

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:

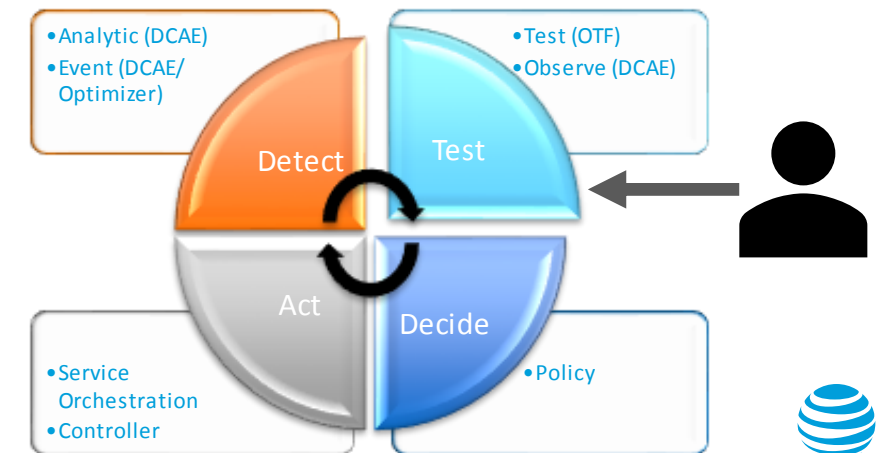
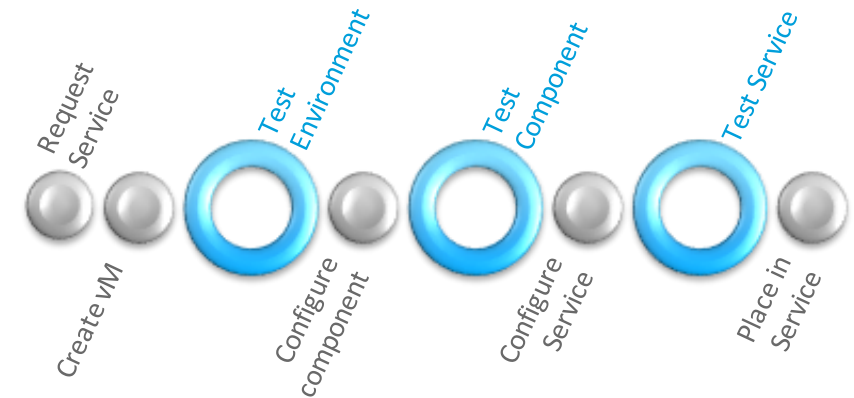
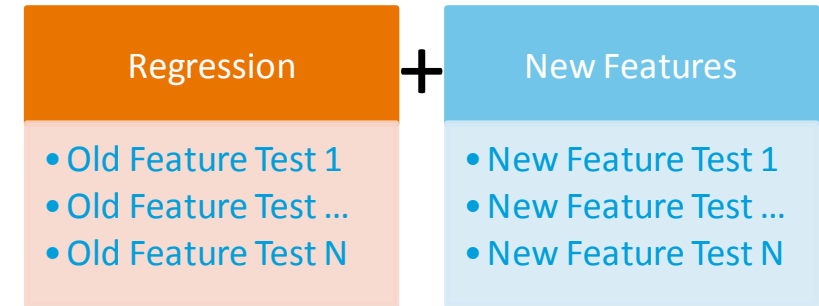
- Done in Lab or Sandbox environment
- Focus on verifying the functionality within the service design process
- May use tools not available in production environment: simulators, emulators, fault injectors
- ❖ **OTF provides the simulators, emulators, fault injectors and data to configure and execute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



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)**

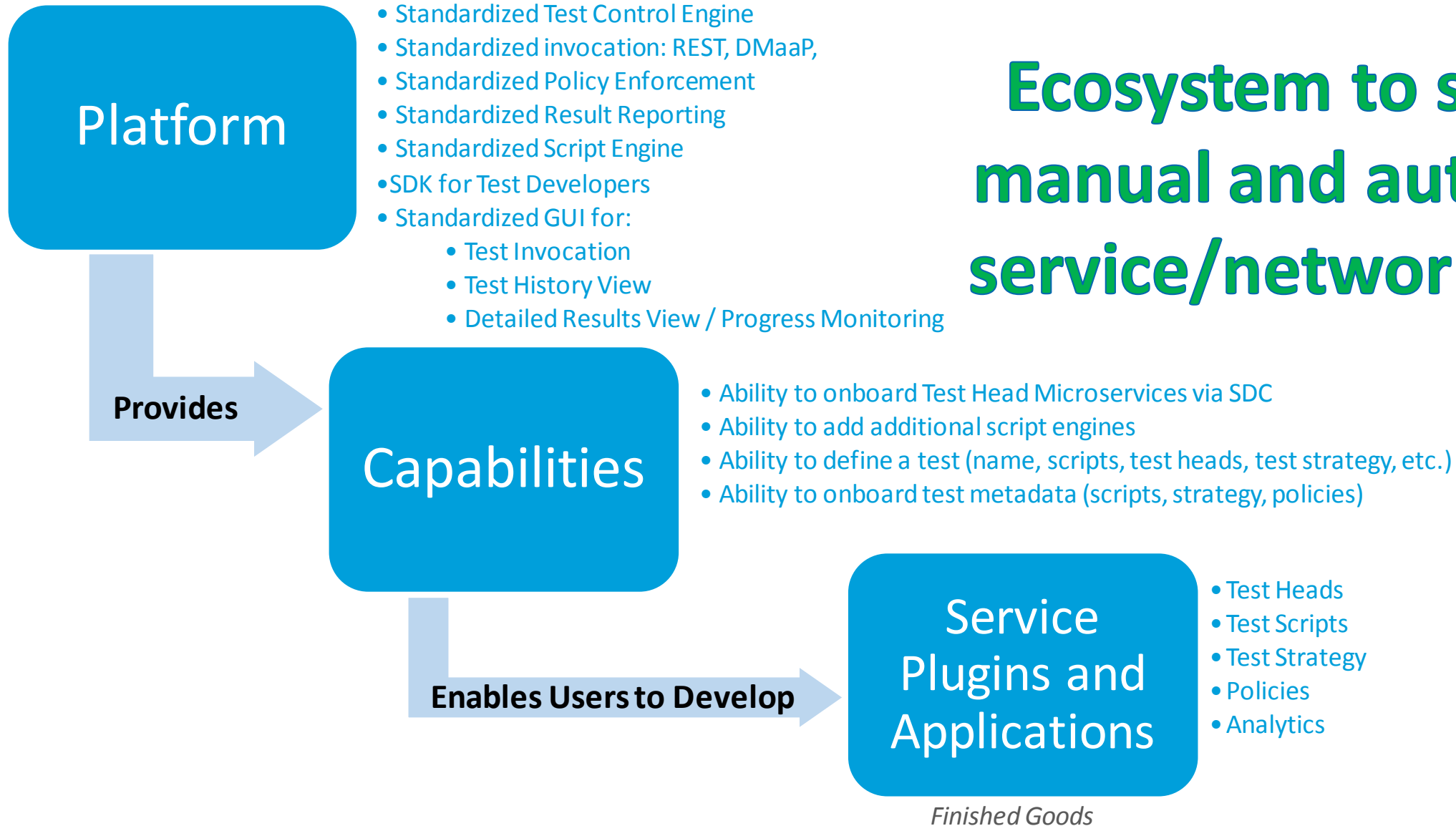
➤ **Test Scripting/Software and Test Head Suppliers**

- 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.**



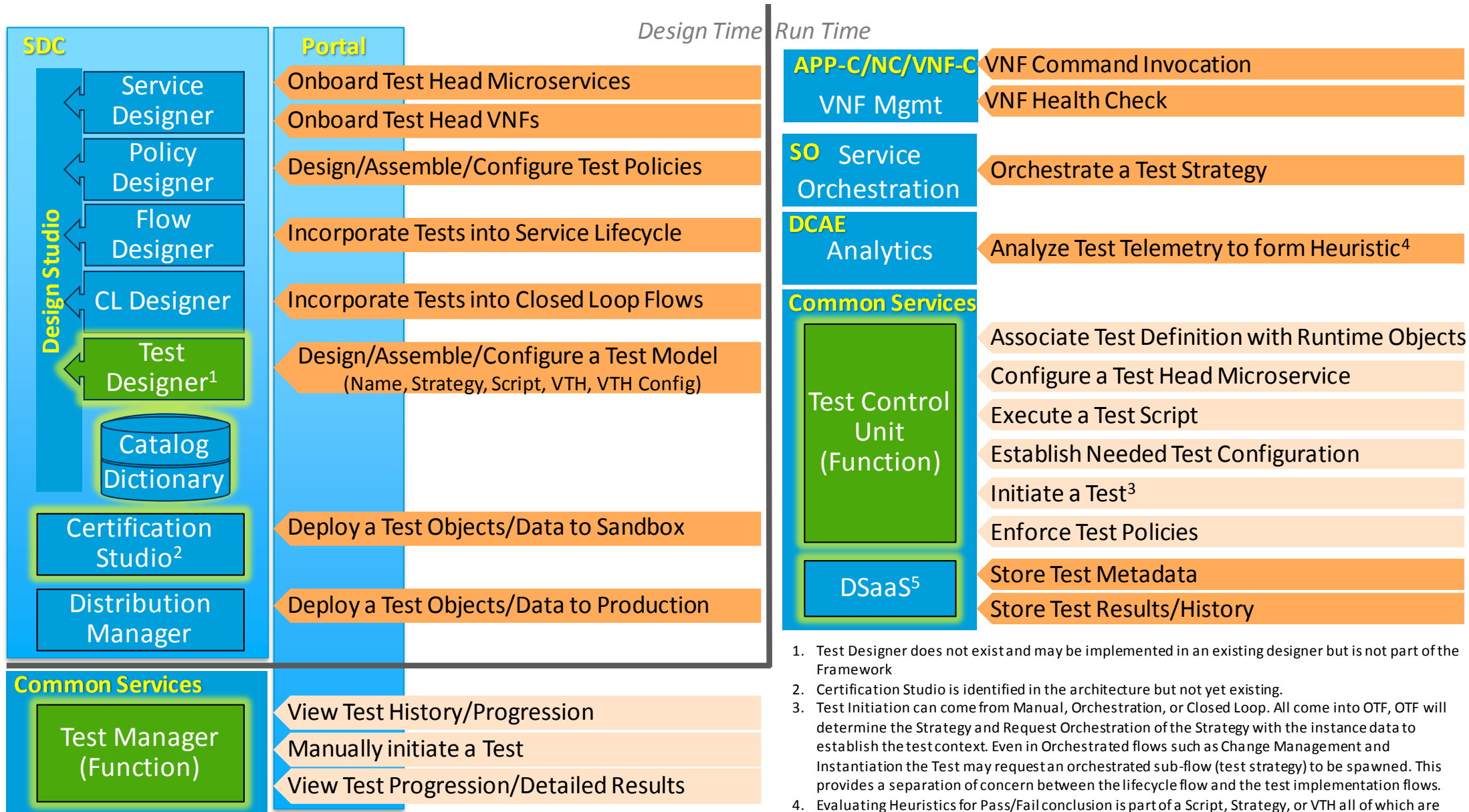
What exactly is the Open Test Framework



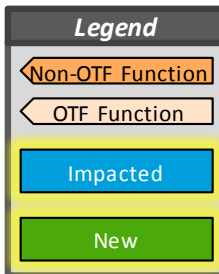
Ecosystem to support manual and automated service/network testing



What is new to ONAP?

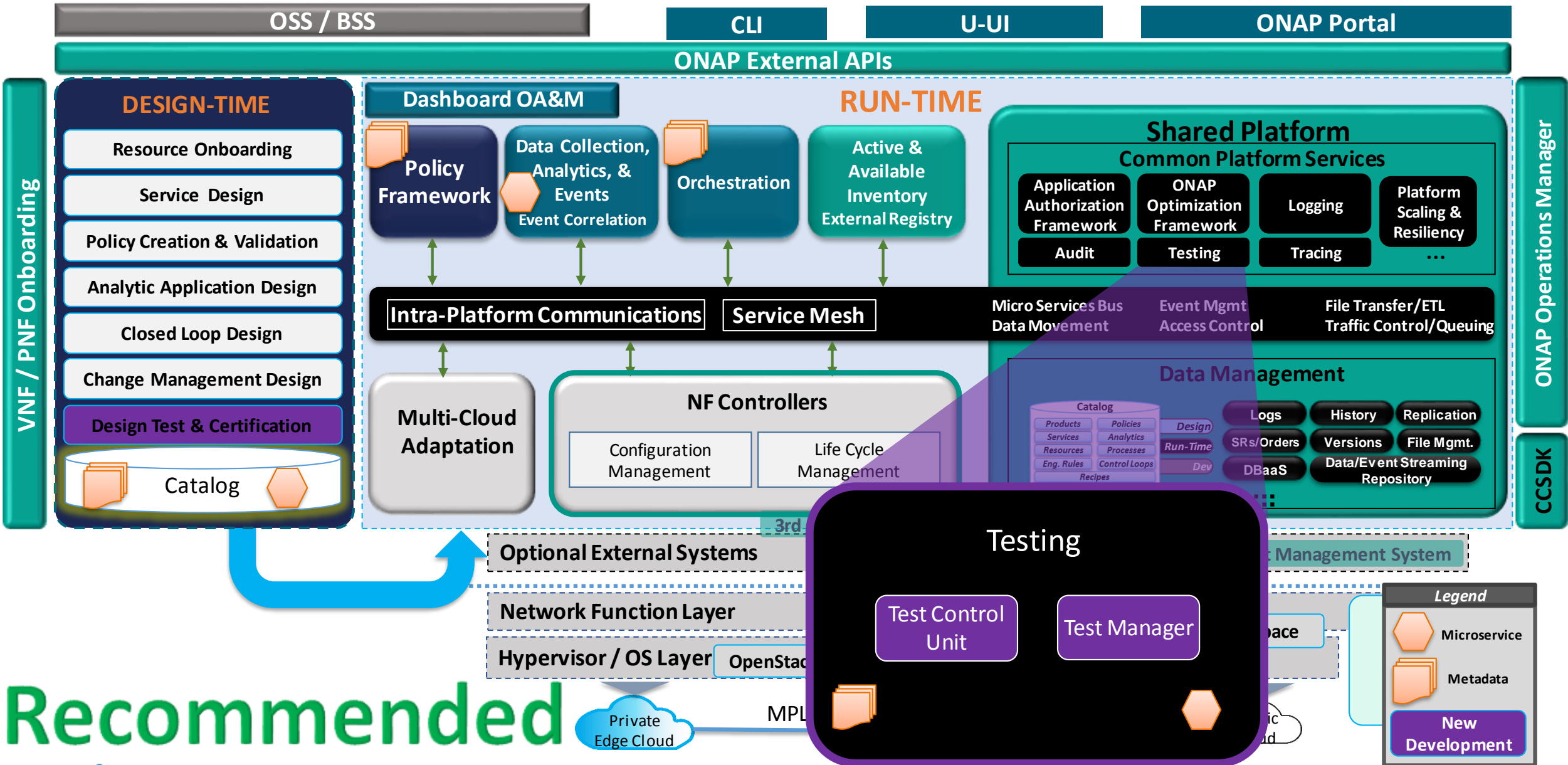


1. Test Designer does not exist and may be implemented in an existing designer but is not part of the Framework
2. Certification Studio is identified in the architecture but not yet existing.
3. Test Initiation can come from Manual, Orchestration, or Closed Loop. All come into OTF, OTF will determine 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



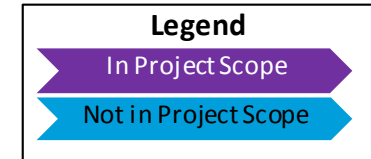
How does it fit in the ONAP Architecture?

As a common platform service similar to OOF

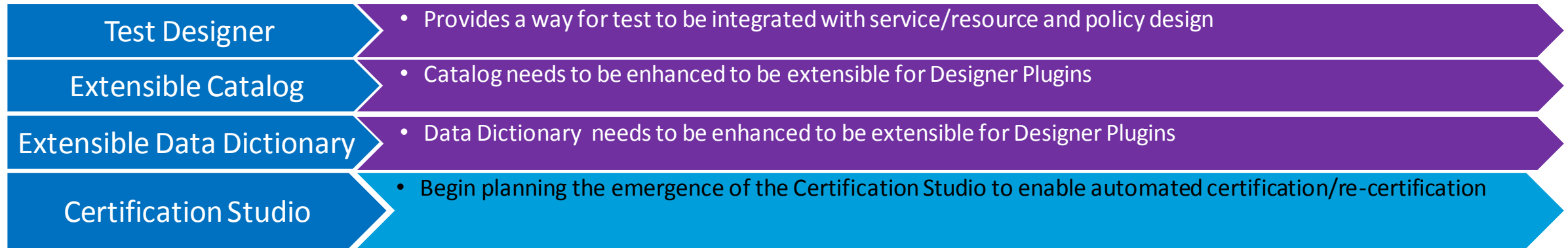


Recommended

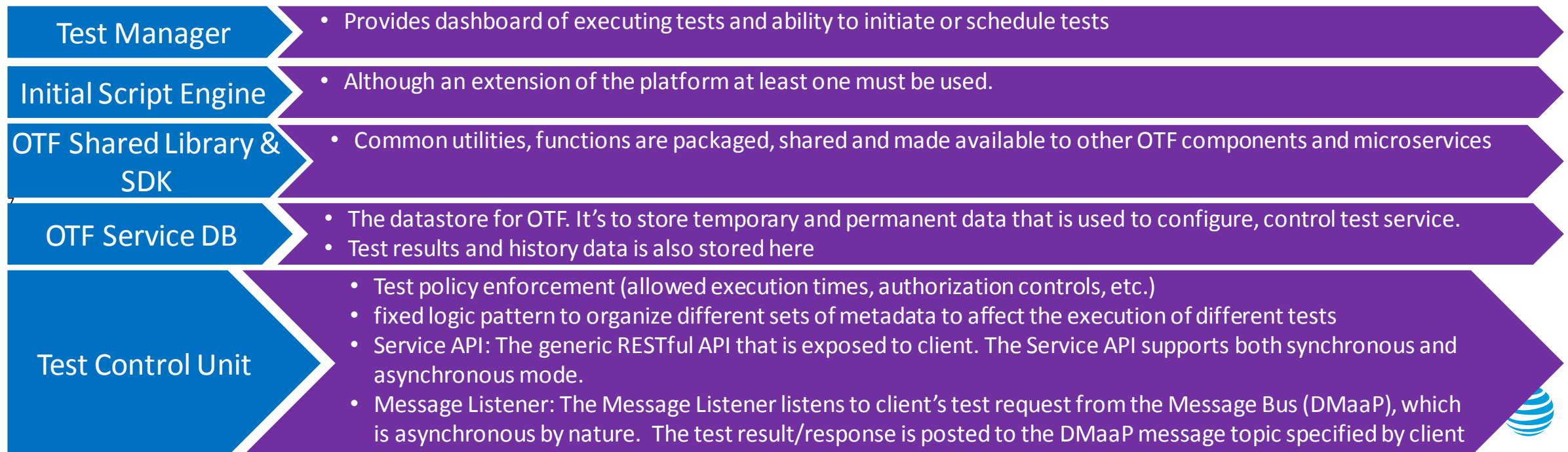
What platform development is needed for OTF?



SDC

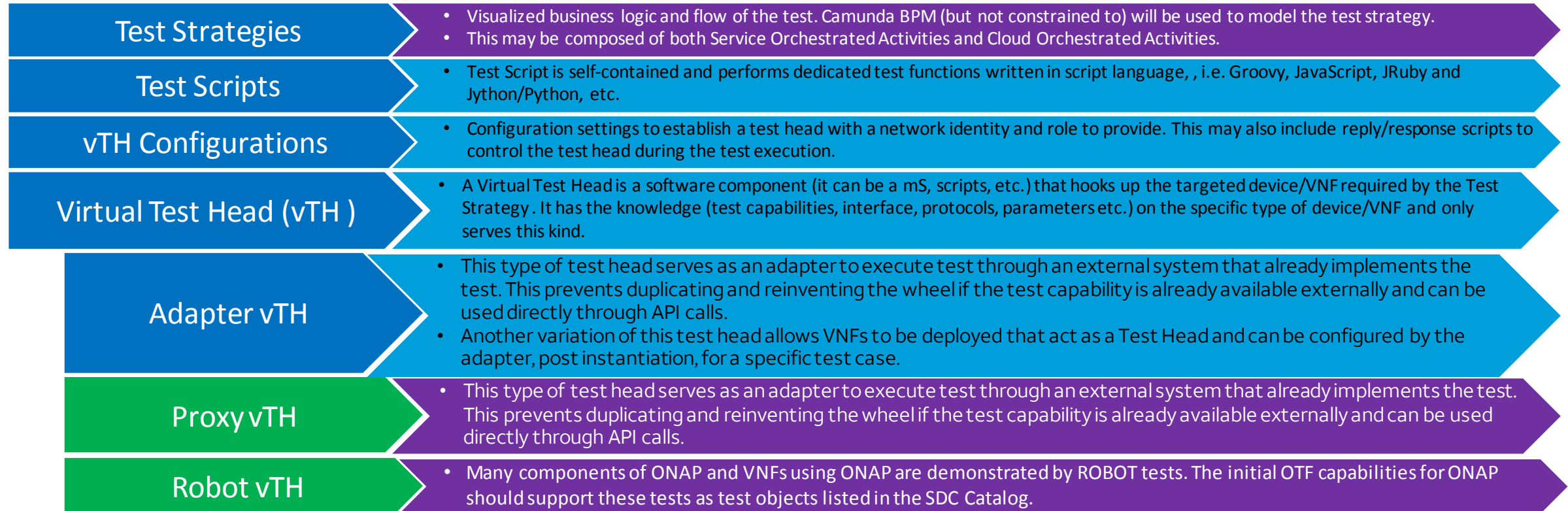
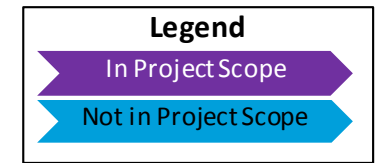


OTF

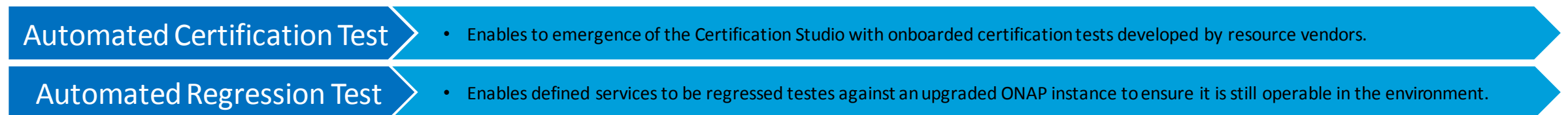


What finished goods and/or improved operations can be developed as a result of this project?

Finished Goods



Operations Improvements



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

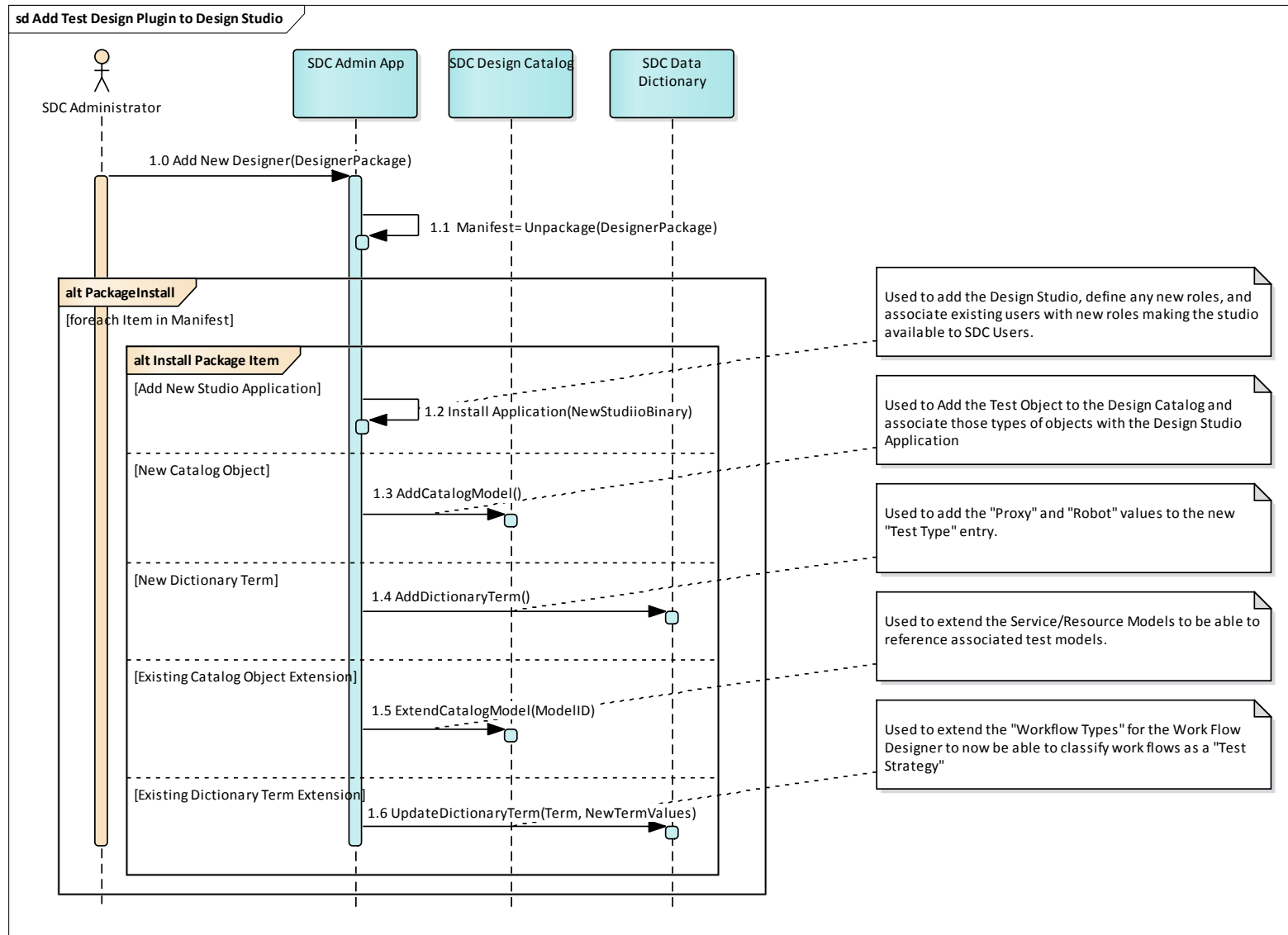
Note: Post deployment of Ansible scripts used for VNF testing would require a Test Object to be created using the Proxy Test Head and then distributed to the OTF – Test Control Unit in order to expose it to operations via OTF Test Manager.

Run Time

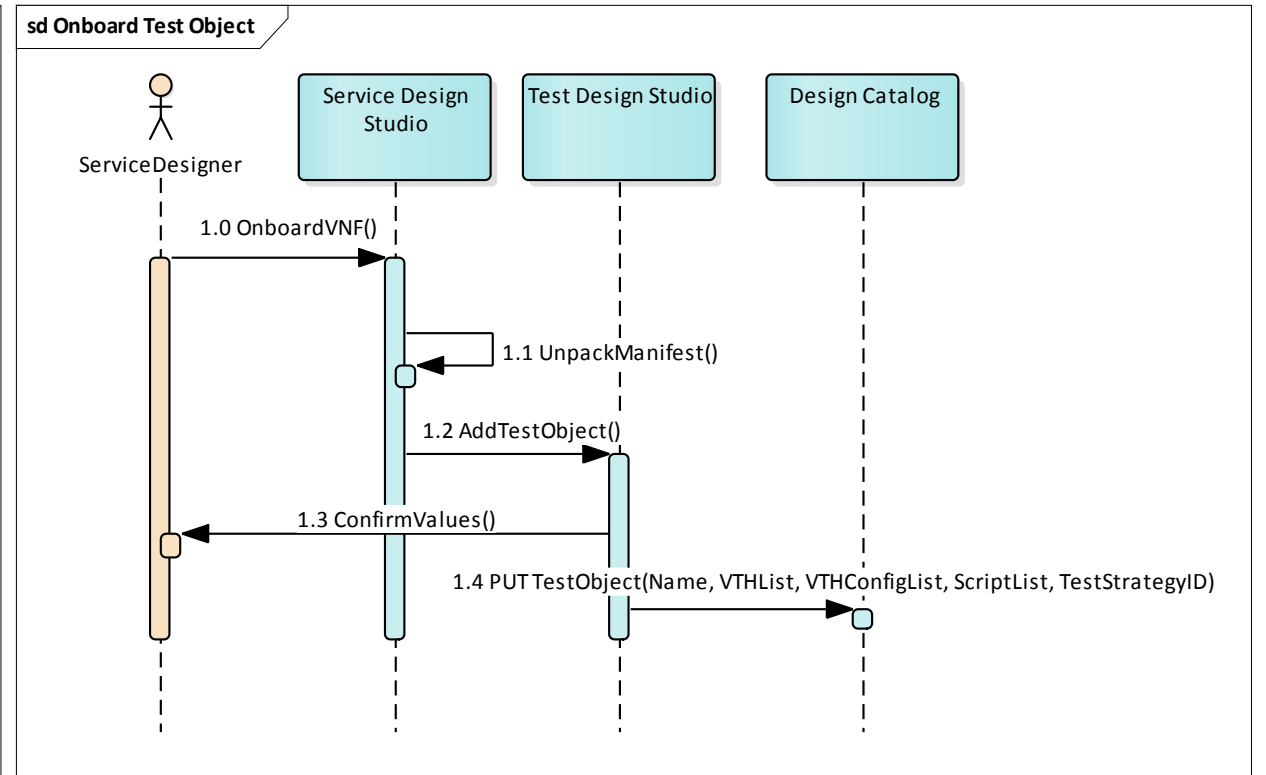
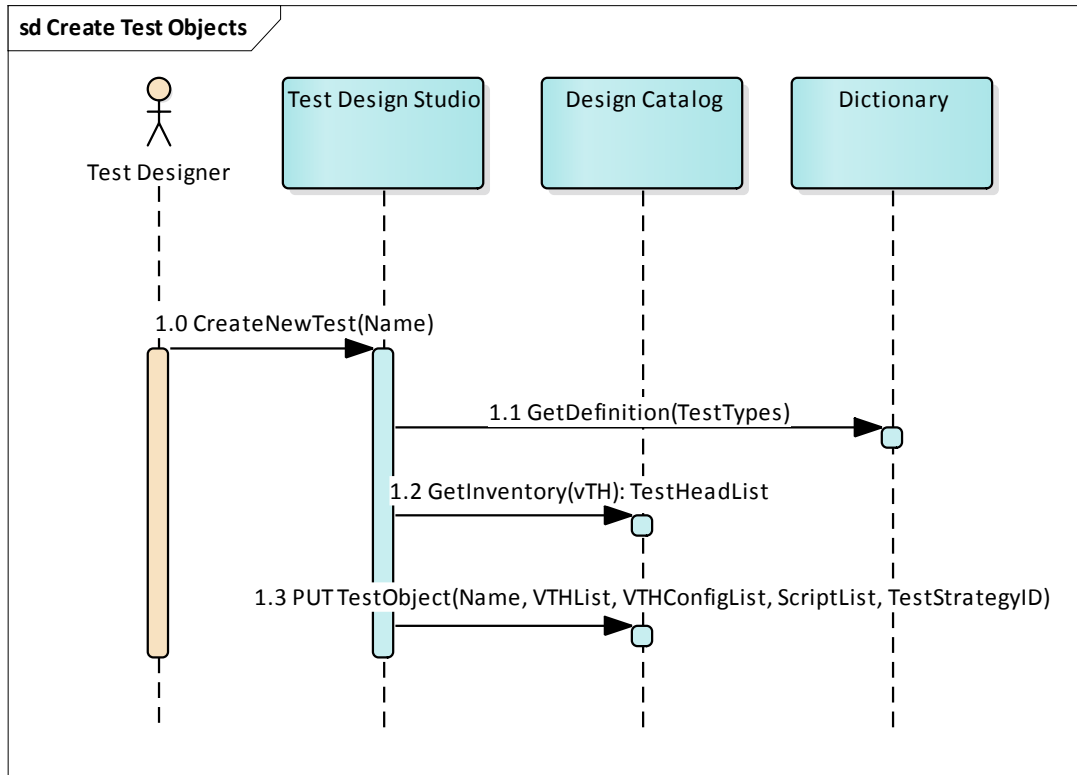
- Manually Execute Test
 - vFW Demo Test (Does Instantiate)
 - vFW Health Check (From Inventory, must be instantiated)



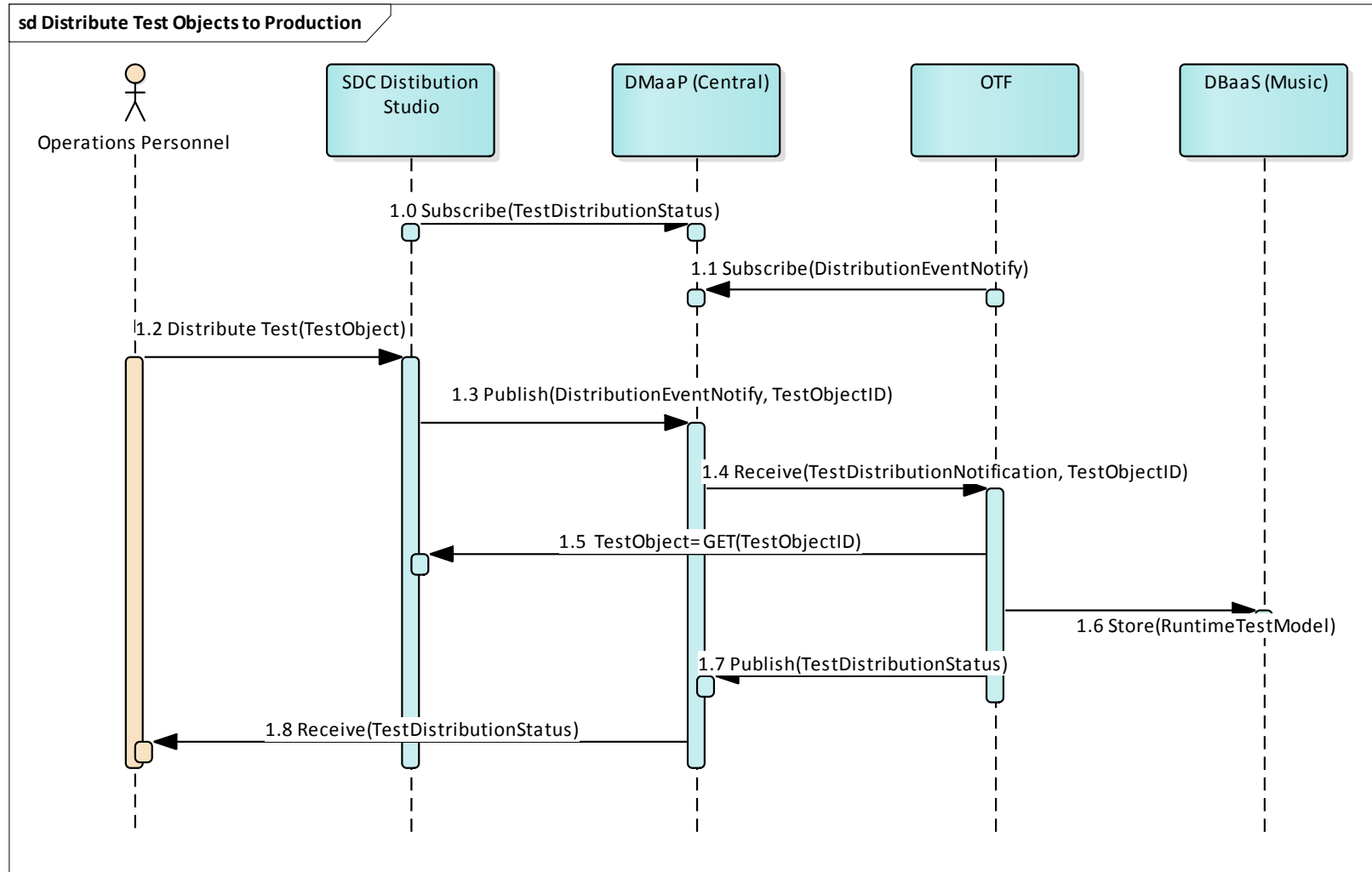
Add Test Designer Plugin to Design Studio



Create Test Objects



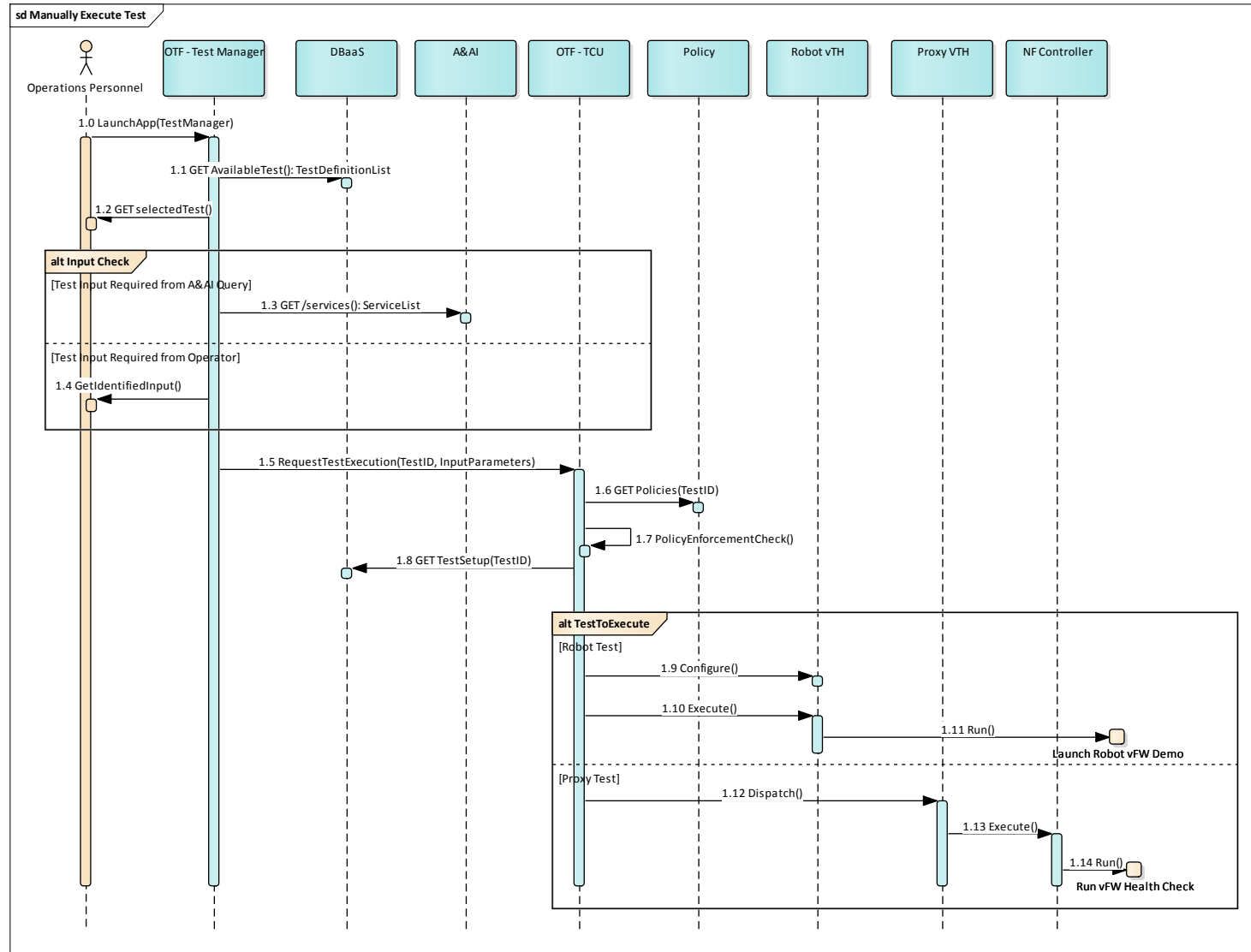
Distribute Test



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



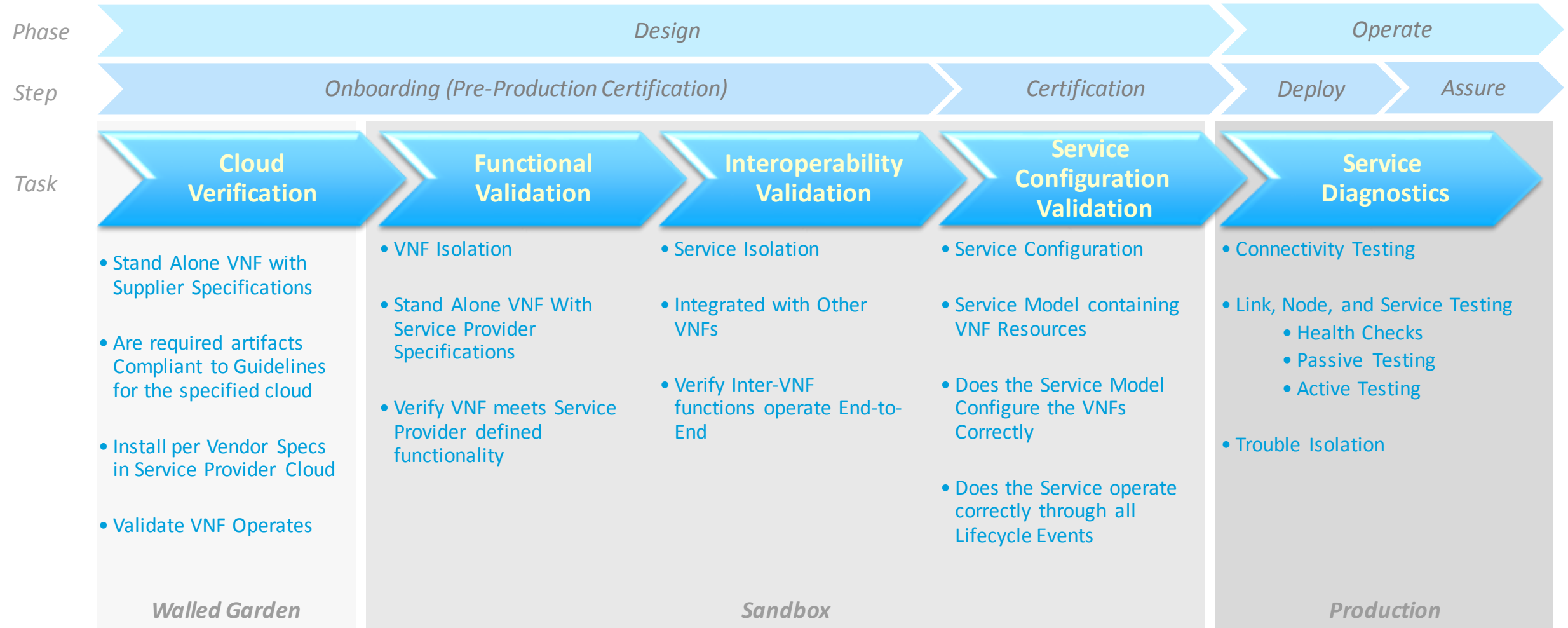
Manually Initiate Test



Backup



Testing & Certification Requirements Across Service Lifecycle

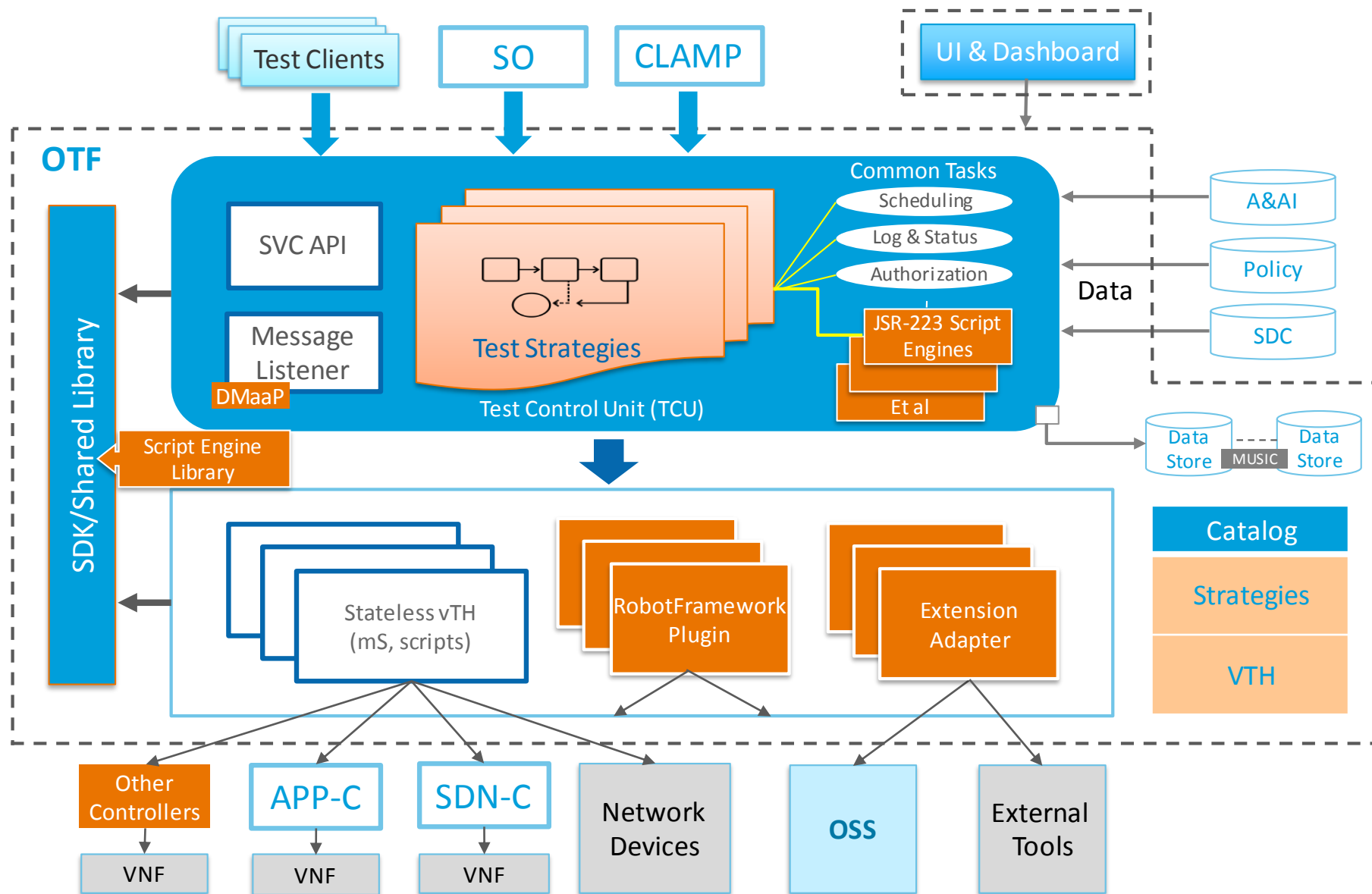


Testing Recommendations

- 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)

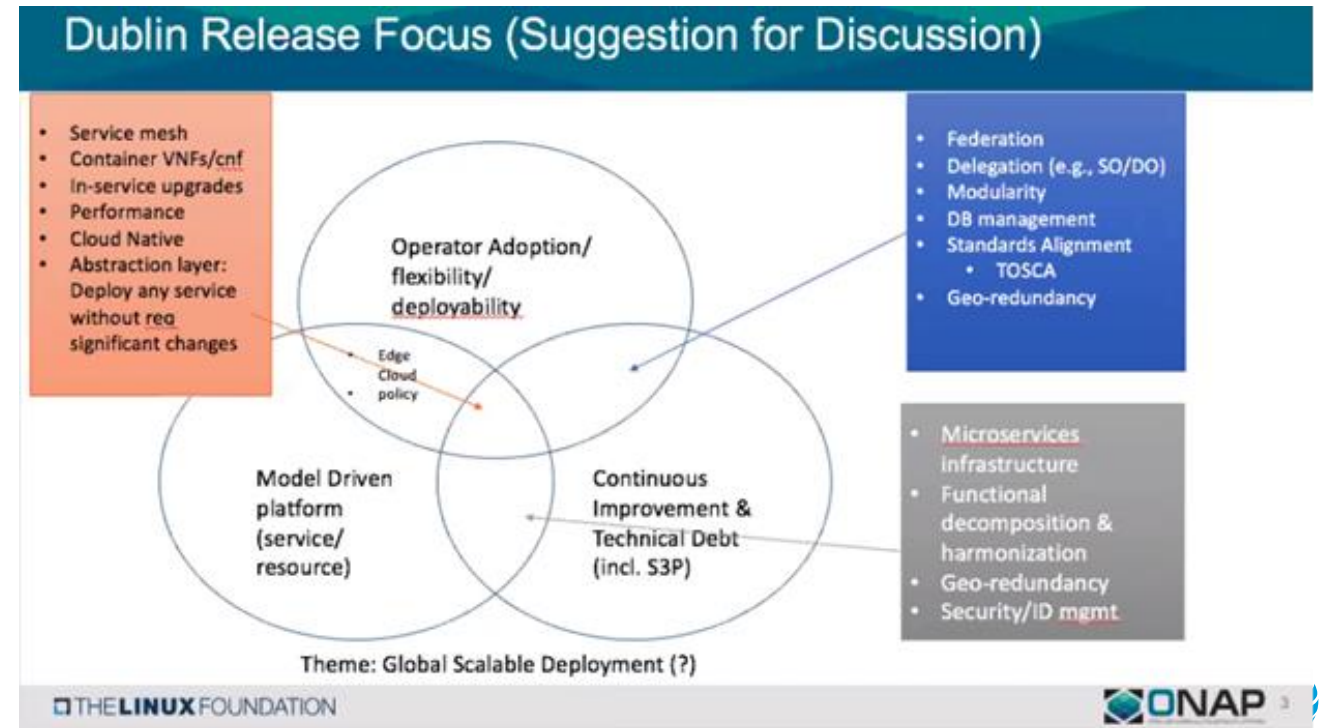
- Formalizes Test as part of Data Model

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)

