Introduction

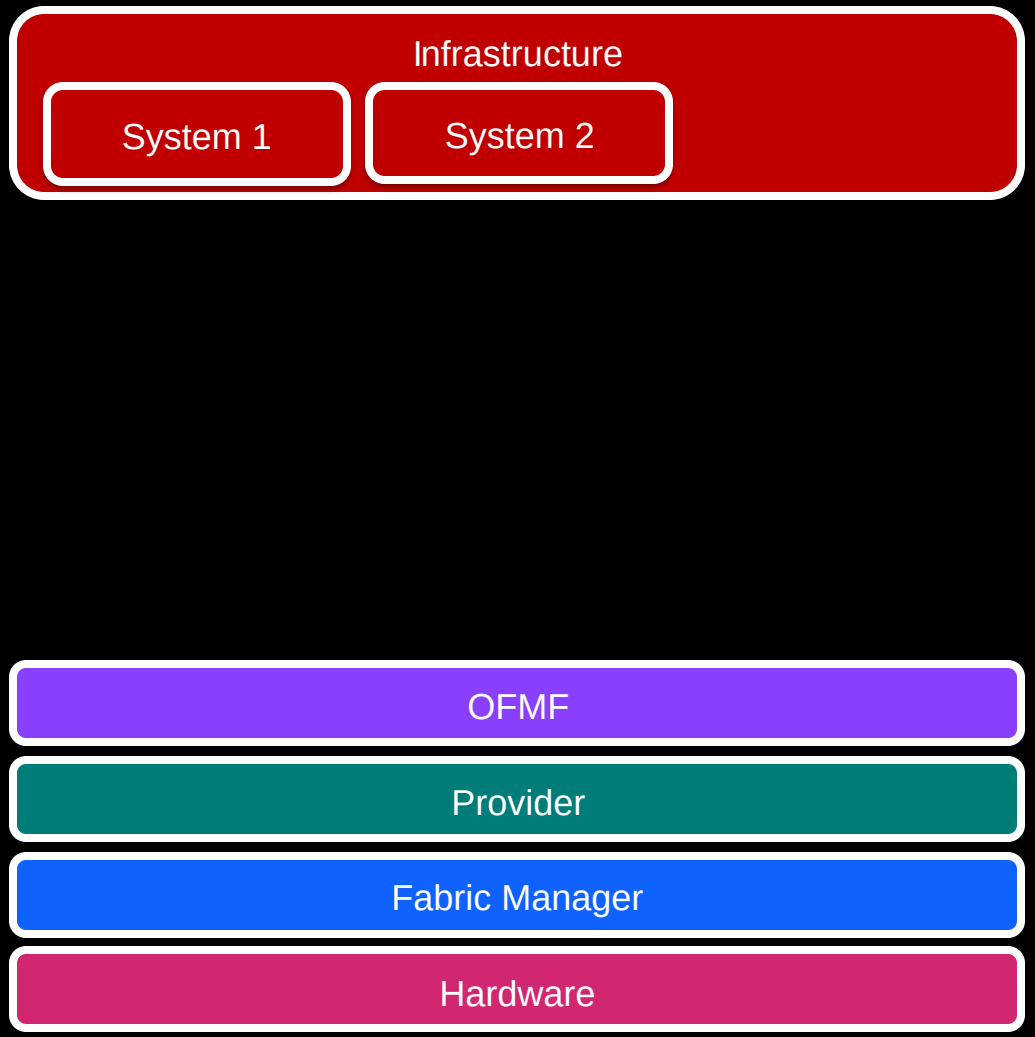
Christian Pinto
Research Scientist
IBM
Location: IBM Research Europe – Dublin

Michele Gazzetti
Research Software Engineer
IBM
Location: IBM Research Europe – Dublin

Motivation

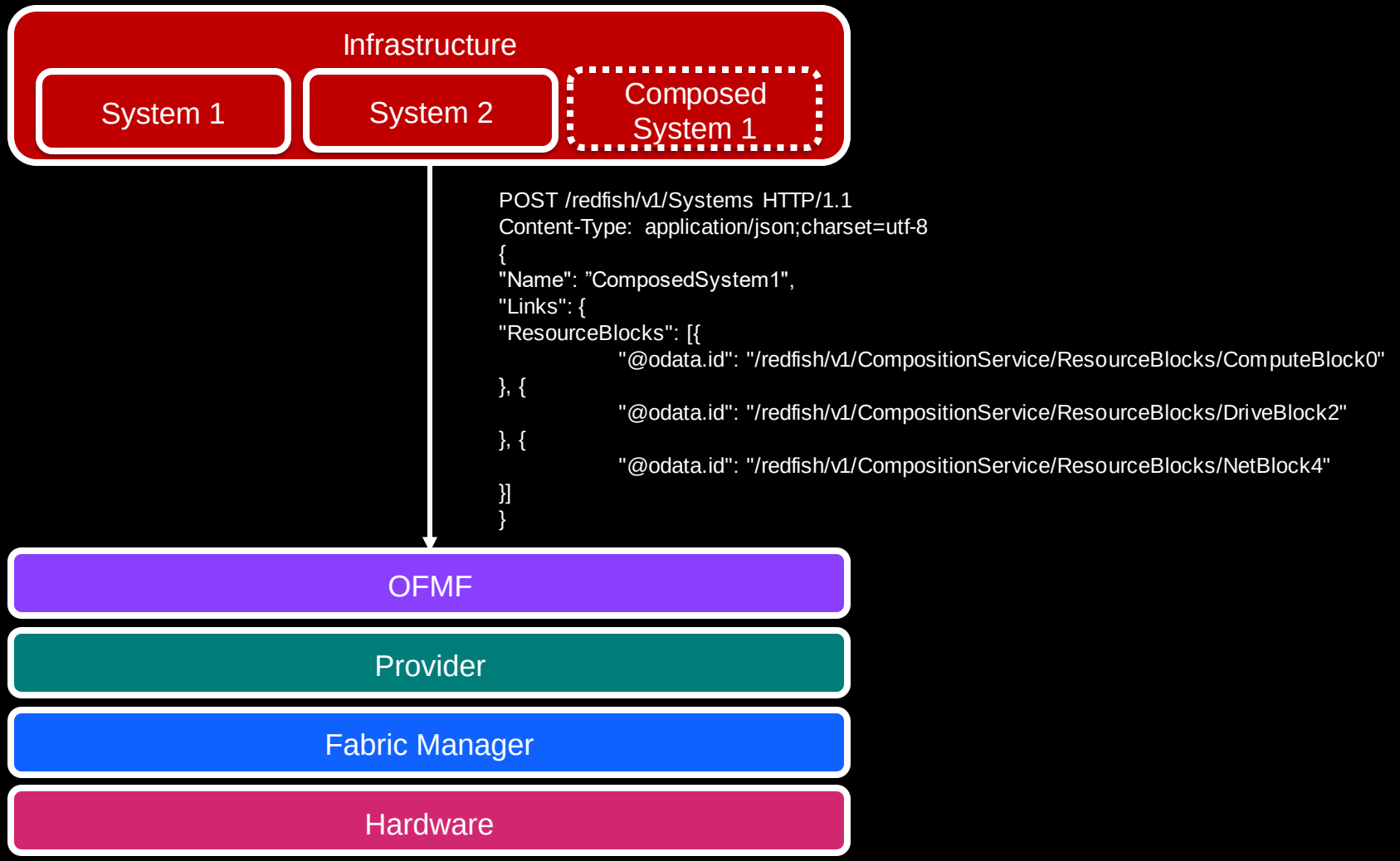
- Fabric attached devices (like GenZ or CXL) can be managed by the OFMF via the appropriate Provider/Agent.
- At the same time, such devices can be seen as composable resources.
- The Redfish specification defines already mechanisms to compose Systems via:
  - Specific Composition requests: list of ResourceBlocks.
  - Constrained Composition Requests: specify characteristics of components
- In some cases, details of the fabric could be abstracted to hide the complexity of managing underlining resources, and for security.
  - In this scenario Constrained Composition Requests allow clients (platforms/infrastructure providers) to leverage composability without knowledge of underlining fabric.

Scenario 1: Composability via Specific Composition Requests

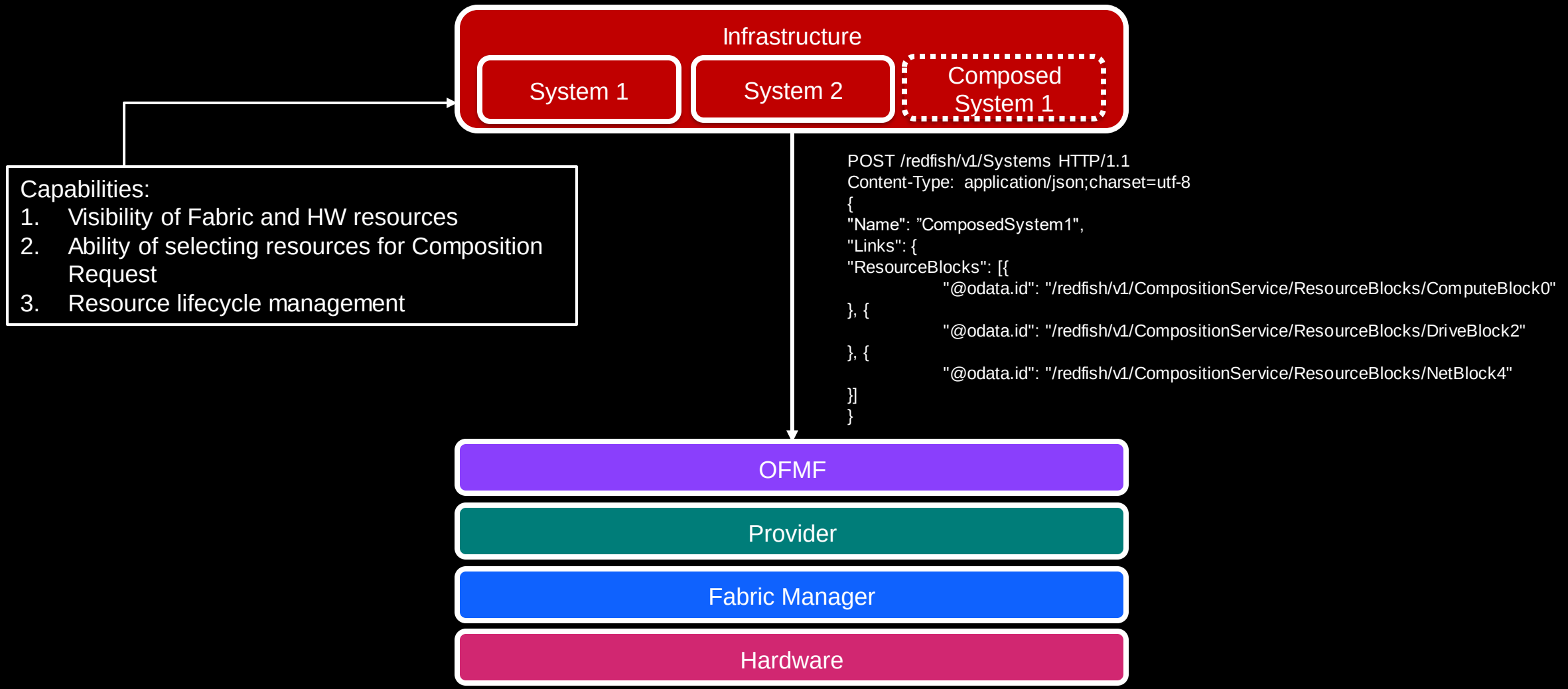


Specific Composition Request: https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.15.1.pdf

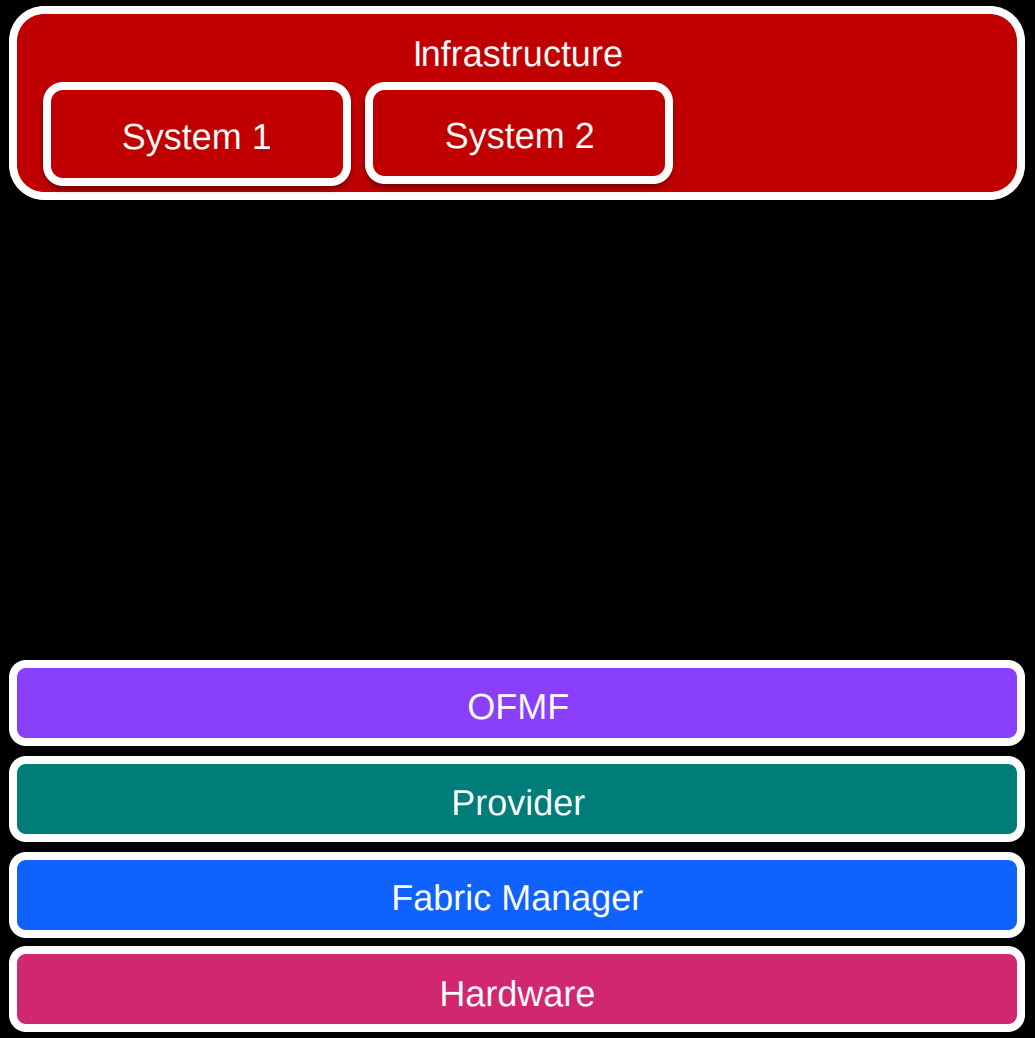
Scenario 1: Composability via Specific Composition Requests



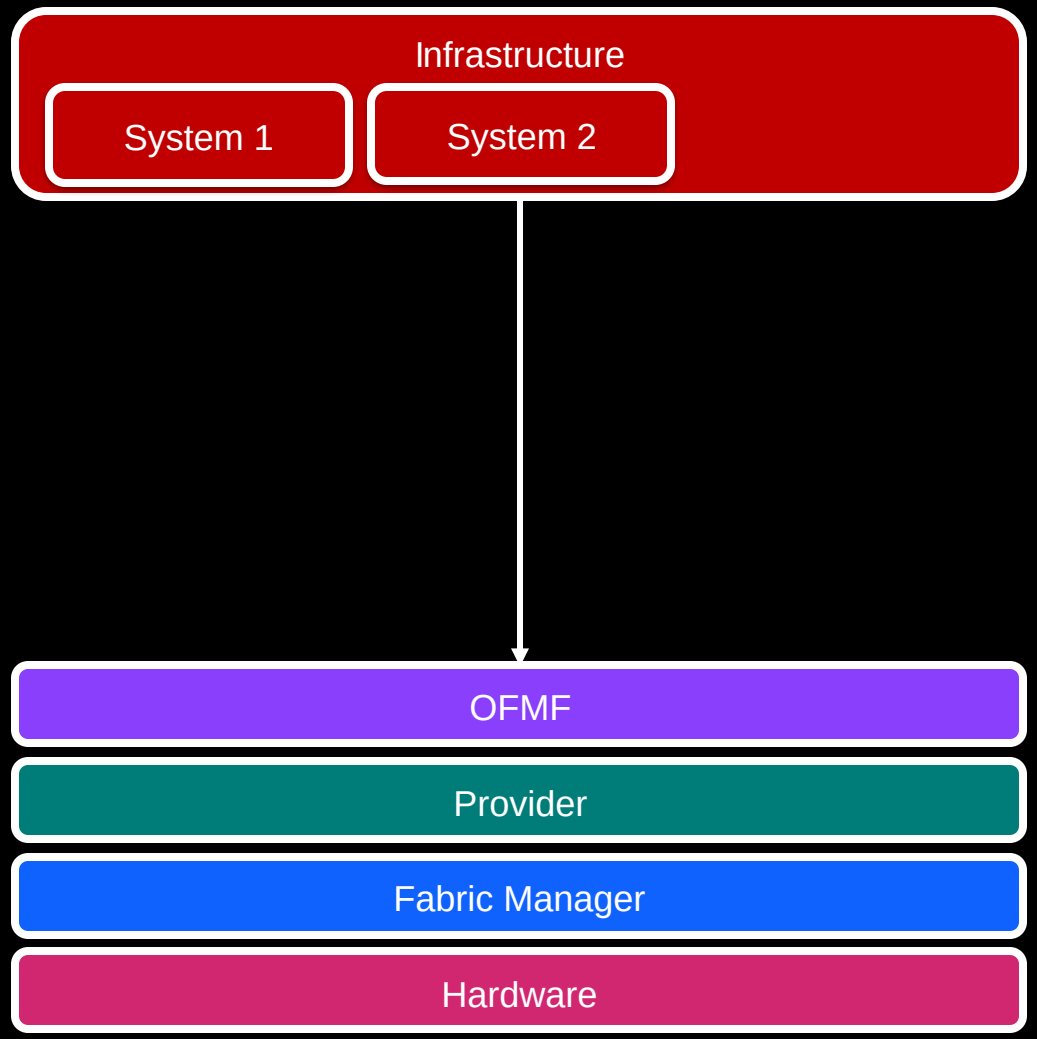
Scenario 1: Composability via Specific Composition Requests



Scenario 2: Composability via Constrained Composition Requests

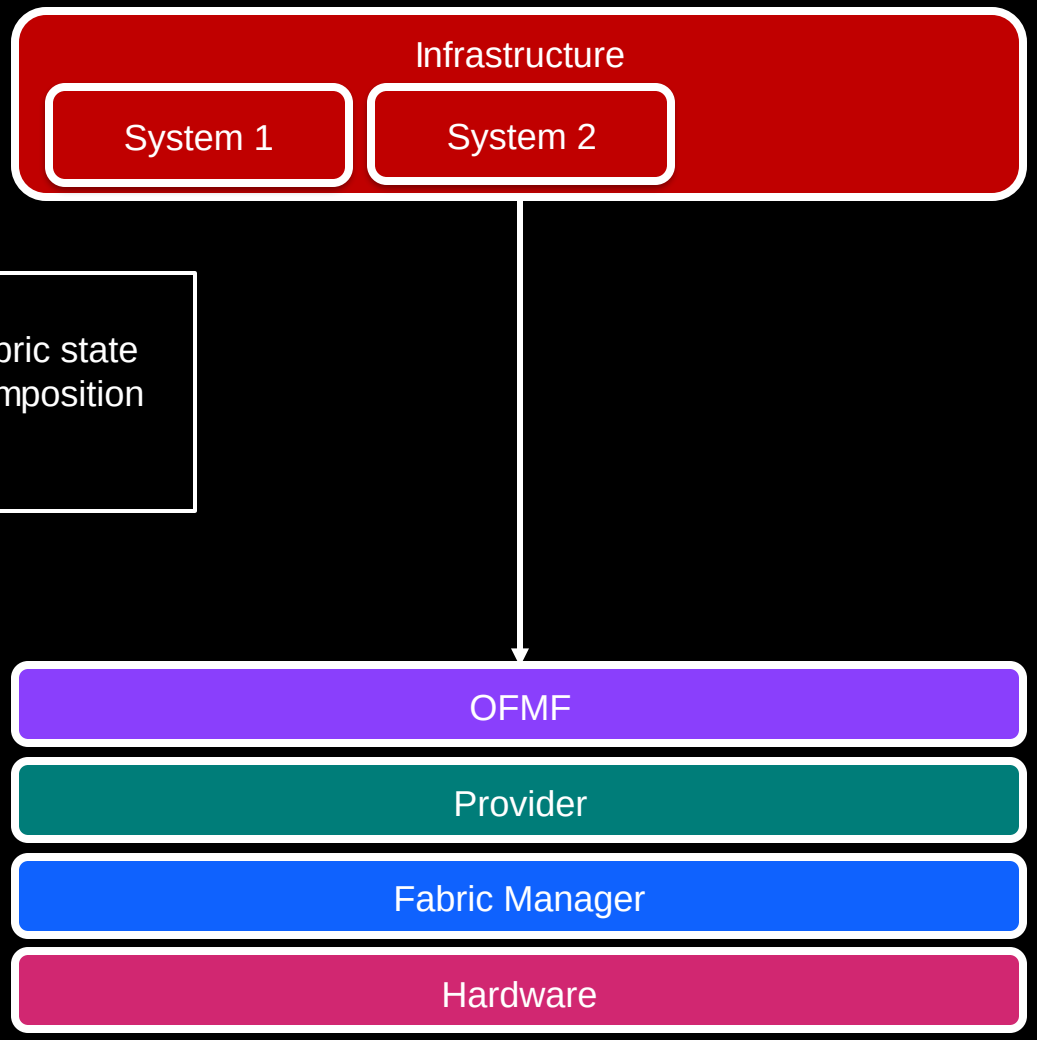


Scenario 2: Composability via Constrained Composition Requests




```
POST /redfish/v1/Systems HTTP/1.1
{
"Name": " ComposedSystem1",
"PowerState": "On",
"Processors": {…},
"Memory": {
"Members": [{
                "@Redfish.RequestedCount":  4,
                "CapacityMiB": 8192,
                "MemoryType": "DRAM",
                "MemoryDeviceType":  "DDR4"
                }]
},
"SimpleStorage": {…}
```

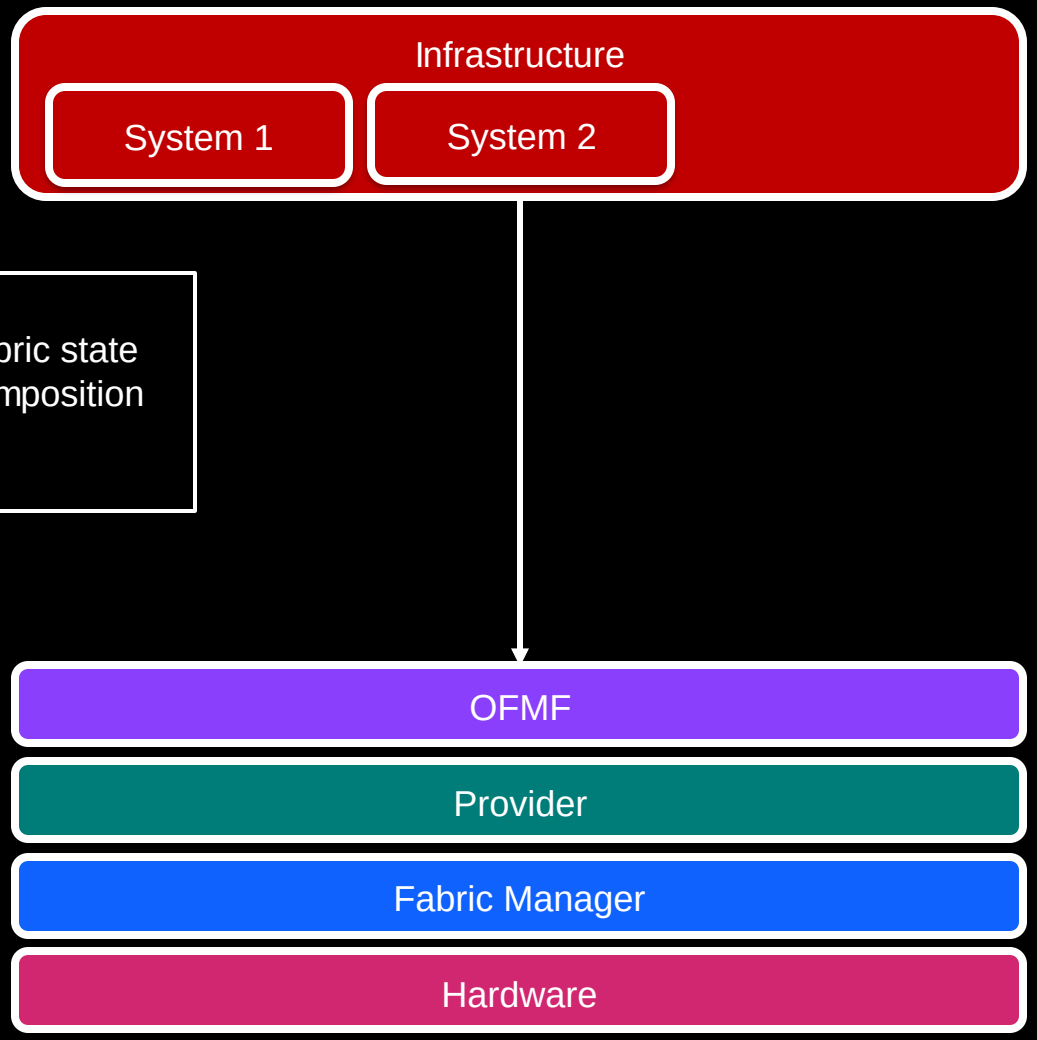
Scenario 2: Composability via Constrained Composition Requests



Capabilities:
1. Visibility of all HW resources and fabric state
2. Ability of selecting resources for Composition Request
3. Resource lifecycle management

```
POST /redfish/v1/Systems HTTP/1.1
{
"Name": " ComposedSystem1",
"PowerState": "On",
"Processors": {…},
"Memory": {
"Members": [{
                "@Redfish.RequestedCount":  4,
                "CapacityMiB": 8192,
                "MemoryType": "DRAM",
                "MemoryDeviceType":  "DDR4"
                }]
},
"SimpleStorage": {…}
```

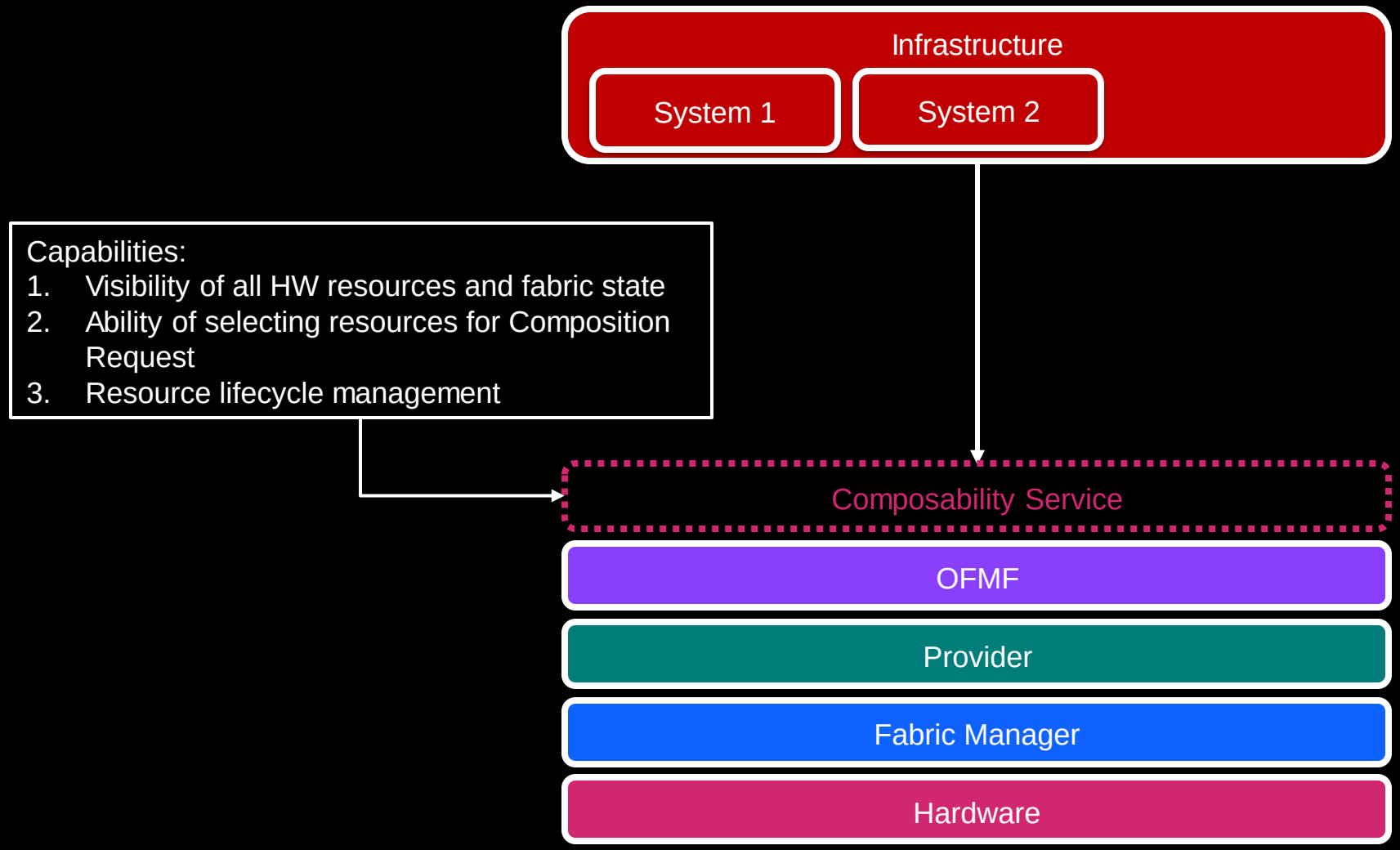
Scenario 2: Composability via Constrained Composition Requests



Capabilities:
1. Visibility of all HW resources and fabric state
2. Ability of selecting resources for Composition Request
3. Resource lifecycle management

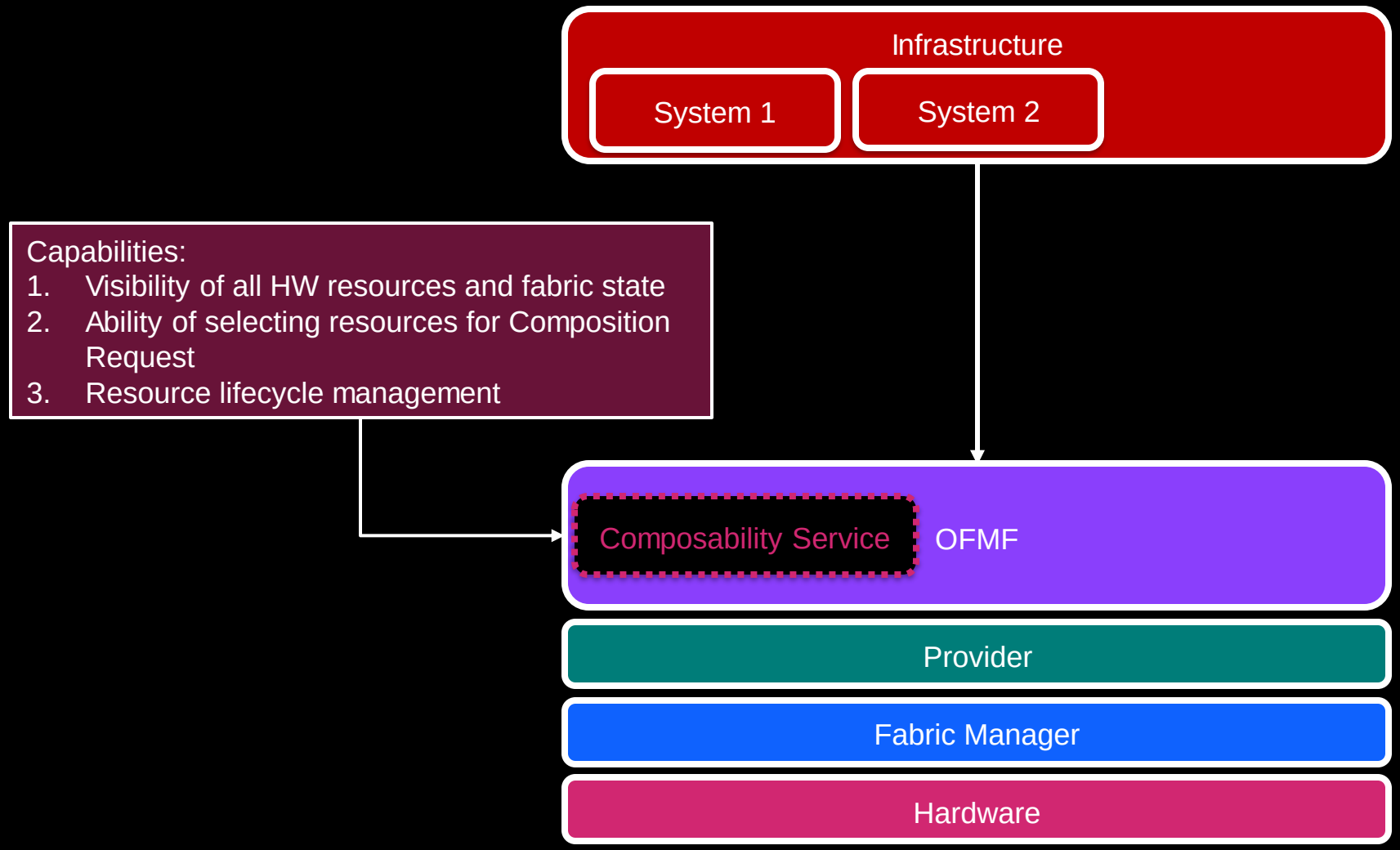
```
POST /redfish/v1/Systems HTTP/1.1
{
"Name": " ComposedSystem1",
"PowerState": "On",
"Processors": {…},
"Memory": {
"Members": [{
                "@Redfish.RequestedCount":  4,
                "CapacityMiB": 8192,
                "MemoryType": "DRAM",
                "MemoryDeviceType":  "DDR4"
                }]
},
"SimpleStorage": {…}
```

Scenario 2: Composability via Constrained Composition Requests




```
POST /redfish/v1/Systems HTTP/1.1
{
"Name": " ComposedSystem1",
"PowerState": "On",
"Processors": {…},
"Memory": {
"Members": [{
                "@Redfish.RequestedCount":  4,
                "CapacityMiB": 8192,
                "MemoryType": "DRAM",
                "MemoryDeviceType":  "DDR4"
                }]
},
"SimpleStorage": {…}
```

Scenario 2: Composability via Constrained Composition Requests




```
POST /redfish/v1/Systems HTTP/1.1
{
"Name": " ComposedSystem1",
"PowerState": "On",
"Processors": {…},
"Memory": {
"Members": [{
                "@Redfish.RequestedCount":  4,
                "CapacityMiB": 8192,
                "MemoryType": "DRAM",
                "MemoryDeviceType":  "DDR4"
                }]
},
"SimpleStorage": {…}
```

Main interests

Composability Service (Part of OFMF or client?):

1. Resource Blocks and Systems lifecycle management
2. Integration of Composition Policies in current OFMF to support Constrained Composition Requests
   (NOTE: policies should be pluggable and selected by the user depending on their context)

Translating Composition Request into actions necessary to setup/tear-down a Composed System

RAS and Security