Continuous Delivery for Critical Network Infrastructure

Enabling Secure and Agile Networks

White paper

Telecom networks are critical in providing essential connectivity for subscribers, businesses, and governments. Secure infrastructure and competitive networking services necessitate that Communication Service Providers (CSPs) and vendors embrace Continuous Delivery (CD) within a highly regulated environment.

This white paper examines the key factors contributing to successful CD in telecommunications. We go beyond software engineering practices to explain what must be accomplished throughout the end-to-end product value stream. Additionally, we outline two use cases demonstrating CD at scale.

Contents

Software Delivery for Critical Network Infrastructure

Continuous Delivery (CD) is a universally recognized practice of software delivery. It involves providing frequent, small, and incremental updates to a product. Changes are regularly deployed to labs, staging, and production environments.

"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale"—The Agile Manifesto [1]

The key benefit of CD is reduced time to market (TTM) for new features. Product companies continuously experiment by releasing functional updates. This creates a fast feedback loop to test the customer value hypothesis before completing the development work. TTM is reduced because the risk of completely missing market preference is minimized over time through adjustments, pivots, or even termination of an experiment. Such experiments inform portfolio managers where to invest or not.

From a product quality perspective, CD represents a positive feedback loop that enhances best practices in an engineering organization. Delivering continuously means a development team must keep the software functional while providing architectural capabilities for continuous business experimentation. Commercial and quality payoffs, i.e., shorter TTM and fewer defects, have made CD widely implemented by software companies.

As consumers, we find CD models in digital products such as apps on our smart devices. As businesses, we need to be a little more cautious. We prefer that limited number of users test new software before an IT administrator rolls out the update across an enterprise. We don't want to risk cutting employees from essential systems in case of a leaked defect.

Similarly, suppliers and businesses involved in the critical infrastructure (CI) acknowledge the commercial and quality benefits of CD and have adopted it. However, CI industry requirements must be thoroughly considered.

CI – critical infrastructure, not to be mistaken for continuous integration in the context of this paper – is well defined by the US Government, referring to sixteen critical sectors.

"(…) so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof" —The US Government [2]

Telecommunications is one of the CI sectors. Following a similar definition, the European Union's NIS2 Directive recognizes networks as the fundamental fabric of digital transformation. This puts the highest regulatory compliance standards and reporting regime on Communication Service Providers (CSPs) and the suppliers of critical network infrastructure (CNI). Consequently, increasing cybersecurity threats make policymakers revisit national CNI protection strategies, expecting lightning-fast responsiveness. The EU's NIS2 Directive holds critical organizations, such as CSPs, accountable for third-party and vendor adherence to robust security practices [3]. The UK's Telecommunications Security Requirements (TSR) mandate CSPs to have rapid incident response mechanisms in place [4]. CNI must be continuously monitored, with vulnerabilities detected, and then swiftly addressed within defined timeframes. Similarly, France's Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI) emphasizes response capabilities as essential to CNI maintenance frameworks.

Regulatory aspects, especially cybersecurity, reliability, and availability requirements, are not the only ones determining the implementation context of CD for telecoms. Another angle to consider is the complexity of CNI. With the introduction of 5G Service-Based Architecture (SBA), flexibility, scalability, and interoperability of networking solutions have been improved as CNI software became more modular, open, and decomposed. This means that multiple interconnected microservices, that constitute network functions, communicate via Application Programming Interfaces (APIs). SBA also allows network functions to be exposed to third-party applications. This approach requires decentralized security vulnerability management that needs to be equally responsive to threats across all microservices. Naturally, this transformation does not occur in isolation from the cloud journey:

- Configuration management (CM) and lifecycle management (LCM) in a multi-layered cloud-native system become increasingly complex to manage with current methods.
- CSPs relying on networking solutions from multiple software vendors often struggle to onboard suppliers uniformly into their deployment systems.
- Release cadence across the complete CNI stack varies. The Kubernetes roadmap underpins cloud providers' release cadence, and open-source projects follow their own release calendars.

The pressure from regulatory requirements and the complexity of modern networks highlight the need for CNI software to update continuously to meet the highest cybersecurity standards and ease network operations. This enhances CD's attractiveness as a sustainable choice for software maintenance contracts. A more secure and easy-to-deploy infrastructure amplifies the traditional CD benefits of competitive advantage and superior product quality. As the case for CD has evolved, so have management practices and technical capabilities in products and accompanying tools.

The Road Towards Autonomous Networks

In the rapidly evolving telecommunications landscape, CD is not just a method—it's a strategic imperative. While the industry may not fully adopt the 'fail fast' philosophy common for business-to-consumer (B2C) organizations and non-critical products, it significantly benefits from CD. Whether enabling a proactive security posture through continuous care or accelerating competitive advantage with continuous feature introduction, CD redefines how CSPs deliver innovation and keep CNI secure.

Customer buy-in is central to a CD framework. A high-stakes CNI environment requires a CSP to have the final say on what gets deployed to production. The digital gates of the software supply chain must fulfill this requirement. Software that makes it to the final gate is quality-checked and security-hardened, thanks to a scalable product creation process guided by the overarching principle of single-trunk development. The scalability of this process is evident through iterative development, continuous integration, left-shifting (i.e., automating and executing tests as early in the process as possible), and system verification.

As we analyze the Voice of Customers reports, the product capability of in-service upgrades emerges as the primary requirement before scaling out a CD model across the network. Customer expectations, such as a shortened maintenance window, no service interruptions, and no need for spare capacity in a data center during upgrades, must be translated into development principles and stringent criteria for continuous release milestones. This approach applies to any deliverables, whether they are bug fixes, security vulnerability patches, or functional service packs.

Various product deliverables are shipped to customers in CD format, so multiple processes are anchored in the same delivery model. For example, continuous vulnerability patching is integrated into the security vulnerability management process, which is part of the broader security design governance implemented by product lines. These processes are optimized for responsiveness and traceability. When embedded into the software maintenance process at the customer interface, they provide CSPs with a continuous stream of targeted security vulnerability patches. This encourages a minimal assessment turnaround and a quick response within the agreed Service Level Agreements (SLAs).

Similarly, continuously introducing new product features and enhancements demands integrating a CD model into the product creation process. Frictionless deployment of major releases and functional service packs requires focused management of non-functional requirements. Professional services and CSPs operations teams value troubleshooting, monitoring, observability, and product lifecycle management (LCM). These capabilities are must-haves for incrementally delivering software pieces of larger business functionality to production environments. Our internal research survey finds that confidence in product quality, automation, and troubleshooting is the key success factor behind well-executed CNI delivery projects. CSPs reaffirm that a series of successful delivery projects, such as software upgrades, architecture modernization, or greenfield deployments, is the most reliable signal for scaling out CD. Software quality assured at every stage of the CD pipeline is imperative to avoid unwanted service impact, and if an unexpected event happens, swift rollback resolves the issue.

There is rarely a single path to a commercially viable CD model for CNI. While fundamental enablers exist, some generic to the software industry and others specific to CNI, various market drivers define a use case and determine the exact logistics behind a CD framework.

Table 1. CNI Continuous Delivery: Two Paths to Value

	Security-Driven CD	Feature-Driven CD
Drivers	Regulatory compliance Vulnerabilities and risks Service Level Agreements (SLAs)	Time-to-market (TTM) Competitive advantage New Product Introduction (NPI)
Characteristics	e.g., monthly On demand if needed; otherwise, planned Security-only patch No functional changes	e.g., monthly, bi-monthly Planned Service pack with new content Incremental, testable functional changes
Key enablers	Design for security Minimal assessment turnaround	DevOps and left-shifting Quality, automation, troubleshooting
Common enablers	Customer buy-in through a CD-ready acceptance process Single-trunk development practice Digital supply chain with automated regulatory & compliance checks In-service upgrade capabilities for minimum service interruption	

The nature of CNI delivery projects has changed over time. Migration from previous generations of bare metal and virtualized network functions (VNFs) continues. Modern infrastructure involves cloud-native network functions (CNFs) running on any cloud platform. Cloud-native development principles are a non-negotiable part of product engineering culture. This also triggers the industry to revisit delivery practices. Drawing inspiration from hyperscalers and cloud providers, the next evolution of CD in CNI embraces several transformative approaches.

GitOps represents a natural progression. By using Git repositories as the single source of truth for declarative CNF states, CSPs can achieve improved visibility, enhanced audit trails, and better disaster recovery capabilities. The GitOps paradigm aligns particularly well with CNI's strict security and compliance requirements, as it is version-controlled and traceable. Intent-based LCM and CM enable CSPs to define desired outcomes, allowing automation to handle the implementation details.

The transition to CD 2.0 must acknowledge the unique constraints of the telecom industry. The challenge lies in adopting these advanced practices while ensuring the five-nines reliability that CNI services require. Extreme automation, which underpins fast-paced, commercially viable CD, combined with regulatory compliance, becomes the foundation of Autonomous Networks.

The path forward requires transformative efforts throughout the complete product value stream.

- Recognize CD as a universal delivery mechanism for all infrastructure deliverables, such as security vulnerability patches, maintenance packs, service packs, and major releases.
- Apply CD regardless of the target production environment, whether a private, hybrid, public cloud, or a software-as-a-service (SaaS) platform.
- Implement GitOps practices and declarative, intent-based frameworks that address CNI requirements, such as role-based access control, audit trails, traceability, and reporting.
- Replace traditional, mostly discrete, delivery pipelines with IT solutions that continuously reconcile discrepancies between declared intent and system state safely and in compliance with regulations.
- Automate pervasively throughout development, delivery, operations, and care, leveraging product capabilities to streamline a truly end-to-end continuous delivery.

Continuous Delivery – Two Paths to Value

Implementing CD practices, whether for rapid security patching or incremental feature deployment, completes the “act” component of Nokia’s Autonomous Networks vision. This is particularly evident in how security vulnerability patches are delivered outside a planned release cadence and how feature-driven CD enables swift adaptation to new requirements, as demonstrated in the 5G SBA implementation cases below.

After recognizing CD as a universal delivery mechanism for modern CNI, a commercially viable delivery model may take a slightly different shape based on the market use case. Here, we examine two scenarios: one in which actors are incentivized for responsiveness (i.e., security-driven) and another that rewards shorter TTM (i.e., feature-driven).

Security-Driven Use Case

Security vulnerabilities with their technical root cause in software typically require a fix to eliminate or mitigate the associated security risk. At Nokia, our product creation process prepares R&D to address vulnerabilities in home-grown code and open-source software (OSS).

Using OSS in CNFs naturally increases the number of attack vectors, and additional measures are required to control them. For example, Nokia’s assured open-source program provides a curated OSS repository that is continuously scanned, updated, and made compliant with the Secure Supply Chain Consumption Framework (S2C2F). This is a specific realization of an internal CD model that supports product engineering teams in receiving timely, secure updates to OSS components commonly used across CNFs. This culture extends from development to system verification, where security scans are run continuously for every new delivery.

However, as new vulnerabilities are discovered, CNI must be promptly patched. Nokia and our customers are not new to deploying corrections to live CNFs in production. It is the fundamental delivery practice underpinning standard SW maintenance services. The key changes in the security context are the heartbeat (frequency of deliveries) and the type (content scope) of deliverables. It is not unusual for CSPs to run regular security scans of their installed base on a monthly or bi-monthly basis. Core Networks serve hundreds of thousands, if not millions, of subscribers, so the stakes are incredibly high in detecting and mitigating risks. Scans and the corresponding assessments typically happen at fixed intervals. An inflow of reported vulnerabilities and their criticality dictate the heartbeat of patch releases. SLAs determine the availability of SW corrections, i.e., how fast product engineering develops and releases a fix for commercial deployment.

The type of software deliverables is changing, too. Instead of maintenance packs providing bug fixes or service packs that bring functional improvements, security vulnerability patches are designed to close only the agreed-upon security vulnerabilities. This ensures the upgrade process is as fast as possible, minimizing any non-essential software changes. This may seem contradictory to the generic CD principle, but the speed and safety of deployment takes precedence. In this case, CSPs and Nokia focus solely on lightning-fast deployment of a security vulnerability patch. The Software Bill of Materials (SBOM) is crucial for identifying impacted system domains among proprietary microservices and OSS components. What makes this cadence a truly continuous delivery model is an operator’s capability to absorb consecutive security fixes and Nokia’s capability to release them any time.

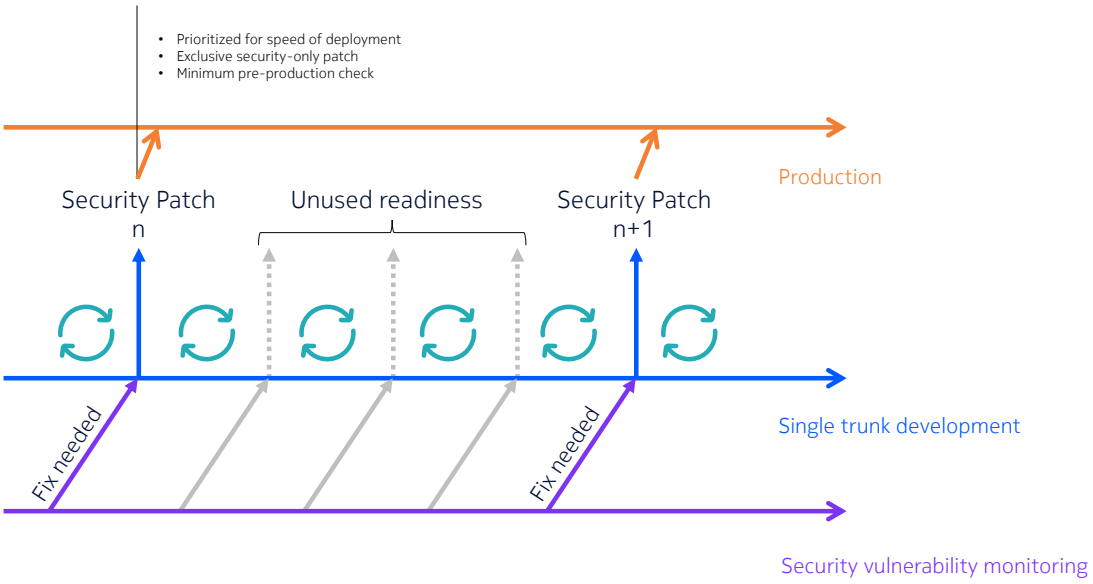

Figure 1 – Continuous security vulnerability maintenance optimized for speed of deployment

For this to happen, all organizations throughout the product value stream must adhere to CD ways of working. Nokia R&D releases security updates outside of the regular cadence. This is enabled by a couple of CD principles:

- Single-trunk development discipline is in place.
- Test cases other than exploratory tests are automated.
- Engineering teams adhere to cloud-native application development principles and in-service upgrade requirements.
- A product security culture is mature, ideally demonstrated by certifications such as the Network Equipment Security Assessment Scheme (NESAS).

Product Security Leaders (PSLs) facilitate the process and ensure full traceability of scans, audits, reports, and fixes. PSLs also help our customers evaluate the severity of reported vulnerabilities, considering CNI system architecture. Nokia's software supply chain adds value by digitally signing artifacts, executing anti-malware scans, and archiving, which is often required for regulatory reasons. A CSP, after adjusting its deployment process, can absorb security updates through a fast-acceptance pipeline. It is an operator's choice to take a new patch, but there is a strong incentive to do so regularly and therefore stay on the supported in-service upgrade paths.

The Nokia Session Border Controller (SBC) is a product of Nokia's IP Multimedia Services (IMS) portfolio. It exemplifies how all these aspects come together in CNFs that provide essential signaling and media planes for voice communications. SBC product engineering releases security vulnerability patches on a continuous basis. Such releases go through the same digital supply chain used for feature deliveries in major releases or service packs. Traceability and reporting are inherited from the security vulnerability management (SVM) process to ensure transparency of software changes and their mapping to security risks. PSLs work together with product engineering, services, and customers to prioritize necessary fixes and identify architecture areas for continuous improvement. Standardized assessments, development practices, and the digital supply chain create a CD framework for continuous security updates, enabling fulfillment of SLAs.

Feature-Driven Use Case

Traditional upgrade cycles in telecom prioritize CNI stability, reliability, and availability. For example, upgrading Core Network elements such as Nokia SBC on a bi-annual basis is considered traditional. Such projects are also believed to be easier from a capital expenditure (CAPEX) planning perspective. It involves preparations, lab testing, acceptance, gradual production rollout, field testing, and mass rollout. These activities have clearly defined start, resources, and exit criteria. These are mostly sequential, with the outcome of CNF running a new major software version.

New product introductions (NPI) often require adjusting this delivery cadence. A framework that serves major releases is not fast enough to govern the exploratory introduction of new interfaces or microservices meant to drive innovations in subscriber services. The word "exploration" may sound inappropriate in the CNI context, but this is still a highly controlled exploration that doesn't impact service quality.

Feature introduction is a classic example of CD. The IMS portfolio mentioned earlier demonstrates the application of CD in the context of 5G SBA architecture evolution. Take the case of Voice over New Radio (VoNR). VoNR is a 5G-native voice service that operates entirely within a 5G Standalone (SA) network. SBA requires IMS to interact with 5G Core (5GC) functions via HTTP-based APIs. The shift from legacy Diameter-based interfaces to HTTP-based APIs required several changes in the IMS P-CSCF function provided within the Nokia SBC product. Such a system architecture change was a perfect CD candidate for several reasons:

- The new interface and its commercial functionality were critical enough to diverge from the standard release model.
- The new architecture required the development of new microservices.
- The new solution needed an in-service upgrade capability from a previous release for ease of deployment.

SBC product engineering applied several CD practices, namely:

- Left-shifting multiple test activities to the earliest feasible development phases , which significantly reduced the use of intermediary test laboratories at the customer interface. The multi-stage continuous integration pipeline enabled product teams to identify defects early.
- Maximizing the use of simulators by SBC software engineers.
- Introducing chaos engineering to ensure the resiliency of newly developed microservices and interfaces. This complemented a comprehensive capacity, performance, regression, and security testing suite.
- Delivering in the DevOps style. R&D collaborated with NPI specialists, who participated in exploratory testing and gained the necessary knowledge to deploy and operate new microservices and interfaces

in the customer environment.

- Leveraging open-source and inner-source practices. SBC product teams focused on IMS-specific implementation while reusing a sidecar microservice owned by another product unit. Continuous development of this sidecar was encapsulated within SBC's development cycles.

The customer was convinced of the need for frequent SW deliveries and adjusted the acceptance process accordingly. This commitment enabled Nokia R&D to reuse the existing digital supply chain for bi-monthly deliveries directly to the customer environment.

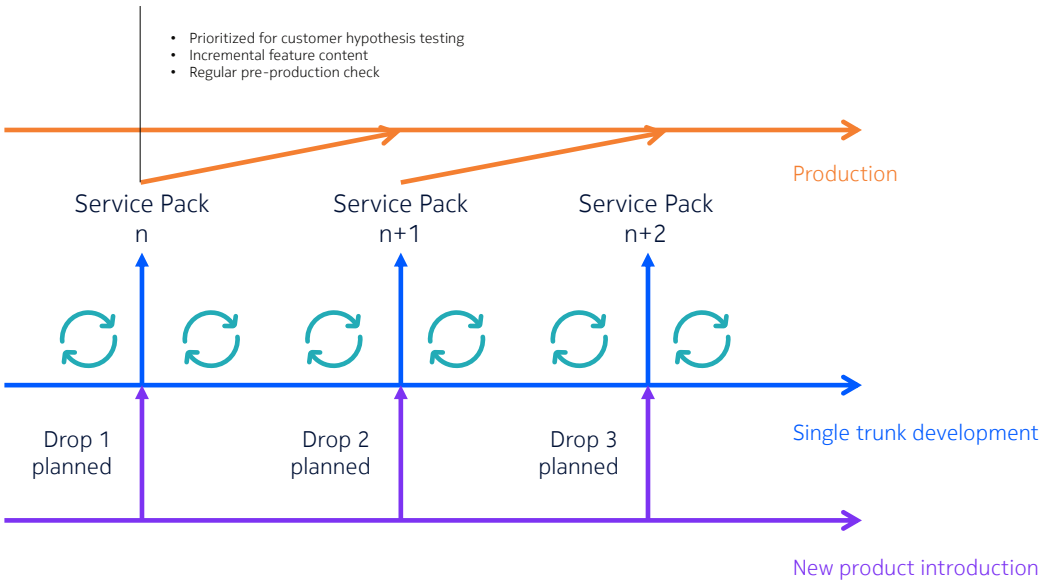

Figure 2 – Continuous product introduction optimized for feature exploration

Conclusion

As demonstrated through both use cases, adopting CD principles to meet the unique CNI challenges paves the way for resilient, agile, and secure telecom solutions. The key takeaway is clear: Continuous Delivery isn't just about faster software updates, it's a crucial enabler of truly Autonomous Networks. By embracing CD principles, CSPs and Nokia can:

- Respond rapidly to security threats through automated, streamlined delivery processes.
- Deploy new features and capabilities with minimal to no disruption.
- Maintain reliability and availability while accelerating innovation in both subscriber services and network operations.
- Lay the foundation for fully Autonomous Networks.

The increasing complexity of critical networks transforms NPI from long-cycles, risk-accumulating projects, to CD. CD enables CSPs to verify network function features and gradually introduce competitive services without compromising the overall subscriber experience. Traditionally, regulatory requirements are perceived as obstacles to fast-paced CD as they incentivize stability over agility. However, CSPs should

recognize security vulnerability management as a significant driving force for applying CD in software maintenance contracts.

At Nokia, CD is a delivery practice essential to the Autonomous Networks vision and SaaS model. CD logistics may vary depending on the use case, but there are several key enablers across the product value stream, many of which go beyond software engineering. Single-trunk development and continuous releases must translate into fast-paced, continuous deployments that address commercially viable use cases and complete the end-to-end CD of critical network infrastructure.

Abbreviations

CD	Continuous Delivery
TTM	Time-to-market
CM	Configuration Management
LCM	Life Cycle Management
B2C	Business-to-consumers
SLA	Service Level Agreement
CNF	Cloud Network Function
SaaS	Software-as-a-service
OSS	Open-Source Software
SBOM	Software Bill of Materials
PSL	Product Security Leader
SVM	Security Vulnerability Management
SBC	Session Border Controller
VoNR	Voice over New Radio
IMS	IP Multimedia Subsystem
5GC	5G Core
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
P-CSCF	Proxy Call Session Control Function
CNI	Critical Network Infrastructure
CSP	Communication Service Provider
CI	Critical Infrastructure
SBA	Service-Based Architecture
R&D	Research & Development
NESAS	Network Equipment Security Assurance Scheme

NPI	New Product Introduction
S2C2F	Secure Supply Chain Consumption Framework
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
TSR	Telecommunications Security Requirements
VNF	Virtualized Network Function

References

[1] K. Beck, M. Beedle, and A. van Bennekum, "Manifesto for Agile Software Development." [Online]. Available: https://agilemanifesto.org/

[2] CISA, "Critical Infrastructure Sectors." Accessed: Jan. 02, 2025. [Online]. Available: https://www.cisa.gov/critical-infrastructure-sectors

[3] Official Journal of the European Union, Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive). 2022. Accessed: Nov. 14, 2023. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2022/2555/oj

[4] C. M. and S. Department for Digital, Telecommunications Security Code of Practice. 2022. Accessed: Nov. 15, 2023. [Online]. Available: https://www.gov.uk/government/publications/electronic-communications-security-measures-regulations-and-draft-telecommunications-security-code-of-practice

Nokia OYJ
Karakaari 7
02610 Espoo
Finland
Tel. +358 (0) 10 44 88 000

CID 215073