CYRAY

MOBULA

WHITE PAPER

Mobula White Paper

DOCUMENT	**1.0.1**
ISSUED	22 August 2026
DOCUMENT	CYR-MOB-WP-1.0.1

Final. All claims approved.

Contents

1. The problem in modern SOC operations

Security operations centres do not fail for lack of tooling. They fail because the work that makes a SIEM actually detect things — writing rules, mapping fields, tuning out noise, keeping content current as the estate changes — is continuous, specialist, and almost always understaffed. A detection rule is not an artifact you install once; it is a living object that must stay correct, enriched, non-noisy and growing as attackers, log sources and the business all move underneath it.

Four failure patterns recur in practice:

- **The rule-authoring burden lands on the customer.** Most platforms hand over an engine and leave the detection content — the part that requires rare expertise — to the buyer's own team. The result is thin coverage, stale rules, and tuning debt that compounds.
- **Collection silently degrades.** Log collectors stop, checkpoints drift, feeds duplicate or gap — and nothing announces it. A detection estate built on a feed that quietly died is worse than no estate at all, because it manufactures false confidence.
- **Parsing fragments.** When field mapping is re-implemented per collector, per site, per integration, the same event means different things in different places, and every downstream rule inherits the inconsistency.
- **Air-gapped and restricted environments are underserved.** Modern platforms assume cloud connectivity — for updates, for telemetry, and increasingly for AI assistance. Sites that cannot connect are left with a frozen product and none of the intelligence.

Underneath all four is a single structural question: how much of security operations can be made deterministic — guaranteed by code and verifiable state — rather than resting on individual heroics and manual supervision? Mobula is built as an answer to that question.

2. What Mobula is

Mobula is CyRay's **AI-orchestrated Security Operations Platform**: a managed-detection platform family in which detection content is engineered centrally by experts and machines, and delivered continuously to every customer environment.

The customer proposition is direct: customers get continuously improving protection without ever having to write or tune a rule themselves. A centrally maintained, expert-vetted detection library — mapped to MITRE ATT&CK and quality-gated before release — is packaged, versioned and distributed automatically to each environment, and kept current as the threat landscape and the customer's own telemetry evolve.

Behind that proposition sits an autonomous detection-engineering factory: roughly fifteen years of enterprise-SIEM detection-engineering expertise encoded into executable pipelines, checks and content, operating under a design rule that human supervision must be made structurally unnecessary —

enforced by code, not by promises. Where an AI agent acts, its actions are attributed, gated and verifiable; where a process must be reliable, it is a pipeline with machine-checkable state, not a runbook someone remembers to follow.

The ecosystem spans the full operational chain: a customer-facing Mobula platform; a central content engine (Mobula Master); a workflow hub (MobulaHub); API log collectors (CyRay Trawler); endpoint sensors (HASH Ocean and the Sysmon Distribution Kit); an autonomous detection-engineering agent (Nemo); and an on-premises, air-gapped AI reasoning layer (`mobula-ai`).

3. Architecture overview

3.1 Hub and spoke, closed loop

Mobula is a hub-and-spoke content architecture. Mobula Master, the central hub, is the single authority for detection content: it builds, reviews, versions, packages and distributes detection rules, exclusions, playbooks, MITRE mappings and connector parsers to every customer platform. Customer environments consume content kits and knowledge packs — and return the raw operational signal: alerts, analyst feedback, exclusions. That signal becomes tuning and training data at the centre, closing a continuous-improvement loop in which every customer's operational experience improves the protection of all.

3.2 Versioned, attributed content

Every detection rule carries a three-part version (Major.Minor.Build). Every change writes a changelog entry, increments the build number, and flags the rule and every kit that carries it for update. All writers route through one canonical service — there is no side door to content. The consequence for the customer is auditability: for any rule in any environment, the platform can answer what version is running, what changed, when, and by whom — including when the "whom" is the platform's own autonomous agent.

3.3 Constitutional data boundaries

Two data rules are architectural constants, enforced in code rather than stated in policy:

- **The master content plane carries zero customer information.** The central authoring environment where detection content is engineered holds no customer data, verified by a standing read-only discovery check.
- **Tenant isolation at the AI boundary.** No customer-identifying data — names, email addresses, hostnames, IP addresses, analyst notes — may reach the AI training, knowledge-pack or retrieval layers without a sanitization pass.

Together these mean a customer's telemetry sharpens the shared detection library without the customer's identity or environment details ever leaving their boundary.

3.4 Collection you can trust

The collection layer is governed by a small set of hard laws:

- **Transport is separated from parsing.** Collectors move raw events; all parsing and field mapping live in exactly one place — the connector parser, the single field authority for every source. Collection can never fork parsing.
- **A new log source is configuration, not engineering.** Vendor integrations are thin, declarative modules on a common framework; adding a vendor never touches the framework.
- **No gaps, no duplicates.** A collector's checkpoint advances only after events are durably handed to intake. Three checkpoint modes cover event feeds (high-watermark), file-granularity APIs (file-manifest) and state-only sources (state-snapshot), plus an event-stream shape for persistent feeds.
- **No silent collector.** Every collector always emits a heartbeat and its checkpoint age. A feed that stops is a visible fact, never a quiet absence.

3.5 Built for scale

The platform follows one scale law throughout: answers are computed at write time and stored; reads are indexed lookups. No screen, report or API call recomputes an answer that the write path already knows. The architecture is designed to hold at one hundred to one thousand times current volume without redesign.

The central stack runs a NestJS backend on Google Cloud Run with a web frontend on Firebase Hosting, PostgreSQL as the single database engine, and file-based, reviewed SQL migrations as the only schema channel.

4. Module capabilities

Module	What it does
Mobula platform	The customer-facing runtime: installs content kits and knowledge packs; the origin of alerts, analyst feedback, exclusions and investigation narratives.
Mobula Master	The content authority and sync hub: builds, reviews, versions, packages and distributes detection rules, exclusions, playbooks, MITRE mappings and connector parsers to every customer platform.
MobulaHub	The workflow and ticketing hub: owns operational work items across the estate.
CyRay Trawler	API log collectors: authenticate to vendor APIs, pull events, checkpoint durably, and hand raw events to connector intake. Transport only — never parsing.
Trawler vendor modules	Declarative per-vendor integration modules on the Trawler framework — a new log source is a small configuration module, not framework work.
Connector-Parsers	The single parser truth: all field mapping and parsing for every source, one parser family per source, the only field authority in the platform.
Content kits (ConKits)	Packaged, versioned detection and connector content, distributed automatically to customer platforms on a regular cycle.

Module	What it does
HASH Ocean	Windows file-integrity and executable-inventory sensor: incremental MD5/SHA-1/SHA-256 hashing, Authenticode signer capture, NEW/CHANGED/UNCHANGED classification, feeding the SIEM via the Windows Event Log.
Sysmon Distribution Kit	Reproducible Sysmon rollout across GPO, SCCM/MECM, Intune and standalone channels, shipped with branded installation, verification, upgrade and removal guides.
Nemo	The autonomous detection-engineering agent: maintains SIEM detection rules end to end, with every write attributed to it by name, carrying a run identifier, and gated fail-loud in code.
`mobula-ai` **(air-gap AI)**	The on-premises reasoning layer for disconnected sites: alert summarization, exclusion suggestion, threat narratives, rule explanation and triage pre-processing — running entirely inside the customer boundary.

Three of these deserve a closer look.

4.1 Nemo — autonomous detection engineering

Rule maintenance at scale is the work customers least want and can least staff. In Mobula it is executed by Nemo, an autonomous agent operating inside the platform's versioning and gating discipline. Every rule Nemo writes is signed with its name and carries a run identifier in the changelog; the attribution is enforced fail-loud in code, so an unattributed write is structurally impossible rather than merely against the rules. The customer sees the outcome — rules that stay correct and current — with a full, queryable audit trail behind every change.

4.2 HASH Ocean — knowing what runs on the estate

HASH Ocean answers a question most estates cannot: what executables exist across this environment, what are their hashes, who signed them, and what changed since last week. It installs as a Windows service on each endpoint, incrementally inventories executable files — recording path, size, timestamps, Authenticode signature and signer, and MD5/SHA-1/SHA-256 hashes — and classifies each as new, changed or unchanged against its local manifest. Results flow to the SIEM through the Windows Event Log, and the agent runs at low priority with configurable pacing so users never notice it.

4.3 `mobula-ai` — AI that works disconnected

The air-gap AI layer runs on a dedicated GPU appliance inside the customer's own boundary, with no internet access. It provides the assistance analysts expect from a modern platform — summaries, suggested exclusions, threat narratives, rule explanations, triage pre-processing — using a local model with retrieval over versioned knowledge packs. Its intelligence is kept current through four independently-cadenced components: base model weights (quarterly), a fine-tuned adapter (daily-eligible, inactive on arrival until an administrator activates it), the knowledge pack (daily) and version-controlled prompt assets (daily). Updates arrive as versioned artifacts through the same content channel as detection kits — never as a live connection.

5. Deployment models

5.1 Cloud

The central platform runs on Google Cloud Platform: Cloud Run services, Cloud Build pipelines, Firebase Hosting and Secret Manager, with kit distribution to customer environments through managed Google Drive channels. Customers consume content; the cloud estate is operated by CyRay.

5.2 Air-gapped and on-premises — the Standalone Law

Mobula treats disconnected operation as a first-class deployment model, governed by one rule: **nothing may be installed in the air-gap environment.** Every on-premises artifact is a self-contained, single-file Windows executable — copy, configure, run — operating as a console application or a Windows service. There are no runtime installations, no package managers, no internet dependencies and no scripts. The installation experience is a double-click, an elevation prompt and a short wizard. Secrets at rest are protected with Windows DPAPI, never stored in plaintext.

Two properties make this model sustainable rather than a one-off port:

- **Dual-mode, one codebase.** Every collector runs identically as a cloud scheduled service (secrets in the cloud store, checkpoints in the database) and as an on-premises local service (secured local configuration, checkpoints on disk). The difference is a mode switch, never a fork — so air-gapped customers run the same code, at the same version, as cloud customers.
- **One shipping channel.** On-premises components ride the existing content-kit distribution chain; there is no separate install channel per tool. Endpoint sensors deploy through the customer's own mechanisms — GPO, SCCM/MECM, Intune, or standalone.

The air-gap model extends all the way up to the AI layer: the reasoning model runs on a customer-side GPU appliance with no internet access, fed by versioned content packs. A disconnected site gets the same continuously improving detection library and the same AI assistance as a connected one — on its own terms.

6. Differentiators

1. **Managed detection content as the product.** Customers never author or tune rules. A curated, MITRE ATT&CK-mapped, quality-gated catalogue is delivered and kept current automatically — with every customer's operational signal improving the shared library.
2. **Air-gap first, all the way up to the AI layer.** The entire stack, including the reasoning AI, runs disconnected. Even the documentation is built to render identically, offline, years later — fonts and artwork inlined, no network dependency anywhere.
3. **Constitutional data boundaries.** Tenant isolation and the customer-data-free master plane are architectural rules verified in code — not policies that depend on compliance.

4. **One parser truth.** Field mapping lives in exactly one place per source, and collection is structurally unable to fork it. Every rule downstream inherits consistency.
5. **Collection that cannot fail silently.** No-gaps-no-duplicates checkpoint semantics and an always-on heartbeat make a silent collector impossible by construction.
6. **Autonomous detection engineering.** Rule maintenance is executed by an agent whose every write is attributed, versioned and code-gated — expertise that scales without scaling headcount.
7. **Documentation discipline as a guarantee.** Every shipped tool carries branded installation, removal, upgrade and verification guides, and the release gate refuses to publish without a complete, current set.
8. **A verified release supply chain.** Releases publish self-contained, with every SHA-256 re-verified from the distribution point — what the customer receives is provably what was built.

Mobula in numbers

100K+ EPS	sustained ingestion
9 years	company track record
5 years	Mobula in production
20+	live deployments
6,000+	expert-engineered detection rules in production
7,500+	MITRE ATT&CK technique mappings across the rule library
4,400+	Sigma rules in the managed detection-content chain
2,300+	connector parser definitions — one field authority per log source
12	integrated modules across the ecosystem

Every figure above is either measured from the platform's own stored state or drawn from CyRay's operating record. This paper deliberately publishes no estimated or extrapolated numbers — including efficiency percentages. Mobula's efficiency claims are structural and countable instead: content changes are data-only writes that propagate to every environment with zero deployments, rule maintenance is executed by an attributed autonomous agent rather than by analyst hours, and the collection-to-tuning chain closes as an automated loop.

Customer references

Mobula deployments span eight sectors: financial services, technology and IT services, government, aviation, energy and water utilities, pharmaceuticals and life sciences, shipping and logistics, and manufacturing. Consistent with the data boundaries described in section 3.3, this paper names no customers. Named references are available on request, under NDA.

7. Conclusion and next step

Security operations improve when the scarce work — detection engineering, tuning, content maintenance — is engineered once, centrally, under machine-verified discipline, and delivered everywhere, including where the internet cannot reach. That is the design premise of Mobula: an AI-orchestrated Security Operations Platform whose guarantees are carried by architecture and code, not by staffing levels or promises.

For a security-operations team evaluating Mobula, the practical next step is a technical briefing and a deployment-model discussion — cloud, on-premises, or fully air-gapped — against your estate's log sources and constraints.

Compliance and certifications

CyRay, the company behind Mobula, holds ISO 27001 certification and a national CERT accreditation. These are company-level credentials — they attest to the organization that builds and operates Mobula.

To arrange that briefing, contact CyRay at **info@cyray.io**, or visit **cyray.io**.