Akraino Feature Project - Regional Controller

Feature	Description	Companies Participating / Committers	Requested Release / Timeline	Informational
Akraino Regional Controller	The current Akraino Portal provides a user interface and a collection of workflows and services to execute the actions requested by the user. This proposal is to separate the workflows and services from the portal user interface so that actions can be performed through the portal, direct REST calls from an external orchestration tool, or a CLI that could be developed as part of a different feature project. 1. Define an API for the various actions that a user might request including user management, blueprint definition, blueprint deployment, monitoring, etc. 2. Modify existing services and workflows to be initiated though the new API. 3. Develop new services and workflows as needed to address common tasks required by blueprints. 4. Coordinate with the Portal feature project to use the new API. 5. Standardize the software definition of a "Blueprint" so that multiple software entities can interact with blueprints in a defined way. 6. Standardize the software definition of a "hardware profile" to enforce some rigor in what types of hardware individual blueprints may use.	AT&T Ericsson	R1	Impacted Blueprint Families - Network Cloud and Radio Edge Cloud but could be leveraged by others See attachment for additional details

Background – Lessons learned from the Initial Portal

The seed code Regional Controller (the RC) taught us several things:

- Too many Blueprint details are hardwired into the RC
- Impossible to dynamically add new Blueprints w/o changing the RC code
- Workflows (Camunda) unreadable and not very flexible
- No documented API which means the portal is the only way to access the RC (no CLI, no programmability)
- The portal itself is tied directly to the workflow engine, and is an inseparable part of the RC
- RBAC model not completed, which means all operations available to all users
- Was never completely Docker-ized, which caused problems with installation and startup (e.g. changing Java)

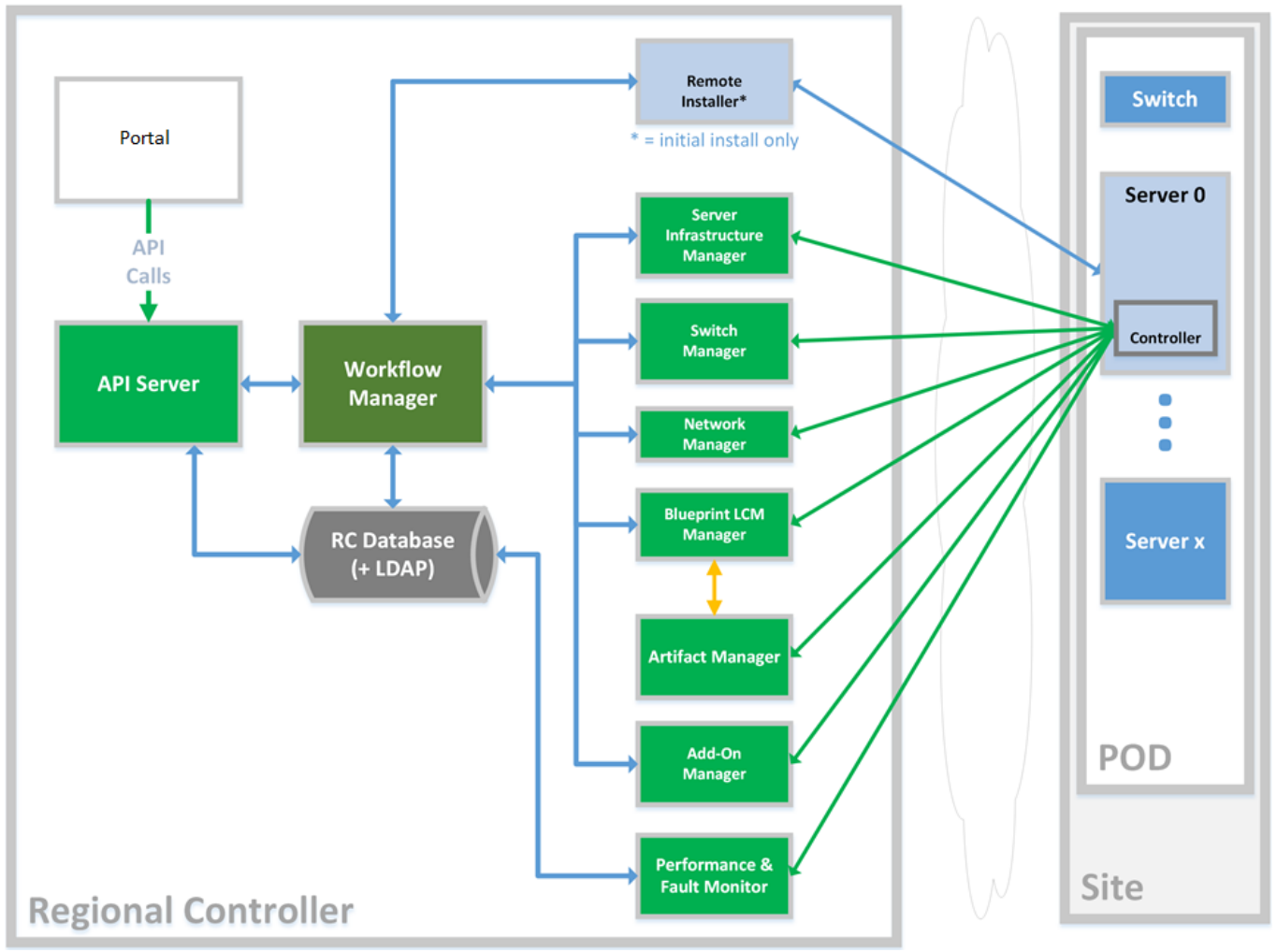
AKRAINO EDGE STACK

Goals for the new Akraino Regional Controller (the ARC)

- Begin a standardization process for defining a "blueprint" in a machine-readable form
  - ARC will define first cut at Blueprints, no doubt these will evolve
  - ARC will be the primary user of Blueprints, but others (e.g. Portal) may also use them
  - This *software blueprint* is all that is needed to instantiate a Blueprint on an Edgesite (a POD)
- Begin a standardization process for defining a "hardware profile"; that is a specific piece of hardware required for and by a Blueprint
  - Again ARC will provide first cut; these really need to be defined by individual Blueprints
  - Hardware profiles should be standardized across all Akraino Blueprints via some registry
- The ARC should manage a high level object model model (Nodes, Edgesites, PODs, Regions, etc.)
- The ARC should provide CRUD operations on the object model (Delete Node, Create POD, Modify Region, etc.)
- The ARC should provide a REST API to access the CRUD model
- The API can be accessed from multiple external actors besides a Portal (e.g. CLI, other programs)
- The ARC should have extensible RBAC model to allow control over who may do what
- The ARC should be completely Docker-ized, meaning it should be much easier to start
- The ARC should be completely agnostic about the how workflows do their work, and what operations are supported per Blueprint.
- Blueprint specifics should be removed from the ARC; pushed into Blueprints, Blueprint workflows, and Blueprint artifacts, thus allowing ARC to dynamically be able to support new Blueprints w/o code mods.
- The ARC should provide the capability to dynamically define a workflow for any action which may be required of a POD

Akraino Feature Project - Regional Controller

- All business logic and automation would be moved from the portal to the regional controller services
- Portal would use REST API calls to initiate actions requested by the user
- Portal and Regional Controller could share same server but not required
- Services are examples and need definition by the community



Akraino Feature Project - Regional Controller

Example Reginal Controller services based on community experience to date. The community will need to define the final list of services and APIs.

- **Remote Installer** – Manages the installation of POD's that utilize blueprints.
- **Workflow Manager** – A workflow engine used to sequence activities that are typically multi-step processes, such as creating PODs.
- **Switch Manager** – Configures switch initially and throughout the POD's lifecycle.
- **Server Manager** – Creates, modifies and decommissions bare metal servers in an Akraino POD.
- **Network Manager** – Configures networking within a given POD.
- **Blueprint LCM** – Interprets and realizes Blueprints as deployed PODs. Manages POD lifecycle. Leverages Server Manager, Artifact Manager, Software Manager and Performance/Fault Manager to manage any given POD from cradle to grave.
- **Artifact Manager** – Repository for all artifacts utilized by the Regional Controller to create and lifecycle manage POD's
- **Software Manager** – Deploys, upgrades and decommissions software on POD.
- **Performance/Fault Manager** – Proactively monitors all POD's under the control of a given Regional Controller. Auto-mitigates performance issues and faults, if possible. Logs performance and state characteristics of all POD's and the Regional Controller itself.
- **HW Management** – Manages hardware deployed.

AKRAINO EDGE STACK