Prospects on the Adoption of a Microservice-Based Architecture in 5G Systems and Beyond

Sebastian Robitzsch[a], Marco Centenaro[b], Nicola di Pietro[b], Luis Cordeiro[c], André S. Gomes[c], Peter Sanders[d], Arif Ishaq[b]

[a]*InterDigital Europe Ltd, London, United Kingdom*
[b]*Athonet, a Hewlett Packard Enterprise Acquisition, Bolzano Vicentino, Italy*
[c]*OneSource Consultoria Informatica Lda., Coimbra, Portugal*
[d]*Everbridge, Deventer, Netherlands*

Abstract

The increasing *softwarisation* of mobile core network functions is fostering the evolution of the mobile network architecture itself, which in its fifth generation (5G) has moved towards a service provider/consumer framework and service-based interfaces. Moreover, the 5G architecture is suitable for the exploitation of the mobile technology for dedicated, non-public uses as an alternative to nation-wide deployments. The *5G core networks* are a crucial part of this architectural paradigm shift, which aims at closing the gap between the telecommunications domain and the information technology world at large. The objective of this article is to discuss the adoption of software design concepts like microservices and cloud-nativeness in the context of mobile networks. Specifically, we will i) advocate the need for a non-trivial adaptation of the 5G core network and a redesign of its functions into a *micro*service-based architecture, ii) identify an approach to achieve this objective and put it into practice by decomposing three exemplary network functions, both theoretically and practically, in microservices in charge of distinct responsibilities, and iii) propose ways forward towards the adoption and further extension of these concepts in beyond-5G mobile systems.

Keywords: 5G, core network, microservice, cloud-native, service-based architecture, standards

1. Introduction

During the last years, the paradigm of *microservices* has gained momentum in various information technology fields, embracing a multitude of business cases and targeting plenty of heterogeneous application scenarios. The concept of microservices yields from the observation that end-to-end digital business services and the underlying computerised functionalities are becoming more and more complex to develop, deploy, interconnect, manage, heal, and update [1]. It relies on identifying independent *responsibilities* within the main service, removing unnecessary dependencies, and isolating them into modular, self-standing logical and operational blocks that relate with each other in a Service-Based Architecture (SBA) via dedicated interfaces and through an event distribution bus. Consequently, microservices conceived in such a manner can be developed, run, and orchestrated independently, because each of them is a self-contained coherent entity.

Recently, thanks to the advent of an SBA also for the Fifth Generation (5G) of mobile networks, the same approaches have become of interest for mobile telecommunications since Third Generation Partnership Project (3GPP)'s Release 15. When transitioning from generic distributed systems to mobile network systems, we have to account for Network Functions (NFs) rather than generic business functions. As part of a critical infrastructure, 5G NFs have to satisfy stringent requirements in terms of latency and dependability. In the past, this was the motivation of having dedicated hardware implementing all network segments and functionalities. However, since the fourth generation of mobile networks, standardisation initiatives like, e.g., the industry specification group for Network Function Virtualisation (NFV) at the European Telecommunications Standards Institute (ETSI), have raised based on the idea to replace physical NFs with Virtual NFs (VNFs). The trend was completed with 5G networks [2], which feature 5G Core Network (5GC) control-plane NFs that interact among each other within an SBA, with the Network Repository Function (NRF) managing NF service registration and discovery. However, despite the adoption of an SBA, the 5GC standardised by 3GPP [2] as components and interfaces is still not sufficient for a straightforward and full-fledged *micro*service-based implementation. More specifically, it is not possible to program the existing 5GC NFs as collections of independent microservices by simply mapping each of the distinct standardised services offered by a NF to a single corresponding microservice.

In this article, we will first discuss in Section 2 the reasons why this cannot be done after setting the necessary foundations on 5G and microservices. Then, we will propose decomposition design patterns in Section 3 of a few functionally heterogeneous 5GC NFs, chosen for their specific diverse roles within the 5GC or importance as enablers of advanced 5G features. For such analysis, we have decided to remain compatible with the architectural specifications of the 5GC [2] at an *inter-NF* level, while we are going several steps beyond the state of the art in the *internal* design of the NFs, fostering the adoption of microservices in their architecture. NFs programmed according to the proposed design patterns can be inserted in the standard SBA of the 5GC without impacting the already defined NFs services and their Service-Based Interface (SBI). Hence, our approach is especially suitable for a step-by-step transition from a more classical to a fully microservice-based 5GC design. Our proposals are not only derived via a theoretical analysis, but are corroborated by the concrete implementation in separate containers of the microservices into which the considered NFs are decomposed. In Section 4 the design patterns are applied in practice to three selected NFs, demonstration the viability of the proposed guidelines. Finally, we will move further and provide a vision on how to embed the outcomes of this study in a beyond-5G system architecture, presented in Section 5.

2. Foundations

This section provides the foundations for the work presented in this paper covering the 5G system architecture, the concepts of microservices and the state of the art published on the topic around 5GC NF decomposition efforts.

2.1. Microservices and Cloud-Native Functions

Often, the classic approach for the design of digital services (usually referred to as *monolithic*) does not fully meet the requirements of the most recent use cases in terms of flexibility, adaptability, continuous development and deployment, scalability, or resource management of the implemented applications. To overcome this, the *microservice* approach is based on identifying independent functionalities (and the corresponding data modules) within the main service, removing unnecessary dependencies, and isolating them into modular, self-standing logical and operational blocks that relate with each other in an SBA via dedicated interfaces and through an event distribution bus. Microservices conceived in such a manner can be developed, run, and

orchestrated independently, because each of them is a self-contained coherent entity. Each microservice can be programmed in different languages, and the computer-scientific development and maintenance of each of them can be adapted to evolving needs without having to reshape the whole service architecture or without impacting how other microservices operate. As opposed to the monolithic approach, the composition of individual microservices into a collection forming the digital service, results in a flexible and scalable solution thanks to its technology-agnostic modularity enabled by a well-defined SBA and its corresponding standardised interfaces. A straightforward consequence of such approach is the natural suitability of microservices for cloud environments. Furthermore, an architecture made of microservices facilitates a software development process based on continuous delivery, enabling the implementation of small changes of the application via rebuilding and redeploying a single or few microservices designed around well-defined *responsibilities*. As a matter of fact, in the following we will embrace the *responsibility-driven* design for microservices [3]. Further, it adheres to principles like, e.g., fine-grained interfaces allowing independent deployment of services, business-driven development, and the DevOps approach [4].

Thanks to their features, microservices play very naturally the role of components of cloud-native (network) functions. Cloud-native refers to a function or an application that is specifically conceived for running in the cloud, taking advantage of the cloud's capabilities to automate infrastructural changes and software management tasks. Cloud-native applications are suitable for automated orchestration and externalised monitoring, and profitably rely on the cloud's built-in resilience, scaling, and self-healing, ensuring forward-compatibility with serverless approaches [5].

The current virtualisation technologies to realise cloud-native applications are containers (mainly Docker) which exemplify the cloud-native approach [1]. Containers package code, libraries, dependencies, and run-time into a single binary image, so that they can be moved easily and can run in any environment. Instances of containers are executed by a common operating system and they fit well in the microservice framework, since each container occupies a well-defined slice of the hosting infrastructure and is isolated from the other containers.

2.2. SBA and Microservices in 5G Networks

Starting with 3GPP's Release 15 [2], a paradigm shift in the system architecture was introduced on how control-plane NFs communicate among each

other. In pre-release 15 systems, all NF instances had a strict one-to-one relationship among each other and used application layer protocols such as Diameter. With the advances of cloud solutions that can scale on demand, Release 15 adopted an SBA for the 5GC (see Figure 1) yielding:

- The decomposition of the mobile network's core functionality into smaller independent NFs.
- The introduction of the concept of consumer (endpoint/clients) and producers (service endpoint/servers) without strict requirements on which consumer is allowed to communicate with which producer.
- The introduction of SBIs for the majority of 5GC NFs moving to Hypertext Transfer Protocol (HTTP) Version 2 (HTTP/2) as the application layer protocol and JavaScript Object Notation (JSON)-encoded payload.

On top of that, in Release 16 an optional network entity called Service Communication Proxy (SCP) was added to take over the responsibility of proxying traffic between a consumer and producer instances [2, Section 4.2]. The deployment of an SCP is optional however and the SCP does not expose a 5GC service itself through a callable Application Programming Interface (API). Instead, it is used for "indirect communication between NFs and NF services" [2, §4.2] and is an addressable endpoint via an IP address or Fully Qualified Domain Name (FQDN). If no SCP is deployed, consumers and producers communicate with each other without an SCP, and Release 16 refers to that as a "direct communication".

The 5GC's SBA and the decomposition of NFs into microservices are envisioned to play a key role to enable an efficient coexistence of traditional Public Land Mobile Networks (PLMNs) and the newly introduced Non-Public Networks (NPNs) [2, Section 5.30] in a truly dynamic and automated manner over the same infrastructure. This holds true *a fortiori* whenever some virtual or physical resources are deployed at the edge of the network or when they are shared, for example, in scenarios where a PLMN operator provides localised and private services to a customer (e.g., by means of a dedicated network slice) while running its usual operations in the same area. To scale with the demand of often time-limited NPNs deployments, the ability of cloud-native implementations of 3GPP's SBA enables a "breathable" network. Furthermore, it should be noted that some prominent NPN use cases

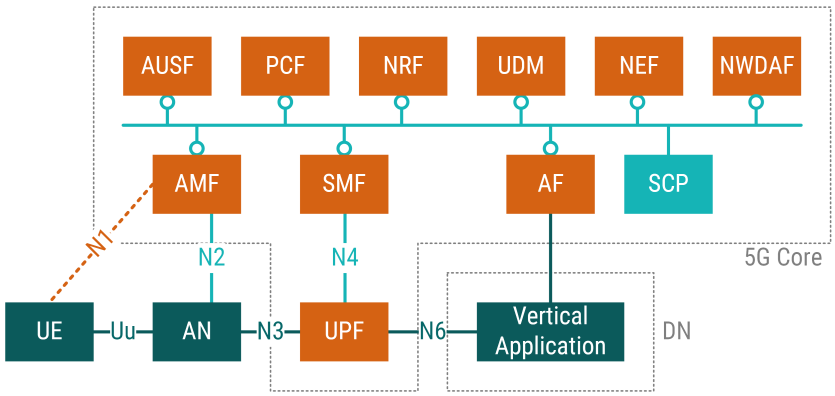

Figure 1: 5G System Architecture and Network Functions as in [2].

require to minimise inter-network operation and external points of contact combined with a rather fine-tuned list of network and radio features to guarantee the Key Performance Indicators (KPIs) promised to the vertical. More details on the requirements and capabilities with 5G Standalones (SAs) deployments in relation to NPNs can be found in the whitepaper of the 5G Public Private Partnership (5G-PPP)[6].

However, there is also a counter argument for mobile networks with very high security requirements or NPNs deployed in isolation from other systems over dedicated infrastructure: those dedicated to Public Protection and Disaster Relief (PPDR) or serving remote locations (e.g., mid-sea oil stations, underground mines, even airplanes), inter-working and dynamic orchestration might not be as important as the apparatus' resilience and security, thus monolithic architectural solutions may still be a viable approach. Nonetheless, microservices remain beneficial in this case for technological forward-compatibility (see Section 3.2 for more details).

2.3. Previous Work on 5G Core Network Network Function Designs and Ways Forward

Some proposals to incorporate the concept of microservices in the design of 5GC NFs have recently appeared in the literature, but the topic has not been extensively studied yet. For instance, [7] analyses the evolution of the core network along successive generations of mobile network technologies. The authors highlight the features of the 5G SBA and mention as an open and not yet investigated challenge the application to 5G systems of microservices as enablers of optimised non-redundant and flexible NFV ecosystems.

They cite the approach of [8] as a promising framework that, though, still needs to be applied to the 5GC. Further, [9] broaches microservices as central elements of beyond-5G mobile network systems. More specifically on 5GC NFs, [10] reports on the cloud-native modular design and the implementation of the functional procedures of an Access and Mobility Management Function (AMF) conceived for microservice-based architectures within the OpenAir Interface (OAI) project. The authors give a description of the different data and functional modules and the architectural implementation layers that constitute their AMF. However, we observe that the focus of [10] is more on the cloud-nativeness of such an approach and the compatibility with cloud environments, rather than an actual microservice-based design of the AMF itself. Some vendors have proposed cloud-native 5GC designs that claim to embrace microservice principles [11], however the lack of a shared approach to NF decomposition weakens the effectiveness of the design. Finally, related work exists on how to provision, manage, and automatically orchestrate microservice-based network function virtualisation service platforms and VNFs in 5G [12, 13], but not with a focus on the 5GC.

In the following, the main principles are analysed that underlie an efficient NF decomposition into microservices, aligned with 3GPP. Based on this, Section 4 goes beyond the state of the art by proposing a novel design for a microservice-based implementation of three exemplary 5GC NFs.

3. Design Patterns for NF Decomposition

This section argues on the degree of 5GC decomposition that is reasonable to achieve in the previously identified deployment scenarios, providing advantages and limitations of the proposed design guidelines. Two approaches have been identified on how to make NFs interact via SBIs:

- Based on interfacing the entire, monolithic NF.
- Based on identifying its sub-functions providing NF services to be interfaced.

The former approach is quite conservative but may be safer/pragmatic while transitioning from monolithic NFs towards a microservice architecture. The advantage of this approach is that the vendor does not need to decompose the monolithic NF into NF services, rather designing a unique SBI to interface different monolithic NFs. The disadvantage is that cloud-native procedures

to manage virtualised NF instances have limited impact, as the NF cannot be scaled based on demand. This is where the described foundations of realising an NF as microservices is of paramount importance (see Section 2.1).

On the other hand, the latter approach is more aligned with the spirit of the 3GPP specifications provide both i) the functional description of each NF [2, §6.2] and ii) the services provided by each NF [2, §7.2]. As a matter of fact, in principle, each of the NF services offered by a NF shall be self-contained, reusable and use management schemes independently of other NF services offered by the same NF (e.g. for scaling, healing). This allows for agile dynamic scaling (horizontal/vertical), independent life-cycle management, and data isolation. This follows the principles of an Service-Based Architecture, as described in Section 2.2.

3.1. General Criteria

Off-the-shelf methodologies like, e.g., the 12 factor app [14], exist on how to convert a monolithic software into a set of microservices. In the specific case of a 5GC, the exercise is how to decompose a monolithic NF into a set of sub-functions, each one implemented as a microservice, that form the NF as a whole. This is up to each vendor to decide. 3GPP itself defines the functionalities of each NF in [2, Section 6.2]. This helps making such an exercise easier especially for control-plane NFs since the services each of them provides are specified in [2, Section 7.2]. In other words, the 3GPP specifications provide initial (but not exhaustive) guidelines for a logical decomposition in microservices of service producers (as known as the server behaviour of a NF).

On the other hand, a service consumer (the client behaviour of a NF) does not offer any services and the criteria for service producers cannot be directly applied to them. Nevertheless, the decomposition of service producers and consumers can be done as per the criteria below, which have been recently discussed also within the Next Generation Mobile Networks (NGMN) Alliance's Operating Disaggregated Networks (ODIN) project [15]:

- *Bottleneck Mitigation and Parallel Execution* – An NF's functionality that poses a bottleneck in terms of, e.g., performance within the same NF or direct interactions with other 5GC NFs, may indicate that an ad-hoc microservice should be created for that. The intention would be to allow the utilisation of more compute capabilities for this microservice to mitigate the bottleneck.

- *Resilience* – Key functionalities of an NF with high-availability requirements shall be isolated in dedicated microservices, so to increase resilience against failures and increase the dependability of the NF.

- *State Dependency* – Whether an application is stateful or stateless depends on how long the state of interaction with it is being recorded and how that information needs to be stored. In this context, there must be a criterion that depends on the data that a NF service needs to read, upload, retrieve or pre-process for running.

It should be noted though that decomposing a monolithic NF in microservices may increase the risk of security breaches due to, e.g., unauthorised access. Thus, it is of paramount importance to adhere to state-of-the-art, recognised security technical implementation guidelines while developing the various microservices, so to assure, e.g., built-in authorisation and authentication by means of JSON Web Tokens (JWTs). Moreover, the adoption of built-in security to the automated procedures for development, testing and release of software, i.e. DevSecOps (see Section 2.1), even allows to bring any security-related assessment of software written, e.g. vulnerable libraries or dependencies, into the automated CI/CD process to develop and deploy 5G systems.

3.2. Advantages and Limitations of a Microservice-Based 5G Core Network

Adopting the proposed criteria for NF decomposition is likely to bring both business and functional benefits to vendors and operators:

1. *Product Flexibility* – A fine-grained modular software architecture allows a vendor to more easily and flexibly customise its solution according to the operator's or vertical customer's requirements, addressing the heterogeneity of needs that differentiate public nation-wide network operators and private network owners.

2. *Forward Compatibility* – Mobile network standards are in continuous evolution. 5G has not been fully deployed yet, but the community has already started investigating the Sixth Generation (6G) [9]. A microservice-based design and development of NFs allows for a more straightforward and natural upgrade of a NF's services and functionalities.

3. *Data Segmentation* – Having separate database implementations, tailored to each microservice's needs and without redundant information, makes user and network data better isolated, thus more easily manageable. This holds even more in distributed deployments, e.g., NFs over remotely located sites or over different network slices.

Nonetheless, the proposed approach may also feature some downsides, which need to be evaluated depending on the application scenario, or at least one needs to account for some trade-offs regarding the following subjects. Considering that the plain, standard 5GC SBA is already microservice-compatible if one develops each entire NF as a single microservice, there could be concerns regarding:

1. *Development Effort* – Further decomposition may introduce too much complexity within functions, e.g., too many internal interfaces to develop or excessively complex event distribution systems within the function, especially at the beginning of the decomposition process.

2. *Security* – 5G comes with more stringent security requirements than the previous generations, and many high-security use cases are envisioned especially for NPN deployments. While NF decomposition helps, e.g., by improving the database security thanks to segmentation and isolation, on the other hand the exposed attack surface as well as the number of potential vulnerabilities increase.

It is worth remarking that the DevOps practices, previously mentioned in Sec. 2.1, can be instrumental in managing the proposed approach for a microservice-based 5GC thanks to the agility and automation they bring in all parts of software development and operations. In particular, with Dev*Sec*Ops software delivery speed and quality is augmented by security thanks to embedded controls and autonomatically generated security compliance artifacts.

4. From Theory to Practice

The goal of this paper is to discuss the decomposability of 5GC NFs into microservices, consistently with the definition of the 5G SBA and to provide design patterns on how to achieve that. Section 3 provided the theory behind this objective. However, it can be clearly asserted that a decomposition

cannot be automatically obtained by creating one microservice for each of the services defined by a 5GC NF. In fact, the specific services produced and consumed by the 5GC NFs as defined by 3GPP [2, 16] are not fully compatible with the definition of microservices that was provided in Section 2, contrarily to what has been believed in consequence of a too high level of analysis, for instance in [17].

More precisely, at the moment of carrying out the actual development of 5GC NFs, it became apparent that the responsibilities [3] assigned to 3GPP NF services are not always independent of one another, and certain responsibilities are duplicated, making it difficult to allocate their implementation to independent developer teams. Hence, it is not always possible to straightforwardly deploy as an actual microservice each 3GPP-defined service of a NF, in what would be a simple one-to-one mapping between a NF's logical sub-functionalities and the microservices into which it is split.

The identification of microservices is facilitated by i) the decomposition of the NF's duties along well-defined, homogeneous responsibilities and ii) by understanding how they collaborate with each other [3]. While the former approach clearly depends on the specific NF under investigation, for the latter an API Gateway is proposed across all decomposed NFs to adopt a unified design in which all NFs feature a dedicated microservice which acts as a coordinator between "external" NFs services and "internal" NF-specific microservices. The API Gateway (GW) (realised as a stand-alone microservice) takes care of two responsibilities, i.e., i) exposing the existing 3GPP-specified SBI towards other NFs (thus, pragmatically ensuring the backward compatibility with the standard) and ii) shielding the custom microservices in the fashion of a De-Militarized Zone (DMZ).

According to this approach, in the following of this section, we propose a possible degree of decomposition of three functionally heterogeneous 5GC NFs via the identification of independent functional or data modules for each service provided by such NFs.

Due to the complexity and numerousness of the 5GC's NFs, an exhaustive and omni-comprehensive analysis of their decomposability into microservices is not feasible. Therefore the following exemplary three NFs were chosen for the following reasons:

- The chosen three network functions were implemented and trialled at technology readiness level 8 (i.e., actual system proven through successful mission operations) as part of the FUDGE-5G project [18].

- Unified Data Management (UDM) because it is a strictly necessary function even in minimum-footprint 5GC deployments, because it stores data (inducing the need to define its composing microservices also based on the non-shared databases they contain), and because it plays an essential role in enabling the statelessness and possible serverless implementation of the other NFs.

- Network Exposure Function (NEF) because it is a crucial enabler of 5G advanced features based on the interaction between the 5GC and external applications, which makes it a requested NF in beyond-minimum-footprint 5GC deployments.

- Cell Broadcast Centre Function (CBCF) because it substantially consists of a so called trusted Application Function (AF), and because it is utilised in some fundamental use cases of private 5G networks, e.g., in PPDR scenarios.

4.1. Example #1 – Unified Data Management

The UDM supports several functionalities including user identity and authorisation [2, Section 6.2.7]. These functionalities are exposed as NF services [2, Section 7.2.5] that are used in the procedures specified in [16, Section 5.2]. We first analyse the NF service specifications and identify the NF's responsibilities:

- *Authorization Management*: To grant service authorisation to positively identified users, based on operator policies, including subscription data.

- *UE Context Management*: To provide access to the dynamic state of the services being provided to the user.

- *Service Events Management*: To monitor events that may require a change in the services provided to the user, and notify consumer NFs that have manifested interest in those events.

We notice that the responsibilities are distributed over the UDM's 3GPP NF services. For instance, the *Non-IP Data Delivery (NIDD) Authorization* service [2, Section 7.2.5] has the responsibility of authorising NIDD, whereas the *Subscriber Data Management (SDM)* service [2, Section 7.2.5]

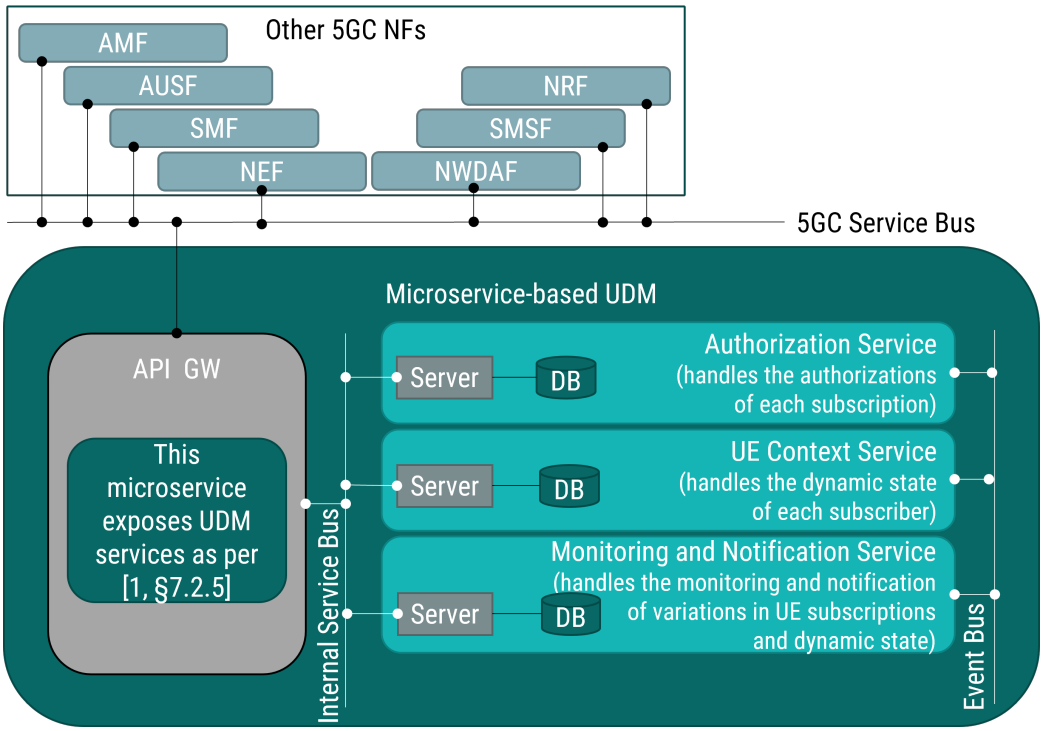

Figure 2: Proposed microservice-based Unified Data Management design.

has the responsibility of managing the data that is used to determine the subscriber's authorised services. Further, how subscriber data is provisioned is not specified, and the SDM is still responsible to notify NFs of subscriber data changes, mixing up the responsibilities of User Equipment (UE) Context Management and duplicating event notification.

After identifying the UDM's three independent responsibilities, we devised the UDM's architecture depicted in Figure 2. It shows the implemented UDM functionalities using three microservices, i.e., Authorisation Service, UE Context Service, Monitoring and Notification Service. In addition, as explained in the initial part of this section, we included a fourth microservice, the API Gateway, which exposes the UDM functionality through the 3GPP-specified SBIs while ensuring their security at the same time. As required by a microservice-based architecture, data is stored within each microservice, depicted by the Database (DB) component. However, the use of an external Unified Data Repository (UDR) is not excluded, which would

make the UDM more stateless with the downside of an increase in signalling traffic to allow each microservice to store data inside the UDR.

It should be highlighted that, as opposed to other UDM designs proposed in the literature [11], the proposed architecture ensures that the three "backend" microservices do not share data among each other, complying simultaneously with the definition of microservices given in Section 2 and with the criterion of state dependency proposed in Section 3.1: in absence of UDR, the UDM is stateful, but microservice-independent states are managed independently. Moreover, the architecture of Figure 2 fully respects the criterion of bottleneck mitigation and parallel execution (Section 3.1), because the split responsibilities guarantee an optimal allocability of computational and storage resources to each of the microservices, without dependencies from each other.

Our work on the UDM has not been exclusively theoretical. We implemented this microservice-based architecture into a prototypical UDM, whose functioning and design features were validated, for instance, in the demonstrational setup reported in [19]. In such a setup, the microservice-based UDM was part of a proof of concept aimed at demonstrating the implementation of an on-demand provisioning procedure of a Network Slice Subnet (NSS) composed of VNFs from different developers, mimicking potentially distinct vendors. The demonstration included a network management system conforming to the 3GPP Service-Based Management Architecture (SBMA), an ETSI MANO orchestrator, and a NFV Infrastructure (NFVI). In particular, this work demonstrated the provisioning, configuration, and control of an exemplary NSS. Although [19] did not specifically focus on microservices and NF architecture but rather on network slice management, we cite it here because it proves that our microservice-based UDM architecture is actually deployable. In particular, such demonstrational setup tangibly benefited from the proposed microservice-based approach in that: i) the UDM's API Gateway guaranteed correct inter-working with other non-microservice-based NFs of the 5GC via the SBI as requested by the standard; ii) in compliance with the definition of responsibilities and purposes of microservices, separate and independent programmers were able to develop the distinct microservices that compose the UDM, achieving a fast and parallelised deployment of the function within the modular architecture.

4.2. Example #2 – NEF

The NEF is located between external AFs and the 5GC. It is responsible for providing a point of access for external applications to access the 5GC securely and consume its exposed services.

As shown in Figure 3, the envisioned microservice-based NEF encompasses a scalable and stateless API Gateway that processes RESTful requests to the NEF and forwards them to the correct NEF microservice instance. Behind it, there are three responsibility-driven microservices, i.e.:

- UE Communication Policies service,
- Subscription and Notification service, and
- Internet of Things (IoT) and Low-power Communication service.

The first one deals with UE policy request and configuration, e.g., Quality of Service (QoS), traffic influence. The second one implements the publish/subscribe operations intended to monitor the network and the UEs. Finally, the third one is responsible for operations towards IoT and low-power devices, e.g., NIDD, Background Data Transfer Policy (BDTP). Each of the three microservices consumes the RESTful APIs exposed by the 5GC NF producers through the 5GC service bus. By having this decomposition, both a logical separation and self-compartmentalisation are achieved, with greater benefits towards reliability, security, and performance. Moreover, it aligns seamlessly with the microservices framework from Section 2, and adheres in full to the criteria outlined in Section 3.1. Notably, bottleneck mitigation, parallel execution and resilience directly stem from the logical separation of microservices above. Concurrently, state dependency is managed through external microservices (e.g., from cloud providers) that handle a very limited set of stateful data in a distributed, yet reliable manner. This approach aligns with the latest cloud principles, and further ensures our microservices remain fully stateless.

An early version of a decomposed NEF [20], with a very simplistic 1:1 split of stateless and stateful components (not microservice-based), was validated in a broader context. This work aimed at similar benefits as our proposal, and achieved good performance and reliability results despite its very simplistics approach. Moreover, there were clear advantages towards security when paired with a security framework tailored for the protection of such decomposed services. In fact, it became clear that the most pressing issue for

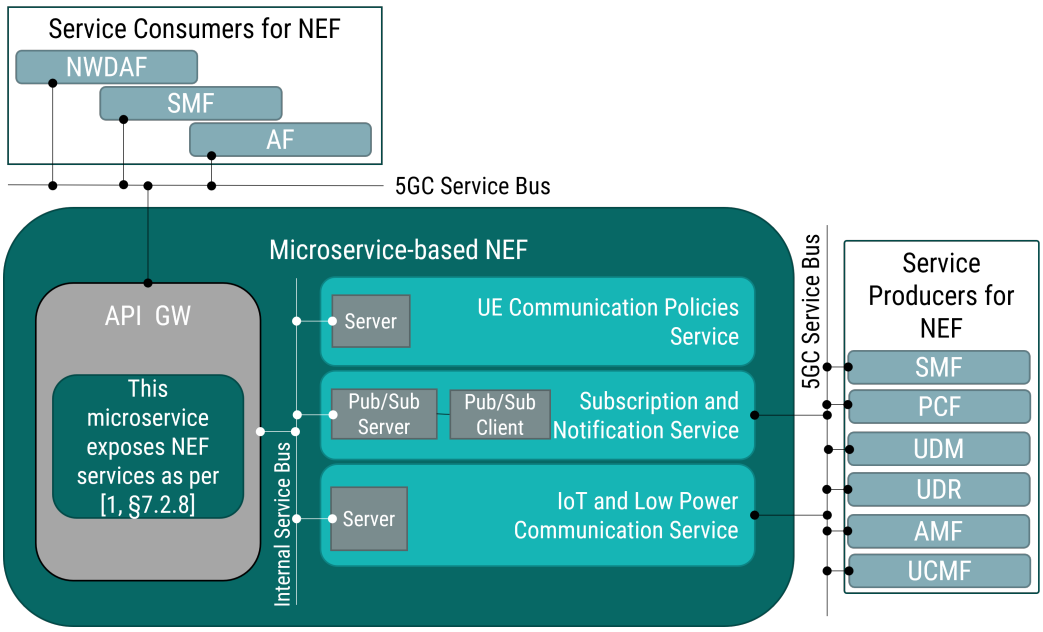

Figure 3: Proposed microservice-based Network Exposure Function design.

the NEF is indeed security, since by design it is the most exposed component of the 5GC and it presents a number of possible attack vectors that need to be considered in order to: i) prevent any threats from reaching critical components within the core; ii) ensure that legitimate AF traffic has a low or null impact from any malicious source; iii) performance and reliability of NEF, as well as its correct operation, are guaranteed. Hence, we fully aim to make our microservice-based approach compatible with a holistic AI-driven security approach that may secure it without compromising any of its isolation, performance and resilience requirements. A very preliminary validation was undertaken to assess the performance contrast between a monolithic NEF and our proposed microservice-based NEF. Stress tests were performed, involving multiple requests for QoS policy changes routed through NEF from AF to Policy Control Function (PCF). Two distinct setups were employed: one with consolidated microservices on a single machine, and another with distributed microservices (1 per machine), while the AF and the rest of the 5GC NFs occupied separate machines. The experiment involved a significant number of AF instances transmitting requests across multiple runs. The outcome of these preliminary tests, while not elaborated upon in this

architecture-focused paper, hinted at performance distinctions between the setups. These findings underscore the potential advantages of a decomposition strategy—enhancing scalability, compartmentalisation, and even performance. Consequently, the decomposition of NEF into a microservice-based architecture and its significance is reaffirmed as a pivotal gateway for 5GC interaction within the proposed architectural framework.

4.3. Example #3 – Cell Broadcast Centre Function

The CBCF is an instantiation of an AF in the 5GC architecture and provides a public warning service in which government organisations can submit text messages to be broadcast to mobile devices in the alert area. Figure 4 shows the architecture of a decomposed CBCF based on a responsibility-driven design as set out in Section 3. The *Ingest API* of the CBCF receives public warning messages from a government-dependent system, i.e., a Cell Broadcast Entity (CBE). In order to support these different CBE instances with minimised effort and maximised product flexibility, it is key to develop the Ingest API as a stateless microservice that can easily be replaced. After validation by the Ingest API, the message is passed on the internal bus to the *CB Kernel* service, which requests the *Cell Selection* microservice to determine the cell sites that cover the alert area as indicated in the area component of the CAP message. The coverage area of each cell is considered in the cell selection procedure. However, the way the coverage area is determined depends on the cell site information the operator can provide, which could lead to different implementations of the cell selection feature; hence, to different microservices. *Drivers* supporting specific network generations (2G, 3G, 4G or 5G) are instantiated as microservices according to the operator's network deployment; they also support static scaling when the number of cells in any network becomes big. It is anticipated that when 6G arrives, a microservice with a 6G driver can be added. Given the low number of public warning messages that need to be broadcast, dynamic scaling plays no role. Finally, a dedicated microservice manages *subscriptions* with the AMF and the NRF and renews subscriptions before they expire. This microservice cancels all subscriptions upon a graceful shutdown of the CBCF. The subscriptions are required for the CBCF to receive indications from the AMF about success or failure of warning message delivery.

The microservices as shown in Figure 4 have been implemented, however, only 5G networks are currently supported. The described microservice-based CBCF has been integrated with an instance of the Cumucore 5GC and has

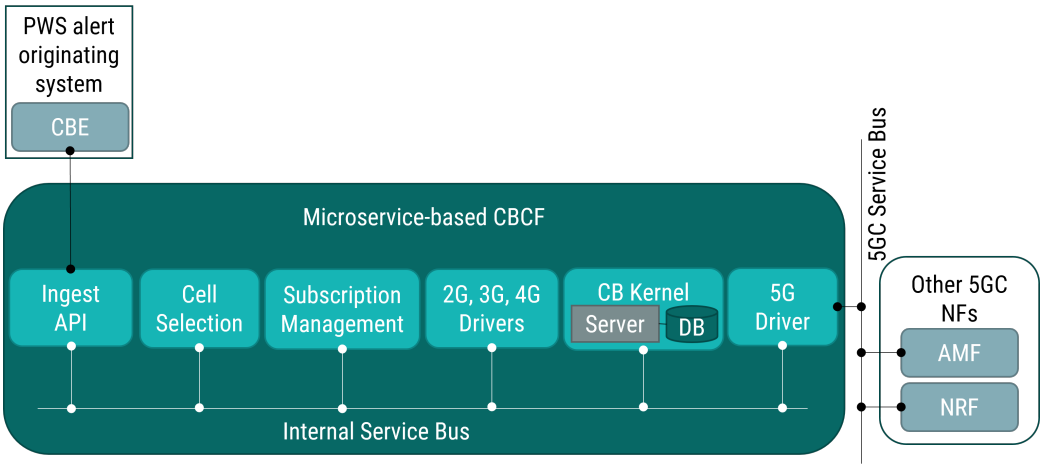

Figure 4: Proposed microservice-based Cell Broadcast Centre Function design.

been demonstrated and documented as part of the FUDGE-5G project at Telenor in Norway [18]. For the demonstration some ten 5G phones of a few different brands were available and they all presented the warning message a few seconds after the message was initiated on the CBCF. An advantage of using a CBCF consisting of microservices is that individual microservices of the CBCF can easily be upgraded without the public warning service becoming unavailable during the upgrade.

5. Considerations for Beyond 5G Systems

The decomposition examples of the design patterns described in Section 4 demonstrate the ability to turn a monolithic software into a set of independent components. These components then use a service bus to communicate among each other, where the service bus brings the required communication methods for the stateless implementation of a component to retrieve information, without the need to figure out from which specific endpoint to retrieve it from, e.g., publish-subscribe.

What becomes apparent from the examples in Section 4 is the differentiation between the 5GC service bus and NF's internal service bus. While the 5GC service bus follows the SBI specification in 3GPP (i.e., HTTP with JSON payload), the internal service bus may consist of a vendor-specific realisation (e.g., Kafka). Following the 3GPP standard, all 5GC service bus communication can be expected to be handled by the SCP, if the 5GC is

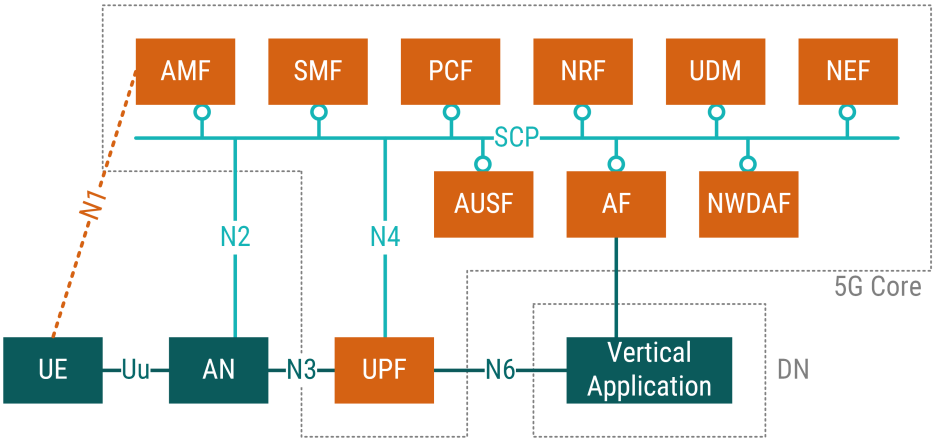

Figure 5: Envisaged beyond-5G system architecture.

operating under the so-called Model C or D [2, Section E.1], as opposed to Model A and B which refer to a direct communication where no SCP is present. Once 5GC's NFs are deployed as microservices, the importance of the SCP for the realisation of a 5GC service is likely to increase in scenarios where the numerical footprint of UEs is in the thousands.

3GPP defines three deployment examples of the SCP [2, Section G.4], i.e. Independent Deployment Units, Service Mesh and Name-Based Routing. All three examples support Model C and D and come with various scopes as well as pros and cons when comparing them among each other. The importance of the SCP will increase when assessing industry-led whitepapers [21, 22] where an actual service routing capability will be required, as discussed hereafter. Figure 5 illustrates the envisaged system architecture based on these whitepapers for a beyond-5G system, where the SCP may substitute the 5GC service bus adopted by all examples of microservice-based NFs presented in Section 4. In order to allow a direct communication between UEs and any NF (and vice versa), one may argue to stop addressing the SCP directly for less signalling required to establish a communication between a consumer and producer.

As can be derived from Figure 5, the non-SBI-enabled interfaces carrying signalling traffic into a 5GC, that are, the N2 and N4, are also routed via the SCP. Even though these interfaces do not utilise HTTP, any SCP should be able to route standard IP traffic. The feasibility of such a beyond-5G system architecture has been demonstrated [23] using a commercial off-the-

shelf UE, access network, and 5GC. The SBA platform was composed of an SCP that is deployed as per [2, Section G.4], i.e., leveraging Name-based Routing [24], alongside a telco-centric, location-aware orchestrator for 5GCs implementing the design patterns described in this paper. This includes the guidelines on how to structure FQDNs for producer components of a 5GC and how consumers shall address them, as discussed within the NGMN Alliance [15]. In short, all SBIs of NF that utilise the 5GC service bus are registered against the SCP with an FQDN that has a parent domain which is identical across all NFs of the same 5GC, e.g., foo.com. Each NF acronym is then used as a sub-domain for the FQDN, e.g., amf.foo.com or nrf.foo.com. When a 5GC is deployed via the SBA platform, each consumer can query the parent domain from the orchestrator in order not to hard-code it into the software implementation. Instead, NFs know that they must communicate with a producer, e.g., the NRF to retrieve an access token, and through the implicit knowledge that all producers carry their acronyms as the sub-domain, the FQDN is nrf."parent domain" (e.g., nrf.foo.com). With the proposed architecture, when assessing the design patterns put into practice in the previous section, the differentiation of the service bus for inter-NF communication and the service bus for intra-NF communication may be revised. One key argument for 3GPP not to further decompose NFs and standardise their SBIs is to leave vendors the ability to differentiate themselves from their competitors through their software. However, it can be argued that if service routing (SCP) capabilities are made mandatory and combined with cloud-native 5GC orchestration capabilities, the internal service bus could leverage the SCP too. This of course requires all design patterns around naming of NFs, registration of these names against the SCP and potential isolation and QoS enforcement requirements to be properly defined and standardised, permitting multi-vendor deployments of a mobile telecommunication network. All in all, the proposed beyond-5G architecture deserves additional investigation efforts from both the scientific community as well as the standardisation community, in order to foster further discussions on it and eventually consensus.

6. Conclusion

This paper presented how the paradigm of microservices can be ported to the world of mobile telecommunications, advocating the need for a real *micro*service-based architecture for the core network. Having illustrated that

the straightforward approach based on the one-to-one mapping between each distinct standardised NF services and a single corresponding microservice does not fully hold, we investigated methodologies and challenges based on the state of the art, as well as on the direct hands-on experience of the authors, and we applied them to three exemplary NFs. The final aim was of defining a framework that can be inherited by researchers, architects, and developers while designing microservice-based (network) functions. The described experiences and the heterogeneity of the service provided by a 5GC make evident the absence of a one-fits-all solution, yet, via some concrete examples, we indicated an approach and the method of analysing the functions for independent responsibilities, allocating a microservice to each, and then adding an API gateway micorservice to provide the desired system interface and orchestration of the independent microservices. Furthermore, an outlook is provided on changes to 3GPP's system architecture in beyond 5G releases based on industry-led whitepapers. The evolution of SBA down to the terminal is complemented with a proposal on naming conventions of NFs for addressing them via FQDNs when a communication between a consumer and producer is desired.

Acknowledgment

This work was supported in part by the European Commission under the 5G-PPP project FUDGE-5G (H2020-ICT-42-2020 call, Grant Number 957242, `https://www.fudge-5g.eu`) and Horizon Europe SNS JU PREDICT-6G (Grant Number 101095890, `https://predict-6g.eu`). The views expressed in this contribution are those of the authors and do not necessarily represent the project.

References

[1] CNCF, Cloud native network function working group charter, 2021. URL: `https://github.com/cncf/cnf-wg/blob/main/charter.md`.

[2] 3GPP, Ts23.501: System architecture for the 5g system (5gs); stage 2 (release 17), 2023. URL: `https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/23501-h90.zip`.

[3] R. Wirfs-Brock, A. McKean, I. Jacobson, J. Vlissides, Object Design: Roles, Responsibilities, and Collaborations, Pearson Education, 2002.

[4] R. W. Macarthy, J. M. Bass, An empirical taxonomy of devops in practice, in: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2020, pp. 221–228. doi:10.1109/SEAA51224.2020.00046.

[5] Y. Li, Y. Lin, Y. Wang, K. Ye, C. Xu, Serverless computing: State-of-the-art, challenges and opportunities, IEEE Transactions on Services Computing 16 (2023) 1522–1539. doi:10.1109/TSC.2022.3166553.

[6] K. Mahmood, A. Gavras, A. Hecker, Non-Public-Networks - State of the art and way forward, 2022. URL: https://doi.org/10.5281/zenodo.7230191. doi:10.5281/zenodo.7230191.

[7] W. d. S. Coelho, A. Benhamiche, N. Perrot, S. Secci, Network function mapping: From 3g entities to 5g service-based functions decomposition, IEEE Communications Standards Magazine 4 (2020) 46–52. doi:10.1109/MCOMSTD.001.1900040.

[8] S. R. Chowdhury, M. A. Salahuddin, N. Limam, R. Boutaba, Re-architecting nfv ecosystem with microservices: State of the art and research challenges, IEEE Network 33 (2019) 168–176. doi:10.1109/MNET.2019.1800082.

[9] Y. Li, J. Huang, Q. Sun, T. Sun, S. Wang, Cognitive service architecture for 6g core network, IEEE Transactions on Industrial Informatics 17 (2021) 7193–7203. doi:10.1109/TII.2021.3063697.

[10] K. Du, X. Wen, L. Wang, T.-T. Nguyen, A cloud-native based access and mobility management function implementation in 5g core, in: 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1251–1256. doi:10.1109/ICCC51575.2020.9345262.

[11] Oracle, Unified data manager (udm) user's guide, 2020. URL: https://docs.oracle.com/communications/F25434_01/docs.10/UDM%20User%27s%20Guide/GUID-BB291E7E-22E2-4484-A902-45F7670A53A2.htm.

[12] T. Soenen, S. Van Rossem, W. Tavernier, F. Vicens, D. Valocchi, P. Trakadas, P. Karkazis, G. Xilouris, P. Eardley, S. Kolometsos, M.-A. Kourtis, D. Guija, S. Siddiqui, P. Hasselmeyer, J. Bonnet, D. Lopez, Insights from sonata: Implementing and integrating a microservice-based

nfv service platform with a devops methodology, in: NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–6. doi:10.1109/NOMS.2018.8406139.

[13] R. de Jesus Martins, A. Galante Dalla-Costa, J. A. Wickboldt, L. Zambenedetti Granville, Sweeten: Automated network management provisioning for 5g microservices-based virtual network functions, in: 2020 16th International Conference on Network and Service Management (CNSM), 2020, pp. 1–9. doi:10.23919/CNSM50824.2020.9269063.

[14] A. Wiggins, The 12 factor app methodology, 2017. URL: https://12factor.net.

[15] NGMN Alliance, Experience on cloud native adoption, 2022. URL: https://www.ngmn.org/wp-content/uploads/220128-Experience-on-Cloud-Native-Adoption-v1.1-Final.pdf.

[16] 3GPP, Ts 23.502: Procedures for the 5g system, stage 2 (release 17), 2023. URL: https://www.3gpp.org/ftp/Specs/archive/23_series/23.502/23502-h90.zip.

[17] G. Brown, Service-based architecture for 5g core networks, 2017. URL: https://img.lightreading.com/downloads/Service-Based-Architecture-for-5G-Core-Networks.pdf.

[18] FUDGE-5G Consortium, Deliverable 4.2: Final technical validation of 5g components with vertical trial, 2023. URL: https://www.fudge-5g.eu/download-file/572/VQcuTXbvGNkN2G2Q40N2.

[19] A. Ishaq, D. Ronzani, A. Spinato, N. Pietro, M. Centenaro, A. Bellin, D. Munaretto, Service-based management architecture for on-demand creation, configuration, and control of a network slice subnet, in: 2022 IEEE 8th International Conference on Network Softwarization (NetSoft), 2022, pp. 275–277. doi:10.1109/NetSoft54395.2022.9844081.

[20] L. Maglaras, I. Kantzavelou, Cybersecurity Issues in Emerging Technologies, Chapter 6: Securing components on a 5G core, CRC Press, 2021. doi:10.1201/9781003109952-6.

[21] M. K. Bahare, A. Gavras, M. Gramaglia, J. Cosmas, X. Li, O. Bulakci, A. Rahman, A. Kostopoulos, A. Mesodiakaki, D. Tsolkas, M. Ericson,

M. Boldi, M. Uusitalo, M. Ghoraishi, P. Rugeland, The 6G Architecture Landscape - European perspective, 2023. URL: https://doi.org/10.5281/zenodo.7313232. doi:10.5281/zenodo.7313232.

[22] NextG Alliance, 6g technologies, 2022. URL: https://www.nextgalliance.org/white_papers/6g-technologies/.

[23] S. Robitzsch, J. Ribes, A. S. Gomes, H. Rexha, L. Cordeiro, M. K. Al-Hares, M. Corici, D. Gomez-Barquero, Under trial: Evolved service-based architecture platform for mobile telecommunication networks, in: 2022 IEEE Future Networks World Forum (FNWF), 2022, pp. 694–700. doi:10.1109/FNWF55208.2022.00127.

[24] D. Trossen, S. Robitzsch, S. Hergenhan, J. Riihijarvi, M. Reed, M. Al-Naday, Service-based routing at the edge, 2019. URL: http://arxiv.org/abs/1907.01293.