

# SO Building blocks

Seshu Kumar Mudiganti - Huawei

# Problem Statement

In the recent meetups of ONAP F2F many of the operators raised concerns on the adoption of ONAP as it is.

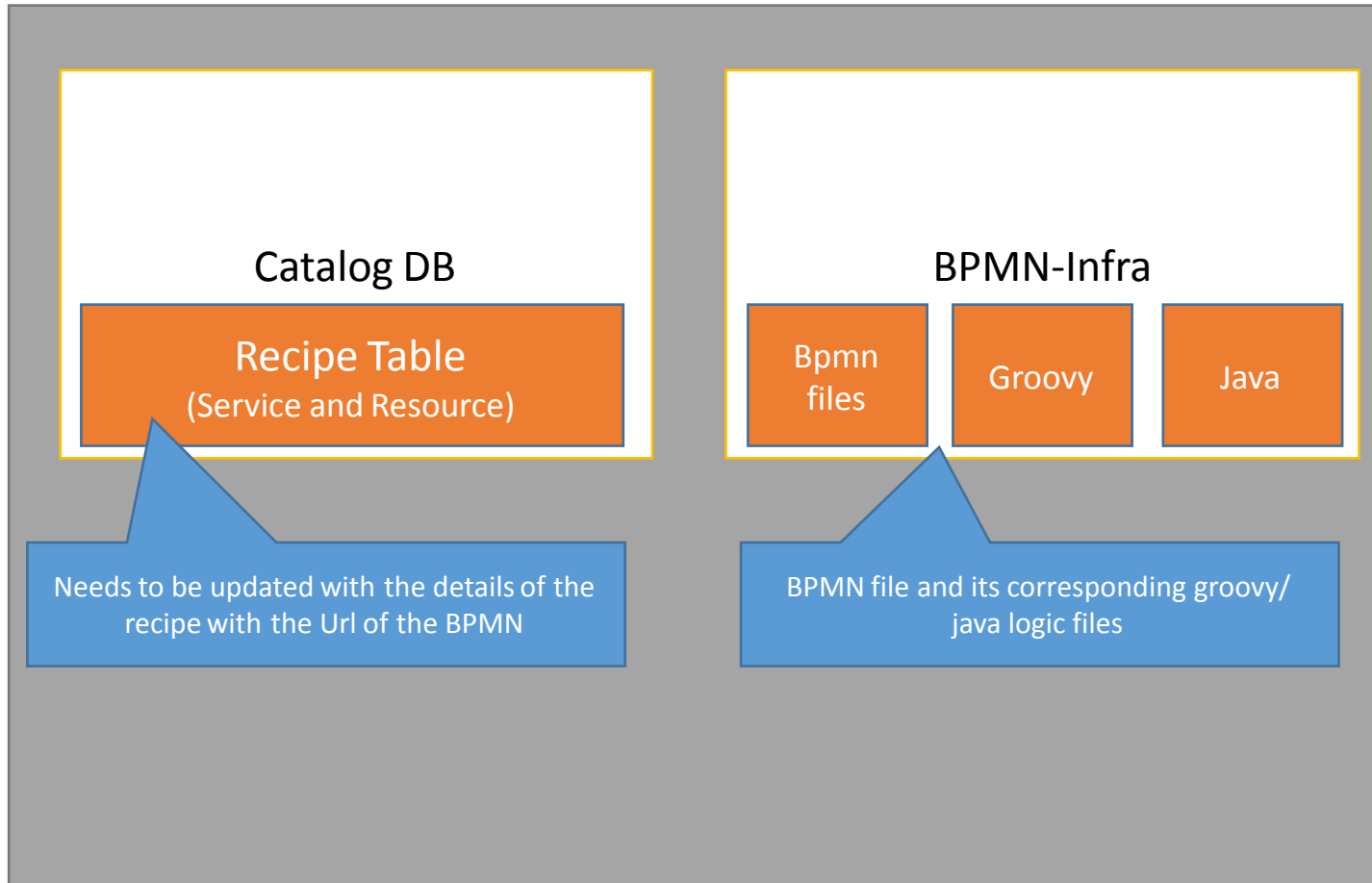
The main challenges raised were around:

- feasibility for the plug and play
- readiness of the modules to be used in the field
- Ease of use and adoption in production

# Challenges in SO

- SO has design time configurations required as recipe that involves the changes in the BPMN workflows and its corresponding logics in groovy / Java
- These configuration though are kept as much as generic would need some amendments from time to time and could pose issues to user who are unaware of BPMN syntax and SO internals
- The need of the hour is to have an generic notations as individual functional blocks that could be stitched together based on the business case need.

# SO Design time - Impacted components



- SO supports 2 kinds of workflows
- **Generic Recipes :** Workflows that are pre-defined in the SO and delivered as part of the installation package
  - **Custom Recipes :** Workflows that could be on-boarded to the SO on need basis.

## Custom Recipes on-boarding process:

- **Manual operation that needs to perform on-boarding of recipes and their configurations**
  - would need changes across Catalog DB for the configuration
  - Needs to be updated with the details of the recipe with the Url of the BPMN

# Introduction SO Building Blocks

- 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
- The building blocks implement generic logic to handle the orchestration of services and various type of resources orchestrated by ONAP, as well as their corresponding actions.

# Ingredients of Building blocks

The building blocks constitutes 2 major components Nodes (Service/Resource) types and 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

## Actions

Set of predefined functions that are trigger points implemented for services & resources.

Examples :

Create

Assign

Activate

Unassign

Delete

Recreate

ScaleOut

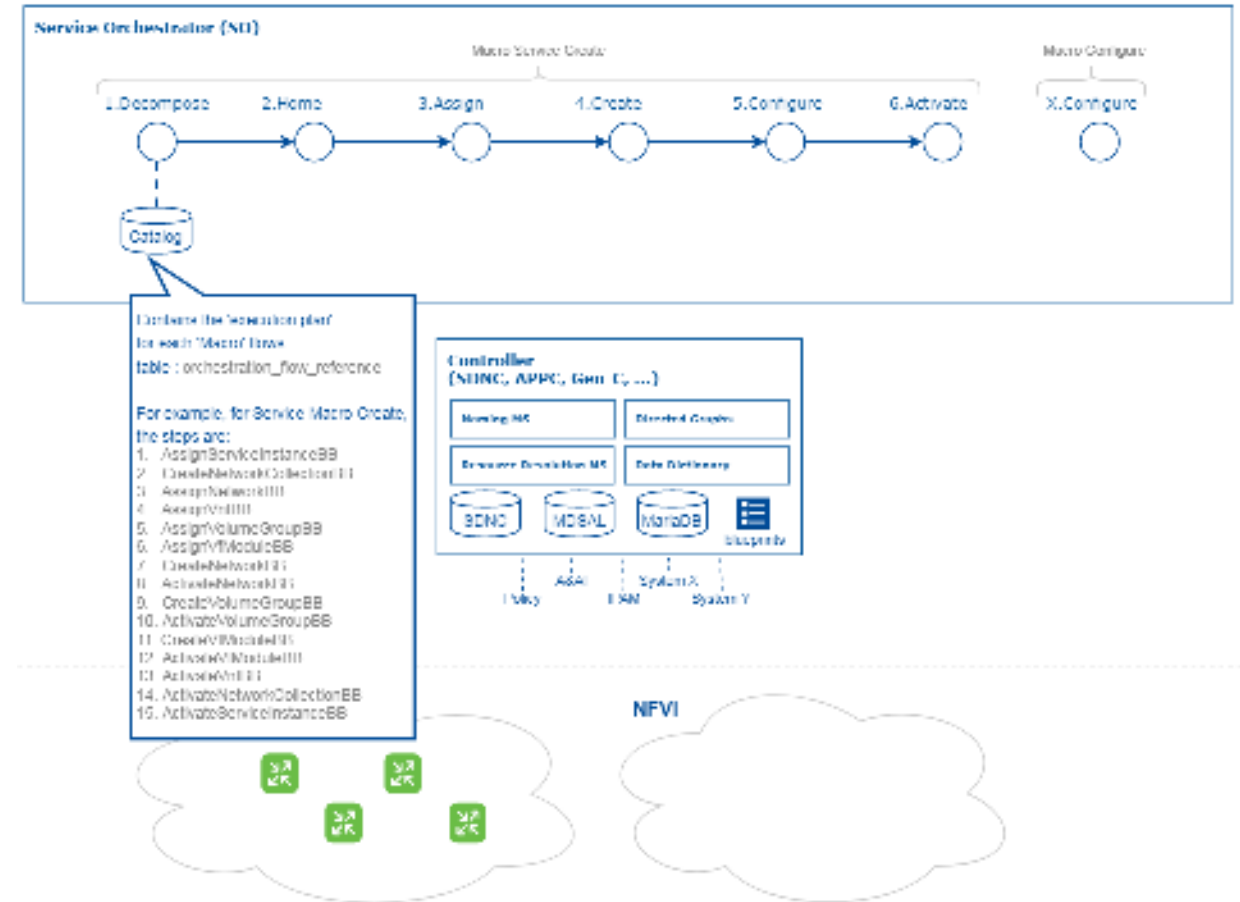
# How it works - Dublin

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 defined in the *orchestration\_flow\_reference* table

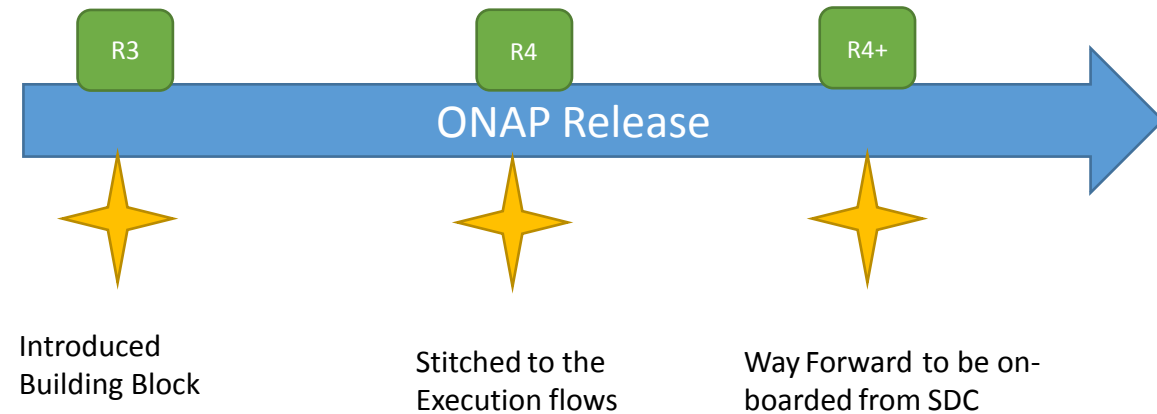
The SO execution engine would be executing those following the sequence numbers defined in the reference data (SEQ\_NO column).

<https://wiki.onap.org/display/DW/SO+Building+blocks>



# Road Map

- SO introduced the building block functionality in the Casablanca Release of ONAP
- In Dublin we are executing the building blocks defined in the *orchestration\_flow\_reference* table
- The future plan is to have the **building blocks** stitched from SDC-workflow designer and distributed to SO.





Thanks



SOL 005

SO

Sol 003

