Akraino TSC F2F Meeting Material
Network Cloud Blueprint & Project Proposals

Kandan Kathirvel
December 6-7, 2018
Agenda

- Network Cloud Blueprint Proposals
  - Unicycle
  - Rover
  - SEBA
  - Serverless

- Akraino Edge Stack Feature Project
  - CHOMP
AT&T initiated Blueprints (Akraino seed code)

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Telecom (5G-Core, Voice, …)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blueprint Family</td>
<td>Network Cloud Blueprint</td>
</tr>
<tr>
<td>Blueprint</td>
<td>Rover (Single Server)</td>
</tr>
<tr>
<td></td>
<td>Unicycle (Multi-server)</td>
</tr>
</tbody>
</table>

12/6/18
Other AT&T’s Blueprint Proposals

**Blueprints**
- SEBA (In collaboration with ONF)

**Use Cases**
- vAccess (XGSPON)
- Microservices
- Realtime RAN, vRAN

**RI Targets**
- Single Rack (NC Family)
- (NC Family)
- Multiple Servers (All Container workloads)

**3rd Party Cloud**
- 3rd Party Cloud Edge Stack – Standard Interfaces

**Customer-premises / Far Edge**
- All-in-one White boxes
- Universal CPE

12/6/18
# Akraino Blueprint Proposal - Unicycle

<table>
<thead>
<tr>
<th>Case Attributes</th>
<th>Description</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Seed Code (Code already in Akraino)</td>
<td></td>
</tr>
<tr>
<td>Blueprint Family - Proposed Name</td>
<td>Network Cloud Family</td>
<td></td>
</tr>
<tr>
<td>Use Case</td>
<td>Network Cloud – carrier edge use case</td>
<td></td>
</tr>
<tr>
<td>Blueprint proposed Name</td>
<td>Unicycle (Single Rack)</td>
<td></td>
</tr>
<tr>
<td>Initial POD Cost (capex)</td>
<td>Unicycle less than $250K</td>
<td></td>
</tr>
<tr>
<td>Scale &amp; Type</td>
<td>Up to 7 servers X86 server</td>
<td></td>
</tr>
<tr>
<td>Applications</td>
<td>5G Core or vRAN (RIC)</td>
<td></td>
</tr>
<tr>
<td>Power Restrictions</td>
<td>Example Only:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Less than 10Kw</td>
<td></td>
</tr>
<tr>
<td>Infrastructure orchestration</td>
<td>OpenStack Pike or above - VM orchestration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Docker 1.13.1 or above / K8 1.10.2 or above- Container</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Orchestration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OS - Ubuntu 16.x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VNF Orchestration - ONAP Beijing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Under Cloud Orchestration - Airship v1.0</td>
<td></td>
</tr>
<tr>
<td>SDN</td>
<td>SR-IOV &amp; OVS-DPDK</td>
<td></td>
</tr>
<tr>
<td>Workload Type</td>
<td>VMs and Containers</td>
<td></td>
</tr>
<tr>
<td>Additional Details</td>
<td>See next slide</td>
<td></td>
</tr>
</tbody>
</table>
Akraino Blueprint Proposal - Unicycle

Hosted @ Telco or Provider (e.g., Network Cloud)

Unicycle POD

- C-Agg
- C-Agg
- M-SW
- Rack POD
- Containerized Control plane
- K8 based resiliency
- Possible - Data plane/Control Plane mixed

Use Cases (e.g.)
- IoT, Wireline (PON), Store
- Remote Edge (Analytics etc.)
## Akraino Blueprint Proposal - Rover

<table>
<thead>
<tr>
<th>Case Attributes</th>
<th>Description</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Seed Code (Code already in Akraino)</td>
<td></td>
</tr>
<tr>
<td><strong>Blueprint Family - Proposed Name</strong></td>
<td>Network Cloud Family</td>
<td></td>
</tr>
<tr>
<td><strong>Use Case</strong></td>
<td>Network Cloud</td>
<td></td>
</tr>
<tr>
<td><strong>Blueprint proposed Name</strong></td>
<td>Rover</td>
<td></td>
</tr>
<tr>
<td><strong>Initial POD Cost (capex)</strong></td>
<td>Rover less than $20k</td>
<td></td>
</tr>
<tr>
<td><strong>Scale &amp; Type</strong></td>
<td>1 server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x86/ARM server or deep edge class</td>
<td></td>
</tr>
<tr>
<td><strong>Applications</strong></td>
<td>5G micro edge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Over the top edge applications.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remote @ Central Offices</td>
<td></td>
</tr>
<tr>
<td><strong>Power Restrictions</strong></td>
<td>Less than 10Kw</td>
<td></td>
</tr>
<tr>
<td><strong>Infrastructure orchestration</strong></td>
<td>OpenStack Pike or above - VM orchestration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Docker 1.13.1 or above / K8 1.10.2 or above- Container Orchestration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OS - Ubuntu 16.x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Under Cloud Orchestration - Airship – in a Bottle</td>
<td></td>
</tr>
<tr>
<td><strong>SDN</strong></td>
<td>SR-IOV &amp; OVS-DPDK or VPP-DPDK</td>
<td></td>
</tr>
<tr>
<td><strong>Workload Type</strong></td>
<td>VMs and Containers</td>
<td></td>
</tr>
<tr>
<td><strong>Additional Details</strong></td>
<td>See next slide</td>
<td></td>
</tr>
</tbody>
</table>
Akraino Blueprint Proposal - Rover

Characteristics
- Telco Remote sites
- Single server deployment (Uniserver)
- Containerized Control plane
- Possible - Data plane/Control Plane mixed

Use Cases
- 5G micro edge
- Over the top edge applications.
## Assessment Criteria – Data Points

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Network Cloud Blueprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each initial blueprint is encouraged to take on at least two Committers from different companies</td>
<td>AT&amp;T, Ericsson, ARM, Intel, Juniper, Radisys, Dell, HP</td>
</tr>
<tr>
<td>Complete all templates outlined in this document</td>
<td>Detailed in this slide</td>
</tr>
<tr>
<td>A lab with exact configuration required by the blueprint to connect with Akraino CI and demonstrate CD. User should demonstrate either an existing lab or the funding and commitment to build the needed configuration.</td>
<td>Validation Lab hosted by AT&amp;T <a href="https://wiki.akraino.org/pages/viewpage.action?pageId=1147237">https://wiki.akraino.org/pages/viewpage.action?pageId=1147237</a></td>
</tr>
<tr>
<td>Blueprint is aligned with the Akraino Edge Stack Charter</td>
<td>All Opensource, Edge use case, Aligned with the Akraino Charter</td>
</tr>
<tr>
<td>Blueprint code that will be developed and used with Akraino repository should use only Open Source software components either from upstream or Akraino projects.</td>
<td>Yes, all Opensource</td>
</tr>
<tr>
<td>For new blueprints submission, the submitter should review existing blueprints and ensure it is not a duplicate blueprint and explain how the submission differs. The functional fit of an existing blueprint for a use case does not prevent an additional blueprint being submitted.</td>
<td>Multiple blueprints under this NC family to support Telco use cases</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>NC Blueprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the project is appropriate (no trademark issues etc.); Proposed repository name is all lower-case without any special characters</td>
<td>NC Unicycle and Rover</td>
</tr>
<tr>
<td>Project contact name, company and email are defined and documented</td>
<td>Kandan Kathirvel, AT&amp;T <a href="mailto:kk0563@att.com">kk0563@att.com</a></td>
</tr>
<tr>
<td>Description of the project goal and its purpose are defined</td>
<td>Multiple blueprints under this NC family to support Telco use cases</td>
</tr>
<tr>
<td>Scope and project plan are well defined</td>
<td>Targeted for release 1</td>
</tr>
<tr>
<td>Resources committed and available</td>
<td>There is a team, resources and lab in place</td>
</tr>
<tr>
<td>Contributors identified</td>
<td>AT&amp;T, Ericsson, ARM, Intel, Juniper, Radisys, Dell, HP</td>
</tr>
<tr>
<td>Initial list of committers identified (elected/proposed by initial contributors)</td>
<td>AT&amp;T, Dell, HP, Ericsson, ARM, Juniper</td>
</tr>
<tr>
<td>Meets Akraino TSC Policies</td>
<td>The Project will operate in a transparent, open, collaborative, and ethical manner at all times</td>
</tr>
<tr>
<td>Proposal has been socialized with potentially interested or affected projects and/or parties</td>
<td>Proposal has been reviewed by the TSC</td>
</tr>
<tr>
<td>Cross Project Dependencies (XPDs). In the case where a project will require changes in other projects, those projects are listed in the proposal, and a sponsoring developer in the project has been identified</td>
<td>Airship, OpenStack, K8, Docker, OS</td>
</tr>
<tr>
<td>Tools have been identified and discussed with relevant partners (Linux Foundation, IT). Once the project passes the review, the tools chain must be created within one week. Tools encompass Configuration Management, CI/CD, Code Review, Testing, Team Wiki, End Users documentation (not exhaustive)</td>
<td>Initial Code already exist in the Akraino CI</td>
</tr>
</tbody>
</table>
SEBA POD Overview

- Deployment model is self-contained, pre-integrated solution
  - Scale to 1000s of central office locations
  - Lowest Cost Solution Required

- Current SEBA POD contains network elements, compute nodes, and software components
  - Aggregation and management switches
  - Three compute nodes required for K8 redundancy
  - About twenty containers running VOLTHA, ONOS, NEM, etc
  - Supports up to 16 OLT
  - All container based
SEBA Logical View

Fault mgmt
Config
Accounting
Performance
Security
Inventory

BNG as a workload option
- External physical BNG
- vBNG in compute
- BNG as PNF in Agg switches
- BNG as PNF in OLT boxes

Subscriber traffic ‘fast-path’ to Internet. Redirected to compute for additional services.

Network Edge Mediator (NEM)

All control software deployed as Docker containers on compute nodes using Kubernetes

Network Edge Mediator (NEM)

ONAP
OSAM-Central

Legacy OSS

ONAP
OSAM-Central

OSAM-Local

EMS/NMS-adaptor

ONOS Cluster

vOLT
dhcp
mcast
SR
FPM
T3

OF

OF

OF

K8

K8

POD Hardware

SEBA EP Peripheral/PNF/Pod

REST
Kafka

Redfish

Redfish

Redfish

UNI

ONT

OLT

NNI

Fault mgmt
Config
Accounting
Performance
Security
Inventory

All control software deployed as Docker containers on compute nodes using Kubernetes

Subscriber traffic ‘fast-path’ to Internet. Redirected to compute for additional services.
SEBA – Key Software Components

- **VOLTHA**
  - The Virtual OLT Hardware Abstraction provides an abstraction for the PON by modeling it as an quasi-Ethernet switch with UNI and NNI ports, to the SDN controller, while hiding internal details about the OLT and ONUs.

- **NEM**
  - The Network Edge Mediator serves as the mediation layer between the edge/access system and the service provider backend and global automation frameworks.
  - XOS will continue to be a core component of NEM. NEM will have additional components to support FCAPS. A variety of operator OSS/BSS and global orchestration frameworks can be integrated northbound for specific deployment needs

- **New Container Orchestration System - Kubernetes**
  - Kubernetes is being used as the container orchestration system and through the use of Helm charts, all control components (VOLTHA, ONOS, NEM etc.) for the access system will be deployed on the compute nodes.

- **ONOS**
  - ONOS clusters will be used to host
    - control Apps providing VOLTHA functionality such as subscriber authentication, vlan assignment, DHCP, and multicast.
    - control Apps providing AGG switch/router functionality such as vlan-termination or forwarding, multicast, failover etc.
SEBA POD after Akraino Integration

- Monolithic design
- Manual steps involved
- Cloud layer not hardened for production

K8 based Edge Compute

- Akraino based community blueprint
- Full Automation (Airship based + Tenant Container support)
- Cloud layer hardened for production

Akraino Blueprint (NC new POD)
### Blueprint Proposal: SEBA

<table>
<thead>
<tr>
<th>Case Attributes</th>
<th>Description</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>New blueprint for fixed wireline access within Network Cloud Blueprint family</td>
<td></td>
</tr>
<tr>
<td><strong>Blueprint Family - Proposed Name</strong></td>
<td>Network Cloud</td>
<td></td>
</tr>
<tr>
<td><strong>Use Case</strong></td>
<td>Virtual broadband access (XGS-PON - Higher bandwidth, symmetric version of GPON)</td>
<td></td>
</tr>
<tr>
<td><strong>Blueprint proposed Name</strong></td>
<td>SDN Enabled Broadband Access (SEBA)</td>
<td></td>
</tr>
<tr>
<td><strong>Scale &amp; Type</strong></td>
<td>3 servers per POD x86 and ARM (with 8-16 cores each)</td>
<td></td>
</tr>
<tr>
<td><strong>Applications</strong></td>
<td>Virtual broadband access – vOLT access &amp; aggregation (5000 edge locations)</td>
<td></td>
</tr>
<tr>
<td><strong>Power Restrictions</strong></td>
<td>Less than 1 kW. NEBS and 48V DC</td>
<td></td>
</tr>
</tbody>
</table>

**Infrastructure orchestration**
- OS - Ubuntu 16.x
- Docker 1.13.1 or above / K8 1.10.2 or above - Container Orchestration
- Under Cloud Orchestration - Airship v1.0
- Open Network Operating System (ONOS) and XOS
- VOLTHA (Virtual Optical Line Terminal Hardware Abstraction – CORD project)
- Network Edge Mediator (NEM)
- ONAP and OSAM
- EMS/NMS Adaptor

**SDN**
- ONOS
- OF & Redfish

**Workload Type**
- Containers

**Additional Details**
- Akraino based community blueprint, Full Automation (Airship based + Tenant Container support)
- Cloud layer hardened for production
- Current SEBA POD contains network elements, compute nodes, and software components.
- Aggregation and management switches
- Three compute nodes required for K8 redundancy
- About twenty containers running ONOS, XOS, VOLTHA, NEM, etc
- Supports up to 16 OLTs.

See next slide for additional details.
Akraino SEBA Blueprint SW Stack

- **Akraino GUI**
  - Dashboard
  - Admin GUI
  - User GUI

- **Akraino Workflow**
  - Platform Workflows
  - Camunda
  - Docker Repo

- **Edge Application and APIs**
  - APIs
  - Applications & VNFs
  - EMS/NMS Adaptor
  - VOLTSA
  - Redfish

- **Edge Application and Orchestration**
  - Lightweight Edge App Orchestration
  - ONOS
  - XOS

- **NFV Orchestration**
  - NFV & Domain Specific Orchestrator
  - ONAP
  - OSAM

- **Edge Platform**
  - Software SKU
  - Infra Orchestration
  - Kubernetes
  - SDS(Ceph)
  - Calico
  - Ubuntu

- **Network Edge**
  - Multinode Cluster
  - Uniserver

- **Network Edge Micro Services**

- **Customer Edge**

**Tools**
- **AirShip Under Cloud Lifecycle Tools**
- **Akraino Upper Cloud Lifecycle Tools**

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

**Source:** AT&T

16
### Assessment Criteria – Data Points

<table>
<thead>
<tr>
<th>Criteria</th>
<th>SEBA Blueprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each initial blueprint is encouraged to take on at least two Committers from different companies</td>
<td>AT&amp;T, ONF, Radisys, Netsia, ARM</td>
</tr>
<tr>
<td>Complete all templates outlined in this document</td>
<td>Detailed in this slide</td>
</tr>
<tr>
<td>A lab with exact configuration required by the blueprint to connect with Akraino CI and demonstrate CD. User should demonstrate either an existing lab or the funding and commitment to build the needed configuration.</td>
<td>Validation Lab hosted by AT&amp;T <a href="https://wiki.akraino.org/pages/viewpage.action?pageId=1147237">https://wiki.akraino.org/pages/viewpage.action?pageId=1147237</a> Plus, Community lab, once it is ready.</td>
</tr>
<tr>
<td>Blueprint is aligned with the Akraino Edge Stack Charter</td>
<td>All Opensource, Edge use case, Aligned with the Akraino Charter</td>
</tr>
<tr>
<td>Blueprint code that will be developed and used with Akraino repository should use only Open Source software components either from upstream or Akraino projects.</td>
<td>Yes, all Opensource</td>
</tr>
<tr>
<td>For new blueprints submission, the submitter should review existing blueprints and ensure it is not a duplicate blueprint and explain how the submission differs. The functional fit of an existing blueprint for a use case does not prevent an additional blueprint being submitted.</td>
<td>This blueprint introduces fixed wireline access based on XGS-PON (higher bandwidth, symmetric version of GPON) within the Network Cloud Blueprint family.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>SEBA Blueprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the project is appropriate (no trademark issues etc.); Proposed repository name is all lower-case without any special characters</td>
<td>Updated NC Unicycle (3 servers)</td>
</tr>
<tr>
<td>Project contact name, company and email are defined and documented</td>
<td>Kandan Kathirvel, AT&amp;T <a href="mailto:kk0563@att.com">kk0563@att.com</a></td>
</tr>
<tr>
<td>Description of the project goal and its purpose are defined</td>
<td>Multiple blueprints under this NC family to support Telco use cases</td>
</tr>
<tr>
<td>Scope and project plan are well defined</td>
<td>Targeted for release 1</td>
</tr>
<tr>
<td>Resources committed and available</td>
<td>There is a team, resources and lab in place</td>
</tr>
<tr>
<td>Contributors identified</td>
<td>AT&amp;T, ONF, Radisys, Netsia, ARM</td>
</tr>
<tr>
<td>Initial list of committers identified (elected/proposed by initial contributors)</td>
<td>AT&amp;T, ONF, Radisys, Netsia, ARM</td>
</tr>
<tr>
<td>Meets Akraino TSC Policies</td>
<td>The Project will operate in a transparent, open, collaborative, and ethical manner at all times</td>
</tr>
<tr>
<td>Proposal has been socialized with potentially interested or affected projects and/or parties</td>
<td>Proposal has been reviewed by the TSC</td>
</tr>
<tr>
<td>Cross Project Dependencies (XPDs). In the case where a project will require changes in other projects, those projects are listed in the proposal, and a sponsoring developer in the project has been identified</td>
<td>Airship, K8, Docker, OS, additional tools to be identified</td>
</tr>
<tr>
<td>Tools have been identified and discussed with relevant partners (Linux Foundation, IT). Once the project passes the review, the tools chain must be created within one week. Tools encompass Configuration Management, CI/CD, Code Review, Testing, Team Wiki, End Users documentation (not exhaustive)</td>
<td>New validation project need to be created</td>
</tr>
</tbody>
</table>
# Blueprint Proposal: Serverless

<table>
<thead>
<tr>
<th>Case Attributes</th>
<th>Description</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>New Blueprint for Serverless capability at the Edge</td>
<td></td>
</tr>
<tr>
<td>Blueprint Family - Proposed Name</td>
<td>Network Cloud</td>
<td></td>
</tr>
<tr>
<td>Use Case</td>
<td>Provide FaaS (Function as a Service) for Serverless Applications</td>
<td></td>
</tr>
<tr>
<td>Blueprint proposed Name</td>
<td>Serverless</td>
<td></td>
</tr>
<tr>
<td>Initial POD Cost (capex)</td>
<td>Leverage Unicycle POD - less than $150k</td>
<td></td>
</tr>
<tr>
<td>Scale &amp; Type</td>
<td>Up to 7 servers, x86/ARM server or deep edge class</td>
<td></td>
</tr>
<tr>
<td>Applications</td>
<td>Ephemeral event driven transactions for IoT gateway. For example, applications for autonomous cars.</td>
<td></td>
</tr>
<tr>
<td>Power Restrictions</td>
<td>Less than 10Kw</td>
<td></td>
</tr>
<tr>
<td>Infrastructure orchestration</td>
<td>Kubeless, Docker 1.13.1 or above and K8s 1.10.2 or above- Container Orchestration, OS - Ubuntu 16.x, Under Cloud Orchestration - Airship v1.0</td>
<td></td>
</tr>
<tr>
<td>SDN</td>
<td>OVS</td>
<td></td>
</tr>
<tr>
<td>Workload Type</td>
<td>Containers</td>
<td></td>
</tr>
<tr>
<td>Additional Details</td>
<td>VerneMQ MQTT Gateway and/or Mosca MQTT Broker, Kafka message bus and Webhook/Nginx middleware, Kubeless function management engine over Kubernetes, Helm chart for platform infrastructure installation integrated with Airship, Regional controller based installer for Functions</td>
<td>See next slide for additional details</td>
</tr>
</tbody>
</table>
Motivation and Benefits for “Serverless” Edge

- Function-as-a-service (FAAS) or serverless architecture improves edge capacity utilization via automatic release of unused resources.
- Functions accelerate development and integration cycles by high-level abstraction of application runtime.
- Edge FaaS users do not have to deal with availability, capacity scaling and planning across large number of edge nodes, removing 3rd party adoption barriers for edge cloud.
- Complementing Akraino with AWS Lambda-like and open-source AWS Greengrass equivalent, with focus on IoT enablement.
# What Workloads Are Great Match for Serverless

<table>
<thead>
<tr>
<th>Workload Characteristics</th>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Transactional – “do and die”</td>
<td>Always on</td>
</tr>
<tr>
<td></td>
<td>Stateless</td>
<td>Stateful</td>
</tr>
<tr>
<td></td>
<td>Working with external data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Event-driven</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workload Examples</th>
<th>Good</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoT back-end</td>
<td>IoT gateway</td>
</tr>
<tr>
<td></td>
<td>Video transcoding</td>
<td>Data plane element (firewall, router, switch)</td>
</tr>
<tr>
<td></td>
<td>Web application services</td>
<td>Database</td>
</tr>
<tr>
<td></td>
<td>Control plane (DNS, AAA, mobile packet core)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In-memory caching (memcached, live streaming)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stream and file processing</td>
<td></td>
</tr>
</tbody>
</table>
Serverless Software Architecture - Autonomous Vehicle Use Case

Network Edge
- End point
- MQTT Gateway: Mosca, VerneMQ
- Middleware: Kafka, NGINX / Webhook
- Function Engine: Kubeless
  - Function invocation
  - Event published
  - Function telemetry collection and visualisation
  - MQTT route and congestion predictions published

End point coordinates and route published over MQTT
Route guidance subscribe over MQTT

Monitoring
- Prometheus
- Grafana

Kubernetes
- Function deployment
- Auto-scaling configuration
- Function invoked
- Endpoint coordinates and route info stored

Regional Datacenter
- Orchestrator
- Persistence: Mongo
- Visualization
  - Endpoints visualized on Google Maps
  - Area-based end point data retrieved

Containers deployment and configuration

End point coordinates and route published over MQTT

AKraino EDGE STACK
Lifecycle Management

Configuration and Deployment
- Kubernetes YAML used to configure persistent containers: access control, back-end service discovery (e.g. MQTT gateway to Kafka or Webhook)
- The K8s containers can be deployed using Helm or external orchestrator
- Kubeless CLI/API used to deploy functions and pass deployment-specific parameters (persistent MongoDB URL, MQTT gateway for end point communications etc.)

Monitoring
- Kubeless supports native Kubernetes monitoring solution Prometheus, which collects key function KPIs (function invocation duration, invocation rate and success rate) per function
- Grafana visualizes the metrics collected by Prometheus data source
- Currently implemented per edge node, centralized metrics aggregation is feasible

Scaling
- The Kubeless engine launches docker container per function
- Additional containers are added using built-in auto-scaling Kubeless function
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Serverless Blueprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each initial blueprint is encouraged to take on at least two Committers from different companies</td>
<td>AT&amp;T, Cloudify Others - TBD</td>
</tr>
<tr>
<td>Complete all templates outlined in this document</td>
<td>Yes</td>
</tr>
<tr>
<td>A lab with exact configuration required by the blueprint to connect with Akraino CI and demonstrate CD. User should demonstrate either an existing lab or the funding and commitment to build the needed configuration.</td>
<td>The ‘CD Environment for Akraino Network Cloud Family Seed Code’ will be used. <a href="https://wiki.akraino.org/pages/viewpage.action?pageId=1147237">https://wiki.akraino.org/pages/viewpage.action?pageId=1147237</a></td>
</tr>
<tr>
<td>Blueprint is aligned with the Akriano Edge Stack Charter</td>
<td>This is part of the network cloud blueprint family which is aligned with the Akraino Charter</td>
</tr>
<tr>
<td>Blueprint code that will be developed and used with Akraino repository should use only Open Source software components either from upstream or Akriano projects.</td>
<td>This is addressed on slide 15 where upstream components and open source components are identified</td>
</tr>
<tr>
<td>For new blueprints submission, the submitter should review existing blueprints and ensure it is not a duplicate blueprint and explain how the submission differs. The functional fit of an existing blueprint for a use case does not prevent an additional blueprint being submitted.</td>
<td>This blueprint introduces serverless compute capability with functions in the network cloud blueprint family</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Serverless Blueprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name of the project is appropriate (no trademark issues etc.); Proposed repository name is all lower-case without any special characters</td>
<td>Proposed projects in LF Geritt: • edgility-mosca - the Mosca broker • edgility-web - the visualization component • edgility-generator – to run the demo for Serverless</td>
</tr>
<tr>
<td>Project contact name, company and email are defined and documented</td>
<td>Kandan Kathirvel, AT&amp;T <a href="mailto:kk0563@att.com">kk0563@att.com</a></td>
</tr>
<tr>
<td>Description of the project goal and its purpose are defined</td>
<td>Yes</td>
</tr>
<tr>
<td>Scope and project plan are well defined</td>
<td>Targeted for release 1 to support microservices use cases</td>
</tr>
<tr>
<td>Resources committed and available</td>
<td>There is a team, resources and lab in place</td>
</tr>
<tr>
<td>Contributors identified</td>
<td>AT&amp;T, Cloudify Others - TBD</td>
</tr>
<tr>
<td>Initial list of committers identified (elected/proposed by initial contributors)</td>
<td></td>
</tr>
<tr>
<td>Meets Akraino TSC Policies</td>
<td>The Project will operate in a transparent, open, collaborative, and ethical manner at all times</td>
</tr>
<tr>
<td>Proposal has been socialized with potentially interested or affected projects and/or parties</td>
<td>Proposal has been reviewed by the TSC</td>
</tr>
<tr>
<td>Cross Project Dependencies (XPDs). In the case where a project will require changes in other projects, those projects are listed in the proposal, and a sponsoring developer in the project has been identified</td>
<td>Leverages Unicycle POD as referenced on slide 14</td>
</tr>
<tr>
<td>Tools have been identified and discussed with relevant partners (Linux Foundation, IT). Once the project passes the review, the tools chain must be created within one week. Tools encompass Configuration Management, CI/CD, Code Review, Testing, Team Wiki, End Users documentation (not exhaustive)</td>
<td>No additional tools will be needed from the Linux Foundation. Existing CI tools can be used.</td>
</tr>
</tbody>
</table>
## Operational Tools Augmentation

CHOMP (Cluster Health and Overload Monitoring Platform) will do log correlation for Akraino lifecycle management

1. Log-based metrics in CHOMP provide key visibility for operations teams beyond the existing traditional monitoring tools such as Prometheus
2. These KPIs could become the beginning of troubleshooting and root cause identification as NC becomes ready for production deployments
3. Broad categories include:
   a. Latencies for common Kubernetes procedures
   b. Failure details - insufficient memory or computing capacity
   c. Procedure details - restart backoffs, pod evictions prior to restarts
   d. Kubernetes Component Availability
4. Modular design and Configurable for easy addition of new log-based metrics as NC platform matures

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Companies Participating / Committers</th>
<th>Requested Release / Timeline</th>
<th>Informational</th>
</tr>
</thead>
</table>
| Operational Tools Augmentation | CHOMP (Cluster Health and Overload Monitoring Platform) will do log correlation for Akraino lifecycle management | AT&T | RI | Impacted Blueprint Family - Network Cloud  
See next slide for additional details |
Akraino Feature Project - CHOMP

Akraino GUI
- Dashboard
- Admin GUI
- User GUI

Akraino Workflow
- Platform Workflows
- Camunda

Edge Application and APIs
- APIs
- Applications & VNFs
- Edge APIs
  - Sample Edge App (CDN)

Edge Application and Orchestration
- Lightweight Edge App Orchestration

Edge Platform Software Components
- Infra Orchestration
- Storage
- OpenStack (Ocata)
- Kubernetes
- SDS(Ceph)
- Calico
- SR-IOV
- OVS
- Ubuntu

NFV Orchestration
- NFV & Domain Specific Orchestrator
- ONAP Amsterdam

NFV Orchestration
- ONAP Amsterdam

Network Edge
- NC – Multinode Cluster
- Single Server (Rover)

Network Edge
- Serverless

Network Edge
- Micro Services

Customer Edge

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

Akraino Upper
Cloud Lifecycle Tools

AirShip Under
Cloud Lifecycle

Dawn*

*To be contributed to Airship

Akraino - new

Upstream

Future release

CHOMP

Documented by

Customer Edge

Akraino - new

Upstream

Future release

CHOMP

Documented by

Network Edge Single Server (Rover)
**CHOMP Containerized Architecture**

- **Configuration files in JSON** - allow for ease of addition for new KPIs
- **New events can be counted** by adding one or more signature files to data collector, new counter config or correlated log config

**Data Signatures**
- Sig 1 Cfg
- Sig 2 Cfg
- ... Sig N Cfg
- Regex File

**Single Log Configuration**
- Counter cfg 1
- Counter cfg 2
- Counter cfg M

**Single-log event Counter**
- Counts events/stats for K8s events based on a single K8s log

**Correlated Log Event Counter**
- Specifies sequence of log signatures required for some event detections

**Correlated Log Configuration**
- Corr 1 Cfg
- Corr 2 Cfg
- ... Corr N Cfg

**ElasticSearch DB**

**Data Collector**

**Redis**

**PostGresDB, Akraiino Portal UI**

---

**NC Cluster**

- **K8S Master**
  - FluentD/FluentBit
- **K8S Node**
  - FluentD/FluentBit
- **K8S Node**
  - FluentD/FluentBit

---

- Specifies log signatures to query from ES
- Specifies counting rules for single-log event counter
- Counts events/stats for K8s events based on a sequence of K8s logs
- Specifies sequence of log signatures required for some event detections
For More Information, Please Visit www.akraino.org