UNEXT
A unified networking experience

White paper

Azimeh Sefidcon, Csaba Vulkán and Markus Gruber

Looking at future networks, their perceived capabilities and services, and their increasing complexity, we see a need for a unified approach to network software, spread across all network elements, that systematically simplifies network operations. Future networks will rely on resources from a wide variety of technologies and stakeholders to meet growing expectations. This will dramatically increase the complexity of running the network and consuming its services — complexity that can turn into an innovation inhibitor. UNEXT is our vision for a unified networking experience that systemically addresses this challenge.

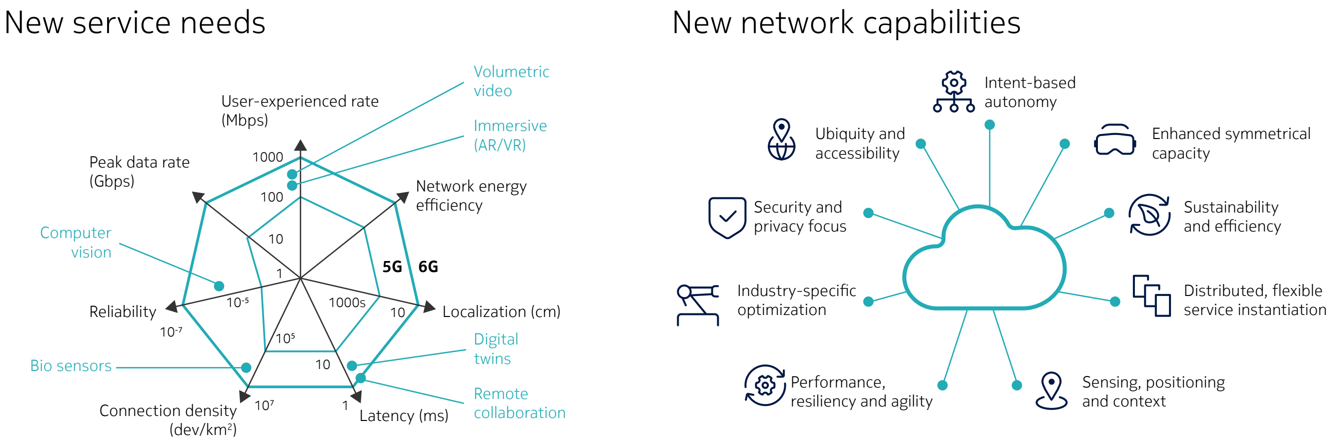
UNEXT decomposes this challenge into sub-problems and wraps solutions into a coherent whole from a software perspective. The goal is to create a secure and reliable experience of the network for operators and users alike. The telecom industry has traditionally lacked a coherent approach to system software, which has made the network increasingly complex to manage and be used directly by enterprise and application developers. The vision presented here offers an alternative, systematic approach to address this growing complexity and create the unified networking experience we need.

Contents

Introduction

Digitalization and communication technologies are transforming businesses and industries, unlocking new value and revenue streams. Future networks will need to be seamlessly integrated into turnkey solutions for vertical industries as they face new service needs and require new network capabilities (Figure 1).

Figure 1. The new service needs and capabilities of future networks.



New service needs

New network capabilities

In future applications like the industrial metaverse, complexities will arise from combinations of physical and digital environments with multiple actors involved in each use case, creating dynamic on-demand value chains, a diverse set of services and expectations, and heterogeneous networking environments.

While the world will require more bandwidth and lower latency from the network in terms of connectivity, the type of services, applications and actors that are connected by the network will pose an additional set of needs that must be accommodated. They will need compute resources, data access, and specific traffic requirements, as well as a new level of autonomy in their interactions with each other and with the network. The network will need to be more adaptable, flexible and robust.

Simplicity and security drivers

Complexity and new security threats usually slow down the acceptance of new telecommunications technologies. Complexity related challenges include:

- Heterogeneous systems with ongoing operations using legacy elements
- A variety of new requirements concerning timescales, performance and power
- Shortages in experts to configure and operate new technologies
- Increasing complexity for maintaining operational updates and bug fixes.

Similarly, on the security side challenges include:

- Individual control of personal data
- Privacy concerns and potential unwanted violations
- The risk of losing mission-critical data (for enterprise users)

- Maintenance of security patch updates
- Control over multiple access points in increasingly complex systems

To summarize, many services that are already extensively on demand today cannot satisfy the security and simplicity criteria necessary to meet the growing scale and density of networks. This will compound in the future given that we are facing an opportunistic, multi-resource approach with additional expectations concerning adaptability, flexibility, and reliability that go far beyond bandwidth and latency. Examples include the needs of vertical industries for seamless integration of their turnkey solutions, as well as many more combinations of physical and digital items, actors, and services in the metaverse. Given these requirements, many of the new 6G requirements are automation related. These automated operations will require intent-based autonomy, distributed, flexible service instantiation, and industry-specific optimization.

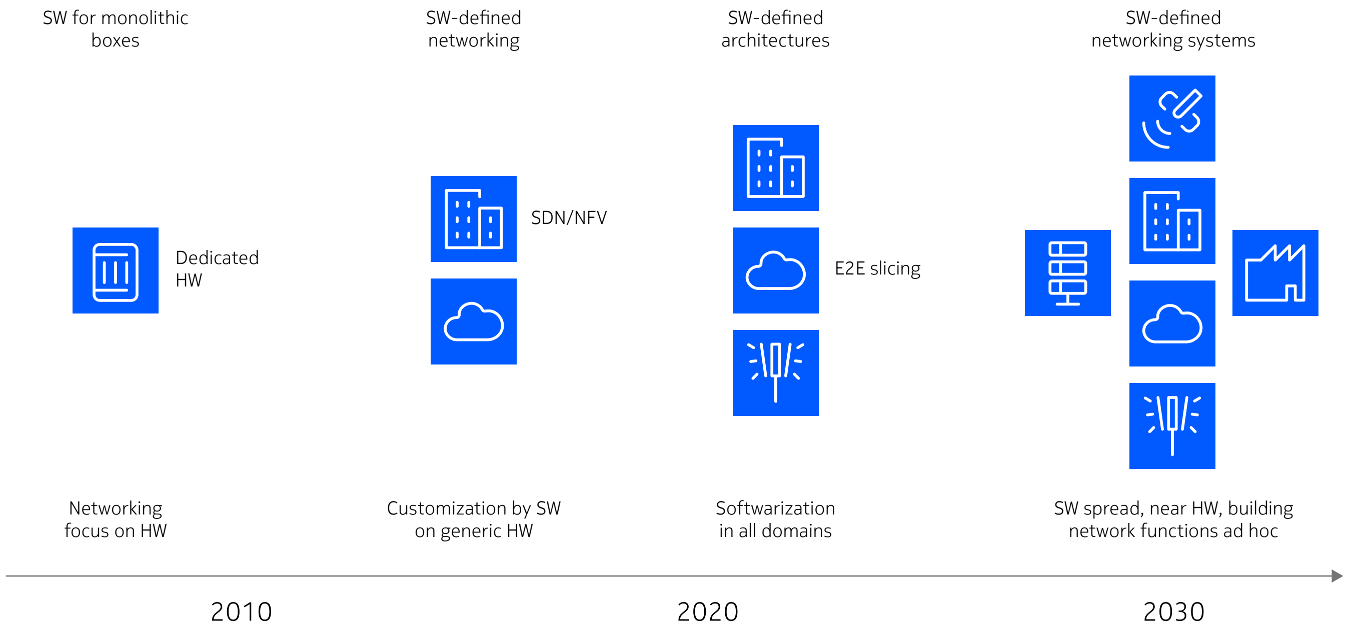
Current trends in unification

Mobile networks are becoming true software (SW) systems, implementing both operational functions and management capabilities as a collection of SW modules interworking via application programming interfaces (APIs). This is particularly necessary because future networks will not only converge fixed and mobile networks but, in the long term, will also integrate non-terrestrial, private and personal networks. APIs will make governance by external systems possible by providing network services and communicating information on operational states throughout these heterogenous networks.

The evolution of telecom hardware (HW) and SW systems is ongoing in several dimensions. First, the evolution of SW technology, especially virtualization and cloud technologies, has enabled the creation of services implemented by HW-independent scalable SW modules. Second, the technologies used by these SW modules, such as programming languages, SW libraries, inter-process communication protocols, and parallel computing architectures, and the associated SW development and management practices, such as test automation, Continuous Integration / Continuous Delivery (CI/CD), canary testing, etc., have also evolved by incorporating novel mechanisms from the IT and webscale domains. Third, AI and machine learning (AI/ML) technologies have been adopted for automated data transformation, context and system modeling, insight generation and increased network autonomy via closed-loop decisions and actions.

The consequences of softwarization and the uptake of AI/ML go beyond pure implementation and network deployment aspects. By using these technologies, telecommunication networks both become and are required to be more flexible, self-organizing and autonomous. They compose AI models of their own capabilities to expose capability driven semantic interfaces. These improve network consumability, allowing natural integration with enterprise solutions and exchanges with future applications like the industrial metaverse. They natively use AI for functions and autonomous management. They dissolve network boundaries by extending network capabilities via applications that natively become part of the network as a system. As well, they create network digital twins that interact with digital twins of devices, applications, workflows or entire production systems.

Figure 2. The evolution of networking SW


SW for monolithic boxes
SW-defined networking
SW-defined architectures
SW-defined networking systems
Dedicated HW
SDN/NFV
E2E slicing
Networking focus on HW
Customization by SW on generic HW
Softwarization in all domains
SW spread, near HW, building network functions ad hoc
2010
2020
2030

The transitions we have seen in networking SW in the last few decades were seismic (Figure 2). We started out with monolithic boxes in the center of everything and SW written explicitly for them. SW-defined networking (SDN) then demoted hardware and its capabilities to a commodity relying on SW to add the needed intelligence. This kind of softwarization was confined to the transport domain, a relatively limited and manageable part of the network. Thus, the evolution to a service-based architecture, with end-to-end (E2E) slicing as an instance of it, was a logical continuation of this trend, expanding softwarization beyond domain limits such as transport and providing a true E2E approach. For E2E slicing the architecture of the entire network must be programmable. For instance, specific architectural elements of the core network also need to be scaled up or down in an automated way depending on the requirements of the slice.

The next logical step is to make the entire network systems SW defined, where systems include combinations of multiple resources, providers, elements of vertical industries as well as physical and digital elements of the metaverse. With this kind of softwarization everything becomes composable and more services are possible. SW-defined network systems provide network functions and services on the fly with SW elements using resources from different providers that are not necessarily arranged in hierarchical relationships.

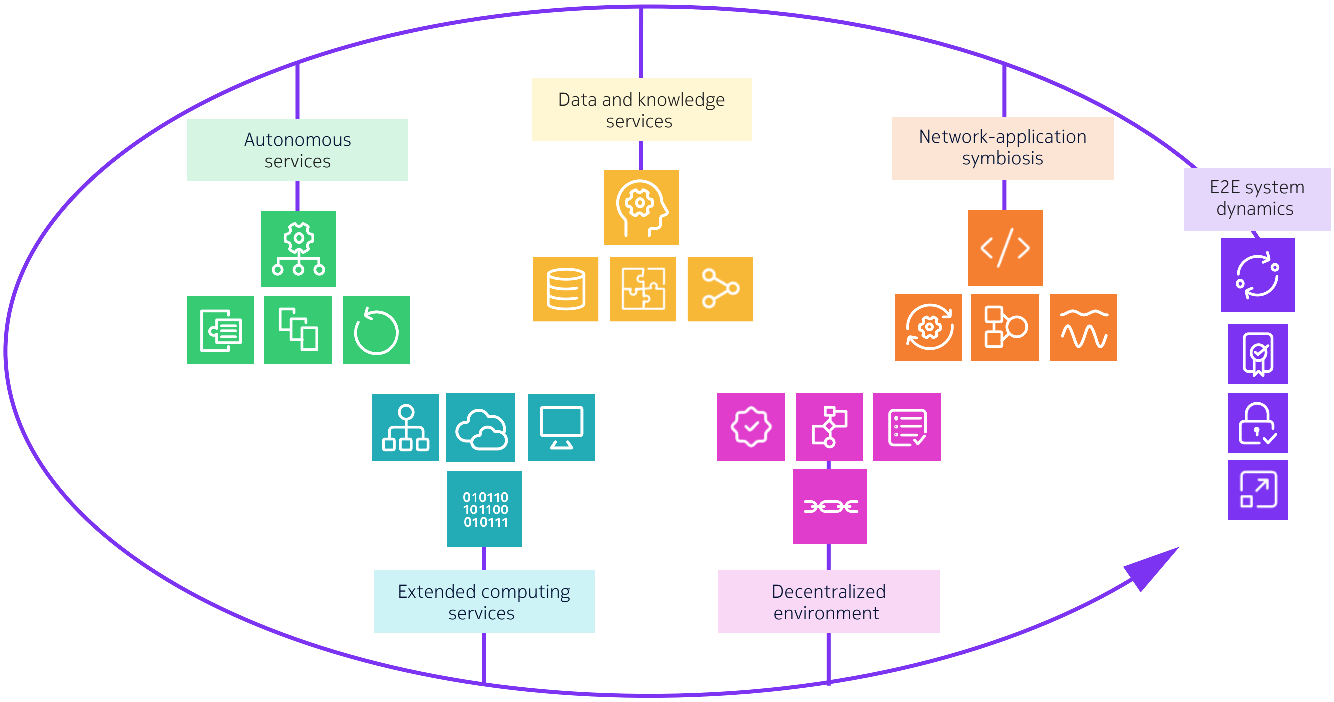
This approach unifies disparate SW and resource elements to optimize the experience of both the operator and the user. We call this unified networking experience UNEXT. UNEXT makes telecommunications technology interoperable with other domains such as vertical industry technologies, and thus serves as a key enabler for the metaverse.

Our response: a unified networking experience

In the complex future of a multi-player, digital-physical world and increasing security threats, UNEXT will be a reliable, intelligent networking platform with composable and decomposable services. It will securely orchestrate compute, data and network elements across heterogeneous environments. As a SW-defined extendable networking system, network functions and applications co-exist and may co-operate (via API exposure mechanisms) on the same technology stack and compute continuum across a multi stakeholder environment. From day one, the technology stack on top of the infrastructure layer should be scaled up to support this highest level, which will also enable services to be run at any point in the extended compute continuum.

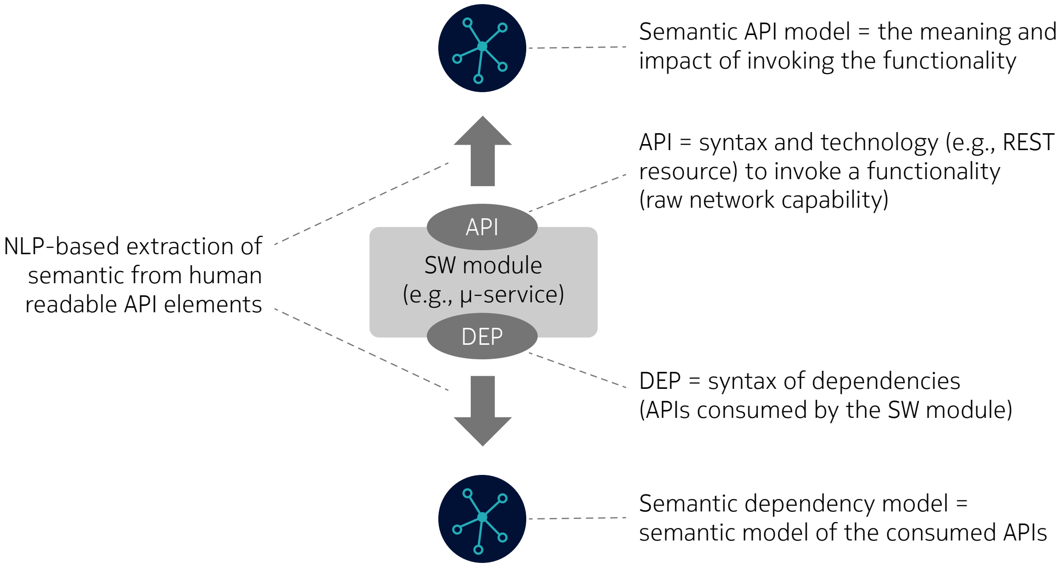
The intrinsic capabilities of UNEXT are defined by the SW modules implementing its constituent network and management functions, data collection and storage modules, automation closed loops, AI models, etc. The capabilities exposed to various users and ecosystem partners in the value chain can be orchestrated to provide a custom combination of the network's intrinsic capabilities. This is a more versatile, flexible and individually optimized approach than can be achieved with a collection of pre-defined network deployment blueprints.

Figure 3. The UNEXT components


Autonomous services
Data and knowledge services
Network-application symbiosis
E2E system dynamics
010110
101100
010111
Extended computing services
Decentralized environment

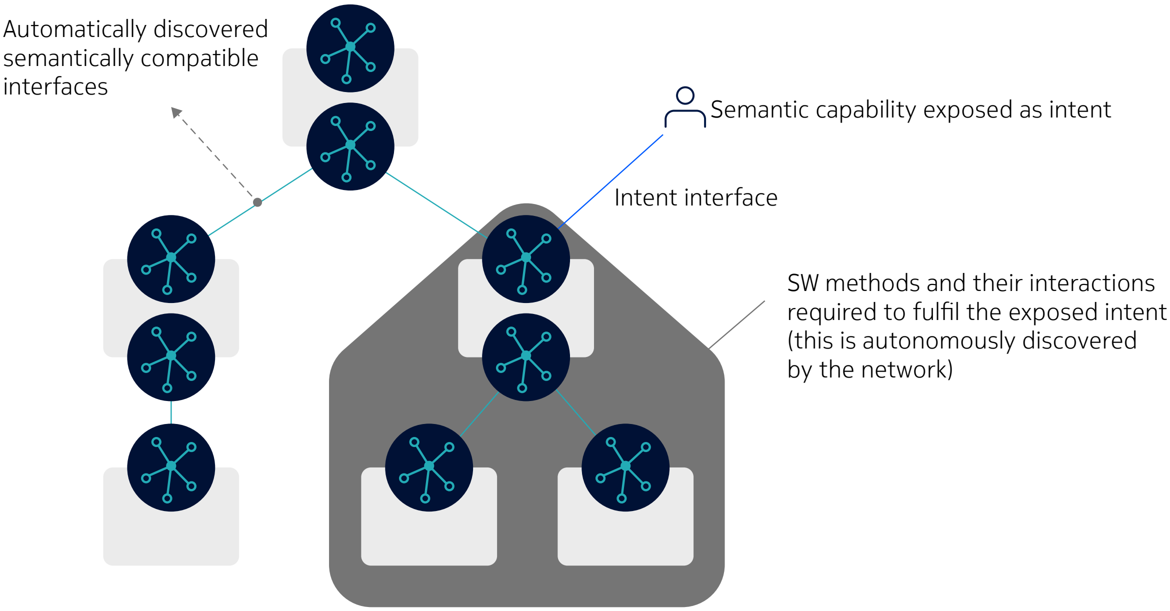
The dynamic synthesis of intrinsic network capabilities into a semantically meaningful combination requires the network to semantically model its own constituent SW modules, covering their functional model, published interfaces (APIs), and dependencies. The combination of two or more SW modules yields a semantically valid bigger unit only if their integration is done via semantically (not only syntactically) compatible APIs (Figure 4). Such semantically compatible interfaces are self-discovered by the network system and updated every time the available SW modules change (e.g., new modules are onboarded or new SW module versions are released).

Figure 4. Semantic model of APIs and network capabilities


Semantic API model = the meaning and impact of invoking the functionality
API = syntax and technology (e.g., REST resource) to invoke a functionality (raw network capability)
NLP-based extraction of semantic from human readable API elements
API
SW module (e.g., μ-service)
DEP
DEP = syntax of dependencies (APIs consumed by the SW module)
Semantic dependency model = semantic model of the consumed APIs

The exposure of combined network capabilities is best achieved via intent-based interfaces, where the interface mechanisms include natural language-based interactions between the user and the system. Ideally, these resemble human dialog to accurately capture an intent that both reflects the user's objectives and the network capabilities (Figure 5).

Figure 5. Network capability exposure through intent interfaces


Automatically discovered semantically compatible interfaces
Semantic capability exposed as intent
Intent interface
SW methods and their interactions required to fulfil the exposed intent (this is autonomously discovered by the network)

UNEXT enables new E2E system dynamics through a collection of system-wide functional capabilities including creation of services autonomously, managing decentralized environments with dynamic trust, establishing and exploiting extended computing services, driving knowledge and creating data services, as well as capabilities for network-application symbiosis.

Areas of unification

Creating services autonomously

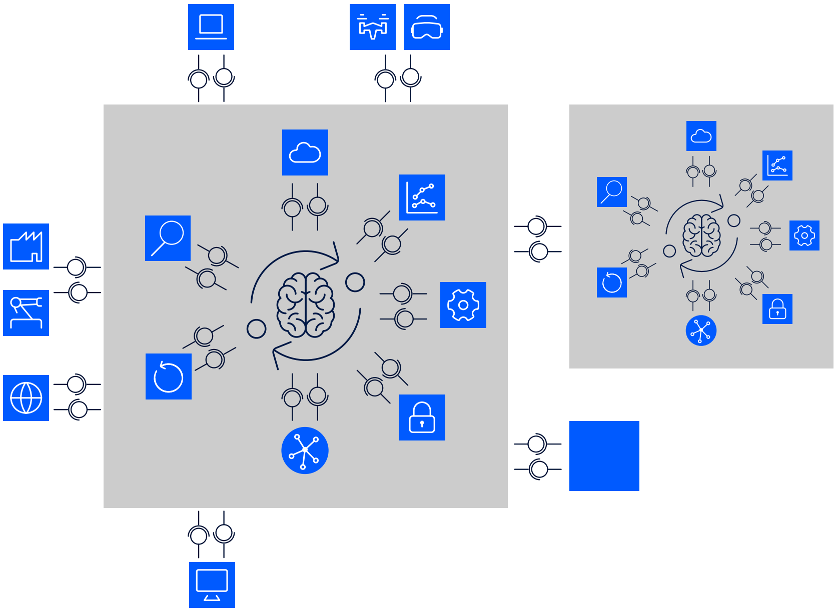
The first set of requirements for the UNEXT system are around composing and executing autonomous services:

- Accommodate different abstraction levels, user expertise and knowledge domains
- Compose services from diverse providers in a multi-stakeholder/multi-tenant environment
- Secure reliability of the composition process by mitigating any semantic ambiguities between intents and descriptions of available capabilities.

The UNEXT autonomous services function covers everything that is needed to compose and orchestrate SW modules to deliver a requested service. A "service" can be anything, from a network management service to an advanced application using those services; a new service can also be composed from existing services. A service can be consumed by expressing intents (intents being defined as declared objectives or states that the system or any of its constituents must be in). In the spirit of UNEXT, all services are composed of SW modules (represented by grey entities in Figure 6) that all follow the same microservice-compatible principles, namely:

- SW modules offer services through APIs and consume services of other SW modules through APIs of those modules
- Producers and consumers of these services can be at any abstraction or technology stack level (e.g., APIs provided by a chipset within a phone, APIs consumed by a GUI with human users)
- Service APIs are declarative and are consumed through expressing intents (i.e., an objective, or desired state, or outcome is declared) rather than operations or workflows, and services assure that the system the service is acting on constantly fulfills the objective expressed in the intent.

Figure 6. Unified SW architecture for simplicity: different icons epitomize the large variety of elements that can be intelligently connected.



While there has already been quite some effort to define these interfaces, the mechanics to run various SW modules as a coherent whole can still be enhanced.

Managing decentralized environments

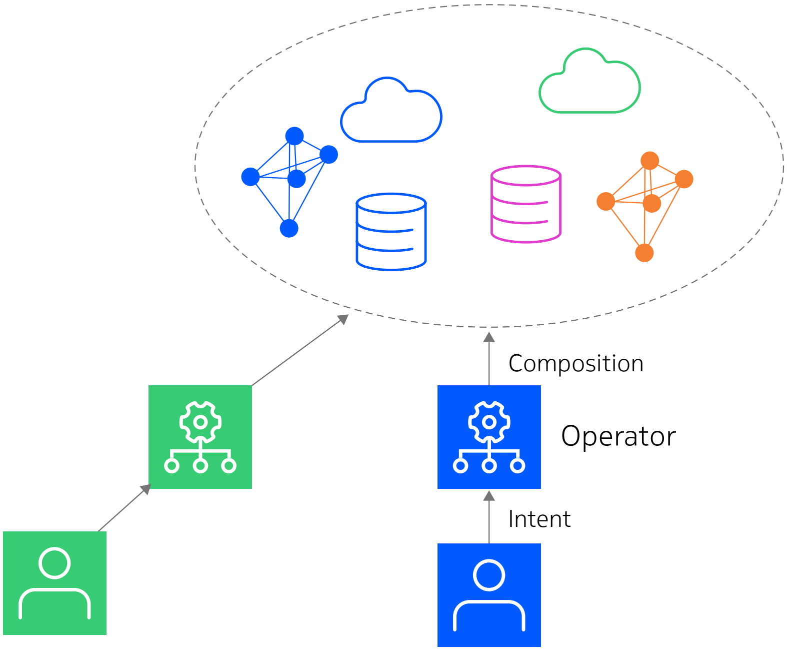
In order to manage decentralized environments with multiple actors where there is no established prior trust agreement, UNEXT will need:

- Orchestration of multiple stakeholders in complex ecosystems
- Ability to address static trust and lack of trust relationships in decentralized ecosystems using trustworthy dynamic orchestration via compositional AI/ML and Web3 technologies.

This UNEXT feature addresses one of the key novelties in 6G, namely the presence of a heterogeneity of stakeholders in the ecosystem. The opportunistic use of resources from multiple parties will require more stringent bandwidth and latency requirements.

Figure 6 illustrates this new ecosystem where different operators can revert to the same pool of resources to compose not only an E2E network slice, but also the corresponding compute sub-cloud. Thus, operators will not only have access to their own resources but also to resources of other operators and vice versa. They could also use compute resources from different cloud providers depending on the current hardware and latency needs. In this spirit, network and compute resources alike are just modules of a richer resource system.

Figure 7. Trusted orchestration of decentralized environments



This comes with three key challenges:

- The integrity of system modules needs to be ensured
- The ownership of data, models, and AI entities needs to be able to be verifiable
- Trust relationships between modules need to be understood.

Extending the compute services

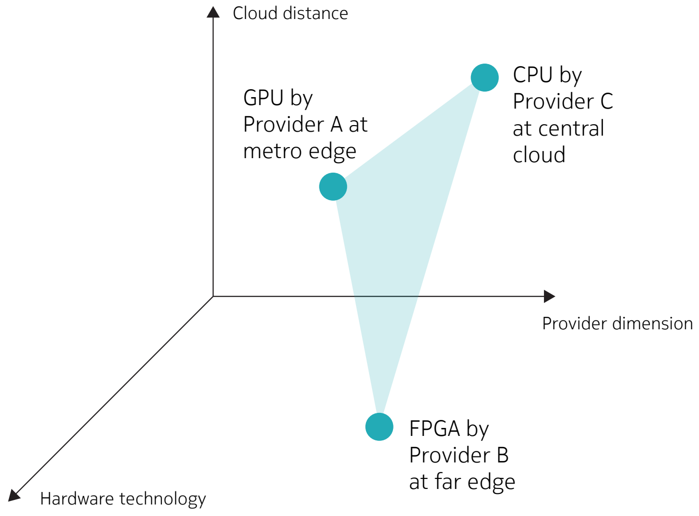
Another set of requirements are focused on distributed computing environments across multiple stakeholders' infrastructure:

- Reliable decentralized execution environments based on ephemeral nodes or connectivity
- Unified placement strategies covering network, data and compute
- Novel application description models.

Network-compute convergence is not a new topic, but the unification principles we are promoting sets it in the context of larger complexity. In its simplest form, network-compute convergence focuses on providing a network service and tying in compute resources from a cloud, whether edge or central. General cloud apps, however, provide the same single, most of the time, stateless function to all endpoints. In contrast, telco-oriented E2E stateful applications are more complex, especially when a heterogenous compute continuum with resources belonging to various domains needs to be formed in order to distribute the elements of the application and network them. For instance, it is conceivable to have a service that runs part of its resources on the edge cloud because of latency requirements, other parts on the central cloud for broader state sharing, and others on extreme edge devices for privacy or other needs particular to a specific application or use case.

This section discusses how we envision UNEXT orchestrating individual elements of a given service on resources with heterogeneous capabilities over a generalized compute continuum. For UNEXT, the compute continuum has three dimensions as illustrated in Figure 8: the "cloud distance" (central, edge, etc.), the "multi-provider" dimension, and a "hardware technology" dimension. The latter considers processing resources that are the most appropriate for a given service. The choice of what type of compute resources to use, where and from whom, is then up to the orchestrator, which needs to consider service requirements such as latency, cost, and efficiency as well as trust, security and available compute resource capabilities. A single service may therefore be partitioned into different compute tasks that are spread across different compute elements, providers, and technologies (e.g., hardware acceleration options), eventually forming a service-specific sub cloud in the continuum. We can think of this as: a sub-cloud is to compute what an E2E slice is to networks.

Figure 8. Orchestration along the compute continuum


Cloud distance
GPU by Provider A at metro edge
CPU by Provider C at central cloud
Provider dimension
FPGA by Provider B at far edge
Hardware technology

Leveraging data and knowledge services

The next set of requirements concerns the efficient use of data, and on-demand creation of knowledge and data services across the network, which includes the ability to:

- Sense and collect pertinent data on demand from heterogeneous sources
- Transform data to information using data fusion, dimensionality reduction and semantic extraction
- Ensure security and privacy along the entire process, while enabling trust and monetization among involved sources.

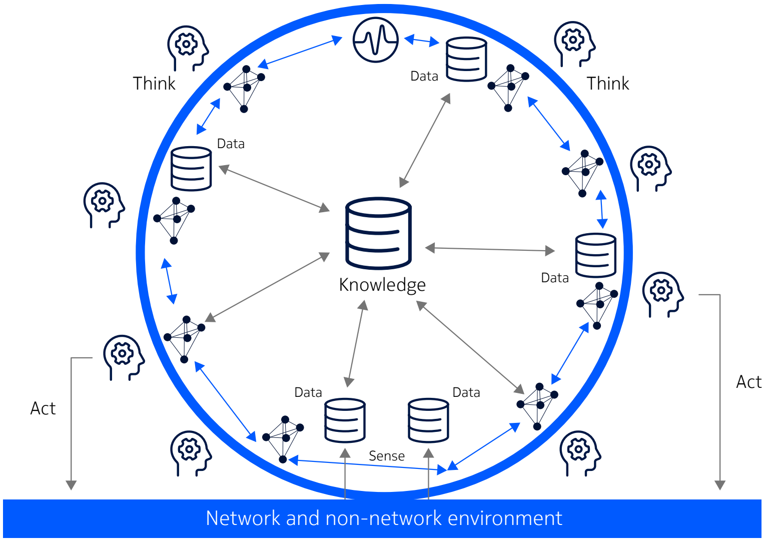
For networks it has become advantageous to use AI to improve, optimize, manage, automate, and enable previously unthinkable innovations. Indeed, 6G networks are being built to be AI-native. Three aspects require particular attention:

1. AI requires an amount of data to be collected both from the network and its environment that may become unsustainable
2. Data for training an AI is usually collected for a specific purpose (e.g., to optimize a process or detect anomalies), and, for many reasons (e.g., organizational separation or data sharing restrictions), these processes are isolated; thus these “artificial brains” work independently
3. Use of AI offers new opportunities for attacks and hence new security mechanisms must be designed to detect, prevent, and mitigate threats in this new landscape.

Thus, we envisage that whenever possible collected raw data should be reduced and processed at the time of collection. Early decisions should be made autonomously to store or not. Furthermore, when processed by an AI, data becomes a machine-usable form of knowledge; it can be the model itself, the weights and features, or the produced insights. This knowledge is more valuable, requires less storage and, if shared, can provide new insights in other contexts.

Applying a unification principle enables the knowledge (and perhaps any intermediary step in the ML processing pipeline) to be an input and basis for AI collaboration. Providing data and knowledge services, thus, becomes the nervous system of the network, as sketched in Figure 9.

Figure 9. Unified AI with knowledge gradually replacing data



Unification does not stop at the borders of the network itself, but also spreads out to non-network environments. For example, when sensing the ambient system conditions from a factory environment, unification enables the specific factory data to be correlated with network data and the AI managing the application to collaboratively act in the best possible manner for the performance of both the network and application.

Just as a service can be anything from network management to an advanced application using those services, this nervous system will provide its services in an intent-based, declarative manner — a network of AIs, managed and constantly evaluated by AIs.

Finally, part of the data and knowledge services will be the ability to trace, prove and verify ownership of any AI asset (e.g., data, AI pipeline and model) and add services that can detect and mitigate adversarial attacks on the AI.

Creating network-application symbiosis

UNEXT places the service front and center and creates enablers that let the network exist in symbiosis with applications. Information is exchanged bidirectionally between the two for their mutual benefit. For UNEXT to create a constructive relation between the network and applications, it requires:

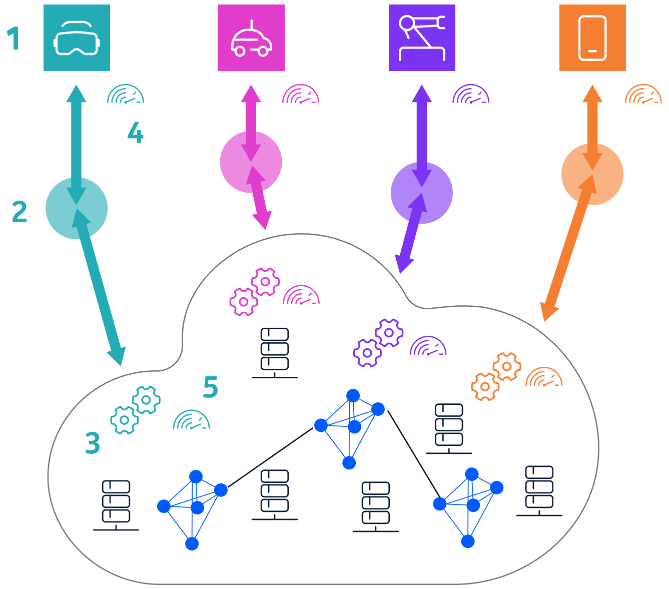
- Operation technology grade reliability (6+ nines) across carrier-grade and webscale technology (3–4 nines)
- Deterministic, SLA-based communications
- In-production testing of SW-defined networks.

The centerpiece of this symbiosis is a UNEXT API that lets any application convey its identity, requirements, quality metrics, network capabilities, and even control commands. Reciprocally, UNEXT can convey resource availabilities and status as well as quality-of-service (QoS) metrics. This interface infuses UNEXT application awareness in every aspect of its operation. Thanks to the API, the application directly contributes to the efficiency of UNEXT.

Figure 10 illustrates the symbiotic coexistence of the network and applications in terms of a series of reciprocal interchanges:

1. As an application is newly deployed, the corresponding UNEXT resources are orchestrated
2. A dynamic, application-tailored API is instantiated, which enables the application to be detected and well understood by UNEXT
3. The resource requirements of the application are enforced by UNEXT
4. The application communicates quality-of-application (QoA) metrics to UNEXT through the API for continuous fine-tuning and orchestration of network and compute resources and control functions
5. UNEXT reciprocally communicates QoS network metrics to the application enabling it to dynamically adapt to the current UNEXT conditions.

Figure 10. Application awareness infuses every step of the application support life cycle



1 New application

2 Dynamic interface

3 UNEXT resourcs for application fulfillment

4 QoA metrics from application

5 QoS metrics from UNEXT

More concretely, network-application interworking is achieved through the extension of semantic API mechanisms. The network capabilities are (selectively) exposed to applications, which can both realize network-aware operation and provide insights to the network (or implicitly “program” the network, depending on the role split between network and application space) so that the network serves the application according to its demands. The increased clarity in the expression of application requirements enables the dynamic instantiation, within the network, of a consumable, customized mesh of network-compute capabilities that only exists while the application is running. The end result is better performance for the application and lower cost of operation for the network.

Emerging differentiators

E2E system dynamics

We believe that by simultaneously addressing the five sets of requirements above, a new set of differentiators will emerge, which would not be the case if the network addressed each requirement in isolation. Some of these properties are:

- Security: various aspects of identity, data, correlation of functionalities and access management provide end user security and privacy guarantees
- Simplicity: with each layer aiming for specific aspects of programmability, the number of layers of management is reduced
- Reliability: security is intrinsically embedded in the design of functionalities while addressing all requirements
- Resiliency: recursive design ensures that the system can be scaled up and down — or clone itself — when required to ensure resiliency.

The five UNEXT features described previously cannot be seen as standalone but work together as part of a unified whole that is more than the sum of its parts. Intents by operators or users are transformed through **network-application symbiosis** into **autonomous services** by the composition of multiple resources in a **decentralized environment** by considering information from disparate **knowledge and data services**, which include **extended compute services**. UNEXT can thus be seen as an operating system of systems.

History suggests that simplification principles such as everything is a file for UNIX (a foundational operating system that underlies Linux, MacOS, Android and iOS), greatly enhance the chances for success of such complex systems. Learning from this history, UNEXT acts as a multi-resource aggregator with the following properties:

- Self-management capabilities
- Semantic interfaces
- Ability to integrate into a communications fabric with other services.

Highly complex systems are prone to racing conditions, instabilities and oscillations. For instance, from a knowledge and data perspective the same data can be consumed by different ML pipelines in an uncoordinated way and with unfavorable outcomes. UNEXT provides appropriate digital twin environments as a sandbox to systematically test new multi-resource configurations in a safe environment before going live.

Simplicity and security by design

There are two aspects of simplicity to be realized:

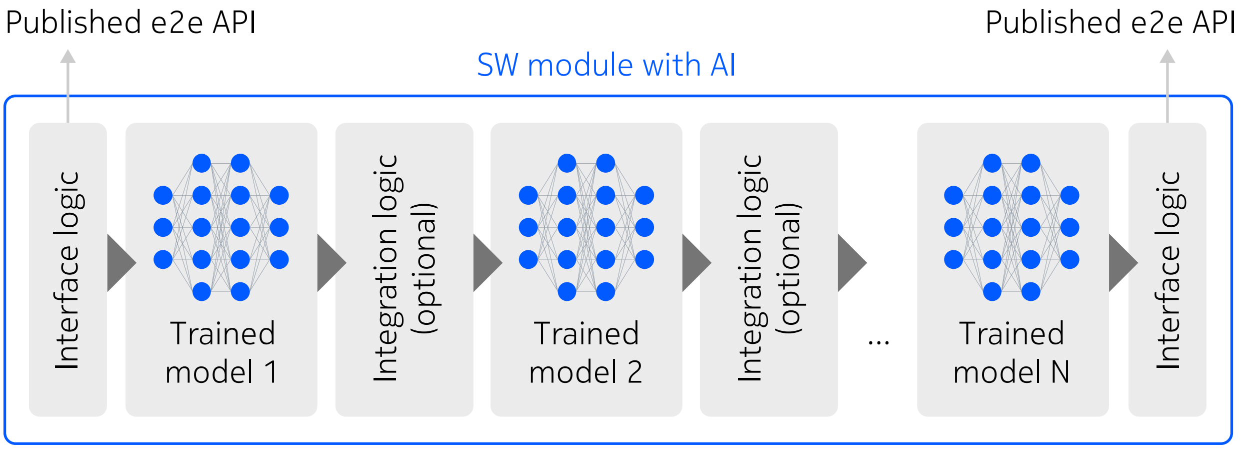
1. Easy consumption of network capabilities by both humans (operators and enterprise experts managing the network) and by machines (applications that can benefit from the extended API capabilities)
2. Optimization of technologies, topologies, and architectures through the adoption of SW, cloud/virtualization and AI/ML (optimization can also include environmental impact, resource efficiency and cost).

As well as easing the consumption of network capabilities for human and machine applications, simplification will also extend to the operator experience with the advent of plug-and-play operations, zero-touch management, autonomous and cognitive network management, and natural language-based interfaces. Flexible, adaptive, on-demand, dynamically defined interfaces and APIs will allow full network service programmability for applications and improve application–network interactions and integration.

There is a set of rapidly developing AI/ML technologies already being applied to various network-related tasks that extract actionable insights from large amounts of data. Often retrofitted components that aim to improve system operations, these technologies are currently additional to and decoupled from existing network and management functions. Data collection, AI modeling, model deployment and lifecycle management are non-standard case-by-case engineering tasks. Thus, the same problems are solved multiple times, each with different tools and to a different extent. The result is non-reusable, non-scalable AI pipelines and interfaces, often dealing with bare AI models that are incubated in non-reproducible lab environments and incapable of being consumed as a service.

Instead, networks need to become AI native. AI models should be just like any other functionality and capability: encapsulated into SW modules to become deployable units with well-defined interfaces providing well-defined services (Figure 11).

Figure 11. AI models integrated into a deployable SW module

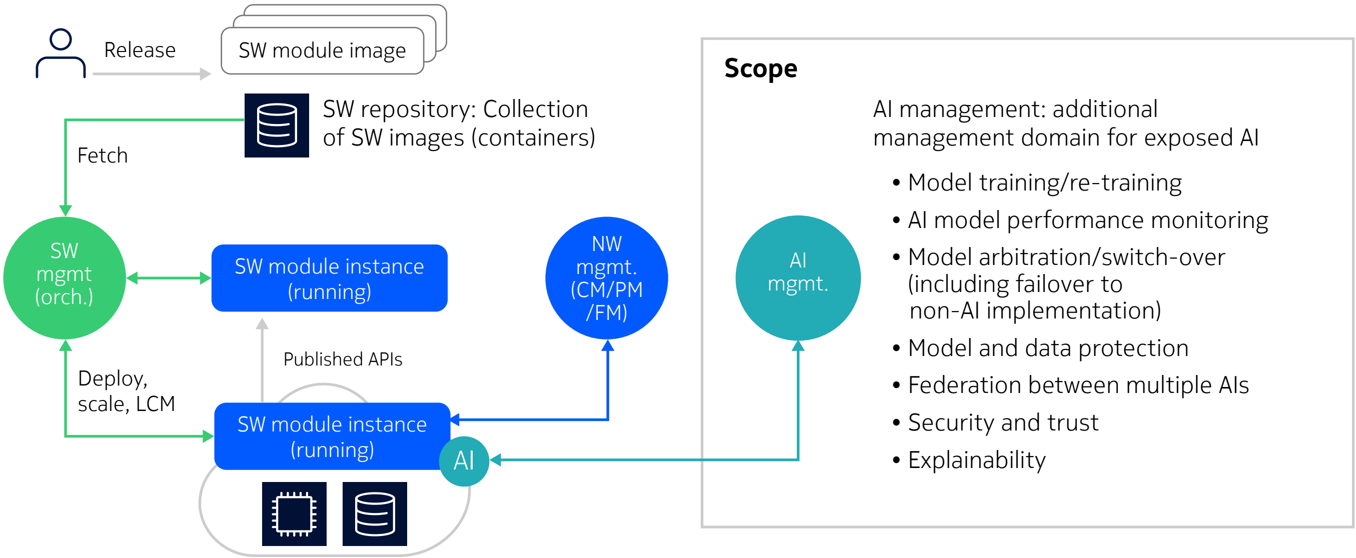


The management of AI-native networks, consequently, will extend to management of AI including (Figure 12):

- Model training and re-training
- Model performance monitoring
- Model arbitration/switch-over (including failover to non-AI implementation)
- Model and data protection
- Federation between multiple AIs
- Security and trust
- Explainability.

These management aspects originate in AI technology and are additional to network and service management aspects (including CM/PM/FM), which relate to the purpose and business logic of traditional network SW modules, as well as SW management and orchestration, which originate in the SW and cloud-native technology used to implement and deploy the SW modules.

Figure 12: Management tasks in AI-native networks


Release
SW module image
SW repository: Collection of SW images (containers)
Fetch
SW mgmt (orch.)
SW module instance (running)
NW mgmt. (CM/PM /FM)
AI mgmt.
Published APIs
Deploy, scale, LCM
SW module instance (running)
AI

Scope

AI management: additional management domain for exposed AI

- Model training/re-training
- AI model performance monitoring
- Model arbitration/switch-over (including failover to non-AI implementation)
- Model and data protection
- Federation between multiple AIs
- Security and trust
- Explainability

SW management (orchestration) + NW management:
Management tasks for any SF module (with/without AI)

With the integration of AI into the network management system, flexibility and autonomy increases — thus the true value of the network for all types of consumers. AI models generated and used by the network may be categorized, as follows.

Network-state models

These capture static or dynamic aspects of select network entities such as equipment, functions, domains and services that are working at different levels of detail ranging from a single physical entity to the aggregation of many entities. They enable the detection and prediction of events such as anomalies, what-if analysis, analysis of the impact and/or efficiency of actions, etc.

Semantic models

These capture the capabilities of individual SW modules (system components) as well as the capabilities that emerge by interworking between multiple modules. Semantic models enable the network to be self-reflective about the types of assured services and actions it may autonomously provide. Pre-defined blueprints are replaced by service implementations that use autonomously and dynamically identified interworking SW modules to unlock network capabilities that were not manually conceived at design time.

Predictive analytical and decision models with self-learning capability

These models provide autonomy in operations, service fulfillment and assurance, and self-management. They have self-learning capabilities, such as reinforcement learning from context-based action impact and efficiency, which enable the automatic adaptation of the network to specific deployments. In this way, it can dynamically respond to the environment without manual pre-configuration of anticipated scenarios that may turn out to be different from what was planned.

Interface models

These models include natural language processing (NLP), large language models (LLMs) and generative AI for human language-based interactions. They can also include enterprise domain specific ontology models for automating the network to service turnkey industrial solutions. Interface models allow the network to become cognitive, i.e., have machine interpretable accurate state information about its users, the demand it serves, and the resources and services on various time scales including historical, real time and predicted. Interface models make possible a fully flexible self-composing network through complete automated rendering of all intrinsic SW capabilities into purpose-driven and dynamically defined closed loops. The result is a set of mechanisms that provides a combinatorically complete repository including all valid combinations of SW modules and their semantically compatible APIs.

A further evolutionary step that builds on the self-composing system capability is network digital twins (NDTs), which extend existing models with their own use-case-specific models and drive them with data collected from the real network to provide new analytics and management automation capabilities. NDTs may be instantiated on demand to run simultaneous virtual experiments on the network without conflicting with each other or with the actual network itself. NDTs may also be integrated into closed-loop automation, in which case they become part of the network's native capabilities.

The network we are envisioning as a system of SW modules and AIs will come to play a fundamental role in delivering critical services provided by future applications like the industrial metaverse, utilities and vertical industries. Network security capabilities, autonomy and response agility must, therefore, be elevated above what is normal for best-effort connectivity. Besides the standard practices of SW and IT security, there should be resilience against threats that target the operational capability of the network and, by extension, any dependent system. Security capabilities must be native to each SW module or networked component, whether implementing a network function, a management service, an analytics or decision model, a data collection and storage service, an AI model or anything else.

Leveraging and orchestrating the various security capabilities of individual components, security is an integrated system responsibility utilizing other key concepts such as dynamic trust composition, reputation management, supply chain security, fine-grained identity and access control, synergistic self-sensing, and situational awareness with appropriate response decisions and their execution. Building in security awareness throughout the system takes advantage of the distributed system architecture and ensures maximum service availability and resiliency even in under-attack systems.

While there are specific security mechanisms required to achieve a unified networking experience, the majority of the end-to-end security questions need to be addressed with a system view. Simplicity in usability and operation is achieved by the elegance of design rules, and generalized intent-based control and interfaces, both at the sub-system/service and global E2E levels.

Unique system properties of UNEXT

To compose and execute services autonomously, UNEXT adheres to the following five principles.

1. Intent-based autonomy

In order to realize the expressed intent, UNEXT must autonomously discover and understand services, meaning the system capabilities that are accessible through service APIs as well as the capabilities of elemental service components. It must also discover and understand states such as resource usage, current and predicted performance, as well as the SW components that provide those services. It must then autonomously select the optimal service composition, which may include components that are provided by different owners. This also applies to elementary services providing orchestration and network management functionalities themselves. The declarative, intent-based management paradigm applies universally to all SW modules down to the device level.

2. Unification

UNEXT must treat all services, regardless of scope, with a unified approach, using the declarative API and autonomous service composition paradigms.

3. Flexibility

New services might emerge automatically if either a component service or an intent changes. UNEXT composition does not follow prescribed blueprints; it adheres only to the desired objective expressed in intents and/or to the changing capabilities and states of the component services.

4. Security

The security of UNEXT services must be inherently included in the service composition procedure.

5. Semantic APIs

Along with being available and discoverable, UNEXT APIs must be semantic, that is, provide a machine-interpretable description of the tasks and capabilities as well as the state and context of the services that are invoked using that API.

Along with the above five principles, UNEXT requires four enabling functionalities for managing the decentralized environment and deploying distributed E2E services and applications over shared infrastructures, resources and devices.

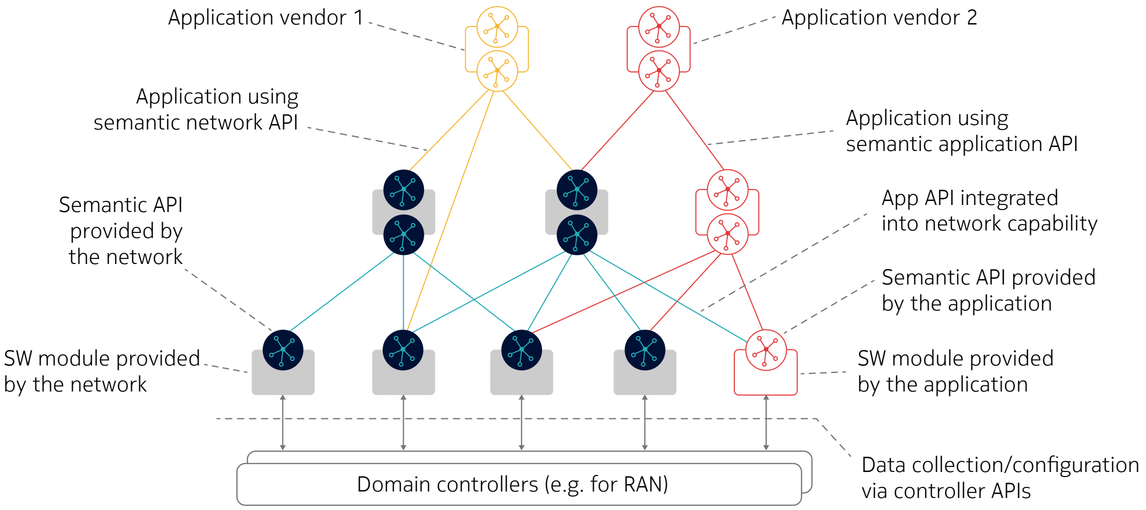
1. **Exposure:** The available assets and resources need to be properly exposed to incentivize sharing
2. **Negotiation:** Cross-tenant intents need to be properly modelled
3. **Conflicts:** Conflicts need to be solved in a fair and automated way
4. **Assurance:** There needs to be a way to monitor and adapt performance across stakeholders.

In the area of extended computing, UNEXT addresses the following:

- Appropriate security mechanisms for the integration of new compute technologies into distributed execution environments (e.g., ensuring secure sharing of FPGA)
- Optimized communications between the compute elements used for a given service depending on performance and/or security requirements
- Ensuring that orchestration of network and compute resources accommodates the scope and level of granularity described in this section.

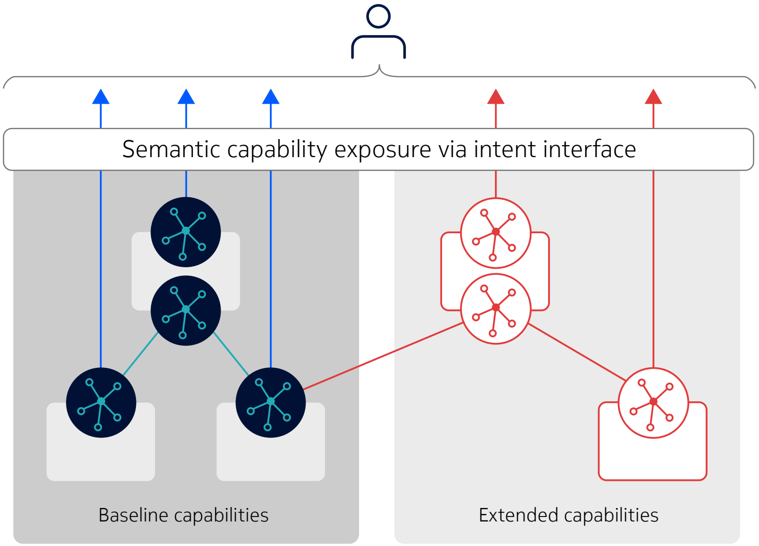
Network-application interworking is achieved through the extension of semantic API mechanisms (Figure 13) that selectively expose network capabilities to applications. This realizes both network-aware operation and provides application insights to the network. Applications can also program the network, depending on the role split between the network and application, so that the network serves the application's needs.

Figure 13. Application embedding via semantic API extension


Application vendor 1
Application vendor 2
Application using semantic network API
Application using semantic application API
Semantic API provided by the network
App API integrated into network capability
Semantic API provided by the application
SW module provided by the network
SW module provided by the application
Data collection/configuration via controller APIs
Domain controllers (e.g. for RAN)

Additionally, the application may publish its own APIs, which are subject to the same semantic modeling as network APIs, thereby creating a semantic API extension over the original network semantic APIs. These additional semantic APIs are usable by other application as well as by network functions themselves, organically combining into an ecosystem of flexible, extendable and interworking SW modules (Figure 14).

Figure 14. Interworking SW modules



Conclusion

The unified networking experience — UNEXT — ensures that a user, whether a mobile subscriber, an application developer or a system operator, easily receives everything they require smoothly and securely. Within this multi-stakeholder environment, users are able to leverage their assets, exercise their legal rights, and engage in monetary exchanges (monetization) with a meaningful ability to make decisions to influence the outcome and reach their objectives. This creates a unifying experience not only for providers and operators but also for the users of their services in a recursive way (power of n), and this will happen for all the parties involved in any use case or scenario.

The name we are proposing, UNEXT, emphasizes unification from one side and experience from the other. It connects and unifies actors across the network, while recognizing that each deserves a unique experience based on their objectives. Conceived as a kind of ecosystem-level operating system, we like the echoes of UNIX in the name UNEXT, as well as the suggestion of future.

UNEXT addresses the dramatic complexity crunch the metaverse community is facing when employing multi-resource solutions from disparate parties. UNEXT provides the cohesion needed for a uniform approach that has the potential to address the complexity challenge. As presented in this paper, unification can act as a catalyst to accelerate the transformation from an era of slow technology adoption to an era of cross-domain technology, inspiration and innovation. The service-based simplification principles underpinning UNEXT may serve as a long-term foundation to build systems of systems in a continuum that is scalable beyond what we can imagine today.

Abbreviations

AI	Artificial Intelligence
AI/ML	Artificial Intelligence / Machine Learning
API	Application Programming Interface
CI/CD	Continuous Integration/Continuous Delivery
CM	Configuration Management
E2E	End-to-End
FM	Fault Management
FPGA	Field Programmable Gate Array
GUI	Graphical User Interace
HW	Hardware
IT	Information Technology
LLM	Large Language Model
ML	Machine Learning
NDT	Network Digital Twin
NFV	Network Function Virtualization
NLP	Natural Language Processing
PM	Performance Management
QoA	Quality of Application
QoS	Quality of Service
SDN	SW Defined Networking
SLA	Service-level Agreement
SW	Software

References

1. Nokia, Discover 2030, Our technology vision for a future where realities merge, https://www.nokia.com/innovation/technology-vision-2030/
2. S. Mwanje and C. Mannweiler (eds), Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond, John Wiley & Sons, 2020.
3. Nokia Bell Labs, Envisioning a 6G future, https://www.bell-labs.com/research-innovation/what-is-6g/6g-technologies/

About Nokia

At Nokia, we create technology that helps the world act together.

As a B2B technology innovation leader, we are pioneering networks that sense, think and act by leveraging our work across mobile, fixed and cloud networks. In addition, we create value with intellectual property and long-term research, led by the award-winning Nokia Bell Labs.

Service providers, enterprises and partners worldwide trust Nokia to deliver secure, reliable and sustainable networks today – and work with us to create the digital services and applications of the future.

Nokia is a registered trademark of Nokia Corporation. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

© 2023 Nokia

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

Document code: CID213573 (November)