Meeting of the **T**echnical **S**teering **C**ommittee of the Akraino Edge Stack Project

January 29, 2019

Agenda

› PTL Election Results

› PTL Responsibilities

› Introduction to Sub-Committees

› List of Blueprints (incubation approved)

› CI / Validation Framework

› Blueprint Process

› Release Targets & Action Items for PTLs

Blueprint / Feature Project PTL Election Results

Blueprint	Winner of PTL Election	Email Address
SDN Enabled Broadband Access (SEBA)	Aaron Byrd- AT&T	ab2745@att.com
Serverless	Itamar Eshet- AT&T	ie2575@intl.att.com
Unicycle	David Plunkett- AT&T	david.plunkett@att.com
Rover	David Plunkett- AT&T	david.plunkett@att.com
OVS-DPDK Unicycle Dell	Rakesh Bohra- Ericsson	rakesh.bohra@ericsson.com
Network Cloud Integration w/ Tungsten Fabric	**Deferred to 1/28**	
Edge Video Processing	Adnan Saleem- Radisys	adnan.saleem@radisys.com
ELIOT IoT Gateway	Khemendra Kumar- Huawei	khemendra.kumar@huawei.com
ELIOT SD-WAN/WAN Edge/uCPE	Khemendra Kumar- Huawei	khemendra.kumar@huawei.com
IEC Type 1 for Integrated Edge Cloud	Trevor Tao (ARM)	trevor.tao@arm.com
IEC Type 2 for Integrated Edge Cloud	Xinhuli (VMWare)	lxinhui@vmware.com
Industrial Edge	**Deferred to 1/28**	
Provider Access Edge	**Deferred to 1/28**	
Micro MEC	Tapio Tallgren- Nokia	tapio.tallgren@nokia.com
Radio Edge Cloud	Paul Carver- AT&T	pcarver@att.com
Starling X Far Edge Distributed Cloud	Bill Zvonar- Wind River	bill.zvonar@windriver.com
Time Critical Edge Compute	Shane Dewing- Intel	shane.dewing@intel.com

Feature Project	Winner of PTL Elected	Email Address
Portal	Mike Hunter – AT&T	mh2094@att.com
Cluster Health & Overload Monitoring Platform (CHOMP)	Mike Hunter – AT&T	mh2094@att.com
Support of OVS-DPK in Airship	Georg Kunz – Ericsson	georg.kunz@ericsson.com

Elections from 1/28:
- Network Cloud Integration w/ Tungsten Fabric
- Industrial Edge
- Provider Access Edge
- Validation

PTL Responsibilities

› The PTL acts as the de facto spokesperson for the project (feature projects and integration projects).

› Actions needed by PTLs:

   › Schedule and conduct kickoff meetings by 2/5

   › Establish regular project team meetings to target April release

   › Setup Jenkins repository for code

› Key Meetings to Join:

Meeting	Cadence
TSC Weekly Call	Tuesdays & Thursdays
TSC Technical Community Call	Thursdays
Sub-Committee Calls	Various

Key Documentation

Document	Description
Akraino Technical Charter	Defines the responsibilities and procedures for technical contribution to and oversight of the Akraino Edge Stack Project
Technical Community Document	Technical Community Document is intended to provide additional operational guidelines for the Project, and is subject to the Technical Charter.

*Note: Changes in documentation can occur as a result of TSC reviews/approvals

Sub Committee Information

Sub Committee	Chair	Description
CI, Blueprint Validation Lab, documentation	Cesar Behro	Further establish process, document and CI process to allow developers to get started
Community	Tapio Tallgren	Calls cover Akraino technical topics such as future plans, discussions about collaboration with other communities, and updates from Akraino projects.
Documentation	Sujata Tibrewala	To promote a consistent documentation across both the Akraino level and all individual Blueprint Families through both the Akraino Wiki and specific whitepapers.
API	TBD	Develop a API plan for the Akraino [collaboration + development].

*Note: The TSC meeting on 2/5 will include PTLs again and feature an overview of the process sub-committee from Jim Einarsson and the documentation sub-committee – Sujata Tibrewala

Sub Committee Information

Sub Committee	Chair	Description
API	TBD	Develop a API plan for the Akraino [collaboration + development].
CI and Blueprint Validation Lab, documentation Sub-committee	Cesar Berho	Further establish process, document and CI process to allow developers to get started
Community	Tapio Tallgren	Organizes weekly community calls, takes feedback from community and reports to TSC.
Documentation	Sujata Tibrewala	To promote a consistent documentation across both the Akraino level and all individual Blueprint Families through both the Akraino Wiki and specific whitepapers.

*Note: PTLs to be invited to TSC Call on 2/7 for introduction to Sub Committee Chairs
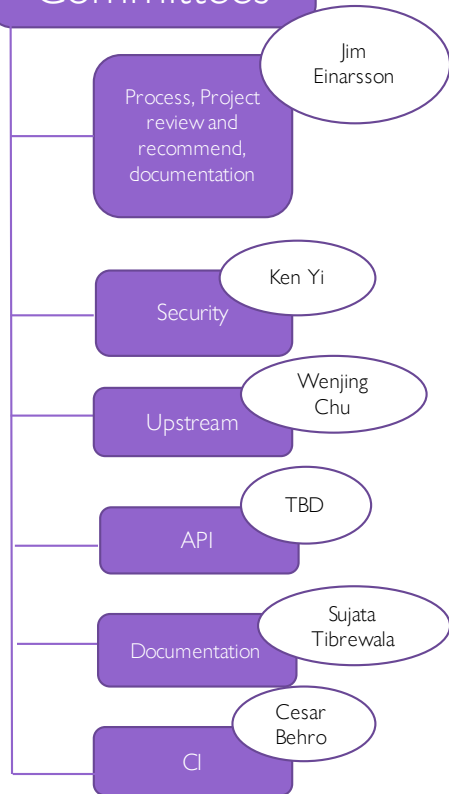
Sub Committee Information

Sub Committee	Chair	Description
Process, Project review and recommend, documentation	Jim Einarsson	Develop and evolve the process by which blueprint and feature project proposals are reviewed, and ultimately approved with recommendation required documentation
Security	Ken Yi	Develop and describe the security architecture, functional security requirements, and implementation recommendations for Akraino. Security encompasses both platform and network security.
Upstream	Wenjing Chu	Collection of coordinators liaison to upstream and provide a single point of liaison and reporting/requests towards TSC.

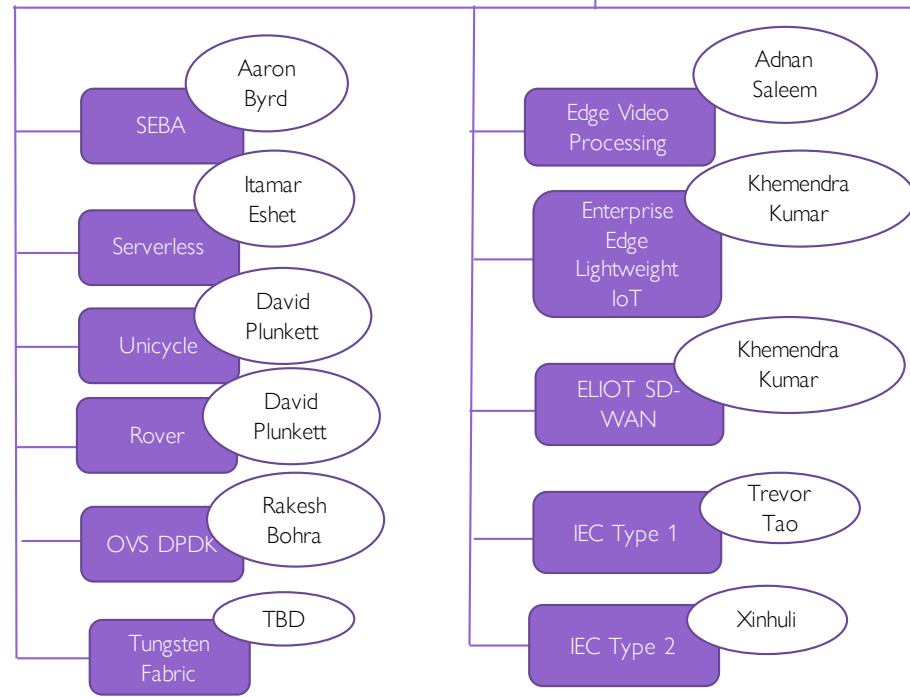*Note: PTLs to be invited to TSC Call on 2/7 for introduction to Sub Committee Chairs

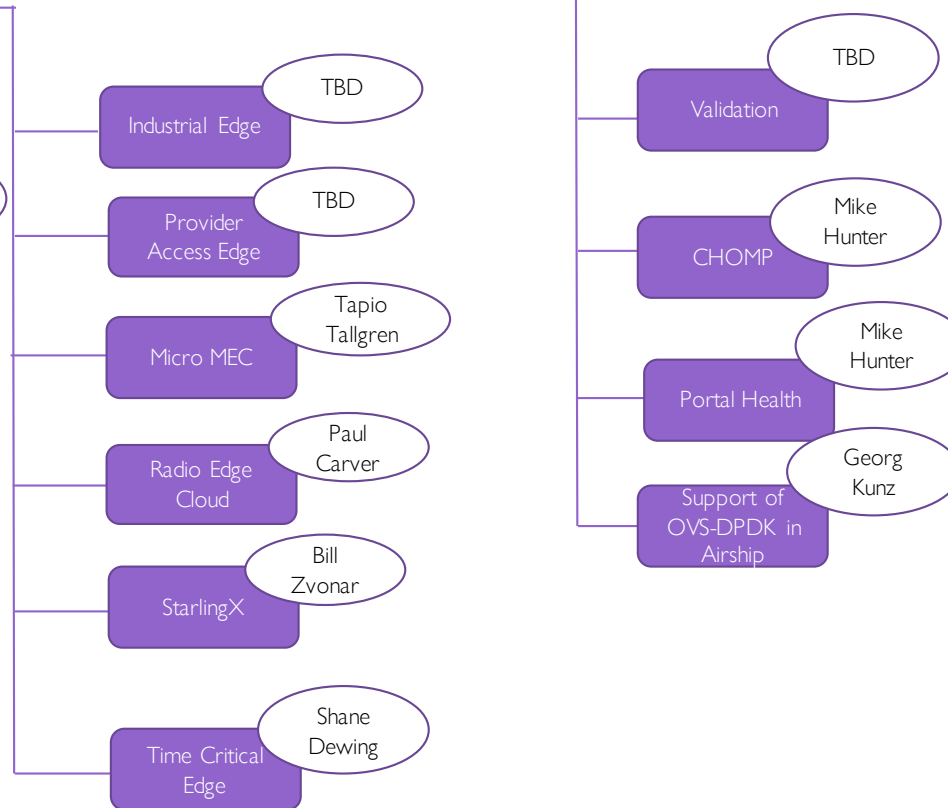# List of Blueprints (Incubation Approved)

Technical Steering Committee (TSC)

Sub Committees

- Process, Project review and recommend, documentation — Jim Einarsson
- Security — Ken Yi
- Upstream — Wenjing Chu
- API — TBD
- Documentation — Sujata Tibrewala
- CI — Cesar Behro

Blueprints

- SEBA — Aaron Byrd
- Serverless — Itamar Eshet
- Unicycle — David Plunkett
- Rover — David Plunkett
- OVS DPDK — Rakesh Bohra
- Tungsten Fabric — TBD
- Edge Video Processing — Adnan Saleem
- Enterprise Edge Lightweight IoT — Khemendra Kumar
- ELIOT SD-WAN — Khemendra Kumar
- IEC Type 1 — Trevor Tao
- IEC Type 2 — Xinhuli
- Industrial Edge — TBD
- Provider Access Edge — TBD
- Micro MEC — Tapio Tallgren
- Radio Edge Cloud — Paul Carver
- StarlingX — Bill Zvonar
- Time Critical Edge — Shane Dewing

Feature Projects

- Validation — TBD
- CHOMP — Mike Hunter
- Portal Health — Mike Hunter
- Support of OVS-DPDK in Airship — Georg Kunz

AKRAINO EDGE STACK
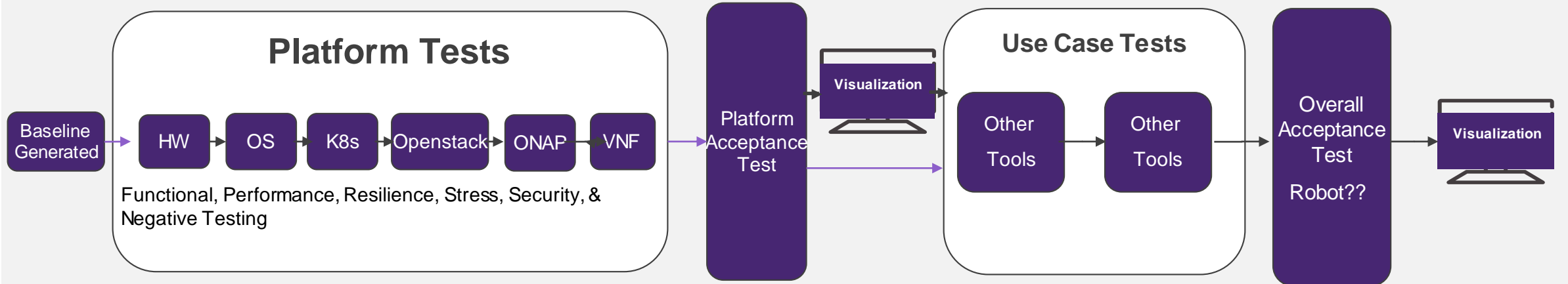
Akraino Blueprint Validation Framework: Background

› The purpose of the Validation framework is to propose a feature project that would define a standard set of tools and tests to evaluate Akraino Blueprints to determine if the blueprints are 'Akraino ready / validated'

› As part of this, we will need a consistent test framework that is automated and will work across all layers of the Akraino Edge Stack and across different BPs/BP Families. Layers of the Akraino Edge Stack include:

  › Hardware
  › OS
  › K8s
  › Kubeless
  › Openstack
  › ONAP
  › VNF (commercially non sensitive opensource VNFs)
  › Application Layer (commercially non sensitive opensource applications)

› Blueprint owners will be expected to conduct the standard tests with the specified tools. Project technical leads will share the results of the platform testing with the Akraino TSC so that the TSC can decide if the blueprint can proceed from the Incubation to Mature or Core stages

  › The proposed test set would be required to be passed for all in-scope layers of Akraino BPs
    › If a BP deploys K8s, the BP must pass the K8s test set using the defined test tool (e.g kubemonkey)
  › The validation framework defines a consistent minimum mandatory test set and test tools
    › Provides consistent and directly comparable results at each layer of BPs

› The validation framework can be extended to meet the requirements of Blueprints that are not included in the standard tests
  › It is incumbent on the blueprint owners to submit additional requirements to extend the validation framework in the feature project

› Leverage pre-existing open source tool sets as much as possible

AKRAINO EDGE STACK
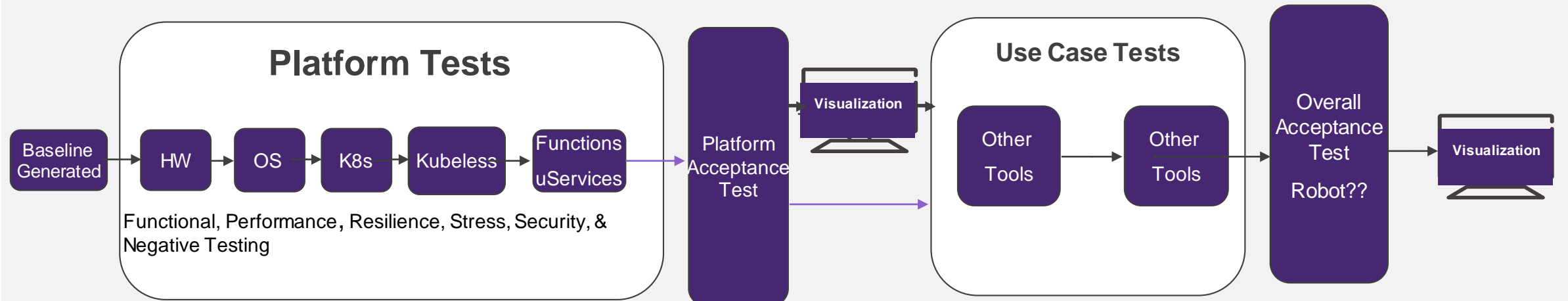
Akraino Validation Framework Feature Project Goals

› Akraino shall identify and utilize open source test frameworks for all BP layers

› Akraino shall extend and contribute to the frameworks to satisfy the blueprint testing needs

› Akraino shall automate testing as quality gates in CI/CD* BP verification pipelines

› Akraino shall employ security scanning to detect potential copyright violations

› Akraino shall develop security compliant testing and certification mechanisms (i.e. least privileges)

› Akraino testing shall ensure that each execution cleans up temporary resources that were required for testing

› Akraino tests shall assert meaningful results that can be easily accessed for visualization and comparison

› Akraino shall provide toolsets and documentation for the development of tests

› Akraino shall encourage re-use of test cases across blueprints for each layer of the Akraino Stack
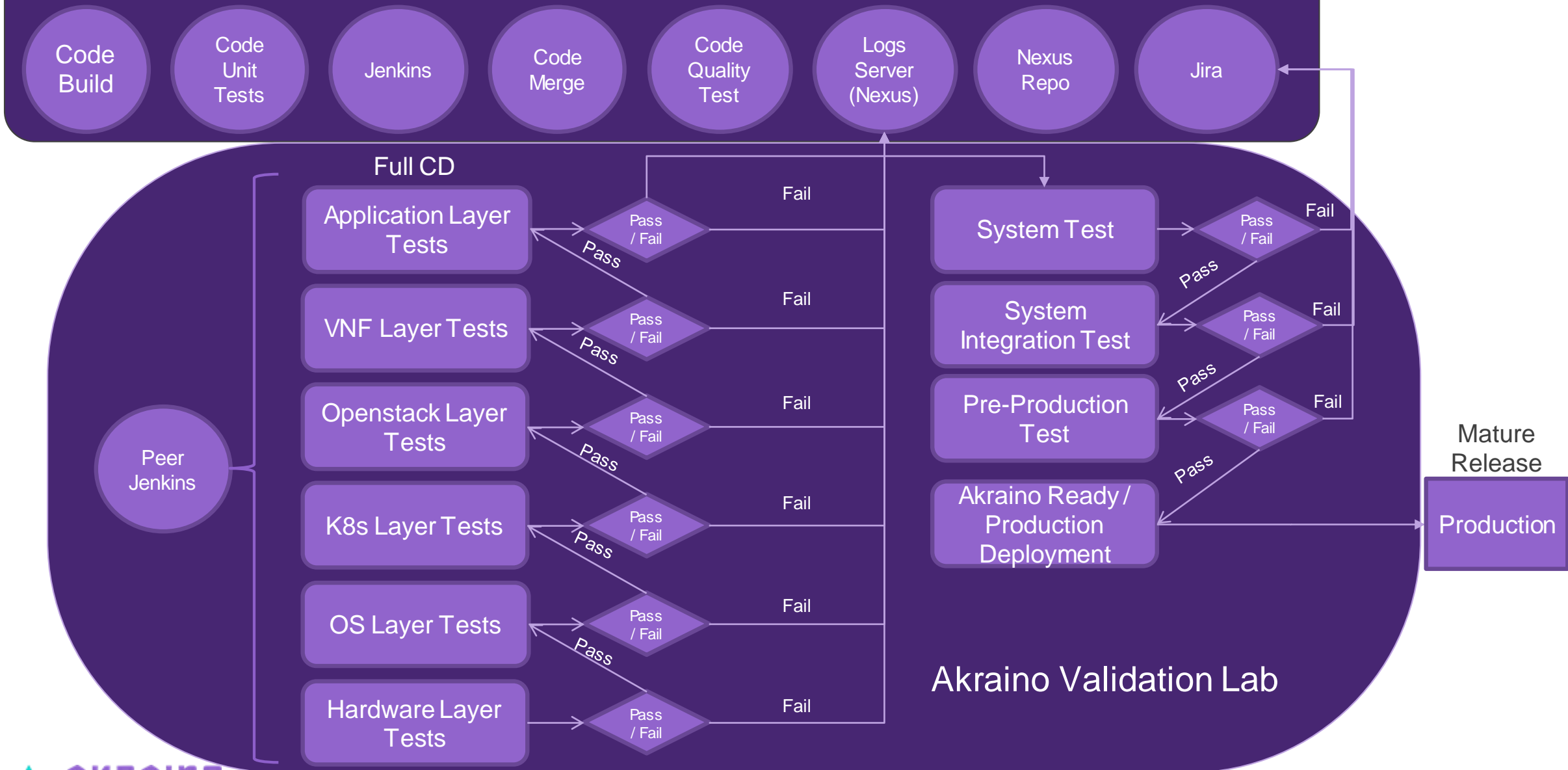
Example Akraino Validation Pipelines

Blueprint 1 Pipeline

Platform Tests

Baseline Generated → HW → OS → K8s → Openstack → ONAP → VNF

Functional, Performance, Resilience, Stress, Security, & Negative Testing

Platform Acceptance Test

Visualization

Use Case Tests

Other Tools → Other Tools

Overall Acceptance Test Robot??

Visualization

Blueprint 2 Pipeline

Platform Tests

Baseline Generated → HW → OS → K8s → Kubeless → Functions uServices

Functional, Performance, Resilience, Stress, Security, & Negative Testing

Platform Acceptance Test

Visualization

Use Case Tests

Other Tools → Other Tools

Overall Acceptance Test Robot??

Visualization

AKRAINO EDGE STACK

Akraino Edge Stack Validation Test Suite & Toolset

Layers	Proposed Tests	Proposed Tools	Proposed Pass / Fail Criteria
Hardware OS K8s Kubeless OpenStack VNF Application	Functional Testing Performance Testing Resiliency Testing Stress Testing Security Testing Negative Testing	TBD	Pass TBD% of tests specified for each layer Pass unit test, system test, integration test, VNF and application test gates applicable for the blueprint

Blueprint Process

1. Project owner submits the template for the Process sub-committee review
2. Process sub-committee comes up with a recommendation to the TSC
3. TSC review and approves the project (features or blueprint)
4. PTL nominated and elected
5. PTL reach out to CI/Validation sub-committee ci@lists.akraino.org to create project under Gerrit and in other LF infra. Including VMs or community lab share they may need
6. CI/Validation sub-committee approves the request and assist PTL to access the resources.
7. PTL coordinates with the project team to start the development

AKRAINO
EDGE STACK

Basic CI Services

› All CI services are run/maintained by the Linux Foundation

› Basic layout/operation of services is similar to other LF-run sites (e.g. *onap.org, acumos.org,* etc.)

› Gerrit: for source code (*gerrit.akraino.org*)

› Jenkins: for code builds and unit tests (*jenkins.akraino.org*)

› Nexus: for repositories (*nexus.akraino.org*)

› SonarQube: for automated code inspection (*sonar.akraino.org*)

› Wiki: for documentation (*wiki.akraino.org*)

› Jira: for issues (*jira.akraino.org*)

› All services are *https* or *ssh* (Gerrit)

› A Linux Foundation ID is required for all services

New Projects

New Projects

› PTL reach out to CI/Validation sub-committee to create project under Gerrit and in other LF infra. Including VMs or community lab share they may need

› CI/Validation sub-committee approves the request and assist PTL to access the resources.

› PTL coordinates with the project team to start the development.

› Projects have a list of committers who can put code into the projects

› Recommend projects be named as such: <blueprint-name>_<subproject-name>

› New Projects: CI committee sends an email to helpdesk@akraino.org with name of project (lower case) and list of committers

› Documentation in wiki generally refers to the latest master branch code (may be overridden by documentation subcommittee)

CI Process - Gerrit

› Code submitted to Gerrit from the developer

› Code is reviewed and voted on via the normal Gerrit code review process

› Reviewers need to have permissions (assigned by LF) to merge code

› LF uses a voting range of -2..+2; +2 code is eligible to merge

› LF guidelines on Gerrit use:

https://docs.releng.linuxfoundation.org/en/latest/gerrit.html

› Code Review process described here:

https://lf-releng-docs.readthedocs.io/en/latest/environment-overview.html

CI Process - Jenkins

› Jenkins jobs are primarily defined by LF code

› All Jenkins jobs are maintained in the *ci-management* project

› Jobs are written as JJB (*Jenkins Job Builder*) templates, on a per-project basis

› LF personnel code review/approve all changes to the *ci-management* project

› More details here:

  › https://docs.releng.linuxfoundation.org/en/latest/

  › https://github.com/lfit/releng-global-jjb

  › https://github.com/lfit/releng-common-packer.git

› Jenkins Sandbox (*https://jenkins.akraino.org/sandbox/*) available to do limited testing of jobs (without Nexus)

CI Process - Nexus

› All built artifacts are stored in Nexus
  › *nexus.akraino.org* - .tar, .war, and.jar files
  › *nexus3.akraino.org* - Docker containers
› CD retrieves from nexus as the result of Gerrit API events
› nexus3 can be used to retrieve Docker containers (supports Docker v2 API) – don't need DockerHub

CI Process – Wiki/Sonar/Jira

› Wiki is the documentation of record for Akraino

› SonarQube - used to check code quality of projects so configured in Jenkins