Akraino Edge Stack Technical Community Document - Framework

TSC Review – 09.20.18

Thanks to all community & TSC members provided inputs and feedback

This slide is the consolidation of all such inputs.

Draft – Yet to be baselined by TSC

Scope of this slide/discussion

- Intent of this slides is not to repeat or cover the content of "TSC charter document".
- The content articulated on this slide is focused on the additional details that TSC need to baseline as "TSC community document"
- This presentation is to kick start the discussion, followed by content documented in the wiki.
- TSC appreciates the feedback shared by the community and this presentation incorporated such feedbacks.

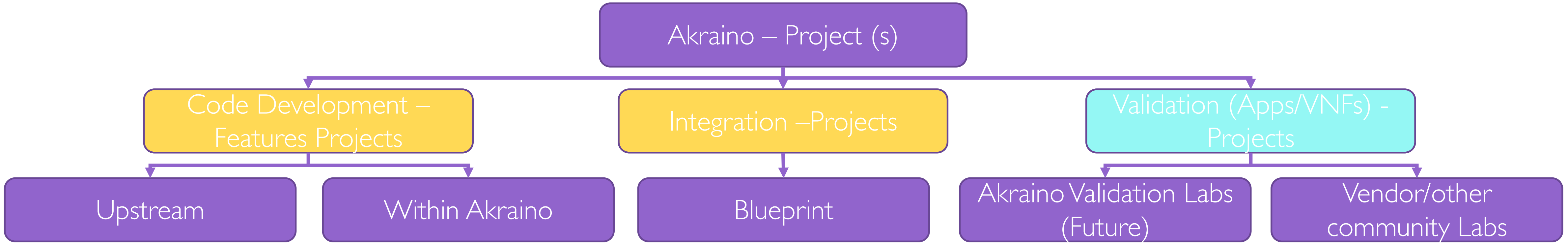
Technical mission

1. Create end to end configuration for a particular Edge Use case which is complete, tested and production deployable { Blueprints}.
2. Develop projects to support such end to end configuration. Leverage upstream community work as much as possible to avoid duplication. { Projects}
3. Work with broader edge communities to standardize edge apis { Socialization, so this community tools can interoperate}
4. Encourage Vendors and other communities to validate VNFs & edge application on top of Akraino blueprints { Facilitate a eco-system}

Akraino Project summary

1. For Simplicity call everything as a project
2. Support three types of projects – Feature projects, Integration, validation
3. Feature Projects
    1. Primary goal is to liaison with upstream project to fill in gaps in the upstream code needed by the edge blueprint (s)
    2. Or develop projects with in community which are not supported in the upstream
    3. Do not fork upstream projects [upstream first]
    4. Project focus area for this community – Common user experience across blueprints, Edge Testing, Integration/Ops/security tools
    5. Primary upstream community – based on what is used within the blueprints.
    6. Example - https://gerrit.akraino.org/r/#/admin/projects/regional_controller
4. Integration projects
    1. Blueprints are integration projects which integrates multiple components for a edge Point-of-Delivery (POD)
    2. Blueprints define the fundamental characteristics / components of any Point-of-Delivery (POD) instantiation
    3. Blueprints should be complete, tested and production deployable
    4. Maintain the Continuous integration at the Akraino Community
    5. Leverage Vendor & Community labs to demonstrate the Continuous deployment and feed back the results to the community to ensure working of "a blueprint"
5. Vendor & Community labs
    5. Akraino community to establish guidelines to connect with Akraino CI and CD feedback to LF.

Akraino Project Types & Scope


Akraino – Project (s)
Code Development – Features Projects
Integration –Projects
Validation (Apps/VNFs) - Projects
Upstream
Within Akraino
Blueprint
Akraino Validation Labs (Future)
Vendor/other community Labs

Infrastructure s/w

- ONAP
- OpenStack
- K8
- Docker
- OS

Integration tools

- Airship
- Starlingx
- Etc.,

Collaborate standards

- APIs (EdgeX Foundry,…)

Common user experience

- Akraino Portal + workflows

Edge Testing

- Blueprint ETE Testing suites
- CI/CD scripts

Integration & ops & Security tools

- xxxx

Seed

- Network cloud – Telco use case – OpenStack/ONAP/ K8/Docker/Airship/ OS agnostic based

Pipeline

- ONF SEBA
- Real-time RAN
- IOT
- Etc.,

- Network Cloud – AT&T
- ONF SEBA
- XXXX

Legend

- 🟨 Continuous Integration (CI)
- 🟦 Continuous Deployment (CD)

AKRAINO EDGE STACK

How to arrive at the blueprints? – 5 step process

Sequence	What	Definition	Action
1	Edge Use Case	Description of the business outcome Defines workload characteristics, design constraints, Cost range, etc.	Community member to use "template" and submit for TSC review
2	Edge use Case Specification	Specifications (HW/SW components, deployment configurations, etc.) designed support Use Case(s) and described in a testable, implementation-agnostic manner ("what", not "how").	Community member to use "template" and submit for TSC review
3	Blueprint	• Reference Architecture to meet the use case need • Implementation-specific declarative configuration file(s) ready to be consumed by that implementation's deployment and LCM tool(s) and resulting in a stack that passes the design's tests.	• Developed and maintained within the Akraino Community (CI) • Project team maintained
4	Validation	• Tested without VNF/Edge Apps – prove it works • Tested with VNF/Edge Apps – Prove ETE works	• Akraino community process • Tested in Vendors, Providers, Community labs • Results published under the blueprint
5	User Deployment	• Production deployment by users/providers/vendors	• Provide feedback to the community (bug and enhancement reports)

AKRAINO EDGE STACK

Thanks to Frank for providing inputs

Akraino Use Cases and Use Case Specifications

Akraino Use Cases Templates

› Business driven

Use Case Characteristics	*Network Cloud Use Case Examples*
Business Need	Network based edge cloud that can be deployed at provider data center and telco offices
User Experience	Single Pane of Glass control - Administrative and User Based GUIs Zero touch provisioning to reduce ops cost
Cost Of Solution	Less 800K a POD [46 servers deployment] – Cruzer POD configuration
Scale	Minimum 10 – Maximum 1000 Locations
Applications	Any type of Edge Virtual Network Functions
Power restrictions	Less than 50K watts

Akraino Use Case Specifications

Specifications (HW/SW components, deployment configurations, etc.) designed support Use Case(s) and described in a testable, implementation-agnostic manner

Blueprint Components

Use Case Specifications	*vEPC service on Network Cloud Specification Examples*
Workload	vEPC or any Edge VNFs
Infrastructure orchestration	OpenStack/ONAP
UCP tool	Airship
Workload Characteristics	VMs and Containers
Under cloud	K8 & Docker
SDN	SR-IOV & OVS-DPDK
OS	Linux (Ubuntu)
Hardware	X86 based G10 and above servers.

Sample templates – not a final version

Existing vs. new blueprints

- Categorize blueprints by Family { e.g., Network Cloud}
- "A" blueprint can support multiple POD types { e.g., Cruzer (6 racks) , Unicycle (1 rack), Rover (single server) }
- "A" Pod could support multiple "configuration types" but within the criteria defined reference architecture for that blueprint { e.g., different Linux OS}
- "A" configuration type is a defined by declarative file { e.g., YAML for the POD type}
- Each committer/project submitter should look at existing blueprint and see if it can support their use case by existing configuration or with new configuration type
- If existing blueprint does not support the use case or with new configuration type then to submit a new blueprint proposal to TSC
- TSC to review the blueprint proposal and approve/disapprove
- Intention is to maximize the "configuration types" supported by a blueprint and minimize the number of blueprint. Discretion applied during review process.

AKRAINO EDGE STACK

Relationship Between Blueprint Specs & PODs

Blueprint Specifications define the declarative configuration for each deployment model or Point of Delivery (POD) of a Blueprint.

- YAML files allow for different configurations within the same blueprint

Blueprint Level	*POD Specification Level*	*Component Level*	*Declarative Configuration Level - YAML File*
Family: Network Cloud	Unicycle	Ubuntu/OS/ODL based	{yaml files U1}
		Centos/OS/ODL based	{yaml files U2}
		Ubuntu/OS/Neutron based	{yaml files U3}
	Tricycle	X	{yaml files T1}
		Y	{yaml files T2}
		Z	{yaml files T3}
	Rover	A	{yaml files R1}

Choices shown is just for illustration and not recommendations

- Point of Delivery (POD) - The method in which a blueprint is deployed to an edge site.
- PODs organize edge devices for deployment and enable a cookie-cutter approach to large scale deployments (e.g., 10,000 plus locations) at a reduced cost.
  - For example, an edge location could have a single server or multiple servers in one or more racks.

EDGE STACK

Thanks to Andrew for providing inputs

Analogy from Andrew’s slide

- Family: Canidae
  - Genus: Canis (wolf like genus)
    - Species: Familiaris (domesticated dog)
    - Species: Lupus (grey wolf)
    - Species: Rufus (red wolf)
  - Genus: Vulpes velox (fox like genus)
    - Species: Vulpus (red fox)
    - Species: Lagopus (arctic fox)
    - Species: Zerda (fenec fox)
  - Genus: Speothos velox (South American genus)
    - Species: Venaticus (Bush dog)
- Family: Felidae ……

Blueprint Components vs. specifications

- Blueprint Specifications (a.k.a Declarative configurations) are built from the component options for the layers contained within a Blueprint.
- Blueprint Specifications can evolve in subsequent releases to add / remove functional layers
- <u>Declarative configuration naturally allows ways to support different components within a same blueprint</u>

Blueprint Specifications

Blueprint Components

UCP tool	Airship	Airship	Starlingx	Starlingx
SDN	SR-IOV/OVS	ODL Boron	TitaniumFabric R1	TitaniumFabric R2
Overcloud	OS Ocata	OS Pike	k8s	
Undercloud CNI	Calico	Multus	Flannel	
Undercloud	K8s 1.9	K8s 1.12		
Host OS layer	Ubuntu 14.04	Ubuntu 16.04	Centos 6	Centos 7
HW layer	Dell R720/ HP DL360	HP DL360		

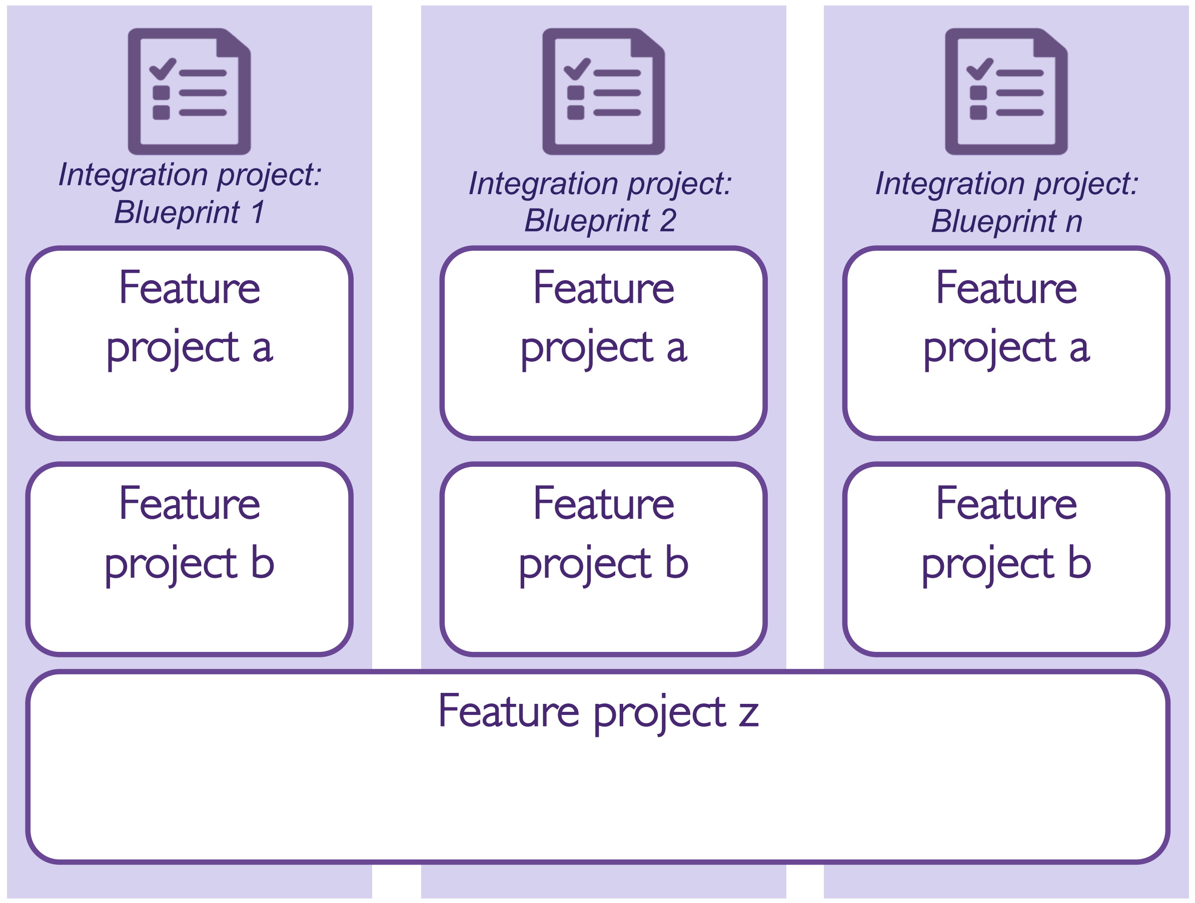
Selections show one possible specification within this blueprint

This is for illustration and doesn't contain all layers required for the NC blueprint

AKRAINO EDGE STACK

Thanks to Andrew for providing inputs

“Feature projects” relationship to “Integration projects”



- Feature project could be specific to a blueprint or across the blueprint
- Integration project = a blueprint
- A Feature Project is a long term endeavor setup to deliver features across multiple releases, which have a shorter lifespan
- A Integration project is a long term endeavor setup to deliver ETE functionalities across multiple releases
- Integration project requires at least one continuous deployment lab supported by vendor or a community. Without such CD lab, blueprint working cannot be validated.

AKRAINO EDGE STACK

Project lifecycle – States and Reviews

- To move from one state to the next state, the Project Team must obtain TSC approval based on a set of evaluation criteria.
- Project teams request TSC reviews to move up the ladder. TSC majority approval is required to advance from one state to the next
- Same process for Feature and Integration projects

Proposal

- Project doesn't exist yet
- May not have real resources
- Proposal to be create project due to business needs.

Incubation

- Project has resources
- Project is in the early stages of development
- The outcome is a minimum viable product (MVP) that demonstrates the value of the project and is used to collect feedback
- Not expected to be used in production environments.

Mature

- Project is fully functioning and stable
- Project has achieved successful releases

Core

- Project provides value to and receives interest from a broad audience.

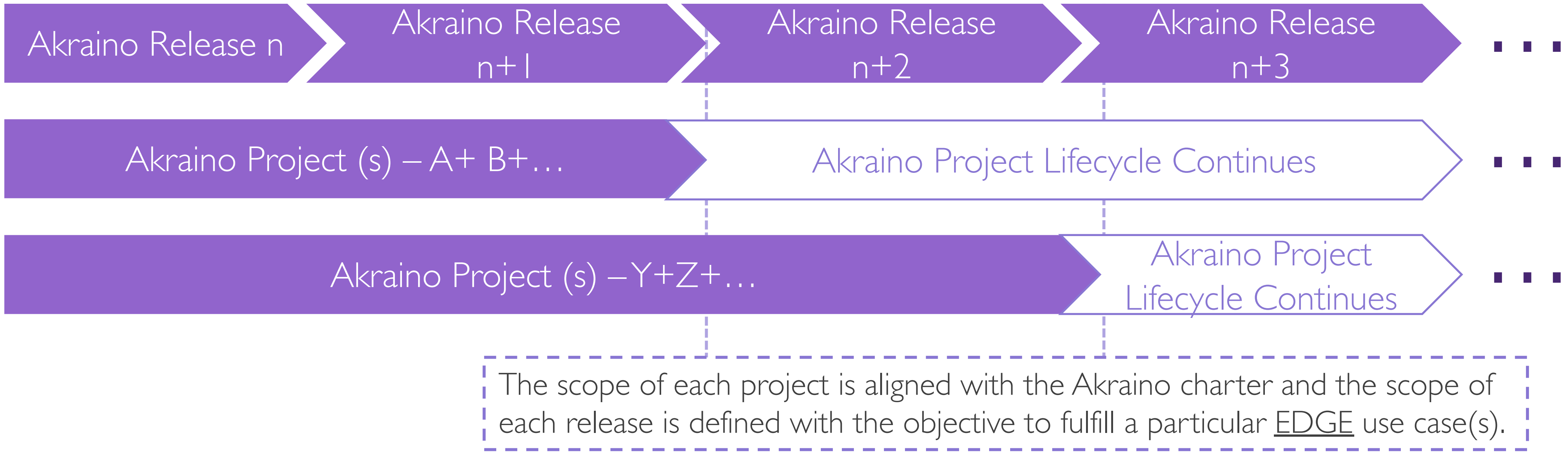
Archived

- Project can reach Archived state for multiple reasons
- Project has successfully completed and artifacts provide business values, or project has been cancelled for unforeseen reasons
- Project in any state can be Archived through a Termination Review.

AKRAINO EDGE STACK

Release plan

- Akraino releases will include a set of project deliverables.
- Akraino releases can be composed of 1 to N projects.
- Akraino projects are long term endeavors setup to deliver features across multiple releases, which have a shorter lifespan.


Akraino Release n
Akraino Release n+1
Akraino Release n+2
Akraino Release n+3
. . .
Akraino Project (s) – A+ B+…
Akraino Project Lifecycle Continues
. . .
Akraino Project (s) – Y+Z+…
Akraino Project Lifecycle Continues
. . .

The scope of each project is aligned with the Akraino charter and the scope of each release is defined with the objective to fulfill a particular EDGE use case(s).

AKRAINO EDGE STACK

TSC Subcommittees

- The TSC may establish subcommittees to assist the TSC with its responsibilities and provide expert guidance in technical subject areas
  - Subcommittees are advisory in nature, and not authoritative. They provide advice to projects and to the TSC.
  - Subcommittee Members - Each subcommittee shall determine its own membership eligibility, in consultation with the TSC
    - Subcommittee Chair - Each subcommittee may elect a Chair and optionally a Vice-Chair who is responsible for leading meetings and representing the subcommittee to the TSC

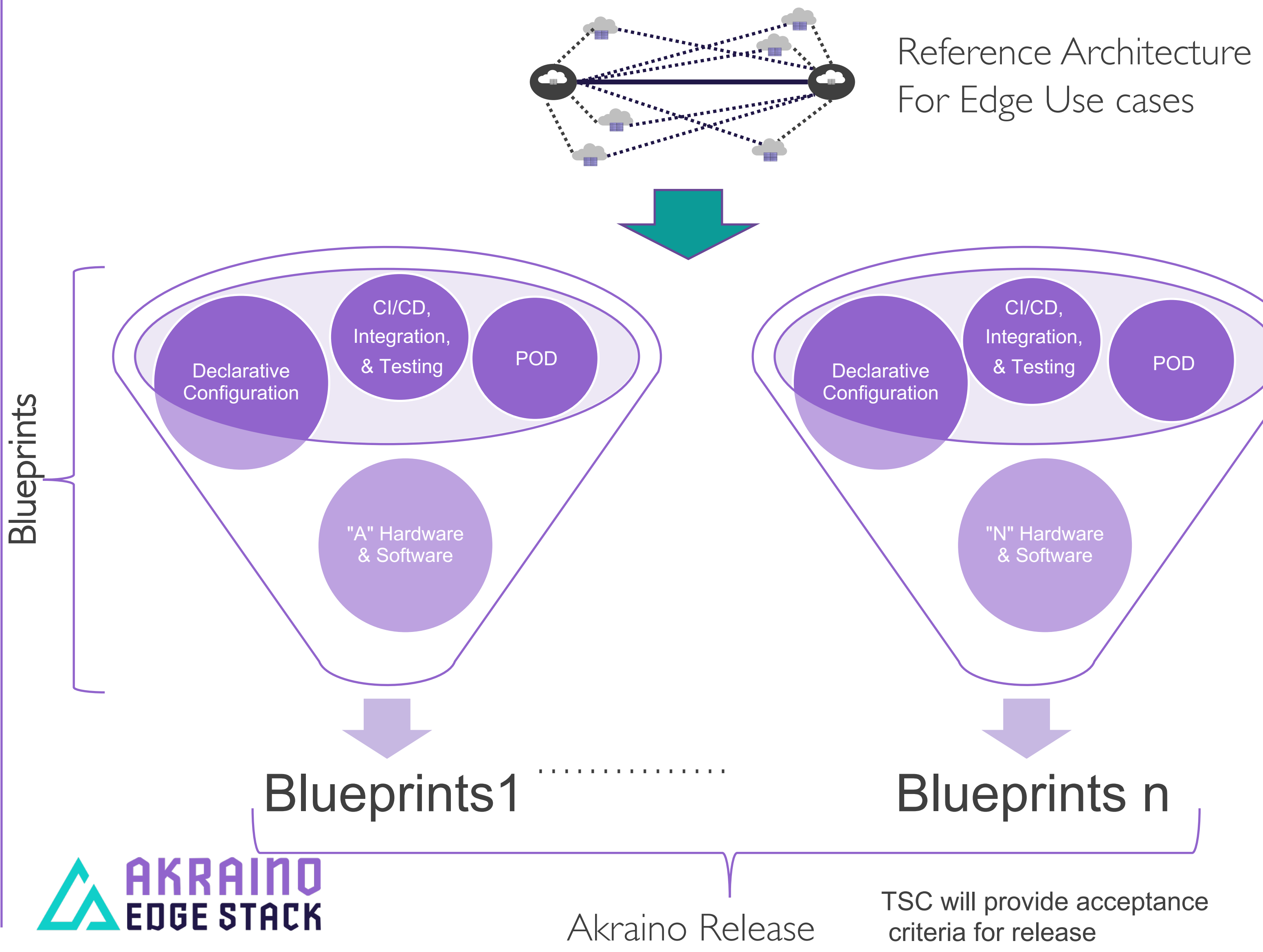
AKRAINO EDGE STACK

Next Steps

- Baseline this deck to agree on the terminology for the community
- Review this slide with the community (September 20th) to get community feedback
- Work on the content of Akraino Technical community document – Early draft available in the google drive
- Target to baseline the document by end of September or early October.

Additional backup slides

What is Akraino Blueprint?



TSC will provide acceptance criteria for release

Blueprints – **Approved and tested declarative configuration based on use cases, set of Hardware & Software, Point of delivery (POD).**

Reference Architecture – Defines Akraino building blocks

Declarative Configuration – Hides lower layer complexity to user

CI/CD, Integration & Testing Tools – Drive product quality

Akraino release – End Product

Why Akraino Blueprint?

Akraino Blueprints

- Use Case Based
- Fully Integrated End to End Solution (CI / CD)
- Proven and Tested by Community
- Life Cycle Support
- Production Quality

Benefits:

- Low Cost
- Large Scale
- Zero Touch Provisioning
- Industry Adoption

AKRAINO EDGE STACK

Blueprints with clear business need

Blueprints	Network Cloud	Real Time	Customer-premises / Far Edge
	Single to Multiple racks	Single – Dual Server/White boxes	All-in-one White boxes
Use Cases	Telecom (5G,…)	Access (RAN), PON, IOT.,	Universal CPE

AKRAINO EDGE STACK

Network Cloud Blueprint (Seed Code)

AT&T Network Cloud Blueprint

airship

Use Case Based

- Telco / 5G / Enterprise Use Cases

Fully Integrated ETE Solution (CI / CD)

- Airship based
- Upstream Integrated
- Full CI in LF
- Automated CD Validation Using Real Hardware

Proven and Tested by Community

- Community Developed and Maintained

Life Cycle Support

- Continuous Integration
- Documentation

Production Quality

- Production deployed at AT&T

AKRAINO EDGE STACK

Akraino Network Cloud Blueprint (Aug 2018)

CICD
(Community)

Akraino GUI	Dashboard	Admin GUI	User GUI
Akraino Workflow	Platform Workflows	Camunda	
Edge Application and APIs	APIs	Edge APIs	Edge Cloud(s) Integration APIs
	Applications & VNFs	Sample Edge App (CDN)	
Edge Application and Orchestration	Lightweight Edge App Orchestration	Community - TBD	
NFV Orchestration	NFV & Domain Specific Orchestrator	ONAP Amsterdam	
Edge Platform Software Components	Infra Orchestration	OpenStack (Ocata)	Kubernetes
	Storage	SDS(Ceph)	
	Network Control Plane	Calico	
	Network Data Plane	SR-IOV	OVS
	Operation System	Ubuntu	
Network Edge	NC – Multinode Cluster	Single Server (Rover)	
Network Edge Micro Services	Serverless		
Customer Edge		Satellite	Rover (whitebox)

Akraino Upper Cloud Lifecycle Tools

AirShip Under Cloud Lifecycle

D a w n *

*To be contributed to Airship

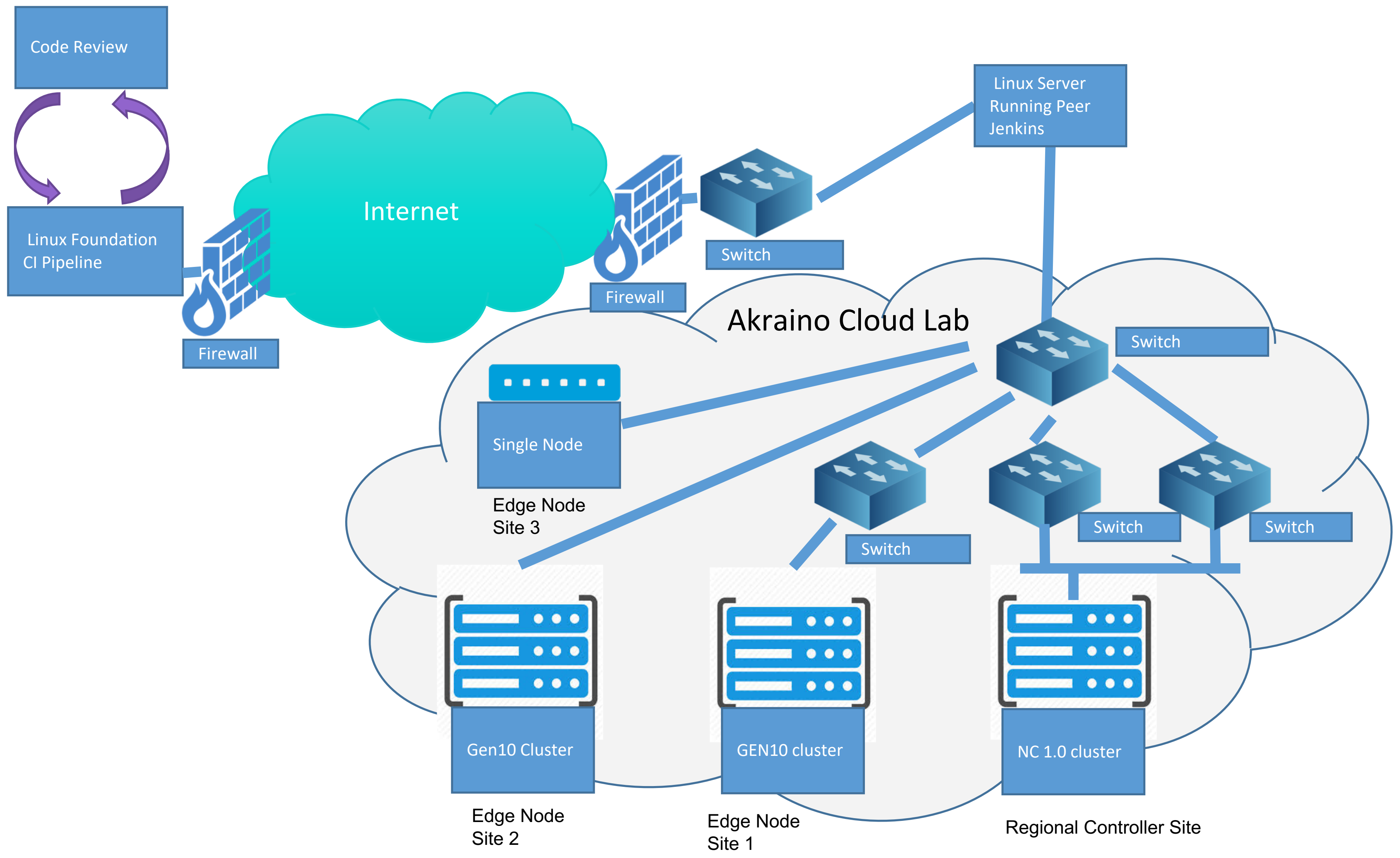
Akraino Chest

- Declarative Configuration
- AI Tools box
- ETE Operations tools
- ETE Security tools
- Narad (Inventory)
- ETE Testing OpenStack Tempest
- PINC (N/W Orchestration)
- Documentation

AKRAINO EDGE STACK

🟩 Akraino - new | Upstream | Future release

Network Cloud - CD Integration Akraino Lab



AKRAINO EDGE STACK

For More Information, Please Visit www.akraino.org

AKRAINO EDGE STACK

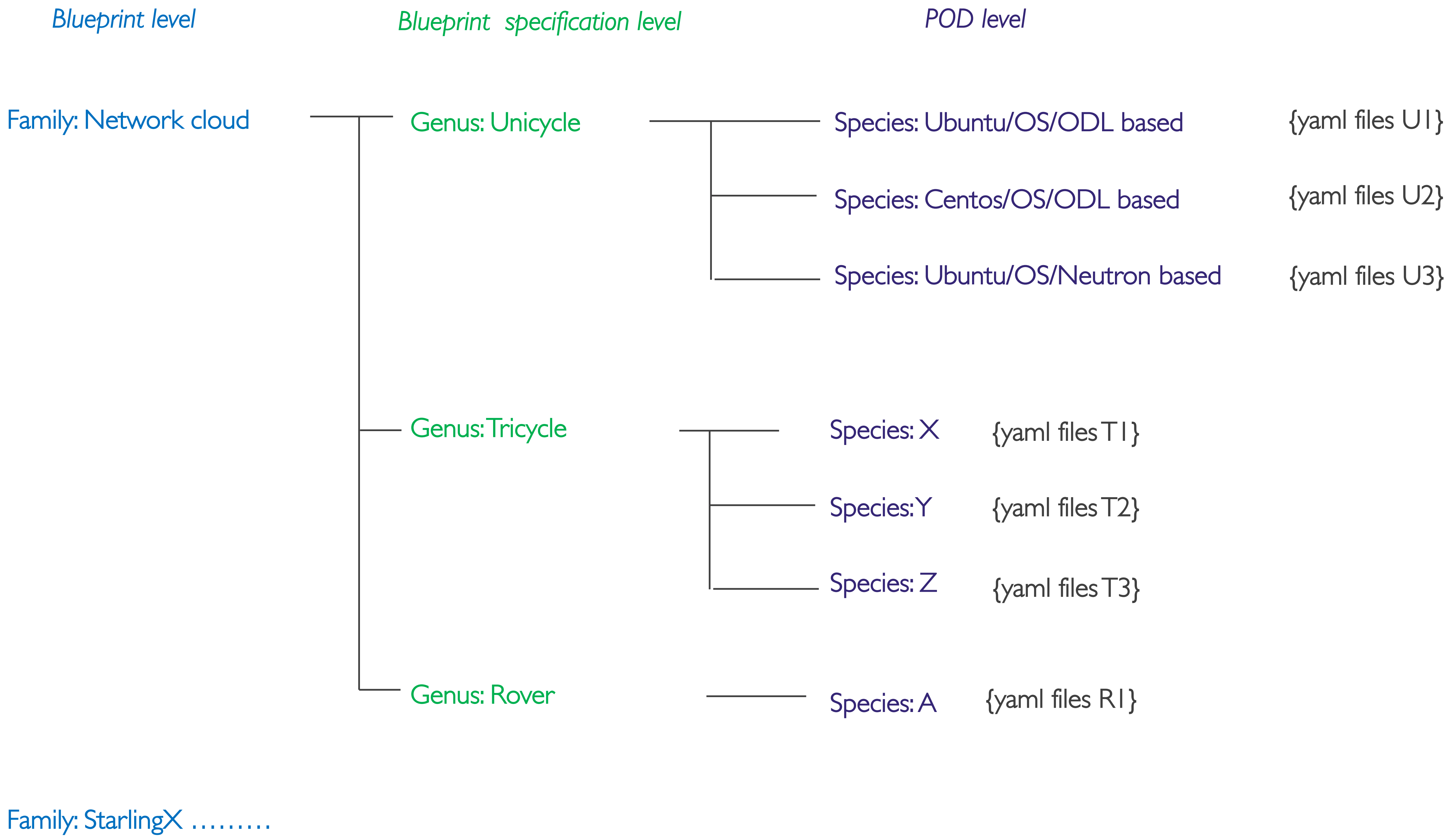
Proposals from Community members – incorporated in the above deck.
Backup materials

AKRAINO EDGE STACK

Akraino Blueprints and Blueprint Specification/Templates

A framework proposal V4.0

- Family: Canidae
  - Genus: Canis (wolf like genus)
    - Species: Familiaris (domesticated dog)
    - Species: Lupus (grey wolf)
    - Species: Rufus (red wolf)
  - Genus: Vulpes velox (fox like genus)
    - Species: Vulpus (red fox)
    - Species: Lagopus (arctic fox)
    - Species: Zerda (fenec fox)
  - Genus: Speothos velox (South American genus)
    - Species: Venaticus (Bush dog)
- Family: Felidae ……



Choices shown are just for illustration and not recommendations

Blueprints and Blueprint Specification/Release Templates

At the highest level the *Blueprint* defines the fundamental must have characteristics/components of any POD deployed using it

- e.g. A "Network Cloud" Blueprint deploys OpenStack using a k8s undercloud with Airship based LCM (etc)
- These are immutable attributes - if they are omitted or replaced a different *Blueprint* results
- Can be considered an Akraino POD's *Family*

Within a given blueprint a POD's deployed components can be tailored by different *Blueprint Specifications*

- e.g. At each Akraino release of the Network Cloud blueprint its *Blueprint Specification Template* would contain the set of all verified possible plugins/options for each layer
- Can be considered an Akraino POD's *Genus*

The exact POD configuration of a given Blueprint Specification is the last level of description

- e.g. This is the contents of the yaml manifests for a Network Cloud blueprint's POD
- Can be considered the final definitive definition of deployment. An Akraino POD's *Species*

Validation of hosted applications (e.g. VNFs) against a *Blueprint* **and** its *Specification* is then possible

Network Cloud Blueprint Specification Template Release 1

SDN	None (neutron)	ODL Boron	TitaniumFabric R1	TitaniumFabric R2
Overcloud	OS Ocata	OS Pike	k8s	
Undercloud CNI	Calico	Multus	Flannel	
Undercloud	K8s 1.9	K8s 1.12		
Host OS layer	Ubuntu 14.04	Ubuntu 16.04	Centos 6	Centos 7
HW layer	Dell R720	HP DL360		

This is for illustration and doesn't contain all layers required for the NC blueprint

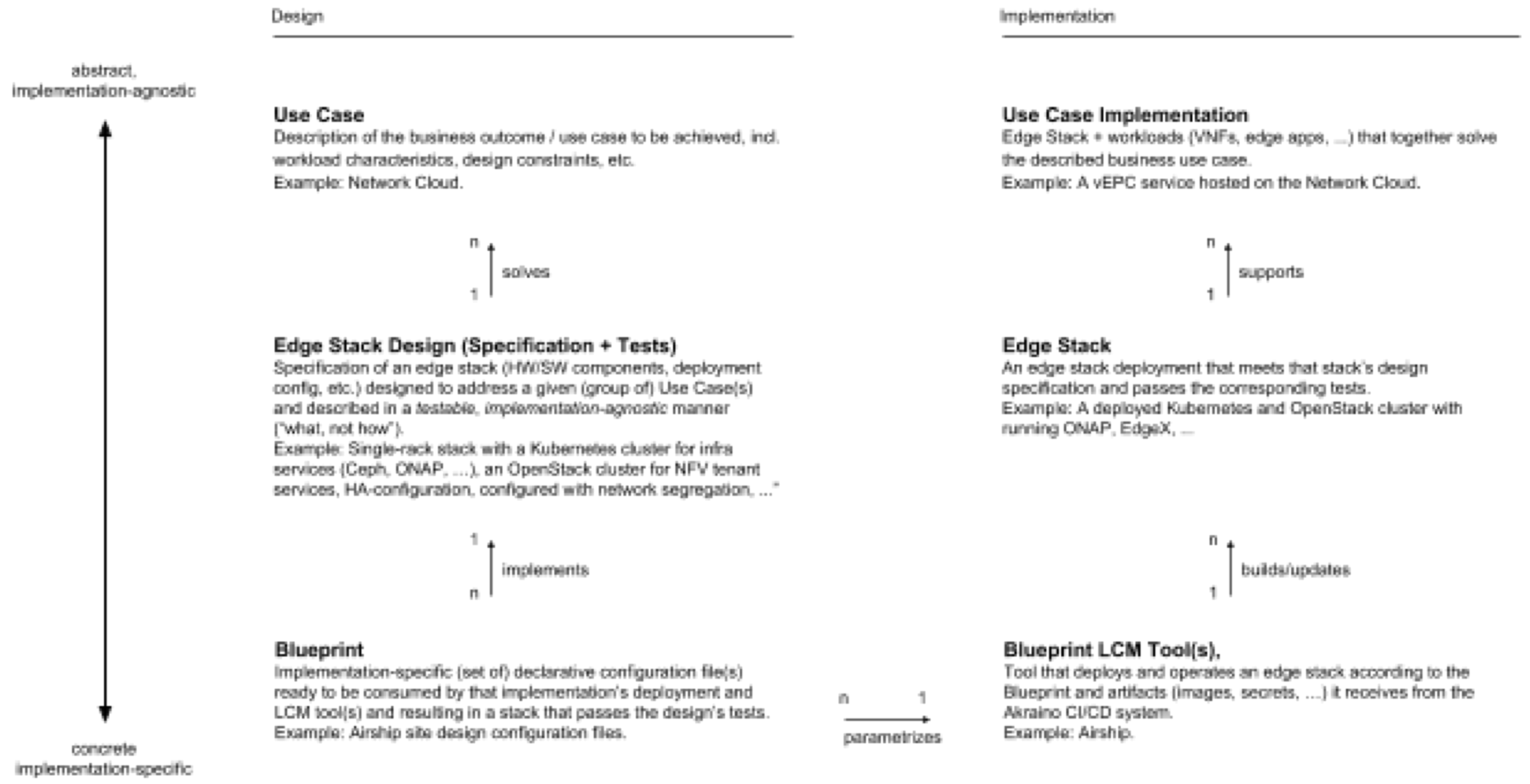
Red box selections show one possible specification within this blueprint

Network Cloud Blueprint and Specification/Release Templates

Different *Blueprints* would have different options to select in the *Blueprint Specification* as the functionality deployed in such a POD would be different

e.g. an IOT blueprint may not use OpenStack as a virtualization

The *Specification Template* of a given *Blueprint* can evolve in subsequent releases to add / remove functional layers


Design

Implementation

abstract,
implementation-agnostic

concrete
implementation-specific

Use Case
Description of the business outcome / use case to be achieved, incl. workload characteristics, design constraints, etc.
Example: Network Cloud.

n ↑ solves 1

Edge Stack Design (Specification + Tests)
Specification of an edge stack (HW/SW components, deployment config, etc.) designed to address a given (group of) Use Case(s) and described in a *testable, implementation-agnostic* manner ("what, not how").
Example: Single-rack stack with a Kubernetes cluster for infra services (Ceph, ONAP, ...), an OpenStack cluster for NFV tenant services, HA-configuration, configured with network segregation, ..."

1 ↑ implements n

Blueprint
Implementation-specific (set of) declarative configuration file(s) ready to be consumed by that implementation's deployment and LCM tool(s) and resulting in a stack that passes the design's tests.
Example: Airship site design configuration files.

n → 1 parametrizes

Use Case Implementation
Edge Stack + workloads (VNFs, edge apps, ...) that together solve the described business use case.
Example: A vEPC service hosted on the Network Cloud.

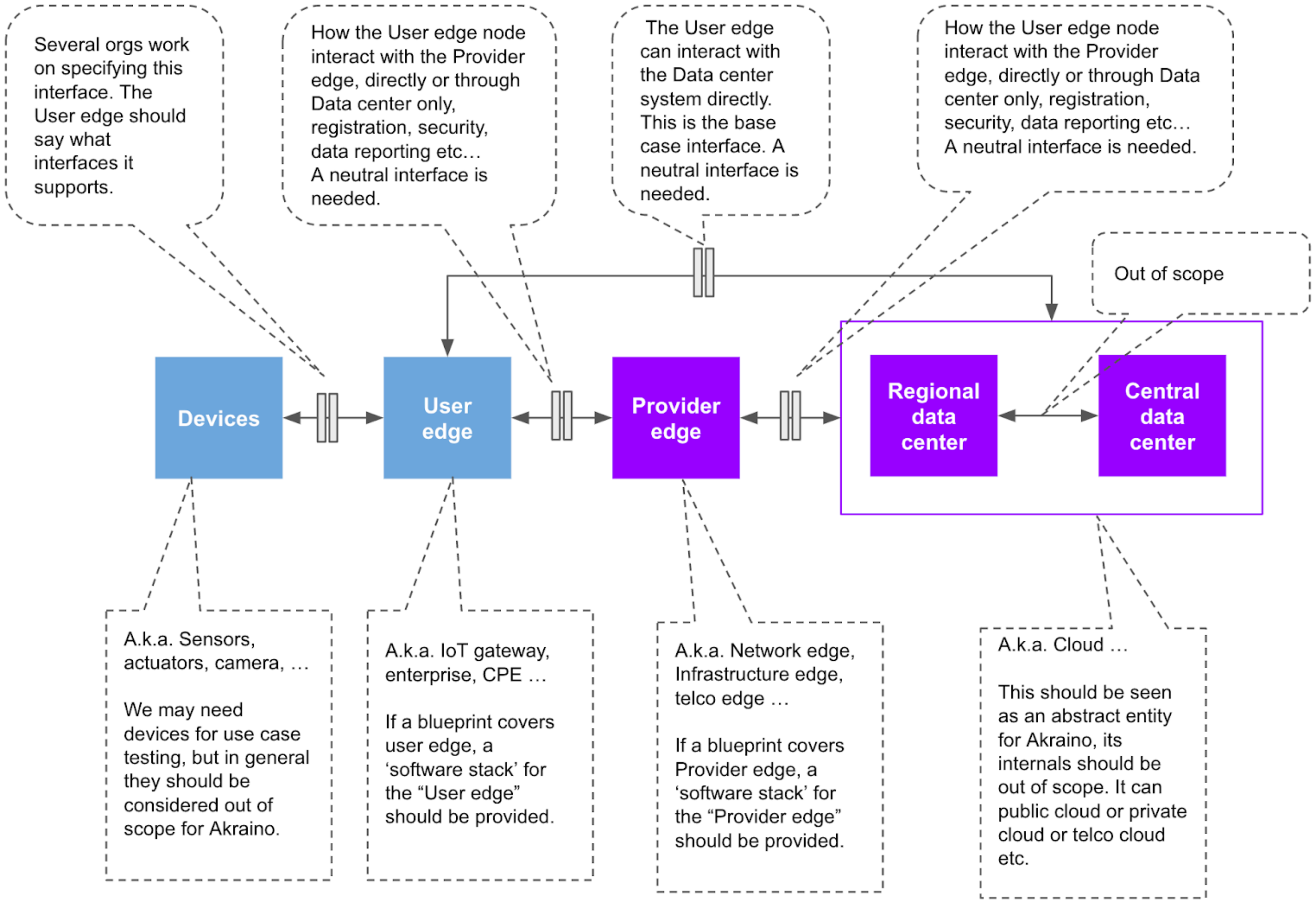
n ↑ supports 1

Edge Stack
An edge stack deployment that meets that stack's design specification and passes the corresponding tests.
Example: A deployed Kubernetes and OpenStack cluster with running ONAP, EdgeX, ...

n ↑ builds/updates 1

Blueprint LCM Tool(s),
Tool that deploys and operates an edge stack according to the Blueprint and artifacts (images, secrets, ...) it receives from the Akraino CI/CD system.
Example: Airship.

Author – Frank Zdarsky


Several orgs work on specifying this interface. The User edge should say what interfaces it supports.
How the User edge node interact with the Provider edge, directly or through Data center only, registration, security, data reporting etc… A neutral interface is needed.
The User edge can interact with the Data center system directly. This is the base case interface. A neutral interface is needed.
How the User edge node interact with the Provider edge, directly or through Data center only, registration, security, data reporting etc… A neutral interface is needed.
Out of scope
Devices
User edge
Provider edge
Regional data center
Central data center
A.k.a. Sensors, actuators, camera, … We may need devices for use case testing, but in general they should be considered out of scope for Akraino.
A.k.a. IoT gateway, enterprise, CPE … If a blueprint covers user edge, a 'software stack' for the "User edge" should be provided.
A.k.a. Network edge, Infrastructure edge, telco edge … If a blueprint covers Provider edge, a 'software stack' for the "Provider edge" should be provided.
A.k.a. Cloud … This should be seen as an abstract entity for Akraino, its internals should be out of scope. It can public cloud or private cloud or telco cloud etc.

Author –
Wenjing Chu