NOKIA BELL LABS

Extended compute services for a unified networking experience

UNEXT White paper series

Mathieu Boussard

Nokia Bell Labs is pioneering the next frontier in intelligent connectivity with the creation of the Unified Networking Experience Technology (UNEXT) platform. UNEXT is an innovative networking solution designed to seamlessly unify diverse network and compute capabilities into a single, intelligent ecosystem. By abstracting complexity, orchestrating heterogeneous technologies and enabling end-to-end services, UNEXT will deliver a streamlined experience for stakeholders across the value chain.

With its foundation in cutting-edge research and decades of expertise, Nokia Bell Labs UNEXT aims to redefine how networks operate, making them more adaptive, efficient and capable of supporting the demands of a hyper-digital world.

Realizing services end-to-end in UNEXT will require defining the abstraction, orchestration and support for heterogeneous network and compute capabilities across a wide range of stakeholders and technologies. This paper describes the foundations of the resulting secure, generalized network-compute continuum, which will serve as UNEXT's execution environment.

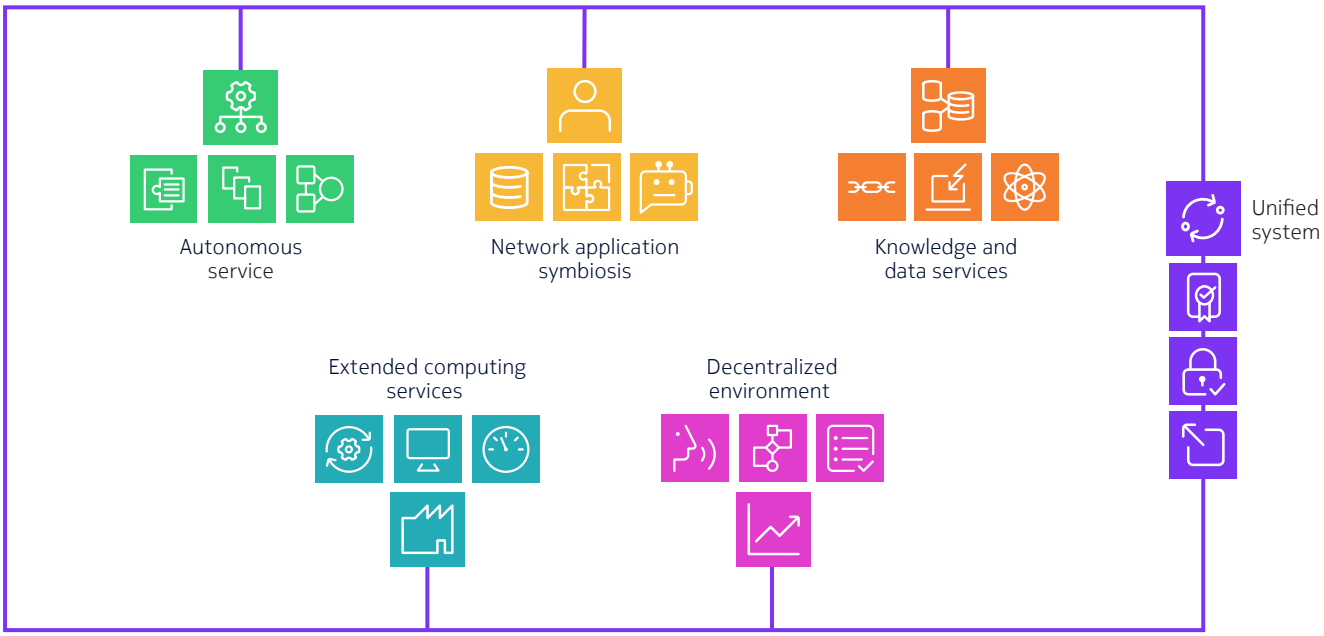
Contents

Introduction

Network systems have become foundational to our modern societies. They are expected to continue to gain importance, evolving in nature and functional scope while incorporating an increasing number of both technologies and stakeholders. This will exacerbate the complexity of both running networks and consuming digital services. UNEXT (Unified Networking Experience Technology, [Sef23]) is our vision for a unified networking experience that systemically addresses this challenge by supporting secure and trusted interactions across stakeholders and seamless service, network and compute orchestration over the resources they each contribute to the system.

To accomplish this ambitious goal, UNEXT proposes a novel approach to creating simple, secure and scalable services based on composable autonomous software agents. UNEXT decomposes the challenge into six main sub-problems (dealt with in as many "focus areas," Figure 1): defining the foundational design rules and autonomous agents signature and behavior (Unified System); enabling interactions between modules of various stakeholders with different goals and requirements, forming de facto a decentralized system (Decentralized Environments, [atk24]); supporting the flows of data and knowledge in the system (Knowledge and Data Services, [con24]); supporting autonomy in system operation and service realization (Autonomous Services); supporting orchestration of network and compute resources across space, stakeholders and technologies (Extended Compute Services); and supporting tight interaction levels between networks and applications (Network-Application Symbiosis).

Figure 1: UNEXT focus areas



This whitepaper describes the context, challenges and proposed approaches related to the Extended Compute Services (ECS) research area. ECS aims to provide UNEXT's execution environment, both to run UNEXT core components and UNEXT application components, defining the abstraction, orchestration, and support for network and compute capabilities across a secure, generalized network-compute continuum.

Extended Compute Services

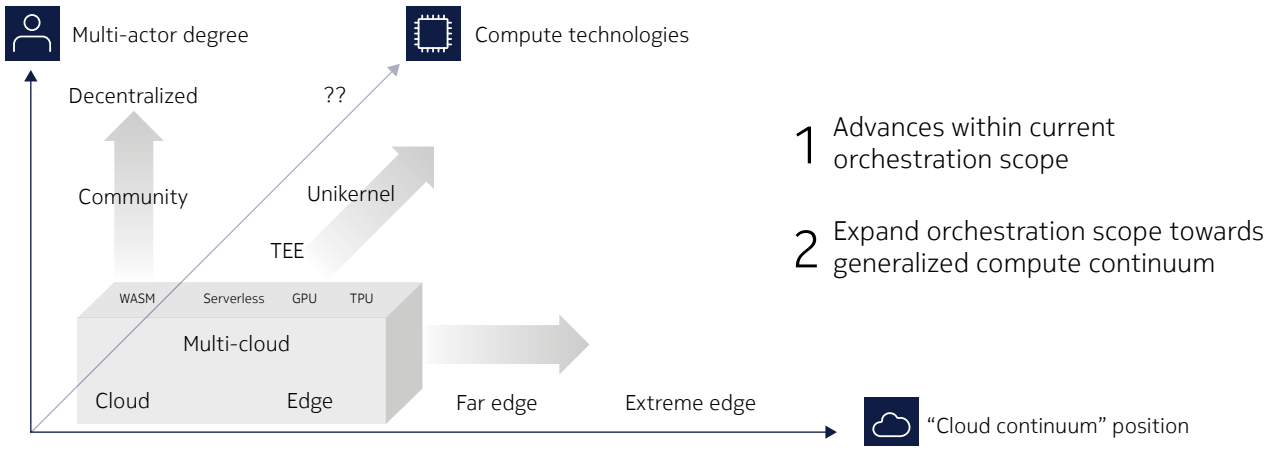
Vision: towards a generalized network and compute continuum

A relatively recent trend in distributed computing has looked at the extension of the realm of cloud orchestration to multiple domains across a "cloud continuum." While this notion has received different and evolving definitions in the literature (Moreschini & al., 2022), proposes the following definition synthesizing prior proposals: "Cloud Continuum is an extension of the traditional Cloud towards multiple entities (e.g., Edge, Fog, IoT) that provide analysis, processing, storage, and data generation capabilities."

As this definition focuses implicitly on cloud (native) technologies and leaves out the network interconnecting the various compute domains, others have proposed to generalize to "network-compute continuum." A network-compute continuum is an abstraction layer for heterogenous, distributed (and possibly decentralized) compute and network resources that forms a uniform execution environment for applications and network functions, so they do not need to care about their kind, physical location or ownership.

In Extended Compute Services, we envision a generalization of the scope of orchestrated distributed compute environments towards a secure, generalized network-compute continuum, which spans the three following dimensions (Figure 2).

Figure 2: Expanding the scope of orchestrated compute environments towards a secure, generalized network-compute continuum.



The first dimension is the position on the cloud continuum, which depicts where service components might be placed in the continuum formed by the central cloud, by the edge (with its various definitions), and beyond, by the end devices themselves, here labeled as "Extreme-Edge." Getting different tiers along such a cloud continuum to jointly contribute to an end-to-end service with specific requirements may require networking them (and the workload they host) with specific characteristics, which is sometimes described as network-cloud. Today's main embodiment of an enabling technology stack for distributed compute platforms is cloud computing, or cloud-native technology, and it has been adapted to fit various tiers (and their intrinsic constraints and advantages) of the cloud continuum. The Telco industry has embraced this technology trend as well, while recognizing it does not come without hurdles in terms of performance (e.g., cloud-native technology limitations for network virtualization) or business (e.g., the competition of hyperscalers entering the telecom market). UNEXT considers a potential future where telcos might expand beyond their traditional role, blending compute and service orchestration in their network service offering.

A second dimension of the generalized network-compute continuum is therefore the multi-actor degree, that is, how many and how dynamically different actors (and more generally administrative domains) are involved in the considered network-compute continuum. In particular, consequences in terms of architectural choices (encompassing centralized, federated or decentralized approaches or a coordinated mix of those) and in terms of multi-objective orchestration (since different domains/owners may have different objectives to be reconciled with service requirements) should be considered. While current continuum discourses usually revolve around a centralized orchestration domain ruled by a main actor (be it the mobile network as an overall orchestration domain of the MNO or the provisioned ensemble of Kubernetes clusters addressable by an application provider through a multi-cloud solution), the ECS vision considers a more decentralized and dynamic continuum where any actor could opportunistically request orchestration above any other (willing) actors' resources ([Bou23]).

And finally, a third dimension is on compute technologies, meaning the mix of different hardware and software compute technology considered in orchestrated compute environments. The generalized network-compute continuum should consider technologies beyond the well-established (albeit evolving) set technologies considered in cloud-native approaches. Their inclusion likely requires adaptations, in particular from a security standpoint, as they may not have been designed with a "shared" context requirement, but also due to the highly heterogeneous nature of compute resources and/or tasks to be executed in such a compute continuum. This represents a paradigm shift from traditional cloud computing, moving away from reliance on an abundance of rather homogeneous resources and corresponding service models towards a model of frugality and resource and service heterogeneity while considering natively "coopetition " among stakeholders of various sizes. At its core, ECS requires new orchestration methods to leverage the opportunistic use of specialized resources. This redesign imposes heterogeneity as the norm, encompassing a variety of resources, networks, and execution platforms, all while embracing multi-tenancy as the standard practice and considering its security implications.

Considering holistically these different dimensions in UNEXT allows to leverage synergies between them. It also opens opportunities to address overarching challenges such as end-to-end sustainability in service realization beyond mere energy-aware orchestration by reducing over-provisioning of resources and improving opportunistic reuse of already deployed resources.

Fundamentals of a Network-Compute Continuum

Papers such as [Kok23] proposed an encompassing view on computing continuum resources and orchestration, highlighting that there are different realization paths to such a continuum (depending on chosen scope, technical assumptions, design choices, etc.). For example, different approaches in orchestration topology can be taken, from logically centralized (e.g., a single network operator-centric proposal) to a fully decentralized/peer-to-peer. But, regardless of such design choices, all approaches require the same (logical) "continuum functions" for discovery, monitoring, federation, security and orchestration. These continuum functions rely on the interworking of multiple interconnected (network-)compute islands (meant here as an administrative domain regrouping compute and network resources), each including the island functions responsible for resource management and orchestration within this domain.

We use the term “island” to avoid the confusion brought by the word “domain” in the context of “network domains,” which may intertwine with the network location of compute resource collections. As depicted in Figure 3, “islands” are likely to belong to different stakeholders of the continuum and can be embodied by collections such as today’s clouds (central or edge), singletons (e.g., single UE devices exposing compute resources to the continuum), or even network domains offering in-network computing capabilities.
The very notion of continuum implies that islands are interconnected; when the networks supporting this interconnection offer solely connectivity services, they may need to be steered for continuum orchestration purposes, typically through mechanisms such as the interfaces/APIs they expose.

Figure 3: Examples of network-compute islands and interconnecting networks.

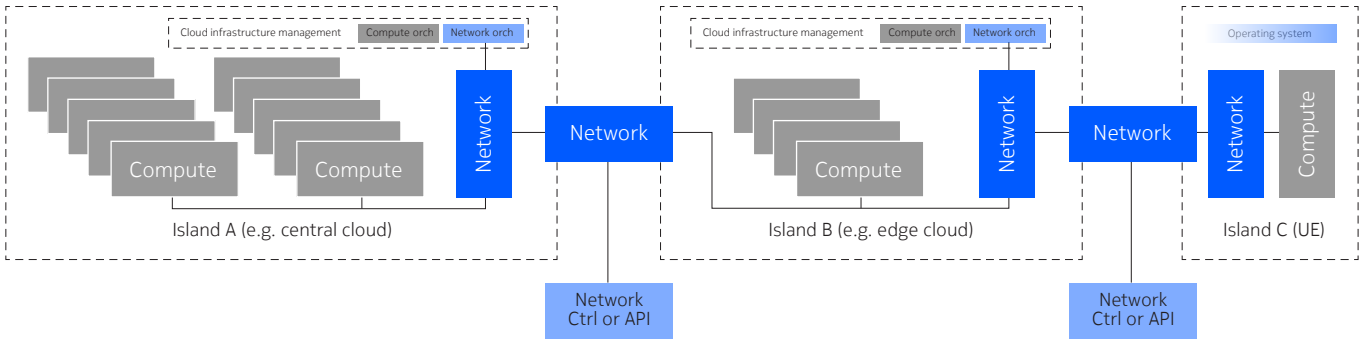


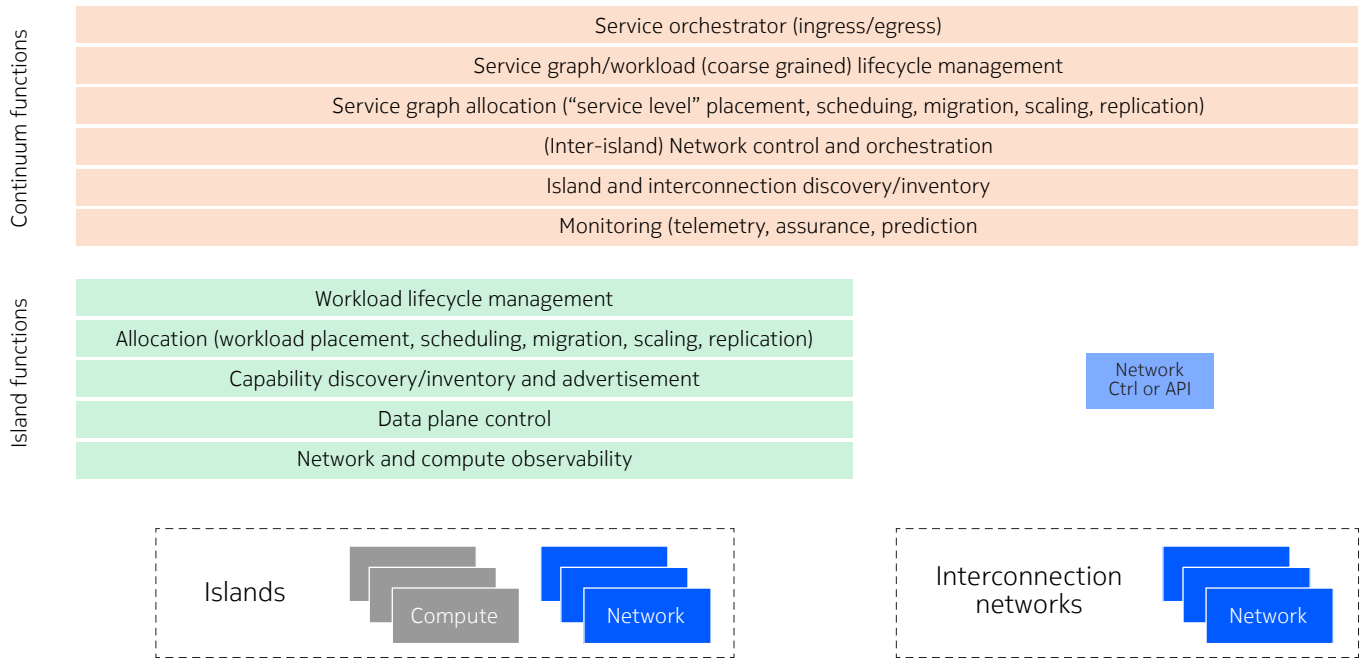
Figure 4 presents the abstract system view of a network-compute continuum functional decomposition.

Islands gather compute and network resources, which are exposed and managed through a number of island functions, including workload lifecycle management and allocation over available compute resources, in-island data plane control, and compute and network observability functions. They also provide advertisements of their capabilities to the broader continuum.

The networks supporting the interconnection of the different compute islands need to be discovered and steered (through network control mechanisms or APIs) to meet the expected connectivity and performance requirements of services spanning several islands.

While island functions may look mostly similar to current cloud-based orchestration fundamentals, continuum functions are more challenging, as they need to provide the abstraction and reconciliation mechanisms across the three dimensions mentioned in Section 2.1. Continuum functions encompass a range of service orchestration functions, such as ingress (from the user) and egress (towards islands and interconnection networks), management of the service graph lifecycle, decomposition and allocation of the service graph to islands, control and orchestration of interconnection networks, discovery of islands and interconnection networks, their respective capabilities and monitoring.

Figure 4: Abstract system view of a network-compute continuum.



Examples and use cases

While the full realization of the ECS vision might seem distant, the need for some form of network-compute continuum orchestration exists in today's use cases. Some examples of these are provided here, illustrating existing or future use cases to which the developed technologies could be applied.

- **Extreme edge computing and orchestration** is already in use in IoT scenarios, where connected UEs can be instructed to run arbitrary compute workloads (e.g., containerized [BAL]) to fulfill a given task, in particular when their connection to the broader network is unavailable. In parallel, **edge computing**, including in telco environments coupled with fine-grained network control (mobile or **multi-access edge computing** [MEC]), is actively pursued as an avenue for services with stringent latency requirements that cannot be fulfilled by relying on larger, more distant clouds. **Mobile offloading** has gained attention lately as a major use case for future 6G networks, supporting the migration of functionality executed in the client device to be offloaded on-demand and executed on a suitable compute environment in the continuum, likely at the edge to support low-latency scenarios. Arguably, those approaches corresponding to the first axis of Figure 2 ("position on the cloud continuum") are proposing actionable solutions in select contexts based on cloud-native technology but are rarely considering the whole spectrum of the cloud continuum.
- **Multi-cloud**, the idea of relying on a mix of cloud computing services from different providers, has become an increasingly important topic for any solution relying on cloud computing. Distributing cloud resources, applications and workloads across public, private or hybrid clouds allows to limit vendor lock-in, optimize performance (through selection of the most suitable cloud offer for given requirements), enhance reliability and optimize costs. Multi-cloud approaches typically illustrate the middle of the second axis of Figure 2 ("multi-actor degree") but lack the dynamicity of a context where a large number of actors could provide some compute and/or be discovered and added opportunistically to a mix of usable environments.
- **AI/ML**, with its heavy compute and data requirements, has become a major use case (and growth relay) for cloud computing, while illustrating the need for supporting virtualization and isolation over heterogeneous compute technologies, as illustrated by the third axis ("Compute technologies") of Figure 2. The extension of its applicability to more specialized, distributed use cases with their own hardware constraints ("edge AI") further emphasizes the need for supporting portability and adaptability of ML pipelines onto various hardware architectures. While GPUs and ML-specialized architecture such as TPUs have already been incorporated into cloud environments, other architectures will eventually need to be added, such as FPGAs ([Inc22]) or even quantum computers ([Sye19]). Beyond the specifics of ML workloads and pipelines, this can be generalized to adding support for sharing of many different software and hardware technologies.

As can be seen, the technological solutions addressing all or parts of the above use cases still require significant evolutions to apply in future use cases that would mix them in the context of a generalized network-compute continuum.

ECS approach and directions

Challenges and requirements

While a number of (typically cloud) technologies are readily available, building any well-scoped network-compute continuum is a technical challenge in itself. The UNEXT vision based on decentralized, multi-stakeholder environments brings further challenges:

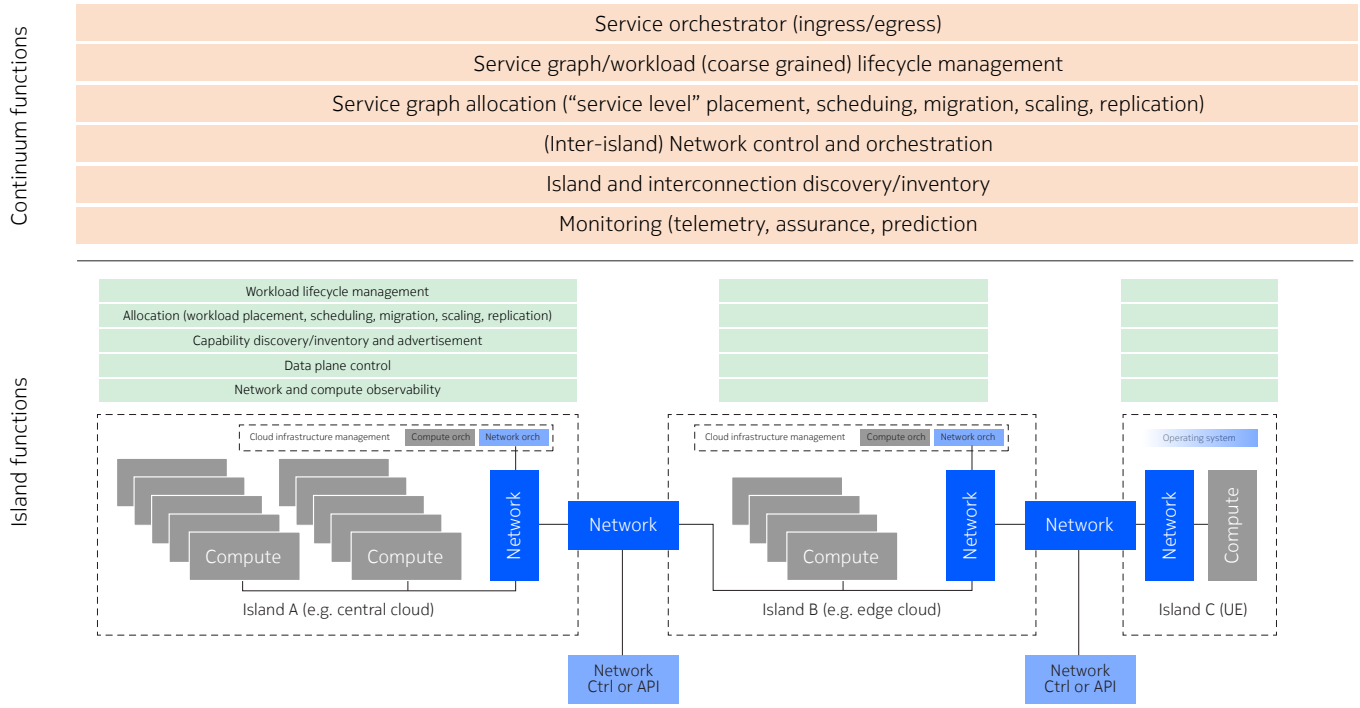
- Just like the physical world, the digital world should be eminently multi-stakeholder. This is reflected in UNEXT being defined as a collection of decentralized environments called "UNEXT domains" [Atk24], i.e., domains belonging to various parties interacting on various reciprocal trust assumptions. As a result, it is likely that the UNEXT network-compute continuum will be composed of a collection of "potentially interconnected" network and compute islands belonging to different actors, each in charge of their own resource orchestration and contributing to the overall service orchestration continuum functions.
- A significant part of a continuum realization is about information exchanges; in UNEXT, this is supported by the knowledge continuum realized by Knowledge and Data Services (KDS) [Con24]. The knowledge continuum supports ECS by supporting the distribution of island (capability) information and island telemetry across decentralized environments, as well as by supporting knowledge transformation and machine learning processes for compute orchestration functions and agents as needed.
- Such a complex, decentralized system requires a level of automation that confines to autonomy, allowing it to run with minimal to no human intervention.
- To support legacy system integration and provide a migration path, UNEXT should support backward compatibility with existing compute orchestration solutions. As a result, Extended Compute Services shall be designed to interface and interact, but go beyond, legacy cloud-native technologies and deployment patterns, supporting heterogeneous island and resource capability inclusion in the continuum.
- UNEXT can be used to orchestrate and deliver indifferently network services and E2E applications, i.e., the UNEXT network-compute continuum should consider equally the complete cloud continuum spectrum, including extreme edge (departing from cloud-centric "server-side" orchestration).

As can be inferred, realization and adoption of a generalized network-compute with the end goal of tapping (securely) into any network and compute, from anybody under any form is an open challenge, the breadth of which will likely require cooperation and standards (if only de facto) across the industry.

The UNEXT network-compute continuum

Figure 5 illustrates how the abstract system view depicted above can be applied in the context of various islands of different scope and capabilities, showing how compute and network resource orchestration can be realized leveraging local cloud infrastructure management functions (e.g., an hypervisor or a container orchestration platform along with a software-defined in-DC network) for cloud-like islands and through the UE's operating system for extreme edge devices. It is also worth noting that the actual island functions realization could be hosted within the island or deported (e.g., for energy or availability reasons).

Figure 5: System view of a network-compute continuum with island deployment example.

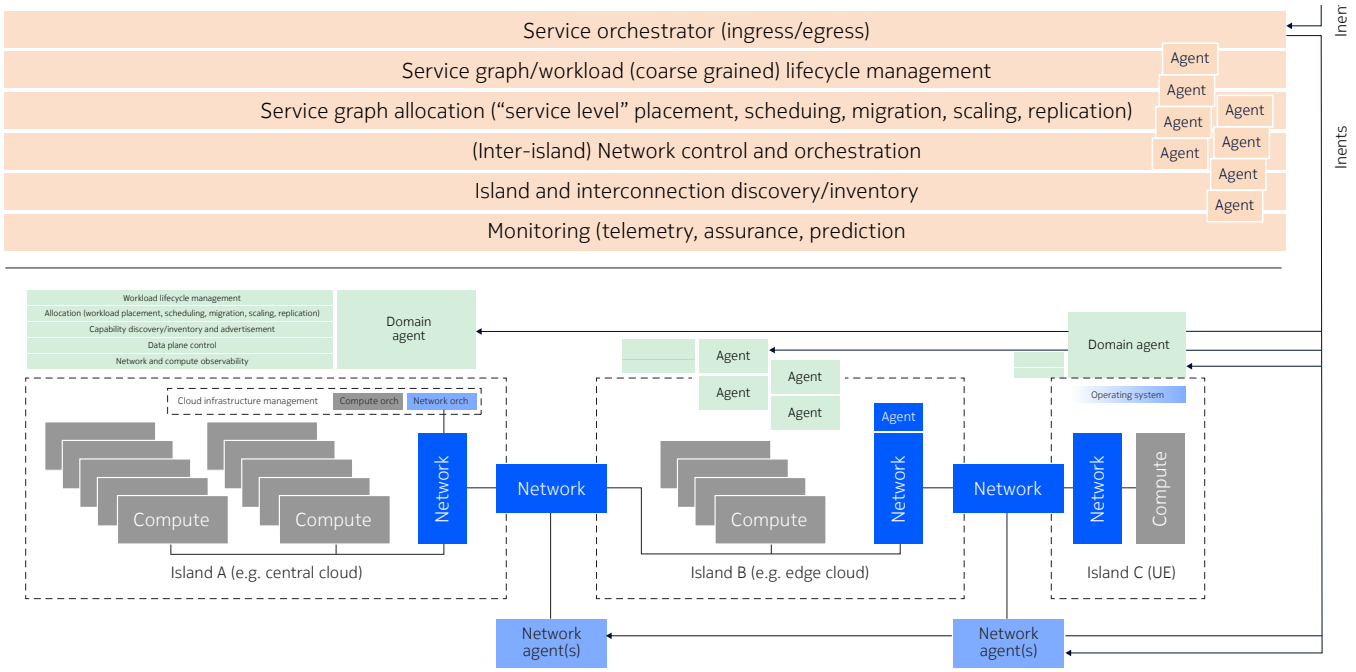


Realizing the **UNEXT network-compute continuum** entails specializing in an abstract system view with UNEXT working assumptions and addressing the challenges and requirements listed in Section 4.1. In particular:

- UNEXT is realized by a collection of autonomous agents exhibiting a set of common characteristics and behaviors. ECS should provide the compute execution and orchestration mechanisms to deploy those agents, and most of those compute execution and orchestration mechanisms should themselves be realized as UNEXT agents.
- Both to accommodate the inherent complexity and to limit visibility onto other stakeholders agents and their capabilities, management and control interactions between those agents are intent-based. However, resources and functional operations (the "services" provided by each agent) can be accessed over a multitude of interfaces, not exclusively intent-based.

Figure 6 below illustrates the implications of those UNEXT working assumptions by presenting the ECS network-compute continuum system view. Note that the actual agent decomposition is an open research question. While it is foreseen, for backward compatibility with existing execution environments, that a single "domain agent" may front compute islands (illustrated by island A in the figure), one of ECS's goals is to explore the agent decomposition of both continuum and island functions. Island function decomposition in agents is illustrated by island B in the figure, with traditional cloud infrastructure management functions replaced by autonomous agent(s) realizing parts or all of the island functions, or even more speculatively, the very network and compute resources being fronted by autonomous agents. Regardless of their functional scope, each of these "ECS agents" will need to be hosted on a compute resource of the continuum.

Figure 6: UNEXT network-compute continuum system view



Research Directions

From the system view described above, we identify the following key concepts and associated research directions.

While it is expected that the UNEXT network-compute continuum would be composed of islands offering heterogeneous capabilities (i.e., involved compute and network domains will have their specific functional, performance or security characteristics), supporting differentiating capabilities in the continuum abstraction layer and function set is critical.

The purpose of the ECS abstraction layer in UNEXT is to be able to eventually reconcile ingress intents with the characteristics of the relevant network-compute island(s) to drive their selection for fulfillment. Those characteristics can be vastly heterogeneous, both due to **different resources and island capabilities**. On the one hand, resources here should be understood as the elements to be orchestrated offered by each island and can be physical/hardware- or logical/software-based. On the other hand, island capabilities could include specifics in the way island orchestration or management is realized and, in some cases, may be guided.

Capability exposure in the continuum is expected to be realized through one or several agents within an island and may vary over time as new capabilities are added or installed and therefore input into or discovered by the agent(s).

A fundamental dimension of a network-compute continuum is the abstraction and supporting mechanisms for **joint network and compute monitoring, management and orchestration**, covering both networks within islands and networks enforcing island interconnection, for instance to support latency-sensitive data paths across (parts of) the continuum. A number of fora have started investigating such "Integrated Network and Compute" (INC) mechanisms under various names (e.g., IETF with Computing Aware Traffic Steering [Li24], 3GPP with Edge Application Server selection in Release 19, or O-RAN's Communication and Computing Integration Networks study item). However, the requirement to extend the network-compute continuum toward the extreme edge is a fundamental difference with respect to the state of the art in most of these fora.

While island and continuum functions are expected to be realized through agents, the number, granularity and resulting network/graph topology [Kok23] of those agents remains an open question. The continuum will likely be realized through islands with **heterogeneous agent decomposition**: islands with a single fronting agent for legacy cloud tiers; in the future, domains decomposing traditional IaaS/CaaS functions into multiple interacting agents; even more speculative situations where the very notion of island (as a "locally centralized collection of resources") may disappear.

As mentioned previously, agent interactions in UNEXT are expected to follow the intent paradigm, which has the advantage of hiding/not requiring the intent handler realization details from the intent owner. ECS network-compute continuum agents are therefore expected to comply with this design choice, although the type of intent they will process will adhere to a given formalism and be already derived from a user or business intent. This is particularly true for the (logical) continuum orchestration function in Figure 6, which should enforce **intent-driven placement** of parts of the service graph across the continuum based on expected end-to-end service characteristics. In some cases, a received intent may include explicit location or target characteristics informing the deployment or execution of a workload ("onboard this end device in the service chain"); in other cases, the target network-compute island for a given service component may have to be derived from "implicit" requirements (e.g., requesting execution on resources trusted by or belonging to a given actor, or requesting an SLA that implies the selection of certain execution domains and specific networking configuration).

Furthermore, UNEXT applications, meant to be deployed in the network-compute continuum, will require novel design patterns and service models ([Kok23] [Alo23]) to enable such "continuum-nativeness." The (continuum) service model used to describe and orchestrate UNEXT applications should enable adaptation mechanisms allowing to choose between alternate E2E application realizations, from selecting alternate application component realizations matching different island capabilities to the more speculative dynamic adaptation across workload nature to fit selected island resource capabilities.

In the advent of a more decentralized continuum orchestration and associated agent decomposition, **intent-driven resource orchestration** leaves room for local decision-making on workload redirection, generalizing the concept of "(mobile) **offloading**" to the transfer of workload execution between any two islands—or even to "self-offloading" in case the decision is made by the service component itself (encapsulated into an agent). A key challenge of any agent decomposition, but which worsens as its granularity increases, is in the stability of the overall autonomous agent-based orchestration, e.g., in terms of placement and scheduling.

To support both new requirements in today's execution environments and take advantage of tomorrow's compute continuum, advances in orchestration processes themselves are needed. One field actively investigated is the application of **AI/ML in orchestration** ([Bou24]), for auto-scaling, resource optimization, scheduling, predictions (e.g., volatile resource availability prediction in extreme edge environments) and accommodating orchestration objectives such as latency or sustainability.

Conclusions

The UNEXT vision of extending the current realm of networking toward the autonomous orchestration and delivery of services over the resources of numerous stakeholders calls for revisiting the way network and compute resources are managed and orchestrated. Extended Compute Services will support decentralized, agent-based realization and orchestration of the continuum formed by an open collection of network and compute islands. These islands will be contributed by a wealth of actors, support heterogeneous and evolving technologies, and be positioned at a variety of positions in the network, offering differentiating properties. This will enable a novel class of "continuum-native" applications and services that autonomously position on and adapt to the best-suiting available environments. Along with the other technical pillars of UNEXT, ECS will enable the transformative potential of a unified networking experience.

References

[Alo23] Alonso, J., Orue-Echevarria, L., Casola, V. et al. Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review. J Cloud Comp 12, 6 (2023). https://doi.org/10.1186/s13677-022-00367-6

[Atk24] G. Atkinson et al., "Operating in Decentralized Environments for a Unified Networking Experience," Nokia Whitepaper, 2024; online resource

[BAL] Balena.io, https://www.balena.io/

[Bou23] M. Boussard, P. Peloso, V. Verdot, R. Douville and N. L. Sauze, "Distributed Personal OS Environments – exploring Cooperative Fog Computing," 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), Istanbul, Turkiye, 2023, pp. 1-4, doi: 10.1109/SmartNets58706.2023.10216015.

[Bou24] A. Bouroudi, A. O.-A. (2024). A Novel DRL Framework for Cross-Domain Network Scaling in 6G Networks. IEEE HPSR.

[Con24] A. Conte et al., "Knowledge & Data Services for a Unified Networking Experience," Nokia Whitepaper, 2024; online resource

[Inc22] S. Ince, D. E. (2022). Token-based authentication and access delegation for HW-accelerated telco cloud solution. International Conference on Cloud Networking (CloudNet), (pp. 109-117). Paris, France. doi:10.1109/CloudNet55617.2022.9978865

[Kok23] Kokkonen, H., & al., e. (2023). Autonomy and Intelligence in the Computing Continuum: Challenges, Enablers, and Future Directions for Orchestration. Retrieved from Arxiv: https://arxiv.org/abs/2205.01423

[Li24] Cheng Li, Zongpeng Du , Mohamed Boucadair , Luis M. Contreras , John Drake. IETF draft-ietf-cats-framework-01, "A Framework for Computing-Aware Traffic Steering (CATS)," IETF Draft, 2024; https://datatracker.ietf.org/doc/draft-ietf-cats-framework/

[MEC] ETSI Multi-access edge computing, https://www.etsi.org/technologies/multi-access-edge-computing

[Sef23] A. Sefidcon, C. Vulkán, M. Gruber, "UNEXT: A unified networking experience," Nokia Whitepaper, 2023; online resource

[Sye19] Nawaz, Syed Junaid, et al., "Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future." IEEE Access 7 (2019): 46317-46350.

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.

With truly open architectures that seamlessly integrate into any ecosystem, our high-performance networks create new opportunities for monetization and scale. 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 OYJ
Karakaari 7
02610 Espoo
Finland
Tel. +358 (0) 10 44 88 000

Document code: CID214403 (December)