ONAP
OPEN NETWORK AUTOMATION PLATFORM

WELCOME TO HONOLULU WELCOME TO

LF NETWORKING

Virtual Technical Meetings

Configuration Persistence Service in R8+

- Virtual Developer Design Forum (vDDF) / Oct 15, 2020 / version 12

Ben Cheung (Nokia)
Marge Hillis (Nokia)
Shankar N K (AT&T)
Ted Johnson (AT&T)
Claudio Gasparini
Zu Qiang (Ericsson)
Tony Finnerty (Ericsson)
Toine Siebelink (Ericsson)
Michela Bevilacqua (Ericsson)
Jacqueline Beaulac (Ericsson)
Rishi Chail (Ericsson)
Ciaran Johnston (Ericsson)
Joachim Blixt
Pawel Slowikowski (Samsung)
Swami N (Wipro)
Bruno Sakoto (Bell Canada)
Philippe Leger (Bell Canada)
Aditya Puthuparambil (Bell Canada)
Fred Feisullin (Verizon Wireless)

R8 DDF Presentation

TIME	AUG 17, 2020 CPS REQUIREMENTS SubComm AGENDA
10 min	**Context for CPS & Agenda** – Context for CPS & Basic problems to be addressed
10 min	**Overview of CPS** – Overview to CPS
15 min	**Model Driven CPS Proof of Concept (PoC) & Demo** – Overview of the Model-Driven CPS PoC for R7
5 min	**R8 & Beyond Roadmap** – Model Driven Proof of Concept (PoC) in R7, way forward in R8 Honolulu, New plan & roadmap
10 min	**Use Cases using CPS Database** – Overview of CPS Applications (State Mgmt)
10 min	**Questions & Answers** – Q&A

TIME	Q&A Session Post-Session
-	**Follow-up questions** – Follow-up meetings at CPS Team Call (Fridays)

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Context for
Configuration Persistence Service (CPS)

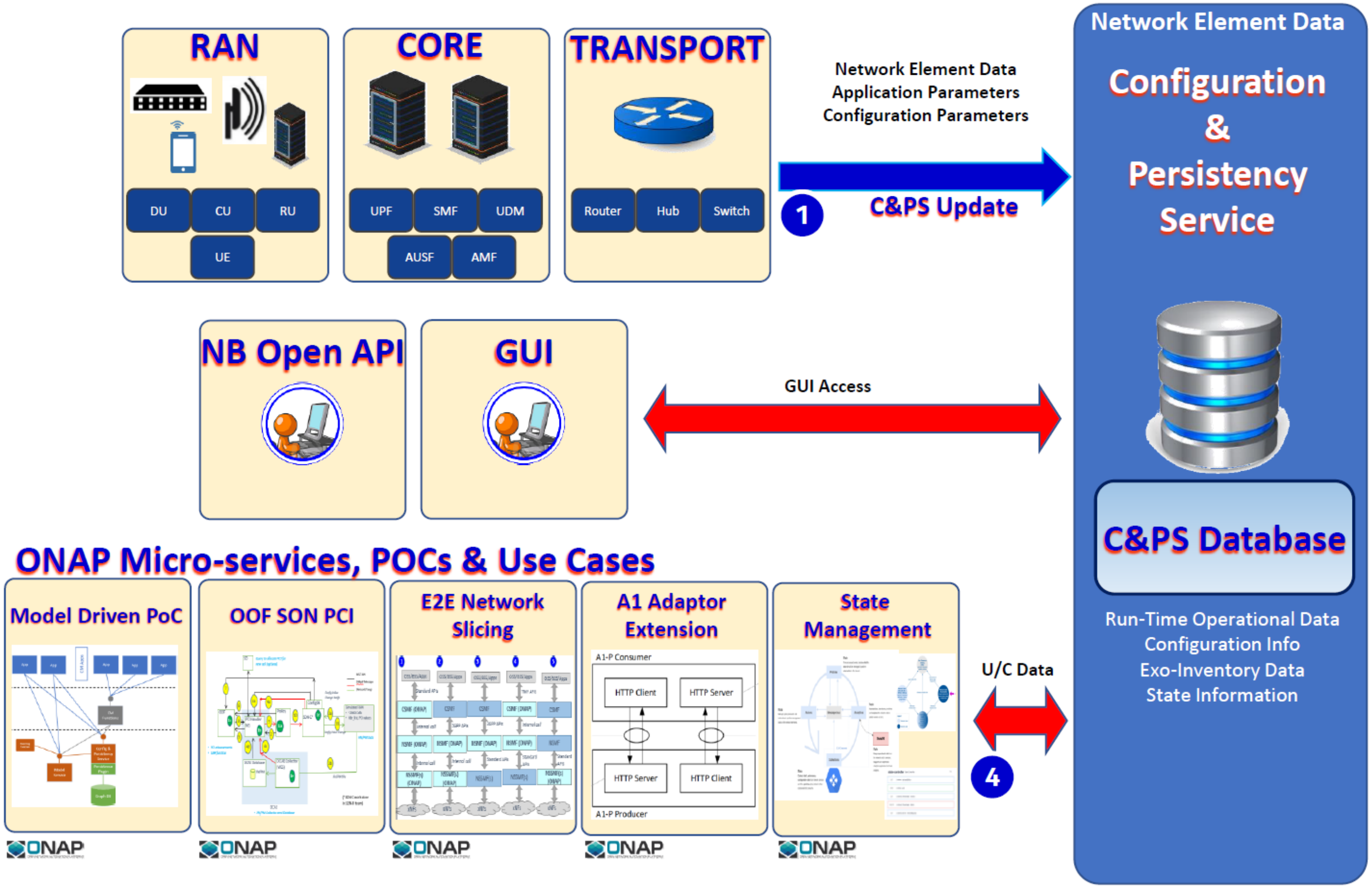
Context

Configuration Persistence Service in R8

TOPIC	PROBLEM	SOLUTION
Heterogeneous Data Sources	Reconciling Multiple domains, multiple vendors, multiple functions and multiple versions	Model Driven Persistence
Shared Data Access	Expensive IO operations that should be shared rather than duplicated	Share Data access through model ownership
Model Handling	Because of the heterogeneous data sources need streamlined approach to support models without having platform life cycle events.	Model Driven Persistence

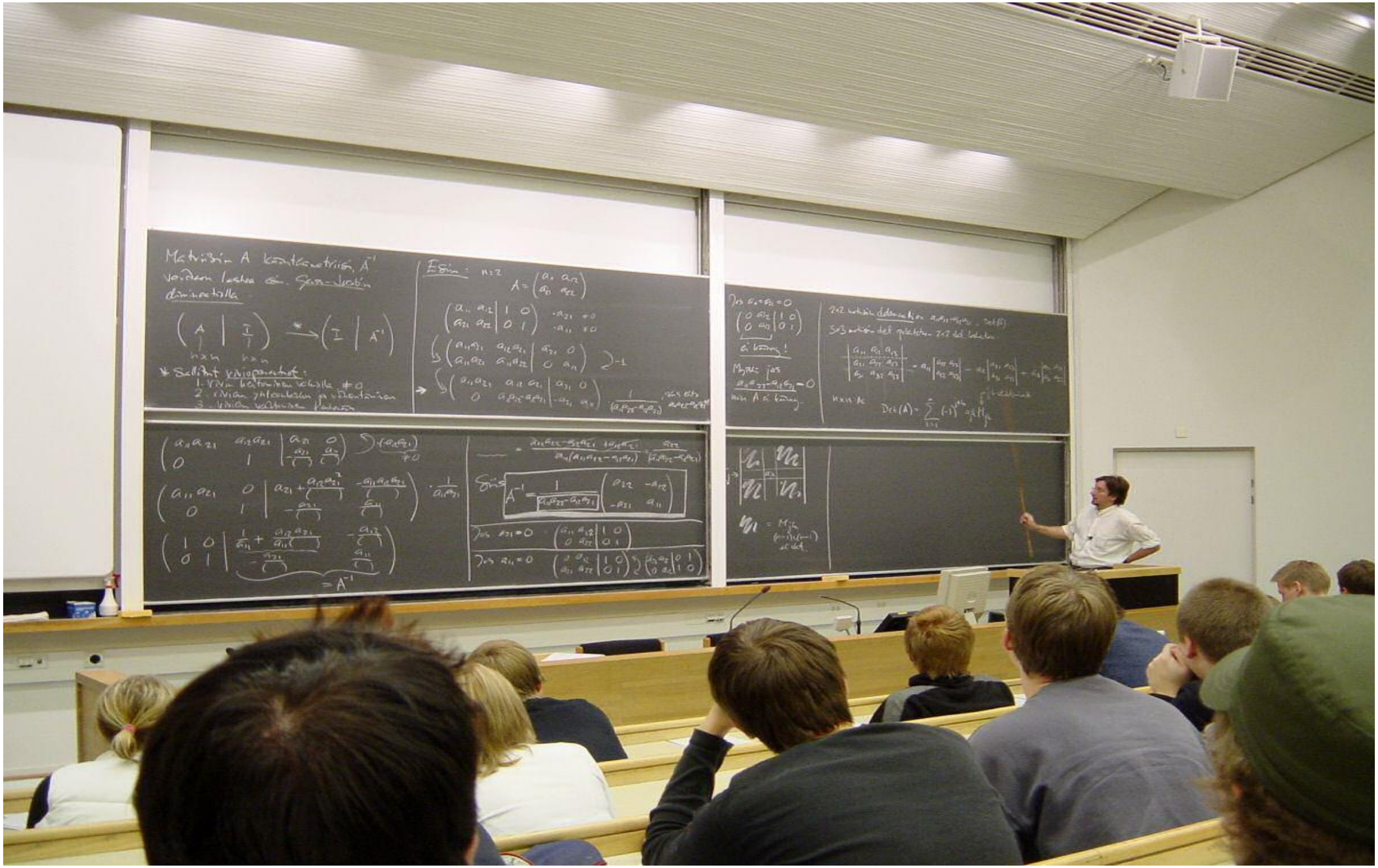
Configuration Persistence Service in R8

- Multiple vendors providing equipment / xNF
- Several telecom networks one ONAP platform
- Evolution of xNF and network standards
- ONAP applications and use cases competing for access to the same CM data



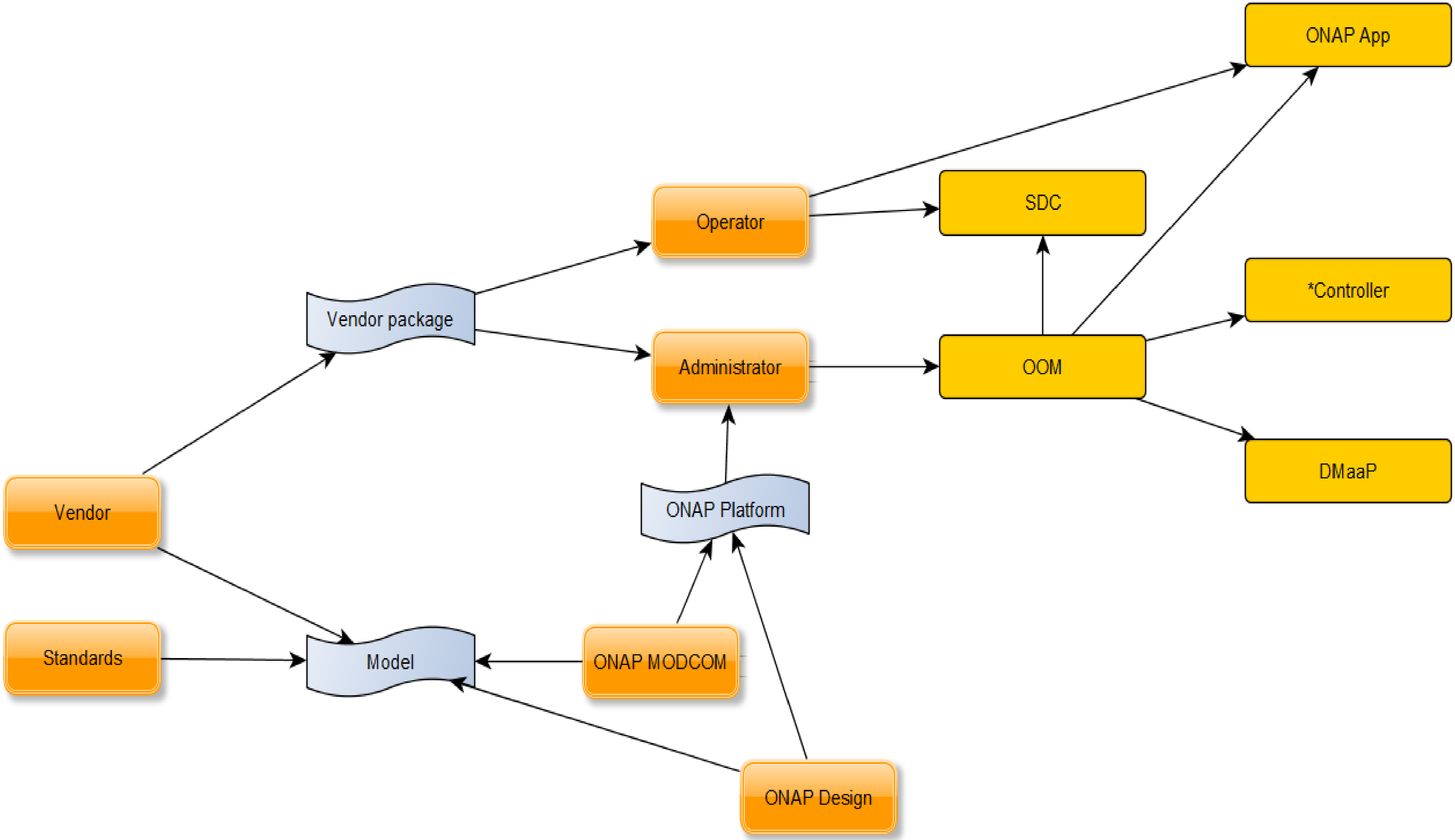
Configuration Persistence Service in R8

Shared data analogy of a teacher sharing knowledge with students


Esim: n=2
$A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$
A^{-1}

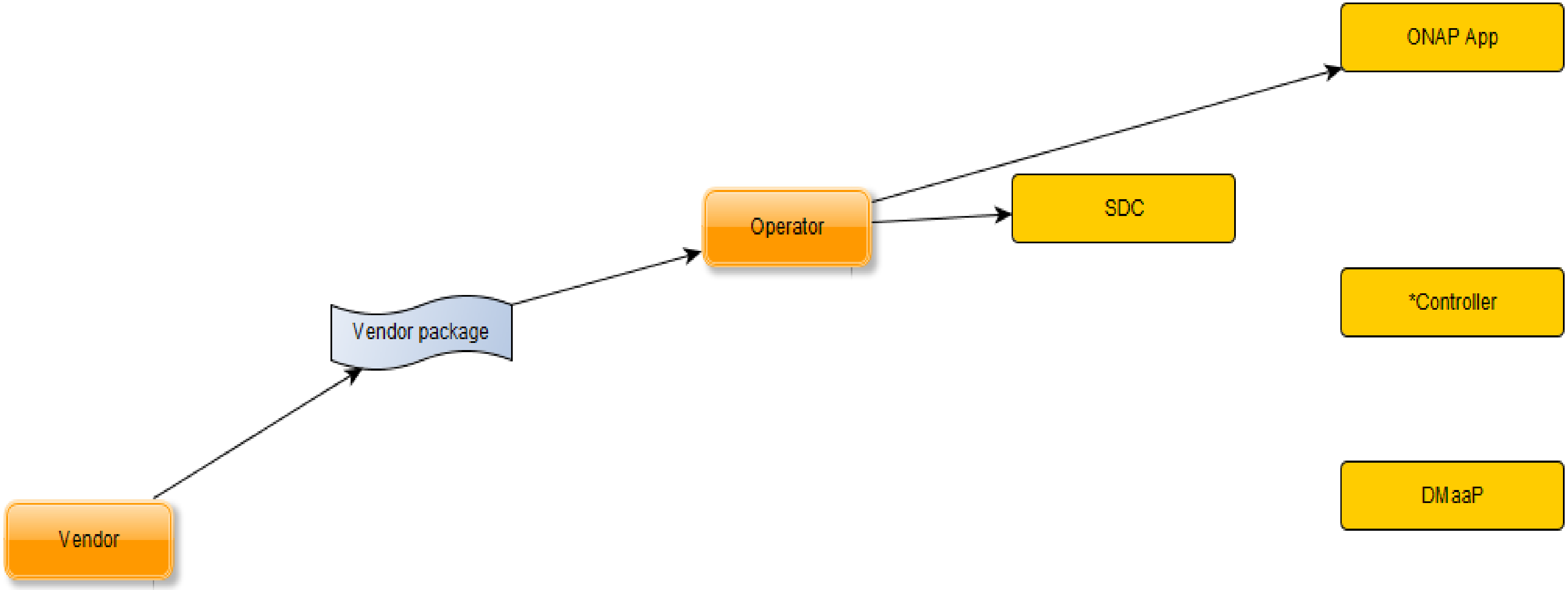
Configuration Persistence Service in R8

Model ingestion during ONAP design & run time



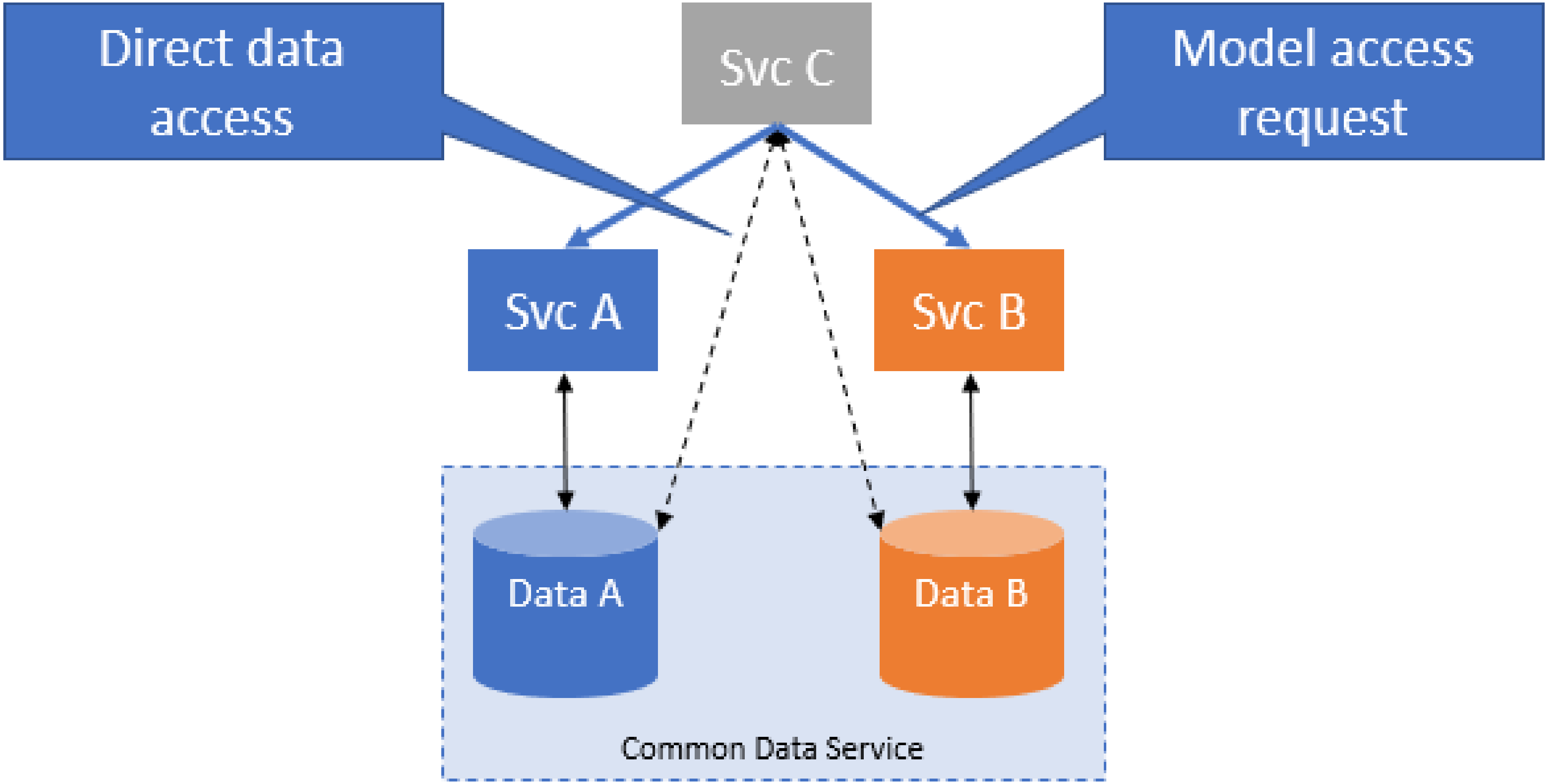
Configuration Persistence Service in R8

Model ingestion with C&PS solution (Run-time)

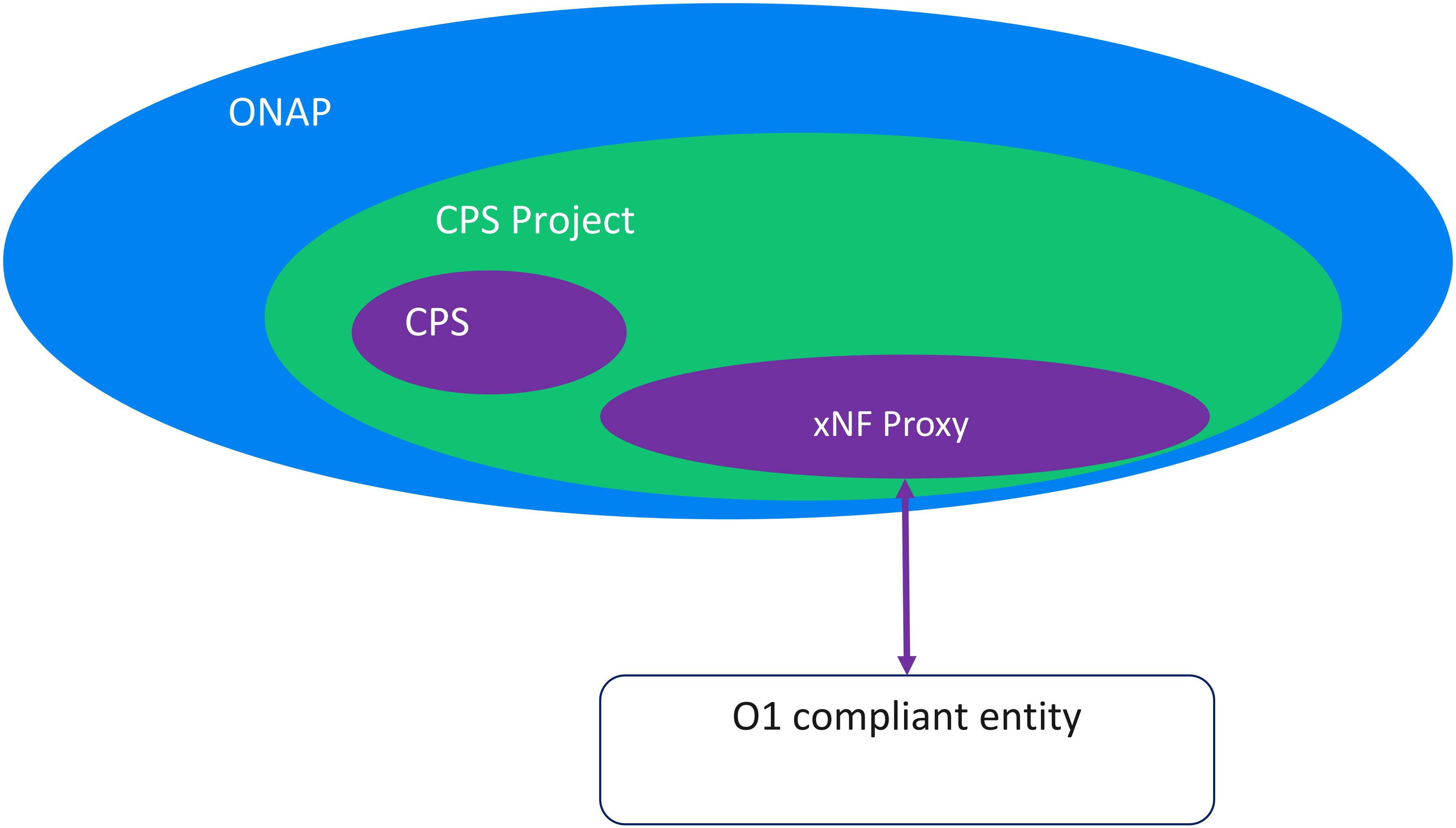


Configuration Persistence Service in R8

Permission & Ownership of data



Configuration Persistence Service in R8



ONAP

OPEN NETWORK AUTOMATION PLATFORM

Overview for
Configuration Persistence Service (CPS)

Business Case

Architecture S/C

Overview

Technical Flows

Configuration Persistence Service in R8

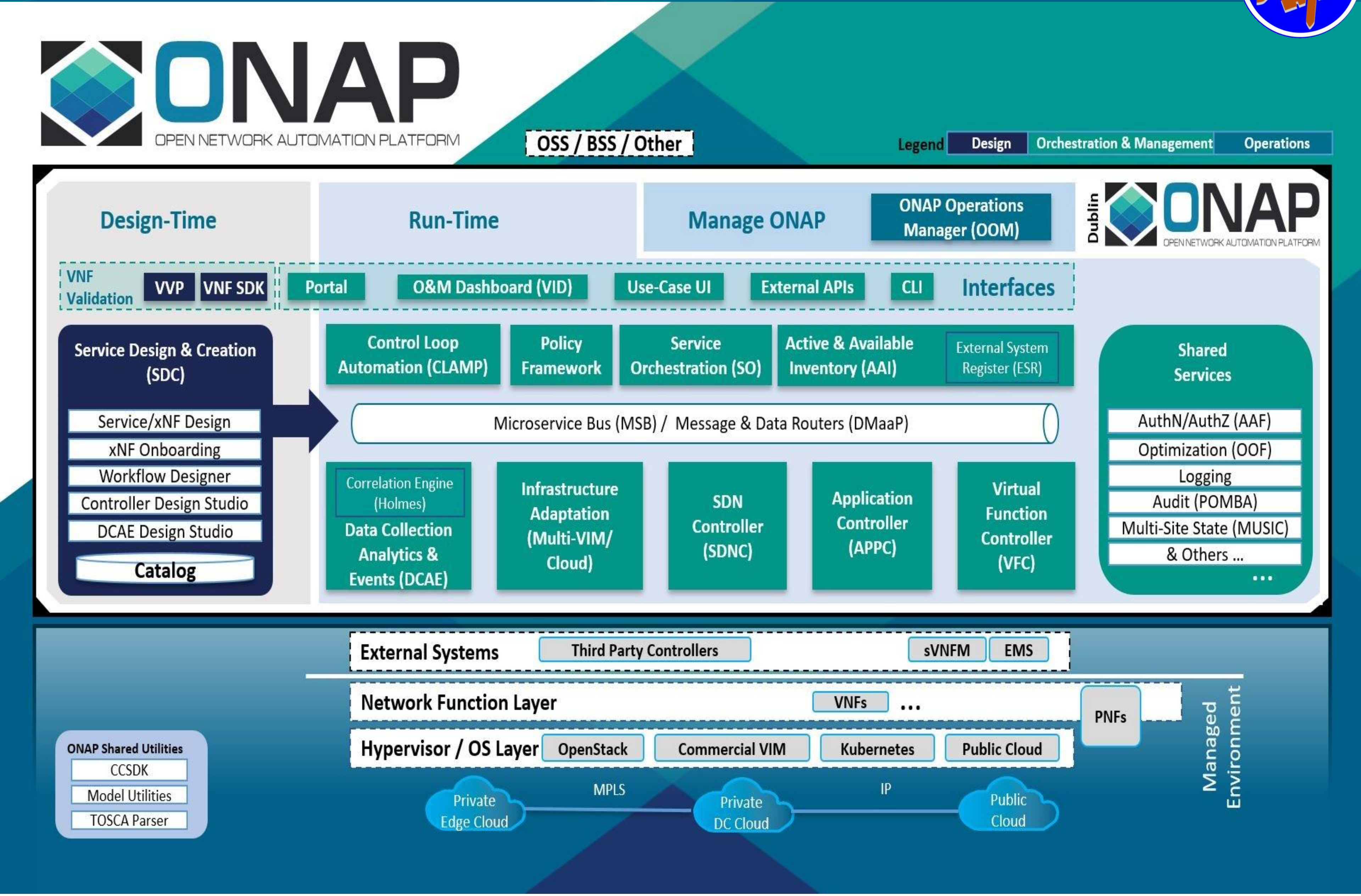
Executive Summary - The Configuration Persistence Service (CPS) is a *real-time service* that is designed to serve as a data repository for Run-time Network Element (configuration) data that needs to be persistent applicable to multiple domain (RAN, Transport, and Core). This was explored as a R7 PoC. *Focus on storing run-time DATA RELATED to NETWORK ELEMENT instances.* In R8, this is being proposed as a stand-alone project.

Business Impact - The ability for service operators to *visualize and manage network element data in a network (PNFs, VNFs, and logical constructs)* with ONAP is a critical business function because they are key Life Cycle Management (LCM) and OA&M operations. The project has business impacts to enhance the operation of data-handling within ONAP by providing efficient data layer services.

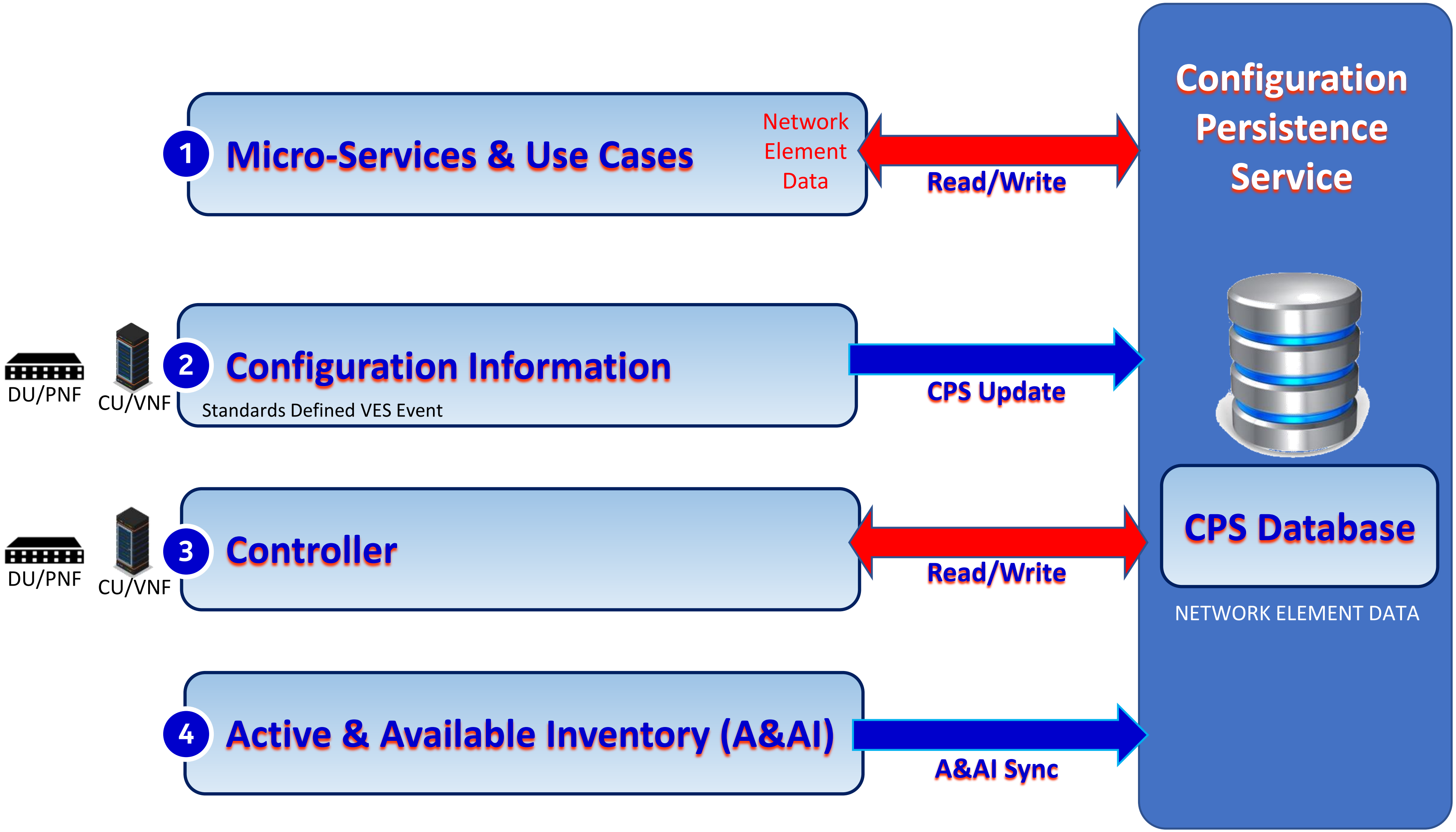
Business Markets - This project applies to any domain (wireless, transport, optical, and wireline) that ONAP may manage. It is not a market or geographical specific capability. It is expected that scaled ONAP installations such as Edge & Core ONAP deployments will also deploy the database across each installation.

Funding/Financial Impacts - This project represents a large potential Operating Expense (OPEX) savings for operators because of the ability to configure networks saving time and expenses.

CPS in Architecture Context



Configuration Persistence Service (CPS)



1 Micro-Services & Use Cases

Network Element Data

Read/Write

DU/PNF CU/VNF

2 Configuration Information

Standards Defined VES Event

CPS Update

DU/PNF CU/VNF

3 Controller

Read/Write

4 Active & Available Inventory (A&AI)

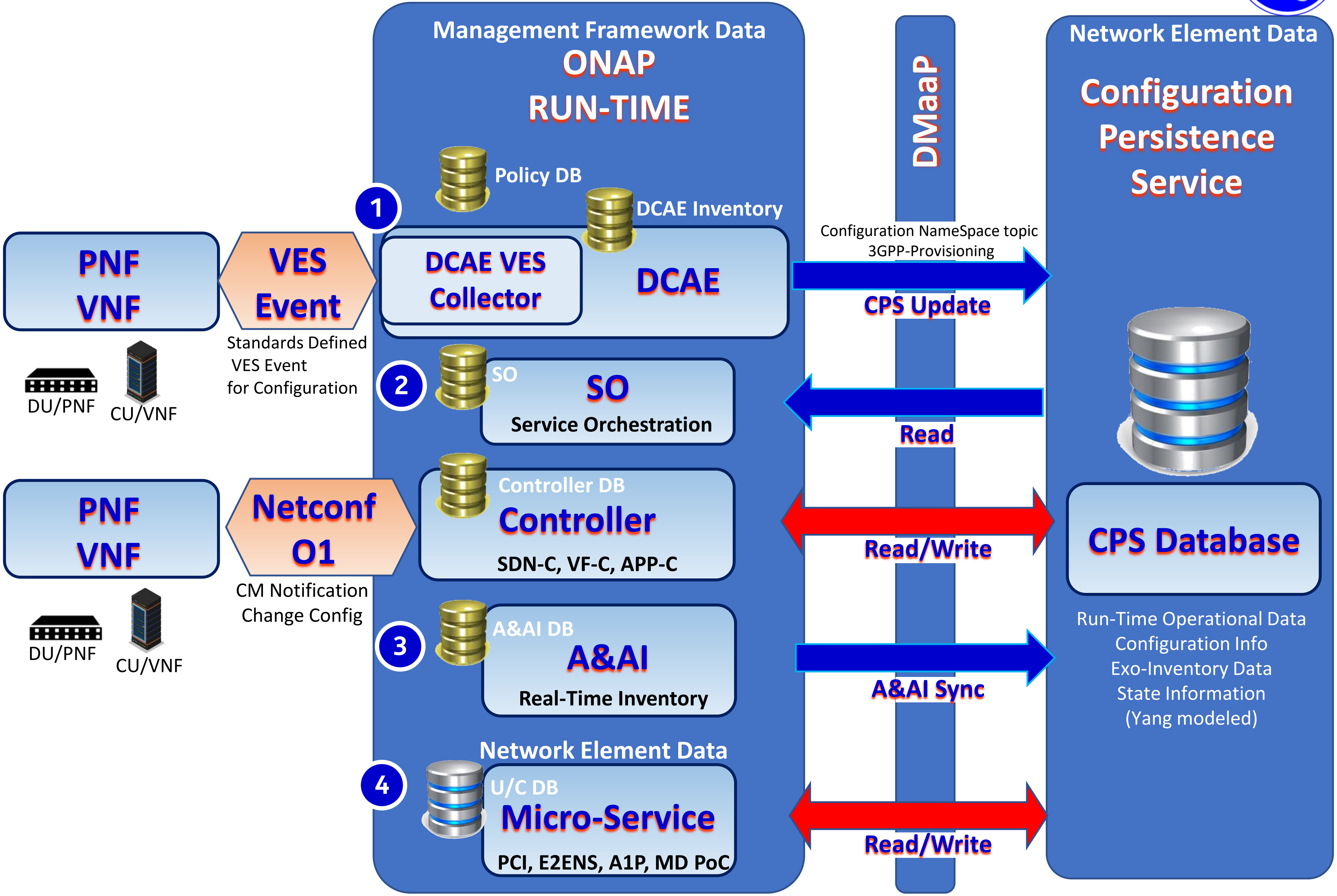
A&AI Sync

Configuration Persistence Service

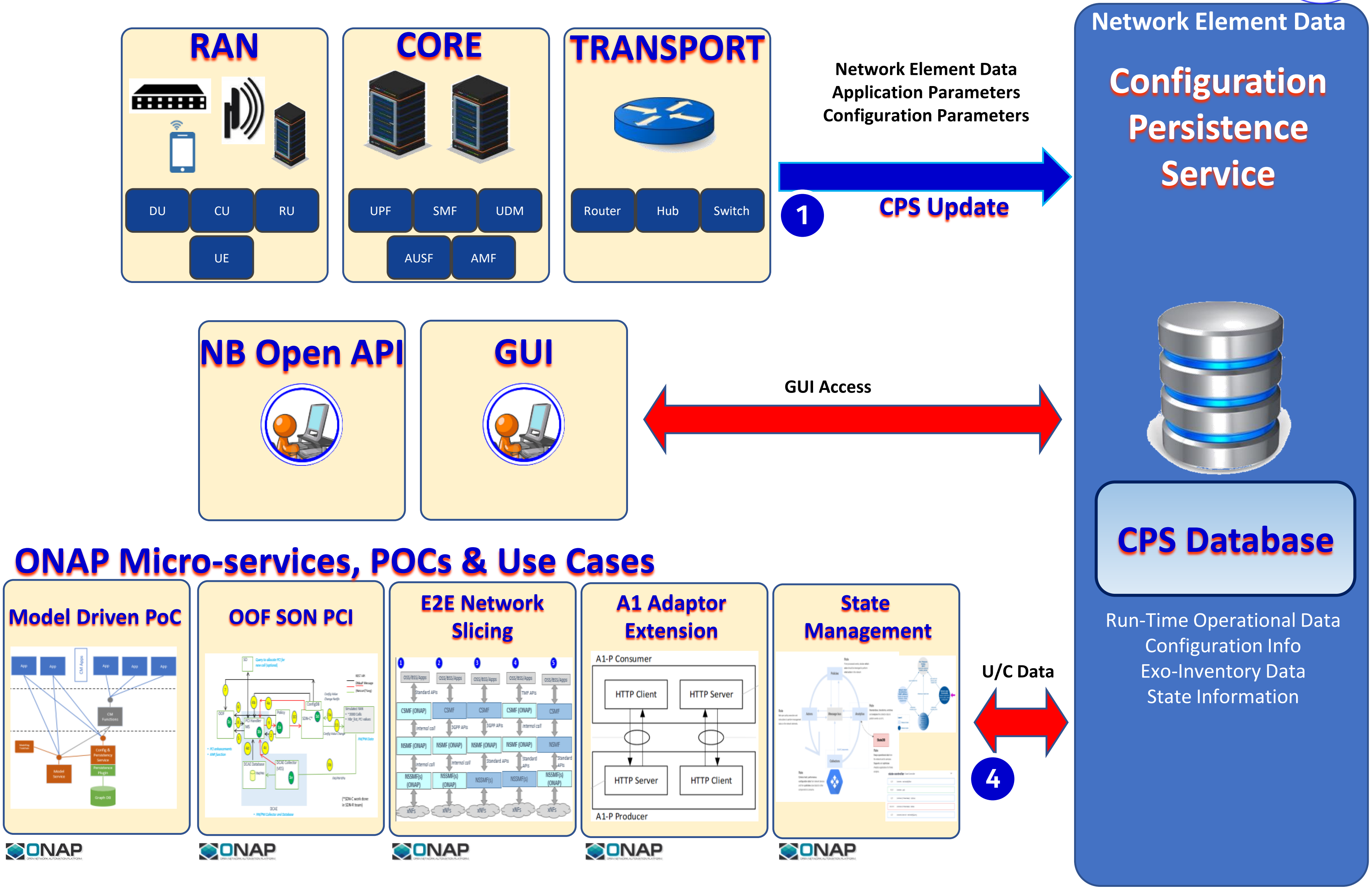
CPS Database

NETWORK ELEMENT DATA

Configuration Persistence Service (CPS)



Configuration Persistence Service (CPS)



RAN
- DU
- CU
- RU
- UE

CORE
- UPF
- SMF
- UDM
- AUSF
- AMF

TRANSPORT
- Router
- Hub
- Switch

Network Element Data
Application Parameters
Configuration Parameters

1 CPS Update

NB Open API

GUI

GUI Access

ONAP Micro-services, POCs & Use Cases

- Model Driven PoC
- OOF SON PCI
- E2E Network Slicing
- A1 Adaptor Extension
- State Management

A1-P Consumer: HTTP Client, HTTP Server
A1-P Producer: HTTP Server, HTTP Client

U/C Data

4

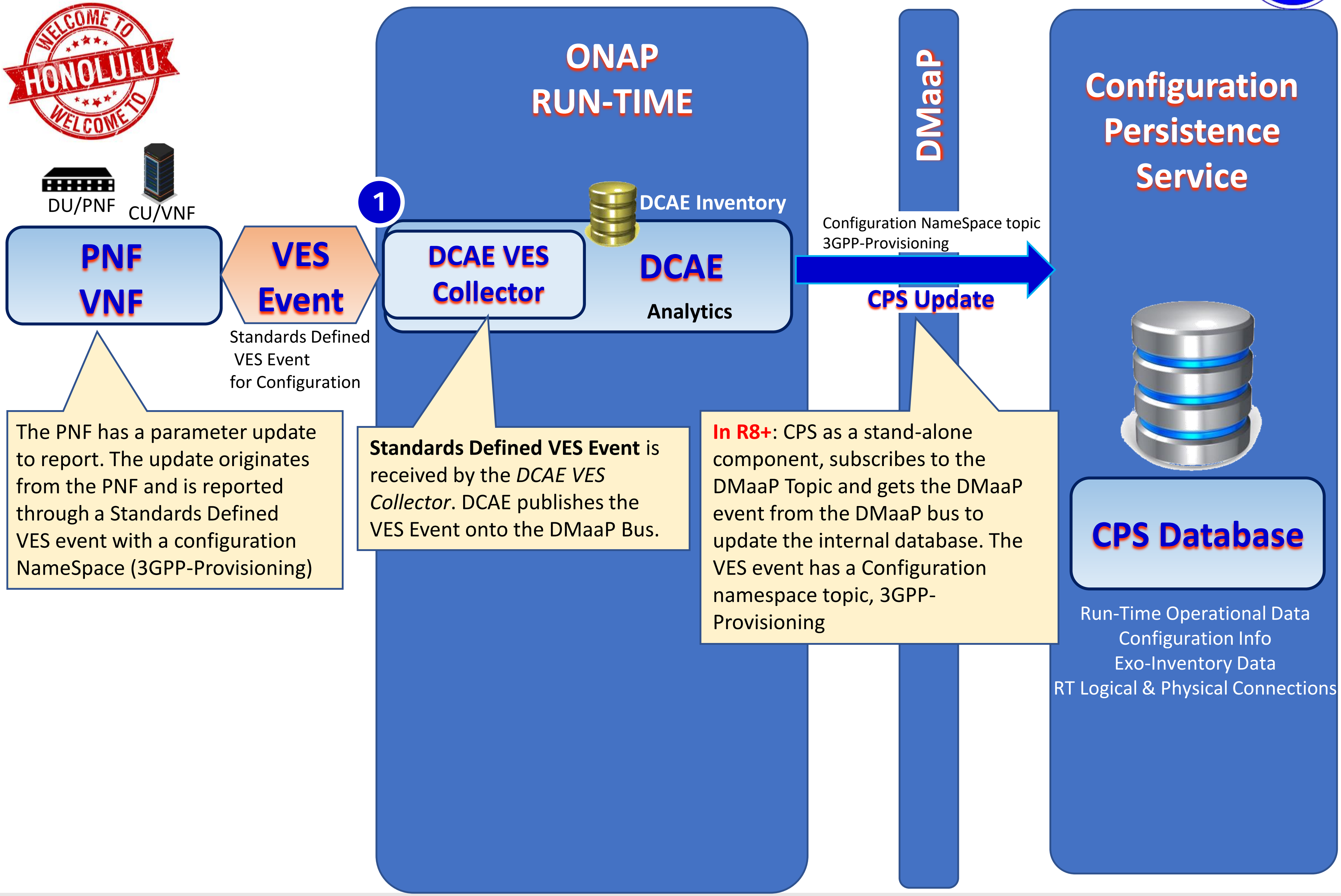
Network Element Data

Configuration Persistence Service

CPS Database

Run-Time Operational Data
Configuration Info
Exo-Inventory Data
State Information

CPS READING: PNF Reports Configuration

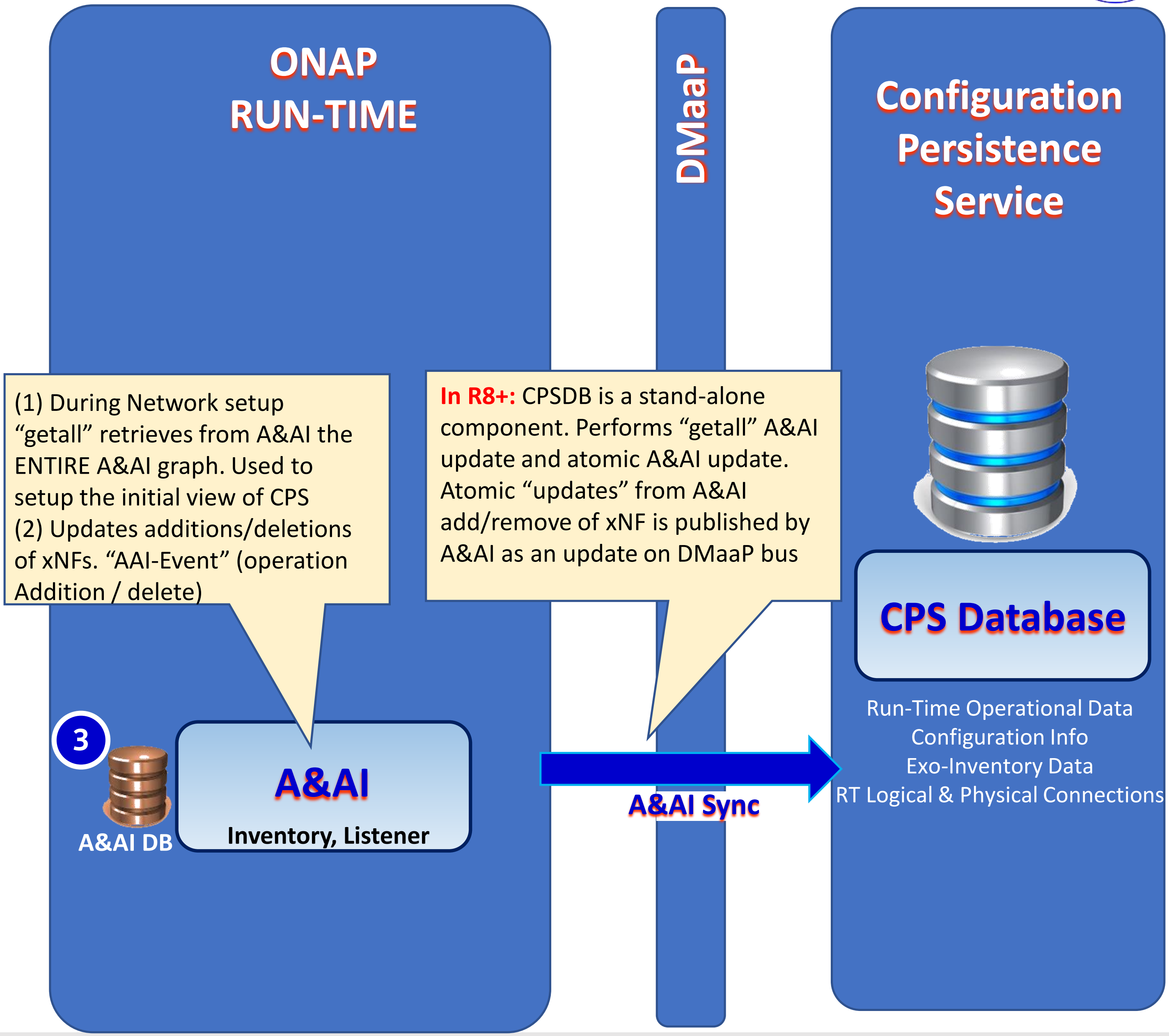


The PNF has a parameter update to report. The update originates from the PNF and is reported through a Standards Defined VES event with a configuration NameSpace (3GPP-Provisioning)

Standards Defined VES Event is received by the *DCAE VES Collector*. DCAE publishes the VES Event onto the DMaaP Bus.

In R8+: CPS as a stand-alone component, subscribes to the DMaaP Topic and gets the DMaaP event from the DMaaP bus to update the internal database. The VES event has a Configuration namespace topic, 3GPP-Provisioning

A&AI Synchronization



ONAP RUN-TIME

DMaaP

Configuration Persistence Service

(1) During Network setup "getall" retrieves from A&AI the ENTIRE A&AI graph. Used to setup the initial view of CPS
(2) Updates additions/deletions of xNFs. "AAI-Event" (operation Addition / delete)

In R8+: CPSDB is a stand-alone component. Performs "getall" A&AI update and atomic A&AI update. Atomic "updates" from A&AI add/remove of xNF is published by A&AI as an update on DMaaP bus

3

A&AI DB

A&AI

Inventory, Listener

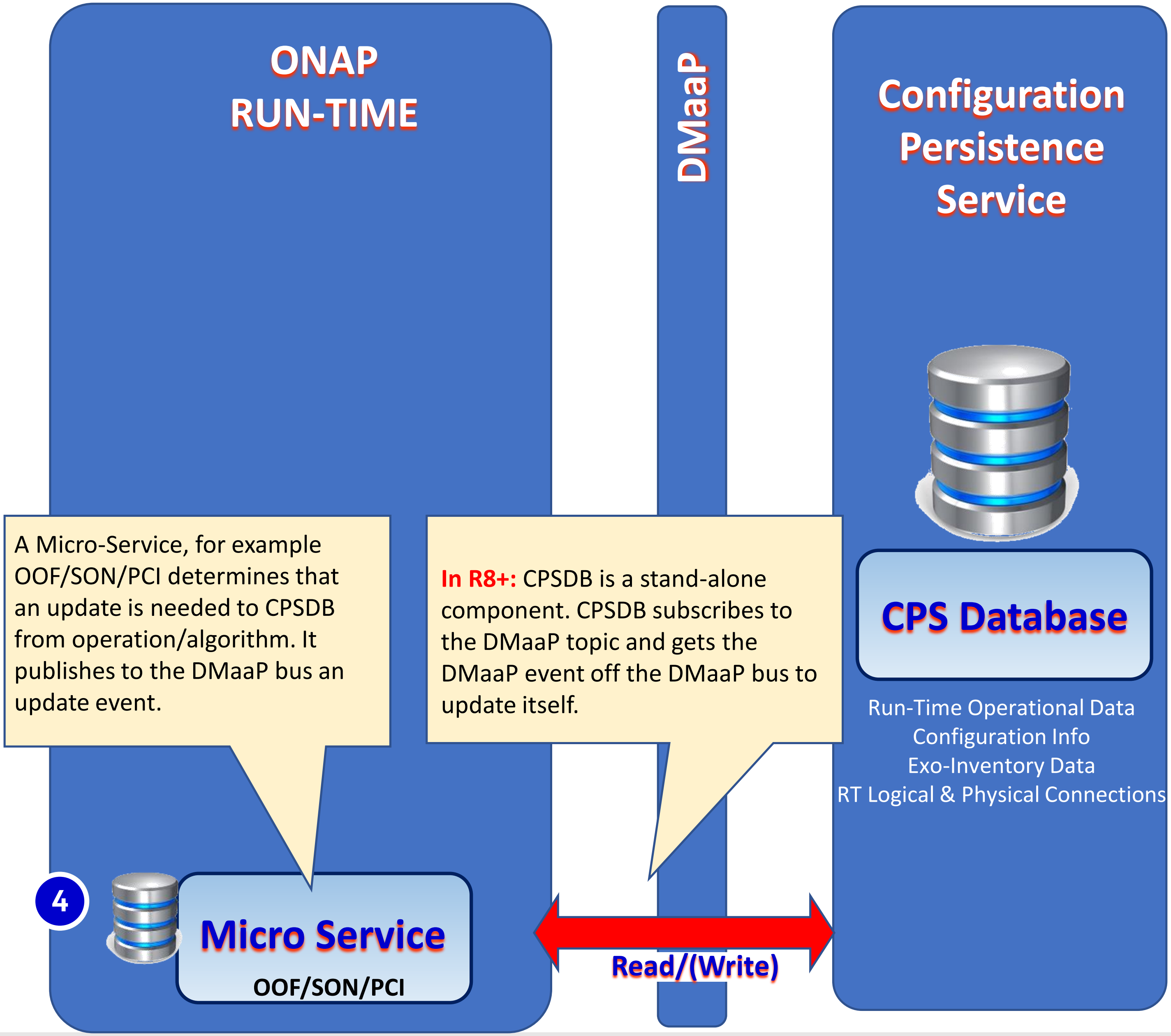
A&AI Sync

CPS Database

Run-Time Operational Data
Configuration Info
Exo-Inventory Data
RT Logical & Physical Connections

CPS WRITING: Micro Service Update

WELCOME TO HONOLULU WELCOME TO

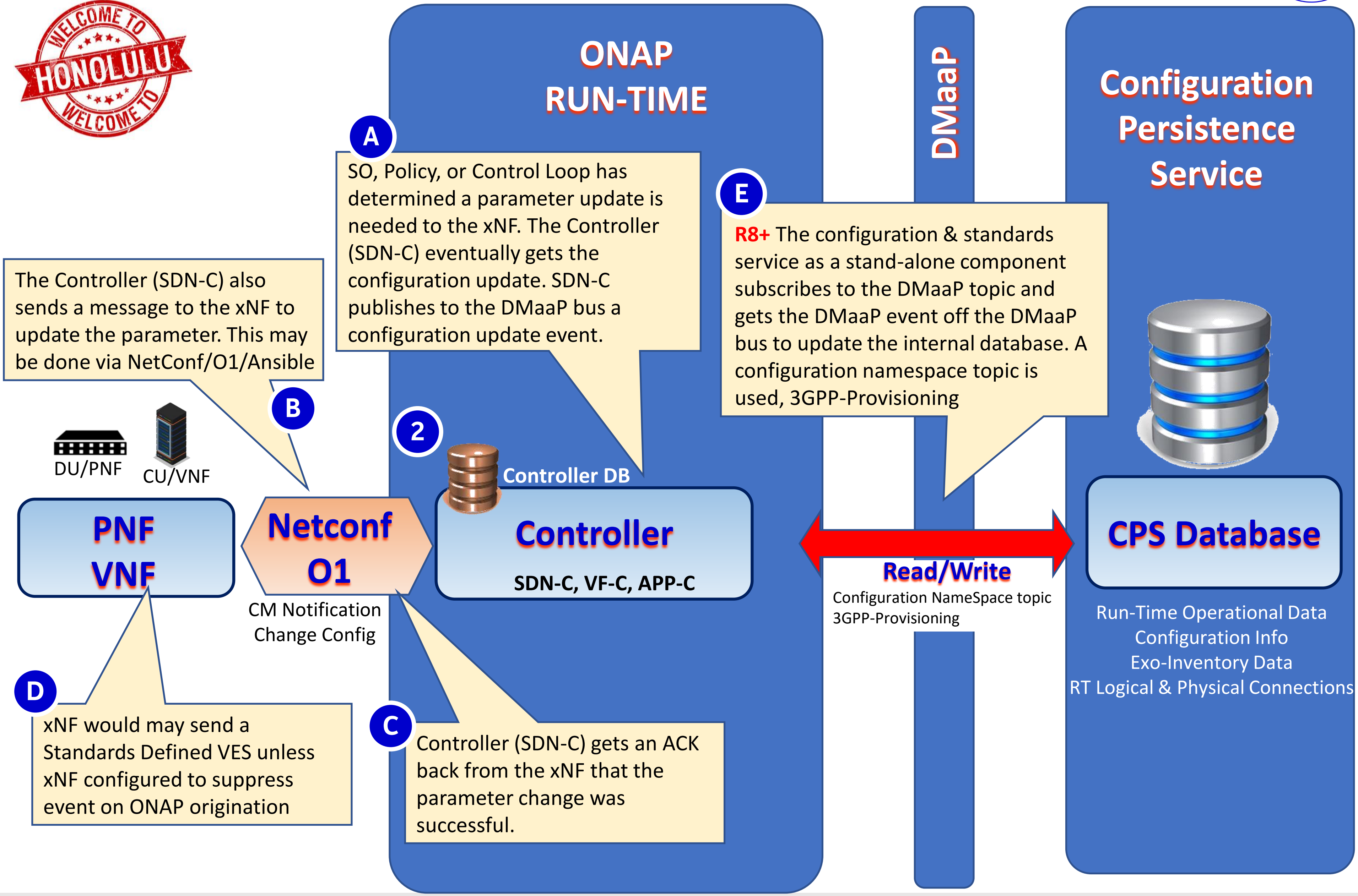


A Micro-Service, for example OOF/SON/PCI determines that an update is needed to CPSDB from operation/algorithm. It publishes to the DMaaP bus an update event.

In R8+: CPSDB is a stand-alone component. CPSDB subscribes to the DMaaP topic and gets the DMaaP event off the DMaaP bus to update itself.

CPS WRITING: From Controller SDN-C

WELCOME TO HONOLULU WELCOME TO



ONAP RUN-TIME

DMaaP

Configuration Persistence Service

A — SO, Policy, or Control Loop has determined a parameter update is needed to the xNF. The Controller (SDN-C) eventually gets the configuration update. SDN-C publishes to the DMaaP bus a configuration update event.

B — The Controller (SDN-C) also sends a message to the xNF to update the parameter. This may be done via NetConf/O1/Ansible

C — Controller (SDN-C) gets an ACK back from the xNF that the parameter change was successful.

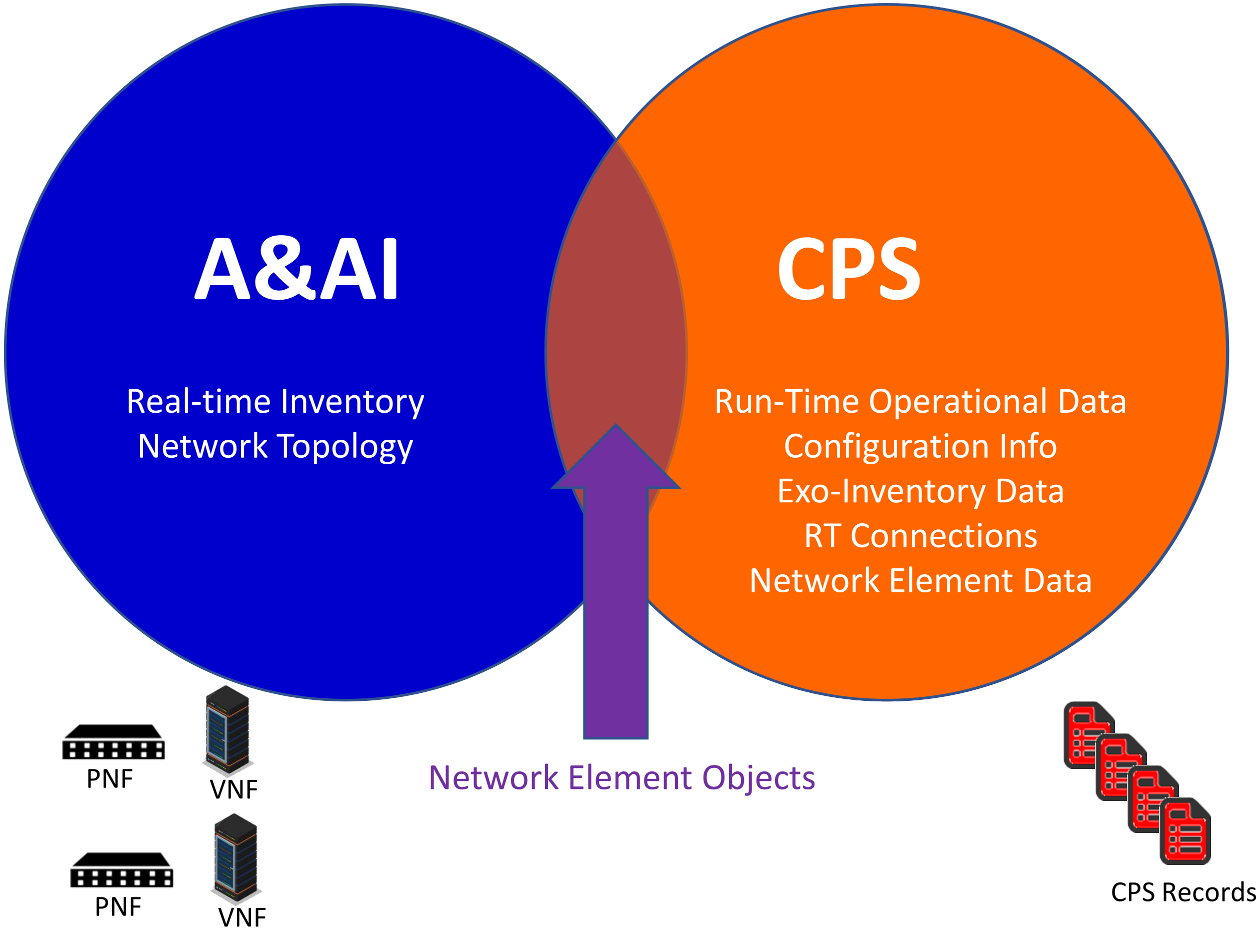
D — xNF would may send a Standards Defined VES unless xNF configured to suppress event on ONAP origination

E — R8+ The configuration & standards service as a stand-alone component subscribes to the DMaaP topic and gets the DMaaP event off the DMaaP bus to update the internal database. A configuration namespace topic is used, 3GPP-Provisioning

Run-Time Operational Data
Configuration Info
Exo-Inventory Data
RT Logical & Physical Connections

A&AI vs CPS

Concepts – A&AI conceptually stores Real-time inventory view of connected and "topology" of xNFs that ONAP sees. CPS stores Network Element Data. A&AI and CPS overlaps because they both need to know about Network Element objects so that can managed & orchestrated.



ONAP
OPEN NETWORK AUTOMATION PLATFORM

R7 Model Driven Proof of Concept for Configuration Persistence Service (CPS)

Proof of Concept

Proof Points for PoC

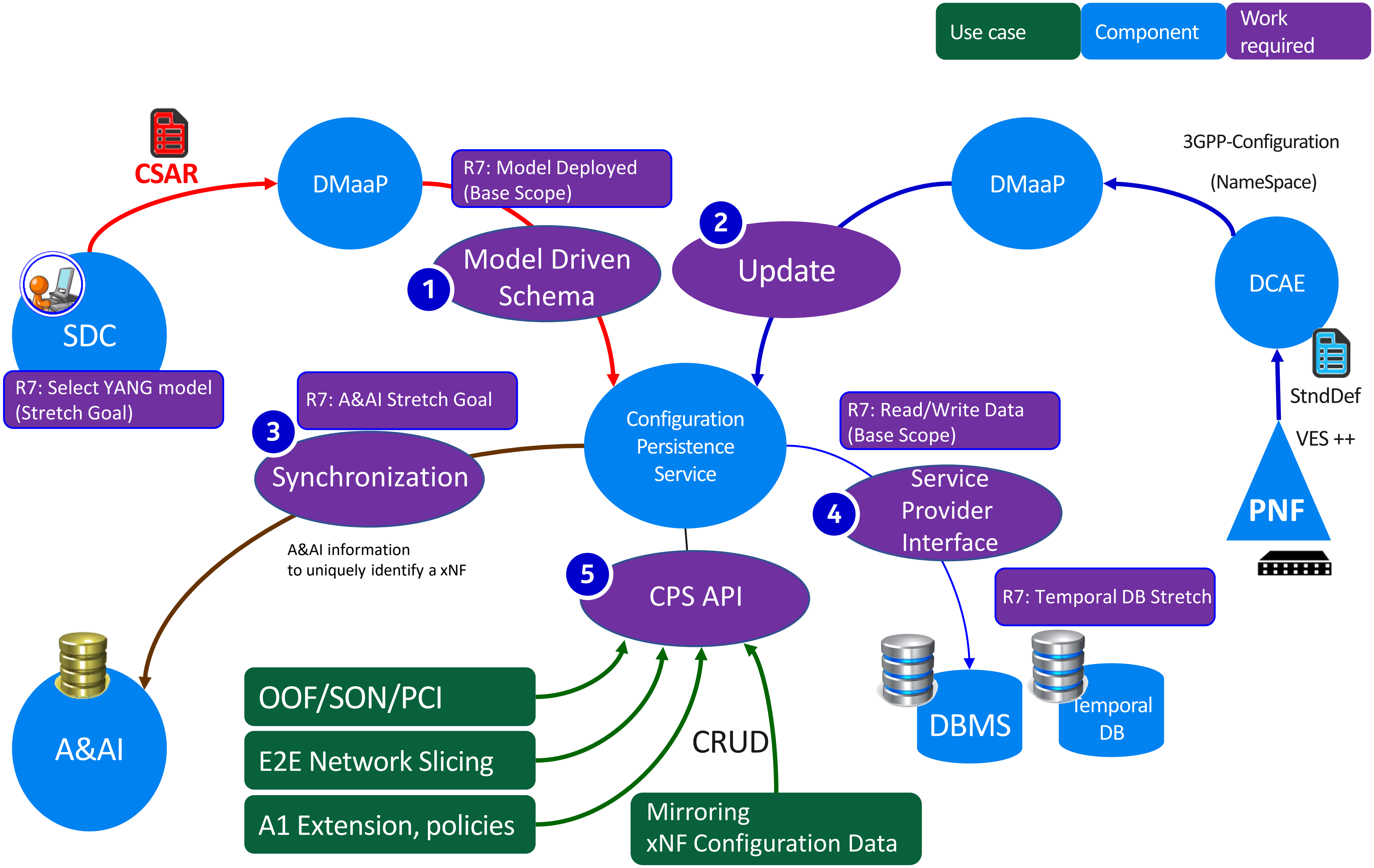
1. Demonstrate write/read operations for YANG data fragments using CPS and store them in a DB with a very simple generic schema

2. Demonstrate ability to deploy / upgrade YANG models at run-time

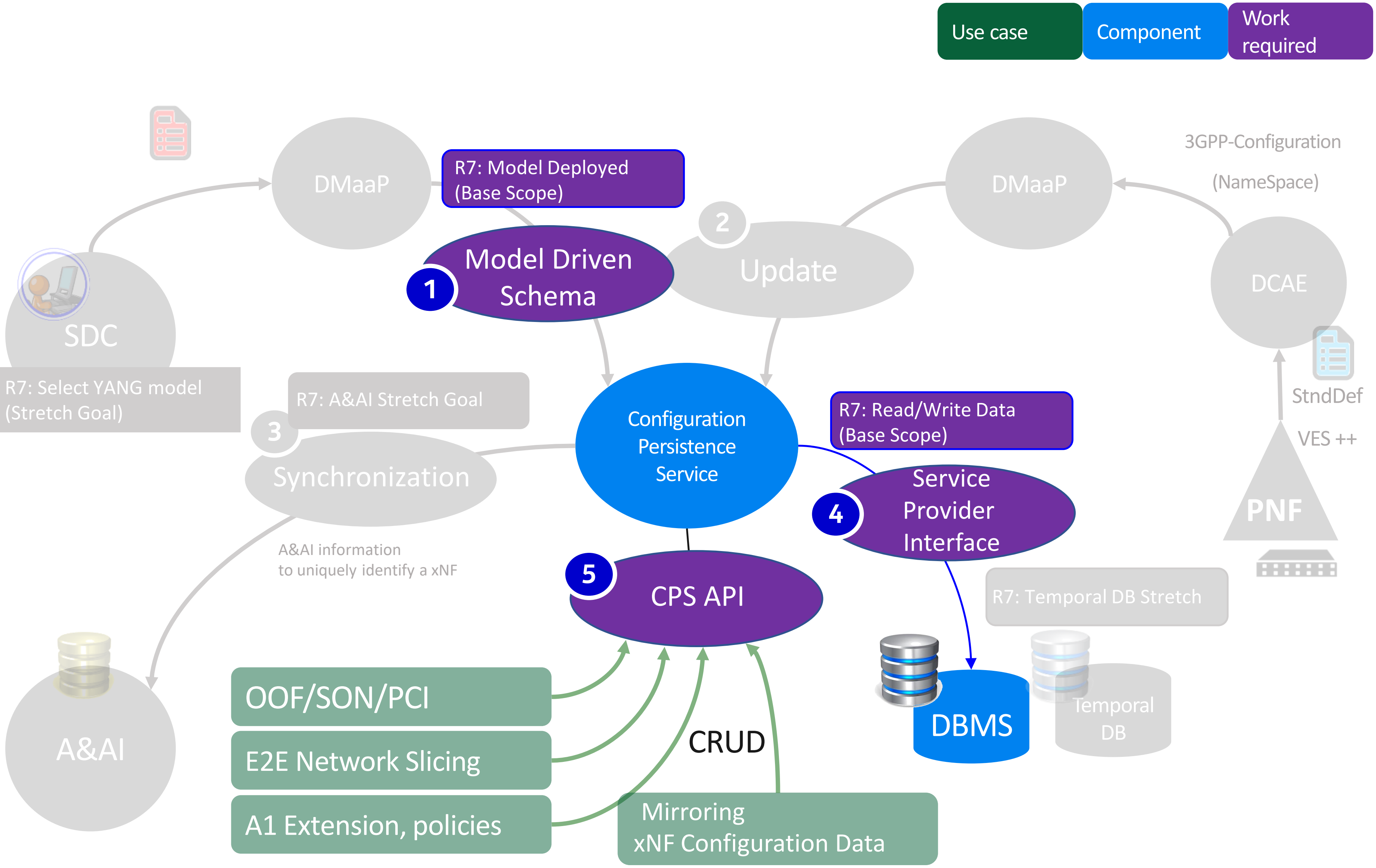
3. Demonstrate CPS behavior driven by YANG model

4. Provide architecture vision and roadmap for a target architecture, supported use cases, non-functional requirements towards an ONAP Project

R7 Model Driven CPS PoC



R7 Model Driven CPS PoC



Community Involvement & Resourcing

- Weekly ONAP meeting hosted by Ben Cheung (Nokia)
- PoC discussed with and approved by TSC
- ONAP R8 Honolulu Project proposal accepted
  - Use Cases & requirements : Configuration Persistence Service (CPS)

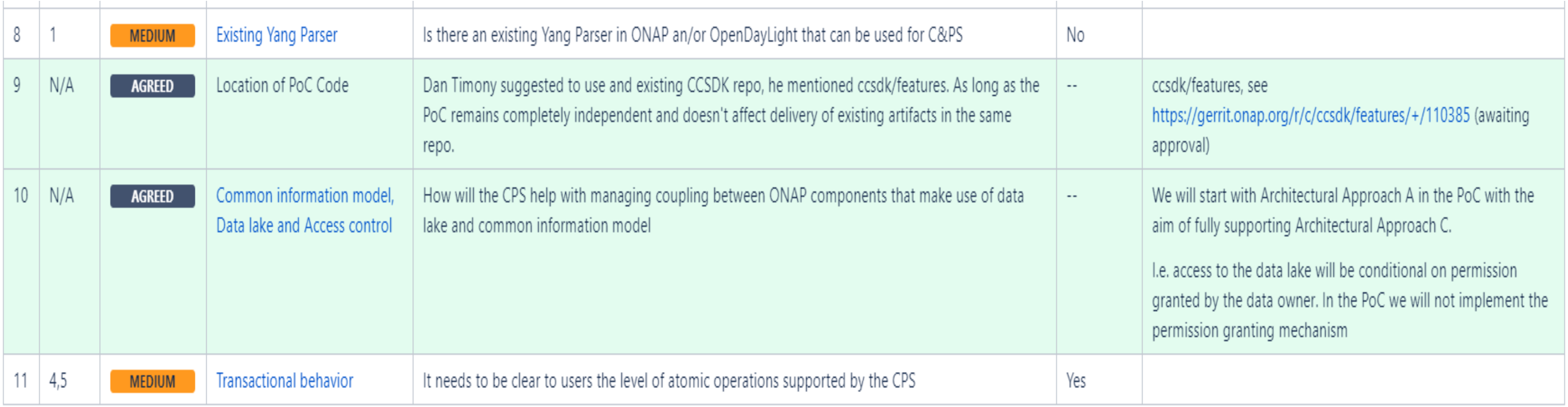
WELCOME TO HONOLULU WELCOME TO

- CPS presentations in many fora, O-Ran PlugFest, LFN Design Developer Forum (DDF) etc.
- Resourcing
  - Ericsson 3 full-time developers
  - Bell Canada 3 part-time developers
- Jira backlog, team ceremonies like scrums etc. established
  - Developers Landing Page

PoC Ways of Working

- Base : N/A (new code)
  Main dependency : ODL Yang Tools 5.x (probably)
  https://javadoc.io/doc/org.opendaylight.yangtools

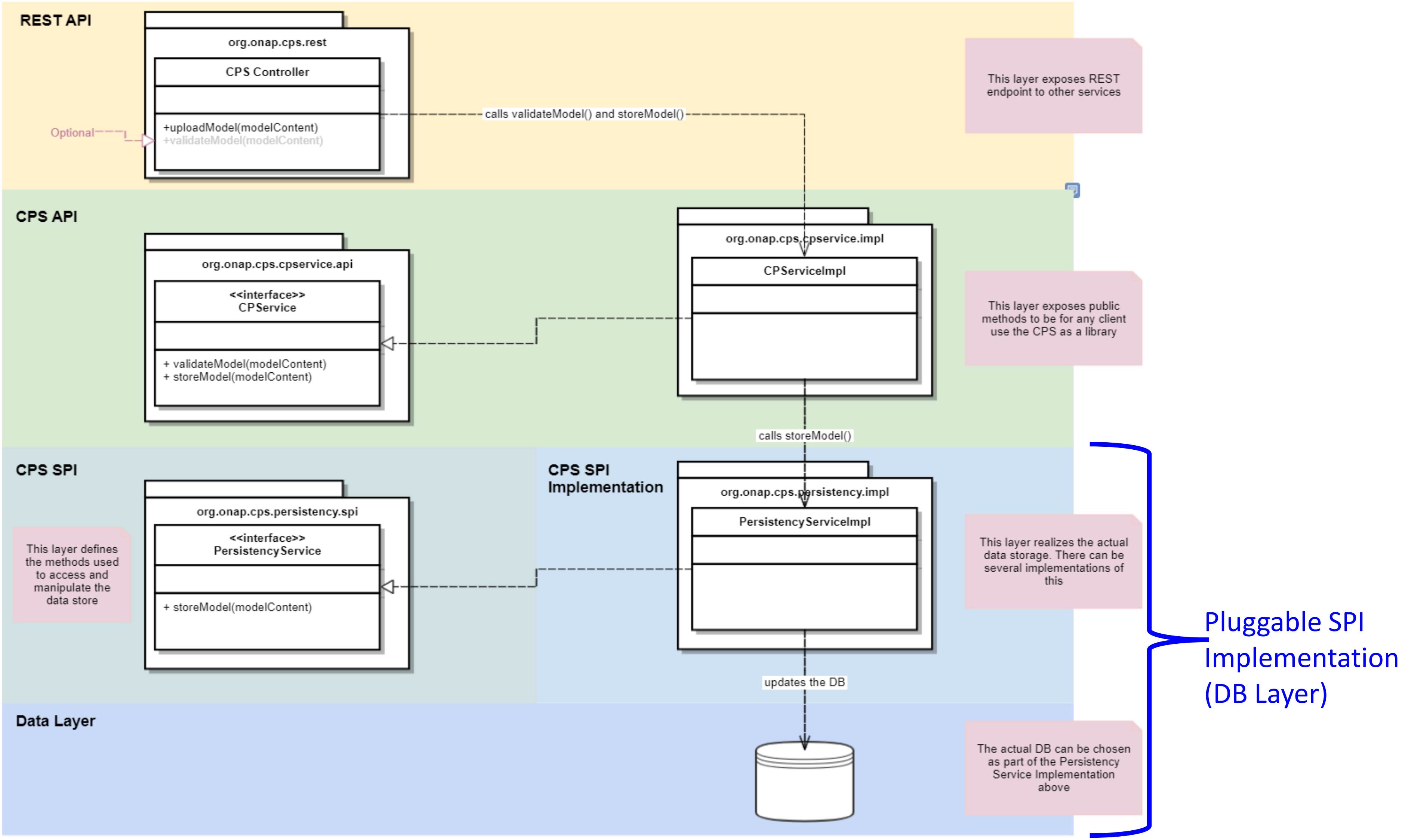
- Design and Architecture discussions ongoing
  https://wiki.onap.org/display/DW/Issues+decisions+and+assumptions



8	1	MEDIUM	Existing Yang Parser	Is there an existing Yang Parser in ONAP an/or OpenDayLight that can be used for C&PS	No	
9	N/A	AGREED	Location of PoC Code	Dan Timony suggested to use and existing CCSDK repo, he mentioned ccsdk/features. As long as the PoC remains completely independent and doesn't affect delivery of existing artifacts in the same repo.	--	ccsdk/features, see https://gerrit.onap.org/r/c/ccsdk/features/+/110385 (awaiting approval)
10	N/A	AGREED	Common information model, Data lake and Access control	How will the CPS help with managing coupling between ONAP components that make use of data lake and common information model	--	We will start with Architectural Approach A in the PoC with the aim of fully supporting Architectural Approach C. I.e. access to the data lake will be conditional on permission granted by the data owner. In the PoC we will not implement the permission granting mechanism
11	4,5	MEDIUM	Transactional behavior	It needs to be clear to users the level of atomic operations supported by the CPS	Yes	

R7 Model Driven CPS PoC

Architecture designed & implemented



REST API

org.onap.cps.rest

CPS Controller

+uploadModel(modelContent)
+validateModel(modelContent)

Optional

calls validateModel() and storeModel()

This layer exposes REST endpoint to other services

CPS API

org.onap.cps.cpservice.api

<<interface>>
CPService

+ validateModel(modelContent)
+ storeModel(modelContent)

org.onap.cps.cpservice.impl

CPServiceImpl

This layer exposes public methods to be for any client use the CPS as a library

calls storeModel()

CPS SPI

This layer defines the methods used to access and manipulate the data store

org.onap.cps.persistency.spi

<<interface>>
PersistencyService

+ storeModel(modelContent)

CPS SPI Implementation

org.onap.cps.persistency.impl

PersistencyServiceImpl

This layer realizes the actual data storage. There can be several implementations of this

updates the DB

Data Layer

The actual DB can be chosen as part of the Persistency Service Implementation above

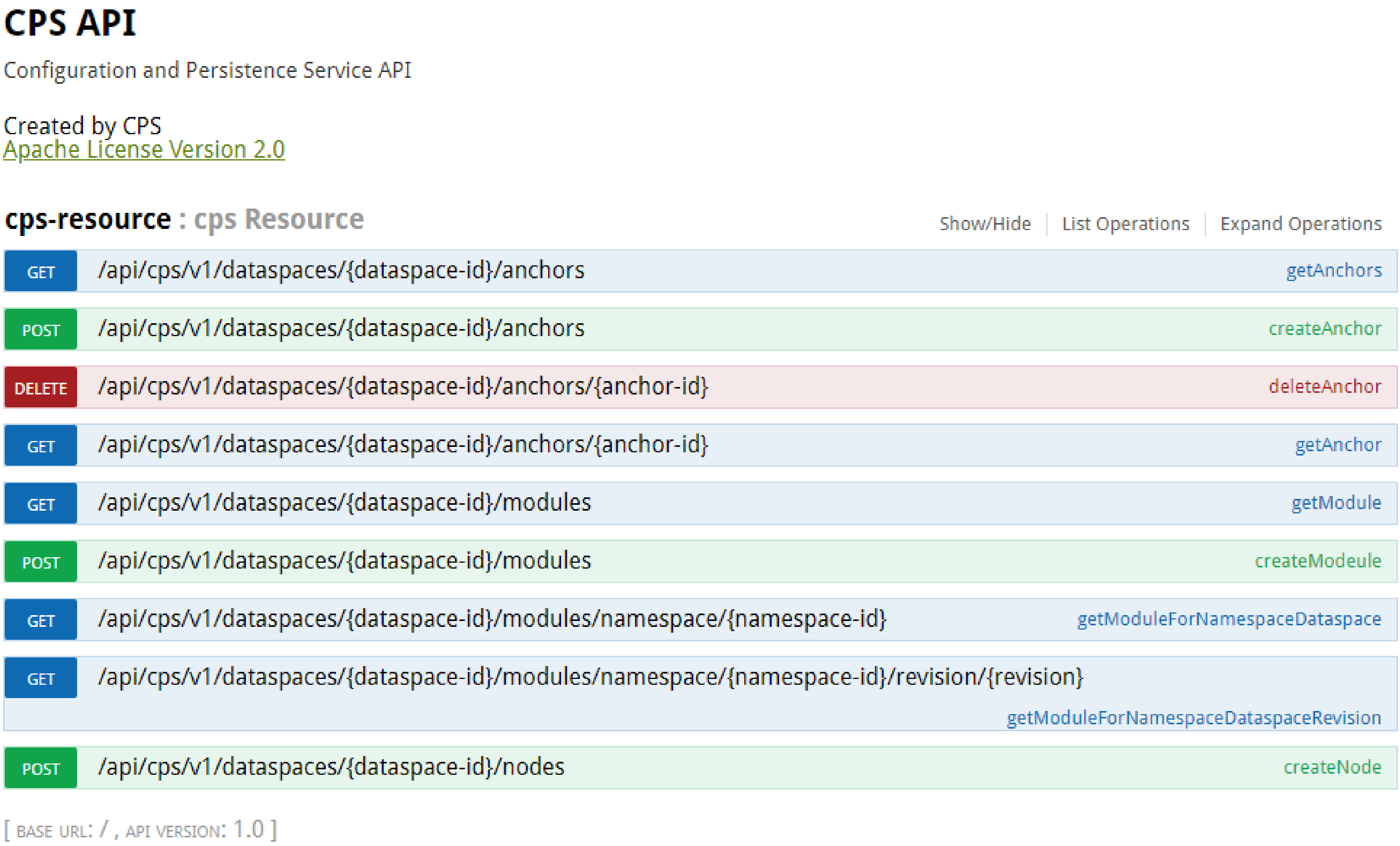
Pluggable SPI Implementation (DB Layer)

R7 Model Driven CPS PoC

Implementation Status

- PoC being developed on Nordix Branch to avoid impacts on production code in CCSDK Repo
- Project structure set up completed
  - Following ONAP standards and conventions
  - Production code Quality (SonarQube & Checkstyle)
- ODL Yang Tools integrated
  - Parsing of Yang models
  - Parsing of Yang data, JSON or XML with validation for given Yang model
- First E2E iteration of 2 use cases completed
  - Upload & Validate Yang Model (part of proof point 2)
  - Upload JSON Data Object (part of proof point 1

REST API Proposal



CPS API

Configuration and Persistence Service API

Created by CPS
Apache License Version 2.0

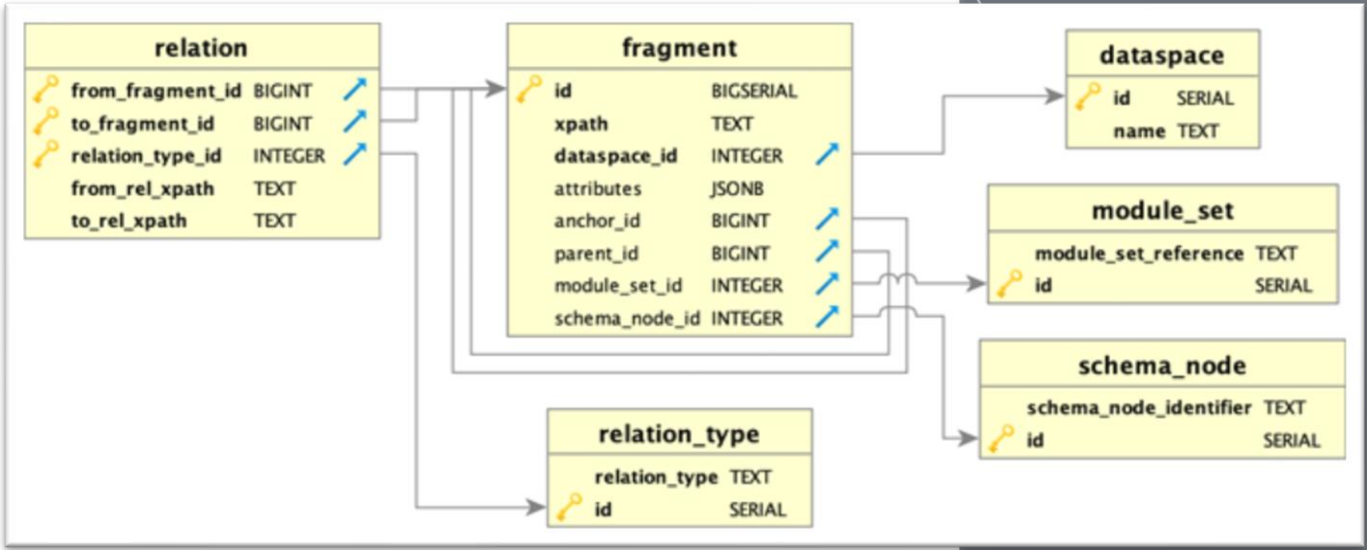
cps-resource : cps Resource

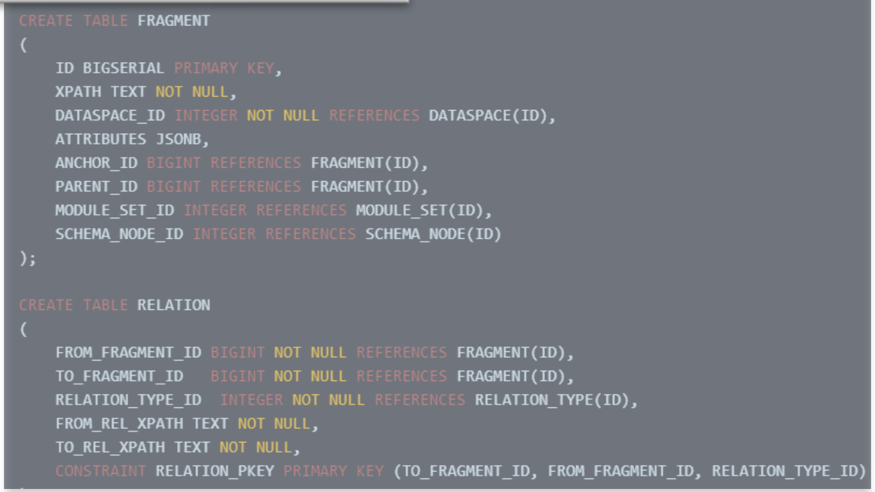
Show/Hide | List Operations | Expand Operations

GET	/api/cps/v1/dataspaces/{dataspace-id}/anchors	getAnchors
POST	/api/cps/v1/dataspaces/{dataspace-id}/anchors	createAnchor
DELETE	/api/cps/v1/dataspaces/{dataspace-id}/anchors/{anchor-id}	deleteAnchor
GET	/api/cps/v1/dataspaces/{dataspace-id}/anchors/{anchor-id}	getAnchor
GET	/api/cps/v1/dataspaces/{dataspace-id}/modules	getModule
POST	/api/cps/v1/dataspaces/{dataspace-id}/modules	createModeule
GET	/api/cps/v1/dataspaces/{dataspace-id}/modules/namespace/{namespace-id}	getModuleForNamespaceDataspace
GET	/api/cps/v1/dataspaces/{dataspace-id}/modules/namespace/{namespace-id}/revision/{revision}	getModuleForNamespaceDataspaceRevision
POST	/api/cps/v1/dataspaces/{dataspace-id}/nodes	createNode

[BASE URL: / , API VERSION: 1.0]

Generic Schema Design Finalized





```
CREATE TABLE RELATION_TYPE
(
```

```
CREATE TABLE FRAGMENT
(
    ID BIGSERIAL PRIMARY KEY,
    XPATH TEXT NOT NULL,
    DATASPACE_ID INTEGER NOT NULL REFERENCES DATASPACE(ID),
    ATTRIBUTES JSONB,
    ANCHOR_ID BIGINT REFERENCES FRAGMENT(ID),
    PARENT_ID BIGINT REFERENCES FRAGMENT(ID),
    MODULE_SET_ID INTEGER REFERENCES MODULE_SET(ID),
    SCHEMA_NODE_ID INTEGER REFERENCES SCHEMA_NODE(ID)
);

CREATE TABLE RELATION
(
    FROM_FRAGMENT_ID BIGINT NOT NULL REFERENCES FRAGMENT(ID),
    TO_FRAGMENT_ID   BIGINT NOT NULL REFERENCES FRAGMENT(ID),
    RELATION_TYPE_ID  INTEGER NOT NULL REFERENCES RELATION_TYPE(ID),
    FROM_REL_XPATH TEXT NOT NULL,
    TO_REL_XPATH TEXT NOT NULL,
    CONSTRAINT RELATION_PKEY PRIMARY KEY (TO_FRAGMENT_ID, FROM_FRAGMENT_ID, RELATION_TYPE_ID)
```

- Can store any modelled data
- Portable, can easily be adapted to any DB technology

R7 Model Driven CPS PoC

Early Performance Indicators (DB)

(proof point 4)

- Insert
  - 450 records/sec (up to 2M records)
- Queries
  - XPath Query (while loading new data) < 100ms
  - Tree-Walk (75K objects) count with condition < 1,000ms
- Delete
  - Delete Tree of 75K objects incl. constraint checks ~ 6 sec

Demos

1. JSON Data
    1. Upload valid JSON, store in DB
    2. Handle invalid JSON
2. Yang Model File (modules)
    1. Valid model file, store in DB
    2. Handle invalid model
3. Split JSON data for given model
    1. Split into ‘fragments’ ready for generic DB
    2. Handle model change

R7 Model Driven CPS PoC

TOPIC	PoC RESULTS
SEED CODE	Seed Code to serve as a basis for CPS has been developed
PROOF POINTS	1. Create/read CRUD operations using YANG fragments using a simple schema or schema-less repository 2. Deploy & upgrade YANG model fragments at run-time 3. Validate based on YANG Constraints 4. Provide architecture vision and roadmap for a target architecture.
ARCHITECTURE AGREEMENTS	Resolve key architectural Issues necessary for CPS as a stand-alone project
CPS CORE FUNCTIONALITY	Can demonstrate some key CPS operations; basic E2E & integration of Yang parser
PERFORMANCE LIMITS	Early performance indicators tested using generic schema with json-blob storage for attributes

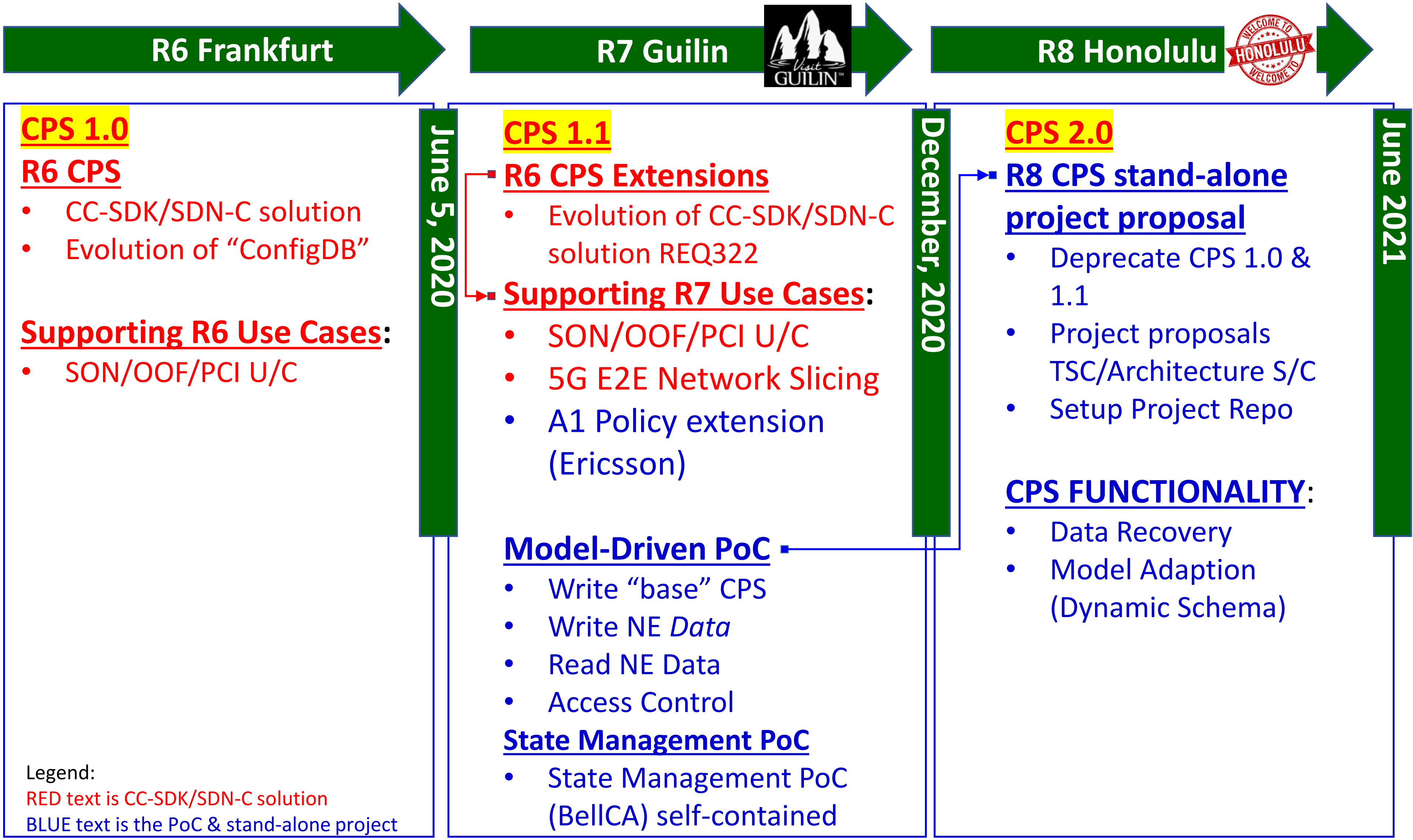
ONAP
OPEN NETWORK AUTOMATION PLATFORM

Roadmap for Configuration Persistence Service (CPS)

Roadmap

CPS Roadmap & R6-R8 Plan

Configuration Persistence Service (CPS) Roadmap –



CPS 1.0

R6 CPS

- CC-SDK/SDN-C solution
- Evolution of "ConfigDB"

Supporting R6 Use Cases:

- SON/OOF/PCI U/C

Legend:

RED text is CC-SDK/SDN-C solution

BLUE text is the PoC & stand-alone project

CPS 1.1

R6 CPS Extensions

- Evolution of CC-SDK/SDN-C solution REQ322

Supporting R7 Use Cases:

- SON/OOF/PCI U/C
- 5G E2E Network Slicing
- A1 Policy extension (Ericsson)

Model-Driven PoC

- Write "base" CPS
- Write NE *Data*
- Read NE Data
- Access Control

State Management PoC

- State Management PoC (BellCA) self-contained

CPS 2.0

R8 CPS stand-alone project proposal

- Deprecate CPS 1.0 & 1.1
- Project proposals TSC/Architecture S/C
- Setup Project Repo

CPS FUNCTIONALITY:

- Data Recovery
- Model Adaption (Dynamic Schema)

CPS Roadmap & R8-R10 Plan

Configuration Persistence Service (CPS) Roadmap –

R8 Honolulu

CPS 2.0

R8 CPS stand-alone project proposal

- Deprecate CPS 1.0 & 1.1
- Project proposals TSC/Architecture S/C
- Setup Project Repo

CPS FUNCTIONALITY:

- Data Recovery
- Model Adaption (Dynamic Schema)

Legend:
RED text is CC-SDK/SDN-C solution
BLUE text is the PoC & stand-alone project

June 2021

R9 Istanbul

Rx (future) development

CPS FUNCTIONALITY:

- Data Auditing Model driven
- Topology Traversal
- Data Syncing

December 2021

R10 Kyoto

Rx (future) development

CPS FUNCTIONALITY:

- Data Auditing Rules Driven
- Data History
- Roll-Back
- Database Backup
- Performance Optimization (Scaling)

June 2021

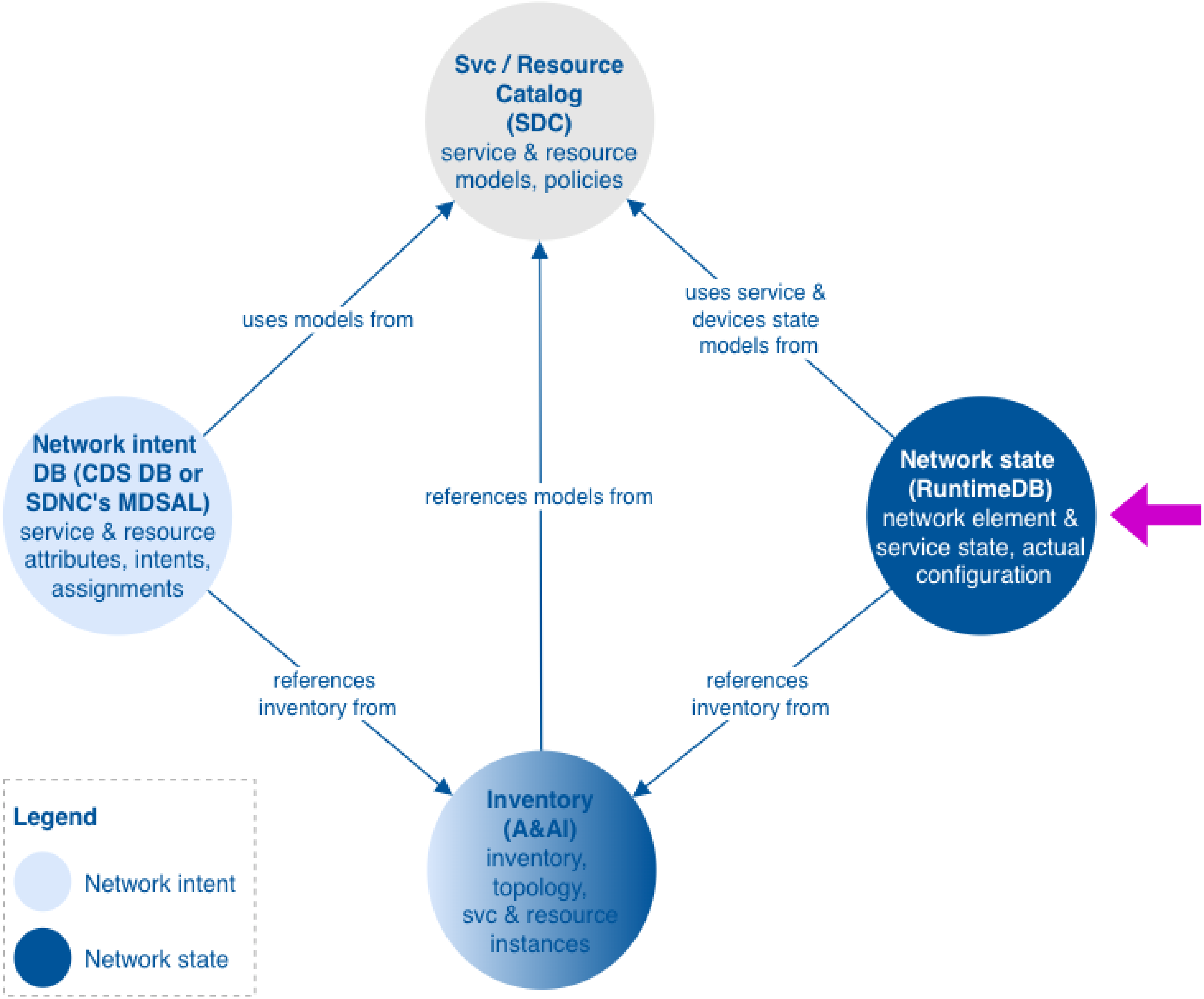
ONAP

OPEN NETWORK AUTOMATION PLATFORM

State Management Use Case for Configuration Persistence Service (CPS) in R8

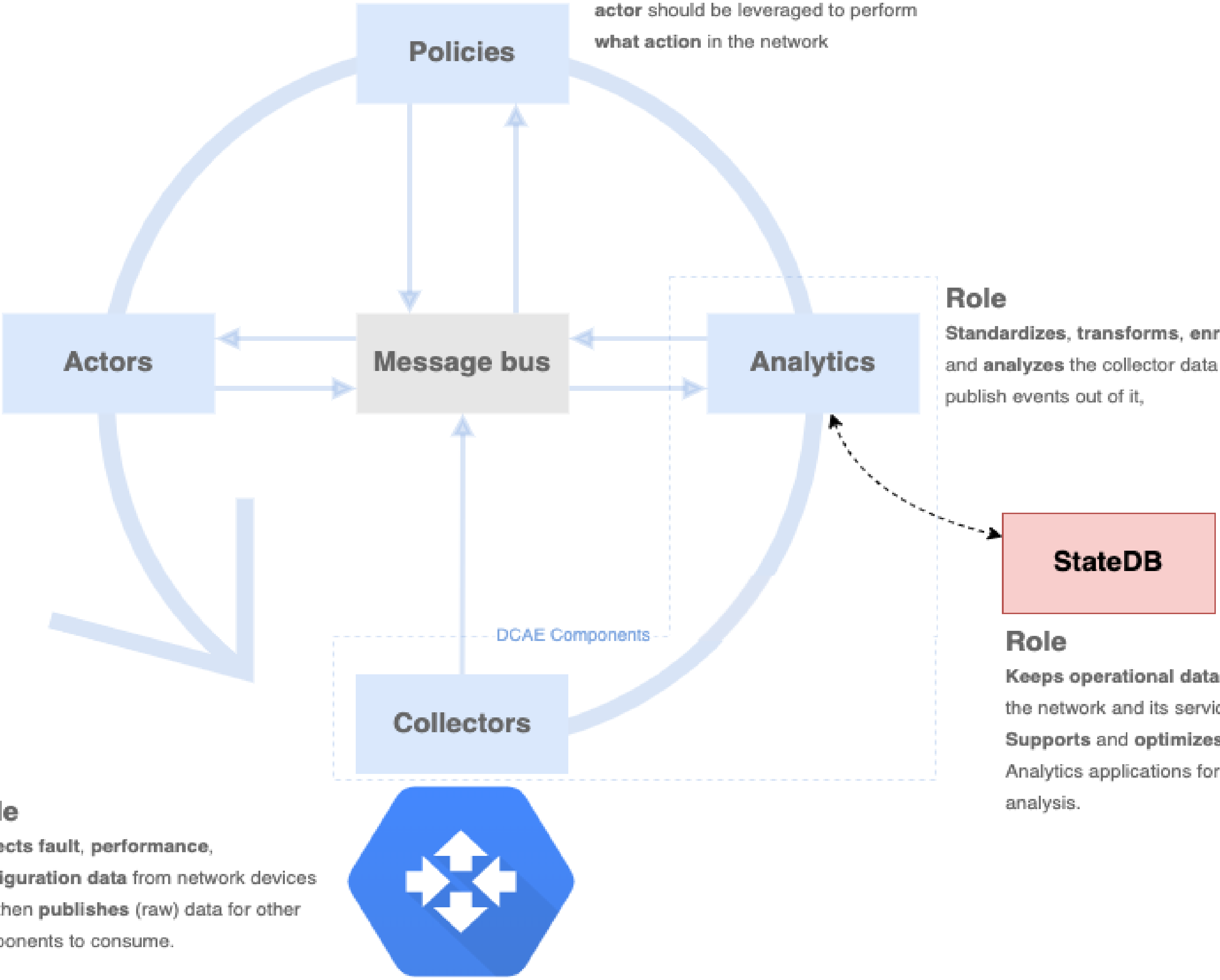
Use Cases

State Management PoC (Bell Canada)



Svc / Resource Catalog (SDC)
service & resource models, policies
uses models from
uses service & devices state models from
references models from
Network intent DB (CDS DB or SDNC's MDSAL)
service & resource attributes, intents, assignments
Network state (RuntimeDB)
network element & service state, actual configuration
references inventory from
references inventory from
Inventory (A&AI)
inventory, topology, svc & resource instances
Legend
Network intent
Network state

State Management PoC (Bell Canada)



Role
From processed events; decides **which actor** should be leveraged to perform **what action** in the network

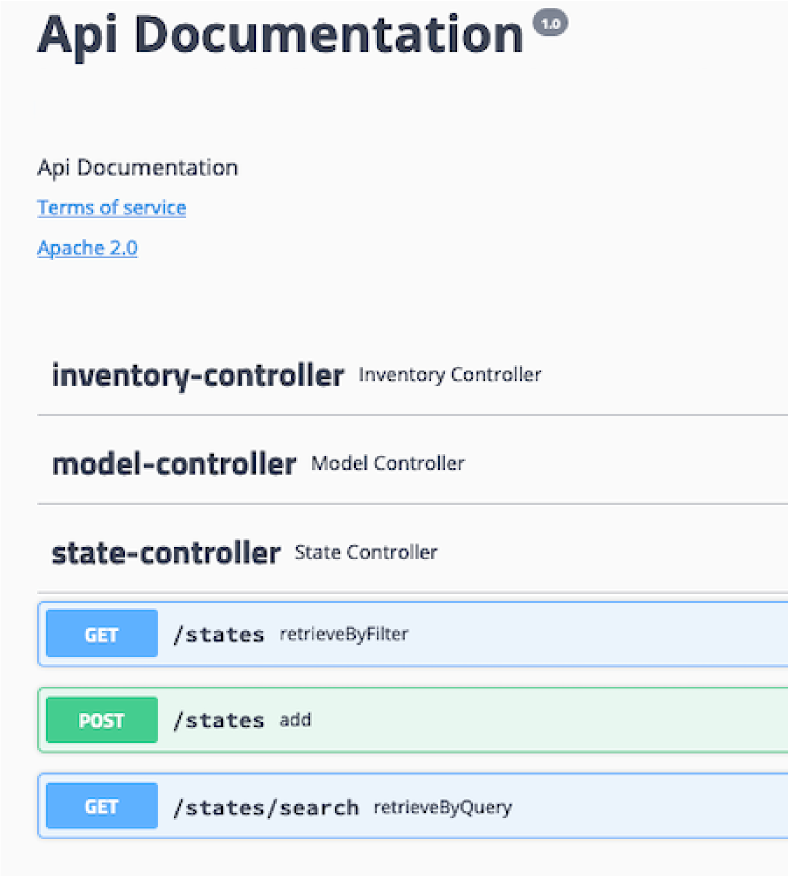
Role
Act upon policy execution and instructions to perform management tasks on the network elements.

Role
Standardizes, **transforms**, **enriches** and **analyzes** the collector data to publish events out of it,

Role
Keeps operational data from the network and its services.
Supports and **optimizes** Analytics applications for theirs analysis.

Role
Collects fault, **performance**, **configuration data** from network devices and then **publishes** (raw) data for other components to consume.

State Management PoC (Bell Canada)



Api Documentation 1.0

Api Documentation

Terms of service

Apache 2.0

inventory-controller Inventory Controller

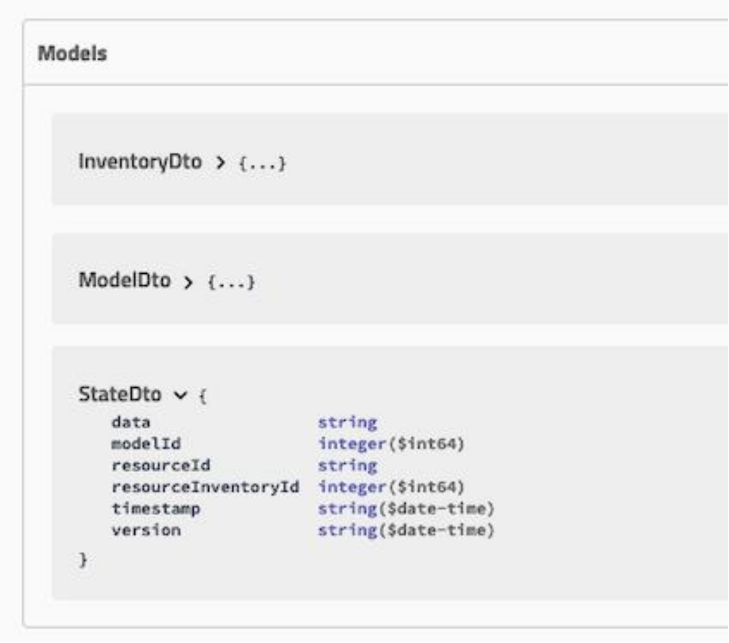
model-controller Model Controller

state-controller State Controller

GET `/states` retrieveByFilter

POST `/states` add

GET `/states/search` retrieveByQuery



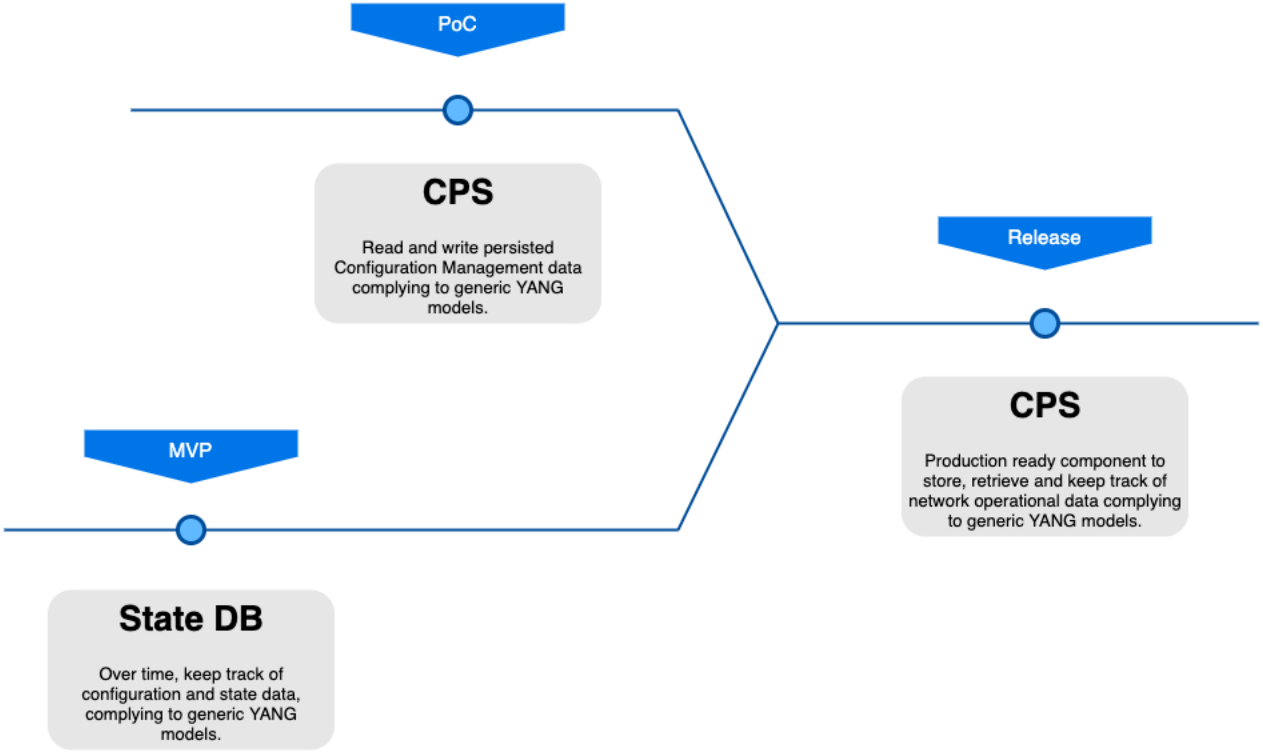
Models

InventoryDto > {...}

ModelDto > {...}

```
StateDto ∨ {
   data                 string
   modelId              integer($int64)
   resourceId           string
   resourceInventoryId  integer($int64)
   timestamp            string($date-time)
   version              string($date-time)
}
```

State Management PoC (Bell Canada)



PoC

CPS

Read and write persisted Configuration Management data complying to generic YANG models.

MVP

State DB

Over time, keep track of configuration and state data, complying to generic YANG models.

Release

CPS

Production ready component to store, retrieve and keep track of network operational data complying to generic YANG models.

State DB and CPS both have the same high level vision:

- To implement the missing run-time operational store to support other ONAP components
- To leverage YANG models to provide a generic structure for data instances

→ Opportunity to converge

ONAP
OPEN NETWORK AUTOMATION PLATFORM

Use Cases for Configuration Persistence Service (CPS)

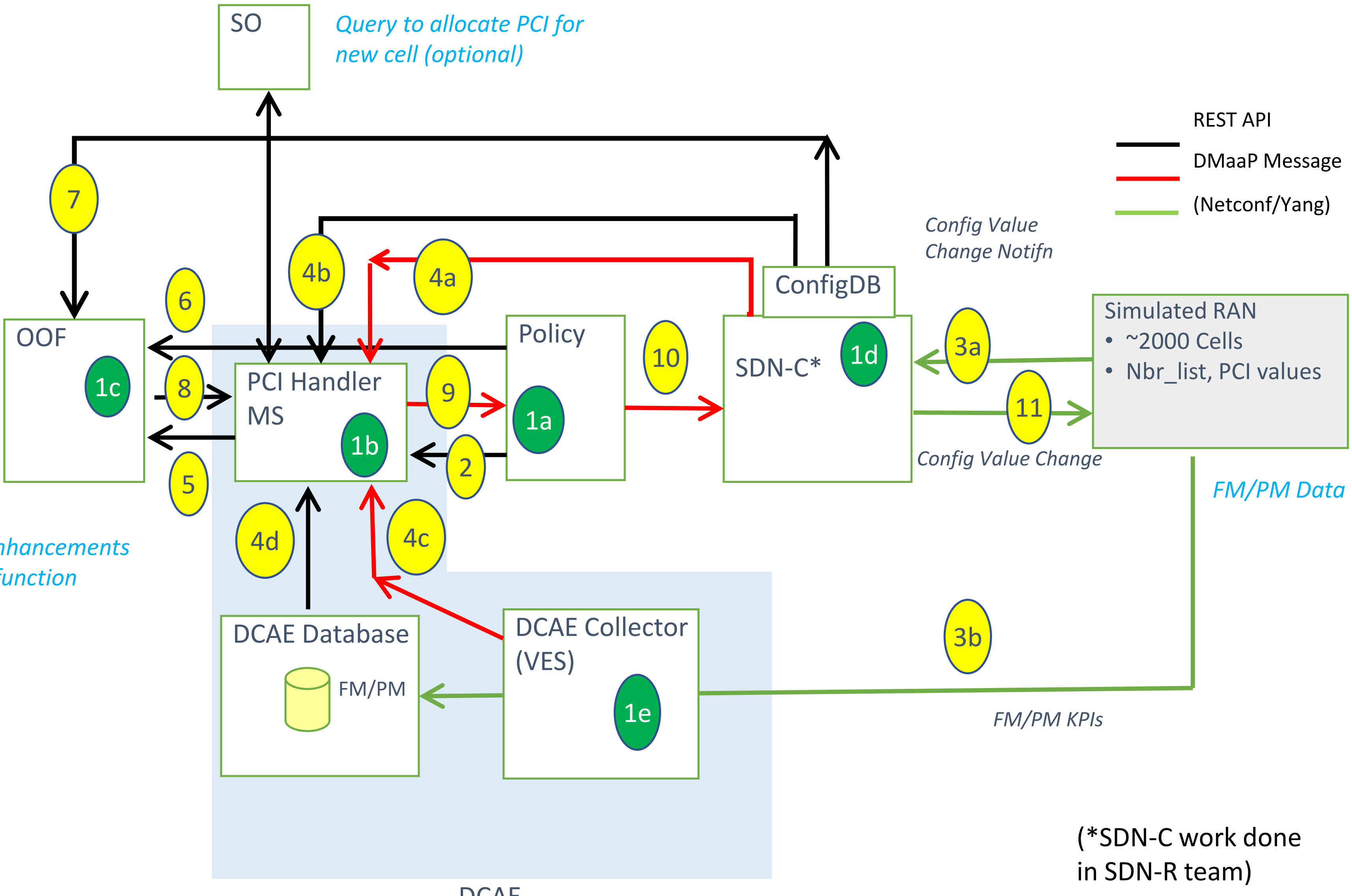
Use Cases

Proof of Concept

CPS Use Cases and Proof of Concepts in R8

5G USE CASE	DESCRIPTION
STATE MANAGEMENT POC	Bell Canada led PoC for State tracking and State management using CPS Integration with CPS (as a platform). Have the State management S/W now work with CPS using available swaggers/APIs
OOF - SON (5G)	Optimization and SON functions for 5G RAN. Self-optimization, Self-Healing, Self-configuration.
NETWORK SLICING (5G Use Case)	Network Slicing defines Slices for 5G RAN systems. Network Slicing is a long-lead (multi-release) development. (will be presented in its own lecture at the Virtual Face to Face)
MOBILITY STANDARDS HARMONIZATION/ A1 adapter	A1 adapter**:** Enhancing the A1 adapter/interface capabilities in ONAP to manage A1 Policies, support multiple A1 targets in the RAN and multi-version A1 interface for different A1 targets, introduce secure TLS communication.

OOF / SON / PCI Use Case

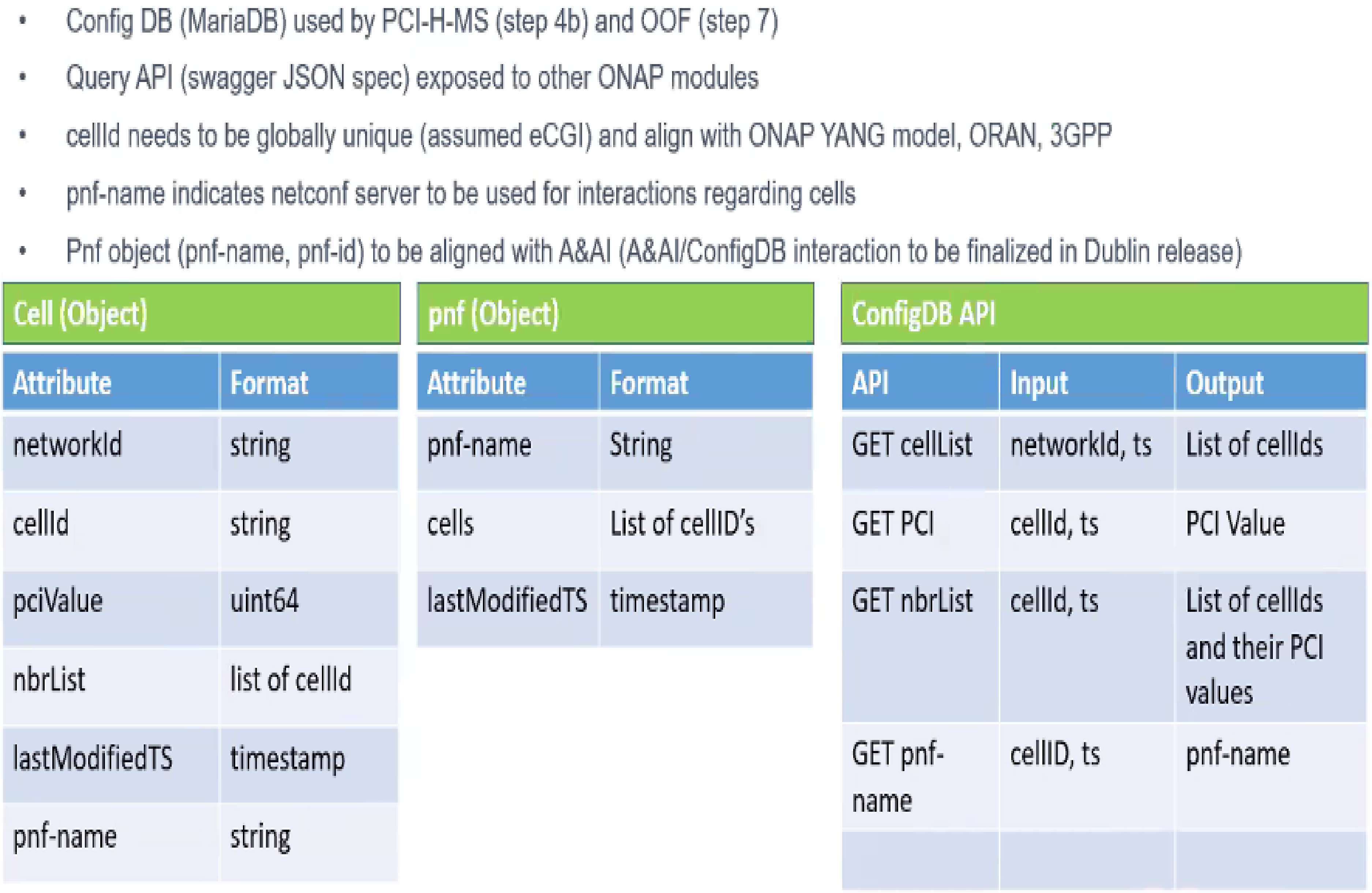


- *PCI enhancements*
- *ANR function*

- *FM/PM Collector and Database*

(*SDN-C work done in SDN-R team)

OOF / SON / PCI Use Case



- Config DB (MariaDB) used by PCI-H-MS (step 4b) and OOF (step 7)
- Query API (swagger JSON spec) exposed to other ONAP modules
- cellId needs to be globally unique (assumed eCGI) and align with ONAP YANG model, ORAN, 3GPP
- pnf-name indicates netconf server to be used for interactions regarding cells
- Pnf object (pnf-name, pnf-id) to be aligned with A&AI (A&AI/ConfigDB interaction to be finalized in Dublin release)

Cell (Object)

Attribute	Format
networkId	string
cellId	string
pciValue	uint64
nbrList	list of cellId
lastModifiedTS	timestamp
pnf-name	string

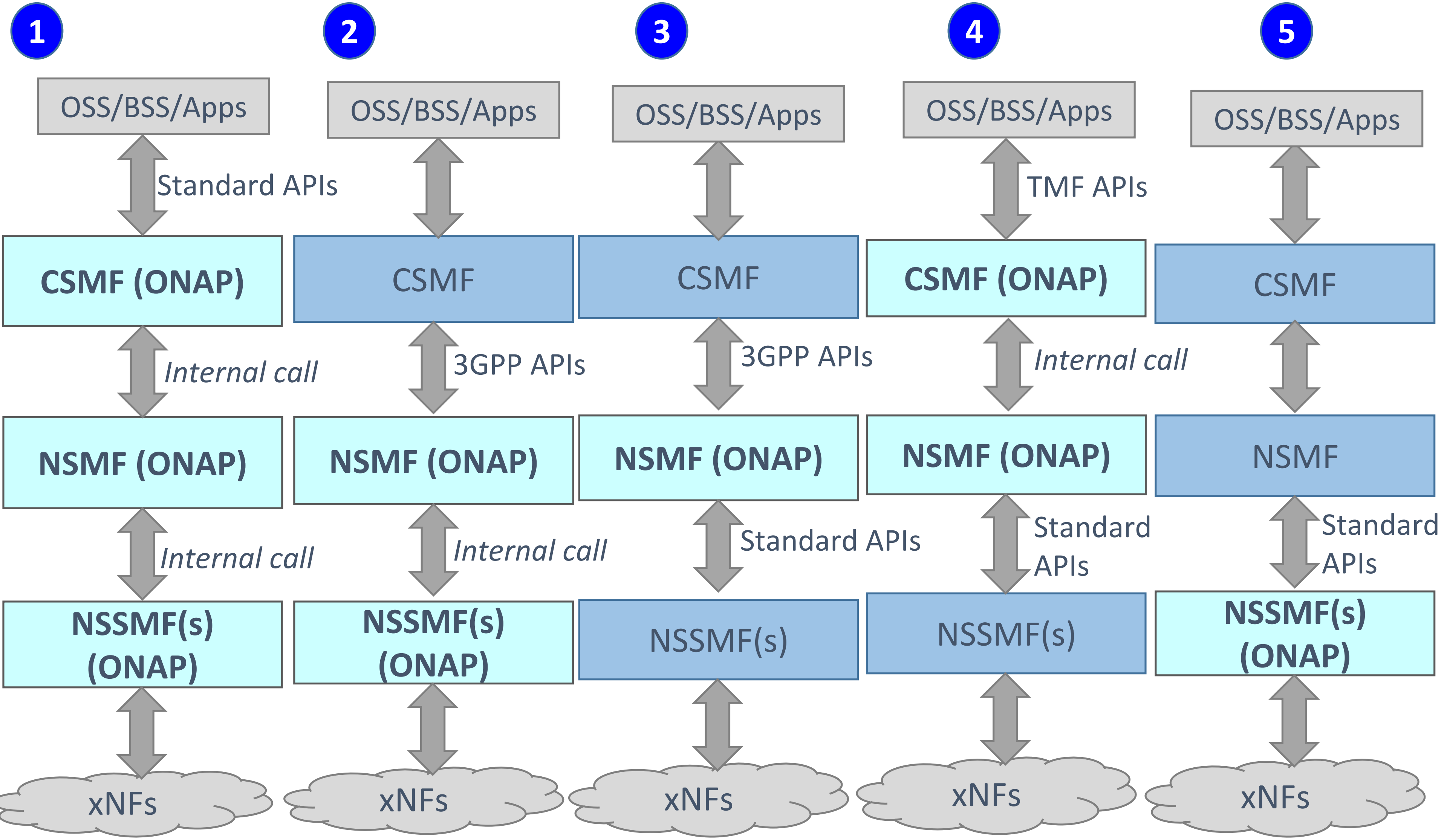
pnf (Object)

Attribute	Format
pnf-name	String
cells	List of cellID's
lastModifiedTS	timestamp

ConfigDB API

API	Input	Output
GET cellList	networkId, ts	List of cellIds
GET PCI	cellId, ts	PCI Value
GET nbrList	cellId, ts	List of cellIds and their PCI values
GET pnf-name	cellID, ts	pnf-name

End to End Network Slicing Use Case



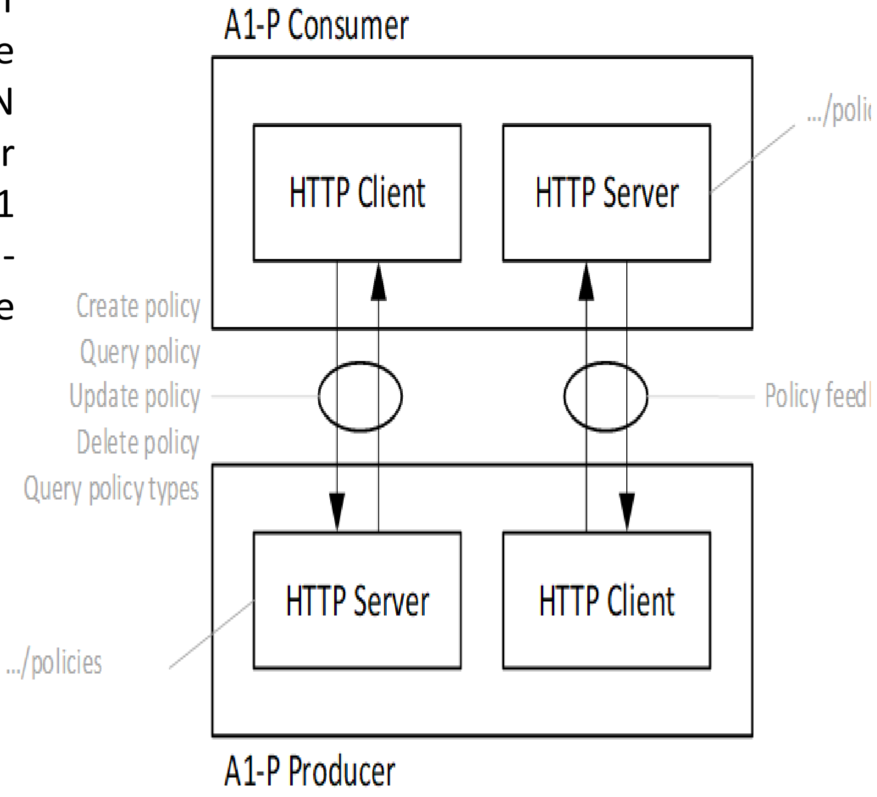
3rd party component

End to End Network Slicing Use Case

NetworkSlice	Network Slice NRM	operationalState
NetworkSlice	Network Slice NRM	administrativeState
NetworkSlice	Network Slice NRM	serviceProfileList
NetworkSlice	Network Slice NRM	networkSliceSubnetRef
NetworkSliceSubnet	Network Slice NRM	operationalState
NetworkSliceSubnet	Network Slice NRM	administrativeState
NetworkSliceSubnet	Network Slice NRM	nsInfo
NetworkSliceSubnet	Network Slice NRM	sliceProfileList
NetworkSliceSubnet	Network Slice NRM	managedFunctionRef
NetworkSliceSubnet	Network Slice NRM	networkSliceSubnetRef
ServiceProfile	Network Slice NRM	serviceProfileId
ServiceProfile	Network Slice NRM	sNSSAIList
ServiceProfile	Network Slice NRM	pLMNIdList
ServiceProfile	Network Slice NRM	perfReq
ServiceProfile	Network Slice NRM	maxNumberofUEs
ServiceProfile	Network Slice NRM	coverageAreaTAList
ServiceProfile	Network Slice NRM	latency
ServiceProfile	Network Slice NRM	uEMobilityLevel
ServiceProfile	Network Slice NRM	resourceSharingLevel
ServiceProfile	Network Slice NRM	sST
ServiceProfile	Network Slice NRM	availability
SliceProfile	Network Slice NRM	sliceProfileId
SliceProfile	Network Slice NRM	sNSSAIList
SliceProfile	Network Slice NRM	pLMNIdList
SliceProfile	Network Slice NRM	perfReq
SliceProfile	Network Slice NRM	maxNumberofUEs
SliceProfile	Network Slice NRM	coverageAreaTAList
SliceProfile	Network Slice NRM	latency
SliceProfile	Network Slice NRM	uEMobilityLevel
SliceProfile	Network Slice NRM	resourceSharingLevel

A1 Policy Extension ORAN-ONAP Harmonize

Executive Summary - This requirement enhances the A1 adapter/interface capabilities provided in Rel 6 as part of 5G/ORAN & 3GPP Standards Harmonization requirement (REQ-38). O-RAN has defined A1 interface specification in the context of the management of 5G RAN elements to provide intent based policies for optimization of the RAN network performance. Planned enhancements for Rel 7 include additional support for managing A1 Policies, multiple A1 targets in the RAN, multi-version support for different A1 targets, and secure TLS communication.



OPEN NETWORK AUTOMATION PLATFORM

Appendix

Benjamin Cheung

Access, Syncing, Indexing Runtime Config DB

ACCESS TO CPS Database (READ/WRITE):

READ ONLY - Run-Time parameters can be READ by any ONAP platform component and any ONAP plug-in. Examples of ONAP platform components are A&AI, SDC, SDNC etc.

READ/WRITE - Parameters can be READ/WRITE from Controllers, DCAE (future), VES Collector/DMaaP, A&AI, Policy/CLAMP (future) and other components with permission settings.

DEFAULT - SO (future), DCAE, A&AI (indirectly), Controllers (CDS, APPC, SDNC) will have default read/write access to CPS Database

DEFINABLE - Other components will have default read-only access to Configuration Persistence Service but can be given Read/Write access on a per record basis.

SYNCING NEW xNF ADDED or DELETED (A&AI):

ELEMENT SYNC - Software keeps the A&AI elements with the elements in the RunTime Config DB in Sync. When the network first being established, a *GetAllPNFs* function from A&AI can be used on startup.

A&AI - A&AI is still the master of valid entities in the network and provides a dynamic view of the assets (xNFs) available to ONAP

CPS Database - The CPS Database is a master of the associate (exo-inventory) data associated with the entities.

DYNAMIC VIEW - When a xNF appears or is removed from the system, CPS Database records will be added/removed based on A&AI entries.

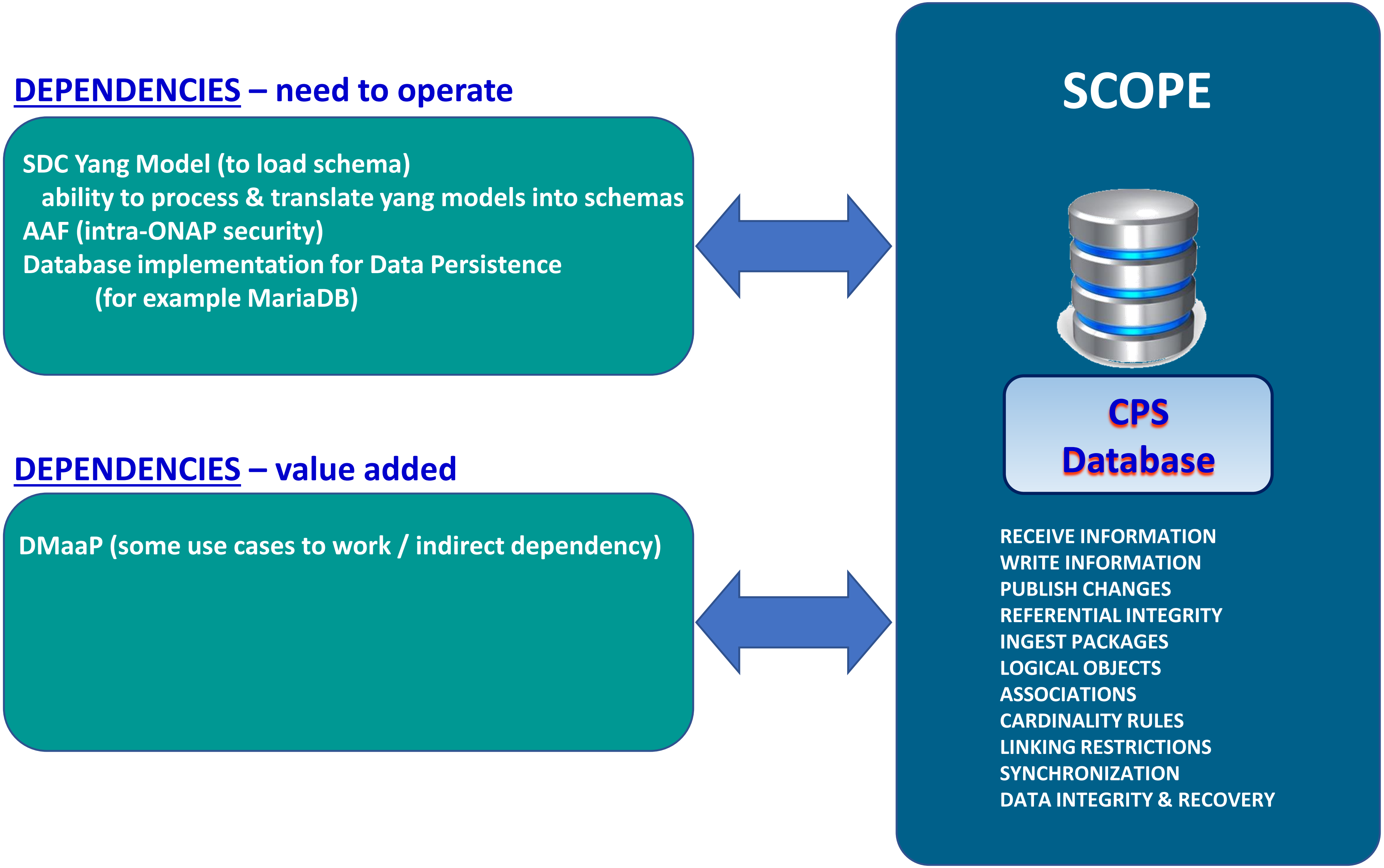
LOGIC - When a xNF appears is removed there is logic to determine how and when something is to be updated. There is some intelligence to know what elements of update.

INDEXING:

INDEXING - Data Records will be indexed by xNF (VNF, PNF, ANF). It would be an objective to have a similar indexing mechanism as A&AI. May also need an index to be a logical object ID.

RETRIEVAL - How are data records retrieved efficiently. This relates how the records are indexed.

Dependencies vs Scope



DEPENDENCIES – need to operate

SDC Yang Model (to load schema)
ability to process & translate yang models into schemas
AAF (intra-ONAP security)
Database implementation for Data Persistence
(for example MariaDB)

DEPENDENCIES – value added

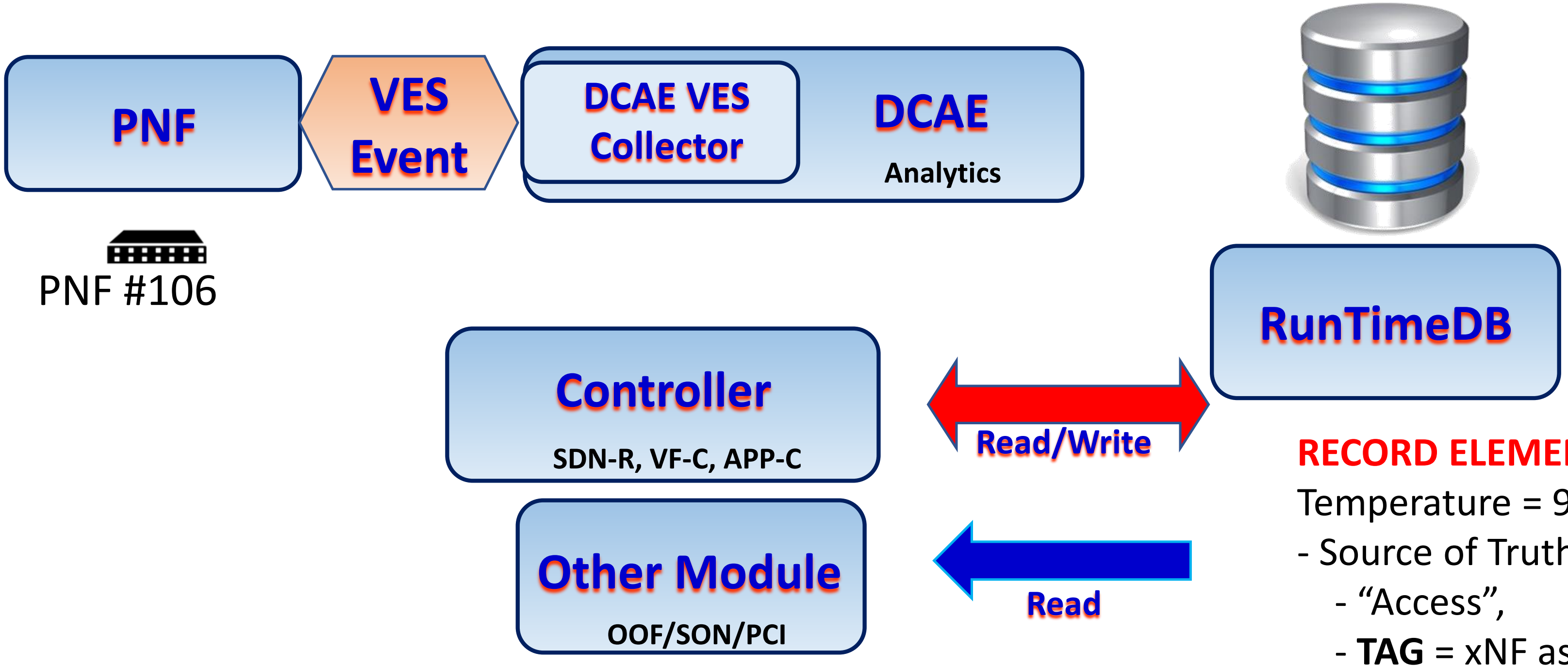
DMaaP (some use cases to work / indirect dependency)

SCOPE

CPS Database

RECEIVE INFORMATION
WRITE INFORMATION
PUBLISH CHANGES
REFERENTIAL INTEGRITY
INGEST PACKAGES
LOGICAL OBJECTS
ASSOCIATIONS
CARDINALITY RULES
LINKING RESTRICTIONS
SYNCHRONIZATION
DATA INTEGRITY & RECOVERY

Configuration Persistence Service (Run-Time)



RECORD ELEMENT

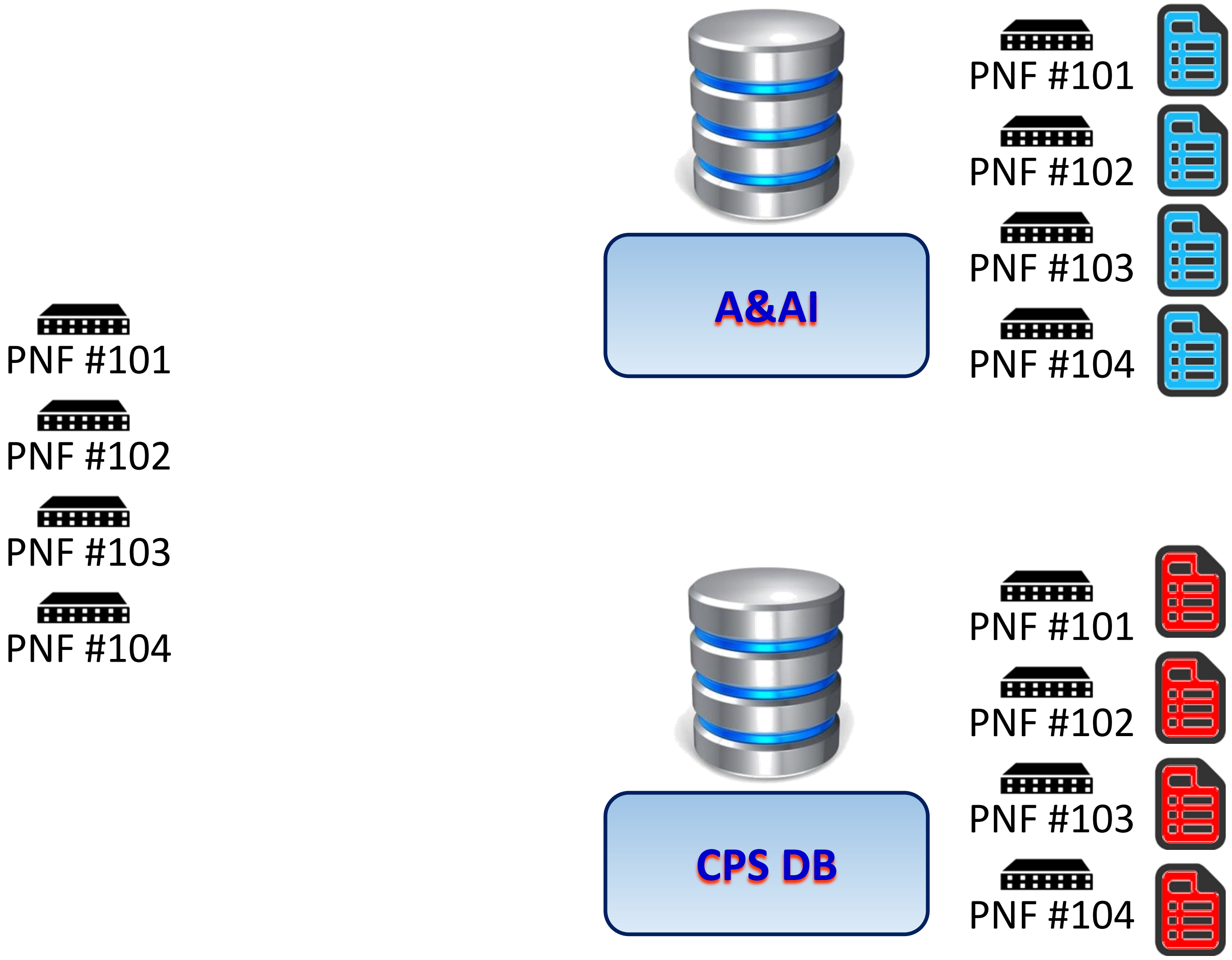
Temperature = 98

- Source of Truth,

  - “Access”,

  - **TAG** = xNF association, Platform ONAP (App-C, ms#10105) (for PNF #106) or Flex-Index

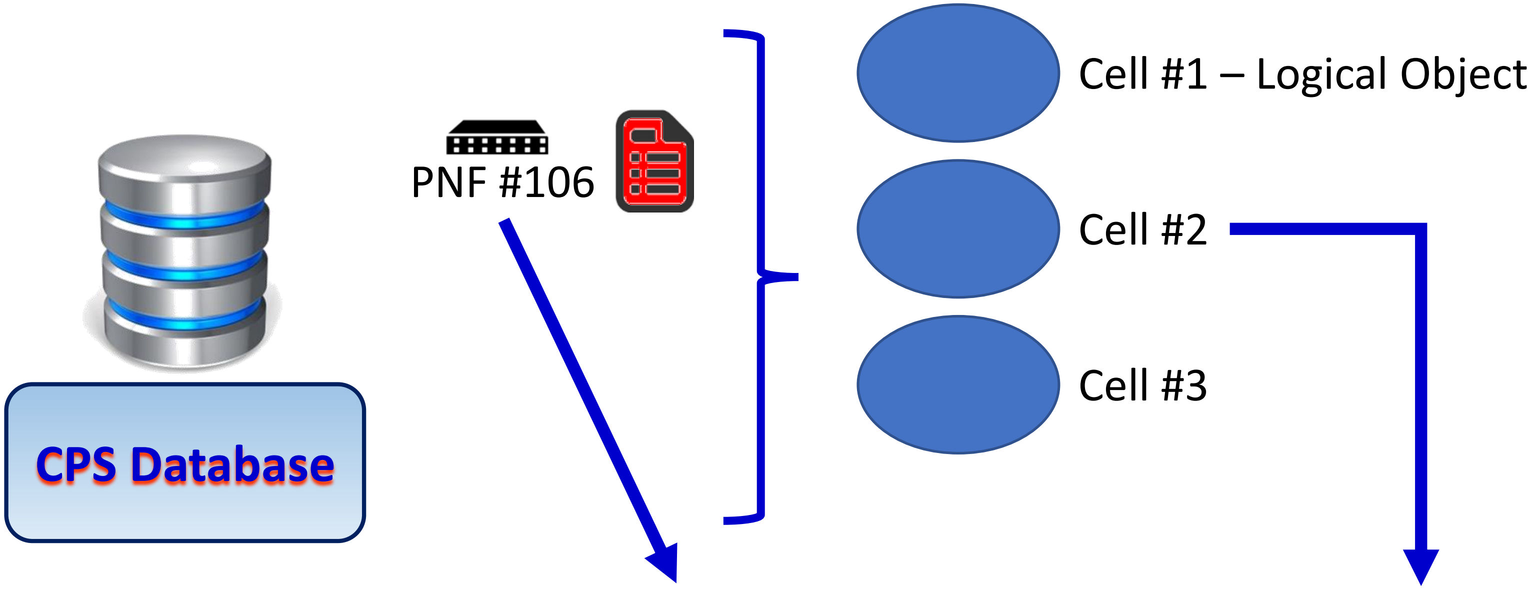
PNF #106



A&AI correlated/Index to RunTimeDB
Publish changes in A&AI, notification on DMaaP

Indices into Configuration Persistence Service may also use Flex-Index (such as CellID)

CPS Database (Run-Time View)

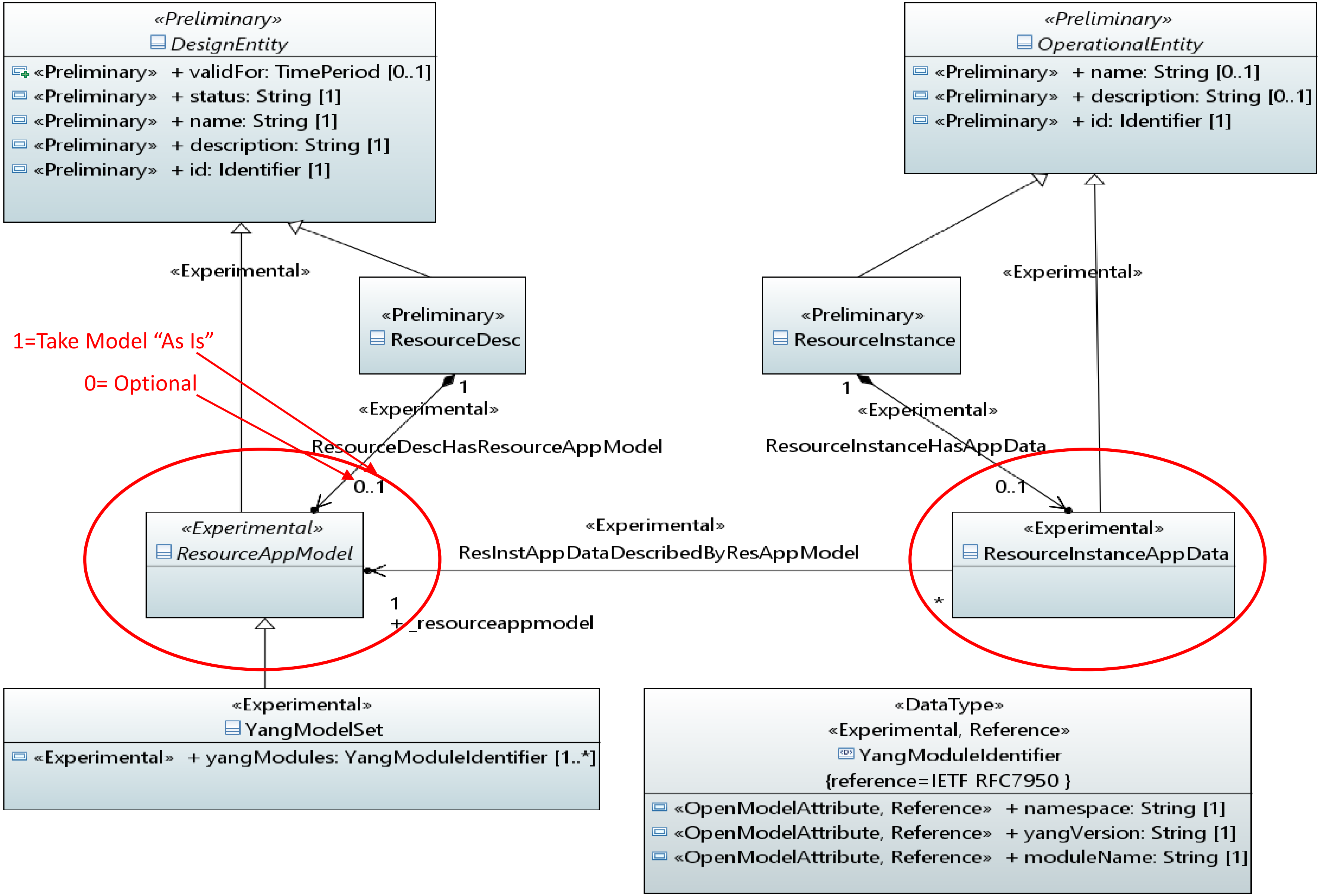


RECORD ELEMENT
INDEX = PNF #106
Parameter #1
Parameter #2
Parameter #3
Logical object, Cell #1
    Cell Parameter #1
    Cell Parameter #2
    Cell Parameter #3

RECORD ELEMENT
INDEX = PNF #106
Parameter #1
Parameter #2
Parameter #3
State Info X.733
Associations
    { Logical Object #111 Cell #2 }
Cardinality Rules
Linking Restrictions

RECORD ELEMENT
INDEX = Logical Object #111
Parameter #1
Parameter #2
Parameter #3
State Info
Associations
    { PNF #106 }
Cardinality Rules
Linking Restrictions

CPS Information Model Design R7



«Preliminary»
DesignEntity
- «Preliminary» + validFor: TimePeriod [0..1]
- «Preliminary» + status: String [1]
- «Preliminary» + name: String [1]
- «Preliminary» + description: String [1]
- «Preliminary» + id: Identifier [1]

«Preliminary»
OperationalEntity
- «Preliminary» + name: String [0..1]
- «Preliminary» + description: String [0..1]
- «Preliminary» + id: Identifier [1]

«Experimental»

«Preliminary»
ResourceDesc

«Preliminary»
ResourceInstance

«Experimental»

1=Take Model "As Is"

0= Optional

1
«Experimental»
ResourceDescHasResourceAppModel
0..1

1
«Experimental»
ResourceInstanceHasAppData
0..1

«Experimental»
ResourceAppModel

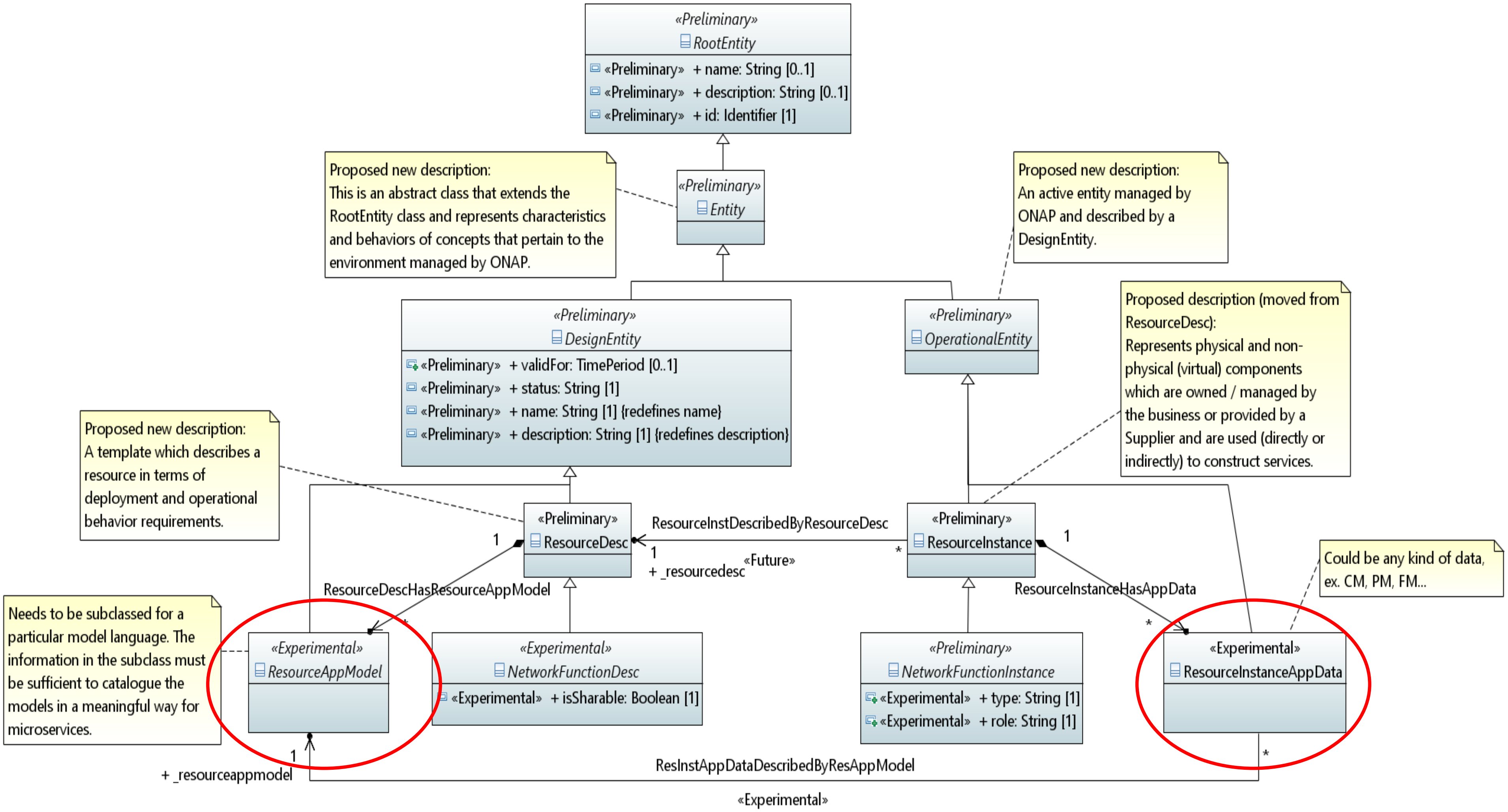
«Experimental»
ResInstAppDataDescribedByResAppModel
1
+_resourceappmodel
*

«Experimental»
ResourceInstanceAppData

«Experimental»
YangModelSet
- «Experimental» + yangModules: YangModuleIdentifier [1..*]

«DataType»
«Experimental, Reference»
YangModuleIdentifier
{reference=IETF RFC7950 }
- «OpenModelAttribute, Reference» + namespace: String [1]
- «OpenModelAttribute, Reference» + yangVersion: String [1]
- «OpenModelAttribute, Reference» + moduleName: String [1]

CPS Information Model Design R8



«Preliminary»
RootEntity
«Preliminary» + name: String [0..1]
«Preliminary» + description: String [0..1]
«Preliminary» + id: Identifier [1]

«Preliminary»
Entity

Proposed new description:
This is an abstract class that extends the RootEntity class and represents characteristics and behaviors of concepts that pertain to the environment managed by ONAP.

Proposed new description:
An active entity managed by ONAP and described by a DesignEntity.

«Preliminary»
DesignEntity
«Preliminary» + validFor: TimePeriod [0..1]
«Preliminary» + status: String [1]
«Preliminary» + name: String [1] {redefines name}
«Preliminary» + description: String [1] {redefines description}

«Preliminary»
OperationalEntity

Proposed description (moved from ResourceDesc):
Represents physical and non-physical (virtual) components which are owned / managed by the business or provided by a Supplier and are used (directly or indirectly) to construct services.

Proposed new description:
A template which describes a resource in terms of deployment and operational behavior requirements.

«Preliminary»
ResourceDesc

ResourceInstDescribedByResourceDesc
1
+ _resourcedesc
«Future»
*

«Preliminary»
ResourceInstance

1
ResourceDescHasResourceAppModel

1
ResourceInstanceHasAppData
*

Could be any kind of data, ex. CM, PM, FM...

Needs to be subclassed for a particular model language. The information in the subclass must be sufficient to catalogue the models in a meaningful way for microservices.

«Experimental»
ResourceAppModel

«Experimental»
NetworkFunctionDesc
«Experimental» + isSharable: Boolean [1]

«Preliminary»
NetworkFunctionInstance
«Experimental» + type: String [1]
«Experimental» + role: String [1]

«Experimental»
ResourceInstanceAppData

1
+ _resourceappmodel
ResInstAppDataDescribedByResAppModel
*
«Experimental»