Integrated Cloud Native NFV & App stack - Proposals
## Integrated Cloud Native NFV & App related blueprints

<table>
<thead>
<tr>
<th>Use cases</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• SDWAN</td>
<td></td>
</tr>
<tr>
<td>• Distributed Analytics as a service</td>
<td></td>
</tr>
<tr>
<td>• IOT framework - EdgeXFoundry</td>
<td></td>
</tr>
<tr>
<td>• Tiled Streaming Video CDN</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Blue print family</th>
<th>Integrated Cloud Native NFV &amp; App family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blueprints</td>
<td>• Multi-Server Cloud native NFV/App stack</td>
</tr>
</tbody>
</table>

| Contributors                                  | • Verizon, Intel, MobileEdgeX, Aarna Networks, VMware |

Will submit another BP with MobileEdgeX in this family
# Blueprint Proposal: Multi-server Cloud native stack

<table>
<thead>
<tr>
<th>Case Attributes</th>
<th>Description</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Blueprint</td>
<td></td>
</tr>
<tr>
<td>Blueprint Family - Proposed Name</td>
<td>Integrated Cloud Native NFV stack</td>
<td></td>
</tr>
<tr>
<td>Use Case</td>
<td>SDWAN, Customer Edge, Edge Clouds – deploy VNFs/CNFs and applications as micro-services</td>
<td></td>
</tr>
<tr>
<td>Blueprint proposed Name</td>
<td>Multi-server Cloud Native stack</td>
<td></td>
</tr>
<tr>
<td>Initial POD Cost (capex)</td>
<td>50K minimum</td>
<td></td>
</tr>
<tr>
<td>Scale &amp; Type</td>
<td>Minimum of 4 Xeon Servers + 1 Xeon server as bootstrap node.</td>
<td></td>
</tr>
<tr>
<td>Applications</td>
<td>SDWAN, ML/DL Analytics, EdgeXFoundry and 360 degree Video streaming</td>
<td></td>
</tr>
<tr>
<td>Power Restrictions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Infrastructure orchestration**
- Bare Metal Provisioning: ironic or equivalent
- Kubernetes provisioning: Cluster Operator with KuD.
- Docker for containers and Virtlet for VMs
- Service Orchestration: ONAP
- MEC framework: OpenNESS
- Site orchestrator: Kubernetes upstream
- Traffic Orchestration within a cluster: ISTIO
- Traffic orchestration with external entities: ISTIO-ingress
- Knative for function orchestration

**SDN**
- OVN, SRIOV, Flannel

**Workload Type**
- Containers, VMs and functions

**Additional Details**
- Future: eBPF based CNI (such as PolyCube)

- Expose HW accelerators
- Storage Orchestration: Ceph with Rook
- Future: AF-XDP for packet processing based containers, OpenShift for site orchestrator, Kubevirt for VMs.
## Assessment Criteria

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Multi-Server Edge NFV/APP stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each initial blueprint is encouraged to take on at least two committers from different companies</td>
<td>Verizon, Intel, MobileEdgeX, Aarna Networks, VMWare</td>
</tr>
<tr>
<td>Complete all templates outlined in this documents</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>Continuous Deployment will be provided in Intel community lab (Similar to the way we did for OPNFV)</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 open source. Actual applications (use cases) in some cases are from third party</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>An comprehensive platform with all kinds of deployment types – VNFs, CNFs, Micro-Services and functions, Advanced networking support, Multi-tenancy, slicing, OVN based data interfaces</td>
</tr>
</tbody>
</table>

### Criteria

- **Name of the project is appropriate (no trademark issues etc.); Proposed repository name is all lower-case without any special characters.**
  - Integrated Cloud Native NFV & App stack

- **Project contact name, company, and email are defined and documents**
  - Identified. Will be documenting them in the page

- **Description of the project goal and its purpose are defined.**
  - Developing an integrated cloud native stack solution for VNFs, CNFs

- **Scope and project plan are well defined.**
  - For release 2. Yes. Various milestones with the release are identified.

- **Resource committed and available**
  - There is a team, resources and lab in place.

- **Contributors identified**
  - Verizon, Intel, MobileEdgeX, Aarna Networks, VMWare

- **Initial list of committers identified (elected/proposed by initial contributors)**
  - Yes. PTL and 4 Engineers are identified.

- **Meets Akraino TSC policies**
  - The project will operate in a transparent, open, collaborative, and ethical manner at all the times.

- **Proposal has been socialized with potentially interested or affected projects and/or parties**
  - ○ Have already reached a consensus with sponsors

- **Cross Project Dependencies.**
  - CNCFs Projec, K8s, CRi, OCI, Virtlet, Kubevirt, Kata container, gVisor, OpenWRT, Docker, OVN, OVS, DPDK, AF, XDP, ONAP
Transformation journey (to Kubernetes)

Two different resource orchestrators
Compute nodes are divided

- K8S for VNFs, CNFs, Micro-Services and functions
- Soft Multitenancy with one K8S
- Strict Multitenancy with K8S clusters from VMs (when required)
<table>
<thead>
<tr>
<th>Goals of ‘Integrated Cloud Native NFV &amp; App’ BPs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Co-existence of multiple deployment types</strong></td>
</tr>
<tr>
<td>(VNFs, CNFs, VMs, Containers and functions)</td>
</tr>
<tr>
<td><strong>Advanced Networking support</strong></td>
</tr>
<tr>
<td>(Multiple networks, Provider networks, Dynamic Route/network creation, Service function chaining)</td>
</tr>
<tr>
<td><strong>Soft and Strict Multi-tenancy</strong></td>
</tr>
<tr>
<td><strong>AI based Predictive placement</strong></td>
</tr>
<tr>
<td>(Collection using Prometheus, Training and inferencing framework)</td>
</tr>
<tr>
<td><strong>Slicing in each tenant</strong></td>
</tr>
<tr>
<td>(QoS On per Slice basis, VLAN networks for slices, VNFs/CNFs/VMs/PODs on per slice basis or slice configuration facility on shared VNFs/CNFs)</td>
</tr>
<tr>
<td><strong>Multi Site Scheduler using ONAP</strong></td>
</tr>
<tr>
<td>(Auto Edge registration, Workload placement, On-demand tenant/slice creation)</td>
</tr>
<tr>
<td><strong>Service Mesh for Micro-services</strong></td>
</tr>
<tr>
<td>(Acceleration using Cilium’ Kernel bypass among service mesh side cars - e.g. Envoys; and others)</td>
</tr>
<tr>
<td><strong>Programmable CNI</strong></td>
</tr>
<tr>
<td>(to allow SFC and avoid multiple protocol layers)</td>
</tr>
<tr>
<td><strong>Security Orchestration</strong></td>
</tr>
<tr>
<td>(Key orchestration for securing private keys of CA and user certificates)</td>
</tr>
</tbody>
</table>

*Prove with either test cases or use cases*
Managed SDWAN use case

- OSS/BSS
- Service Orchestrator (ONAP)
- Subscriber
- Operator

Network Edge
- SDWAN VNF
- WAN Opt CNF
- Security VNF
- Operator slice

Cloud
- Advanced network services (e.g., DLP)
- AR/VR Services
- Media Analytics
- MEC Services
- Training

ISP A
- Tunnel
- ISP X
- ISP B
- ISP Y
- CoSP / Internet

Subscriber Edge1

Kubernetes

Advanced network services (e.g., DLP)

May be tested with dummy apps and OpenWRT, if no real apps from community

AR/VR Services
- Media Analytics
- MEC Services
- Training
How does NFV based deployment with Cloud-native applications look like (Taking SDWAN with security NFs as an example)

What it proves:
- Coexistence of VNFs, CNFs and Micro-services
- Advanced networking support
- Soft-Multitenancy
- Multi-Cluster scheduler via ONAP
- Service mesh for Micro-services
- Programmable CNI

Initial testing to be done using Ubuntu for all VNFs/CNFs.
Distributed Analytics as a Service
(Each site to have self contained inferencing and few sites with training)

Service Orchestrator (ONAP)

1. Onboard Analytics framework (With 6 bundles) in catalog
2. Activate framework
3. Onboard an analytics app
4. Activate analytics app

Federated Learning (future)
EdgeXFoundry use case

**CORE SERVICES**
- Simple Rules Engine
- Databases
- Security

- Kubernetes

**Additional Services**
- Service Orchestrator (ONAP)
- Machine Learning
- Logging
- Security
- UI
- Graphs

**Operator**

**“SOUTHBOUND” DEVICES, SENSORS AND ACTUATORS**
VR 360 streaming—Enable remote users to view the events/games via Edge-computing

1. Onboard App (With 6 services) in catalog with deployment intent
2. Active App (when event starts)
3. New users join the event (Auto bring up of services at Edge1 and Edge3)
4. Users join in a geo that requires additional context to be added to the stream.
5. More users join near edge1
6. Users disappear Edge2
Cloud Native App & NFV Stack – Putting it all together

**Use cases/Apps**
- Analytics as a Service
- SDWAN + Security NFs
- EdgeXFoundry
- Tiled streaming with Video CDN

**Kubernetes**
- ISTIO
- gVisor
- Kata
- CollectD
- Virtlet, Kubevirt
- Ceph/Rook

**K8S App Components**
- MetallB
- Flannel, OVN
- Prometheus
- OpenNESS
- knative

**K8S Specific components**
- Multus
- SRIOVNIC
- OVN4K8SNFV
- SFC Mgr
- OVN4SFC
- NTWRK/Route Mgr

**Host Operating System**
- Ubuntu (start with this)
- RH
- Clear

**Hardware**
- S1
- S2
- S3

**ONAP**
- Tenant Mgr
- Edge Label Mgr
- Slice Mgr
- Multi-Site scheduler
- K8S HPA
- MC – K8S Plugin Service (Instantiation, Day0, Day2 config)

**PaaS**

**Global ZTP (Zero Touch Provisioning) system**

**Infrastructure Provisioning & Configuration**
- KUD (with Cluster API)
- Ironic with Metal3(or equivalent) for bare-metal provisioning

**Future**

**Use cases/Apps**
- Analytics as a Service
- SDWAN + Security NFs
- EdgeXFoundry
- Tiled streaming with Video CDN

**Kubernetes**
- ISTIO
- gVisor
- Kata
- CollectD
- Virtlet, Kubevirt
- Ceph/Rook

**K8S App Components**
- MetallB
- Flannel, OVN
- Prometheus
- OpenNESS
- knative

**K8S Specific components**
- Multus
- SRIOVNIC
- OVN4K8SNFV
- SFC Mgr
- OVN4SFC
- NTWRK/Route Mgr

**Host Operating System**
- Ubuntu (start with this)
- RH
- Clear

**Hardware**
- S1
- S2
- S3

**ONAP**
- Tenant Mgr
- Edge Label Mgr
- Slice Mgr
- Multi-Site scheduler
- K8S HPA
- MC – K8S Plugin Service (Instantiation, Day0, Day2 config)

**PaaS**

**Global ZTP (Zero Touch Provisioning) system**

**Infrastructure Provisioning & Configuration**
- KUD (with Cluster API)
- Ironic with Metal3(or equivalent) for bare-metal provisioning

**Future**
BACKUP
Edge Computing – Cloud gaming app example

1. Game Selection
2. Slice bring up and App bring up
3. New player joins up, expand slice & add new micro-services
4. Game interaction
5. Distributed Rendering
6. Playout

1. **Game Selection**
2. **Slice bring up and App bring up**
3. **New player joins up, expand slice & add new micro-services**
4. **Game interaction**
5. **Distributed Rendering**
6. **Playout**

---

**Edge Com**

**Game Selection Service**

**Service Orchestrator**

**Cloud/DC**

**Edges**

100s to Millions (Millions in case of client edges)

**P1 - Edge1**

**P1 - Edge2**

**P2 - Edge 3**

**Pm Edge N**

**Game Simulation Engine**

**Render**

**Encode**

**Streamer**

**Distributed Rendering**

**Playout**

**Thin Client**

**Select Game**

**Game interaction**

**Codec, blend & Playout**

**Render**

**Game Services**

**GPU Offload**

**Bring up services with the help of service orchestrator (e.g. ONAP)**

---

**PC Client**

**Example: Cloud gaming app**

**Thin Client**

**Select Game**

**Game interaction**

---

**Px Edge Provider x**

**Edge**

**Slice**

**Game Services**
Thank you