ottia

OTTIA WHITE PAPER

Software-defined tempo.

Software-defined assurance was the first step. What decides the fight is tempo — how fast a trusted, self-owned loop turns, all the way to the field. A perspective on where the software-defined shift goes next.

Tempo · Tooling · Reach · Ownership

17 OCTOBER 2026 · HELSINKI

© OTTIA 2026

OTTIA

In brief

- The defence debate has rightly turned to **software-defined assurance** — how to trust a capability that now changes constantly.
- But assurance is a property of a **loop**, not a one-off gate. In the field the binding constraint is the loop's **tempo**: the time from a new signal to a trusted update in operators' hands.
- Tempo is decided less by the assurance method than by three things the debate under-weights — the **tooling** inside the loop, how far it **reaches** and at what energy cost, and whether it is **owned** end to end.
- Take these seriously and software-defined assurance becomes something you can field: a **sovereign loop**. The organisations that internalise it — likely smaller and more heavily tooled than today's integrators — will set the pace.

> BUILDS ON
>
> In 2024 **Helsing** — one of Europe's largest defence-technology companies — published a white paper, *Software-Defined Assurance*, arguing that confidence in a capability must move at software speed rather than at the pace of once-a-decade certification. This paper extends that argument. We write from building such loops ourselves, but what follows is about the shift, not a product.

Helsing, *Software-Defined Assurance* (2024) · helsing.ai/download/whitepapers/software-defined-assurance

01

The shift so far

Modern capability increasingly lives in software running on cheap, adaptable hardware. Helsing's white paper drew the right conclusion: assurance — the confidence that a system is safe and effective — can no longer be a once-a-decade certification. It has to become continuous; it has to cover one software build used across many hardware platforms; it has to handle systems that combine and recombine into larger structures; and it has to find new ways to assure AI.

This reframes assurance from a gate into a process, and it is correct. We agree with all of it. Our argument begins where that one pauses — with a plain observation from the field.

The people operating these systems do not experience the problem as *assurance*. They experience it as **speed**. The adversary updates in weeks; a capability fixed today is behind within two years. Trust and speed are not separate problems to be solved in sequence. They are two faces of one object — a loop that has to turn quickly and stay trustworthy at the same time.

02

Assurance is a property of a loop

State the loop plainly: **signal → decide → build → ship**, continuously, with what ships becoming the next signal. New information — from the field, from intelligence, from operations — enters as work; a decision becomes a build; a release goes out; experience returns and starts the next turn.

In this picture assurance is not a stage bolted to the end. It is encoded in the loop itself: a continuous, tamper-evident record of what changed and why, with deployment gated on requirements being met. This is the deeper form of Helsing's continuous assurance — not the same assurance performed more often, but assurance as a property the loop carries at every turn.

The change in the unit of analysis is the point. Once the loop is the object, the question is no longer *"is this system assured?"* but **"how short can the loop be while remaining trusted?"** Everything that sets that cycle time — the tools inside the loop, how far it reaches, who owns it — stops being an implementation detail and becomes part of the capability.

03

Three dimensions the debate under-weights

If the loop is the capability, three dimensions decide how fast it turns. The assurance conversation tends to treat them as background. They are not.

01 **Tooling density.** A loop turns only as fast as the tools inside it. Two matter most: AI that carries the volume of building and analysis, and the ability to take an adversary's system apart and turn what is learned into one's own requirements. The second is also an *assurance input* — requirements grounded in the real threat, with known provenance — and it is largely absent from standards written for a world of hand-coded software.

02 **Reach and energy.** A software-defined capability is only worth as much as the places it can run. The same loop must reach from the rear to the frontline, over thin and contested links, on edge hardware measured in **watts, not kilowatts**. Assurance that holds only in a lab, or a capability that cannot be powered where it is needed, is incomplete. Physical reach and energy budget belong inside the assurance picture, not beside it.

03 **Sovereignty as ownership.** Helsing calls for customer-owned digital assets and infrastructure. The sharper point is that sovereignty is not geography — it is **self-ownership of the whole environment**. You can only guarantee that the loop keeps turning, run the models the work requires, and hold your own data and record, if the loop is yours end to end. Ownership is the precondition for assurance you can actually trust.

04

What it implies

These dimensions point somewhere uncomfortable for the current industrial model. A short, trusted, self-owned loop that reaches the field rewards **density of tooling over size of organisation**. A small, heavily-tooled team can turn such a loop faster than a large integrator — not as an exception, but as the logic of the shift.

This is not a claim about any one company. It is where the argument leads. Software-defined defence will be realised by whoever runs the fastest trusted loop, in their own hands, reaching the field — and that is a wider set of actors than today's procurement assumes.

The strategic task, then, is to recognise the **loop** — not assurance alone — as the capability, and to build the conditions for it: tooling, sovereign infrastructure, and acquisition that can contract for a loop rather than a box. Europe has made the first move by taking assurance seriously. The next is to see the whole loop, and to let it be owned and run by those who can turn it fastest.

ottia

info@ottia.com · ottia.army · Helsinki, EU

© OTTIA 2026 · ALL RIGHTS RESERVED