EMERSA

INTERNAL. PRODUCT DEFINITION.

Two Products, One Family

The server platform, the endpoint agent, and how to keep them apart

Emersa Labs, London. August 2026.

1. The decision

There are two products. They share a codebase, a record format and a name, and they do different jobs for different buyers. Writing them as one product with weaker platforms was wrong and every existing document that does so needs correcting.

	Server platform	Endpoint agent
Runs on	SUSE Linux Enterprise Server. The preferred and supported target.	Windows and macOS. Small, local, on ordinary processors.
Its job	Prevent, divert and absorb. It acts.	Watch and record. It tells you.
Deception layer	Yes. This is the product.	No.
Enforcement	Yes, inside the operating system.	No. It observes and alerts.
Bait	Yes.	Yes, and it is the main detection capability.
Signed record	Yes.	Yes. This is the main deliverable.
Buyer	Data centres, government bodies, any organisation running servers.	Small organisations, individual machines, anyone who needs to know what happened.
Scale	Designed to grow across many machines.	One machine, running quietly.

THE ONE SENTENCE THAT KEEPS THEM APART

The server platform stops things happening. The endpoint agent tells you what happened. A customer who buys the second expecting the first will ask for a refund, and they will be right to.

2. The server platform

This is the full architecture: three tier data plane, six nodes, deception, metering, the signed ledger. It is the product the white papers describe and the thing worth building for the next two years.

Who it is for

- Data centres, where a single deployment covers many workloads and the economics of absorption make sense.
- Government organisations, where the requirement to show what happened is as strong as the requirement to prevent it.
- Any organisation running its own servers, which is the honest boundary of where this product applies.

Why SUSE is the preferred target and not just a supported one

- One reference platform means one set of behaviours to prove. Supporting every distribution multiplies the testing surface before anything has been measured once.
- The certification and the catalogue listing already exist, so the route to a buyer is shorter.
- The post quantum cryptography in the platform belongs to SUSE and is checkable, which is a stronger position than making the claim ourselves.
- Container and edge packaging is already part of that ecosystem, which is how the product reaches many machines without a separate deployment story.

3. The endpoint agent

Small, local, and running on ordinary hardware without special permission. It already runs on Windows and macOS. Its purpose is not a reduced version of the server product. It answers a different question.

What it watches

- Whether anything is listening on the machine that should not be, and whether that changed.
- Whether anything is looking through files, mail or documents in a way the machine does not normally do.
- Whether planted files or credentials have been touched or used.
- Whether the machine's own configuration has changed in ways that weaken it.

What it produces

A signed, ordered record of what happened on that machine. That is the deliverable and it is worth more than another alert feed, because most small organisations discover a compromise weeks later with no account of it at all. An agent that can say what was touched, when, and by which process turns an unknown into an incident report.

> **THIS IS THE ACTUAL PRODUCT**
> For a small organisation the valuable output is not prevention, which they will not configure, but evidence, which they cannot produce any other way. Sell the record, not the monitoring.

Why bait matters more here than anywhere

Bait needs no kernel access, no driver and no special permission. It works identically on Windows, macOS and Linux, on shared hosting and on a laptop. On the endpoint agent it is not one detection method among several. It is the detection method, because everything else the agent does is observation.

4. What is shared, and what must not be

Shared	Consequence
The event format	One record shape for both products, so the same tooling reads both.
The bait implementation	One implementation, used by both. Any defect in it affects both products.
The ledger and signing	One record format. An endpoint record and a server record are comparable and can be held together.
The core scoring engine	Same code. The endpoint agent simply has fewer inputs and no enforcement path.

Not shared	Reason
The deception layer	It requires enforcement inside the operating system and a place to put a fabricated environment. Neither exists on the endpoint.
The governor	It exists to bound the cost of an engagement, and the endpoint agent has no engagements.
Multi machine agreement	The endpoint agent is a single machine product and adding coordination would make it something it is not.

5. What this changes

- DEF-002, bait firing on backup agents and scanners, is now the most important defect in the programme. It gates both products, and on the endpoint agent it gates the only detection capability there is.
- The endpoint agent can ship long before the deception layer is proven, because it makes no claim that depends on session fidelity or measured efficacy. It is honest to sell an agent that watches and records.
- The Windows and macOS work leaves the open questions list. It is not unfinished platform parity. It is a different product with a smaller scope, and the documents should say so.
- Ukrainian Sisters and organisations like them become addressable. They have no server estate, so the server platform never fitted. The endpoint agent does.
- The penetration testing service sits naturally alongside the endpoint agent for small customers, where examining a website and testing an estate is often more valuable than any software.

6. Naming

The two products need to be distinguishable on the first page a buyer reads. Three options, in the order I would rank them.

- KRUPIQ Server and KRUPIQ Agent. Plain, immediately understood, and the relationship between them is obvious.
- KRUPIQ and KRUPIQ Watch. Keeps the full name for the flagship and gives the endpoint product a job title rather than a diminutive.
- Two separate names. Cleanest separation and the most expensive, because it means building a second brand.

Whichever is chosen, the endpoint product should never be described as a lighter version of the server product. A buyer hearing lighter assumes the same thing with less of it, which is the misunderstanding that produces a refund.

7. What needs deciding

- The naming, before any document is corrected.
- Whether the endpoint agent is sold, given away to seed the market, or bundled with an assessment service. Each produces a different product.
- Whether endpoint records can be collected centrally for an organisation running many machines, which is useful and is also the point at which the endpoint agent starts becoming a different and much larger product.
- What the endpoint agent does when it finds something. Alert only, or alert and act. Acting requires permissions we have said we do not need, so the answer is probably alert only, and that should be a decision rather than a default.
- Whether the server platform supports anything other than SUSE. Every additional target multiplies testing before anything has been measured once.

Summary One family, two products, two buyers. The server platform acts and is the thing worth building. The endpoint agent records and is the thing that can ship first. Bait is the component both depend on, which makes fixing it the highest priority work in the programme.