### ComputerSystem creation via Constrained Composition through the ComposabilityService

ss

Figure 1: Option 1 — The Composability Manager defines Resources Blocks objects



Figure 2: Option 2 — The Provider defines the Resource Blocks and replicates the objects in the OFMF

|  |  |
| --- | --- |
| Use Case Description | * User requests a new ComputerSystem creation in the form of a Specific Composition request to the Composability Service.
 |
| Actors | * User (System administrator, Infrastructure Providers), Composability Service, OFMF & Provider.
* The Composability Service is a client of the OFMF and implements a policy to compose new ComputerSystems. It receives Constrained Composition requests containing the resource constrains that the new system must meet (# CPU, GBs of volatile memory, storage,etc..).
 |
| Description | * We want to hide low level details of the Fabric to users — this could be for many reasons like complexity or security — but still provide the capability to compose systems in an easy way.
* The Redfish specification already supports Composability .
* As part of Redfish Composability, ResourceBlocks group resources in an inventory of composable components, providing a layer of abstraction that clients can leverage to implement new composition policies and define new ComputerSystems.
 |
| Comments | * NOTE: The following comments are here to guide the group during the discussion on understanding how to create ResourceBlocks and identify the actor in charge of this task.
* The Redfish Composition Service defines a ResourceBlock as the smallest element for composition.
* The definition of a ResourceBlock depends on attributes of the physical resources. Example: CPU and local DRAM could be part of the same resource block. Fabric Attached Memory Chunks could be separated in multiple ResourceBlock.
* One of the actors in this use-case must define ResourceBlocks and Links to the physical resources belonging to the each ResourceBlock.
* The actor providing this capability is open to discussion. Following, two possible approaches:
	1. The Composability Service could define ResourceBlocks. In this case, the service scans the Redfish tree and creates ResourceBlocks. For each a Composition Request, the Composability Service can translate the request into changes in the Redfish tree that enable connectivity between resources (example: creates fabric connections and, if necessary, Memory Chunks). To support this approach, the Composability Service needs to know how to group resources belonging to the same ResourceBlocks and connect resources belonging to different ResourceBlocks. These capabilities must be available for each Provider supporting composability.
	2. The Provider could define ResourceBlocks. In this case, having knowledge of the underlying hardware resources managed by the Fabric Manager, the Provider defines ResourceBlocks depending on physical or logical constrains. The ResourceBlocks will be exposed to Composability Service and other clients via the OFMF. A Composition Request coming from the Composability Service will be forwarded by the OFMF to the appropriate Provider managing the ResourceBlocks. As a note, not all Providers will need to define ResourceBlocks, only the one supporting Redfish Composability.
 |
| Input Data | * The User provides a description of the resources required by the new ComputerSystem
* Example: { "Name": " ComposedSystem1", "Processors": {…}, "Memory": { "Members": [{ "@Redfish.RequestedCount": 4, "CapacityMiB": 8192, "MemoryType": "DRAM", "MemoryDeviceType": "DDR4" }]}, "SimpleStorage": {…}}
 |
| Preconditions | * Composability Service can explore and inspect ResourceBlocks
 |
| Postconditions | * OFMF Redfish tree contains the description of a new ComputerSystem matching the criteria described in the Constrained Composition request.
* OFMF Redfish tree contains an updated view of the Fabric. This could include new Connections.
* Actual Fabric hardware state matches the OFMF Redfish description and desired Fabric functionalities are enabled (resource are accessible as part of the new ComputerSystem)
 |
| Trigger | * ComposabilityService receives a Constrained Composition Request
 |
| Normal Flow   | Note: For simplicity, flows are described as synchronous operations. Implementation details are beyond the scope of this first draft. Always for simplicity, in both figures I am omitting updates to Redfish trees.**Option 1: Composability Service defines the ResourceBlocks** (Figure 1)1. The Composability Service inspects the locally defined ResourceBlocks (created by scanning the Fabric)
2. It selects the subset that fulfills the request for a new ComputerSystem. In the example, these are called RBA and RBB
	1. The policy for composability is beyond the scope of this use-caseNote: ResourceBlocks must contain enough resource metadata to implement the desired policy (latency, bandwidth, etc...)
3. It reserves the resources belonging to the selected ResourceBlocks RBA and RBB
4. It sends multiple POSTs/PATCHs to the OFMF to connect all resources in ResourceBlocks RBA and RBB. Each resource will require one or more requests for a change to the OFMF.
5. The OFMF forwards each request to the Provider in charge of the Fabric attached resources.
6. The Provider translates each request in actions to be carried out by the Fabric Manager.
7. Once all resources are configured and ready, the Composability Service creates a new ComputerSystem.
8. Finally, the ComposabilityService returns the new ComputerSystem description to the user.

**Option 2: Provider defines the ResourceBlocks** (Figure 2)1. Composability Service inspects the ResourceBlocks querying the OFMF.
2. It selects the subset of ResourceBlocks that fulfills the request for a new ComputerSystem (RBA and RBB similar to Option 1)
3. It creates a single Specific Composition request listing the selected ResourceBlocks RBA and RBB to be composed and sends it to the OFMF.
4. The OFMF forwards the request to the Provider in charge of the ResourceBlocks
5. The Provider validates that the request is valid and reserves the ResourceBlocks RBA and RBB.
6. The Provider translates the request in a set of actions to connect the resources belonging to RBA and RBB via the FabricManager. Once every resource is setup and ready, the Provider returns the description of the new ComputerSystem and the current state of the Fabric.
7. The OFMF registers the new ComputerSystem and returns the new ComputerSystem state to the ComposabilityService.
8. The ComposabilityService forwards the new ComputerSystem definition to the user.
 |
|  |  |