AAI R4 Architecture Review - 5 Mar 2019

• Brief Project Overview (brief as it should be known
  • Active and Available Inventory (AAI) is the ONAP subsystem that provides real-time views of Resources and Services and their relationships. AAI not only forms a registry of active, available, and assigned assets, it also maintains up-to-date views of the multidimensional relationships among these assets, including their relevance to different components of ONAP.

  This project targets a logically centralized reference point for service and resource details serving other ONAP components and non-ONAP systems to enable fulfillment, closed loop, reporting, and other operational use cases. A&AI is critical to ONAP as the existing sources of truth do not provide a cross domain view and are not designed to serve this information to multiple clients.

• New component capabilities for Dublin, i.e. the functional enhancements.

  - AAI-2096 - PNF Plug and Play
  - AAI-2097 - PNF Pre-Onboarding & Onboarding
  - AAI-2099 - 5G Network Slicing
  - AAI-2100 - Change Management - Traffic Migration Workflow
  - AAI-2104 - 5G PNF Software Upgrade
  - AAI-1990 - BBS: Support BBS use case in AAI
  - AAI-1998 - Support CCVPN Extension use case in AAI
  - AAI-1340 - Schema Changes for K8S

• New or modified interfaces
  • Most of the changes are related to the functional use cases; some of the functional use cases are test-only, they use existing types and edge rules.
  • The most contentious change is in the PNF object, there is a request to change the key of the object to pnf-id from pnf-name

  If they are modified, are they backwards compatible?

  • All are backward compatible except for AAI-2148 - Change key on PNF object from pnf-name to pnf-id

• Interface naming (point to an example)
  • AAI REST API

Reference to the interfaces.
  • AAI REST API Documentation - Dublin
  • AAI REST API Documentation on readthedocs.io

• What are the system limits?
  • Multithreaded concurrent operations on the same objects can cause duplicates in the graph - we have implemented stickiness in the haproxy that sits between AAI and clients to keep a client bound to the same node which can mitigate this, but we do not force locking on updates which means that malicious or careless clients can cause duplicates in the graph. We provide cleanup utilities for data grooming, but we recommend that clients are careful not to send multiple requests on the same object within microseconds of one another.
  • Operators should pay special attention to giving cassandra enough space, because we have known issues where things go pretty bad (like unhelpful or just wrong error messages) when the disks fill up on the cassandra cluster nodes.

• Platform Maturity Targets
  • AAI lacks resources to move beyond what we achieved in Casablanca.
  • Consider CII Badge silver and security Level 2 a stretch goal
  • 55% code coverage, most repos already passing but a few are between 50% and 55%

• Involved use cases, architectural capabilities or functional requirements.
  • AAI R4 Use Case and Functional Requirements Impacts

• Listing of new or impacted models used by the project (for information only).
  • The modeling subcommittee has requested that AAI leads an effort to produce a UML data model from the existing run time model (Dublin). Future decisions and updates to the model will then be approved and updated on the Papyrus model, thereby providing better linkage between ONAP modeling efforts and the downstream systems.

  • The PNF model in particular (see above AAI-2148 - Change key on PNF object from pnf-name to pnf-id needs attention in Dublin, we have reached out to Andy Mayer to have it considered in the context of the overall Modeling effort.