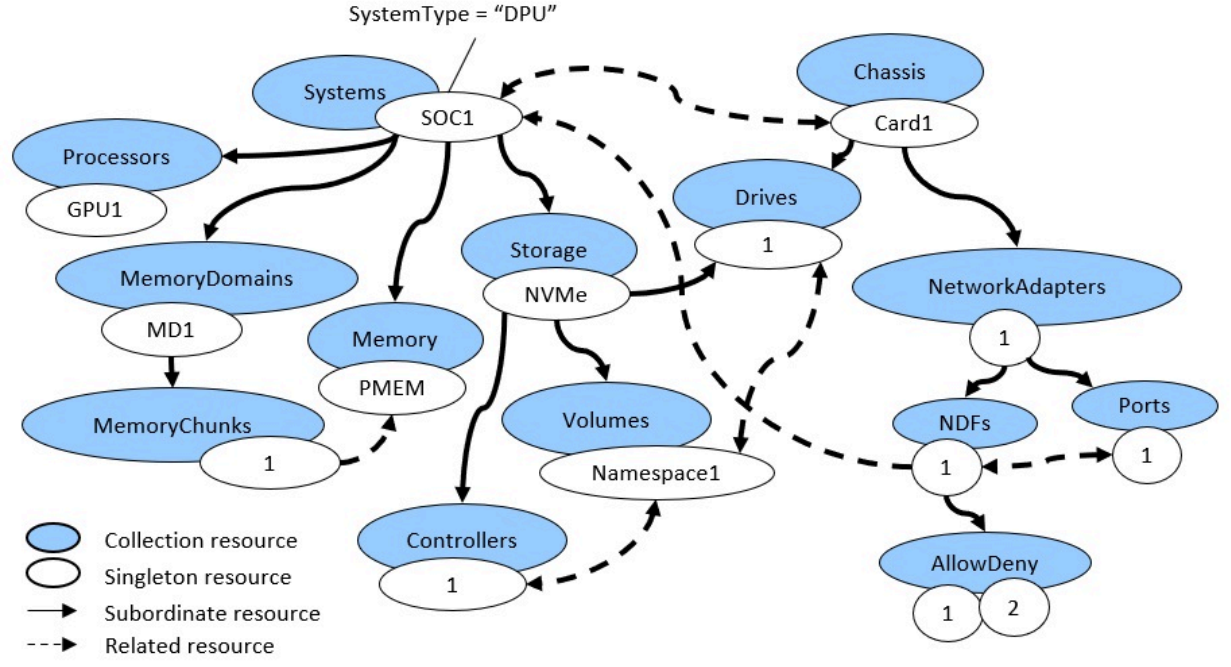

Figure 5

6 Appendix A: References

- "SmartNICs" Mockup: http://redfish.dmtf.org/redfish/v1

7 Appendix B: Change log

Version	Date	Description
1.0.0	2021-12-02	Initial release.