Portal Platform Project Proposal (5/12/17)

ONAP Portal

Project Name:

- Proposed name for the project: ONAP Portal Platform
- Proposed name for the repository: "portal", "ecompsdkos"

Project description:

- The ONAP Portal is a platform that provides the ability to integrate different ONAP applications into a centralized Portal Core.
- The intention is to allow decentralized applications to run within their own infrastructure while providing common management services and connectivity.
- The Portal core provides capabilities including application onboarding & management, centralized access management, and hosted application widgets.
- Using the provided SDK, application developers can leverage the built-in capabilities (Services / API / UI controls) along with bundled tools and technologies.

Scope:

High-level Scope Overview:

- The Portal provides a web-based user interface that provides access to all of the subsystems of an instance of ONAP.
- It offers both design-time tools and run-time monitoring and control (for example: access to SDC, Policy, VID applications).
- Any user seeking access to an ONAP application will first visit the Portal, where authentication will be performed. Based on the user's configured access level, the Portal will let the user access different application widgets, and might also redirect the user to a specific run-time environment.
- From the Portal, users access applications and Key Performance Indicators.
- Administrators onboard and manage applications and hosted application widgets, and manage user access.

Detailed scope and purpose of Portal

ONAP applications can be divided into three main categories. Applications that are required at the design time, such as SDC, Policy creation, etc. Applications that support runtime functions such as AAI, SO, DCAE, SDNC, APPC and VFC. Finally, the applications that support administrative functions on the ONAP platform. As the number of applications grows, there is a need to have control over:

- What technologies are being used to develop?
- How can we ensure a consistent user experience is from one application to another?
- Who controls the access to the applications?
- How are applications managed and monitored. And how are new applications on-boarded?
- How can the applications expose features and APIs to be consumed by other systems?

To address this, a platform and an SDK are being built to aid teams, who are tasked with building these ONAP applications. The platform includes two major modules: the SDK and the Application Portal Core. The SDK will provide a base framework for development teams, and will help in developing and releasing applications into production in a rapid and consistent manner. The applications built on this SDK will be controlled by a centralized ONAP Portal Core. Here are the details of both.

SDK

The SDK module will be used as a base for any ONAP application being built. It bundles together a host of different tools, technologies, and standards that will assist the teams during the development phase. It includes reusable UI components, authentication & authorization components, visualization & reporting engine, collaborative services, workflow manager, GIS / Map, web component, and widget development framework.

Any application based on the SDK will have implicit connectivity with a centralized ONAP Portal Core via a message-oriented middleware. It will be used for sending and receiving any data from the ONAP Portal.

ONAP PORTAL CORE / RUNTIME

This will be a centralized web-based application running in the cloud environment and will be responsible for orchestrating all the other ONAP web applications. The ONAP Portal Core will include centralized access control, user management for internal/external users, dashboards, application administration, common UI controls, system monitoring/alarms/audit, system notifications, user messaging widget, context-aware UI controls, access history, personalization, service registration/discovery etc.

Any user seeking access to an ONAP application will first land on the ONAP Portal Core where authentication will be performed. Based on the user’s access level, the ONAP Portal Core will let the user access different application widgets, and may also redirect them to their own run-time environments.

Designing Services

The Portal provides the following design-time features:
Instantiating Services

The Portal offers a Virtual Instantiation Deployment (VID) GUI to trigger SO instantiation of Services and components that have been certified and distributed for production. These services may include:

- Infrastructure Services (such as compute and storage resources)
- Network Services (Virtual Network Functions)
- Application Services (such as a load-balancing function)

VID reads the models created in SDC, and, in turn, forwards the appropriate information to SO during the Service instantiation process.

Administration

From the ONAP Portal, administrators:

- access the same functionality accessible to users
- manage users and application admins
- onboard applications and widgets (developed using the Portal as a platform)
- edit the functional menu

Future enhancements in the following releases:

- Portal SDK - Digital Experience Control/UI Upgrade.
- Ability for admin to use notification and act on it w/o copy/paste, e.g. hyperlink to target function with context transfer.
- Enabling centralized Authentication and Authorization (AAF): Ability for centralized User Management and administrative tasks such as Role based access.

Architecture Alignment:

Overall ONAP Architecture showing relation to Portal component with other components:
How does this project fit into the rest of the ONAP Architecture?

- Please include architecture diagram if possible - [+][https://wiki.onap.org/display/DW/Portal]+
- What other ONAP projects does this project depend on? No dependecies. But will impact components - Policy, VID, SDC, DBC (DMaaS Bus Ctrl)

How does this align with external standards/specifications?

- For security - O-Auth

Are there dependencies with other open source projects?

- Application Authorization Framework (AAF)

Resources:

- Primary Contact Person: Manoop Talasila
- Names, gerrit IDs, and company affiliations of the committers
  - Manoop Talasila, talasila@research.att.com, AT&T
  - Sunder Tattavarada, statta@research.att.com, AT&T
- Names and affiliations of any other contributors
  - Aftab Syed Anwar, anwar@research.att.com
  - Chris Lott, cl7781h@att.com, AT&T
  - Wei-Ting(Robert) Lo, w8849v@att.com, AT&T
  - Anandarup Sarkar, as0481@att.com, AT&T
  - Leiming Shi, ls3297@att.com, AT&T
  - Lorraine A Welch, lb2391@att.com, AT&T
  - Abbas Fazal, af9900@att.com, AT&T
  - Ikram Ikramullah, iI041c@att.com, AT&T
  - Farhan Mir, fm7890@att.com, AT&T
  - Morales Victor, victor.morales@intel.com, Intel
  - Gary Wu, gary.i.wu@huawei.com, Huawei
  - Kwiecień Krzysztof, Krzysztof.Kwiecien1@orange.com, Orange
  - Kishore Koya, kk707x@att.com, AT&T (Primary Tester)
- Project Roles (include RACI chart, if applicable)

Other Information:

- link to seed code (if applicable) –
  - https://gerrit.onap.org/r/#/admin/projects/portal
  - https://gerrit.onap.org/r/#/admin/projects/ecompsockopt
- 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? yes
    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)
Key Project Facts

Project Name: Portal

- JIRA project name: Portal
- JIRA project prefix: Portal-

Repo name: portal, ecompsdkos
Lifecycle State: enhancement
Primary Contact: Manoop Talasila
Project Lead: Manoop Talasila
mailing list tag [Should match Jira Project Prefix]
Committers: talasila@research.att.com; statta@research.att.com
*Link to TSC approval:
Link to approval of additional submitters: