ANR SCALER

2023-2026

D 4.1

Use Cases Requirements

Chef de file pour le D 4.1 : Ilhem FAJJARI
Co-rédacteurs : Wassim AROUI,
Joquim SOARES
Vania MARANGOZOVA

January 15, 2024

Contents

1 Introduction

In the dynamic and rapidly evolving landscape of telecommunications and e-commerce, the need for robust, scalable, and efficient technological solutions has considerably grown. This document provides a comprehensive study of the technical and functional requirements of a 5G-core network service and a web store application in e-commerce. It also provides insights into the SCALER platform's capabilities and the execution infrastructure designed to support these use cases.

In Section 2 the document first explores the shift towards cloud-native network functions and service-based architectures for 5G core networks. It gives insights into the specific technical and functional requirements of the 5G-core network service, detailing the network functions and their roles in ensuring seamless, secure, and efficient network operations.

In the second part of the document (Section 3), we focus on the e-commerce sector, where the adoption of microservices architecture is revolutionizing the way web store applications are developed and managed. The document gives an overview of the key components of an e-commerce application, the technical requirements for a microservices-based approach, and the functional requirements that ensure a user-friendly and secure online shopping experience.

In the last part (Section 4), the SCALER use case execution infrastructure is introduced, highlighting two environments: i) the private cloud environment provided by Eolas and ii) Grid'5000 national research infrastructure.

2 Real-world UC # 1 - 5G-core network service

The integration of 5G technology into telecommunications has triggered a shift in the potential of Telco's cellular networks. The advent of this advanced network generation has not only triggered unprecedented speeds but has also paved the path for a swift evolution of envisioned services. Among these services are innovative advancements such as augmented reality, virtual reality, autonomous vehicles, and Internet of Things (IoT). This transformative wave requires a substantial overhaul from telecom operators, compelling them to reconfigure their architectures and fortify their infrastructures to seamlessly align with these burgeoning demands.

From this viewpoint, 5G technology has introduced a groundbreaking concept that advocates for the disaggregation of networks, aiming to make them fully cloud-native. This innovative paradigm promotes the breakdown of network services into a collection of adaptable, autonomous, elastic, and segregated microservices, each securely enclosed within its container. These microservices, representing distinct network functions, aim to form a novel breed of scalable, self-repairing, and nimble network function known as CNF (Cloud Native Network Function) [1]. This evolution drives towards a more resilient, dynamic, and efficient network architecture poised to meet the evolving demands of modern connectivity. We remind the reader that a network function represents a software entity dedicated to executing specific tasks related to the management, control, or facilitation of network services

The 5G Core brings a significant transformation by replacing traditional network architectures with service-based interactions between Network Functions. This eliminates the need for nodes or network elements connected by interfaces. In the 5G Core architecture, each Network Function offers services to other Network Functions in the network, encompassing both control plane and data plane functionalities. Control services are accessed through Network Function interfaces connected to the common Service-Based Architecture (SBA). This means that the specific functionality provided by a Network Function is exposed and accessible through an API (Application Programming Interface) [2].

Among the network functions that the 5G core network is relying on, we find :

- **AMF—Access and Mobility management Function** is the gateway to the core network that ensures all control interactions with gNB (i.e., base station) and end users (i.e., UE).
- **SMF—Session Management Function** controls packet data unit (PDU) sessions on one or more instances of the User Plane Function (UPF), providing end-to-end connectivity to the UE.
- **UPF—User Plane Function** processes and forwards user data. The functionality of the UPF is controlled by the SMF. It interconnects with external IP networks and acts as an anchor point for the UEs towards external networks.
- **NRF—Network Repository Function** is the function responsible for discovering services in the 5G control plan. It is used by all other network functions in the control plane to record or retrieve information about other functions.
- **UDM—Unified Data management Function** is a front-end for the user subscription data stored in the UDR (see below). The UDM uses subscription data that may be stored in the UDR to execute application logic like access authorization, registration management and reachability for terminating event e.g., SMS.
- **UDR—Unified Data Repository** is the database where various types of data are stored. Important data is of course the subscription data or data defining network or user policies. UDR storage and access to data is a service offered to other network functions, specifically UDM, PCF, and NEF.
- **UDSF—Unstructured Data Storage Function** is an optional function that allows other NFs to store dynamic context data outside the NF itself. This is sometimes referred to as a “stateless” implementation.
- **AUSF—Authentication Server Function** is responsible for authentication of UEs requesting 5G network services. Specifically, it communicates with the Unified Data Management (UDM) function to retrieve subscriber information, which is recorded by the telco operator.
- **PCF—Policy Control Function** provides policy control for session management, for access and mobility, for UE access selection and PDU session selection.
- **NSSF—Network Slice Selection Function** The NSSF selects the set of network slice instances for the UE and the set of AMFs that should serve the UE.
- **NEF—Network Exposure Function** exposes events and capabilities from the 5G system towards applications and network functions inside and outside the operator’s network.
- **SEPP—Security Edge Protection Proxy** is used to protect the signaling between operators in roaming scenarios.
- **N3IWF—Non-3GPP Inter Working Function** is used for integrating non-3GPP accesses with the 5G core.
- **LMF—Location Management Function** serves the purpose of determining the User Equipment’s (UE) location. It achieves this by retrieving location estimates directly from the UE and obtaining additional location measurements and data from the Next-Generation Radio Access Network (NG-RAN).

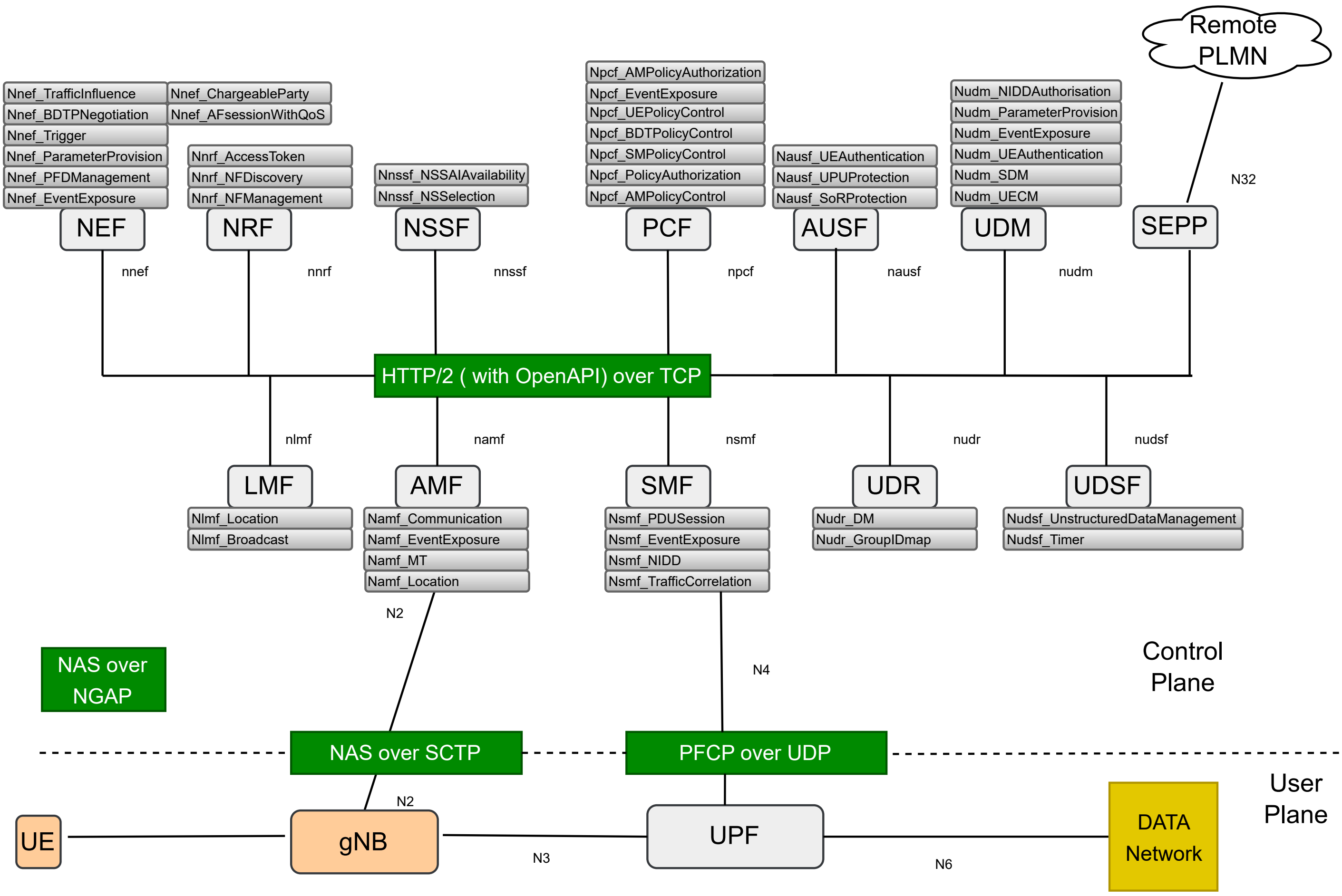

Figure 1: 5G System Architecture

2.1 Key 5G Core Technical Requirements and KPIs

As the next generation of mobile communication, the 5G Core network brings forth a range of technical requirements and Key Performance Indicators (KPIs) that are specifically addressed to meet the evolving needs of the industry. These requirements and KPIs are crucial in ensuring the successful deployment and operation of the 5G Core network. Let's delve into the key technical requirements and KPIs that the 5G Core network focuses on:

- **Seamless mobility**: The 5G Core network is designed to support seamless mobility across various cell types and radio access technologies. It enables smooth handovers and transitions between different network elements, ensuring uninterrupted connectivity for users.
- **Context-aware optimal connection**: Leveraging contextual information such as device, user, environment, and network conditions, the 5G Core network aims to provide the best possible connection and services at all times. This ensures optimal user experiences and personalized connectivity.
- **Wired/wireless terminal switching**: The 5G Core network facilitates fast and efficient switching between wired and wireless devices, enabling seamless device and session mobility. This capability allows for quick transfers between different types of devices, enhancing user convenience.

Requirement	KPI	Value
E2E latency	The time needed to transfer a given piece of information from a source to a destination measured at the communication interface from the moment it is transmitted by the source to the moment it is successfully received at the destination.	User plane (1 ms) Control plane (20 ms)
Reliability	The percentage of packets successfully delivered within the required time constraint out of all transmitted packets.	99.999 %
Availability	The percentage of time during which the network is operational and accessible to the users.	99.999 %
Mobility Interruption Time	The shortest time duration supported by the system during which a user terminal cannot exchange user plane packets with any base station during transitions	0 ms
Peak Data Rate	The maximum achievable data rate under ideal conditions (in bit/s)	Downlink 20 Gbit/s Uplink 10 Gbits/s
Packet Error Rate	The upper bound of the rate of packets that have been processed by the sender of a link layer protocol but that are not successfully delivered for the corresponding receiver of the upper layer	10^{-6}
Energy Efficiency	The capability of the network to deliver high performance and data speeds while minimizing power consumption	Qualitative

Table 1: Technical Requirements KPIs Examples for 5GC Use Case

- **Single ID for multiple access**: Irrespective of the access network, the 5G Core network recognizes a mobile terminal as a single entity. This means that users can maintain a consistent identity and experience regardless of the network they are connected to, providing a seamless and unified connectivity experience.
- **Lightweight signaling**: The 5G Core network requires efficient signaling protocols capable of accommodating various terminals, including massive Machine Type Communication (MTC) devices. These protocols are designed to be lightweight, minimizing overhead and optimizing network resources.
- **Distributed architecture**: To handle the expected traffic increase of up to 1000 times, the 5G Core network maintains a distributed network structure. This architecture ensures scalability and efficient resource management, enabling the network to handle the growing demands of future communication.
- **Multiple radio access technologies interworking**: The 5G Core network requires an architecture that enables seamless flow over multiple radio access technologies. This ensures uninterrupted service continuity, even in scenarios with limited bandwidth in wireless access. The network efficiently interworks with different radio access technologies to deliver high-volume services cost-effectively.
- **Fine-grained location tracking**: The 5G Core network provides a virtualization environment

and robust support for reconfiguration and upgrades within the core network. This allows for cost-effective alterations without the need to make changes to the physical network infrastructure. The network can adapt and evolve efficiently to meet changing requirements and technological advancements.

- **Flexible reconfiguration and upgrade**: The 5G Core network provides a virtualization environment and robust support for reconfiguration and upgrades within the core network. This allows for cost-effective alterations without the need to make changes to the physical network infrastructure. The network can adapt and evolve efficiently to meet changing requirements and technological advancements.
- **Network on-demand**: The 5G Core network is capable of constructing the network infrastructure based on specific parameters such as Quality of Service (QoS), Quality of Experience (QoE), charging mechanisms, and distinct service attributes. This ensures a tailored network configuration that meets various requirements and improves the user experience. The network can be dynamically provisioned to meet specific needs, optimizing resource allocation and enhancing service delivery.

Examples for the technical requirements KPIs and their values are given in Table 1.

2.2 Key UC Functional Requirements and KPIs

In the following are described some major scenarios of the 5G network functions.

2.2.1 Registration

Registration is the initial procedure that a User Equipment (UE) undergoes after being powered on. Its purpose is to establish a connection with the network and enable the UE to receive services. The registration procedure can be summarized in the following steps:

1. The UE sends a NAS Registration Request message to AMF via the radio access network (R)AN. If temporary UE identities (5G-S-TMSI or GUAMI) are included and the (R)AN can map these to a valid AMF, the (R)AN forwards the NAS message to that AMF. Otherwise the (R)AN selects an AMF, based on Requested NSSAI [reference to slicing] or a configured default AMF, and forwards the NAS message to that AMF.
2. In case a new AMF is selected (e.g., because the UE registers in an area not served by the old AMF), and the UE provided a temporary UE identity containing the identity of the old AMF, the new AMF retrieves the UE context from the old AMF.
3. Authentication within the network is executed through the utilization of either the 5G Authentication and Key Agreement (AKA) protocol or the Extensible Authentication Protocol for AKA (EAP-AKA), ensuring robust security measures for user verification and access control.
4. In case a new AMF has been selected, the new AMF indicates to the old AMF that it is now taking over as serving AMF for the UE.
5. The AMF registers as serving AMF for the UE in the specific access technology (3GPP access or non-3GPP access) and requests subscription data and subscribes to subscription data updates using from the UDM. The UDM notifies the old AMF that it is deregistered in the UDM.

6. In case Access and Mobility management policies are deployed, the AMF initiates establishment of the AM policy association with PCF and retrieves the AM policies.

7. If the UE has indicated that it wants to activate User Plane connection for existing PDU Sessions, the AMF in the invokes the required SMF operation for those PDU Sessions.

8. If the Registration procedure is successful so far, the AMF provides a NAS Registration Accept to the UE. In some cases, the UE sends a NAS Registration Complete message to the AMF.

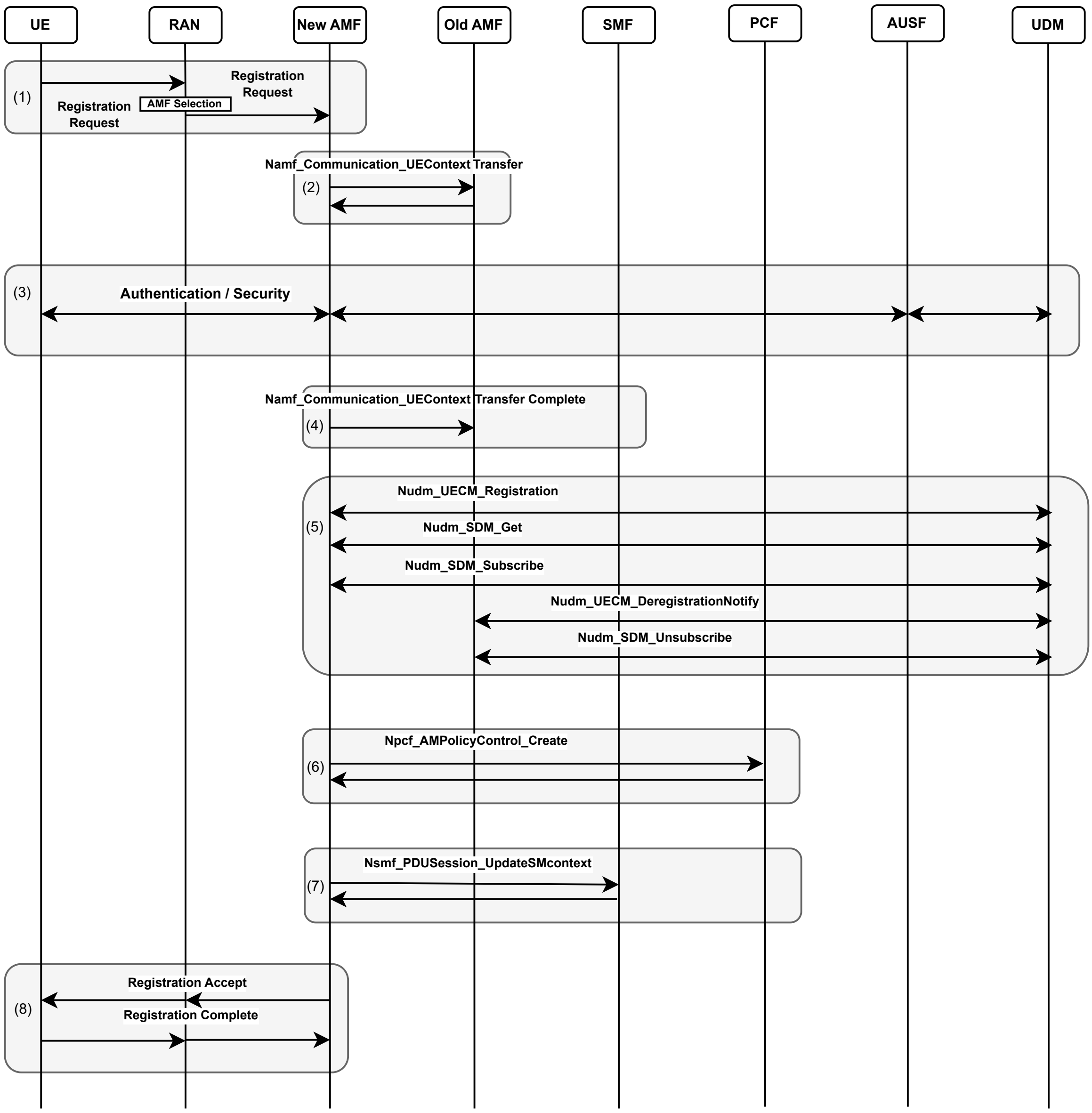

Figure 2: UE Registration Procedure

2.2.2 PDU Session Establishment

The PDU (Packet Data Unit) Session Establishment procedure is a process that starts when a User Equipment (UE) wants to create a new PDU Session or handover a PDU Session between non-3GPP access to 3GPP access. The PDU Session Establishment procedure is always initiated by the

UE, but sometimes the network might send a signal (a device trigger message) to an application on the UE. When the UE's application receives this message, it might decide to start the PDU Session Establishment procedure based on the information it received. The actual steps involved in this procedure are the following :

1. The UE transmits a 5GSM (5G Session Management) NAS PDU Session Establishment message to the AMF. This message contains crucial details like the PDU Session Id, the requested DNN (Data Network Name) along with the desired S-NSSAI (Single Network Slice Selection Assistance Information), and the type of PDU Session needed. The AMF, upon receiving this message, handles the NAS security. If the PDU Session Establishment is a request for a new PDU Session, the AMF selects a new SMF. The AMF may use the NRF to discover available SMFs serving the specific DNN and S-NSSAI. If the PDU Session Establishment is a request for handing over an existing PDU Session, the AMF uses its context about the UE to identify which SMF currently manages that particular PDU Session Id.

2. The AMF directs the 5GSM container, which holds the PDU Session Establishment message, to the SMF. Upon receiving this container, the SMF retrieves the UE's subscription data related to Session Management from the UDM and also subscribes to subscription data updates from UDM.

3. A secondary authentication may be performed between the UE and an external AAA (Authentication, Authorization, and Accounting service) on the Data Network.

4. The SMF selects a PCF (Policy Control Function) and initiates a policy session to obtain the initial set of PCC (Policy and Charging Control) rules. The SMF also selects a UE IP address and UPF and initiates a N4 session establishment towards that UPF.

5. The SMF transmits a 5GSM NAS PDU Session Establishment accept message to the UE, providing both the UPF GTP-U tunneling endpoint specifics and QoS details directed to the (R)AN. The AMF serves as the conduit for sending this message.

6. The AMF generates an N2 message that carries both the NAS message (PDU Session Establishment accept) and the essential PDU Session specifics for the (R)AN. This includes details about GTP-U tunneling and QoS information. Subsequently, the (R)AN configures the necessary resources toward the UE and replies to the AMF, providing information about the (R)AN's GTP-U tunnel endpoint. Once this process is completed, the up-link data path becomes operational and available for use.

7. The AMF forwards the PDU Session information received from (R)AN to the SMF so that SMF can provide the (R)AN GTP-U tunnel endpoint to the UPF for downlink forwarding. With this action completed, the downlink data path is now operational and available for use.

8. Finally, the SMF then registers itself in UDM, indicating its role as the entity serving the specific PDU Session Id.

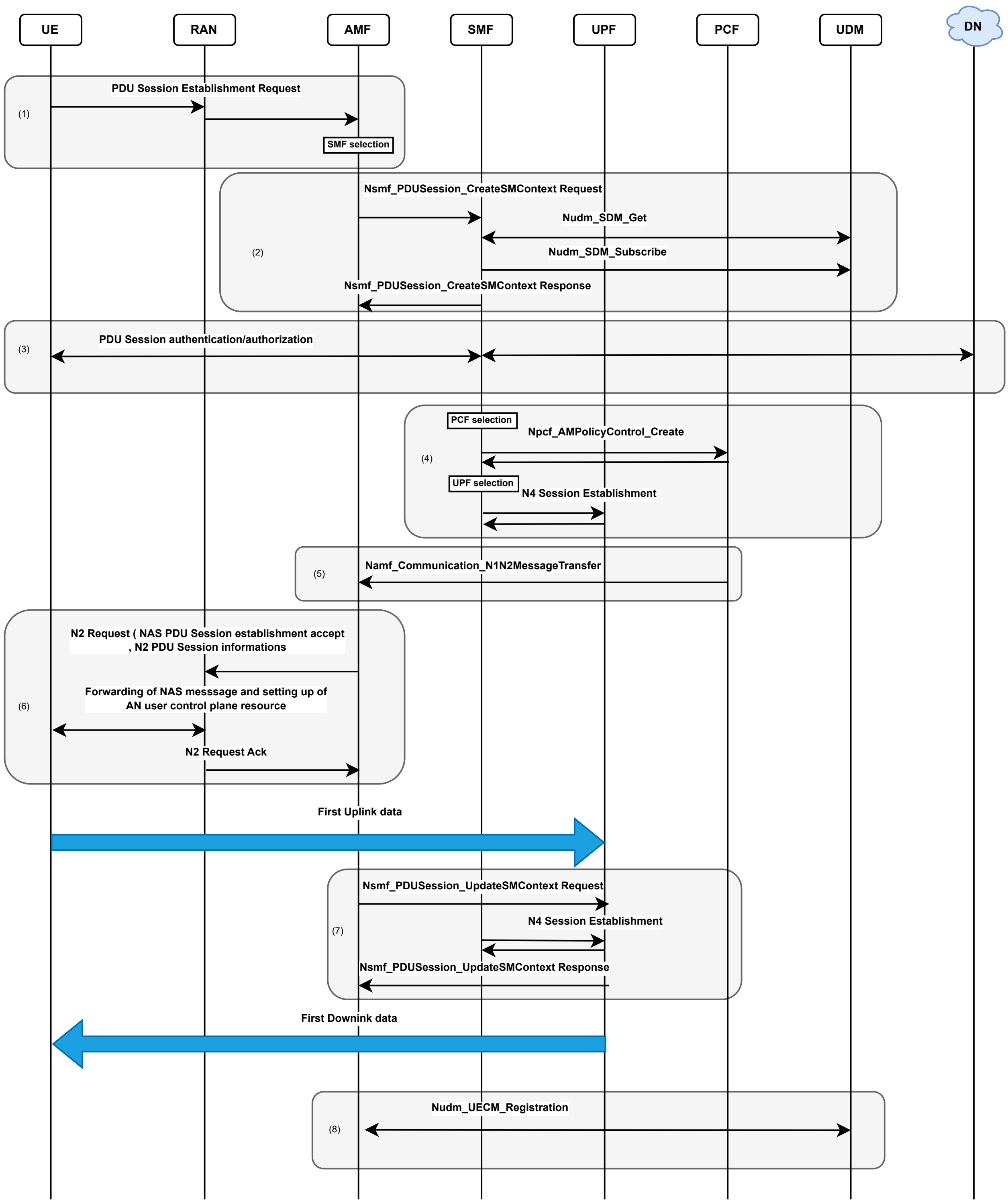

Figure 3: PDU Session Establishment Procedure

2.2.3 Deregistration

The deregistration process lets the UE tell the network it doesn't want 5G system access any longer, and vice versa—the network can inform the UE it no longer has 5G system access. The procedure contains the following steps:

1. The UE sends a NAS Deregistration Request message to the AMF through the (R)AN.
2. The AMF notifies every SMF that holds an active Session Management context, about the release of this specific context. Subsequently, each SMF informs other Network Functions about the PDU Session release, involving actions such as terminating N4 sessions and associated resources, ending policy associations with the PCF, and deregistering from the UDM for that specific PDU Session ID.
3. The AMF terminates the AM (Access and mobility) policy association with the PCF, if such an association exists.
4. The AMF terminates the UE policy association with the PCF, if such an association exists.
5. The AMF sends a Deregistration Accept message to the UE
6. Finally, the AMF instructs (R)AN to release the N2 UE context.

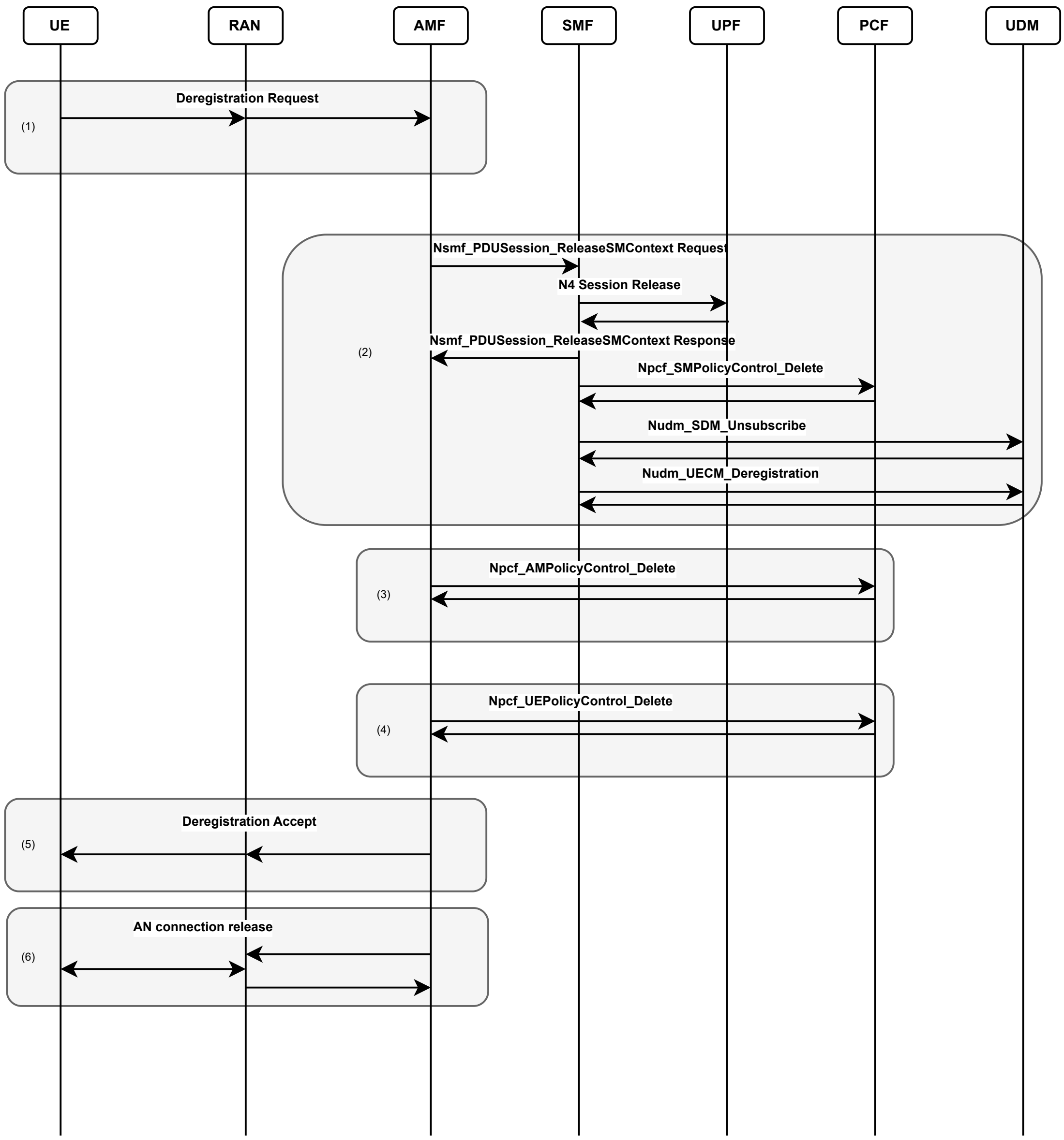

Figure 4: UE Deregistration Procedure

2.2.4 Handover

Handover procedures in 5G system are used to handover a UE from a source NG-RAN node to a target NG-RAN node. The procedure is briefly described in the following.

1. Before starting the handover process toward the core network, NG-RAN conducts essential handover preparation and executes signaling both within NG-RAN and between NG-RAN and the UE.

2. Once the handover is confirmed within NG-RAN, a N2 Path Switch Request is transmitted to notify that the UE has moved to a new target cell. This message contains details regarding the PDU Session to be switched.

3. The AMF informs every relevant SMF holding a PDU Session impacted by the handover. These SMFs subsequently provide the NG-RAN N3 tunnel details to the UPFs. This enables the transmission of down-link data to the new NG-RAN node.

4. Following the down-link path shift to the target NG-RAN, the UPF directs down-link packets to this new destination. To help the Target NG-RAN's reordering process, the UPF transmits "end marker" GTP-U packets for each N3 tunnel on the old path (toward the source NG-RAN). This action signals the arrival of the final down-link packet on the old path, enabling the removal of the forwarding tunnel between the source and target NG-RAN. Consequently, the target NG-RAN ensures sequential delivery of down-link packets to the UE.

5. Upon receiving responses from all SMFs in step 3, the AMF consolidates the gathered UPF N3 tunnel information and transmits this compiled data to the NG-RAN. This action enables the NG-RAN to set up the necessary up-link N3 data path(s).

6. The target NG-RAN then informs source NG-RAN that the handover is successfully completed so that source NG-RAN can release the associated resources.

7. At times, following the completion of a handover, the UE may need to trigger a Mobility Registration procedure. This situation can arise if the handover causes the UE to relocate beyond its designated Registration Area.

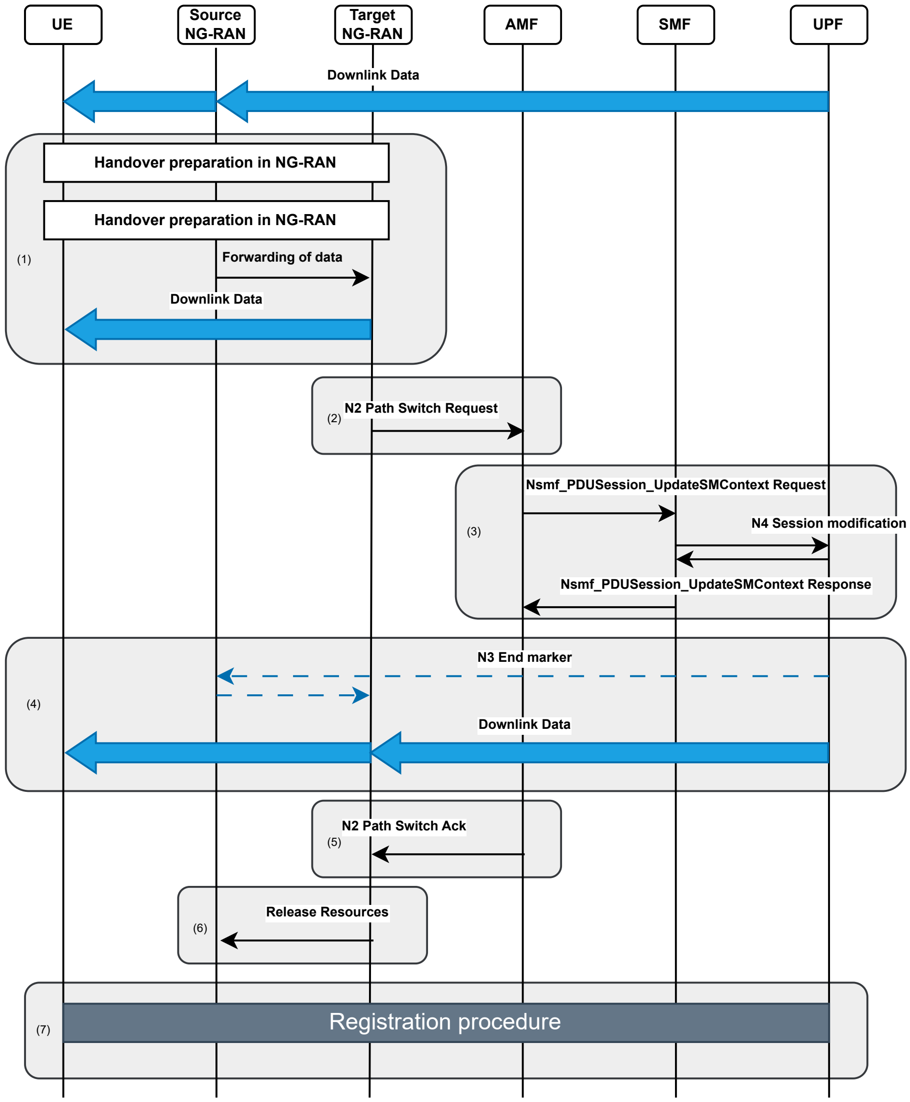

Figure 5: Handover Procedure

3 Real-world # 2 - Web store application (e-commerce, retail)

In the ever-evolving e-commerce industry, an increasing number of companies are turning to a microservices-based software development approach to replace their legacy monolithic applications. This migration represents a significant shift in how companies design, develop, and deploy their online commerce platforms. The transition to microservices offers numerous benefits, including improved scalability, agile development, enhanced error management, and greater technological flexibility. However, it is essential to acknowledge the expected complexities, such as management, inter-service communication, testing, and security. Companies considering this transition must engage in careful planning and execution to maximize the advantages while minimizing potential challenges.

Microservices have introduced several aspects that have proven to be very beneficial for e-commerce applications, namely:

1. **Scalability on Demand:** e-commerce websites often experience fluctuating levels of traffic, typically during sales, holidays, or special events. Microservices allow e-commerce platforms to scale individual services to meet increased demand, ensuring a smooth and responsive shopping experience for customers.
2. **Enhanced Resilience:** applications need to be highly available and resilient. Microservices, with their isolated nature, make it easier to design redundancy and fail-over mechanisms, ensuring uninterrupted service even if one component fails.
3. **Faster Development Cycles:** the microservices architecture encourages agile development practices. E-commerce companies can release updates, new features, and bug fixes more frequently, responding quickly to market changes and customer feedback.
4. **A/B Testing and Experimentation:** microservices facilitate A/B testing and experimentation. Different versions of a service can be deployed and tested independently to measure their impact on user engagement, conversion rates, and overall user experience.
5. **API-First Design:** microservices are designed with well-defined APIs, making it easier for different services to interact with each other. This API-first approach simplifies the integration of new services or third-party tools.
6. **Adaptation to Market Trends:** e-commerce is a dynamic industry, and market trends change rapidly. Microservices provide the flexibility to adapt to these changes by allowing for the introduction of new services or the modification of existing ones without overhauling the entire system.

The microservice paradigm thus ultimately leads to improved customer experiences and increased competitiveness in the e-commerce sector.

An e-commerce application typically consists of various key functions, that will be detailed later on this document, each serving a specific purpose to enable a smooth shopping experience for users. These functions collectively create a comprehensive e-commerce application that offers a user-friendly, secure, and feature-rich shopping environment, meeting the demands of both customers and businesses in the digital marketplace. The experiments for the web store application use case will use one or more open source e-commerce projects adapted to industry standards and the needs of the SCALER project. We can cite the following:

- Teastore [3, 4]
  - Emulates a basic web store that contains 6 microservices (5 services and 1 registry)
  - Each service may be replicated without limit and deployed on separate devices as desired
- DeathStarBench [5, 6]
  - Includes multiple use cases like Social Network, Hotel reservation or E-commerce site
  - Focuses on large-scale microservice applications that stress typical datacenter design (inspired by Amazon.com, Netflix and Twitter web applications)
- OpenTelemetry Astronomy Shop [7, 8]
  - Created by OpenTelemetry team to illustrate code instrumentation
  - Composed of microservices written in different programming languages that talk to each other over gRPC and HTTP; and a load generator which uses Locust [9] to fake user traffic
- TrainTicket [10, 11]
  - Proposes a service architecture that contains 41 microservices
  - Can be used with tracing software like Jaeger [12]
- Online Boutique [13]
  - Proposes a service architecture that contains 11 microservices.
  - Google uses this application to demonstrate the use of technologies like Kubernetes, GKE, Istio, Stackdriver, and gRPC.
- Robot Shop [14]
  - Stan's Robot Shop is a sample microservice application you can use as a sandbox to test and learn containerised application orchestration and monitoring techniques.
  - Instana[1] uses this application to demonstrate the use of its tracing techniques.

3.1 Key UC Technical Requirements and KPIs

In a microservices architecture for an e-commerce web application, several technical requirements are crucial for successful deployment and operation. These requirements include:

- **Scalability:** the ability to scale individual components of the application independently to handle varying loads, especially during high-traffic events like sales, promotions or daily peak load.
- **Performance and Efficiency:** optimized services to ensure fast response times and efficient resource usage, which is crucial for maintaining a positive user experience (i.e. E2E latency, number of requests per seconds, core web vitals [16]).

[1] https://www.ibm.com/fr-fr/products/instana

Requirement	KPI	Metric	Value
Scalability	Service Scaling Time	Time taken to spin up additional instances of a service.	< 5 minutes to scale up
Performance and Efficiency	Response Time	Average response time of service requests.	< 300 milliseconds
Reliability and Availability	SLA, SLO [15]	Percentage of time the application is operational.	SLA : 99.9% uptime SLO : 99.5% uptime
Security	Number of Security Breaches	Count of security incidents per year.	0 incident
API Gateway	API Error Rate	Percentage of API requests that result in errors.	<0.5%
Monitoring and Logging	Mean Time to Detection (MTTD)	Average time to detect an issue.	< 30 minutes
Database Management	Database Query Performance	Average response time for database queries.	< 50 milliseconds
Inter-Service Communication	Message Failure Rate	Percentage of failed messages between services.	< 0.1%
Resilience and Error Handling	Error Rate	Number of errors per thousand transactions.	< 10 errors

Table 2: Technical Requirements KPIs Examples for the e-commerce Usecase

- **Reliability and Availability:** high availability and fault tolerance are vital. This includes strategies like replication, load balancing, and the use of circuit breakers to prevent failures from cascading. To improve reliability, stakeholders can implement custom health check or status page that generate alerts in the event of an technical outage (e.g. : valid database connection, valid third-party service, etc ...).
- **Security:** robust security measures to protect sensitive data, such as user information and payment details. This includes secure communication between services, data encryption, and adherence to compliance standards like PCI DSS for payment processing.
- **API Gateway:** a single entry point for all clients (web, mobile, etc.), which routes requests to the appropriate microservice and handles cross-cutting concerns like authentication, SSL termination, and rate limiting.
- **Monitoring and Logging:** comprehensive monitoring and logging to track the health of each microservice, user activities, and system anomalies.
- **Database Management:** each microservice should ideally have its own database to ensure loose coupling, but this must be balanced with the complexity of managing multiple databases.
- **Inter-Service Communication:** efficient and reliable communication mechanisms, like RESTful APIs or messaging queues, depending on the use case.
- **Resilience and Error Handling:** implementing strategies for graceful degradation in case of partial system failure, including retries, timeouts, and fallback methods.

Depending on the use case, Table 2 shows some examples of KPIs that could be defined to achieve efficient e-commerce application.

3.2 Key UC Functional Requirements and KPIs

An e-commerce application typically consists of various key functions, each serving a specific purpose to enable a smooth shopping experience for users. Functional requirements for an e-commerce web application describe what the system should do and how it should perform specific functions. These requirements are typically centered around user interactions, business processes, and data management. Here's a non exhaustive list of common functional requirements for an e-commerce web application:

1. **User Account Management.** The application must allow users to create, manage, and delete their accounts, providing features like registration, login, password recovery, and profile editing.

2. **Product Catalog** The application should enable the listing, categorization, detailed description, prices, and availability of products, including features like search, filter, and sort options to help users easily find products.

3. **Search and Navigation.** This function provides search capabilities and navigation tools to help users find products efficiently. It may include features like filters, sorting options, and category browsing.

4. **Shopping Cart.** Users must be able to add products to a virtual shopping cart, modify quantities, remove items, and view a summary of their potential purchases.

5. **Order Management.** The system should facilitate the entire ordering process, from placing an order to tracking its status, and allow for order modifications, cancellations, and returns.

6. **Checkout Process.** The application should secure the processing of payments with various methods (like credit cards, PayPal, or digital wallets), provide delivery options and handle transactions, including invoicing and receipt generation. This involves a series of steps for users before finalizing the purchase.

7. **Shipping and Delivery Management.** The application should manage the logistics of shipping, including options for different shipping methods, costs, and tracking of deliveries.

8. **Customer Service.** The application is to provide features for customer support, such as live chat, email, or phone support. A FAQ section is also necessary for addressing customer inquiries and issues. Users should also be able to rate and review products, with mechanisms in place to manage and display these reviews authentically.

9. **Inventory Management.** This feature tracks inventory levels, updating stock status, and managing supply chain logistics in real-time.

10. **Marketing and Promotions.** The application is to support marketing campaigns, discount codes, and promotional events to attract and retain customers.

11. **Analytics and Reporting.** The application should provide analytics on sales, customer behavior, and other relevant metrics, along with reporting tools for business insights.

12. **Security and Compliance.** Implementing robust security measures to protect user data and transactions is crucial. The application must ensure compliance with relevant legal and regulatory standards is critical.

13. **Reviews and Ratings.** Users can leave reviews and ratings for products, providing valuable information to other shoppers.

14. **Wishlist and Favorites** users can create wishlists or mark products as favorites for future reference or as a way to keep track of items of interest.
15. **Recommendation Engine.** This function uses algorithms to suggest products to users based on their behavior, preferences, and purchase history, enhancing personalization.
16. **Returns and Refunds.** The application should be able to manage the process for customers who wish to return products, request refunds, or exchange items.
17. **Multi-Language and Currency Support.** To cater to a global audience, e-commerce applications may offer multiple language options and the ability to display prices in various currencies.
18. **Social Media Integration.** Integration with social media platforms allows users to share products and reviews, increasing brand visibility and traffic.
19. **Multi-platform Compatibility.** Ensuring the application is compatible across various devices and platforms (like desktops, mobile devices, and tablets) is necessary for broad accessibility.

Additionally, stakeholders use custom dashboards to track e-commerce application performance from multiple perspectives such as sales, marketing, customer service, manufacturing and project management [17]. Figure 6 shows a dashboard used by Eolas customers. Each indicator focuses on the financial goal to be achieved on a daily, weekly or monthly basis. Here are some examples of KPIs :

- Sales (in local currency) for the desired period (day, week, month, quarter, year).
- Gross profit (in local currency) is calculated from the total sales minus the cost of goods.
- Profit margin (in local currency) is calculated from the total sales minus the cost of the business.
- Number of orders with the evolution (in %) compared to last year.
- Number of new customers with the evolution (in %) compared to last year.
- Conversion rate (in %) tracks the rate of users purchasing items in the shopping cart.
- Bounce rate (in %) tells how many users exit your site after viewing only one page.
- Shopping cart abandonment rate (in %) tells how many users are adding products to their shopping catr but not checking out.
- Website traffic refers to the total number of visits to the site during a specific period.

4 SCALER Use Case Testbeds

This section delves into the technical setup and infrastructure that underpins the SCALER project's testbeds. These testbeds are pivotal components of the SCALER project, providing the necessary infrastructure for the development, deployment, and evaluation of the use cases discussed earlier in this document. These testbeds are designed to replicate real-world conditions as closely as possible, offering a controlled environment where the project's innovative solutions can be rigorously tested and refined.

Figure 6: Eolas custom business metrics dashboard

The testbeds are equipped with state-of-the-art hardware and software, enabling the SCALER team to simulate the complex interactions between various components of 5G-core network services and e-commerce web applications. This section will outline the technical setup of these testbeds, including the virtual private cloud (VPC) hosted by Eolas and the Grid'5000 testbed, detailing their respective roles in supporting the project's objectives.

4.1 Eolas Testbed

The testbed proposed by Eolas is designed to provide a secure and controlled environment for the SCALER project team to rigorously test the functionalities and performance of the use cases. It includes a description of the hardware specifications of the bare-metal servers, the virtualization software used, and the network configurations that support the deployment of multiple Kubernetes clusters and virtual machines. This section is essential for stakeholders to understand the technical capabilities and resources available for the SCALER project, which will facilitate the development, deployment, and scaling of the use cases.

Private Cloud As depicted in Figure 7, the Scaler testbed is hosted by Eolas as a virtual private cloud (VPC). The VPC consists of a VMware vSphere vCenter and two hypervisors running VMware vSphere ESXi, with sufficient compute resources to run multiple Kubernetes clusters and virtual machines (VM) for the project needs. For security purposes, all accesses to the testbed are secured by VPN.

Each bare-metal server includes the following hardware :

- 2 processors Intel Gold 6226 2.9Ghz 12 cores
- 192 Gigabytes of memory
- 40 Terrabytes of hard disk drive

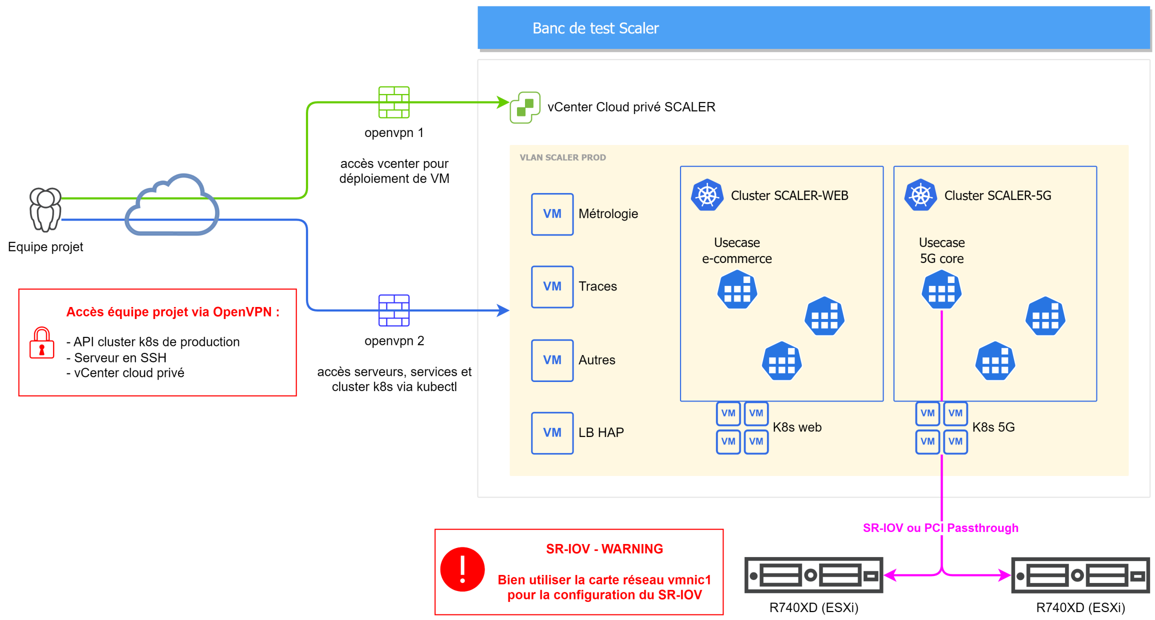

Figure 7: Eolas SCALER Testbed Architecture

- Less than 2 Terrabytes of solid-state drive
- Dual port network card with SR-IOV enabled (used by 5G use case)

Observability Tools Used in the Testbed Within the SCALER Use Case Testbed, several observability tools are employed to ensure effective monitoring, logging, and data storage. These tools are integral to maintaining the health and performance of the system. Here is the list of the used tools.

- **Minio (S3 storage)** Minio is a high-performance, Kubernetes-native object storage service that is compatible with Amazon S3 cloud storage service. It is generally used in private cloud environments due to its scalability and ease of deployment. Minio provides a secure way to store large amounts of unstructured data, such as photos, videos, log files, backups, and container/VM images. It is designed for high availability and supports features such as distributed mode, erasure coding, and bitrot detection to protect data integrity.
- **Grafana (Dashboard tool)** Grafana is an open-source analytics and interactive visualization web application that provides charts, graphs, and alerts for the web when connected to supported data sources. In the context of the SCALER project, Grafana will be used to create dashboards that visualize metrics, logs, and traces from the various components of the testbed. It helps the project team to understand the system's behavior and performance through real-time data visualization, making it easier to detect and diagnose potential issues.
- **Loki (Logs storage)** Loki is a horizontally-scalable, highly-available, multi-tenant log aggregation system inspired by Prometheus. It is designed to be cost-effective and simple to operate. Unlike other logging systems, Loki does not index the contents of the logs, but rather a set of labels for each log stream. This approach allows for more efficient querying, especially when paired with Grafana for visualizing the log data. Loki is particularly well-suited for storing and querying large volumes of log data.

- **Mimir (timeseries aggregation, storage and retention)** Mimir (formerly known as Cortex) is a long-term storage solution for Prometheus, which is a widely-used open-source system monitoring and alerting toolkit. Mimir allows the horizontal scaling of Prometheus across multiple nodes and provides long-term storage capability for Prometheus metrics. It is designed to be highly available and supports multi-tenancy, which means it can serve multiple separate Prometheus sources. Mimir is useful for aggregating, storing, and querying large amounts of time-series data generated by the SCALER testbed.

4.2 Grid'5000 Testbed

The partners of the SCALER project also have access to the experimental infrastructure Grid'5000 [18]. Grid'5000 is a French national research infrastructure for all areas in computer science with a particular focus on HPC, Cloud, BigData and AI. It spans over 9 sites and counts 69 clusters with 770 nodes (cf. Figure 8). The computational power is provided by 17016 CPU cores and 395 GPUs with 1876672 GPUs cores. It has 111.27 TiB of RAM and 6.0 TiB of PMEM. The storage is composed of 564 SSDs and 904 HDDs for a total of 1.48 PB. The maximum computational capacity is 725.8 TFLOPS (excluding GPUs).

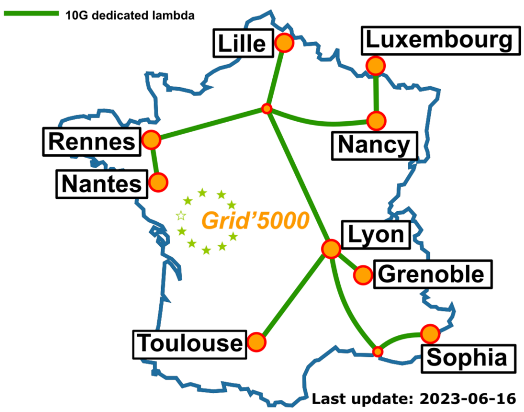

Figure 8: Grid'5000 Backbone Architecture

Grid'5000 comes with a set of useful tools allowing for the reservation and for the complete control over a set of resources. It is possible to deploy a full software stack (from OS to applications) on nodes, as well as to configure a virtual network.

Together, these tools form a comprehensive observability stack that enables the SCALER project team to store, monitor, and analyze the vast amounts of data generated by the use cases in the testbed environment. This ensures that the team can maintain high levels of service quality, reliability, and performance throughout the development and testing phases of the project.

5 Conclusion

This document serves as a cornerstone for future developments within the SCALER project. It gives insights about the framework that will guide the different SCALER partners through the implementation and optimization of 5G-core network and e-commerce applications.

References

[1] *Cloud Native Thinking for Telecommunications*. https://github.com/cncf/telecom-user-group/blob/master/whitepaper/cloud_native_thinking_for_telecommunications.md.

[2] S. Rommer, P. Hedman, M. Olsson, L. Frid, S. Sultana, and C. Mulligan. *5G Core Networks: Powering Digitalization*. Elsevier Science, 2019. ISBN: 9780081030097. URL: https://books.google.fr/books?id=d9eewgEACAAJ.

[3] *TeaStore*. https://github.com/DescartesResearch/TeaStore.

[4] Jóakim von Kistowski, Simon Eismann, Norbert Schmitt, André Bauer, Johannes Grohmann, and Samuel Kounev. "TeaStore: A Micro-Service Reference Application for Benchmarking, Modeling and Resource Management Research". In: *Proceedings of the 26th IEEE International Symposium on the Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*. MASCOTS '18. Milwaukee, WI, USA, Sept. 2018.

[5] *DeathStarBench website*. http://microservices.ece.cornell.edu/.

[6] *DeathStarBench Github*. https://github.com/delimitrou/DeathStarBench.

[7] *OpenTelemtry website*. https://opentelemetry.io/docs/demo/.

[8] *OpenTelemtry Github*. https://github.com/open-telemetry/opentelemetry-demo.

[9] *Locust*. https://locust.io/.

[10] *TrainTicket Github*. https://github.com/FudanSELab/train-ticket.

[11] Xiang Zhou, Xin Peng, Tao Xie, Jun Sun, Chenjie Xu, Chao Ji, and Wenyun Zhao. "Benchmarking microservice systems for software engineering research". In: *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018.* Ed. by Michel Chaudron, Ivica Crnkovic, Marsha Chechik, and Mark Harman. ACM, 2018, pp. 323–324. DOI: 10.1145/3183440.3194991. URL: https://doi.org/10.1145/3183440.3194991.

[12] *Jaeger: open source, end-to-end distributed tracing*. https://www.jaegertracing.io/.

[13] *Online Boutique*. https://github.com/GoogleCloudPlatform/microservices-demo.

[14] *Robot Shop*. https://github.com/instana/robot-shop.

[15] Betsy Beyer, Niall Richard Murphy, David K Rensin, Kent Kawahara, and Stephen Thorne. *The site reliability workbook: practical ways to implement SRE*. 2018.

[16] *Core Web Vitals*. https://web.dev/articles/vitals#core-web-vitals.

[17] *70 Ecommerce KPIs for Tracking Business Success*. https://www.shopify.com/blog/7365564-32-key-performance-indicators-kpis-for-ecommerce.

[18] *Grid'5000: a large-scale and flexible testbed for experiment-driven research in all areas of computer science.* https://www.grid5000.fr/w/Grid5000:Home.