Linux Foundation's Project EVE: A Cloud-Native Edge Computing Platform

Roman Shaposhnik AKA @rhatr
Founder, VP of Product & Strategy @ZEDEDA Inc.
Board member @Apache Software Foundation & Linux Foundation
Edge Computing is... “cloud-native” IoT
Edge ain’t your gramp’s
Embedded and/or IoT
Data must be pre-processed at the edge due to bandwidth, latency and cost.
Are you ready to live on the Edge?

• **Edge is one final “Cloud” we’re building**
  o Remember: everything has to be “Cloud Native”
  o Edge DevOps anyone?

• **So… Edge is just another cloud?**
  o Yes and no. It is more like mobile + DC

• **Can we rub some Kubernetes on it?**
  o APIs – most likely
  o Implementations -- absolutely not!

• **Economics of the Edge**
  o Super-heterogeneous ownership
  o Huge business opportunity seen by VCs
  o AI (especially autonomous) is a ”killer app”
Challenges at the Edge

- Diversity of hardware and apps
  - Infrastructure management
  - Orchestration of apps

- Scale and automation
  - Geographically disperse
  - Deployment and maintenance

- Security – increased threat vector
  - No perimeter network security
  - No perimeter physical security

- Vendor lock-in is impossible
  - Distributed Ownership…
  - …hence it has to be open
Guarding Against Physical Attacks: The Xbox One Story — Tony Chen, Microsoft

Tony Chen
Microsoft
Platform Security Summit 2019
10/1/2019
From the people who brought you CNCF
Go West, young man, go Edge!

EDGE
MEC server, AI/ML, IoT, 5G

EDGE
97% of operators plan VNF execution in Smart CO

EDGE
vNFs, vEPC, MEC, distributed RAN, vRAN, BBU hotel, FMC, vCPE, AI/ML, IoT go here

PARTIAL EDGE
85% of operators plan VNF execution in DC Near CO

NOT EDGE
70% of operators plan VNF execution in DC Not Near CO

20msecs

THE LINUX FOUNDATION
LF Edge Projects

IOT Interoperability framework

Telco Edge Use Case

EDGE VNFs, vEPC, MEC, distributed RAN, vRAN, BBU, hotel, PM, vCPE, AI/ML, IoT go here

PARTIAL EDGE
85% of operators plan VNF execution in DC Near CO

NOT EDGE
70% of operators plan VNF execution in DC Not Near CO

Data Center Near CO [Regional DC]

Data Center Not Near CO [Central DC]
LF Edge Projects

Drivers

› Complementary and aligned vision on multiple LF projects
› Fuels faster adoption and deployment
› Edge market is fragmented and creating a larger entity provides leadership

Projects
Edge Requirements

- **ZERO TOUCH**
- **FREEDOM OF ANY**
  APP | HARDWARE | CLOUD
- **IoT SCALE**
- **CLOUD NATIVE**
- **ZERO TRUST**
App deployment is but the tip of the iceberg

Edge Virtualization Engine

EVE-EVC API - config, status, metrics, logs

Device connectivity

Instance connectivity

Remote instance consoles

EVErouter:
- DHCP
- DNS
- ACLs
- LISP
- VPN

EVEagent:
- config
- status
- metrics

EVEmanager:
- instance orchestrator

Verifier
- sha, sigs

HW info, metrics

Log manager

Domain ring

I/O virtualization and assignment

Driver domain(s)

Instance A
Instance B
Instance C
Instance D

Linux watchdog
Baseos manager
Network interface manager

Mesh network
Downloader

Device onboarding

Crypto device identity

Eth, wlan, wwan

Eth, wlan, wifi

TLS 1.2/1.3 OCSP stapling

Instance B

Instance C

Instance D

Run “apps” at the edge

Support any app on any HW

Manage connectivity

Secure the data & device

Monitor & manage all edge resources and EVE image

TEE/TPM
Hardware Layer
Eth, RS 485, BTLE etc

© 2019 ZEDEDA, INC
A complete Edge ”Cloudification” proposal
Edge Infrastructure Challenges Solved with **Edge Virtualization**

**EVE Controller**
- Web Console
- App Marketplace
- Deploy, Secure and Manage Gateway and Apps at Scale
- SaaS

**Edge Virtualization Engine (EVE)**
- Abstraction layer *designed* for the edge
- *Created* and donated by ZEDEDA to LF
- Open sourced under Apache License v2
- Part of Linux Foundation LF Edge Project

---

**Any Gateway at IoT Scale**
- Historian, SCADA or On-Premises System
- Any Application
- VM or Container
- No Compromise to Security (TPM and vTPM)
- Any Cloud
4 pillars of complete Edge “Cloudification”
EVE’s architecture

- **Type** - 1 Hypervisor (currently Xen)
- Networking, I/O virtualization, etc. (AKA side-cars you don’t have to think about)
EVE is going to be for the Edge…
....what Android is for Mobile

<table>
<thead>
<tr>
<th></th>
<th>EVE (Edge Virtualization Engine)</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>App and OS Sandboxing</td>
<td>Hardware Assisted Virtualization</td>
<td>JVM</td>
</tr>
<tr>
<td>App Bundling</td>
<td>Edge Container (ECOs)</td>
<td>APK</td>
</tr>
<tr>
<td>App Deployment</td>
<td>Cloud Orchestrated &amp; Pre-loaded</td>
<td></td>
</tr>
<tr>
<td>H/W support</td>
<td>Intel, ARM (+RISC V)</td>
<td>Intel, ARM, MIPS</td>
</tr>
</tbody>
</table>
EVE: a post-, post-modern OS
EVE deep dive... could be pretty deep
LF Edge’s EVE deep dive

• Inspired by QubesOS, ChromeOS, SmartOS
• Based on Type-1 Hypervisors (Xen or ACRN)
  o No KVM allowed!
  o Containers are fine, but everyone gets a lightweight VM
• DomU is…
  o linuxkit
  o Alpine Linux
• But wait, there’s more:
  o We are driving towards unikernel architecture
  o Everything is Golang based
  o Moving to AtmanOS (GOOS=xen go build …)
• Introducing: Edge Containers
Edge Containers

• A true extension to the OCI specification
  • Image specification (not much of a change)
  • Runtime specification
  • Registry Support (via OCI Artifacts Initiative)

• Related initiatives
  • Kata Containers, Singularity Containers, etc.
  • Weave.works’s Project Ignite (Firecracker MicroVMs)
  • Rancher’s K3S + K3OS

• Top 3 goals:
  • Filesystem-level composition (aka OCI layers)
  • Block-level composition (VMs and Unikernels)
  • Hardware mapping

• Registry as a ”nexus of Liquid Software”
EVE’s networking is intent based

- Directly assigned hardware (Edge Container capability)
- Switch Network
  - A simple, virtualized L2 network (Ethernet++)
- Local Network
  - A traditional, L3 (IP++), NATed network
- Cloud Network
  - "Please connect me to this AWS VPC"
- Mesh Network
  - Based on LISP RFC 6830
  - Gives you a flat IP6 overlay with…
  - …crypto-identity based routing
  - …service mesh (regardless of NATs, etc.)
EVE’s trust model – Zero Trust

- Trusted systems don’t exist, trustworthy ones may
- Root-of-trust
  - Always derived from a hardware element (TPM, TEE, etc.)
  - Hardwired root CA cert for Controller Trust
- Measured boot with EVE Controller fencing
- Crypto identity for all elements in the system
- No ssh access, no usernames/passwords
- Defense-in-depth (kudos to Qubes OS)
  - Hypervisor-enforced isolation
  - Stub domains for drivers
  - Microservices running as Unikernels
EVE’s software update model

- **Prevent bricking by**
  - Applications are easy: just redeploy
  - EVE itself: dual partitioning + multiple levels of failover

- **Avoid the need for physical contact with Edge Nodes**

- **Manage everything starting from Firmware**
  - Good news: we are based on UEFI…
  - …which also happens to be bad news
  - Coreboot is really exciting
  - Don’t deploy things you don’t need (ILOs, BMCs)
Hardware-protected vTPM 2.0

**Current Landscape**

Multiple vTPMs published or under development, but few vTPMs are protected with hardware mechanisms.

No public vTPM addresses the TPM 2.0 requirements of shielded functions:

- vTPM contents can easily be influenced

**QEMU Virtual TPM:**

- Instances run as user space processes
- Separation provided by OS kernel

**Proposed Approach**

Based on TPM 2.0 spec and reference code (Microsoft)

Provide a BSD-licensed vTPM implementation that isolates each vTPM instance on a platform, provides the complete TPM 2.0 interface, and can be used by standard OS drivers for TPM 2.0

**Strong Isolation Properties:**

- Isolate runtime for the Protected Capabilities and the Shielded Location for the Protected Objects
- Platform Security: leverage SGX, memory encryption and other hardware-based separation technologies
Hardware-protected vTPM 2.0

Use Cases:

● Cryptographic key generation and protection, e.g. Windows Bitlocker or other disk encryption keys
● Measured Launch (SRTM/DRTM)
● Integrity Measurement & Attestation
● Local Hardware Security Module (HSM)

Initial Participants:

● TrenchBoot, OpenXT, QubesOS / Invisible Things Lab, LF Edge Project EVE / Zededa

Target for open-source implementation delivery:

● Q3 2020

Collaborators Welcome:

● Requirements & Design
● Implementation & Validation
● Crowdfunding & OSS/commercial adoption

CONTACT

● DPSmith@ApertusSolutions.com
● trenchboot.github.io
● LF Edge
Key takeaways

• Edge Computing today is where Public Cloud was in ’06
  • It is a pioneer’s land - sorry “settlers” and “town planners”

• Edge Computing is the one final Cloud left and it is the only one that can NEVER be taken away from us

• Edge Computing represents a HUGE TAM
  o VC activity is really picking up

• Kubernetes (implementation) is dead – long live Kubernetes (APIs)

• Edge Computing is a lot of fun, so…
  • Help us build LF Edge EVE…
  • …or pick any other LF Edge Project
THANK YOU!

Follow EVE and LF Edge:
@eve_edge
@LF_Edge