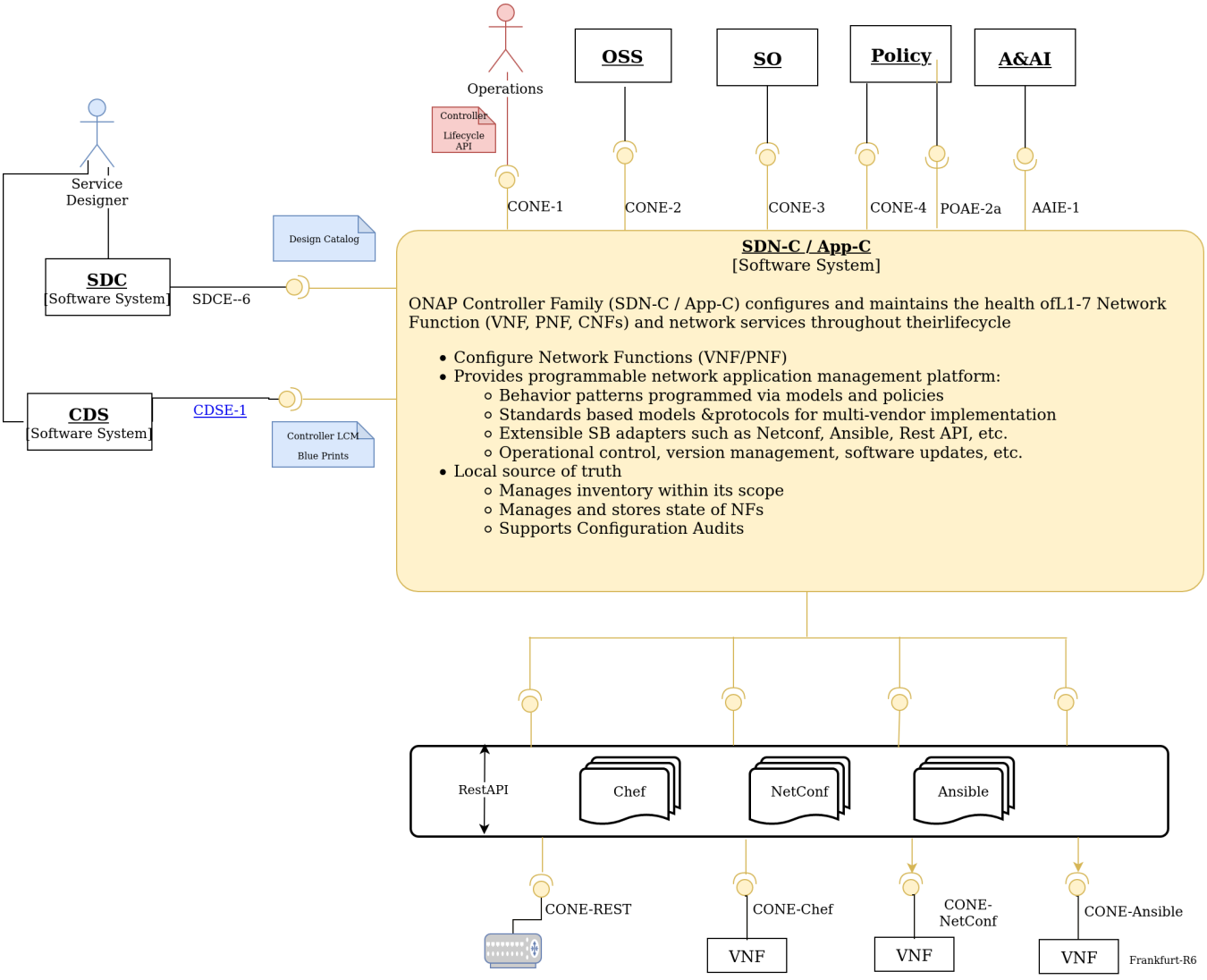
ARC Controller Component Description – Frankfurt

STATUS: Under Construction

SDN-C / App-C:

Note: ONAP has two application level configuration and lifecycle management modules called SDN-C and App-C. Both provide similar services (application level configuration using NetConf, Chef, Ansible, RestConf, etc.) and life cycle management functions (e.g. Stop, resume, health check, etc.). They share common code from CCSDK repo. However, there are some differences between these two modules (SDN-C uses CDS only for onboarding and configuration / LCM flow design, whereas App-C uses CDT for the LCM functions for self service to provide artifacts storing in APP-C Database). SDN-C has been used mainly for Layer1-3 network elements and App-C is being used for Layer4-7 network functions. This is a very loose distinction and we expect that over time we will get better alignment and have common repository for controller code supporting application level configuration and lifecycle management of all network elements (physical or virtual, layer 1-7). Because of these overlaps, we have documented SDN-C and App-C together.

1 High Level Component Definition and Architectural Relationships



ONAP Controller Family (SDN-C / App-C) configures and maintains the health ofL1-7 Network Function (VNF, PNF, CNFs) and network services throughout their lifecycle

- Configure Network Functions (VNF/PNF)
- Provides programmable network application management platform:
  - Behavior patterns programmed via models and policies
  - Standards based models &protocols for multi-vendor implementation
  - Extensible SB adapters such as Netconf, Ansible, Rest API, etc.
  - Operational control, version management, software updates, etc.
- Local source of truth
  - Manages inventory within its scope
  - Manages and stores state of NFs
  - Supports Configuration Audits

2. SDN-C/APP-C API definitions

Controller provides the following interfaces:

Interface Name	Interface Definition	Interface Capabilities
CONE-1	Operations Interface	An interface to request for Lifecycle management operations on network resources
CONE-2	OSS Interface	An interface to request for Lifecycle management operations on network resources
CONE-3	Service Order Interface	An interface to request for Configuration and Lifecycle management operations on network resources
CONE-4	Policy Interface	An interface to support LCM requests such as Restart, Rebuild, Migrate, Evacuate operations on network resources (APP-C interfaces with openstack to send those LCM requests to VNF/VNF-C/VM)

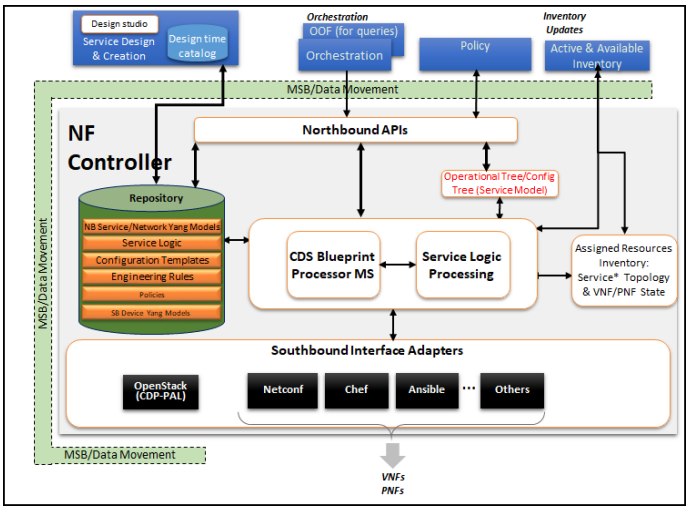
The current API documents can be found at:

https://onap.readthedocs.io/en/casablanca/submodules/appc.git/docs/index.html

Controller consumes the following interfaces:

Interface Name	Interface Definition	Interface Capabilities
CONE-5	Rest API	An interface for communication with external systems such as IP management
CONE-6	Resource Chef API	An interface for configuration and Lifecycle management of network resources using Chef protocol
CONE-7	Resource NetConf API	An interface for configuration and Lifecycle management of network resources using NetConf protocol
CONE-8	Resource Ansible API	An interface for configuration and Lifecycle management of network resources using Ansible protocol
SDCE-6	SDC Interface	An interface to receive resource Templates from SDC design catalog
CDSE-1	CDS Interface	An interface to receive resource blueprint from CDS
AAIE-1	Inventory Service Interface	An interface to create, update, query, and delete resource information and relationships
POE-2a	PDP Query API	Policy Decision Point query for IP address

3. Component Description:



4. known system limitations

- Lack of clarity & roles in the controller (which controller does what?)
- Proliferation of controller instances (many similar yet different controller instances)
- Divergence of controller implementation (lack of common controller framework)
- Duplicate and uncoordinated interfaces (lack of uniform common services in southbound interfaces)
- Controller resiliency and horizontal scaling

5. Used Models

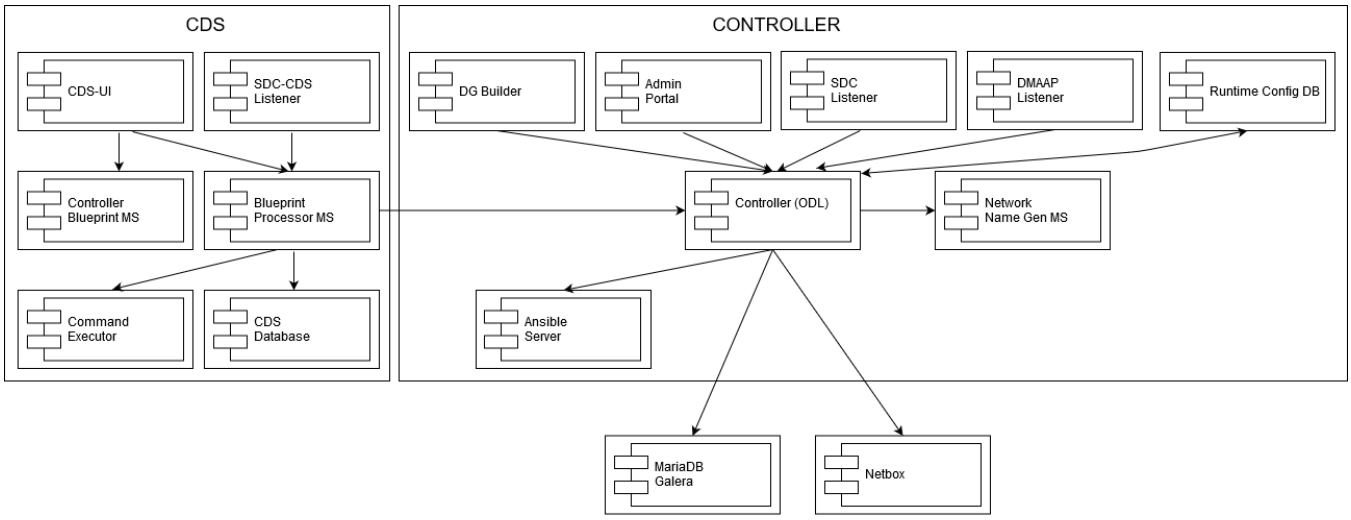
Controllers use the following models:

- TOSCA
- YANG

6. System Deployment Architecture

Controller consists of the following containers:

- CDS-UI container - Design Time
- SDC-CDS Listener container - Design Time
- CDT-UI container - Design Time
- Controller Blueprint MS Container - Design Time
- Blueprint Processor MS Container - Run Time
- Command Executor Container - Run Time
- CDS Database Container - Run Time
- DG Builder Container - Design Time
- Admin Portal Container - Run Time
- SDC Listener Container - Run Time
- DMAAP Listener Container - Run Time
- Controller (ODL) Container - Run Time
- Network Name Gen MS Container - Run Time
- Ansible Server Container - Run Time
- MariaDB Galera - Platform Service
- Netbox - Platform Service
- RunTime Config DB - Run Time



7. New Release Capabilities

- Upgrade of ODL to Neon SR1
- Change Management use case support
- Scaling use case support
- Closed loop use case support
- Platform maturity advancement (Scalability, Stability, Security, and Performance)
- Network discovery enhancements
- RunTime Config DB is an independent component that is integrated with the ODL SDN-C controller in R6 (Frankfurt release). The RunTime Config DB capabilities are part of CC-SDK. q.v. CONFIGURATION & PERSISTENCY SERVICE for more information.

8. References

1. APP-C overview & User Guide: https://docs.onap.org/en/casablanca/guides/onap-developer/developing/index.html#application-controller
2. SDN-C overview & User Guide: https://docs.onap.org/en/casablanca/guides/onap-developer/developing/index.html#software-defined-network-controller