

Trust You Can See: CNTi Test Suite and the business case for Cloud-Native Network Functions

Authors

Cédric Ollivier
Network Automation Expert, Orange - Linux Foundation Networking Leader - CNTi Technical Steering Committee Member

Martin Matyáš
Lead Telecom Cloud Engineer, Tieto - CNTi Technical Steering Committee Member

Philippe Ensarguet
VP Cloud & Software Engineering, Orange Fellow - Co-Chair of Sylva Governing Board

The case in brief

Every **network function** today is **sold as cloud native. Operators** often **find out** whether that is true **when it's too late**, during onboarding and acceptance, where each iteration costs precious time. The **CNTi Test Suite**, an Apache-2.0 project of LF Networking's Cloud Native Telecom Initiative, has offered an open, vendor-neutral way to check those claims for years.

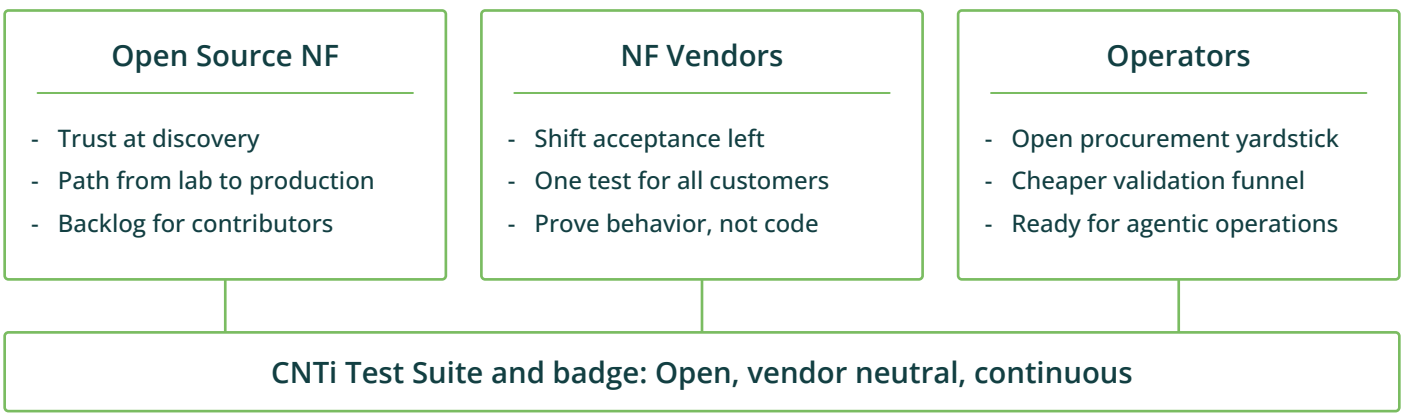
What is new is the introduction of the Cloud Native badge that removes friction: running the suite no longer needs a lab, a setup or a person to read the result. With the v2 release line and its companions, the GitHub Action and the GitLab CI/CD component, **any project hosted on GitHub or GitLab can run the CNTi essential tests on every change** to its default branch and **show the result as a badge** on its repository page. That turns cloud native compliance from a private, late-stage, often manual testing into **continuous and automated public evidence**, and this paper sets out the value to open source projects, vendors and operators.

This paper argues three things:

- **For open source network functions**, the badge is a trust signal at the point of discovery and a path from lab project to deployable component.
- **For network function vendors**, running the same open tests in their own CI shifts operator acceptance left, replaces many operator-specific checklists with one shared reference, and lets closed-source products compete on verifiable behaviour.
- **For operators**, a shared yardstick simplifies procurement and aligns directly with the Sylva Validation Center, where passing the CNTi Test Suite is one of the requirements for a CNF to be considered validated by Sylva.

The badge is not a certification, a CI runner-hosted cluster is not a telco cloud. Used for what it is, as the first and most economical layer of a trust stack that continues with formal certification and platform validation, it can make cloud native quality something the market sees rather than something it assumes.

One open yardstick, three kinds of gain



A note on terms: evidence, certification and validation

This paper uses three terms that are often blurred in the network function market. They are not interchangeable.

Test evidence (the CNTi badge). A network function project runs an open, published test suite itself and publishes the results. Nobody else reviews them. What makes them trustworthy is that anyone can rerun the same tests, read every finding and check which pipeline produced them. A badge shows how a given version of a network function behaved against a given version of the tests, at a given time. It is not an endorsement by LF Networking, CNTi or Sylva.

Certification. An independent body or a network vendor reviews a product against a defined, versioned set of requirements and grants a mark, usually for a limited period and a specific product release. Certification answers the question: *does this product conform to the program's requirements, as confirmed by a third party?*

Validation. A network function is tested in a specific target context, such as a given platform, configuration or operator environment, to confirm that it deploys and runs as intended there. Validation answers the question: *does this product work here, for this use?* The Sylva Validation Center validates network functions on Sylva-compliant platforms; operators then run their own acceptance on their own clouds.

In this paper, the CNTi badge is test evidence: self-run, continuous and public. It is the first and fastest layer of trust, a prerequisite rather than a substitute for certification or validation. A green badge says that a network function follows essential cloud native practices on Kubernetes. It does not say that a third party has certified the product, or that the product is fit for a particular telco cloud or network or that its operational lifecycle (e.g. GitOps workflows, automated upgrades/rollbacks via external tooling, or day-2 configuration management processes) is automated or free of manual or proprietary steps.

Why this white paper?

We are publishing this white paper to introduce the new version of our badge. It now provides the easiest way to demonstrate and share that a network function (NF) has implemented essential security and cloud native practices.

Why now

- The v2 release line and the GitHub Action make continuous, public CNTi results possible for the first time, with only a handful of adopters so far.
- Habits set in the next few months by early adopters, vendors and procurement teams will decide whether the badge becomes an industry-accepted market signal.
- Telco cloud platforms such as Sylva are entering production at scale, which makes workload quality a deployment bottleneck rather than a lab concern. With the EU Cyber Resilience Act, any verifiable evidence of cloud native behaviour also becomes valuable input to security analysis.

What we want readers to take away

- **Business leaders** see proven cloud native compliance as a real cost and time-to-market lever.
- **Open source network function projects** see the badge as a credible path from lab project to deployable component.
- **Vendors** have a concrete case for running CNTi in their own CI and sharing results publicly with every potential customer.
- **Operators** have a shared, open yardstick to reference in procurement and acceptance.
- **Everyone** understands the badge scope and how it fits with certification and platform validation.

What we hope it triggers

- Encourage vendors to embrace cloud native patterns fully
- Establish the CNTi badge as an accepted telecom-industry indicator, starting with open source network functions.
- CNTi results requested as a standard procurement artefact.
- More maintainers and contributors invested in the CNTi Test Suite.

The operability gap

Cloud native claims are inexpensive to make but expensive to verify, and the cost of verification lands on the operator, late.

A decade into network function virtualisation and containerisation, "cloud native" has become a default line in every (new) product sheet.

In practice, operators still find network functions that need local storage to keep state, ignore SIGTERM, run privileged containers, hard-code IP addresses, require a separate day-2 configuration management lifecycle with legacy protocols like YANG, or cannot survive a node drain. None of these are functional defects. They are operability defects, and they decide whether a network function can live on a shared Kubernetes platform alongside others.

These defects typically surface during operator onboarding or acceptance. By then, the release is frozen, the fix belongs to the vendor's next cycle, and each round trip costs weeks. The same discussion repeats with every operator, each with its own checklist and vocabulary.

The gap is not a lack of good practice. Cloud native best practices are well documented. The gap is the absence of a shared, economical, continuous and visible way to show that a given network function follows them.

CNTi: A shared, vendor-neutral yardstick

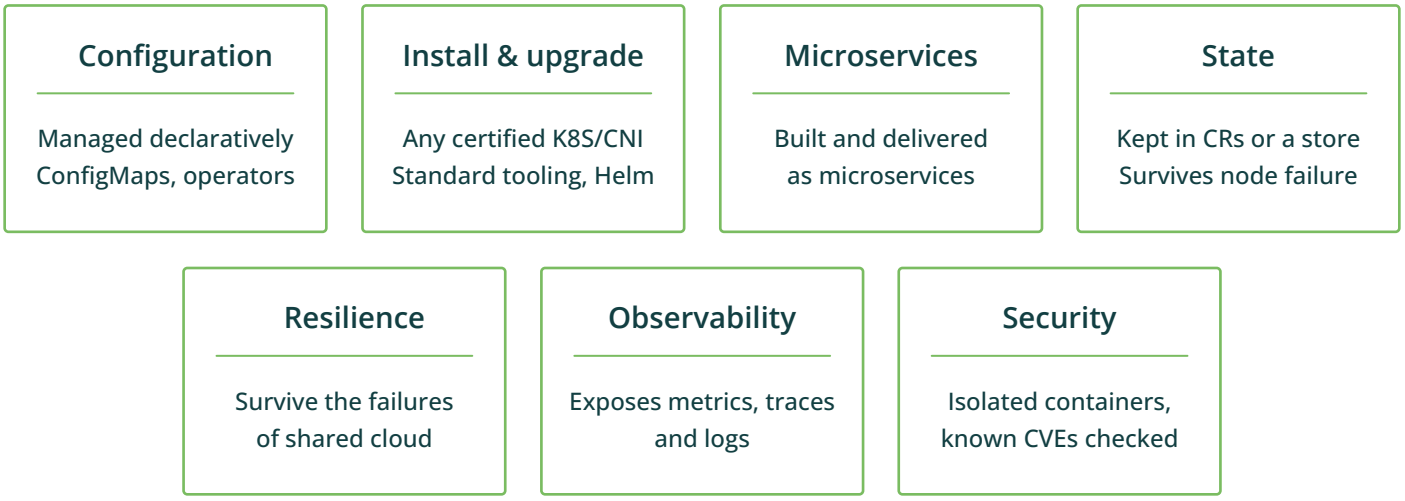
The Cloud Native Telecom Initiative (CNTi) chains best practices, automated tests and test execution evidence, so that "cloud native" means the same thing to a vendor and to an operator.

CNTi is an LF Networking initiative that absorbed the CNCF CNF Certification program and CNF Test Suite and continued enhancing them to align with the evolution of Cloud Native technology.

Its work today centres on the **CNTi Test Suite**: **defining** what **cloud native behaviour** means for a network function as executable tests, and **automating** their **run** so that the check happens in a project's own CI rather than in a lab. The tests are grounded in the initiative's **Best Practices**, a **vendor-neutral description of cloud native practices**[1] **for network functions**, and each failed test points to the practice it stands for and **how to meet it**. What the suite produces is evidence: a **report and a badge** that show which tests a network function passes, on which version, and as of when.

1. https://12factor.net/

Seven categories of cloud-native behaviour



The Test Suite inspects a network function across seven categories depicted here after, focusing on how it behaves on Kubernetes (e.g. Kubernetes resource lifecycle management: creating, updating, and deleting resources via Kubernetes APIs or Helm).

Scope note: CNTi validates Kubernetes compliance (how the CNF behaves on Kubernetes: security, resources, networking, resilience, observability, and Kubernetes object lifecycle). It does not assess broader operational readiness such as GitOps delivery pipelines, CI/CD automation, secret/config management workflows, or whether day-2 operations rely on manual or proprietary tooling. A CNF can satisfy CNTi's essential tests while still requiring operator-specific operational processes.

Four properties make it a **credible yardstick** rather than one more checklist:

1. **It is open source**, so anyone can rerun the exact tests used to evaluate a product.
2. **It is vendor-neutral**, governed under LF Networking.
3. **It is anchored in 5G workloads: OCUDU**[2], **free5GC**[3], and **Aether SD-Core**[4], leading open source telecom building blocks, are the reference CNFs and run the essential tests nightly in the project's CI.
4. **It is recognised by industry requirements**: GSMA's NG.139[5], Cloud Infrastructure Reference Architecture managed by Kubernetes, cites the CNTi Test Suite (under its former name, cnf-testsuite), and its documented rationale, as the reference for 29 of its Kubernetes workload requirements.

2. https://ocudu.org/
3. https://free5gc.org/
4. https://aetherproject.org/sd-core/
5. https://www.gsma.com/newsroom/wp-content/uploads/NG.139-v2.0.pdf

The v2 release line, released on 15 September 2026, introduces a major usability overhaul. It declares scoring at the level of each test, attaches a documented remediation to every failure, derives the exit code from whether the run met its objective, and drops the platform, RAN and 5G-specific suites to focus squarely on workloads. It also renames the tool from cnf-testsuite to cnti-testsuite.

About the Sylva project

Sylva[6] is an open source project hosted by Linux Foundation Europe[7]. It was launched in late 2022 by European operators (Deutsche Telekom, Orange, Telefónica, TIM, Vodafone) and vendors (Ericsson, Nokia). Its goal is to reduce fragmentation in telco cloud infrastructure by providing a common, production-grade cloud native stack and reference implementation for running network functions on bare metal or IaaS (OpenStack, OpenNebula and KubeVirt).

Its **Validation Center** tests vendor network functions on Sylva-compliant platforms. It checks both that the platform offers what the network function needs, and that the network function deploys and runs correctly on it. The CNTi Test Suite is a mandatory step in every Sylva validation. CNTi asks whether a network function behaves well on Kubernetes; Sylva asks whether it runs on a real telco cloud platform.

The badge: from point-in-time compliance to continuous public evidence

The badge makes a network function's CNTi result visible where people discover it, and keeps it current on every change to the main branch.

It is delivered by two companion projects that CNTi maintains, the **CNTi Test Suite GitHub Action**[8] and the **GitLab CI/CD Component**[9]. They are feature-identical today, and CNTi keeps them at parity as far as the two platforms allow, so a project hosted on GitLab gets the same badge and evidence as one on GitHub, without building its own pipeline. In one workflow step, the action creates a three-node kind cluster, installs the latest cnti-testsuite release (v2.0.0 or newer), deploys the network function from its Helm chart or configuration, and runs the essential tests. It writes a job summary, annotates failed tests, uploads the results file, and fails the job if the objective is not met. For more complex network functions, it offers the possibility to run the checks on privately provided runners and/or more production-ready clusters.

6. https://sylvaproject.org/
7. https://linuxfoundation.eu
8. https://github.com/lfn-cnti/testsuite-action
9. https://gitlab.com/lfn-cnti/testsuite-component

On the default branch, the action also publishes a badge showing:

- The CNTi logo,
- The task and the score, for example `CNTi | essential | 16/19`,
- Green when the run met its objective.

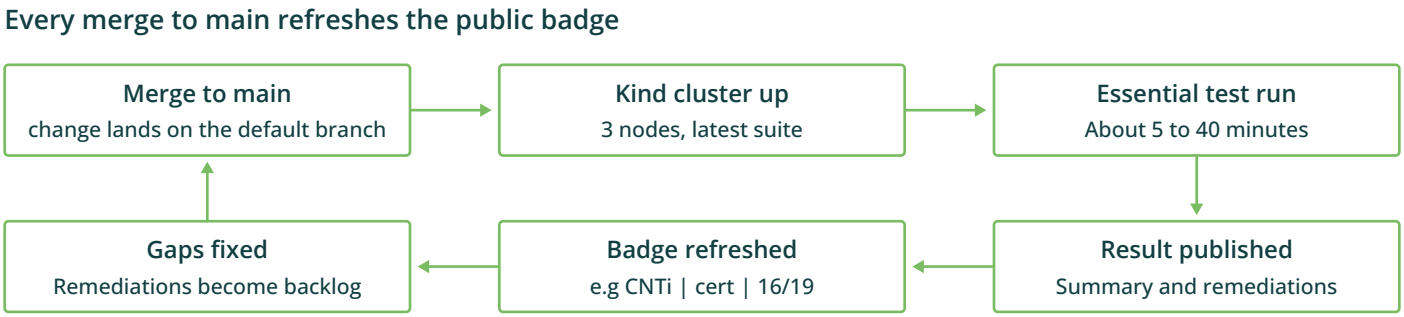
Example here after:



The badge reflects the latest network function status for its default branch.

Three features change the economics of compliance:

- **Continuous.** The result is recomputed on every merge, not frozen at a date. Regressions show within a day.
- **Public.** The score sits on the README next to other trust signals such as the OpenSSF Scorecard. A buyer sees it before deploying anything.
- **Inexpensive.** It requires one step in an existing CI pipeline and runtime depends on the complexity of the network function, with simpler functions completing in as little as five minutes.



The loop runs only on the default branch; pull requests can be tested as well, but their results don't change the published badge.

Public or private repository, the evidence is the same.

A badge alone is only a number; what makes it evidence is what stands behind it: the full results with every finding and its remediation, the configuration the run used, and a signed statement of which pipeline produced them. The action and the component publish that bundle together with the badge. An open source project keeps it in its own repository. A vendor whose code is private publishes the bundle to a public place it controls, or attaches it to the chart in its registry, so a buyer can see the badge, read the results and verify their origin without seeing the source.

Adoption is early but real. As of October 2026, Jitsi Meet and Aether SD-Core run the CNTi validation in their own CI and display the badge, and OCUDU and free5GC are being checked as reference network functions inside the CNTi Test Suite repository. The CNTi maintainers are already designing an "adopter canary" that replays each adopter's own configuration against new suite builds, so that suite changes do not silently move adopters' badges. That is evidence that the project now treats its badge holders as a downstream ecosystem to protect, not just as users.

Using the badge. A badge is only as trustworthy as the rules around it. Because it is self-run and self-published, its value depends on every project displaying it the same way: from an unmodified, released version of the test suite, on the configuration the project actually ships, kept current by its own CI and linked to the run behind it. The rules will also set how the badge may be named and presented, so that it is never mistaken for a certification or an endorsement by LF Networking, CNTi or Sylva, and what happens when a badge no longer reflects its project.

CNTi will define these rules in the CNTi Test Suite repository, alongside the tests themselves, so that they are versioned, public and open to change through the same review as the suite. Projects adopting the badge today should treat them as part of the badge.

Proposed badge usage rules

A CNTi badge is valid only if:

1. It was generated, by the cnti-testsuite, the GitHub Action or the GitLab component, from a real run. Never redrawn or edited.
2. It links to its evidence: results, configuration, suite version, date and signed origin.
3. It covers what was tested: one network function, one release, one configuration.
4. It is current: refreshed on a supported suite release, or removed.
5. It is described accurately: "CNTi essential tests: 16/19", never "CNTi certified", "LF Networking endorsed" or "Sylva validated" unless separately obtained.

The same rules apply on product pages, datasheets and tenders. The CNTi name and logo are used under the Linux Foundation's trademark guidelines. Anyone can report a misleading use to the CNTi maintainers; LF Networking may then ask for it to be corrected or removed.

Value for the ecosystem

The same open tests and badges create a different kind of value for each of the three stakeholders: open source projects, network function vendors, and telecom operators.

Value for open source network functions

For open source network functions, the badge gives an early, public indication on the question that often slows adoption: how well does this behave as a cloud native workload, beyond the lab?

Trust at the point of discovery. Today, an operator evaluating an open source 5G core has to deploy it to learn how it behaves on Kubernetes. A badge puts a first answer on the repository page. Paired with the OpenSSF Scorecard, it gives two complementary signals: how the project manages its supply chain, and how its workloads behave on a cluster.

A path from lab to production. Operators rarely reject open source network functions on features. They hesitate on operability: state, upgrades, resilience, observability, security posture. Those are exactly the CNTi categories. A rising score is a concrete, external measure of production readiness that a project can point to in conversations with operators and integrators.

A contributor and funding magnet. Every failed test comes with a documented remediation. That turns a score into a ready-made backlog of well-scoped issues, a natural on-ramp for new contributors. It also gives sponsors and foundations a simple progress story: from 12/19 to 18/19 in two releases.

A test of the tool on network workloads. Commercial workloads validated by Sylva such as PowerDNS and Titan.ium, open source adopters such as OCUDU, free5GC and SD-Core also harden the suite itself, as recent fixes driven by their runs show. Each new adopter makes the yardstick more credible for everyone else.

Value for network function vendors

For vendors, the business case is an affordable, faster path through operator acceptance, plus a way to prove behaviour without opening the code.

Shift acceptance left. A cloud native defect found by the operator costs a release cycle. The same defect found by the vendor's own CI on the day it is merged costs an afternoon. Running the CNTi essential tests on every merge moves the discovery of operability issues to the cheapest point in the lifecycle. This is the core return on investment, and it holds whether or not the vendor ever publishes a badge.

One test, many customers. Each operator brings its own cloud native requirements, wording and evidence format. A shared, open test suite gives vendors one reference to meet and one result to share. The CNTi essential tests make the same point from the buyer side: any end user can rerun the identical open source tests used to validate a product.

Arrive at Sylva Telco Cloud platform validation with no surprises. The CNTi Test Suite is mandatory in every validation run of the Sylva Validation Center, which validates vendor network functions on Sylva-compliant platforms. Vendors that run the suite continuously arrive at that validation and at the operator deployments built on Sylva already knowing their result. As a benefit, it is accelerating the Sylva Validation process.

Differentiate on verifiable behaviour. Closed-source vendors cannot show their code, but they can show how their product behaves. Publishing CNTi results, as a badge on a public chart repository or as an artifact in tender responses, turns “cloud native” from a marketing adjective into a checkable claim. Early movers will set the reference others are measured against. The evidence behind the badge can be published without the code: a vendor pushes the results, the configuration and a signed statement of origin to a public location of its choice, or alongside the chart in its registry, and the consumer verifies the signature rather than the vendor’s word.

Remediation built in. With v2.0, every failed test carries a documented remediation. Engineering teams get actionable guidance rather than a pass/fail verdict, which lowers the cost of closing each gap.

Value for operators

Operators gain a common language for procurement, a first validation funnel, and a precondition for the next wave of automated operations.

A procurement lever. Operators can ask for recent CNTi essential test results as a standard artefact in RFIs and RFPs, alongside SBOMs and security attestations. Because the tests are open, the operator can rerun them and confirm the claim. Because they are shared, vendors answer once for the whole market.

A first validation funnel. Platform validation, such as that of the Sylva Validation Center, is expensive: real hardware, real CNIs, real integration work. Network functions that already pass CNTi workload tests reach that stage with fewer basic defects, so scarce validation capacity goes to platform-specific issues.

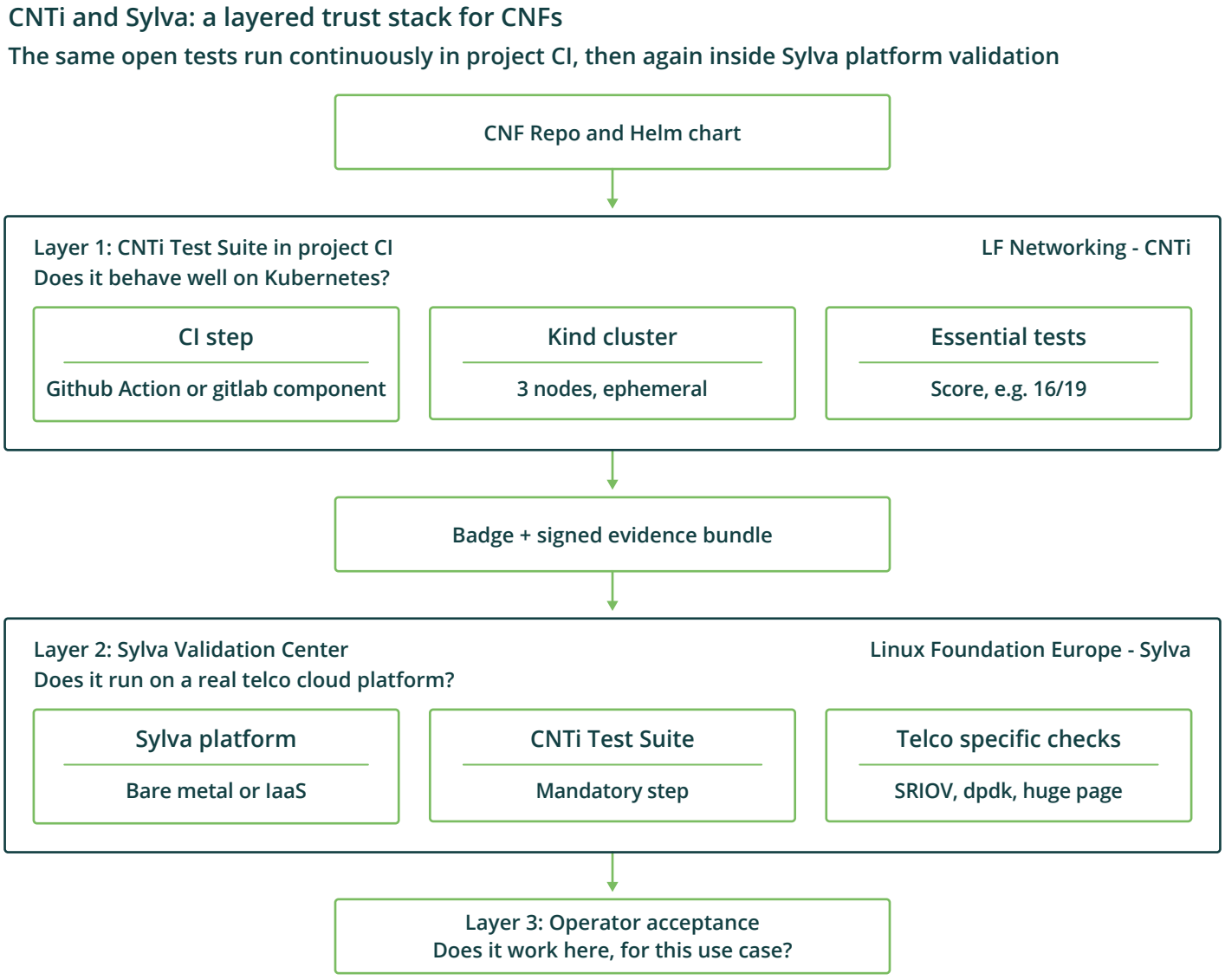
A coherent trust stack. The badge combines naturally with other open signals: the OpenSSF Scorecard for supply-chain hygiene, CNTi essential execution score, and Sylva validation on the target platform. Together they describe a trustworthy network function from source code to deployment.

A precondition for agentic operations. Network functions that maintain state in declared resources, expose proper health probes, handle termination cleanly, and externalise their telemetry are better equipped to support safe, reliable automation and AI-driven operations.

CNTi behaviours are, in effect, a precondition for agentic network operations, but not sufficient on their own. Safe agentic or autonomous network operations also require trusted state management, controlled actuation, governance, authorisation, guardrails, and operational safeguards. The suite's own roadmap points the same way, with recent work on using it from an AI agent through MCP.

Honest limits and complementarity

The badge is the first and most economical layer of trust, not the last. Presented honestly, its limits strengthen the case for the layers that follow.



How they fit together: CNTi validates cloud native compliance and the Kubernetes resource lifecycle. Sylva validation covers telco platform integration and telecom-specific lifecycle. Operator acceptance covers site-specific needs. Telecom-specific lifecycle validation is expected primarily in Sylva and operator acceptance, not CNTi.

Each layer answers a question the previous one cannot, at a higher cost; the badge is valuable because it makes the first answer free and continuous.

A kind cluster is not a telco cloud. The default environment is a small kind cluster. It cannot exercise SR-IOV, DPDK, hugepages or multi-interface data planes. A perfect score shows that a workload is well-behaved on Kubernetes, not that it is carrier-grade on a given platform. In addition, telecom CNFs often introduce state management, recovery, and service continuity considerations (e.g. state replication, graceful failover, and restoration after failure) that are not exercised by CNTi tests. These domain-specific requirements call for complementary validation under realistic platform and network conditions. That is precisely why platform validation such as Sylva's remains necessary.

Scope has narrowed. v2.0 removed the RAN and 5G-specific suites to focus on generic workload behaviour. Telco-specific behaviour now has to be covered elsewhere, and the paper's readers should know where.

Sustainability is a real risk. Recent releases rest on a very small group of maintainers. An ecosystem asked to build procurement and CI on this tool must also invest maintainers in it.

Cost and visibility are uneven. A full validation run usually takes tens of minutes, which suits nightly or merge-time runs rather than every pull request. Proprietary vendors will mostly run it on private CI, so their badges depend on them publishing the evidence; the first public badges come from open source projects.

Call to action

The badge only becomes a market signal if each part of the ecosystem takes one concrete step.

Open source network function projects

- Add the CNTi GitHub Action or GitLab component to your chart repository and publish the badge.
- Treat failed tests as a public, well-scoped backlog and invite contributors to close it.
- Run the CNTi workload suite as a second step, which includes both essential and normal tests (~60 tests), all hardened and extensively tested for CNTi v2.0.0.

Network function vendors

- Run the CNTi essential tests in your internal CI on every merge, before any customer sees a release.
- Share results with operators as a standard artefact, and publish a badge wherever you have public charts.

Operators

- Reference recent CNTi results in RFIs and RFPs, and rerun them yourself.
- Keep CNTi mandatory in platform validation programmes such as the Sylva Validation Center.
- Fund maintainers: the tool your procurement relies on needs more than a handful of contributors.

References

CNTi

- CNTi Test Suite repository: https://github.com/lfn-cnti/testsuite
- CNTi Test Suite 2.0.0 blog post: https://lfnetworking.org/cnti-test-suite-2-0-0-tests-that-show-their-work/
- CNTi Test Suite v2.0.0 release notes: https://github.com/lfn-cnti/testsuite/releases/tag/v2.0.0
- CNTi Test Suite GitHub Action: https://github.com/lfn-cnti/testsuite-action
- CNTi Test Suite GitLab Component: https://gitlab.com/lfn-cnti/testsuite-component
- LF Networking, Cloud Native Telecom Initiative: https://lfnetworking.org/cloud-native-telecom-initiative/
- LF Networking, free5GC becomes a reference CNF for the CNTi Test Suite: https://lfnetworking.org/?p=41088
- LF Networking, strengthening the CNTi test suite: https://lfnetworking.org/?p=40489

Sylva

- Sylva project website: https://sylvaproject.org/
- Sylva project on GitLab: https://gitlab.com/sylva-projects/sylva
- Sylva core stack (sylva-core): https://gitlab.com/sylva-projects/sylva-core
- Sylva Validation Center: https://gitlab.com/sylva-projects/validation_center

About the authors

Cédric Ollivier is a network automation expert at Orange with strong experience in open source, where he has played all the top Linux Foundation Networking roles, from multiple project leads and steering committees, to the Governing Board. Convinced about the need to automate, Cédric evangelises the integrator job and the open source process benefits to all the layers, from executives to developers and testers. With the desire to share his expertise, Cédric gives many speeches at Orange and open source conferences, writes blog posts and, on some occasions, has been interviewed at the request of Orange or the Linux Foundation.

Martin Matyáš is a Lead Telecom Cloud Engineer at Tieto, working on cloud native platforms and applications for telecom operators. With over 20 years in the industry, he builds and operates scalable, resilient systems based on Kubernetes, with a particular focus on testing and automation. A passionate open source contributor, he is a maintainer of the CNTi Test Suite within LF Networking, where he drives the CNTi badge, its GitHub Action and GitLab CI/CD component. As a leader, Martin favours structured approaches that let teams own their outcomes, and he works to bridge the gap between complex systems and solutions people can actually use.

Philippe Ensarguet is an Orange Fellow, VP Cloud & Software Engineering, and 2021 award-winning CTO, championing at the crossroads of AI, cloud native, and platform innovation, currently driving telco-to-techco transformation. A three-time CTO with 25+ years' experience, he advises executives, VC funds, and tech startups. Board Advisor for Linux Foundation Europe, Sylva project governing board co-chair and active in open source communities, Philippe shapes technology strategy through advisory, mentorship, and thought leadership. Proud member of Tech.Rocks community.

Acknowledgement

Reviewers

Authors want to thank **Guillaume Nevicato** (Orange), **Ranny Haiby** (The Linux Foundation), **Richard Lopez** (F5 Inc.), **Luis Velarde** (Telefónica), **Eric Debeau** (Orange), **Morgan Richomme** (Orange), **Feng Tu** (free5GC), **Andre Puschmann** (SRS), **Tom Kivlin** (Vodafone), **Guillermo Carreto** (Telefónica), **Jill Lovato** (The Linux Foundation), **Ashan Senevirathne** (Telstra), **Patrik Rokyta** (Titan.ium), **Joel Studler** (Swisscom), **Neil Cook** (PowerDNS), **Gabriel Arrobo** (Intel and Aether SD-Core), **Arpit Joshipura** (The Linux Foundation), and **Kai Steuernagel** (VP Cloud Technology at Deutsche Telekom) for their precious feedback and contributions during working on this white paper.

Use of AI in preparing this paper

In the spirit of transparency, we used AI tools in two ways:

- An AI-powered “second brain” to manage the context of the topic and keep facts, terms and arguments consistent across the paper.
- Claude, by Anthropic, as a sparring partner to challenge and shape the narrative flow and propose the document’s initial organisation.

The positions, choices and conclusions expressed here are ours as authors, and a human has reviewed every statement before publication.