Active and Available Inventory (AAI) (5/11/17)

Project Name:
- Proposed name for the project: Active and Available Inventory
- Proposed name for the repository: aai

Project description:
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.

Scope:

- **Inventory for services and resources** (core capability to be delivered in R1)
  - Problem being addressed: Service and Resource information is critical to operational processes throughout the life cycle. When information is needed it should be made available quickly, accurately, reliably without overly impacting components in the ecosystem. Information must be made immediately available for newly defined service and resources.
  - Delivering a single point of reference for service and resource details
  - a central registry to create a global view of inventory and network topology
  - Receives updates from various Clients, to persist & maintain a global view.
  - As transactions occur, AAI persists asset attributes and relationships into the view based on configurable metadata definitions.
  - Provides a set of APIs to enable queries from various clients regarding inventory and topology. Queries can be supported for a specific asset or a collection of assets.
  - metadata-driven, new resources and services can be added quickly with Service Design and Creation (SDC) catalog definitions, using the AAI model loader.
  - A&AI will capture, maintain, and expose the data required to support the 3 ONAP R1 use-cases

- **Deliverables:**

<table>
<thead>
<tr>
<th>Service /Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Loader</td>
<td>Pulls in models from SDC topic queue and loads them to AAI inventory.</td>
</tr>
<tr>
<td>Resources</td>
<td>CRUD REST API endpoints for AAI inventory resources</td>
</tr>
<tr>
<td>Traversal</td>
<td>REST APIs for graph traversals. Some APIs use the Gremlin Server.</td>
</tr>
<tr>
<td>UI</td>
<td>used by operations for visualizing and searching. Available through ONAP Portal.</td>
</tr>
<tr>
<td>Data Router</td>
<td>Makes decisions about workloads to be dispatched to database abstraction services (including the search service). Includes logic to recognize and direct requests based on request archetypes.</td>
</tr>
<tr>
<td>Search</td>
<td>Enable complex searches for sub graphs and functions that need to perform well across deeply nested structures using Elastic Search. Used by UI.</td>
</tr>
<tr>
<td>Champ</td>
<td>Abstraction from underlying graph storage systems that A&amp;AI would interface with.</td>
</tr>
<tr>
<td>Gizmo</td>
<td>CRUD REST API endpoints for resources and relationships, delivering atomic interactions for improved scalability</td>
</tr>
</tbody>
</table>

- **Move to Active Open Source Graph Database** (R1 candidate)
  - Problem being addressed:
    - OpenECOMP AAI is today delivered on TitanDB in ONAP
    - TitanDB has no active Open Source community. No community updates for over 1 year.
    - ONAP AAI needs to move away from alignment to Titan There is risk in aligning AAI to TitanDB due to lack of support, maintenance and ongoing development.
  - Main scope for this feature is technical evaluation, selection, implementation of open source graph database options.
    - Janus good candidate ([http://janusgraph.org/](http://janusgraph.org/))
    - Janus supports Tinkerpop Abstraction implemented today in AAI.
    - Janus is an evolution of TitanDB (fork)
For support of Active-Active, Cassandra back-end is needed (rather than HBASE)

**Extend Model-driven AAI Use cases (R1 candidate)**

- Problem being addressed:
  - AAI needs to dynamically operationalize new and updated models at run-time, with minimal downtime and no coding, so that ONAP is always on, and that latest service and resource models can be delivered by ONAP quickly, and without release boundary.
  - To do this, AAI must update its internal model, external API and behavior at run-time to respond to change to service and resource models, including schema changes.
  - OpenECOMP AAI today delivers a model loader capability that ingests native A&A models and generates the API.
  - This capability needs to be extended to support:
    - Retrieve the common model artifact(s) (TOSCA assumed) for schema and for model
    - Translate the common model artifact(s) to AAI native artifact(s)
    - Ingest and operationalize the new models - making appropriate changes to data stores, APIs, configurations
    - Operationalization should include an internal 2 phase commit to ensure that relevant AAI services are successfully updated
    - Operationalization should include an external 2 phase commit across all relevant ONAP components to ensure they are successfully updated.
  - Decomposing AAI model/schema artefacts - today’s OXM is a monolith, a more granular approach will better enable extensibility and support logical subdivision of mods
  - First priority is the aspects of this behavior required for the 3 ONAP R1 use-cases

- **Track Change through Time (R2+ candidate)**
  - Problem being addressed:
    - Some assurance use-cases require investigation of the historical state of a service or resource (i.e. there was severe service degradation last Tuesday, what did the network/service look like then)
    - A&AI today delivers insight only into current service and resource.
    - Tracking the changes to Service and Resource inventory across time
    - Support for point in time service/resource detail, e.g. for assurance
    - Updated API to support query of time series data

- **AAI Reconciliation from Network/Cloud.** (dependencies on Multi VIM project - assume R2 earliest)
  - Problem being addressed:
    - Information supplied by AAI needs to be accurate to minimize fallout in operational processes.
    - Maintaining an accurate view of the service/resource is made more challenging by increased autonomy within the network
    - AAI needs mechanisms to ensure tight alignment with network, without significant additional load to the controllers, etc.
    - Need to refresh from VIM
    - data integrity checks and reconciliation
    - event based updates from VIM/SO/Controller
    - Different VIMs have different levels of detail.
    - SO has this UC also.
    - Variety of different options here (VIM/SO/Controller) - all valid.

**Related work**

Scalable, HA AAI (assumed covered by ONAP level project - Project does not yet exist for this)

- Back end needs to be HA. Need to confirm, explore.
- Testing vs performance, throughput
- Scaling AAI

Distributed AAI (assumed covered by ONAP level project - Project does not yet exist for this)

- How does AAI serve local orchestrator, local DCAE across large geographical regions
- Resource data, interim data cant be centralized - too costly.

**Architecture Alignment:**
How does this project fit into the rest of the ONAP Architecture?
- AAI is one of the key functional blocks in the ONAP Architecture
- What other ONAP projects does this project depend on?
  - SDC / Modelling
  - Multi VIM / SO / SDN-C
  - CommServ1 / MSB
  - Integration
  - OpenLab
- How does this align with external standards/specifications?
  - APIs/Interfaces - TinkerPop, Gremlin
  - Information/data models - ONAP TOSCA model
- Are there dependencies with other open source projects?
  - APIs/Interfaces
    - TitanDB
    - Hbase
    - AJSC
    - EELF
    - DMAAP
  - Integration Testing

Resources:
- Primary Contact Person - Manisha Aggarwal (ma9181@att.com), Colin Burns (Colin.Burns@amdocs.com)

Table of Committers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Gerrit ID</th>
<th>Company</th>
<th>Email</th>
<th>Time Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manisha Aggarwal</td>
<td>016991</td>
<td>A&amp;T</td>
<td><a href="mailto:ma9181@att.com">ma9181@att.com</a></td>
<td>Middletown, NJ, USA, UTC -4</td>
</tr>
<tr>
<td>Jimmy Forsyth</td>
<td>016992</td>
<td>A&amp;T</td>
<td><a href="mailto:jf2512@att.com">jf2512@att.com</a></td>
<td>Michigan, USA, UTC -4</td>
</tr>
<tr>
<td>Steven Blimkie</td>
<td>sblimkie</td>
<td>Amdocs</td>
<td><a href="mailto:Steven.Blimkie@amdocs.com">Steven.Blimkie@amdocs.com</a></td>
<td>Ottawa, Canada, UTC -4</td>
</tr>
<tr>
<td>Tian Lee</td>
<td>TianL</td>
<td>Amdocs</td>
<td><a href="mailto:tian.lee@amdocs.com">tian.lee@amdocs.com</a></td>
<td>Bath, UK, UTC +1</td>
</tr>
<tr>
<td>Heliu Zhong</td>
<td>016994</td>
<td>BOCO</td>
<td><a href="mailto:zhongheliu@boco.com.cn">zhongheliu@boco.com.cn</a></td>
<td>Beijing, China, UTC +8</td>
</tr>
<tr>
<td>Yuanwei Yang</td>
<td>016995</td>
<td>BOCO</td>
<td><a href="mailto:yangyuanwei@boco.com.cn">yangyuanwei@boco.com.cn</a></td>
<td>Beijing, China, UTC +8</td>
</tr>
</tbody>
</table>

Table of Contributors:
### Project Roles (include RACI chart, if applicable)

### Other Information:
- link to seed code (if applicable)
  - gerrit.onap.org/r/a/aai/
- Vendor Neutral
  - if the proposal is coming from an existing proprietary codebase, have you ensured that all proprietary trademarks, logos, product names, etc., have been removed?

  The current seed code has been already scanned and cleanup to remove all proprietary trademarks, logos, etc. except openecomp to be replaced by onap

  Subsequent modification to the existing seed code should continue to follow the same scanning and clean up principles.
- Meets Board policy (including IPR)

*Use the above information to create a key project facts section on your project page*

### Key Project Facts

**Primary Contact:** Manisha Aggarwal (AT&T), Colin Burns (Amdocs)  
**Project Lead:** Manisha Aggarwal (AT&T), Steve Blimkie (Amdocs)

**Project Name:**
- JIRA project name: Active and Available Inventory
- JIRA project prefix: AAI-

**Repo name:** Lifecycle State: **Primary Contact:** Project Lead: mailing list tag [Should match Jira Project Prefix]

**Committers:**
- ma9181@att.com AT&T
jf2512@att.com AT&T
Steven.Blimkie@amdocs.com Amdocs
TianL@amdocs.com
zhongheliu@boco.com.cn
yangyuanwei@boco.com.cn

Link to TSC approval:
Link to approval of additional submitters: