Infosec OS.1:
High-Assurance Architecture on Synrc Hypervision

Namdak Tonpa
Synrc Research

August 2026

Abstract

Infosec OS.1 is a high-assurance operating system built on Synrc Hypervision, a hybrid Type-1 architecture that hosts an unmodified Erlang/OTP BEAM runtime on the formally verified seL4 microkernel while confining device drivers to a minimal Alpine Linux guest. The system is constructed entirely statically under the Microkit framework. This article presents the architecture, derives technical requirements according to ISO/IEC/IEEE 29148:2018, maps selected NIST SP 800-53 control families onto seL4 capabilities, and shows how elevating key POSIX applications (BE, TV, SH, SKYNET) into dedicated Protection Domains further strengthens isolation, least privilege, and auditability.

Contents

1 Introduction

Erlang/OTP and its BEAM virtual machine provide robust concurrency, supervision trees, and soft real-time behaviour that are attractive for high-availability and distributed systems. These strengths, however, traditionally rest on a large general-purpose operating system whose trusted computing base (TCB) undermines strong security and certification arguments.

Formal isolation has advanced substantially with the seL4 microkernel, which offers machine-checked proofs of functional correctness, integrity, and (for selected configurations) information-flow security. Running a full Linux guest under seL4 merely relocates the TCB problem; pure unikernel approaches often force substantial rewriting of application runtimes or sacrifice device support.

Synrc Hypervision resolves this tension by combining three elements:

- the formally verified seL4 microkernel and its Microkit framework for static, capability-based system construction;
- Synrc, a minimal host that supplies only the approximately fifty Linux-compatible system calls required by a musl-linked, unmodified OTP BEAM;
- a carefully confined Alpine Linux virtual machine that supplies real device drivers, communicating with the BEAM domain solely through capability-mediated shared-memory channels defined by the seL4 Device Driver Framework (sDDF).

The resulting architecture keeps the TCB under the BEAM extremely small (seL4 + Synrc) while still permitting practical use of complex hardware. All isolation boundaries, scheduling parameters, and device ownership are fixed at image-construction time, yielding a static system amenable to analysis and aligned with selected families of NIST SP 800-53 controls.

Infosec OS.1 extends this foundation into a coherent security-oriented stack that also includes a NuttX-based POSIX real-time layer (RT), a drone-swarm coordination protocol (SKYNET), and a set of zero-dependency Infosec tools (Binary Editor, Terminal Vision, and a formally modelled POSIX shell). The present article describes the overall architecture, formalises the technical requirements, maps the relevant security controls, and demonstrates the security gains obtained by placing the principal POSIX applications into their own Protection Domains.

2 Architecture of Infosec OS.1

Infosec OS.1 is organised as a static collection of seL4 Protection Domains (PDs) whose memory regions, communication channels, interrupt bindings, and Mixed-Criticality Scheduling (MCS) contexts are declared in a Microkit system description and materialised at build time. No protection domain or channel can be created dynamically at runtime.

The principal components are:

Component	Type	Responsibility
seL4	Microkernel	Isolation, capabilities, MCS scheduling, IRQ routing
Microkit	Framework	Static system description and capability layout
Synrc	PD	Minimal syscall surface and host for BEAM
BEAM / OTP	Runtime	Erlang/Elixir processes, schedulers, supervision
VMM (libvmm)	PD	Creates and manages the Alpine Linux guest
Alpine Linux	Guest VM	Real device drivers (network, block, ...)
sDDF	Protocol	Shared-memory rings and notifications
Console PD	PD	Exclusive ownership of UART / virtio-console
Monitor PD	PD	Metrics, health, and audit events
Storage / Network	PD	Block and Ethernet access (native or mediated)
BE, TV, SH, SKYNET	PD	Isolated POSIX applications (see §2.3)

All authority is expressed by seL4 capabilities granted at system construction. There is no ambient authority. Scheduling follows the seL4 MCS model; typical priority ordering (highest to lowest) places the Console PD first, followed by Synrc/BEAM, the Monitor, the VMM/Alpine domain, and background domains. Core affinity and budget/period parameters are supplied via compile-time flags.

2.1 Technical Requirements (ISO/IEC/IEEE 29148:2018)

Technical requirements are derived and organised according to the principles of ISO/IEC/IEEE 29148:2018 (Systems and software engineering — Life cycle processes — Requirements engineering). They are stated as necessary properties of any conforming implementation of Infosec OS.1.

TR-1 Static construction. All Protection Domains, memory regions (with explicit read/write/execute rights), channels, interrupt bindings, and MCS scheduling contexts shall be declared in a Microkit (or equivalent) system description and fixed at image-construction time. Dynamic creation of domains or channels is prohibited.

TR-2 Minimal syscall surface. The host for the primary application runtime (Synrc) shall implement only the Linux-compatible system calls required by a musl-linked OTP binary (memory management, threading and futex family, time, basic file descriptors, and an in-memory VFS). The host shall never own real device registers.

TR-3 Capability-mediated I/O. All device access shall occur exclusively through sDDF-style shared-memory rings and notifications, or through

native seL4 driver PDs. The Alpine guest may own devices only under the control of a confined VMM PD.

TR-4 Exclusive device ownership. The platform console (UART or virtio-console) shall be owned exclusively by a dedicated Console PD. Storage and network devices shall be assigned either to native driver PDs or to the Alpine guest; they shall never be shared ambiently.

TR-5 MCS scheduling and affinity. Each PD shall receive an explicit budget, period, and core affinity. Priority ordering and resource parameters shall be fixed at build time.

TR-6 Minimal guest images. Any Linux (or NetBSD Rump) guest shall be constructed from a minimal, musl-based root filesystem. Only devices explicitly assigned to the guest shall appear in its device tree.

TR-7 Isolated POSIX applications. The Binary Editor (BE), Terminal Vision (TV), formal POSIX shell (SH), and SKYNET swarm coordinator shall each execute as a distinct Protection Domain with least-privilege capabilities (see §2.3).

TR-8 Zero-dependency userland tools. BE, TV, and SH shall be implemented in ANSI C99 using only POSIX and termios interfaces; external libraries (ncurses, S-Lang, etc.) are prohibited.

TR-9 Formal verification preservation. The seL4 configuration used by Infosec OS.1 shall retain a verified status (functional correctness and integrity). The SH PD shall be derived from a formal model where claimed.

TR-10 Measurable static image. The final system image shall support measurement and attestation of its static configuration and code.

These requirements are both necessary precursors of the architecture and directly matched by the current Synrc Hypervision design and its Infosec OS.1 extensions.

2.2 NIST SP 800-53 Controls

Selected control families of NIST SP 800-53 are mapped onto seL4 capabilities and the static compile-time configuration as follows.

AC — Access Control
Least privilege and information-flow enforcement are realised by the explicit capability layout (AC-3, AC-4, AC-6). Separation of duties is obtained by placing distinct functions into distinct PDs.

SC — System and Communications Protection
Boundary protection (SC-7) is provided by seL4 Protection Domains and stage-2 translation for the Alpine guest. Information-flow control follows from the seL4 proofs. All inter-domain communication occurs through declared channels only.

CM — Configuration Management
The Microkit system description together with compile-time flags constitutes the baseline configuration (CM-2, CM-3, CM-6, CM-7). No runtime reconfiguration of the capability space is possible.

AU — Audit and Accountability
The Monitor PD collects health, performance, and audit-style events via fixed shared metrics regions and notifications, supporting AU-2, AU-3, AU-6, and AU-12. When MONITOR=0 the domain may be omitted, further reducing the TCB.

SI — System and Information Integrity
Software integrity (SI-7) is supported by the static, measurable image and by the formal verification of the seL4 core.

CP — Contingency Planning
Fault isolation is obtained from both BEAM supervision trees and seL4 spatial/temporal isolation. MCS budgets prevent resource-exhaustion interference between domains.

Additional high-assurance properties beyond the NIST catalogue include the absence of ambient authority, deterministic resource allocation, and the ability to omit non-essential PDs from production images.

2.3 POSIX Applications as Protection Domains

Elevating BE, TV, SH, and SKYNET into their own Protection Domains constitutes a deliberate isolation upgrade that applies the same static capability model already used for Synrc, the VMM, and the device PDs.

- **Spatial isolation.** Each tool receives a private address space. A compromise in the Binary Editor (while processing untrusted binaries) cannot reach the memory of the shell, the text editor, the BEAM runtime, or the SKYNET control plane.
- **Least-privilege capability surfaces.** Capabilities are granted individually:
  - BE obtains mediated storage channels and a console channel;
  - TV obtains only a console channel;
  - SH obtains a console channel and a tightly restricted filesystem view;
  - SKYNET obtains the network/radio channels required by the swarm protocol and a high-priority MCS context.

No domain receives ambient authority.

- **Fault and temporal isolation.** Crashes, infinite loops, or resource exhaustion remain confined. Independent MCS budgets allow SKYNET to run with real-time parameters while the interactive tools receive lower-priority contexts.
- **Containment of untrusted input.** Malicious binaries (BE), scripts (SH), free-form text (TV), and external swarm messages (SKYNET) are processed inside dedicated domains whose outward channels are minimised.
- **Improved auditability.** The Monitor PD can receive distinct notification channels from each application PD, producing attributable events rather than mixed userland logs.

Console multiplexing is performed by the existing Console PD, which continues to own the physical UART exclusively and exposes virtual character streams. Storage access is mediated through the Storage PD or a small helper domain. The NuttX-based real-time layer used by SKYNET may itself reside inside the SKYNET PD or communicate with it via declared channels.

The net result is a system that more closely approximates a high-assurance separation kernel: a small number of carefully interfaced application domains running on a formally verified microkernel with a minimal TCB under the primary workload.

3 Conclusion

Infosec OS.1 demonstrates that a production-grade BEAM application can operate with a dramatically reduced trusted computing base while retaining practical device support and a coherent suite of Infosec and real-time components. The architecture rests on three pillars: the formally verified seL4 microkernel, a purpose-built minimal host (Synrc), and capability-mediated confinement of drivers inside an Alpine guest.

Technical requirements have been stated in accordance with ISO/IEC/IEEE 29148:2018 and are directly realised by the static Microkit construction. Selected NIST SP 800-53 control families map cleanly onto seL4 capabilities and compile-time configuration. Elevating the principal POSIX applications (BE, TV, SH, SKYNET) into dedicated Protection Domains further raises the assurance level by enforcing spatial isolation, least privilege, fault containment, and clearer audit trails—without abandoning the static model or introducing dynamic authority.

The design therefore offers a practical route toward higher-assurance, certifiable distributed and embedded systems that combine the productivity of Erlang/OTP with the isolation guarantees of a separation kernel.