# Proposal for ONAP to be able to support Service Testing Open Test Framework (OTF)

#### What is OTF?

- A framework to support patterns for automated E2E Service testing and fault isolation support.
- Comprised of existing discrete ONAP functions and new discrete functions/code that support OTF Patterns.

#### Questions to answer:

- Why is this project important to the ONAP Community?
- What is new to ONAP?
- How does this project fit in the ONAP Architecture?
- What platform development is needed for this project?
- What finished goods and/or improved operations can be developed as a result of this project?
- Who is interested in developing this project?
- What is the projected ongoing cost to develop and consume this project from ONAP?
- Why is this an important problem to solve?
- Project Use Cases?



## OTF Role in Test Design Patterns

#### **Certification Test:**

- Done in Lab or Sandbox environment
- Focus on verifying the functionality within the service design process
- May use tools not available in production environment: simulators, emulators, fault injectors
- ❖ OTF provides the simulators, emulators, fault injectors and data to configure and execute each test in a suite of tests

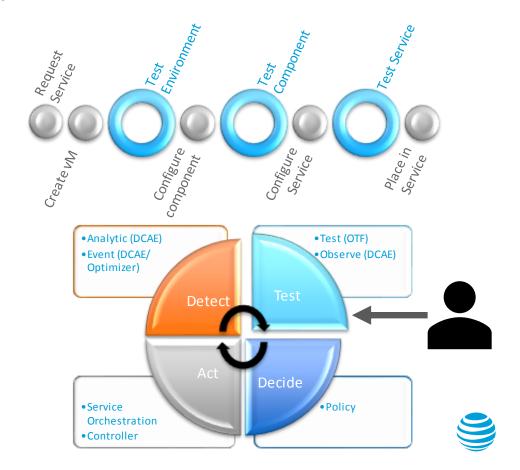
#### Lifecycle Test:

- Done in Lab, Sandbox and Production environments
- Verifies progress at major orchestration milestones
- Enables incremental testing to validate intermediate work steps (Turnup Testing)
- OTF separates test details from lifecycle steps

#### Monitoring & Troubleshooting:

- Done in Lab, Sandbox and Production.
- Focus on testing used to isolate and determine root cause of detected condition.
- Participates as part of an active closed loop or to validate functionality of a closed loop
- Manual Initiation by Operations personnel
- OTF provides tool set for closed loop test activities

Regression
Old Feature Test 1
Old Feature Test ...
Old Feature Test N
New Feature Test 1
New Feature Test ...
New Feature Test N



## Why is OTF Important to the ONAP Community?

#### Service Providers

- Consistent Framework in the Management Platform (ONAP) to support E2E Service testing and fault isolation.
- Creates a **common** environment to access the test implementations and tools to support a service of diverse "elements".
- Provider Collaborative contributions will provide a more robust and accepted approach to standardize testing.
  - Lower costs of adoption to emergent script engines in the industry
  - ✓ Lower finished goods costs due to standardization
  - ✓ Enables Marketplace for standardized tests and test tools

#### VNF Suppliers

- Automates the setup and "connect the dots" that needs to occur to enable a certification test to be executed.
- Enables **streamlined** process for evolution of proprietary (secret sauce) test tools (simulators/emulators).
- Provide the ability to isolate faults to/from vendor products.
  - ✓ Decrease Vendor time to market by simplifying VNF design pattern
  - ✓ Decrease VNF costs by simplifying test development (certification, operations)

#### Test Scripting/Software and Test Head Suppliers

- Enable a marketplace for "tests" as finished goods that may be service specific but not VNF specific.
- Provides common mechanism for ONAP Integration and promotes test interface standards.
  - ✓ Provide market competitive tests and test tools that are ONAP Finished Goods that can be bought by VNF Vendors or Providers to provide them a competitive.



## What exactly is the Open Test Framework

Platform

**Provides** 

- Standardized Test Control Engine
- Standardized invocation: REST, DMaaP,
- Standardized Policy Enforcement
- Standardized Result Reporting
- Standardized Script Engine
- •SDK for Test Developers
- Standardized GUI for:
  - Test Invocation
  - Test History View
  - Detailed Results View / Progress Monitoring

#### Capabilities

- **Ecosystem to support**
- manual and automated
- service/network testing
- Ability to onboard Test Head Microservices via SDC
- Ability to add additional script engines
- Ability to define a test (name, scripts, test heads, test strategy, etc.)
- Ability to onboard test metadata (scripts, strategy, policies)

**Enables Users to Develop** 

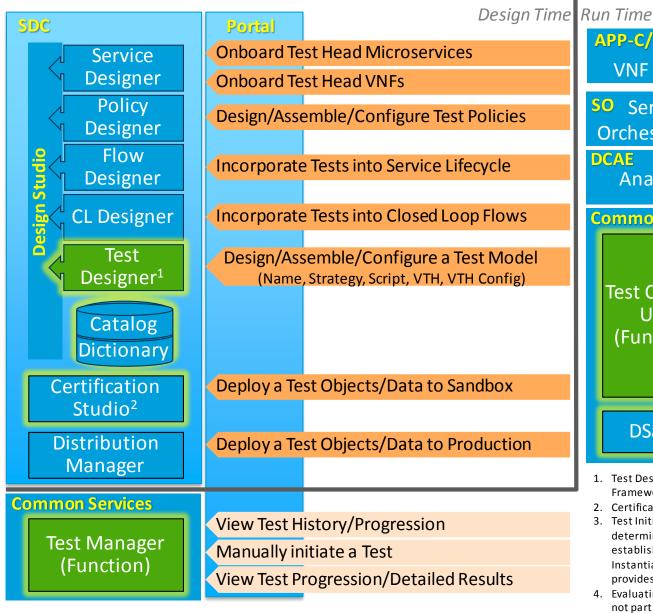
Service
Plugins and
Applications

- Test Heads
- Test Scripts
- Test Strategy
- Policies
- Analytics

Finished Goods



#### What is new to ONAP?



APP-C/NC/VNF-C VNF Command Invocation

**VNF Health Check VNF Mgmt** 

Orchestrate a Test Strategy

DCAE Analytics

Orchestration

Service

Analyze Test Telemetry to form Heuristic<sup>4</sup>

**Common Services** 

Test Control Unit (Function)

Associate Test Definition with Runtime Objects

Configure a Test Head Microservice

Execute a Test Script

**Establish Needed Test Configuration** 

Initiate a Test<sup>3</sup>

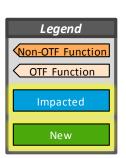
**Enforce Test Policies** 

DSaaS<sup>5</sup>

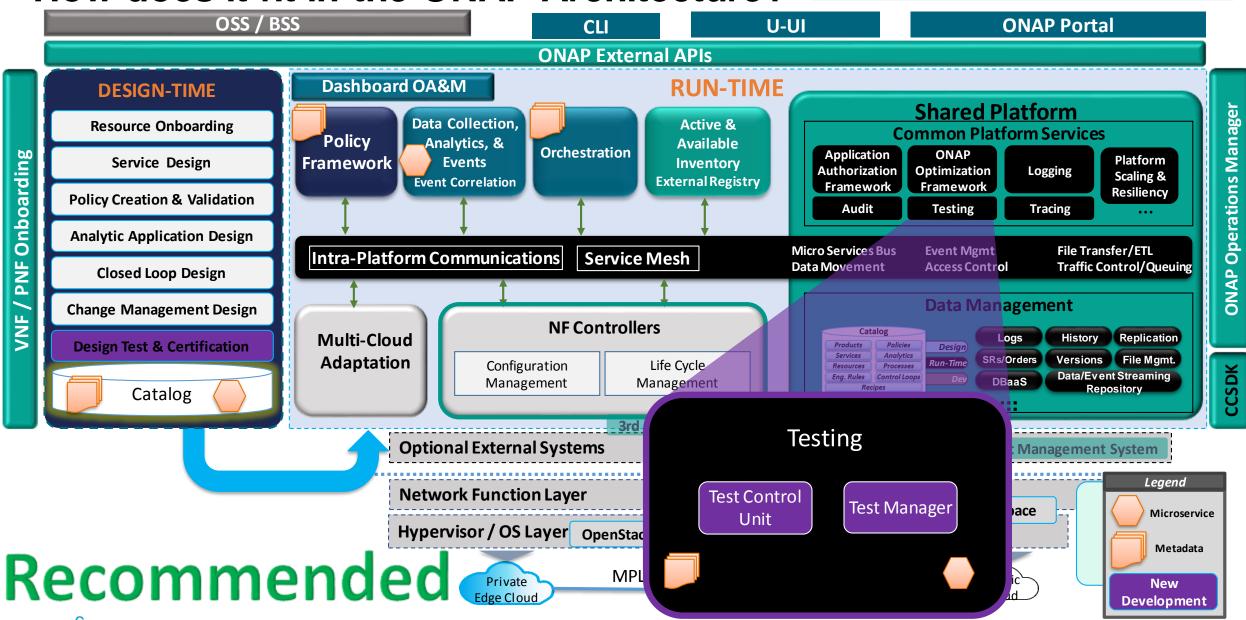
Store Test Metadata

Store Test Results/History

- 1. Test Designer does not exist and may be implemented in an existing designer but is not part of the Framework
- 2. Certification Studio is identified in the architecture but not yet existing.
- 3. Test Initiation can come from Manual, Orchestration, or Closed Loop. All come into OTF, OTF will determine the Strategy and Request Orchestration of the Strategy with the instance data to establish the test context. Even in Orchestrated flows such as Change Management and Instantiation the Test may request an orchestrated sub-flow (test strategy) to be spawned. This provides a separation of concern between the lifecycle flow and the test implementation flows.
- 4. Evaluating Heuristics for Pass/Fail conclusion is part of a Script, Strategy, or VTH all of which are not part of OTF or any other ECOMP Component
- 5. Data Storage as a Service is proposed Casablanca project. OTF intends to use such a platform abstraction for its Data Centric approach to Test Data







#### What platform development is needed for OTF?

## Legend In Project Scope Not in Project Scope

#### SDC

Test Designer

Provides a way for test to be integrated with service/resource and policy design

Extensible Catalog

• Catalog needs to be enhanced to be extensible for Designer Plugins

**Extensible Data Dictionary** 

• Data Dictionary needs to be enhanced to be extensible for Designer Plugins

**Certification Studio** 

• Begin planning the emergence of the Certification Studio to enable automated certification/re-certification

#### **OTF**

**Test Manager** 

• Provides dashboard of executing tests and ability to initiate or schedule tests

Initial Script Engine

• Although an extension of the platform at least one must be used.

OTF Shared Library & SDK

• Common utilities, functions are packaged, shared and made available to other OTF components and microservices

**OTF Service DB** 

- The datastore for OTF. It's to store temporary and permanent data that is used to configure, control test service.
- Test results and history data is also stored here

Test policy enforcement (allowed execution times, authorization controls, etc.)

- fixed logic pattern to organize different sets of metadata to affect the execution of different tests
- Service API: The generic RESTful API that is exposed to client. The Service API supports both synchronous and asynchronous mode.
- Message Listener: The Message Listener listens to client's test request from the Message Bus (DMaaP), which
  is asynchronous by nature. The test result/response is posted to the DMaaP message topic specified by client

**Test Control Unit** 

## What finished goods and/or improved operations can be developed as a result of

Legend
In Project Scope
Not in Project Scope

## this project? Finished Goods

Test Strategies	<ul> <li>Visualized business logic and flow of the test. Camunda BPM (but not constrained to) will be used to model the test strategy.</li> <li>This may be composed of both Service Orchestrated Activities and Cloud Orchestrated Activities.</li> </ul>
Test Scripts	Test Script is self-contained and performs dedicated test functions written in script language, , i.e. Groovy, JavaScript, JRuby and Jython/Python, etc.
vTH Configurations	<ul> <li>Configuration settings to establish a test head with a network identity and role to provide. This may also include reply/response scripts to control the test head during the test execution.</li> </ul>
Virtual Test Head (vTH )	<ul> <li>A Virtual Test Head is a software component (it can be a mS, scripts, etc.) that hooks up the targeted device/VNF required by the Test Strategy. It has the knowledge (test capabilities, interface, protocols, parameters etc.) on the specific type of device/VNF and only serves this kind.</li> </ul>
Adapter vTH	<ul> <li>This type of test head serves as an adapter to execute test through an external system that already implements the test. This prevents duplicating and reinventing the wheel if the test capability is already available externally and can be used directly through API calls.</li> <li>Another variation of this test head allows VNFs to be deployed that act as a Test Head and can be configured by the adapter, post instantiation, for a specific test case.</li> </ul>
ProxyvTH	<ul> <li>This type of test head serves as an adapter to execute test through an external system that already implements the test.         This prevents duplicating and reinventing the wheel if the test capability is already available externally and can be used directly through API calls.     </li> </ul>
Robot vTH	Many components of ONAP and VNFs using ONAP are demonstrated by ROBOT tests. The initial OTF capabilities for ONAP should support these tests as test objects listed in the SDC Catalog.

#### Operations Improvements

**Automated Certification Test** 

• Enables to emergence of the Certification Studio with onboarded certification tests developed by resource vendors.

Automated Regression Test

Enables defined services to be regressed testes against an upgraded ONAP instance to ensure it is still operable in the environment.

## Who is interested in developing OTF?

#### Providers:

AT&T

Reliance Jio

#### Vendors:

Spirent

Need to reach out to broader ONAP community for additional support



## Why is this an important problem to solve?

By creating a common test framework that can be used by both AT&T internal developers and external vendor partners will increase consistency and reuse across AT&T. This can result in major cost savings in both operations and development costs for AT&T.

Driving this capability as a part of ONAP will allow end-to-end service testing to be developed as finished goods which can either be developed in OpenSource reducing development cost for AT&T by "crowd sourcing" of the finished goods. Or by suppliers wanting testing as part of their onboarded package. Additionally this will allow all ONAP Suppliers and ONAP Providers a common mechanism to onboard VNFs "certified" in a Vendor ONAP Lab, to be re-certified in the Provider ONAP environment. Shortening the time between onboarding and deployment of the VNF.



#### **OTF Initial Use Cases**

#### Design

- Add Test Designer Plugin to Design Studio
  - Add new Design Studio Applications (Plugins)
  - Add new Model Types to Design Catalog
  - Extend existing Model Types in Design Catalog
  - Create new Dictionary Terms
  - Extend existing Dictionary Terms
- Create Test Objects
  - Create vFW Demo Test Object (Creates vFW VNF)
  - Onboard vFW Healthcheck Test Object
- Distribute Test Objects
  - This would follow standard SDC Distribution practices

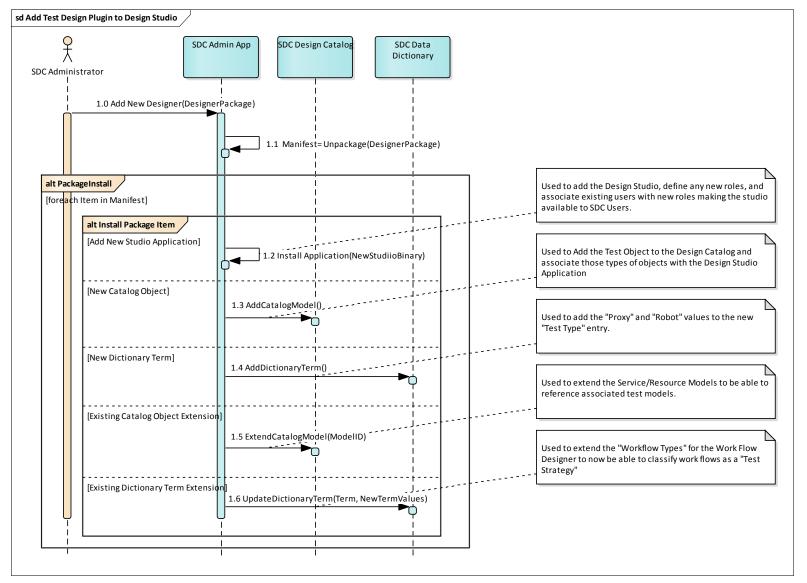
#### Run Time

- Manually Execute Test
  - vFW Demo Test (Does Instantiate)
  - vFW Health Check (From Inventory, must be instantiated)

Note: Post deployment of Ansible scripts used for VNF testing would require a Test Object to be created using the Proxy Test Head and then distributed to the OTF — Test Control Unit in order to expose it to operations via OTF Test Manager.

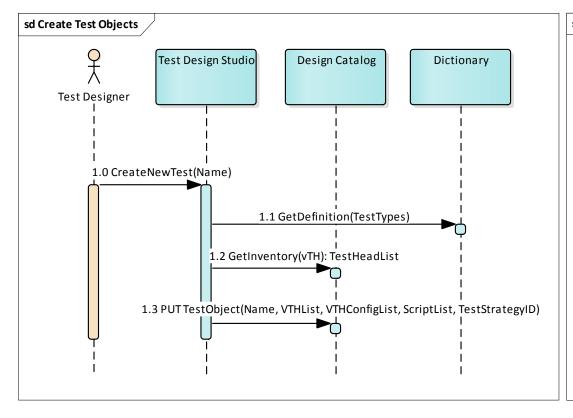


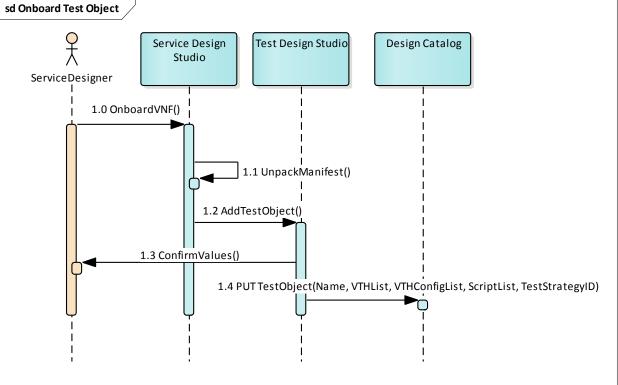
## Add Test Designer Plugin to Design Studio





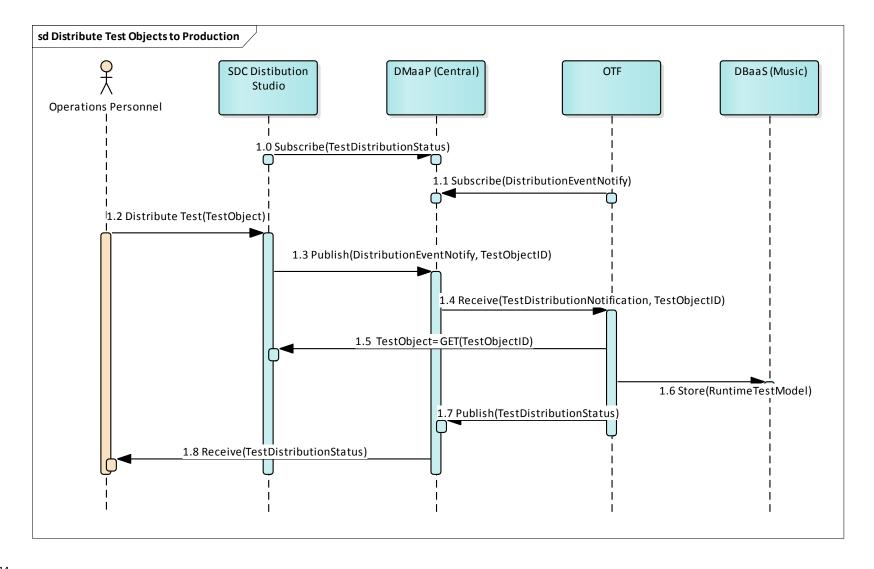
## Create Test Objects







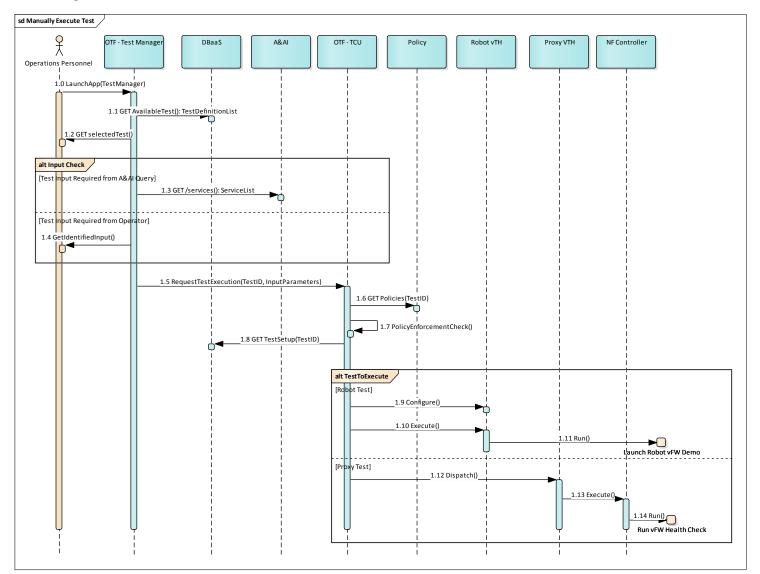
#### Distribute Test



Assumes pre-Run Time Catalog flow structure. Message flow would be different once RTC is available.



## Manually Initiate Test





## Backup



#### Testing & Certification Requirements Across Service Lifecycle

**Operate** Phase Design Certification Onboarding (Pre-Production Certification) Deploy *Assure* Step Service Cloud **Functional** Interoperability Service Configuration Task Verification **Validation Validation Diagnostics Validation**  VNF Isolation Service Isolation • Service Configuration Connectivity Testing Stand Alone VNF with **Supplier Specifications**  Stand Alone VNF With Integrated with Other Service Model containing • Link, Node, and Service Testing Service Provider **VNFs VNF** Resources Health Checks Are required artifacts **Specifications**  Passive Testing Compliant to Guidelines Verify Inter-VNF Does the Service Model Active Testing for the specified cloud Verify VNF meets Service functions operate End-to-Configure the VNFs Provider defined End Correctly • Trouble Isolation • Install per Vendor Specs functionality in Service Provider Cloud • Does the Service operate correctly through all Validate VNF Operates **Lifecycle Events** Walled Garden Sandbox **Production** 

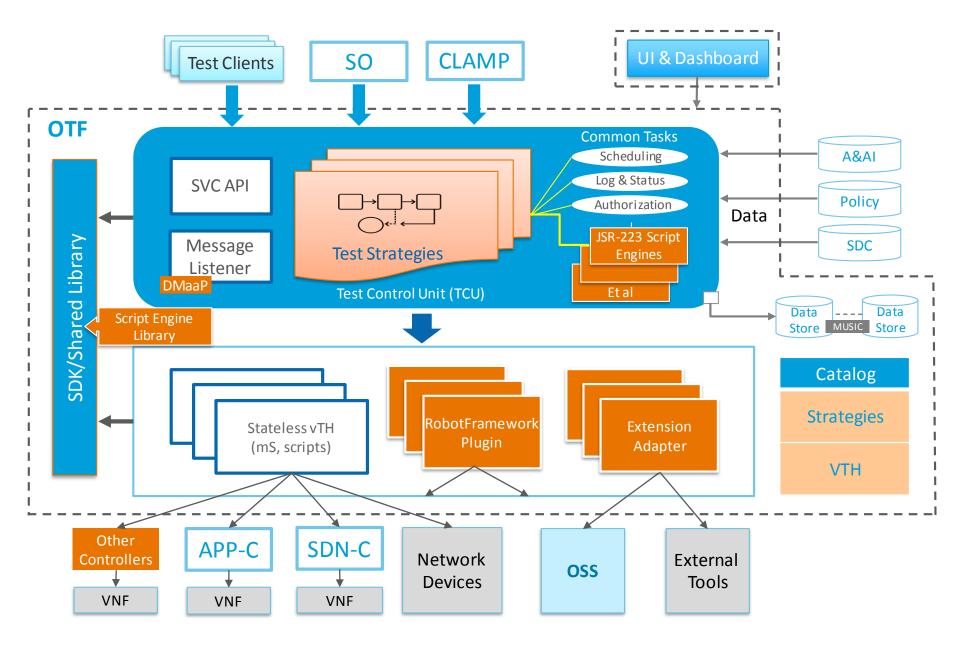


## Testing Recommendations

- Develop mechanisms for standardized onboarding of Network Tests even regardless of developer
  - Could be data only, leveraging existing test functions
- Develop model driven mechanisms for standardized test invocation:
  - As part of Certification Testing
  - On Demand
  - As part of Service Orchestration
  - As part of a Control Loop
- Standardize access to Network Functions
  - Via a managing controller, not direct to the element
- Standardized tooling, but not force it to be fixed
  - Tooling may differ for each Providers implementation of ONAP
  - Tools that are provided work in a cohesive pattern or roles within the framework
  - Tool set should be extensible
- Provide an SDK to standardize common functions leveraging platform capabilities



#### **OTF Component Development Design/Deep-dive**





## Dublin Focus Alignment

#### Operator Adoption/Flexibility/Deployability

Vendor Provided Certification Tests can be easily re-executed in Operators Instance

#### Model Driven Platform (Service/Resource)

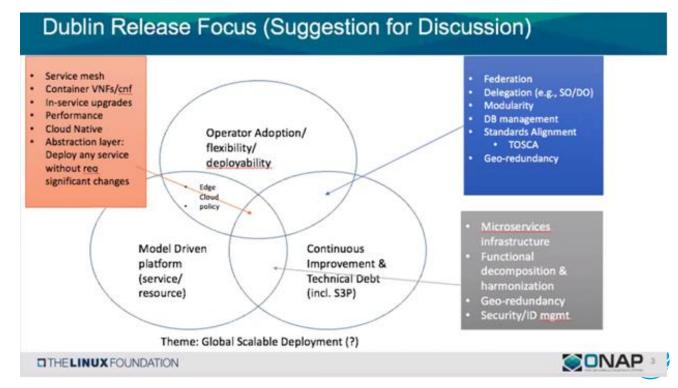
- Formalizes Test as part of Data Model

#### Continuous Improvement & Technical Debt

Employs Data Centric Architecture

#### Aligns with ARC Priorities

- Microservices
- Modular
- Standards Alignment
- API Standardization/Improvement
- Containers
- Service Mesh



## Sample Ping Test Strategy (BPM)

