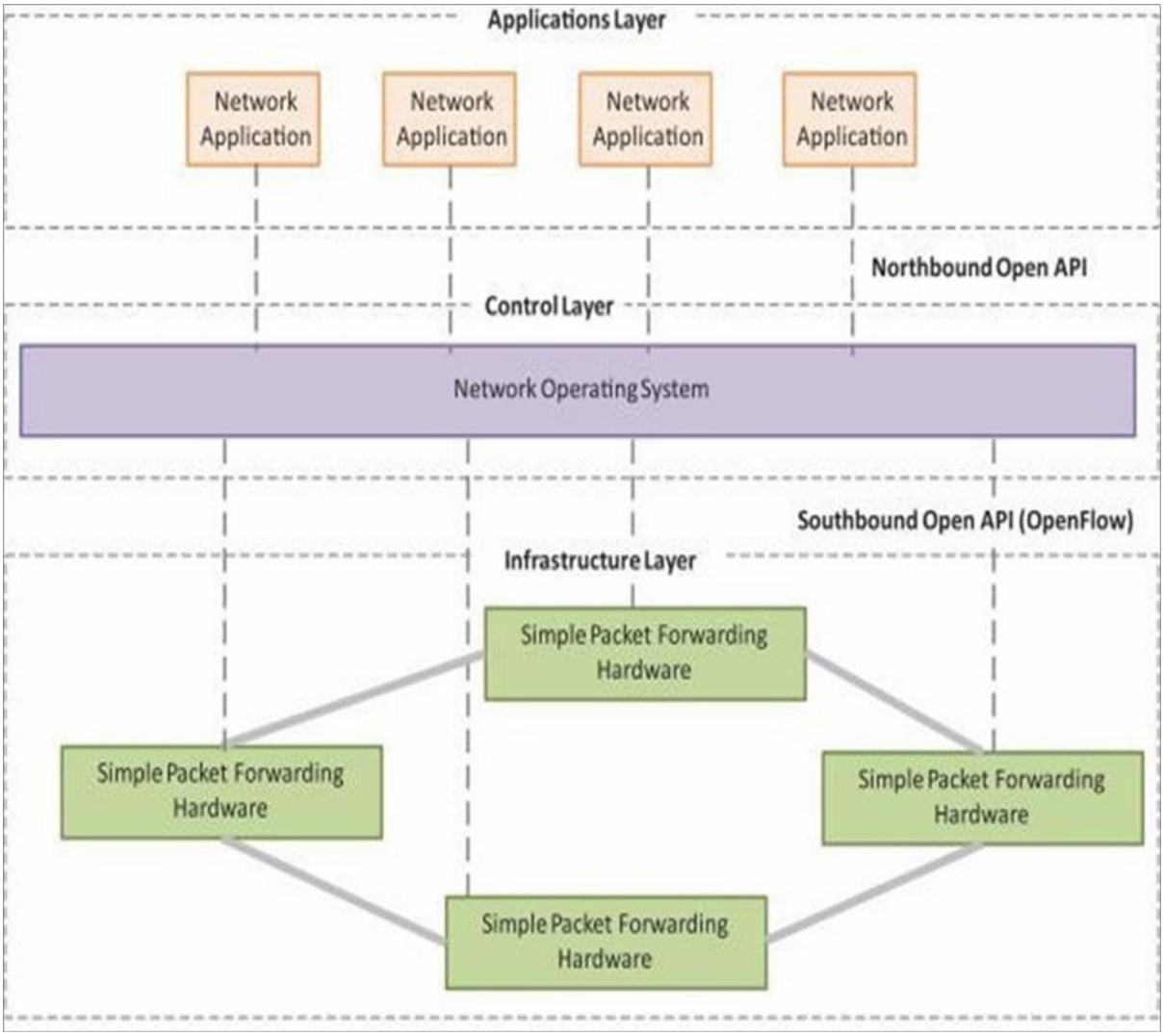
Software Defined Networking(SDN):

- Software-DefinedNetworking (SDN) isanetworking architecture that separates the control plane from the data plane and centralizes the networkcontroller.
- Software-based SDN controllers maintain a unified view of thenetwork
- Theunderlying infrastructure in SDN uses simple packet forwarding hardware as opposed to specialized hardware in conventionalnetworks.



SDN Architecture

Key elements of SDN:

1) **Centralized NetworkController**

   With decoupled control and data planes and centralized network controller, the network administrators can rapidly configure the network.

2) **Programmable OpenAPIs**

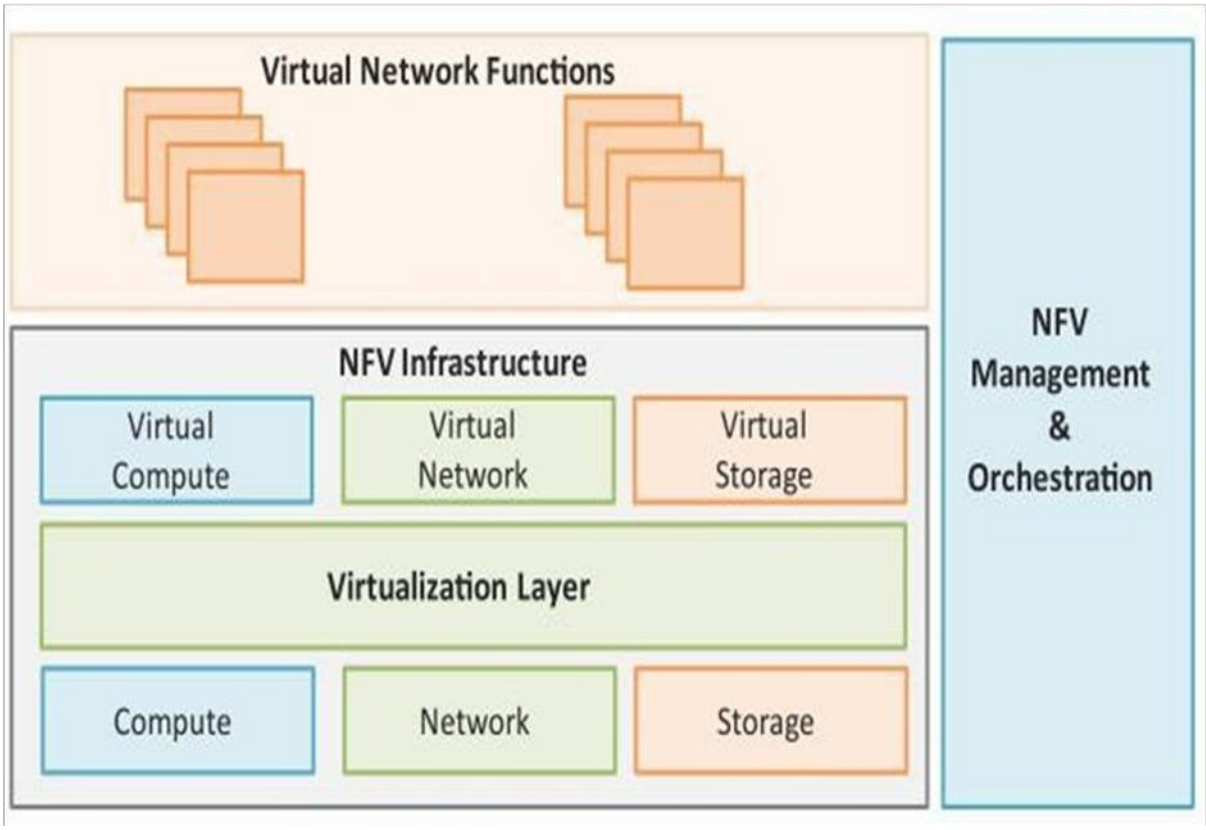
   SDN architecture supports programmable open APIs for interface between the SDN application and control layers (Northbound interface).

3) **Standard Communication Interface(OpenFlow)**

   SDN architecture uses a standard communication interface between the control and infrastructure layers (Southbound interface). OpenFlow, which is defined by the Open Networking Foundation (ONF) is the broadly accepted SDN protocol for the Southboundinterface.

Network Function Virtualization(NFV)

- Network Function Virtualization (NFV) is a technology that leverages virtualization to consolidate the heterogeneous network devices onto industry standard high volume servers, switches andstorage.
- NFV is complementary to SDN as NFV can provide the infrastructure on which SDN canrun.



NFV Architecture

Key elements of NFV:

1) **Virtualized Network Function(VNF):**

   VNF is a software implementation of a network function which is capable of running over the NFV Infrastructure (NFVI).

2) **NFV Infrastructure(NFVI):**

   NFVI includes compute, network and storage resources that are virtualized.

3) **NFV Management andOrchestration:**

   NFV Management and Orchestration focuses on all virtualization-specific management tasks and covers the orchestration and life-cycle management of physical and/or software resources that support the infrastructure virtualization, and the life-cycle management of VNFs.

Need for IoT Systems Management

Managing multiple devices within a single system requires advanced management capabilities.

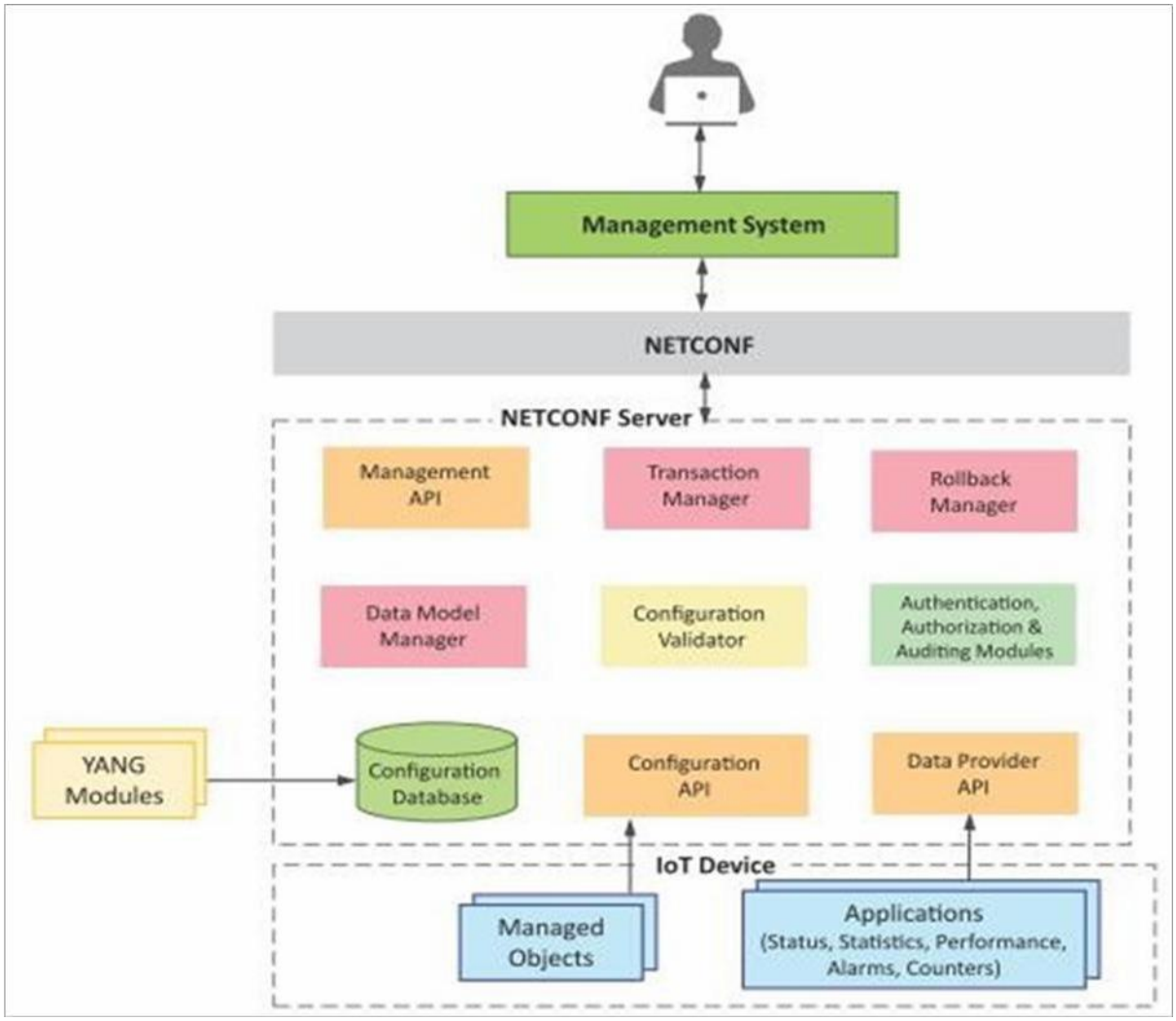
1) **Automating Configuration :** IoT system management capabilities can helpin automating the systemconfiguration.
2) **Monitoring Operational & Statistical Data :** Management systems can help in monitoring opeartional and statistical data of a system. This data can be used for fault diagnosis orprognosis.
3) **Improved Reliability:** A management system that allows validating the system configurations before they are put into effect can help in improving the systemreliability.
4) **System Wide Configurations :** For IoT systems that consists of multiple devices or nodes, ensuring system wide configuration can be critical for the correct functioning of thesystem.
5) **Multiple System Configurations :** For some systems it may be desirable to have multiple valid configurations which are applied at different times or in certainconditions.
6) **Retrieving & Reusing Configurations :** Management systems which have the capability of retrieving configurations from devices can help in reusing the configurations for other devices of the sametype.

IoT Systems Management with NETCONF-YANG

YANG is a data modeling language used to model configuration and state data manupulated by the NETCONF protocol.

The generic approach of IoT device management weith NETCONF-YANG. Roles of various componentsare:

1) ManagementSystem
2) ManagementAPI
3) TransactionManager
4) RollbackManager
5) Data ModelManager
6) ConfigurationValidator
7) ConfigurationDatabase
8) ConfigurationAPI
9) Data ProviderAPI



1) **Management System :** The operator uses a management system to send NETCONF messages to configure the IoT device and receives state information and notifications from the device as NETCONFmessages.
2) **Management API :** allows management application to start NETCONFsessions.
3) **Transaction Manager:** executes all the NETCONF transactions and ensures that ACID properties hold true for thetrasactions.
4) **Rollback Manager :** is responsible for generating all the transactions necessary to rollback a current configuration to its original state.
5) **Data Model Manager :** Keeps track of all the YANG data models and the corresponding managed objects. Also keeps track of the applications which provide data for each part of a datam,odel.
6) **Configuration Validator :** checks if the resulting configuration after applying a transaction would be a validconfiguration.
7) **Configuration Database :** contains both configuration and operastionaldata.

8) **Configuration API :** Using the configuration API the application on the IoT device can be read configuration data from the configuration datastore and write opeartional data to the opearationaldatastore.
9) **Data Provider API:** Applications on the IoT device can register for callbacks for various events using the Data Provider API. Through the Data Provider API, the applications can report statistics and opeartionaldata.