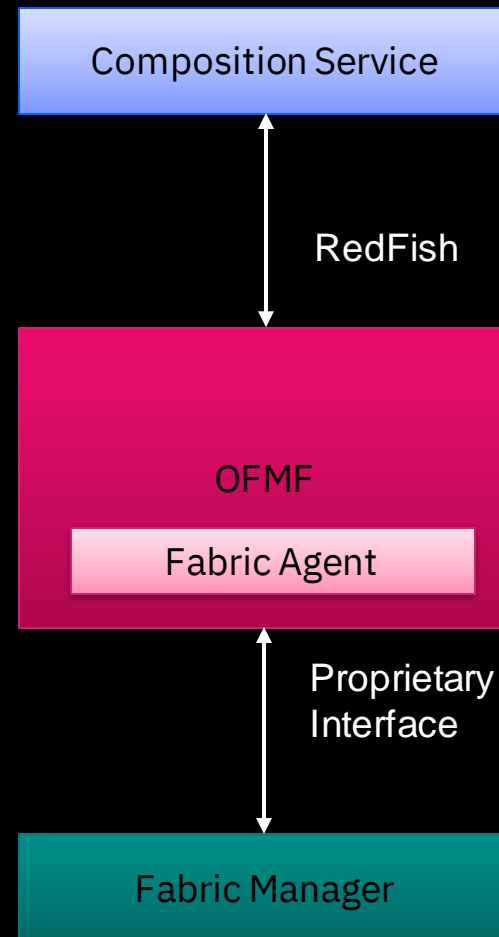


# OFMF system composition via RedFish

Christian Pinto  
Michele Gazzetti

IBM Research Europe - Dublin

# Current SW stack template

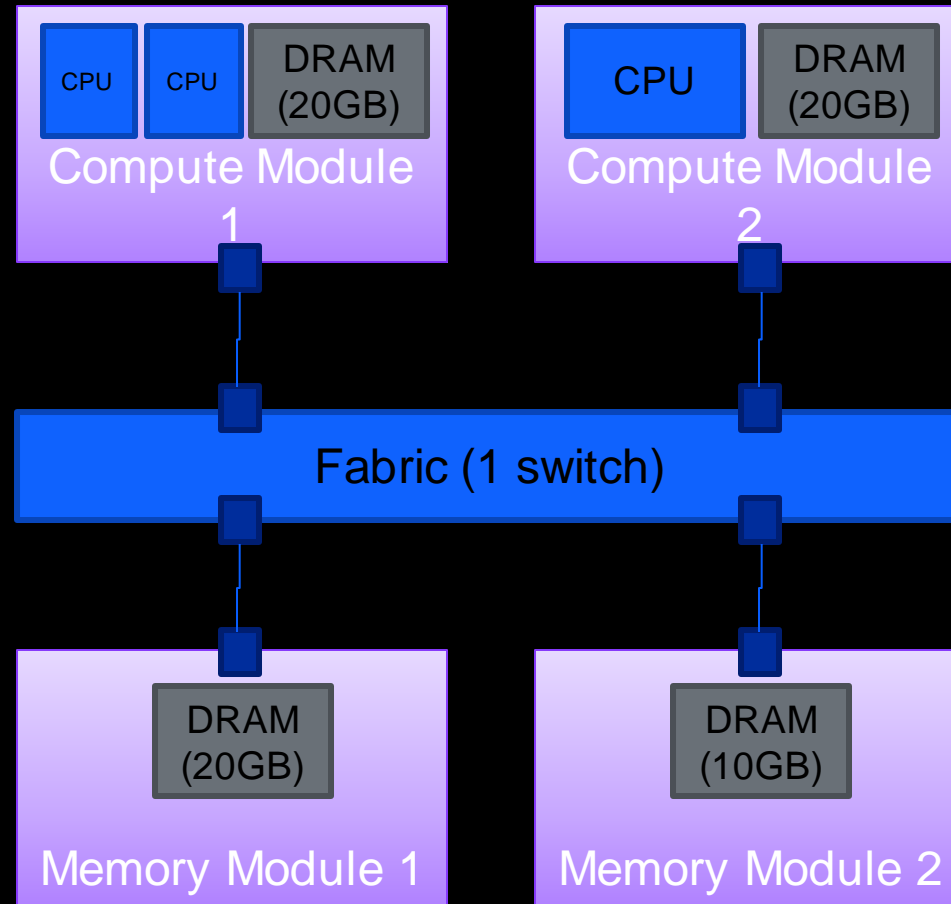


- Groups resources in ResourceBlocks
- Handles composing Computer Systems out of existing hardware blocks
- Client of the OFMF

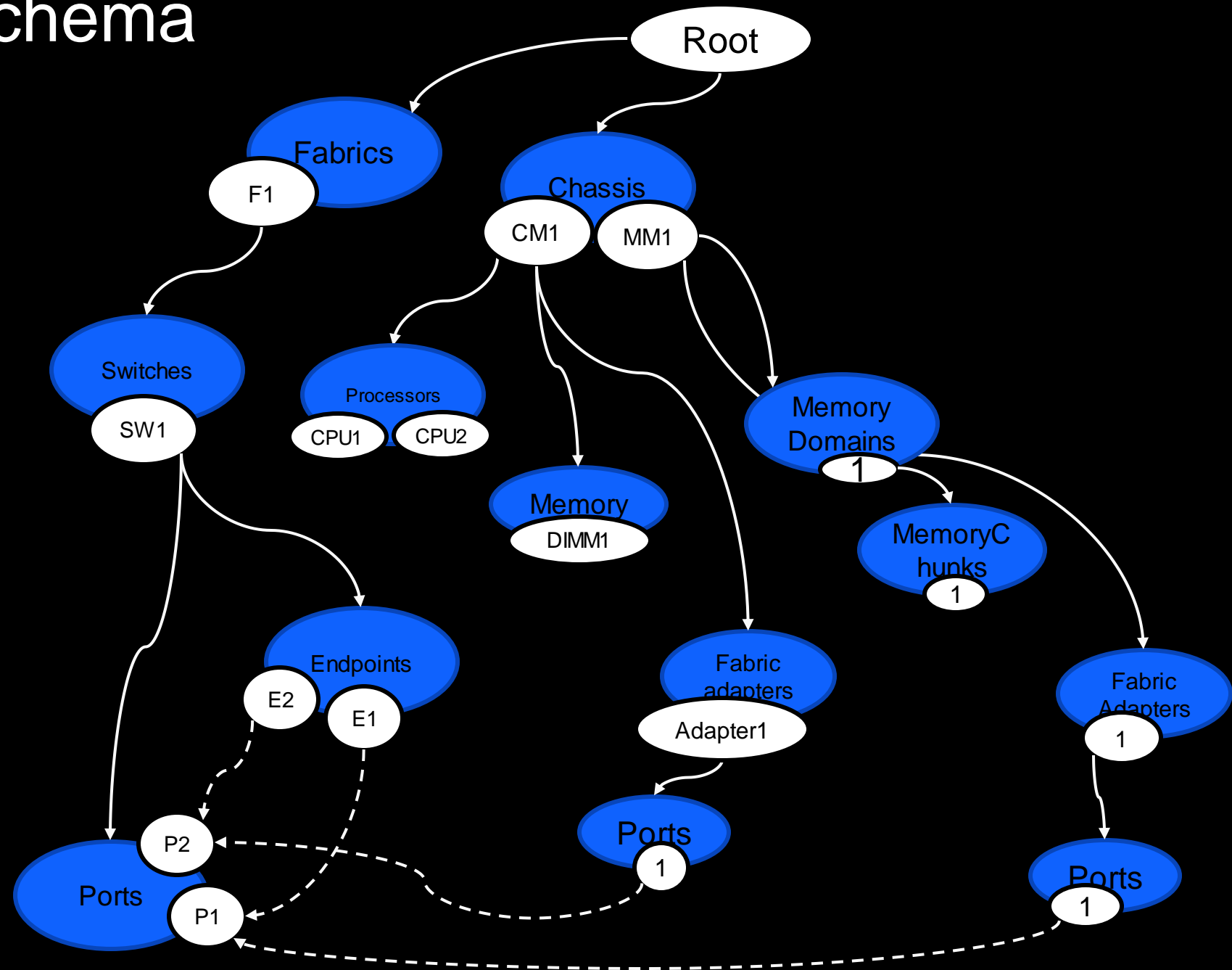
- Keeps full hardware inventory
- Handles interactions with Fabric via Agents
- No Composability Awareness

- Interacts with Fabric

# Example system template



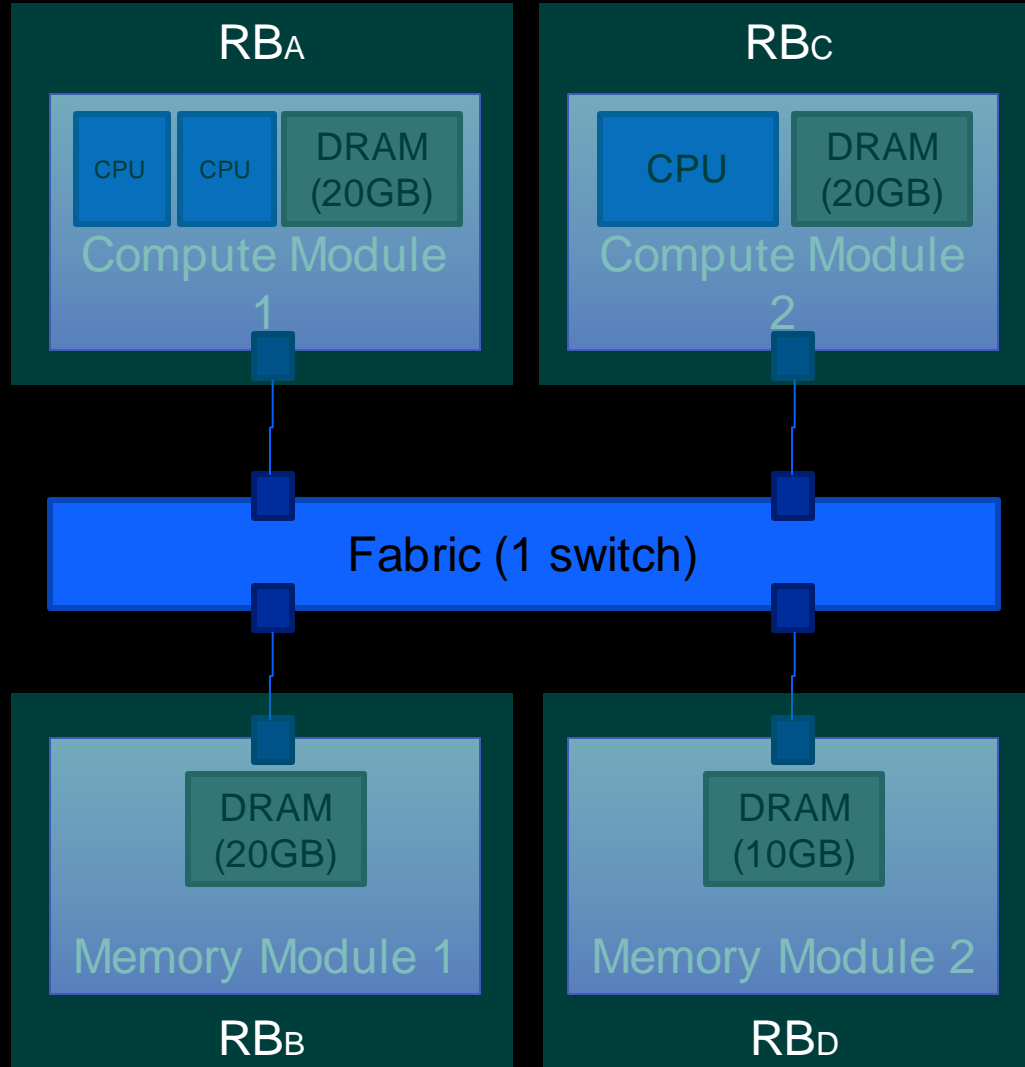
# OFMF RedFish schema



No Computer Systems are available at t0. Computer Systems are created upon the successful completion of a composition request.

# Example system - ResourceBlocks

The composition service will group hardware resources in ResourceBlocks that are the units for composing Computer Systems.



# Assumptions in the creation of ResourceBlocks

For the sake of this use-case take the following assumptions **(to be discussed)**:

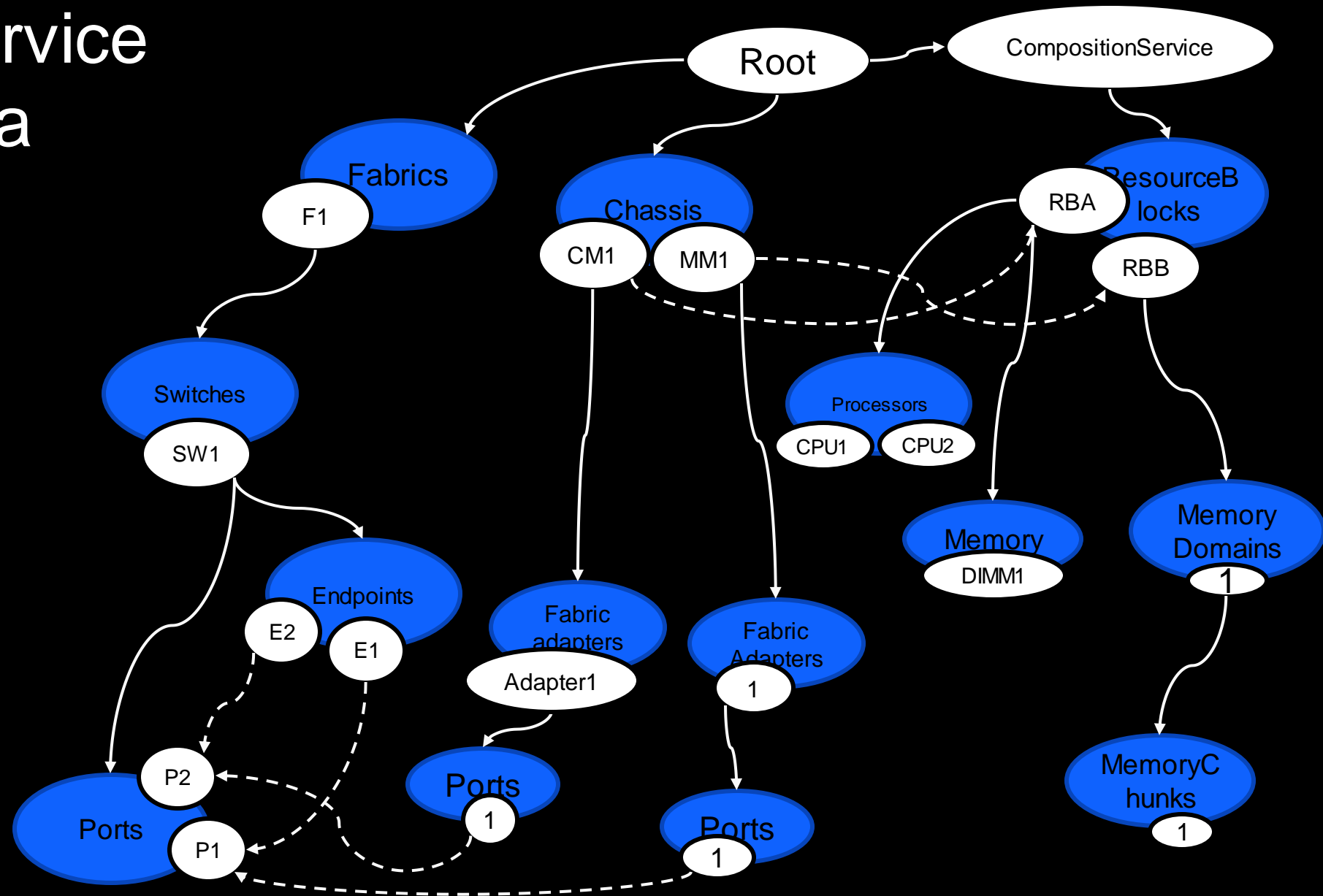
1. A **Chassis** containing CPUs, local memory and a connection to the fabric a **FabricAdapter** can be represented as **ResourceBlock**
2. A **MemoryChunk** belonging to a **MemoryDomain** connected to the fabric via a **FabricAdapter** can be represented as a **ResourceBlock**
3. Other kind of resources (**Accelerators**) connected to the **Fabric** via a **Fabric Adapter** can be represented as a **ResourceBlock**

**Note:** The list of assumption above is far from exhaustive and we should explore more complex and dynamic cases to see if they hold.

# Composition Service RedFish schema

The composition Service needs to:

- Read original RedFish tree from OFMF
- Group resources part of the same composition unit into Resource blocks.
  - Possible multiple RB per Chassis.
  - How does the CS know how to divide resources within the same Chassis in RBs?
- Rebuild the tree to have resources belonging to RBs and not Chassis.
- Do we need to have the Fabric information replicated too?



# Example Composition Request

Request sent to the Composition service as:

```
POST /redfish/v1/Systems HTTP/1.1
```

```
{
  "Processors": {
    "Members": [
      {
        "@Redfish.RequestedCount": 2,
        "ProcessorType": "CPU",
        "TotalCores": 8
      }
    ]
  },
  "Memory": {
    "Members": [
      {
        "@Redfish.RequestedCount": 2,
        "MemoryType": "DRAM",
        "CapacityMiB": 20480
      }
    ]
  }
}
```

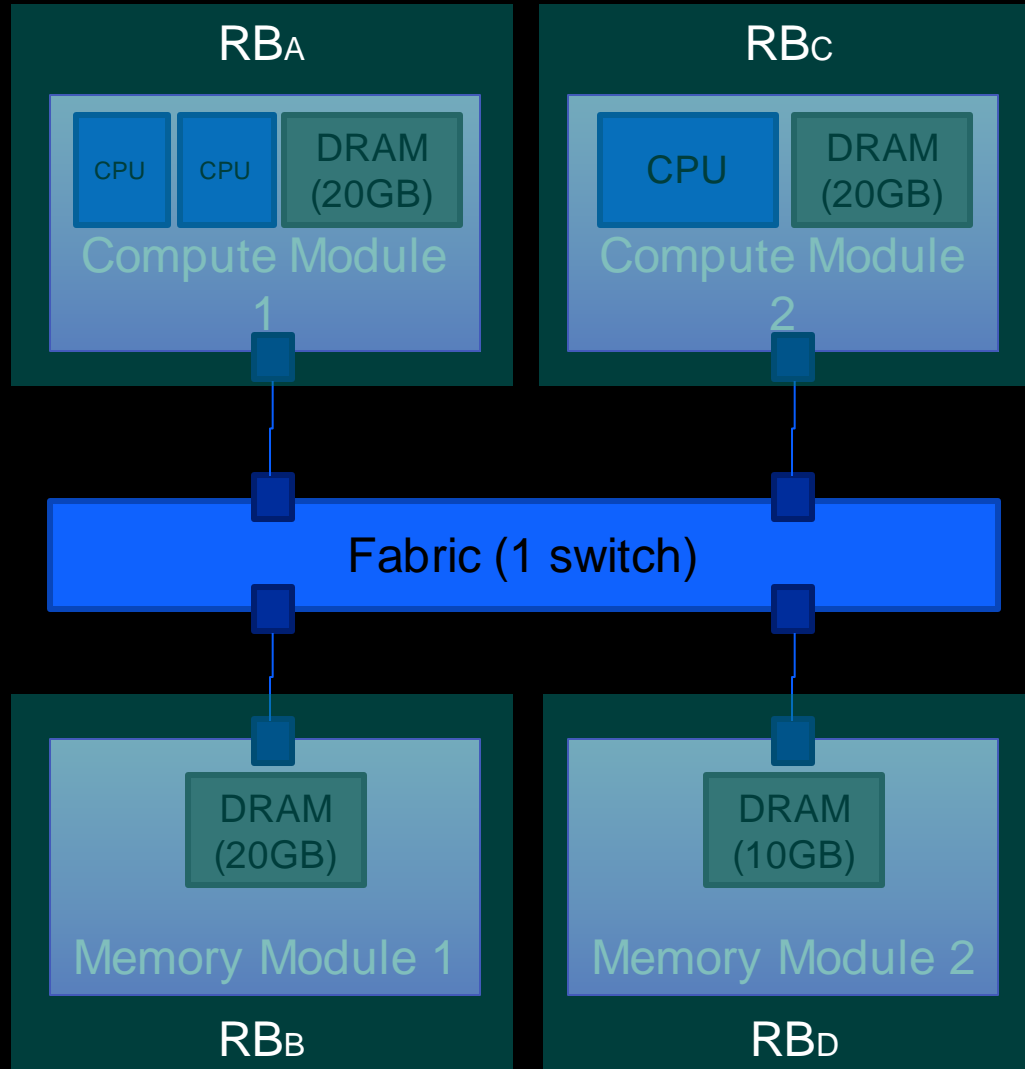
New system  
specification:

2 x 4 Cores CPUs

2 x 20GB DRAM  
Modules



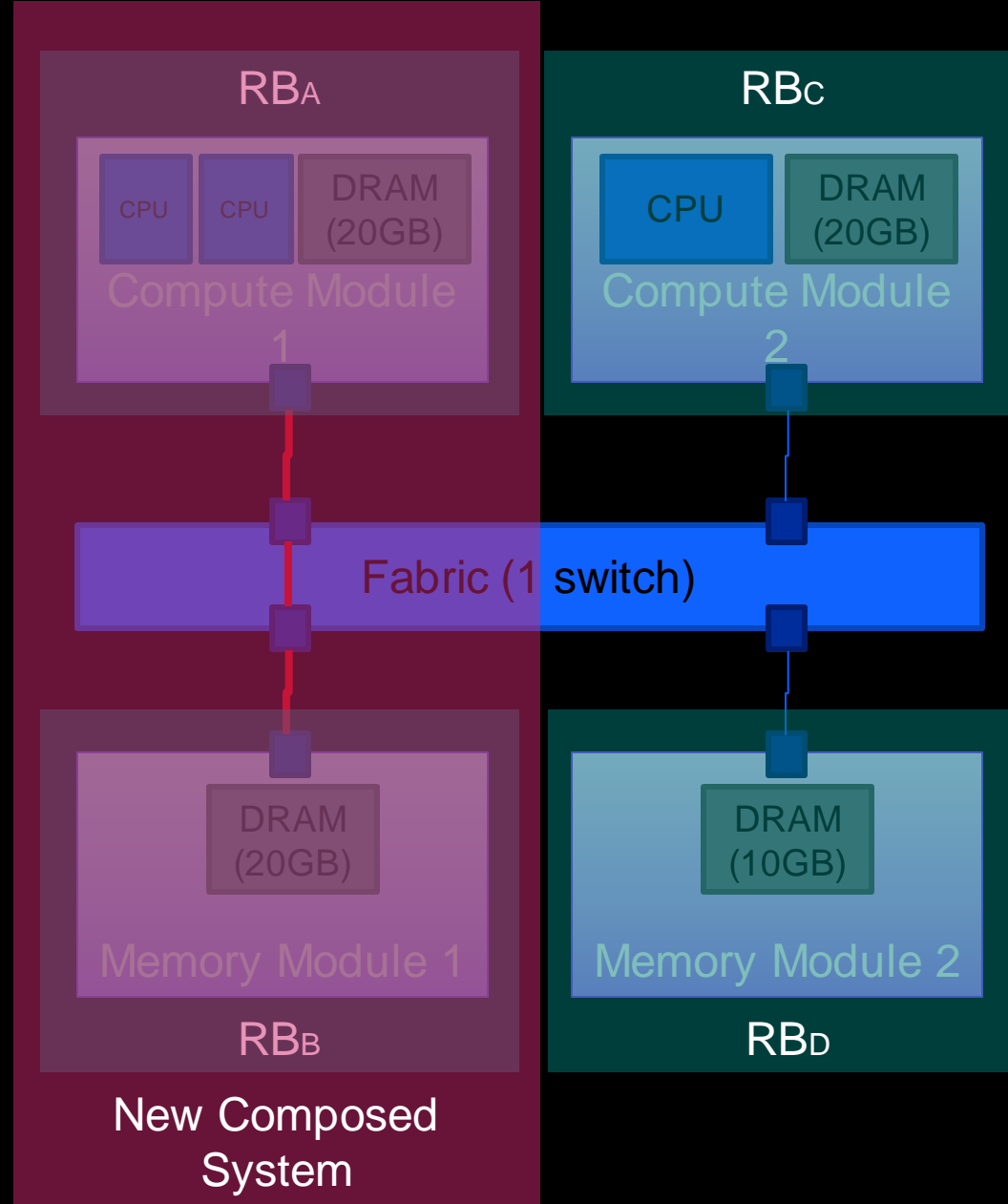
# Example system - ResourceBlocks



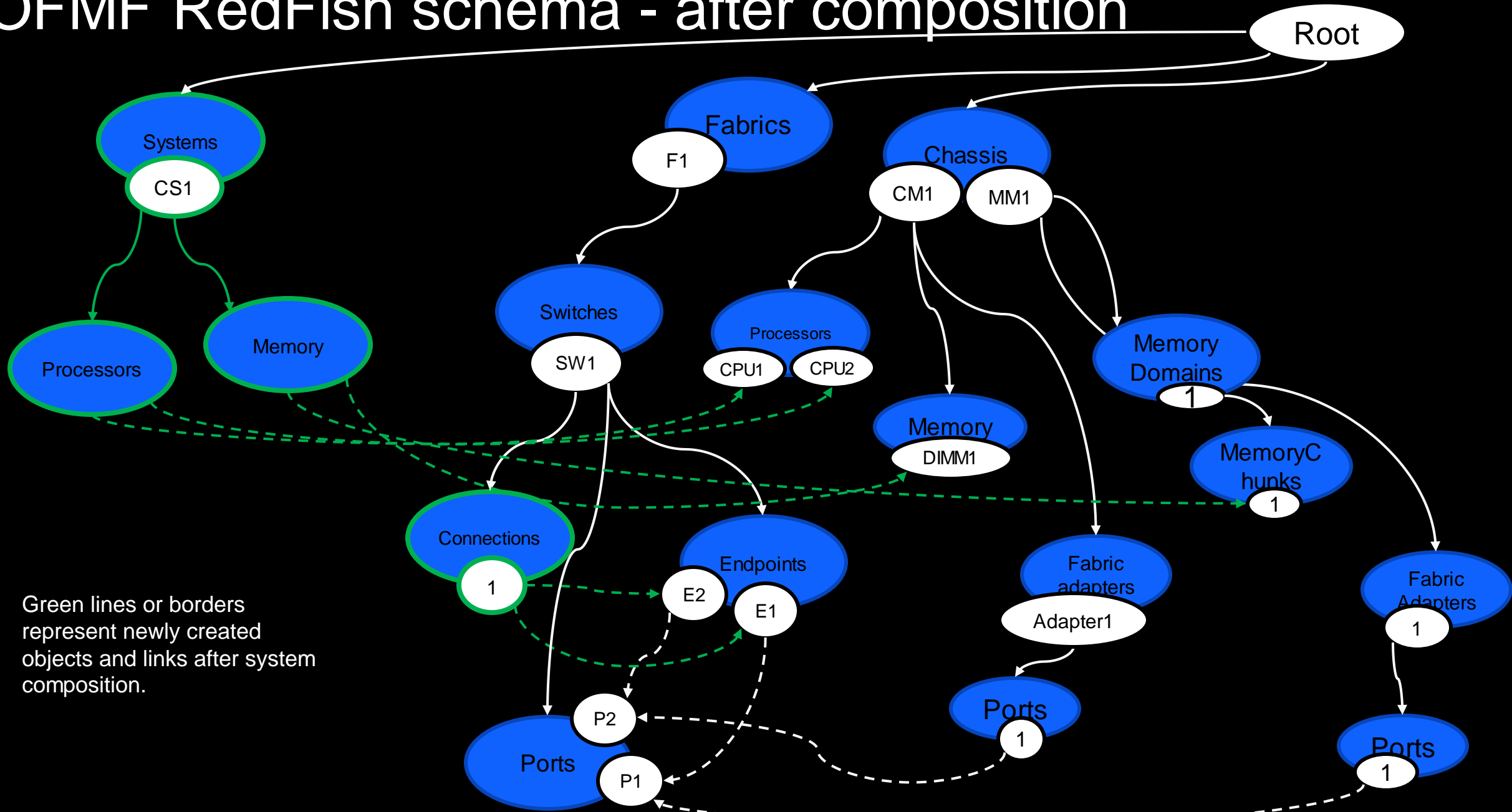
In this example composition request one Compute Module alone (RBA or RBc) does not have enough resources for satisfying the request. The Composition Service can select RBA and RBb for composing a new Computer System.

# Example system – Modules composition

Composition Service identifies RB<sub>A</sub> and RB<sub>B</sub> as candidates and composes them into a Computer System by creating a Fabric Connection through the OFMF.



# OFMF RedFish schema - after composition



IBM