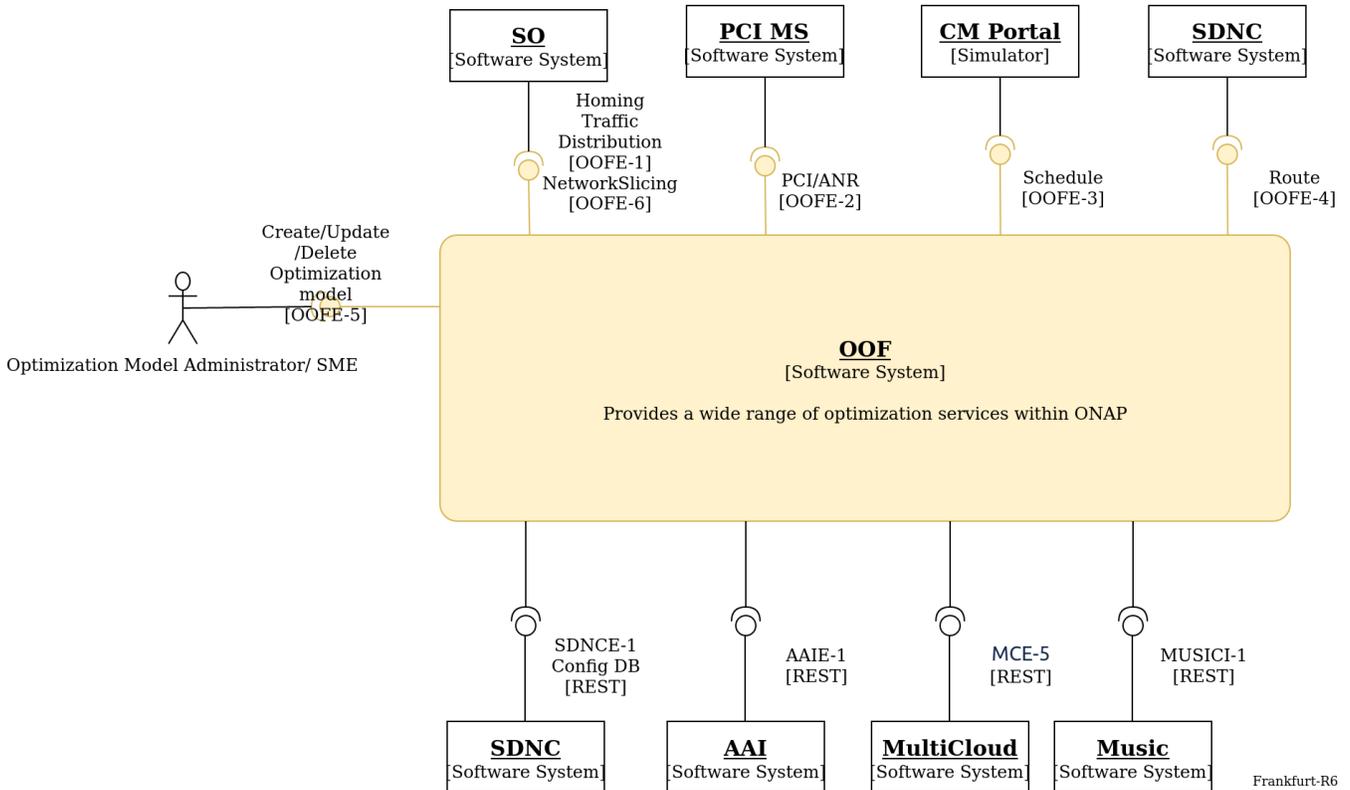


ARC OOF Component Description - Frankfurt

STATUS:

OOF (ONAP Optimization Framework):

1. High Level Component Definition and Architectural Relationships



2. API definitions

OOF provides the following interfaces:

Interface Name	Interface Definition	Interface Capabilities
OOFE-1	<ul style="list-style-type: none"> Homing Traffic Distribution Network Slicing 	It enables placement based on a wide variety of policy constraints including capacity, location, platform capabilities, and other service specific constraints.
OOFE-2	PCI/ANR Optimization	Enables PCI/ANR optimization API for SON.
OOFE-3	Schedule Optimization	a policy driven workflow schedule optimizer for change management planning. This interface schedule workflows in time to maximize parallel change management activities, while respecting dependency between the workflows.

OOFE-4	Route Optimization	Provides an interface for Router Optimization.
OOFE-5	OOF Model Administrator	This is for the OOF Model Administrator API. This API is a way to dynamically change the optimization models that will be used to find solutions for various optimization problems. This API will be used to Create, Update or Delete Models.
OOFE-6	Network Slicing	This interface enables slice selection recommendations.

Note: xxxI interface is a Component internal interface. xxxxE interface is a component external interface

The current API documents can be found at:

- The OOF user guide can be found at: [OOF latest user guide](#)
- OOF internal APIs can be found: [OOF API specification](#)

OOF consumes the following Interfaces:

Interface Name	Purpose Reason For Use
SDCE-1	For PCI/ANR optimization, OOF Retrieves the Cellsite inventory details from the configdb API, which is hosted as part of the SDNC /R component
AAIE-1	OOF interfaces with AAI to retrieve the inventory
MCE-5	OOF queries Multicloud for real-time available capacity information
MUSICI-1	OOF interfaces with MUSIC to persist service state.

3. Component Description:

ONAP Optimization Framework (OOF) is an Umbrella project, with the primary goal of addressing the optimization needs of ONAP. OOF is a framework that supports creating and running a suite of Optimizing applications including:

- Change Management Scheduling optimizer
- Homing/Placement optimizer
- PCI optimizer
- Route optimizer

OOF is targeted to be an optimization platform with the following goals:

- Declarative, policy-driven approach to solving optimization problems
- Reusable components: data and policy adapters/libraries, execution environment
- Support General-Purpose as well as Custom optimizers
- Extensible to multiple optimization problems

Internally, OOF has the following components:

- OSDF: Optimization Service Design Framework, which is a collection of APIs and libraries, along with a generic runtime optimizer
 - Status: Part of ONAP since Beijing Release.
- HAS: Homing and Allocation Service, which provides a policy based constraint driven selection optimizer
 - Status: Part of ONAP since Beijing Release.
- CMSO: Change Management Schedule Optimizer, providing schedule optimization
 - Status: Seed code up-streamed in R3
- FGPS: Fine Grained Placement Service
 - Status: Seed code up-streamed, POC in Frankfurt

4. Known system limitations

Please find the known system limitations in the following links

- [has release notes](#)
- [osdf release notes](#)
- [cmso release notes](#)

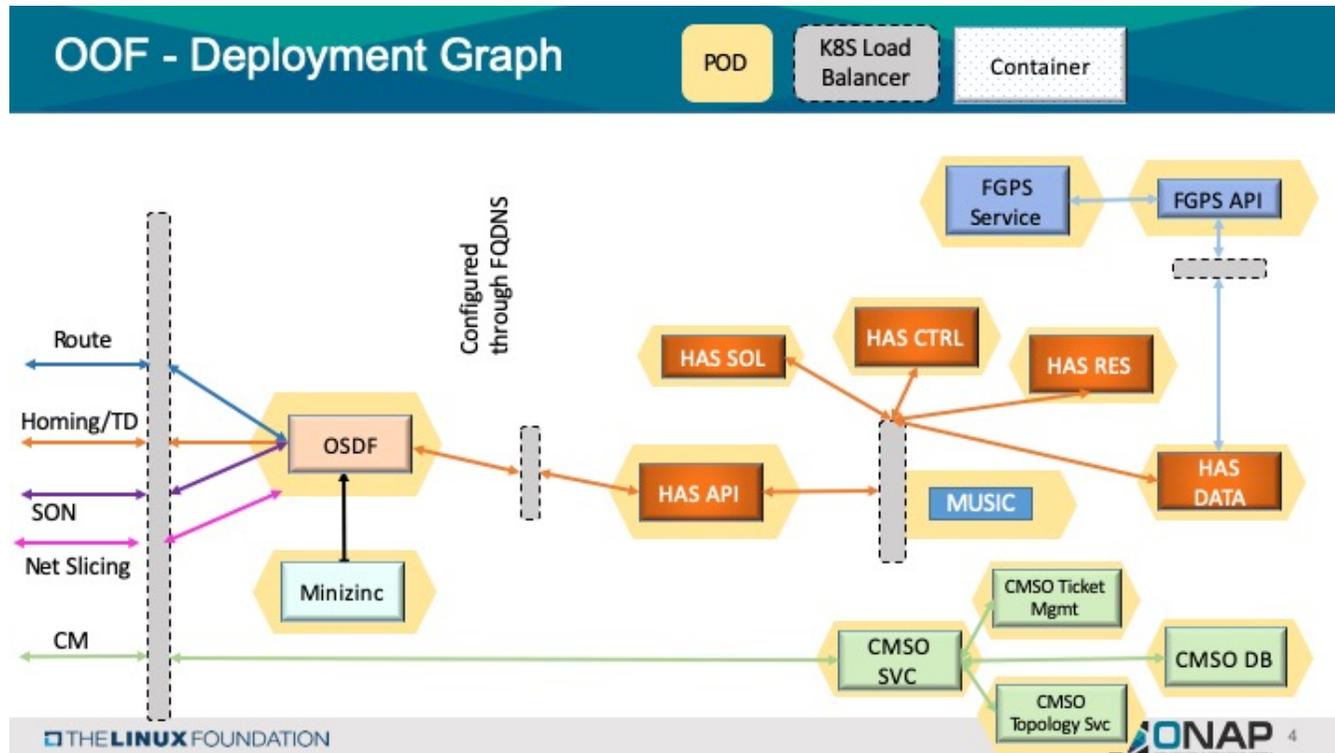
5. Used Models

OOF doesn't directly create models, but indirectly consumes them via AAI, Policy and Multi Cloud.

- Service and Resource Info, from: AAI
- Network Topology for CM: AAI

- HPA Flavors/Capabilities/CapacityInfo, from : AAI
- Policy Models (homing, PCI) from: Policy
- Infrastructure Metrics Info (capacity), from: MultiCloud
- Cloud agnostic Intent Info, from: MultiCloud
- AZ level capacity Info, from: MultiCloud (for F-GPS)
- PCI configuration data(not yet a part of SDC model)
- Slice/Subnet Profile and Slice/Subnet Instance models, from AAI

6. System Deployment Architecture



7. New Capabilities in this Release

- Component Upgrades to new Policy Lifecycle API
- Support for E2E Network Slicing
- Support to perform CRUD operations on certain optimization models during runtime.

8. References