

SO Building blocks

- Background
 - Services & Resource types
 - Actions
 - Key differences from Amsterdam/Beijing
- Execution flow
 - Decomposition
 - Homing
 - Assignment
 - Creation
 - Configuration
 - Activation

Background

Introduced with the Casablanca release, the SO building blocks are a set of database-driven, configurable and generic process steps to be leveraged through several actions defined as 'Macro' flows.

For each of the macro flows, there are a set of actions to be performed which are implemented as building blocks - which then implement generic logic to handle the orchestration of services and various type of resources orchestrated by ONAP, as well as their corresponding actions.

Services & Resource types

These resource types are essentially the ones defined in the model - through the SDC framework. With Casablanca, SO has

- Services (which can be a resource as well, through resource allotment; a service can be a resource for another service)
- Networks (L2, L3, or custom types)
- VNFs
- VF modules (i.e. a deployment unit, such as a HEAT stack)
- Volume groups
- AVPN bonding (AT&T VPN bonding)

Actions

- Create
- Assign
- Activate
- Unassign
- Delete
- Recreate
- DeactivateAndCloudDelete (i.e Deactivate & remove VM)
- ScaleOut

* Note : Not all services & resources implement all these actions, for reference see `orchestration_flow_reference` table in SO's MariaDB.

Key differences from Amsterdam/Beijing

- Pre-loading data into SDNC is no longer required - the service inputs, or VNF/C attributes & properties can be supplied in the SO request in a structured format (JSON).

Execution flow

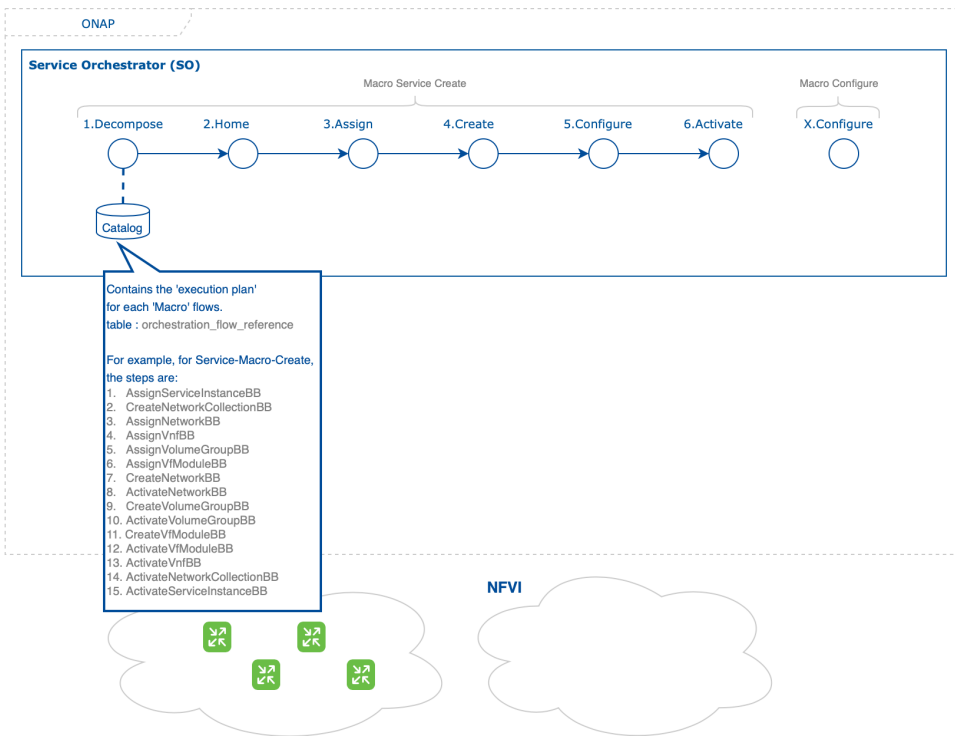
The following is a **highly simplified view** of how SO would handle a Service Create Macro, or in other words the 'one-click' instantiation for a complete service.

The execution order will be based on the `orchestration_flow_reference` table and will be executing those following the sequence numbers defined in the reference data (SEQ_NO column).

1. Decomposition

The first function is the request decomposition : essentially deriving all the required actions for this specific macro flow.

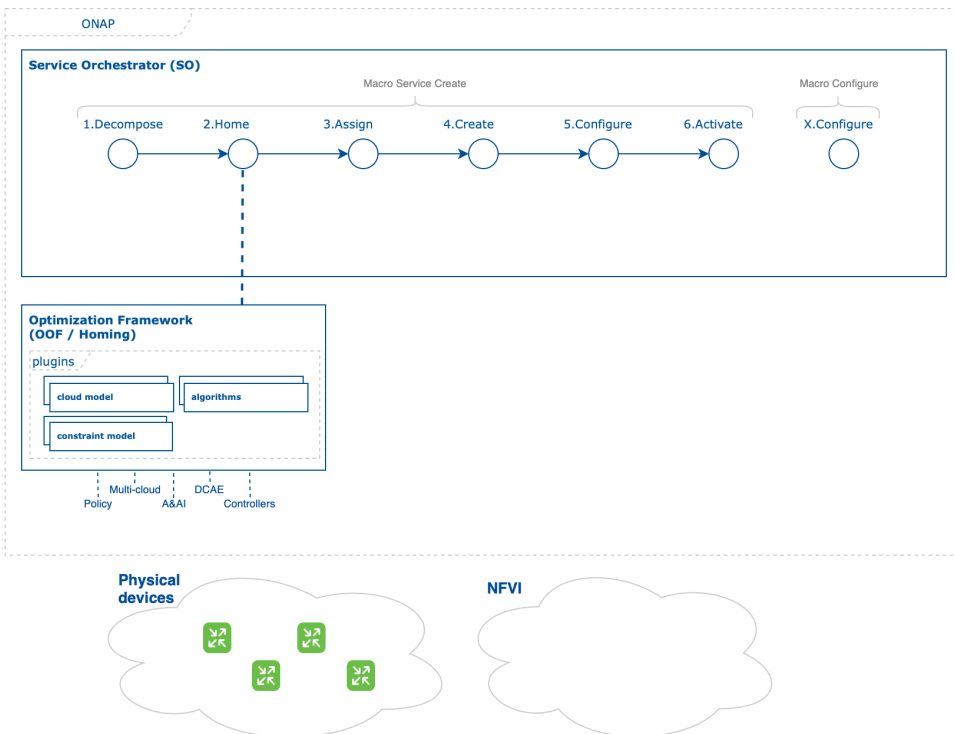
This step is executed every time the service recipe leverages the 'main' building block - *WorkflowActionBB*.



2. Homing

The second function will be the homing of the service and resources, although for now not all macro flows implement the homing step.

Homing consists of determining 'where' to place the service and all its resources, leveraging different models (through the OOF framework) based on various data points (location, inventory data, resource availability, business rules, etc.)



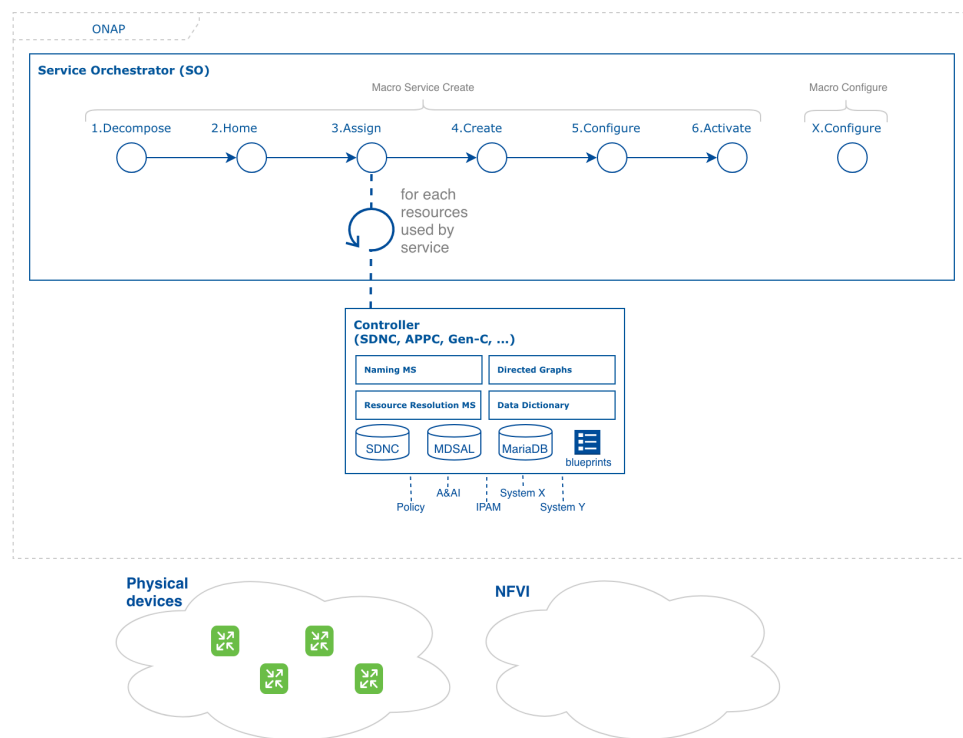
3. Assignment

The third step is to perform assignment. Assignment will be performed per the orchestration plan - and will start from the service-level, then iterating through the various resources contained within the service.

Assignment can involve different systems, business logic (policies) or other ONAP components in order to derive automatically what the service inputs or resource attributes/properties values will end up being.

For example: for a '*management_ip*' property on a specific VNF component, representing the management interface address, the system may have to reach out to an IPAM system, pulling information from a specific subnet (either rules-based, leveraging a database such as the controller data store, or provided through input). The ONAP Controller Design Studio (CDS) initiative implements an exhaustive framework to tackle this (through data dictionary, controller blueprints or other means as it evolves).

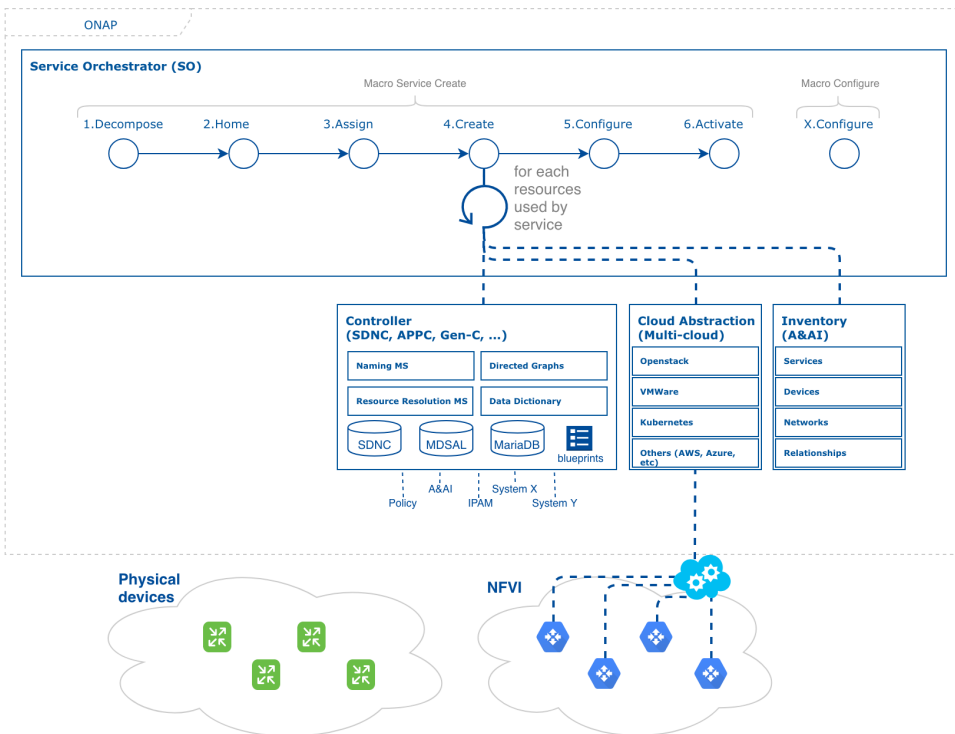
These assigned values will be stored in the service context, inside the controller's data store (MDSAL). These can be leveraged at any time afterwards - this replaces the legacy preload function.



4. Creation

The fourth function is the service creation - which, when executed by a Macro request, would perform the 'single click' actions required to bring up a service and all its resources.

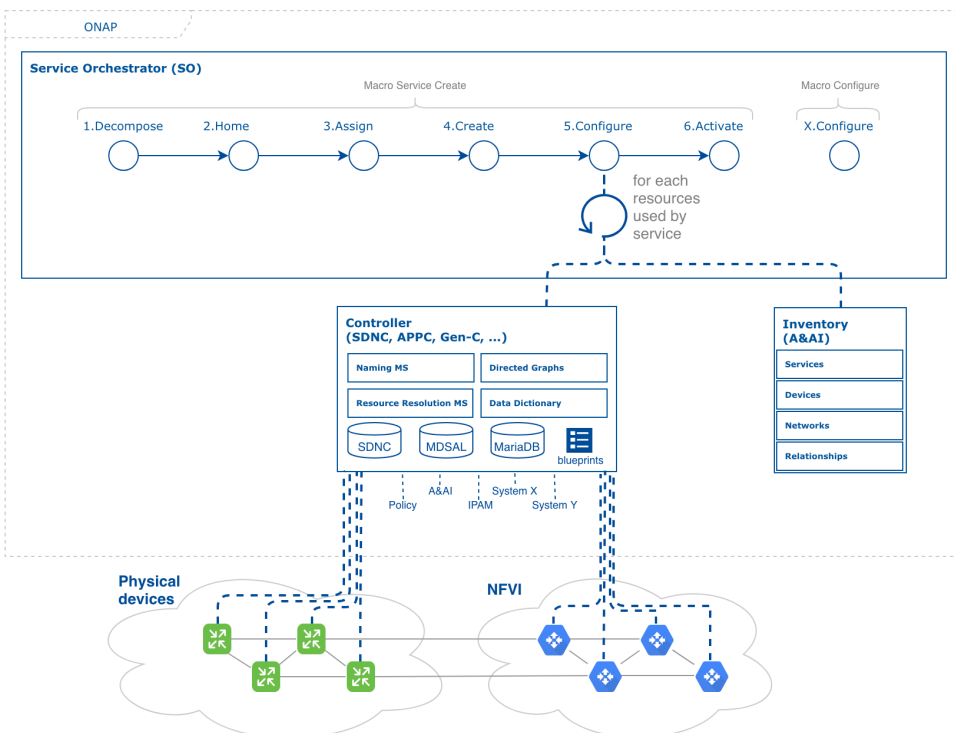
This applies mostly to VNFs and ANFs, but isn't applicable to PNFs for obvious reasons. It will spin up the VM/Containers in the NFVI, and provide basic device configuration in order to have an operational, manageable VNF - but not yet into the traffic path.



5. Configuration

The fifth function is the configuration of the device - essentially applying service configuration or application type configuration to the device so it can become operational.

This is done involving the right controllers, and again leverages the service & resource context stored in the controller data store, the directed graphs and/or Controller Design Studio blueprints artifacts (which can include DGs, code, etc.). It will then transformed all the assigned values into configuration payload for the device, using the right protocol (Netconf/Restconf or just Rest APIs), and when triggered through CDS Blueprints will use Velocity templating for transformation/mapping. This applies to PNFs or VNFs - it is purely network device configuration. If any aspect of the configuration needs to be represented in the inventory, it will perform such updates.



6. Activation

The last function is activation of the service - which is bringing the device in the 'real' network traffic. It can be separate from the configuration so that you can fully configure the device to be operational, and then start steering traffic or attaching the device to the rest of the network afterwards. After this step the device should be considered 'live'. It also updates the device state in the inventory accordingly.

