LYNX SOFTWARE TECHNOLOGIES

LYNX MOSA.ic™

# Simplifying Software Systems Through Modular Design

LYNX MOSA.ic™ is a software development framework for building, comprehensible software systems from independent application modules, delivering the Modular Open Systems Approach (MOSA) vision. It enables developers to collapse existing development cycles to create, certify, and deploy robust, secure, mission-critical platforms. It achieves this by giving developers deeper insight.

## Customer Adoption of LYNX MOSA.ic

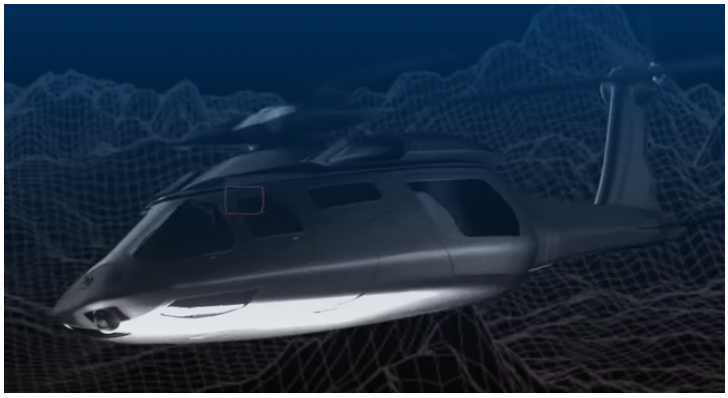
LYNX MOSA.ic is the foundational software technology at the core of a number of safety critical platforms. One specific customer has stated that our architecture saved them hundreds of thousands of lines of code and tens of millions of dollars in certification costs. Publicly disclosed programs harnessing LYNX MOSA.ic include:

### Lockheed Martin F-35

LYNX MOSA.ic is used in the Panoramic Cockpit Display (PCD) and Integrated Core Processor (ICP) avionics platforms. They have also articulated the simplicity of development flow to migrate software from a Linux environment to the LynxOS-178 real-time operating system. This multicore Intel platform drove Lynx's DO-178C DAL A certification work for LYNX MOSA.ic.

### General Atomics Gray Eagle Extended Range Uncrewed Aerial System (UAS)

This Arm-based (Xilinx MPSoC) architecture selected Lynx to create a mixed criticality system (Linux & LynxOS-178) from what was previously a monolithic stack.



### Collins Perigon

The mission computer in this system supports bare metal applications on LynxSecure, the foundational separation kernel in LYNX MOSA.ic, on three processor architectures, including Arm.

The figure to the right shows an example of LYNX MOSA.ic on a LRU processing board. The runtime architecture is comprised of two core platform technologies:

- **Control Plane** - Multicore hardware time and space separation control based on a separation kernel hypervisor
- **Data Plane** – Guest operating system and application development tools for building Real-time POSIX and ARINC applications within VMs

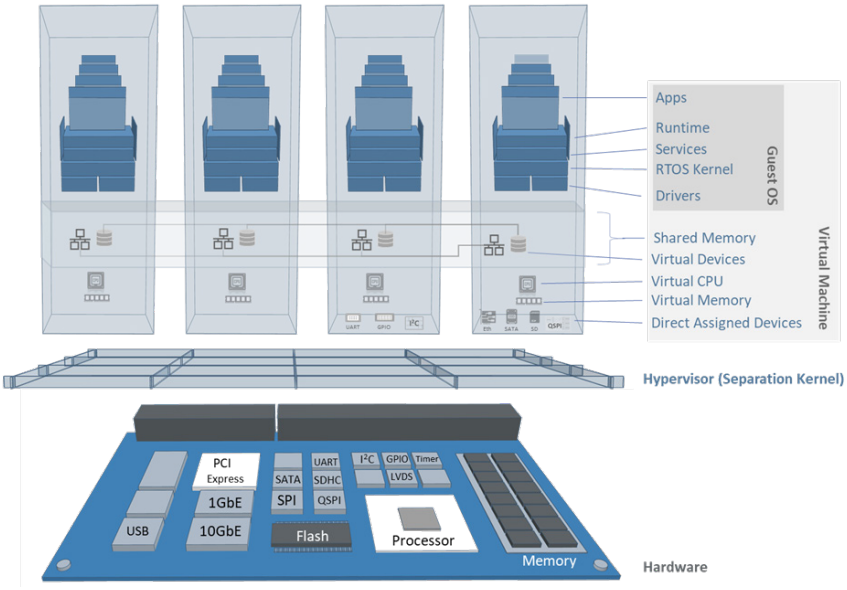



*Figure 1 - LYNX MOSA.ic™ Platform Composition Overview*

## Key capabilities include the following

### Multicore Support

The hypervisor permits flexible allocation of CPU cores to virtual machine environments – including AMP, SMP, and BMP. The picture below shows the flexibility of hosting various execution configurations simultaneously.

### Flexible Scheduling Support

When allocating a physical CPU shared by multiple virtual machines, the hypervisor has configurable options to schedule virtual machines contexts:

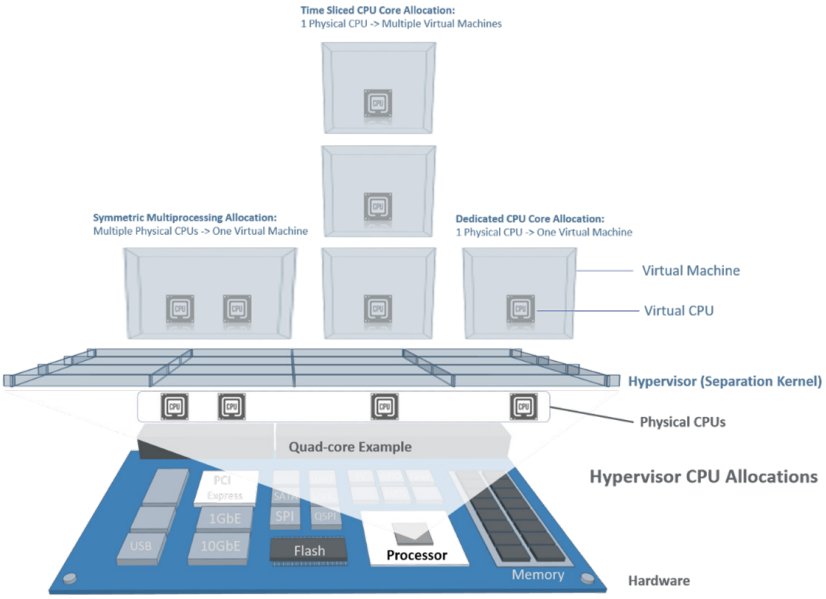



*Figure 2 - Hypervisor CPU Allocation*

- **Static Cyclic** – Each virtual machine is assigned an execution duration and cyclic period following ARINC 653 time partitioning standards
- **Static/Adaptive Cyclic** – Multiple static execution policies may be defined with different duration and period parameters per virtual machine
- **Priority Preemptive** – Virtual CPUs allocated to virtual machines are assigned priorities acting as threads to a priority pre-emptive scheduler in the hypervisor
- **Aperiodic Reservation** – Virtual CPUs may be designated for hosting sporadic workloads and are guaranteed a user defined budget of execution time per major frame of execution time
- **Co-operative "Z-Scheduling"** – Context switching capability can be delegated to a VM with the ability to control context switching with conditional and bounded execution periods granted by the hypervisor

## Virtual Device Emulation

The hypervisor can allocate peripheral interfaces to virtual machines that are not directly mapped to physical hardware. Virtual devices are commonly used to facilitate inter-VM communication and for multiplexing the use of physical devices. Device virtualization is a foundational feature that allows RTOS runtimes to exercise POSIX file system and networking features across all CPU cores.

## Lynx Partners Providing Complementary Technology

We value our relationships with all of our partners. Below we highlight speciifc partners who have product SKUs whose technology is delivered as part of LYNX MOSA.ic. Systems continue to increase in complexity at a time when there has never been a stronger focus on improving development timescales. Lynx works closely with a diverse set of partners (see below) to prove our technology works optimally with LYNX MOSA.ic. Lynx is increasingly focused on this, with MOSA.ic being the Integration Center of future systems. Consequently we are reselling proven third party technology as part of LYNX MOSA.ic. Current technology includes:

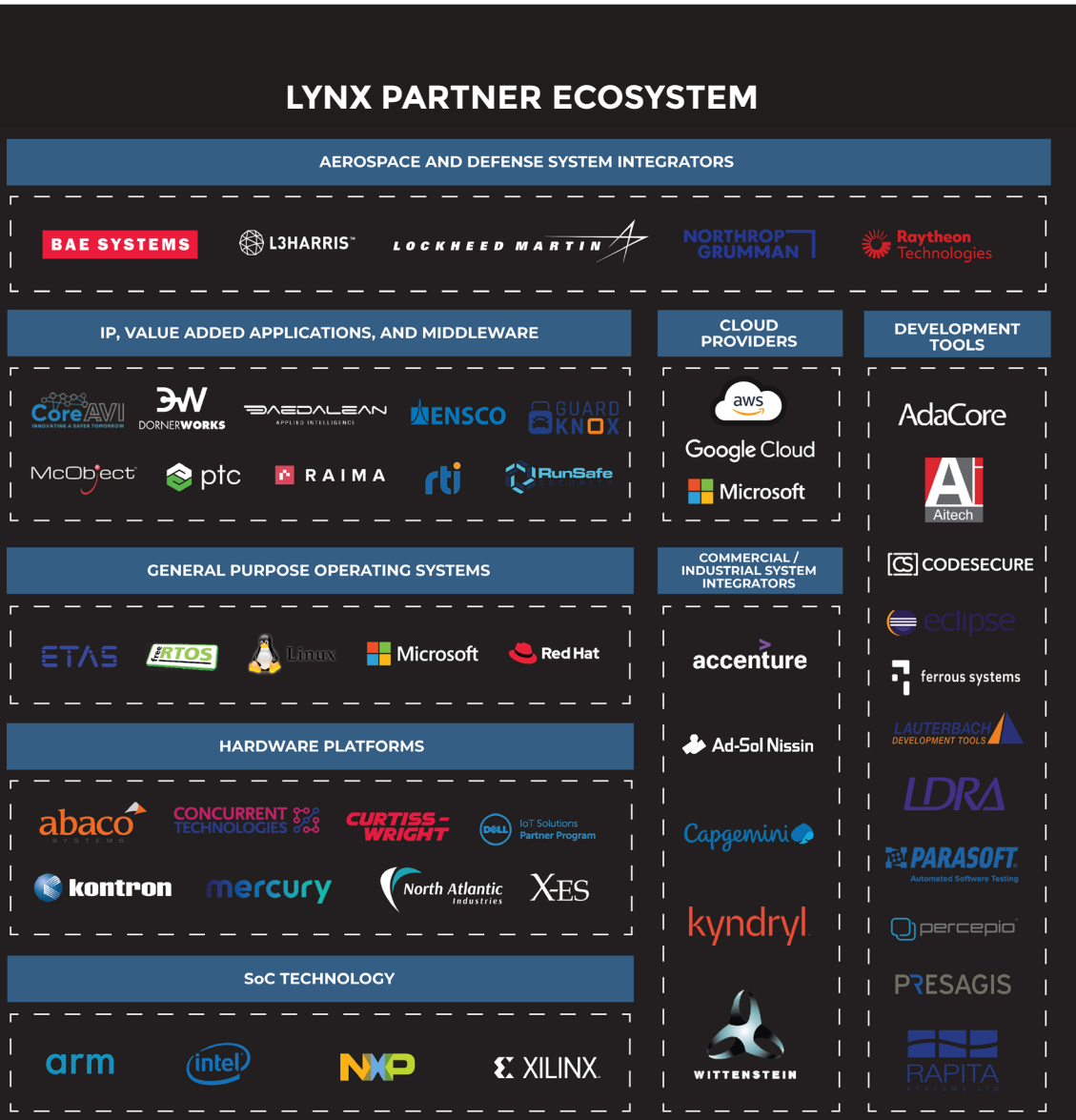
### RunSafe Security

In partnership with RunSafe Security, we bundle their technology as part of our LYNX MOSA.ic product, protecting it against 70% of the most common vulnerabilities with no developer impact. This RunSafe protection applies to the build root Linux operating system, applications, and customer software.

### Percepio

The rising system complexity coupled with the intense pressure on improving development cycles is such that Lynx felt it needed to improve the level of system observability available with LYNX MOSA.ic. A review of make versus buy options led us to commit to expanding our Spyker product family by partnering with Percepio. Lynx is licensing Percepio technology to:

- Re-sell TraceAlyzer™ product as-is
- Deliver TraceAlyzer as part of LYNX MOSA.ic, branded as “SpyKer TZ”



## LYNX MOSA.ic Technical Details

On a traditional RTOS platform, all hardware control, real-time scheduling, security, multi-media, and application services are integrated into a common stack, servicing all applications on all CPU cores, LYNX MOSA.ic allows system architects to subdivide systems into smaller independent stacks which include only the runtime dependencies needed per application.

The benefits of stack separation yield high program values through the reduction of software complexity including:

- Promoting comprehensible traceable architectures
- Improved formalism of system composition to aid security and safety analysis
- Reducing time to debug
- Increasing the speed of system integration

This approach greatly reduces development costs while the critical aspects of safety and security are robustly supported at the CPU control level.

RTOS applications are built as virtual machines partitioned by a separation kernel hypervisor. Use of LynxSecure, Lynx's separation kernel, addresses the complexity challenges of building multi-core systems that must conform to reference architecture standards (Ex: IMA, FACE), while providing greater platform robustness and application portability properties over conventional RTOS design.

LynxSecure's primary role is to partition, allocate, and arbitrate access to physical resources. It is developed according to DO-178C DAL A software development standards and supports ARINC 653 architecture requirements. The lightweight hypervisor strictly serves as a hardware control plane for the overall system. The hypervisor does not provide application or data services. All resource allocation and policy enforcement capabilities provided by the hypervisor apply to the definition of virtual machines and their assigned resource and access control permissions. This establishes a hardware enforced software architecture for a given system configuration.

Lynx's CDK includes support to build the following guest software types:

- **Bare-metal (LSA)** – Main C program, supported by 64-bit C libraries and GCC toolchain to build simple applications with minimal runtime complexity.
- **RTOS (LynxOS-178)** – Safety certified, preemptive hard real-time operating system. Provides multi-process, multi-threaded POSIX and ARINC runtime services and hard real-time scheduling primitives.
- **Unikernel (LynxElement)** - A consolidated multithreaded, single process RTOS environment that removes redundant architectural constructs shared between the RTOS guest kernel and separation kernels. Unikernel applications are provided standard ARINC and POSIX libraries and a fully featured stack.
- **Linux (Buildroot)** – Lightweight embedded Linux toolchain. Provides broad peripheral support using native Linux kernel.org kernel images. Buildroot provides customization tools to select kernel modules and application packages to include in RAM disk and boot images.

A virtual machine may also host guest software 3rd party binary OS distributions for example: Windows, Red Hat, Ubuntu etc.