# LYNX MOSA.ic™

Lynx MOSA.icTM is a software development framework for rapidly building comprehensible software systems out of independent application modules and delivering the vision of the Modular Open Systems Approach (MOSA). Lynx MOSA.icTM gives developers deeper insight and increased control over how applications are realized onto modern CPUs and introduces a new perspective to application development that simplifies the creation, certification, and maintenance of inherently complex software systems.

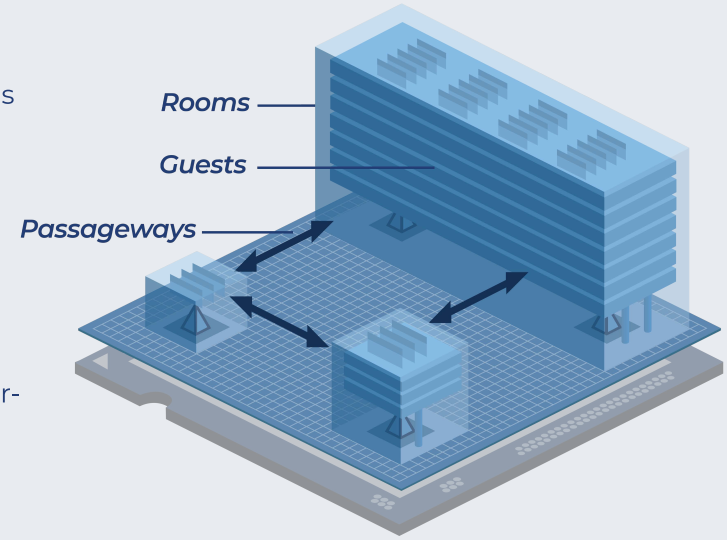
## Product Overview

The Lynx MOSA.icTM framework enables the construction of robust, comprehensible systems where hardware is statically partitioned into rooms and passageways to host guests:

*Rooms* are collections of hardware resources created by the Lynx MOSA.icTM Programmable Processor Partitioning System.

*Passageways* are explicit point-to-point memory regions that link rooms together via standard inter-process communication (IPC) interfaces.

*Guests* are Lynx, legacy, competitor, or partner-provided environments for hosting applications.



Lynx MOSA.icTM provides tools to define rooms and passageways, along with policies to enforce least-privilege access when connecting guests together.

The Programmable Processor Partitioning System of Lynx MOSA.icTM is provided by the LynxSecure® separation kernel hypervisor. First released as a standalone product in 2006, LynxSecure® is a mature and widely deployed product. Following initial deployments in high assurance Government applications, the product has been deployed in volume in mission-critical commercial and industrial applications, and across market sectors including automotive, avionics, industrial IoT, robotics, transportation, and UAV's.

At its core, Lynx MOSA.icTM enables simpler software systems by harnessing CPU virtualization to partition systems into components. Simplicity is achieved by subdividing the hardware into smaller compute platforms and by eliminating the need for an operating system (OS) or hypervisor to act as a global resource manager. For example, a modern quad-core system on chip (SoC) could be subdivided into four mono-core compute platforms. The SMP RTOS scheduling processes across the four cores could be eliminated, replaced instead with four bare-metal applications.

This approach removes as much complexity as possible between application interfaces and hardware. If an application requires the use of a filesystem, network stack, RTOS or OS then it may be used, but the developer is not forced to include software unnecessary to their design.

## Open Architecture

The objective of the U.S. Department of Defense's (DoD) Modular Open Systems Approach (MOSA)[1] is to design systems with highly cohesive, loosely coupled, and severable modules that can be competitively sourced from independent suppliers. As a software development framework for rapidly building comprehensible software systems out of independent application modules, Lynx MOSA.ic$^{TM}$ is designed specifically to deliver on the open architecture vision of MOSA.

In traditional platform architectures, applications are compiled against APIs provided by an OS and run as processes on the OS. The means by which multiple applications are realized onto the CPU are abstracted away by the OS. Lynx MOSA.ic$^{TM}$ provides a Modular Development Framework that gives system integrators and platform suppliers precise control over how applications are realized onto the CPU. Unlike OSs, Lynx MOSA.ic$^{TM}$ provides system integrators clear understanding of the capabilities of their CPU and visibility of its capacity as software loads are added to it.

The Lynx MOSA.ic$^{TM}$ framework enables platform suppliers and system integrators to partition hardware into Rooms and Passageways. Application suppliers have the freedom to use their preferred application development tools and runtime environments for hosted applications. Lynx MOSA.ic$^{TM}$ is designed to deliver the five primary benefits of MOSA as anticipated by the DoD:

1. Enhance competition. In contrast with a fixed-function "black box" provided by a single system integrator, Lynx MOSA.ic$^{TM}$ is an enabler of open architectures in which complex software systems can be decomposed into components that can be competitively sourced.

2. Facilitate technology refresh. Lynx MOSA.ic$^{TM}$ enables the delivery of new capabilities or replacement technology without changing all components in the entire system.

3. Incorporate innovation. The precise control over hardware resources enabled by Lynx MOSA.ic$^{TM}$ provides system integrators with the flexibility to configure and reconfigure available resources to enhance capability wherever possible.

4. Enable cost savings/cost avoidance. Lynx MOSA.ic$^{TM}$ encourages the development and reuse of components, reducing the effort, cost, and risk of building and maintaining complex software systems through the entire lifecycle. Furthermore, as application suppliers are free to use their preferred application development tools and runtime environments, Lynx MOSA.ic$^{TM}$ does not impose additional taxes or tolls to populate rooms with guests.

5. Improve interoperability. The system module integration tools included in Lynx MOSA.ic$^{TM}$ connect legacy, competitor, or partner-provided guests together, and can move guests from room to room in current and future designs.

[1] https://www.acq.osd.mil/se/initiatives/init_mosa.html

In many industries, standardization activities have led to the development of open architectures – based on a rich set of standard APIs, components and services – as a means of dealing with the growing challenge of software complexity. Examples include:

- Automotive – AUTOSAR Adaptive Platform
- Industrial – Industrial Internet Reference Architecture (IIRA) and Reference Architecture Model Industrie 4.0 (RAMI 4.0)
- Military Avionics – Open Group Future Airborne Capability Environment (FACE™) Consortium
- Military Vehicles – NATO Generic Vehicle Architecture (NGVA)
- UAVs – Joint Architecture for Unmanned Systems (JAUS) and Unmanned Systems (UxS) Control Segment (UCS) Architecture

When the APIs, components, and services of these open architectures are realized on traditional platforms, the centralized resource management model common to OS and hypervisor designs can result in unintended coupling between modules. This coupling can be exploited to create vendor lock-in at all levels of the supply chain; from platform suppliers, system integrators and application suppliers, to the vendor of the platform itself. In giving architects and developers increased control over how independent application modules are realized onto modern CPUs, Lynx MOSA.ic™ avoids unintended coupling between modules and enables open architecture to be delivered in practice.

## Guest Flexibility

As rooms can host any software load, Lynx MOSA.ic™ provides a significant degree of flexibility in the type of guests that can be deployed on the platform, subject to the constraints of the CPU architecture. To enable the rapid creation of partitioned systems, Lynx MOSA.ic™ includes the following cross development kits (CDKs) for guests of varying size, assurance, and complexity:

- Lynx Simple Application (LSA) – A "bare metal" guest environment for simple applications.
- LynxOS-178 – A DO-178C DAL A certified UNIX-like Real Time Operating System guest environment for real-time and safety critical applications using ARINC, FACE or POSIX APIs.
- Buildroot Linux – An embedded Linux toolchain for general purpose Linux applications.

As system architects have precise control over how applications are realized on the CPU, it is possible to construct mixed criticality systems which span the entire spectrum from hard real-time deterministic guests to enterprise class Linux and Windows guests. At the high integrity end of the spectrum, Ada applications can be deployed in an LSA guest running on a zero footprint (ZFP) runtime. On the low integrity side, applications including databases, digital twins, IoT frameworks and platforms, digital mapping, and simulations may be deployed in an enterprise-class OS guest such as Linux or Windows.

## Virtualization

Lynx MOSA.ic™ is designed to support 64-bit multi-core processors (MCPs) that provide hardware support for virtualization. The first release of Lynx MOSA.ic™ supports Arm (Armv8 processors from NXP and Xilinx) and Intel (Intel® Core™ i5, Intel® Core™ i7, Intel® Xeon and Intel® Atom) processors. Support for NXP T-Series QorIQ® Processors Based on Power Architecture Technology is in development.

Lynx MOSA.ic™ is compatible with a wide range of commercial off the shelf (COTS) hardware platforms from leading suppliers including Abaco, Aitech, Concurrent Technologies, Curtiss-Wright, Kontron, Mercury, Teledyne e2v and X-ES. It is normally a straightforward process to validate Lynx MOSA.ic™ on any new COTS or custom hardware platform designed around a supported processor.

Lynx MOSA.ic™ is the first software development framework that unlocks the full potential of hardware virtualization, enabling system integrators to leverage high-performance MCPs to significantly reduce SWaP (size weight and power) by consolidating multiple Line Replaceable Units (LRUs) onto a single platform.

## Assurance – Safety

Lynx MOSA.ic™ provides a clearer path to multi-core safety certification by forgoing the traditionally inherited complexities of a centralized resource management model common to OS and hypervisor designs. The stack of software implementing the resource management is a liability; not only does it add significant cost to the certification effort, but it over-complicates robust partitioning and contributes to the multi-core interference problem.

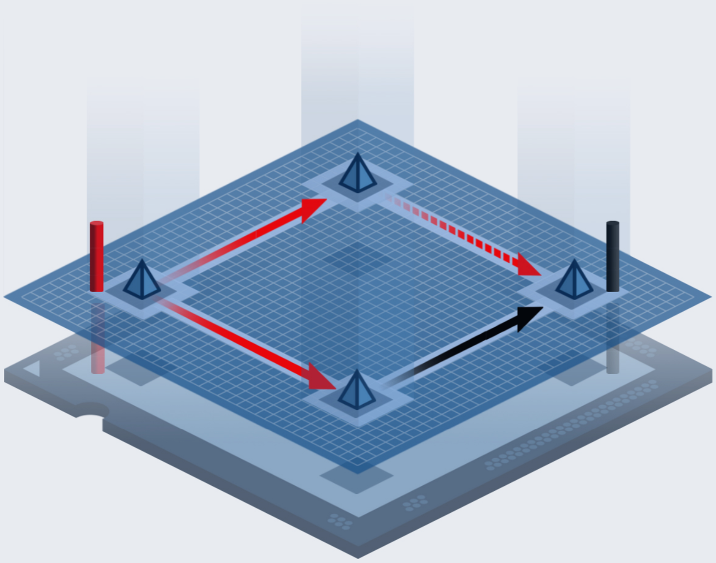
Lynx MOSA.ic™'s assurance strategy stands in marked contrast to strategies that rely on reasoning about software calls, loops, and conditions in that it rests on MMU translation results verifiable at physical memory locations. Ironically, every argument about software calls, loops, and conditionals will also eventually resolve to assumptions about hardware, such that said arguments arrive at the same point, but at substantially greater cost. In short, rather than being forced to deal with the complexities of centralized resource management, the system integrator has precise control over the partitioning of hardware resources and the resulting impact on hardware interference.

Moving safety critical applications to multi-core processors practically necessitates the adoption of Integrated Modular Avionics (IMA). The separation provided by the Lynx MOSA.ic™ Programmable Processor Partitioning System is verifiably robust and suitable for IMA incremental certification and mixed assurance use cases. The benefits of IMA include reduced certification efforts due to incremental acceptance, component acceptance, and reuse of accepted components.

These cost-benefits can accrue over many years and over multiple subsystems, components, suppliers, and aircraft systems.

### Assurance – Security

Lynx MOSA.ic™ is founded on the LynxSecure® Programmable Processor Partitioning System. Inspired by the Rushby Separation Kernel, LynxSecure® harnesses CPU virtualization to simplify resource control abstraction layers into a distributed model in contrast to traditional OS-based central resource management models. LynxSecure® isolates computing resources into rooms – independent distributed environments which are uniquely capable of managing themselves. Its distributed, least privilege design and trusted hardware control abstraction layers minimize attack vectors and make Lynx MOSA.ic™ naturally resilient to advanced persistent threats and side channels.



## Sustainment

Lynx MOSA.ic™ encourages the development and reuse of components. Enabling the delivery of new capabilities without touching other components in the system, Lynx MOSA.ic™ minimizes the cost, risk, and effort of technology refresh programs. The reuse of binary components is feasible, provided that silicon vendors continue to embrace backwards compatibility across hardware generations.

## Summary

Lynx MOSA.ic™ is a software development framework for rapidly building comprehensible software systems out of independent application modules. Lynx MOSA.ic™ enables simpler software systems by harnessing CPU virtualization to partition systems into components. Simplicity is achieved by subdividing the hardware into smaller compute platforms and by eliminating the need for an OS or hypervisor to act as a global resource manager.

Lynx MOSA.ic™ is designed specifically to deliver on the open architecture vision of the U.S. DoD Modular Open Systems Approach (MOSA), in which complex software systems can be decomposed into components that can be competitively sourced from independent vendors. Lynx MOSA.ic™ avoids unintended coupling between components and minimizes the scope for any actor in the supply chain to exploit vendor lock-in.

Lynx MOSA.ic$^{TM}$ provides a significant degree of flexibility in the type of components that can be deployed on the platform and includes development kits for “bare metal”, hard real-time, safety critical, and embedded Linux applications. High integrity applications can happily coexist alongside low integrity applications such as digital twins deployed in an enterprise-class OS guest such as Linux or Windows.

Avoiding the complexities of centralized resource management and giving the system architect precise control over the partitioning of hardware resources, Lynx MOSA.ic$^{TM}$ provides a clearer path to multi-core safety certification. Furthermore, its distributed, least privilege design minimizes attack vectors and makes Lynx MOSA.ic$^{TM}$ naturally resilient to advanced persistent threats and side channels.

Finally, Lynx MOSA.ic$^{TM}$ encourages the development and reuse of components, reducing the effort, cost, and risk of building and maintaining complex software systems through the entire lifecycle.