Solution Brief

Virtualized Network Functions (VNFs): Works on Arm™

Understand Open Linux Virtualization for vCPE solution on Armv8

Intelligent devices have revolutionized the way we experience the world and have created a limitless platform for innovation. The infrastructure that enables and powers these intelligent devices, the network, is in need of its own revolution. Multicore processing and virtualization are becoming ubiquitous in software development.

Fig 1: Next Generation Networks Drivers

MontaVista continues to participate in the way networks are created and behave by providing necessary software, tools and support to help custom designs, meet the need to be:

- **Flexible:** Next Gen. network will be characterized by a much higher native flexibility and programmability for all non-radio network segments including SDN, NFV And IoT Networks. This level of complexity necessitates the selection of a reliable system software like MontaVista CGX.
- **Reliable: MontaVista CGX is a** Yocto™ based Linux solution with pre-certified Linux BSP (Board Support Packages) on the latest and greatest hardware from our SOC partners. Additional profiles for Virtualization and Data-Path along with bring-up support extend feature support for these next generation networks.
- **Secure:** MontaVista CGX Security profile can be customized to enable secure gateway partitions, utilizing an Arm TrustZone® enabled secure world environment. This offers secure boot, applications including certificate management, secure firmware upgrades, and secure data storage.

BENEFITS

- OUT OF BOX EXPERINECE WITH PRE-TESTED BSPs
- REAL TIME LINUX + Virtualization (KVM + Containers)
- Multi-Architecture & HW/FPGA Optimization
- BUILT IN FLEXIBILITY, RELIABILITY AND SECURITY
- SECURE LIVE KERNEL AND APPLICATION UPDATE

APPLICATIONS / USE CASES

- 5G CARRIER GRADE INFRASTRUCTURE
- IOT AND SMART GATEWAY
- SMART HOME AMONG OTHERS

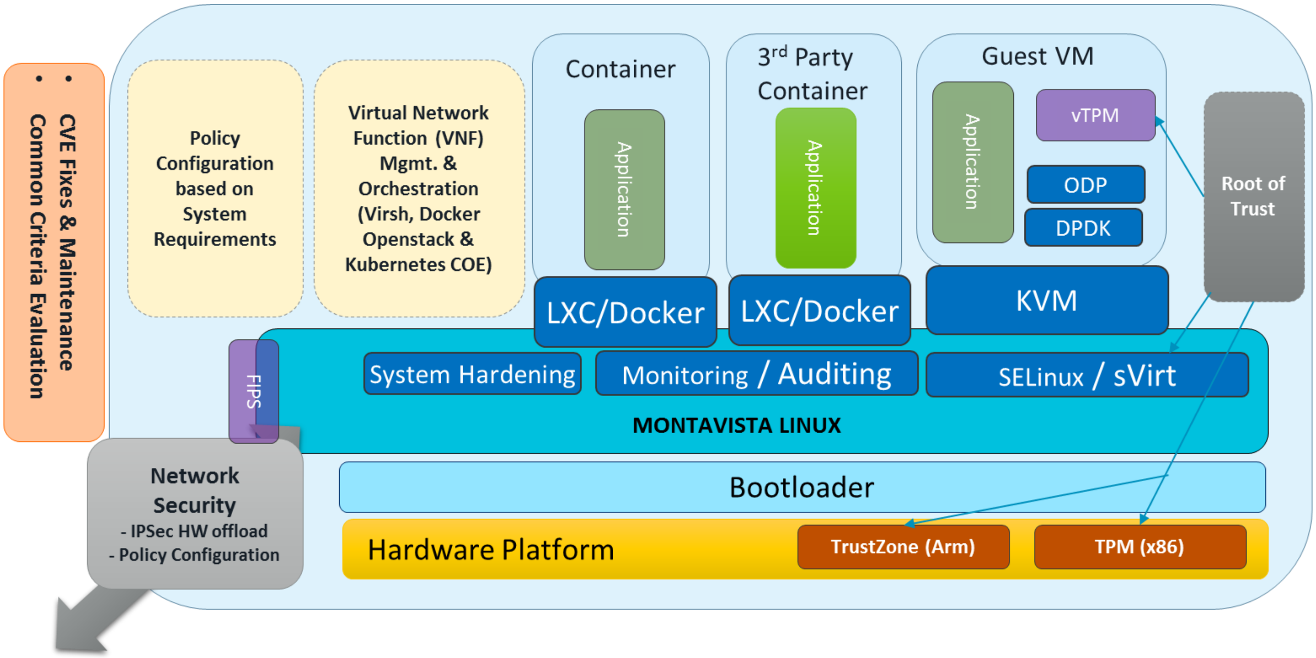

Fig 2: Virtualization & Security Solution Architectural Design with MontaVista™

Next Generation Networks

Network data centers are undergoing major transformations by introducing virtual network devices to provide the agility and efficiency required today. Until recently, deployment of new services was done with purpose built telecommunications equipment. However, telecommunication service provides are demanding for higher agility with better costs to keep up with the rapid expansion in the user base, the increasing services offerings along with the rate technology innovation.

Multicore processing and virtualization are rapidly becoming ubiquitous in software development. They are widely used in the commercial world, to

- Reduce CAPEX by Isolating application software from hardware and operating systems, enabling different applications to share underutilized computers or processors.
- Improve OPEX through system scalability at a fraction of time and cost along with high reliability and robustness by limiting fault and failure propagation and support failover and recovery.

Virtualization

Virtualization can be described as a method for dividing the resources of a computer into multiple execution environments that enable software applications to run on

- A virtual hardware either through full virtualization (KVM or similar) or a proprietary Hypervisor.
- A virtual operating system (such as a container)-by extending the resources of the native OS, without emulating the underlying hardware.

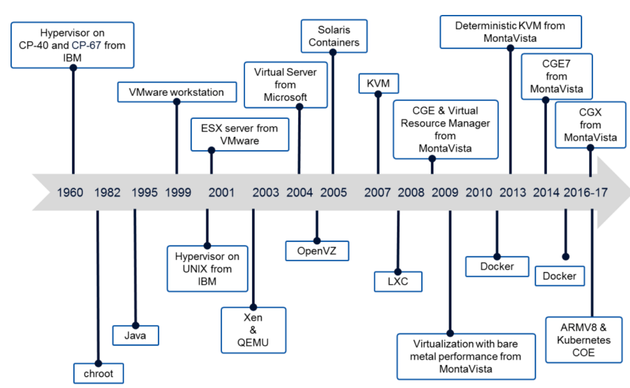

Fig 3: Brief History of Virtualization

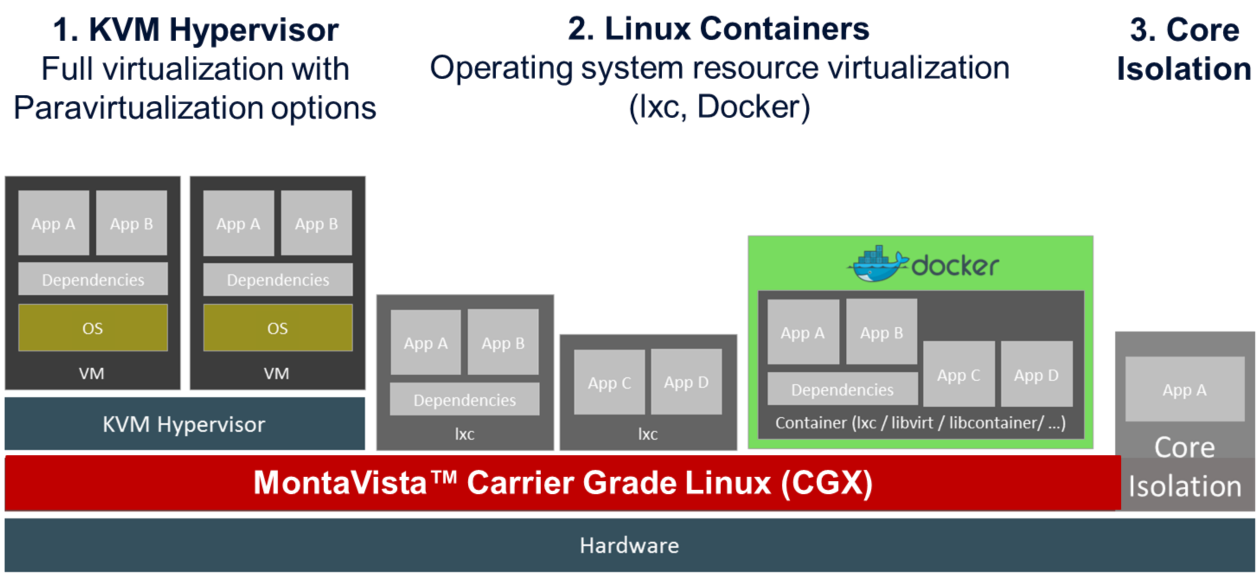

Fig 4: Summary, Secure Boot and Key Management on the reference HW

Let's explore some of these methods,

KVM Hypervisor

KVM (Kernel-based Virtual Machine) is a hypervisor-based full virtualization solution for Linux on x86 and Arm hardware with ports under development for MIPS~~,~~ and PowerPC architectures. It consists of a loadable kernel module that provides the core virtualization infrastructure and a processor-specific module for ARM and x86 processors, and the equivalent for other architectures. KVM also uses a modified QEMU to support the I/O requirements of the virtual environment.

Using KVM, one can run multiple virtual machines hosting both Linux and non-Linux operating systems native to the underlying processor architecture. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc. For embedded SoCs, each implementation of KVM needs to determine which of the SoC-specific hardware devices will be incorporated into the KVM environment. The kernel component of KVM has been mainline Linux as of revision since 2.6.20.

LXC/Container

An OS-level virtual environment, Containers provide lightweight virtualization that isolates processes and resources without the complexities and overhead of full virtualization. LXC provides a virtualized environment that supports multiple process and network name space instances as well as extensive resource allocation and control capabilities. Access to resources for a Container can be controlled with Control Groups (cgroups). Namespaces provide resource isolation for implementing Containers.

Container starts at the same place as a regular Linux distribution: it starts by running init. Containers may be used for a number of different use cases. These include:

- Resource partitioning with maximum performance
- Multiple secure applications instances (e.g. a walled garden)
- Process isolation (e.g. process jails)
- GPL insulation

Docker™

Docker™ is a project by Docker Inc , initially based on the LXC project to build single application containers. Docker has now developed their own implementation libcontainer that uses kernel container capabilities directly. Designed specifically to support a single application per container.

Virtual Network Function

Historically, network functions components of a network infrastructure that have specialized functional behavior (e.g. routers, firewall etc.), have been deployed as physical appliances. Simply put here, the software is tightly coupled with specific, proprietary hardware.

These physical network functions need to be manually installed and configured into a network, creating operational challenges and preventing rapid deployment of new network functions. A Virtual Network Function (VNF), on the other hand, refers to the implementation of a network function using software that is de-coupled from the underlying hardware. This can lead to more agile networks, with significant Opex and Capex savings. These are also known as virtual appliance.

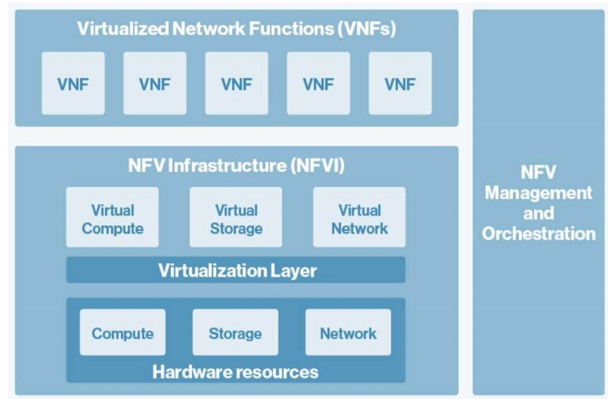

Fig 5: NFV Arch. Overview

Virtualized Network Function – The Evolution

While the move from physical appliance to a more versatile virtual appliance has been well understood, the details around the framework for implementing and managing virtual network functions (VNFs) have been evolving over time. It is interesting to note that the earlier implementation utilized either an open virtual machine (that usually had some hardware specific optimization) or a proprietary hypervisor solution. This led to scalability and interoperability issues limiting adoption by SLA driven real world contract. In addition, they also posed a technical roadblock for application developers and service providers.

Another challenge was the use case them e.g. Industrial control solution that often have to smaller footprint & power needs. Such observations have led to a much larger adoption of –

- Containers based implementation (Docker™ & Kubernetes™)
- Light weight & configurable Yocto™ compliant e-Linux®
- Higher adoption of Arm CPUs

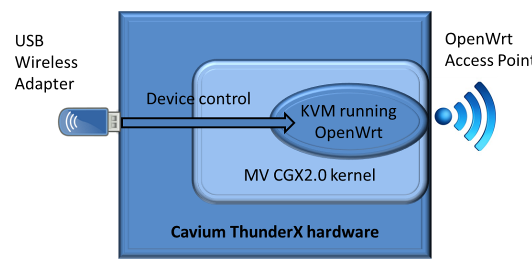

Fig 6: Sample VNF

This transition has been driven by open communities like OpenNFV (OPNFV), who have embraced the integration of NFV solutions incorporating ARM servers for creating usable and cost efficient NFV use cases.

MontaVista and Arm – Ready for mainstream

Increasingly, configurable (small footprint) Linux like CGX is being considered along with Armv8 architecture to achieve the highest level of optimization. These are especially true on the price to performance conscious virtual Client Premise Equipment's (vCPE). The reasons are not hard to understand, given the benefits:

BENEFITS (ARM SOCs: range of accelerators/offload **)**

- Packet Processing
- Crypto functions
- Traffic management
- Full IPSec Offload
- Virtual Switching Offload
- Network Virtualization protocols

BENEFITS (MontaVista CGX)

- Out of Box Commercial Linux (BSPs, Yocto™ User-land, Tools, Eng. Support)
- Proven workload efficiencies on Armv8 CPUs
- Up to 10x performance jump (DPDK, ODP & OVS)
- Lower latencies with native Real-Time

Solution Brief

Smart IoT Gateway: Deploy Secure Virtual Network Functions

Arm “mBed” device cloud connector with Open Virtualization

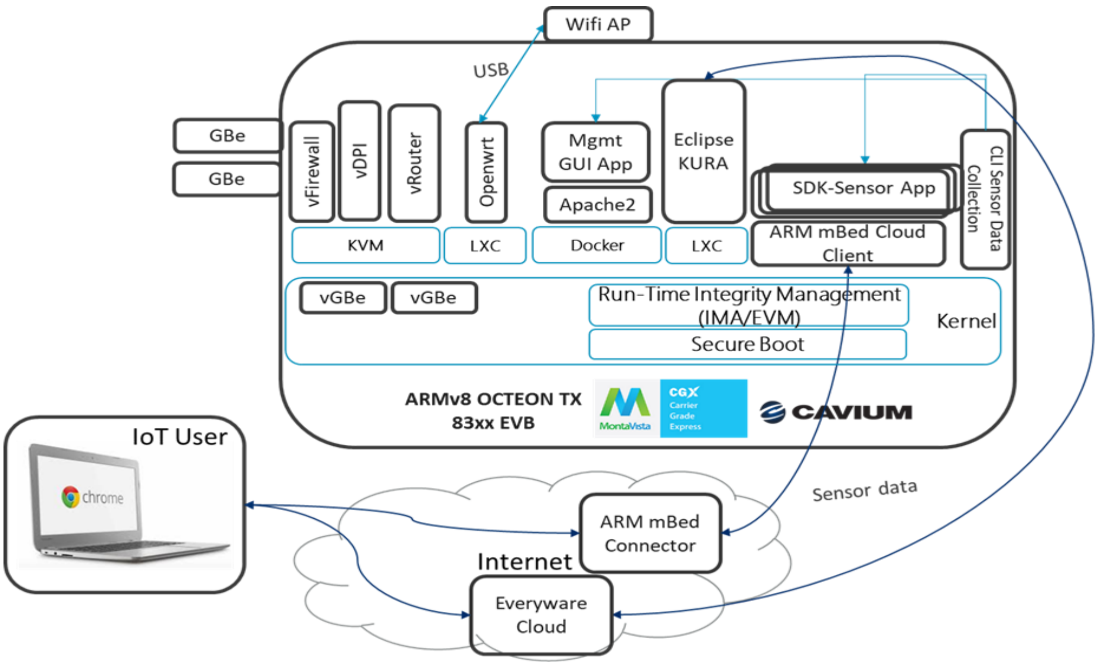

Fig 7: Case Study: SMART IoT Gateway

SMART IoT Gateway

Fig. 7 shows a reference solution for a smart IoT gateway that showcases the ecosystem of software components that are optimized for MontaVista Carrier Grade Linux. These include: Open source virtualization that allow multiple virtual network function to share the resources, an advanced Security and Arm mBed device cloud integrations for remote device management.

Virtualization and Virtual Network Functions (VNFs)

This solution showcases use of open virtualization technologies like KVM, LXC/Docker & Kubernetes (Production-Grade Container Orchestration) for isolating Virtual Network Functions (VNFs), applications and system software.

Security

If an attacker were to compromise the IoT gateway, not only the data passing through the gateway is at risk, but control of the physical things connected to it are at risk as well. By implementing a true end-to-end, security solution that ensured a system wide “Root of Trust” and a “Trusted Execution Environment” for applications, we demonstrate the capability to meet the need for security in a modern connected IoT device/gateway.

Device Cloud Integration

ARM mBed IoT Device Platform provides operating system, cloud services, tools and developer ecosystem for creation and deployment of commercial, standards-based IoT solutions. It is made up of components such as device software and cloud based device management services that enable movement of data from sensor to server.

The example VNF solution has a mBed Linux client running on ThunderX gateway

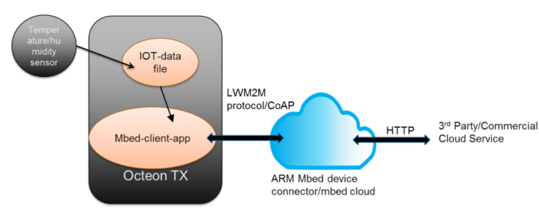

Fig 8: ARM mBed Cloud

that reads data from a cluster of such devices (ex: Temperature and Humidity sensors from outside world) in real time. The data is then encapsulated into a resource using client libraries and published to mBed cloud or mBed device connector using LWM2M (Light Weight Machine 2 Machine) protocol. Security key is also embedded so that mBed cloud registers it as valid device through registration services/APIs. The sensor data is now made available on the med-device-connector or on mBed-Cloud.

Deliver Flexibility, Reliability & Security for the 5G Wireless Network

“Network” for “Things” : MontaVista CGX®+ 5G Application Service

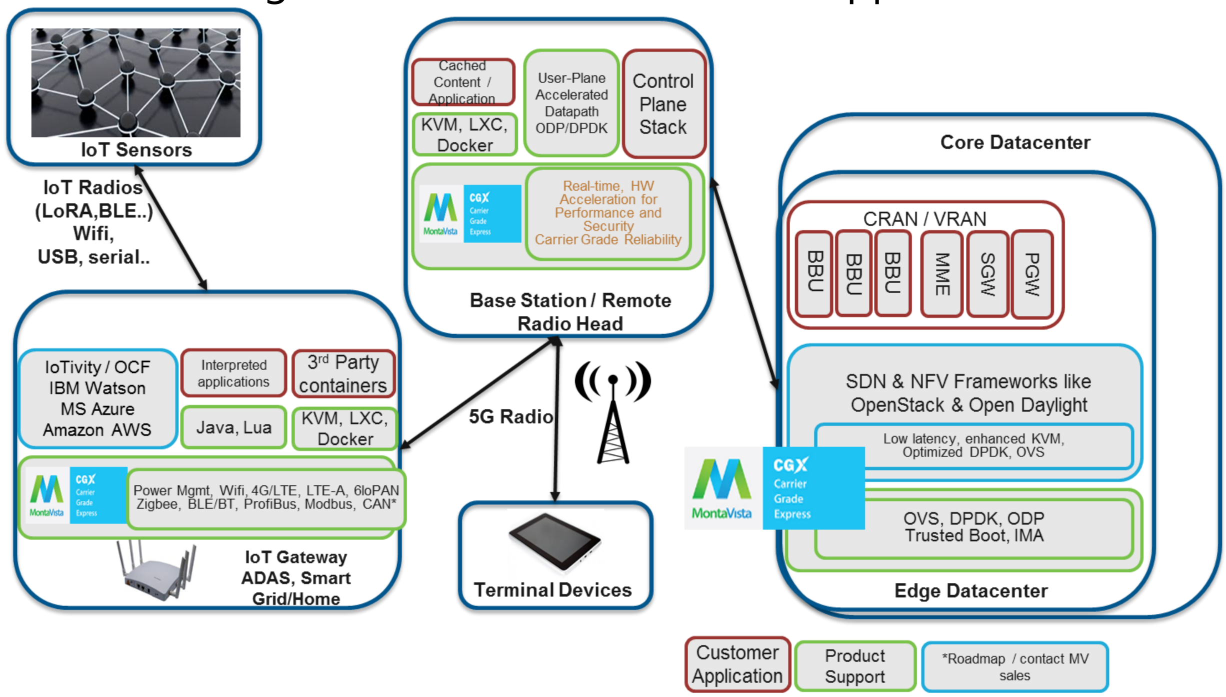

Fig 9: Case Study: 5G Architectural Design Consideration for “Network for Things”

“NETWORK” for “THINGS”

The setup shown in Fig 9 highlights a typical 5G application use case to service the need for a flexible and scalable network that be used as a reliable Access and Core Networks considerations—

- Low Latency and Bandwidth
- Content Caching
- Heterogeneous Networks / IoT Convergence
- Radio and Baseband separation
- Network Slicing
- Edge Computing, Multi-Tier Data Centers
- Control- and User Plane Separation
- Reliability
- Security

End-to-end Encryption

Communications between the things, the gateway, and the Cloud service in the 5G realm must be cryptographically secured to preserve confidentiality, integrity, and authenticity.

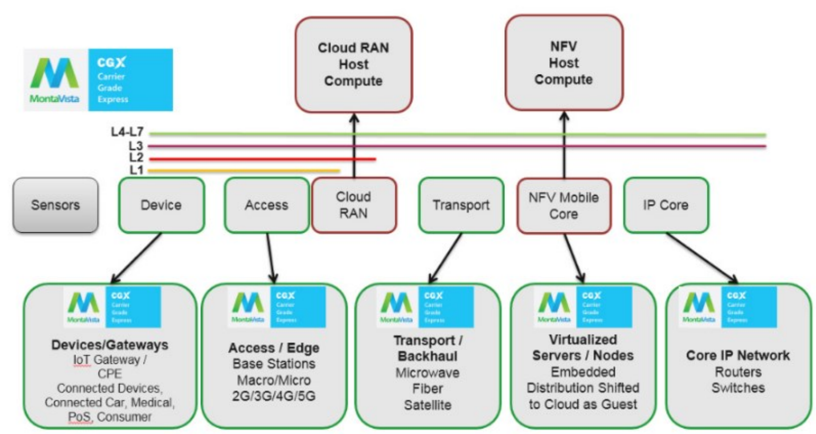

Fig 10: CGX for Next Generation Networks

Solution Brief

MontaVista Carrier Grade eXpress (CGX), meets the demands of the interconnected intelligent devices, providing application portability, dynamic configuration, field maintenance, and real-time performance in a single platform that is optimized for NFV Applications and Embedded Internet of Things (IoT) Devices.

eXpress.Connected.Everything.

About MontaVista Software

MontaVista Software, LLC, a wholly owned subsidiary of Cavium Networks (NASDAQ:CAVM) is a leader in embedded Linux commercialization. For over 15 years, MontaVista has been helping embedded developers get the most out of open source by adding commercial quality, integration, hardware en-ablement, expert support, and the resources of the MontaVista development community.

All Ways Open

MontaVista Software
2315 North First St, 4th FL
San Jose, CA 95131
Email: sales@mvista.com
Tel: +1-408-943-7451