|  |  |
| --- | --- |
| 3GPP TR 23.700-36 V18.1.0 (2022-12) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Study on Application Data Analytics Enablement Service;  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2022, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 5

Introduction 6

1 Scope 7

2 References 7

3 Definitions of terms, symbols and abbreviations 7

3.1 Terms 7

3.2 Symbols 8

3.3 Abbreviations 8

4 Key issues 8

4.1 Key issue #1: Support for application performance analytics 8

4.2 Key issue #2: Support for edge analytics enablement 8

4.3 Key issue #3: Support for data collection for application layer analytics 9

4.4 Key issue #4: Key Issue on interactions with SEAL services 9

4.5 Key issue #5: Support for slice-related application data analytics 10

4.6 Key issue #6: Support for slice configuration recommendation enablement 10

4.7 Key issue #7: Support for location accuracy analytics 10

4.8 Key issue #8: Support for service API capability analytics 11

5 Architecture aspects 12

5.1 Architecture related requirements 12

5.1.1 Description 12

5.1.2 General Requirements 12

5.1.3 ADAE internal architecture requirements 12

5.2 ADAE capability related requirements 12

5.2.1 Application performance analytics requirements 12

5.2.1.1 Description 12

5.2.1.2 Requirements 12

5.2.2 Edge load analytics requirements 12

5.2.2.1 Description 12

5.2.2.2 Requirements 13

5.2.3 Slice related application data analytics requirements 13

5.2.3.1 Description 13

5.2.3.2 Requirements 13

5.3 Functional Architecture 13

5.3.1 General 13

5.3.2 On-network Functional Architecture 13

5.3.3 Off-network Functional Architecture 15

5.3.4 ADAE internal architecture based on 3GPP data analytics framework 15

6 Solutions 17

6.1 Mapping of solutions to key issues 17

6.2 Solution #1: Support for application performance analytics 17

6.2.1 Solution description 17

6.2.1.1 Procedure on VAL server performance analytics 17

6.2.1.2 Procedure on VAL UE / session performance analytics 19

6.2.2 Corresponding Analytics API 21

6.2.3 Solution evaluation 22

6.3 Solution #2: Data Analytics Enablement 22

6.3.1 Solution description 22

6.3.1.1 General 22

6.3.1.2 Procedures 23

6.3.1.2.1 Generic server-side initiated data analytics 23

6.3.1.2.2 Generic client-side initiated data analytics 25

6.3.1.2.3 Generic data collection procedure 27

6.3.1.3 Information Flows 29

6.3.2 Evaluation 30

6.4 Solution #3: Support for edge load analytics 31

6.4.1 Solution description 31

6.4.2 Corresponding Analytics API 33

6.4.3 Solution evaluation 34

6.5 Solution #4: Support for performance analytics for UE-to-UE sessions 34

6.5.1 Solution description 34

6.5.2 Corresponding Analytics API 36

6.5.3 Solution evaluation 36

6.6 Solution #5: Service experience to support application performance analytics 36

6.6.1 Solution description 36

6.6.1.1 Pull service experience information 37

6.6.1.2 Push service experience information 38

6.6.1.3 Service experience information based on triggers 39

6.6.2 Solution evaluation 40

6.7 Solution #6: Support for slice related application data analytics 40

6.7.1 Solution description 40

6.7.2 Corresponding Analytics API 42

6.7.3 Solution evaluation 42

6.8 Solution #7: Slice configuration recommendation 42

6.8.1 Solution description 42

6.8.2 Corresponding Analytics API 44

6.8.3 Solution evaluation 44

6.9 Solution #8: support for location accuracy analytics 44

6.9.1 Solution description 44

6.9.2 Corresponding Analytics API 46

6.9.3 Solution evaluation 46

6.10 Solution #9: support for service API analytics 46

6.10.1 Solution description 46

6.10.2 Corresponding Analytics API 49

6.10.3 Solution evaluation 49

7 Deployment scenarios 49

7.1 General 49

7.2 Deployment model #1: Cloud-deployed ADAES 49

7.3 Deployment model #2 Edge-deployed ADAES 50

7.4 Deployment model #3: Hierarchical ADAES deployment 51

8 Overall evaluation 51

8.1 General 51

8.2 Solution evaluations 51

8.2.1 General 51

8.2.2 ADAE analytics services 53

9 Conclusions 54

9.1 General conclusions 54

9.2 General conclusions for normative work 54

9.2 Conclusions of solutions 55

Annex A (informative): Change history 56

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# Introduction

Data analytics is a useful tool for the operator to help optimizing the service offering by predicting events related to the network or slice or UE conditions. 3GPP introduced data analytics function (NWDAF) [2] to support network data analytics services in 5G Core network, and management data analytics service (MDAS) [3] to provide data analytics at the OAM.

Considering vertical-specific applications and edge applications as the major consumers of 3GPP-provided data analytics services, the application enablement layer can play role on the exposure of data analytics services from different 3GPP domains to the vertical/ASP in a unified manner; and on defining, at an overarching layer, value-add application data analytics services which cover stats/predictions for the end-to-end application service.

This technical report identifies the key issues and corresponding application architecture and related solutions with recommendations for the normative work.

# 1 Scope

The present document is a technical report which identifies the application enabling layer platform architecture, capabilities, and services to support data analytics enablement at the application layer.

The aspects of the study include the investigation of application data analytics services to optimize the application service operation, edge/cloud analytics enablement, data collection aspects per identified application data analytics service, coordination of data collection from multiple sources and unified exposure of data analytics to the vertical/ASP. This study will also identify potential enhancements to existing enablement layer entities (SEAL, eEDGEAPP, vertical specific enablers) to consume application data analytics enablement services.

The study takes into consideration the work done for data analytics in 3GPP TS 23.288 [2] and 3GPP TS 28.104 [3] and other related work outside 3GPP.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.288: "Architecture enhancements for 5G System (5GS) to support network data analytics services".

[3] 3GPP TS 28.104: "Management and orchestration; Management Data Analytics".

[4] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[5] 3GPP TS 23.434: "Service Enabler Architecture Layer for Verticals (SEAL); Functional architecture and information flows".

[6] 3GPP TS 23.558: "Architecture for enabling Edge Applications".

[7] 3GPP TS 26.531: "Data Collection and Reporting; General Description and Architecture"

[8] 3GPP TR 23.700-96: "Study on 5G-enabled fused location service capability exposure"

[9] 3GPP TS 23.222: "Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs"

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**example:** text used to clarify abstract rules by applying them literally.

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

<symbol> <Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

ADAE Application Data Analytics Enabler

A-ADRF Application layer - Analytical Data Repository Function

A-DCCF Application layer - Data Collection and Coordination Function

ADRF Analytical Data Repository Function

ASP Application Service Provider

DCCF Data Collection and Coordination Function

MEP Multi-access Edge Platform

MDAS Management Domain Analytics Service

NWDAF Network Data Analytics Function

OAM Operation, Administration and Maintenance

RNIS Radio Network Information Service

SEAL Service Enabler Architecture Layer

SEALDD Service Enabler Architecture Layer – Data Delivery

VAL Vertical Application Layer

# 4 Key issues

## 4.1 Key issue #1: Support for application performance analytics

Data analytics related to the application end-to-end QoS and in particular statistics and predictions on the application server or application session performance and load can be useful for the application specific layer, so as to proactively identify potential adaptations of the application service and to trigger adaptations at the communication layer. One example is the utilization of analytics by the application specific layer e.g. for selecting the least loaded EAS for an application session, or for selecting the optimal PLMN for communicating the application service in a given area.

This key issue will study:

- whether and how the application data analytics enablement service provides application QoS related analytics for the application session /service?

- whether and how the application data analytics enablement service provides application QoS related analytics tailored for different communication means (e.g. different PLMNs, RATs, slices)?

- what data needs to be collected from 3GPP system and application specific layer for performing application QoS related analytics?

- how to enable the exposure of application QoS related analytics to the vertical / ASP in a unified manner?

## 4.2 Key issue #2: Support for edge analytics enablement

Edge deployments are vitally important for applications that require performance levels that cannot be met by existing cloud deployments. Edge data analytics may relate to stats/predictions on computational resources and expected/predicted load of the platform which hosts the edge applications and may be necessary to be exposed as a service to EAS (which can be either edge native applications or edge enhanced applications at a centralized cloud). Particularly, for edge native applications which need to be light designed and high portable, the use of edge analytics at the edge platform can help improving the application service operation.

The support for edge analytics at the enablement layer (related to the edge performance, failure, service availability), would be useful for the edge applications to allow for dynamically deciding to scale-in, scale-out, migrate from the edge to the cloud in heavy load situations, or migrate from the cloud to the edge to improve the quality of experience for the end user. Also, the need for edge application relocation between edge platforms could be supported by using analytics which can be leveraged by the EDGEAPP layer and could be exposed as a service to the application developer for supporting the edge service operation optimization.

Hence, in this key issue the following points shall be studied:

- Whether and what edge data are needed to be collected by the application data analytics enablement server to allow for edge analytics enablement (related to the edge performance, failure, service availability)?

- Whether and how the application data analytics enablement server (deployed in an edge or centralized data network) can be utilized by the corresponding Edge Enabler layer architecture (as specified in 3GPP TS 23.558 [6]) to optimize edge services?

- Whether and how the analytics enablement layer needs to align with the EDGEAPP layer for allowing the edge services to utilize edge analytics enablement service to optimize their operation (e.g. triggering pro-active ACR based on edge analytics)?

- Whether and how the application data analytics enablement server needs to align with ETSI MEC system to utilize MEC services?

## 4.3 Key issue #3: Support for data collection for application layer analytics

For deriving application layer analytics, the data collection may be provided by different sources (e.g. vertical-specific server, application of the UE, EAS, 3rd party server, SEAL/SEALDD) and it needs to be identified how these data can be collected to allow for stats/predictions by the analytics enablement layer.

The application data analytics enablement layer needs to be capable of receiving data from different data producers and prepare the data to be used for deriving analytics. Such data can be measurements or analytics from the 5GS (5GC, OAM), the applications of the VAL UEs, other application enablers etc.

For example, for application QoS related analytics, such data can be potentially derived by the OAM, monitoring of network QoS by 5GC, subscribing and receiving QoS and network analytics from NWDAF, performance data from the application server, QoS data from enabler layer client-server sessions, etc. The consumer of the ADAE service may not be aware of the data that need to be collected from different sources, however the ADAE needs to be capable of selecting the optimal sources to collect data, subscribe to different data producers and also retrieve supplementary data samples based on the data producers' availability and load.

Hence, this key issue will discuss the following open issues:

- How to enable the collection and preparation of data at the application data analytics enablement service for data analytics derivation, when the data to be collected target the same performance metrics and are originated from different sources (UE, networking layer, application specific layer, non-3gpp domains)?

- Whether and how the application data analytics enablement layer needs to collect data from multiple sources, at the DN side or locally at the VAL UE side?

- Whether and how to leverage the UE data collection support provided by the SA4 EVEX study?

## 4.4 Key issue #4: Key Issue on interactions with SEAL services

SEAL is the service enabler architecture layer common to all vertical applications over 3GPP systems. It provides the functions like location management, group management, configuration management, identity management, key management, network resource management and network slice capability management as defined in 3GPP TS 23.434 [5].

This key issue will study:

- the applicability of the usage of SEAL services for application data analytics enablement services considering different deployment and business models

- whether any enhancements are required at the SEAL services for exposing data to the application data analytics enablement service?

- whether and how application data analytics at the application data analytics enablement service can be used to optimize SEAL service operation?

NOTE: This KI does not preclude the case where ADAE service is a SEAL service.

## 4.5 Key issue #5: Support for slice-related application data analytics

Data analytics related to slicing are provided by the 5GS, from NWDAF (e.g. slice load analytics) and MDAS (e.g. NSI/NSSI performance analytics). The slice capability enablement layer (based on NSCALE) discusses enhancements to NSCE SEAL service (as specified in TS 23.434 [5]). According to Solution #5 of TR 23.700-99, the NSCE server is expected to consume 5GS services related to analytics (from MDAS, NWDAF) and to re-expose them to the VAL server (slice customer).

If further analytics is required on top of the consumed analytics services (MDAS/NWDAF), the ADAES can be utilized by NSCE service to perform further analytics related to applications for certain slice / NSI. Such analytics service is not overlapping with NWDAF/MDAS services since it will provide at application layer data analytics (per session or VAL server) which are bound to a given slice or NSI (e.g. per VAL session performance statistics when using slice #1).

This key issue will investigate:

- what is the possible interaction between NSCE service and ADAES, for providing application layer analytics bound to a slice or an NSI?

- whether and what data need to be collected by NSCE layer for supporting per slice or NSI app layer analytics?

## 4.6 Key issue #6: Support for slice configuration recommendation enablement

Slice data analysis can analyze the slice usage pattern based on the collected network slice performance and analytics, and provide analysis-based slice management suggestions, such as the slice scale in and scale out, which can be exposed to VAL or provided to NSCE as a service. One example is, to support the application layer automatic network slice lifecycle management, in which the NSCE server is supposed to send out some management recommendation based on the collected network slice performance analytics from the 5GC, OAM and the application layer. The recommendation is usually an empirical value given by experienced network operations, ADAES can help to output the recommendation according to the analysis based on historical network slice status and network performance.

Hence, in this key issue the following points shall be studied:

- How ADEAS supports the slice configuration recommendation based on the slice related information from NSCE.

## 4.7 Key issue #7: Support for location accuracy analytics

According to SA1 TS 22.261 (6.27) and TS 22.104, positioning services aim to support verticals and applications with positioning accuracies better than 10 meters, thus more accurate than the ones of TS 22.071 for LCS. High accuracy positioning is characterized by ambitious system requirements for positioning accuracy in many verticals and applications, including regulatory needs. For example, on the factory floor, it is important to locate assets and moving objects such as forklifts, or parts to be assembled. Similar needs exist in transportation and logistics, for example rail, road and use of UAVs. In some road user cases, UE's supporting V2X application(s) are also applicable to such needs. In cases such as guided vehicles (e.g. industry, UAVs) and positioning of objects involved in safety-related functions, availability needs to be very high. In SA1, different service levels are mapped to different positioning performance attributes including vertical and horizontal accuracies. Such accuracies (e.g. cm-level, dm-level, meter-level) may depend on the positioning methods which are used, the LCS producers, as well as the UE mobility and the environment.

When the VAL consumer requests a positioning service, the accuracy is calculated at the entity which produces a location estimate and whether the accuracy can be maintained along an application session (for a given time/area) is challenging to answer at the time of the request/subscription. In this scenario, there needs to be a translation of the per UE location report accuracy to an expected /predictive location accuracy derivation for the application requiring positioning services. Such location accuracy analytics and in particular the sustainability of vertical and horizontal accuracy per VAL application (e.g. group of field devices in industrial use cases) based on per UE reported location accuracies could be needed to make sure that LMS will meet the VAL customer location reporting requirements for a given time/area of location request validity. Such information will help deciding from application side whether for a particular service (e.g. process automation, AR in factories) adaptation of the application behavior if the accuracy cannot be maintained e.g. programs the IIOT devices to maintain a bigger distance etc.

This key issue aims to investigate:

- whether and how ADAES needs to be enhanced to perform analytics on vertical and horizontal accuracy for positioning services requested by a VAL customer?

- what criteria need to be considered (e.g. environment, UE mobility, service type, positioning method, fusion) and what data are needed to be collected from 5GS (e.g. NWDAF, LMF) and VAL side for performing location accuracy analytics for the VAL application?

- what enhancements are needed in SEAL LMS to support location accuracy analytics/data per VAL application?

## 4.8 Key issue #8: Support for service API capability analytics

The service APIs (assuming also EAS provided APIs, enablement service APIs and OAM API), cannot be assumed uniformly available and offering the same service level across the entire network. For CIoT service, 3GPP SA2 has already defined a NEF monitoring service to allow the AF to monitor the API availability and service level (e.g. via invoking a Nnef\_APISupportCapability API as part of the Monitoring Event in TS 23.522) for the target API. However, this doesn't provide analytics on NEF/SCEF APIs and doesn't support all ranges of service APIs (produced or offered at the platform) and focuses on the CIoT scenarios. Furthermore, CAPIF supports the monitoring of service API invocations and can provide API monitoring via the *Availability of service APIs event notification* or *Service Discover Response* as specified in TS 23.222.

Service API analytics (such as the statistics on the successful/failed API invocation or predicted API availability for a given deployment) can be a tool to be used by the API provider (ASP, ECSP, MNO) to help optimizing the API usage by enabling him to trigger API related actions like API mashups, API rate limitations/throttling events, or pro-actively detecting API termination point changes which may affect service performance. Such service could be also useful for the API invoker to allow for early notification on expected API unavailabilities.

One example for such API analytics can be the statistics or prediction of NEF API or SEAL API invocation request failure probability, or the predicted number of API invocations for a particular EDN area and time of day or even the number of unauthorized API invocation requests. Such analytics can be matched to different APIs and API operations and can be used as a service for example to help the service API invoker to identify what is the best time and means to perform a request e.g. so as to avoid possible failure due to high number of invocations expected for this service API.

This key issue will investigate:

- whether and how the application data analytics enablement service needs to provide data analytics for service APIs?

- what data / API logs and from which entities need to be collected for performing service API analytics?

- what enhancements are needed in CAPIF (CCF, API management function) for supporting service API analytics?

NOTE: Data/analytics related to MNO provided APIs are only possible if ADAES is deployed by the MNO. For ADAES outside MNO domain, this key issue covers only non-MNO provided APIs.

# 5 Architecture aspects

## 5.1 Architecture related requirements

### 5.1.1 Description

This subclause specifies the general and ADAE internal requirements for application data analytics enablement layer functional architecture.

### 5.1.2 General Requirements

[AR-5.1.2-a] The ADAE client and the ADAE server shall support one or more VAL applications.

[AR-5.1.2-b] Supported ADAE capabilities may be offered as APIs to the VAL applications.

[AR-5.1.2-c] The ADAE shall support interaction with 3GPP network system to consume network data analytics services.

[AR-5.1.2-d] The ADAE client shall be capable to communicate with one or more ADAE servers of the same ADAE service provider.

### 5.1.3 ADAE internal architecture requirements

[AR-5.1.3-a] The ADAE layer shall be able to provide a data collection coordination functionality to enable the collection from diverse data sources (OAM, 5GC, UE) per application data analytics event type.

[AR-5.1.3-b] The ADAE layer shall include a data analytics repository function to store application data analytics.

[AR-5.1.3-c] The data collection coordination and repository capabilities may be offered as APIs to ADAE server.

## 5.2 ADAE capability related requirements

### 5.2.1 Application performance analytics requirements

#### 5.2.1.1 Description

This subclause specifies the requirements for application performance analytics capability.

#### 5.2.1.2 Requirements

[AR-5.2.1.2-a] The ADAE server shall be capable of providing data analytics for the VAL server performance.

[AR-5.2.1.2-b] The ADAE server shall be capable of providing data analytics for the VAL application sessions (for both Uu-based and PC5-based sessions).

[AR-5.2.1.2-c] The ADAE server shall be able to collect application performance measurements and analytics from one or more ADAE clients.

### 5.2.2 Edge load analytics requirements

#### 5.2.2.1 Description

This subclause specifies the requirements for edge load analytics capability.

#### 5.2.2.2 Requirements

[AR-5.2.2.2-a] The ADAE server shall be capable of collecting edge data from one or more edge platforms

[AR-5.2.2.2-b] The ADAE server shall enable the exposure of edge data analytics to the VAL applications

### 5.2.3 Slice related application data analytics requirements

#### 5.2.3.1 Description

This subclause specifies the requirements for slice related application data analytics capability.

#### 5.2.3.2 Requirements

[AR-5.2.3.2-a] The ADAE server shall be capable of providing data analytics for the VAL server or VAL session performance for a requested slice or slice instance.

[AR-5.2.3.2-b] The ADAE server shall be able to collect slice related measurements and analytics from one or more 3GPP network system domains (OAM, 5GC).

## 5.3 Functional Architecture

### 5.3.1 General

This clause provides the overall functional architecture description, which includes the on-network and off-network functional models.

### 5.3.2 On-network Functional Architecture

For the on-network functional architecture, both service-based representation and reference point representation are provided.

Figure 5.3.2-1 depicts the application data analytics enablement architecture in the non-roaming case, using the reference point representation showing how various entities interact with each other.



Figure 5.3.2-1: Architecture for application data analytics enablement – reference points representation

The application data analytics enablement client communicates with the application data analytics enablement server over the ADAE-UU reference point. The application data analytics enablement client provides the support for application data analytics enablement functions to the VAL client(s) over ADAE‑C reference point. The VAL server(s) communicates with the application data analytics enablement server over the ADAE-S reference point. The application data analytics enablement server, acting as AF, may communicate with the 5G Core Network functions (over N33 reference point to NEF and N6 reference point to UPF) and OAM (over ADAE-OAM interface).

Figure 5.3.2-2 exhibits the service-based interfaces for providing and consuming application data analytics enablement services. The application data analytics enablement server could provide service to VAL server and ADAE client through interface SAdaee.



Figure 5.3.2-2: Architecture for application data analytics enablement – Service based representation

Figure 5.3.2-3 illustrates the service-based representation for utilization of the 5GS network services based on the 5GS SBA specified in 3GPP TS 23.501 [4].



Figure 5.3.2-3: Architecture for application data analytics enablement utilizing the 5GS network services based on the 5GS SBA – Service based representation

The application data analytics enablement server can be deployed as a SEAL server; hence enhancements to SEAL architecture (as specified in TS 23.434 [5]) are needed to incorporate the ADAE service. Figure 5.3.2-4 illustrates the service-based representation including ADAE server as part of the SEAL framework.



Figure 5.3.2-4: SEAL functional model representation using service-based interfaces and including ADAE function

### 5.3.3 Off-network Functional Architecture

Figure 5.3.3-1 illustrates the generic off-network functional model for ADAE.



Figure 5.3.3-1: Generic off-network functional model

In the vertical application layer, the VAL client of UE1 communicates with VAL client of UE2 over VAL-PC5 reference point. An application data analytics enablement client of UE1 interacts with the corresponding application data analytics enablement client of UE2 over ADAE-PC5 reference points. The UE1, if connected to the network via Uu reference point, can also act as a UE-to-network relay, to enable UE2 to access the VAL server(s) over the VAL-UU reference point.

If the ADAE server is deployed as SEAL server, the off network functional architecture is similar to SEAL off-network architecture (as specified in TS 23.434 [5]).

### 5.3.4 ADAE internal architecture based on 3GPP data analytics framework

In 3GPP SA2 (TS 23.288 [2]), the following entities have been defined as part of the data analytics framework:

- NWDAF provides network data analytics services at the 5GC

- DCCF coordinates the collection and distribution of data requested by NF/AF consumers. Data Collection Coordination is supported by a DCCF. Data Consumers can send requests for data to the DCCF rather than directly to the NF/AF Data Source.

- ADRFstores historical data and/or analytics, i.e., data and/or analytics related to past time period that has been obtained by the consumer. After the consumer obtains data and/or analytics, consumer may store historical data and/or analytics in an ADRF. Whether the consumer directly contacts the ADRF or goes via the DCCF or via the Messaging Framework is based on configuration.

ADAE server can reuse the existing 3GPP data analytics framework for the data collection coordination, delivery and storage as provided by DCCF and ADRF functionalities. As illustrated in Figure 5.3.4-1, A-DCCF and A-ADRF can be defined as functionalities within the internal ADAE architecture and can offer similar functionality as proposed in 5GC but at application layer.

Figure 5.3.4-1 illustrates the generic functional model for ADAE when re-using the existing data analytics model.



Figure 5.3.4-1: Generic functional model based on network data analytics model

In this model, an Application layer - Data Collection and Coordination Function (A-DCCF) is used to fetch data or put data into an Application level entity (e.g. A-ADRF, Data Source). Such A-DCCF coordinates the collection and distribution of data requested by ADAE server (over ADCCF-1, ADAE-X). ADAE server can also directly interact with the Data Sources via ADAE-Y.

Also, Application layer – Analytics and Data Repository Function (A-ADRF) can be used to store historical data and/or analytics, i.e. data and/or analytics related to past time period that has been obtained by the ADAE server (via AADRF-1) or other NFs/NWDAF. ADAE server can also fetch historical data from ADRF. Whether the ADAE server directly contacts the ADRF or goes via the A-DCCF is based on configuration.

Data Sources can be 5GS data sources (5GC, OAM) or enablement layer data sources (SEAL, EEL) or external data sources at the DN side (VAL server/EAS) and VAL UEs. A-DCCF and A-ADRF can be used only for interacting with certain data sources (e.g. 5GC, OAM) based on configuration, and can be hidden from the VAL layer.

NOTE: If the Data Source is the VAL UE, then the data collection mechanism shall reuse the SA4 mechanism based on EVEX study (TS 26.531 [7]).

# 6 Solutions

## 6.1 Mapping of solutions to key issues

Table 7.1-1: Mapping of solutions to key issues

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | KI #1 | KI #2 | KI #3 | KI #4 | KI #5 | KI #6 | KI #7 | KI #8 |
| Sol #1 | **X** |  |  |  |  |  |  |  |
| Sol #2 | **X** | **X** | **X** | **X** |  |  |  |  |
| Sol #3 |  | **X** |  |  |  |  |  |  |
| Sol #4 | **X** |  |  |  |  |  |  |  |
| Sol #5 | **X** |  |  |  |  |  |  |  |
| Sol #6 |  |  |  | **X** | **X** |  |  |  |
| Sol #7 |  |  |  | **X** |  | **X** |  |  |
| Sol #8 |  |  |  | **X** |  |  | **X** |  |
| Sol #9 |  |  |  |  |  |  |  | **X** |

## 6.2 Solution #1: Support for application performance analytics

### 6.2.1 Solution description

This solution addresses Key Issue #1.

This solution introduces application layer analytics to provide insight on the operation and performance of an application (VAL server or EAS, application session) and in particular statistics or prediction on parameters related to e.g. VAL server number of connections for a given time and area, VAL server rate of connection requests, connection probability failure rates, RTT and deviations for a VAL server or VAL UE session, packet loss rates etc.

In this solution, two procedures are described in more detail:

- one procedure for VAL server related analytics where an example in provided for VAL server performance,

- one procedure for VAL session/UE related analytics.

#### 6.2.1.1 Procedure on VAL server performance analytics

Figure 6.2.1.1-1 illustrates the procedure where the VAL server performance analytics are performed based on data collected from the ongoing VAL sessions as well as data from the DN (VAL server, DN database or networking stack at DN).

Pre-conditions:

1. ADAEC is connected to ADAES



Figure 6.2.1.1-1: ADAES support for VAL server performance analytics

1. The consumer of the ADAES analytics service sends a subscription request to ADAES and provides the analytics event ID e.g. "VAL perf prediction", the target VAL server ID, the time validity and area of the request, the required confidence level, whether offline and/or online analytics are needed etc.

2. The ADAES sends a subscription response as an ACK to the consumer.

3. The ADAES maps the analytics event ID to a list of data collection event identifiers, and optionally a list of data producer IDs. Such mapping may be preconfigured by OAM or may be configured at ADAES based on the analytics event type / vertical type.

4. The ADAES sends a subscription request to the Data Producers (at the DN side or UE side) with the respective Data Collection Event ID and the requirement for data collection. This message includes the Data Collection event ID and/or the analytics event ID, the target VAL server ID, the ADAES ID, the time validity and area of interest, the required confidence level etc.

5. The Data Producer(s) sends a subscription response as an ACK to the ADAES.

NOTE: The ADAES acting as AF may also subscribe to NEF/SMF/PCF/NWDAF to monitor network/UE situation or network data analytics required for the application data analytics event.

6. The ADAES based on subscription, may receive offline stats/data on the VAL server performance based on the analytics/data collection event ID. Such offline data can be average/peak throughput, average/maximum e2e delay, jitter, av. PER, availability, VAL server load, number of failed transactions, and can be for a given area and time of the day (based on the time/area of the request). Th edge / cloud DB depending on the deployment can be also part of ADRF (for MNO-deployed ADAES).

A session starts between the VAL server #1 and a UE (this could happen for more than one UEs)

7. The Data Producer starts collecting/listening to real-time networking or application data (from networking start at DN or VAL server itself). Such data can be the RTT, the PER, throughput etc.

8a. The Data Producer sends the real-time data to the ADAES, where the data correspond to the data collection ID or the analytics event ID for which the ADAES subscribed.

8b. The ADAES may receive also data (periodically or if a threshold is reached based on configuration) from the application of the UE within the ongoing session (via ADAEC). Such data can be about the RTT, average/peak throughput, jitter, QoE measurements (MOS, stalling events, stalling ratios, etc), QoS profile load, VAL server load, etc.

9. When the VAL UE session with VAL server finishes, the ADAEC notifies the ADAEC of the completion of the reporting.

10. The ADAES abstracts or correlates the data based on the analytics event and the data collection configuration. Such correlation can be filtering of data for the same metrics but with different granularities or be combining/aggregating the data of segments of the end-to-end path (end to end is between VAL client and server). The outcome is an abstracted/correlated/filtered set of data.

11. The ADAES derives application layer analytics on VAL server #1 performance, based on the analytics ID and type of request. Such analytics can be stats or prediction for a given area/time and based on the event type for a given network configuration.

12**.** The ADAES sends the analytics to the consumer, where these analytics include the VAL server 1 predicted or statistic performance for a given area and time horizon, including also the confidence level, whether offline/online analytics were used.

#### 6.2.1.2 Procedure on VAL UE / session performance analytics

Figure 6.2.1.2-1 illustrates the procedure where the VAL session performance analytics are performed based on data collected from the ongoing VAL sessions.

Pre-conditions:

1. ADAEC is connected to ADAES



Figure 6.2.1.2-1: ADAES support for VAL session performance analytics

1. The consumer of the ADAES analytics service sends a subscription request to ADAES and provides the analytics event ID e.g. "VAL UE perf prediction", the target VAL UE ID, VAL server ID/VAL application ID, the time validity and area of the request, the required confidence level, exposure level for providing UE analytics. If the consumer is the VAL server, the VAL server can provide to ADAEC application data related to the UE expected route/trajectory and VAL application traffic schedule / expected session time.

2. The ADAES sends a subscription response as an ACK to the consumer.

3. The ADAES selects the corresponding ADAEC of the VAL UE for which the local analytics need to be performed.

4a. The ADAES sends a subscription request to the ADAEC with the analytics event ID and the configuration of the reporting required (e.g. periodic, based on threshold or event)

4b. The ADAEC sends a subscription response to ADAES

5. The ADAEC maps the analytics event ID to a list of data collection event identifiers or data collected IDs at the VAL UE or other UEs within the service and in proximity (in group-based communications)

6. The ADAEC subscribes to the VAL clients and/or requests UE local data based on the respective Data Collection Event ID (or the analytics event ID if they already know the mapping). This data may come from the PDU layer of the UE (via listening the traffic), or via VAL client of one or more UEs (if an application consists of a group of UEs).

A session starts between the VAL UE #1 and a VAL server.

7. The ADAEC (after being aware from the VAL client that the session started) sends a notification to ADAES that a session started, and it could be possible to provide real-time data analytics for VAL UE performance in the target area.

8. The ADAEC starts collecting data from the corresponding VAL UE(s) based on subscription. Such data can be about the RTT, throughput, jitter, QoE measurements, QoS profile load, etc. It can be also possible that VAL client provides to ADAEC application data related to the UE expected route/trajectory and VAL application traffic schedule / expected session time.

9. The ADAEC filters or correlates the data based on the analytics event and the data collection configuration.

10. When the VAL UE session finishes, the ADAEC (optionally) derives VAL session analytics to ADAES on VAL UE #1 performance, based on the analytics ID and type of request. Such analytics (if performed at the ADAEC can be stats or predictions on the RTT or RTT deviation, average/peak throughput, av. jitter, QoE measurements (MOS, stalling events, buffer related events), QoS profile load, VAL application traffic load etc. In case of prediction, a confidence level shall be also present and a time horizon for the predicted parameters.

11. The ADAEC sends the data of step 8 or the analytics of step 9 (if ADAEC performs analytics) to the ADAES.

12. The ADAES derives application layer analytics on VAL session performance (based on the data or analytics received by the ADAEC), based on the analytics ID and type of request. Such analytics can be stats or prediction for a given area/time and based on the event type for a given network configuration. Such analytics (if no analytics is performed at ADAEC) at ADAES can be stats or predictions on the RTT or RTT deviation, average/peak throughput, av. jitter, QoE measurements, QoS profile load, VAL application traffic load etc. In case of prediction, a confidence level shall be also present and a time horizon for the predicted parameters.

13**.** The ADAES sends the analytics to the consumer, where these analytics include the VAL UE 1 session predicted performance for a given area and time horizon, including also the confidence level, whether offline/online analytics were used.

### 6.2.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #1.

For VAL server performance analytics, this includes:

- Inputs: per VAL server performance measurements (application QoS measurements such as latency, channel losses, data rate, jitter), historical data/stats for VAL server performance, network/KPI monitoring from 5GS

- List of Data Sources

- Data Source #1 information: VAL UE #1 (or more VAL UEs having a session with VAL server #1), VAL Server #1

- Data required from Data Source #1: application QoS measurements

- Data Source #2 information: OAM

- Data required from Data Source #1: PM data

- Data Source #3 information: 5GC

- Data required from Data Source #3: service experience analytics, network and QoS monitoring

- Output: Predicted application QoS metrics per VAL server, Statistics on VAL server application QoS/performance metrics

For VAL session performance analytics, this includes:

- Inputs: per VAL session performance measurements (application QoS measurements such as latency, channel losses, data rate, jitter)

- List of Data Sources:

- Data Source information: VAL UE #1, VAL Server #1

- Data required from Data Source: application QoS measurements

- Output: Predicted application QoS metrics per VAL session

### 6.2.3 Solution evaluation

This solution addresses Key Issue #1 and introduces application layer analytics to predict the performance of a VAL server or an application session between a VAL UE and a VAL server. Such solution enables the VAL layer to get statistics or predictions for expected deviations of application performance metrics (e.g. RTT) based on data collected from the ADAE clients, as well as from 5GS. This solution is complementary to Solution #4 which covers only the VAL UE-to-UE sessions. Also, this Solution doesn't overlap with Solution #2 which provides a generic mechanism for data analytics enablement (and could be re-used).

This solution is technically viable and doesn't introduce any impact on 5GS.

## 6.3 Solution #2: Data Analytics Enablement

### 6.3.1 Solution description

#### 6.3.1.1 General

The following solution corresponds to the key issue #1, 2, 3, 4 on support for application performance analytics, data collection, edge analytics enablement, and interaction with SEAL service.

ADAE supporting application and service enablement layer analytics collects application or service enablement layer related information from the vertical application layer, service enablement layer, 5GC, and/or OAM to perform data analytics and to provide statistics or predictions.

The solution addresses the following aspects:

- Enablement of data analytics services by the ADAES layer, by addressing the types of interactions possible between ADAES and other service layer entities.

- Description of how higher-level data analytic requests can be provided to the ADAES layer, and on how analytics requests are translated into generic data collection tasks.

- Description of the information flows for generic data analytics and data collection.

A data source profile is proposed for a data source which describes capabilities related to data generation, data availability and pre-processing, as shown in table 6.3.1.3-1.

A data analytics request from a requesting server/client to ADAE provides criteria for the data source profile elements to indicate the applicable data sources. The data analytics request also includes criteria for determining the information applicable for collection via the data collection procedure. The data analytics request also includes task processing requirements and operations used to determine the necessary level of processing by the ADAE, as shown in table 6.3.1.3-2

A data collection request from ADAE to a source provides information about the original data source and the data required, as shown in table 6.3.1.3-4.

NOTE 1: Data collection tasks may be performed by ADAE using available dedicated functionality, e.g. Application Data Collection and Coordination Function deployed by MNO at the service layer.

NOTE 2: Further definition of the data processing operations will be provided in the normative phase, including stage 3.

Interaction between ADAE and the vertical application layer can be done over ADAE-S or ADAE-C reference points.

If ADAE is implemented in the SEAL layer, then:

- interaction between ADAE server and application enabler servers can be done over SEAL-S (e.g. for EES see TS 23.558 Figure A.4.1-1, for VAE Server see TS 23.286 Figure 6.2-2). Interactions between ADAE client and the corresponding application enabler client can be done over SEAL-C.

- interaction between ADAE server and SEAL server could be done over SEAL-X (TS 23.434 Figure 6.2-3),

#### 6.3.1.2 Procedures

##### 6.3.1.2.1 Generic server-side initiated data analytics

Pre-conditions:

1. ADAE is provisioned with data source profiles (Table 6.3.1.3-1) for data sources in the vertical application layer, application/service enablement layer (e.g., SEAL server/client, EES/EEC, CAPIF entities), core network (e.g., OAM, DCCF, NWDAF), etc. Alternatively, ADAE may perform a discovery for the data source profiles of data sources of interest.



Figure 6.3.1.2.1-1: Server-side initiated generic data analytics request

1. The requesting server sends a data analytics request to the serving ADAE server to initiate data analytics, using either a one-time request or a subscription request. The request may specify the type of data analytics and the requirements/ preferences of the required analytics as defined in Table 6.3.1.3-2.

2. If the request is authorized, the ADAE server may determine which input data should be collected and the input data sources based on the request and local policies.

3. Based on the determination in step 2, the ADAE server may collect input data from various sources by performing data collection procedure, as described in clause 6.3.1.2.3.

NOTE 1: A data repository may be assumed to be available to the ADAE server for corresponding data storage tasks, as described in clause 6.3.1.2.3. The repository functionality may be provided e.g. by a SEALDD storage server, or an Application – Analytics and Data Repository Function deployed by MNO at the service enablement layer.

Depending on the type of data source, this step can be performed with several alternatives as follows:

(a) from other servers:

3a. If the requested analytics requires server-side input data, the ADAE server may collect input data from the server-side entities such as vertical application servers (and EAS, if the target application is an edge application) via ADAE-S reference point.

NOTE 2: In this alternative, the ADAE server may also collect input data from SEAL server(s)and application/service enabler server(s) (e.g. EES, VAE-S).

(b) from ADAE clients:

3b1. If the requested analytics requires client-side input data, the ADAE server performs the data collection procedure with the corresponding ADAE client as source. The request specifies what input data is required from the client-side.

NOTE 3: In this alternative, the ADAE client may also collect input data from SEAL client and application/service enabler clients(s) (e.g. EEC, VAE-C) , or other ADAE clients (via ADAE-PC5).

(c) from other functions external to the service enablement layer (e.g. NWDAF, OAM):

3c. The ADAE server performs the data collection procedure with other analytics functions in the system, such as NWDAF or OAM, as sources. For example, the ADAE server may collect input data from 5GC via monitoring events or via interactions with NWDAF, receive input data from OAM system, receive analytics result from NWDAF by subscription, etc.

4. If the step 1 request requires data collection without higher-level analytics, ADAE sends a data analytics response (collected data) to the requester with the collected data or the location where the collected data is stored, as defined in Table 6.3.1.3-5.

5. If the step 1 request includes analytics tasks to be performed, and based on the collected input data, the ADAE server derives the analytics result. The ADAE server performs the analytics operations provided in the data analytics request in step 1. The result of the analytics task on the collected data can be optionally stored in the repositories available, such as a SEALDD storage server, Application-ADRF, etc.

6. Following an analytics task, the ADAE server may perform or trigger other analytics actions (using requests to the corresponding entities) based on the analytics result, before providing a final response to the requestor.

7. The ADAE server provides the requested data analytics task outputs to the requestor, using either a response or a notification message, depending on the service used in step 1.

##### 6.3.1.2.2 Generic client-side initiated data analytics

Pre-conditions:

1. ADAE is provisioned with data source profiles (Table 6.3.1.3-1) for data sources in the vertical application layer, service enablement layer (e.g., SEAL server/client, EES/EEC, CAPIF entities), core network (e.g., OAM, DCCF, NWDAF), etc. Alternatively, ADAE may perform a discovery for the data source profiles of data sources of interest.



Figure 6.3.1.2.2-1: Client-side initiated generic data analytics procedure

1. The requesting client sends a data analytics request to the serving ADAE client to initiate data analytics, using either a one-time request or a subscription request. The request may specify the type of data analytics and the requirements/ preferences of the required analytics, as defined in Table 6.3.1.3-2.

2. If the request is authorized, the ADAE client determines which input data should be collected and where to collect the input data based on the request.

3. Based on the determination in step 2, if client-side input data is required, the ADAE client may collect input data from the client-side entities such as vertical application clients (via ADAE-C) and other ADAE clients (via ADAE-PC5) by performing data collection procedure, as described in clause 6.3.1.2.3.

NOTE 1: In this step, the ADAE client may also collect input data from SEAL client and application/service enabler clients(s) (e.g. EEC, VAE-C).

4. The ADAE client sends the data analytics request to the ADAE server, as defined in Table 6.3.1.3-2. If client-side input data has been collected by the ADAE client, the collected client-side input data or the location of the data will also be sent to the ADAE server.

5. The ADAE server may determine which input data should be collected and the input data sources, based on the request from the ADAE client and/or local policies

6. Based on the request in step 4 and/or the determination in step 5, the ADAE server may collect input data from various sources by performing data collection procedure, as described in clause 6.3.1.2.3.

NOTE 2: A data repository may be assumed to be available to the ADAE server for corresponding data storage tasks, as described in clause 6.3.1.2.3. The repository functionality may be provided e.g. by a SEALDD storage server, or an Application – Analytics and Data Repository Function deployed by MNO at the service enablement layer.

Depending on the type of data source, this step can be performed with several alternatives as follows:

(a) from other servers:

6a. The ADAE server collects input data from server-side entities such as vertical application servers (and EAS, if the target application is an edge application) via ADAE-S reference point.

NOTE 3: In this alternative, the ADAE server may also collect input data from SEAL server(s) and application/service enabler server(s) (e.g. EES, VAE-S).

(b) from other functions external to the service enablement layer (e.g. NWDAF, OAM):

6b. The ADAE server may collect input data and/or request for analytics service from other analytics functions in the system.

7. If the step 1 request requires data collection without higher-level analytics, ADAE server sends a data analytics response (collected data) to the requester with the collected data or the location where the collected data is stored, as defined in Table 6.3.1.3-3.

8. If the step 1 request includes analytics tasks to be performed, and based on the collected input data, the ADAE server derives the analytics result. The ADAE server performs the analytics operations provided in the data analytics request in step 4. The result of the analytics task on the collected data can be optionally stored in the repositories available, such as a SEALDD storage server, Application-ADRF, etc.

9. Following an analytics task, the ADAE server sends a data analytics response (or a notification) to the ADAE client with the analytics result or the location where the resulting data is stored, as defined in Table 6.3.1.3-3.

10. Following an analytics task, the ADAE server or client may perform or trigger other analytics actions (using requests to the corresponding entities) based on the result, before providing a final response to the requestor.

11. The ADAE client provides the requested data analytics task outputs to the requestor, using either a response or a notification message, depending on the service used in step 1.

##### 6.3.1.2.3 Generic data collection procedure

Figure 6.3.1.2.3-1 describes a generic data collection procedure, i.e,. using abstract data sources.

Precondition: ADAE is provisioned with data source profiles (Table 6.3.1.3-1) for data sources in the vertical application layer, service enablement layer (e.g., SEAL server/client, EES/EEC, CAPIF entities), core network (e.g., OAM, DCCF, NWDAF), etc. Alternatively, ADAE may perform a discovery for the data source profiles of data sources of interest.



Figure 6.3.1.2.3-1: Generic data collection procedure

1. ADAE determines data collection sources and processing operations based on the requirements in the data analytics request. For example, ADAE may determine whether data should be collected from the application layer, the service enablement layer, the core network, or whether a data processing task should be performed using data from multiple layers/sources.

2. ADAE may collect existing data that can meet or partially meet the requirements of the data analytics request from sources with the "Data source role" IE set as "repository" in the data source profile (e.g., SEALDD storage server, Application-ADRF). The request and response for data collection are defined in 6.3.1.3-4 and 6.3.1.3-5.

NOTE 1: ADAE data collection requests/responses may be realized via subscriptions/notifications.

3. ADAE collects data from other identified data sources. The request and response for data collection from a data source are defined in 6.3.1.3-4 and 6.3.1.3-5.

4. ADAE performs data processing operations as determined in step 1 and/or required by policies. For example, data samples that target the same performance metrics but originate from different sources may be normalized and validated. Such processing may remove samples that are inconsistent across different sources and keep samples that achieve consensus across all sources.

5. The collected (and optionally processed) data can be optionally stored in available repositories, such as a SEALDD storage server, Application-ADRF, etc.

#### 6.3.1.3 Information Flows

The data source profile includes information about the data generation/production capability of the data source to support data collection for data analytics service and the availability/accessibility of the generated/produced data, as defined in Table 6.3.1.3-1.

Table 6.3.1.3-1: Data source profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Information element | | Status | | Description | |
| Source ID | | M | | ID of the source | |
| Data source entity | | M | | Specifies the type of the entity, such as a vertical application server, a SEAL server/client, EES/EEC, EAS, etc.(NOTE 1) | |
| Information type | | M | | Type of information can be provided by the data source, e.g., performance indicators, resource usage data, server load data, etc. The information types may also include those obtained from NWDAF or OAM events, or from service layer original sources such as application performance (solution #1), edge load (solution #3), (NOTE 2) | |
| Data generation schedule | | O | | The schedule of data generation, e.g. when the data source is active to produce data. | |
| Data source role | | O | | Role of the data source, e.g., original source, repository, logging server, etc. | |
| Original source | | O | | If the data source role is not original source, specifies the original data source of the data provided by this data source. | |
| Data freshness | | O | | If the data source role is not original source, length of time elapsed after the data is generated until is available at the data source. Alternatively, the data collection rate supported by the source is provided | |
| Data storage capability | | O | | Indicates data storage capabilities, e.g. how long the data can be stored. | |
| Anonymization capability | | O | | Indicates whether the data available at this data source can be anonymized before collection. | |
| Pre-processing capabilities | | O | | Indicates capabilities of the data source to provide pre-processing functionality, such as aggregation, validation, etc. | |
| Original source communication constraints | | O | | Constraints of the original source such as geographic constraints, access technology associated with the original data source, etc. | |
| NOTE 1: The list of possible choices may be determined in the specification phase, based on ADAES capabilities to interact with other service layer entities  NOTE 2: The values available for "information type" may be determined in the specification phase. | | | | | |

Table 6.3.1.3-2 describes information elements in the data analytics task request from the requester to ADAE or from ADAE client to ADAE server.

Table 6.3.1.3-2: Data analytics request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| Requester ID | M | ID of the requesting server/client making the request |
| Data source profile criteria | M | List of criteria for determining the data sources, based on the IEs present in the Data source profile table 6.3.1.3-1. |
| Information filter | O | Parameters and constraints of the information collected for the task. For example, the filter may specify a PLMN, a service provider, area of interest, time window of interest, reporting threshold (for NWDAF subscriptions), etc…. |
| Task processing operation | O | Specifies data processing operations that need to be performed on the collected data to produce the analytics results, such as normalization, rounding, clean-up, etc. to generate the task results.  When the task requires data to be collected from multiple sources, this parameter may specify operations to combine/merge the data from different sources such as aggregation, validation, data alignment, etc. |
| Task requirements | O | Specifies requirements on the data analytics task or the resulting data, e.g., when the task results are needed, where the resulting data should be stored, required data sampling/updating rate, required number of data samples, required amount (size) of collected data (dataset), required data granularity/accuracy, required level of confidence, etc. |
| Existing input data | O | Specifies input data that may already be available. |

Table 6.3.1.3-3 describes information elements in the data analytics response from ADAE to the requester.

Table 6.3.1.3-3: Data analytics response

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| ADAES ID | M | ID of the ADAES |
| Task results | O | Output of the data analytics task. |

Table 6.3.1.3-4 describes information elements in the data collection request from ADAE to a data source.

Table 6.3.1.3-4: Data collection request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| ADAES ID | M | ID of the ADAES |
| Original data source | M | Identifies all the original data sources. |
| Data required | M | Identifies the data to be collected. |

Table 6.3.1.3-5 describes information elements in the data collection response from the data source to ADAE.

Table 6.3.1.3-5: Data collection response

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| ADAES ID | M | ID of the ADAES |
| Data collection output | O | Output of the data collection request. |

### 6.3.2 Evaluation

The solution addresses Key Issue #1, 2, 3, 4 on support for data collection for application and provides generic procedures and APIs to support different types of data analytics (e.g., application performance analytics, edge analytics, etc.).

The solution defines generic procedures for application data analytics enablement service enablement, including interactions with other service enablement layer entities (e.g., SEAL, EEL) or the core network. The solution also enables ADAES to select data sources and to collect data for the analytics services provided. The solution specifies the API for requesting application data analytics enablement service and for enabling the collection and preparation of data that are originated from different sources.

## 6.4 Solution #3: Support for edge load analytics

### 6.4.1 Solution description

This solution addresses Key Issue #2.

This solution introduces edge load analytics to provide insight on the operation and performance of an EDN and in particular statistics or prediction on parameters related to:

- the EAS / EES load for one or more EAS/EES

- edge platform load parameters, which include the aggregated load per EDN or per DNAI due to the edge support services and e.g., load level of edge computational resources.

Such analytics can improve edge support services by allowing the pro-active edge service operation changes to deal with possible edge overload scenarios. For example, this can trigger EAS migration to a different EDN / central DN, or pro-active EAS reselection for a target UE or group of UEs.

Figure 6.4.1-1 illustrates the procedure where the edge analytics are performed based on data collected from the EDN (EAS and/or EES, edge database or networking stack at EDN side).

Pre-conditions:

1. ADAES has discovered the APIs to access the edge services at EDN.



Figure 6.4.1-1: ADAES support for edge analytics

1. The consumer of the ADAES analytics service sends a subscription request to ADAES and provides the analytics event ID e.g. edge performance prediction or stats, the DNN / DNAI, the time validity and area of the request, the required confidence level, whether offline and/or online analytics are needed etc.

2. The ADAES sends a subscription response as an ACK to the consumer.

3. The ADAES maps the analytics event ID to a list of data collection event identifiers, and optionally a list of data producer IDs. Such mapping may be preconfigured by OAM or may be configured at ADAES based on the analytics event ID. Such Data Producers can be EASs onboarded to EDN, EESs, SEALDD server, MEP services (e.g. RNIS).

4. The ADAES sends a subscription request to the Data Producers (EASs onboarded to EDN, EESs, SEALDD server, RNIS, N6 endpoint at EDN, NWDAF, OAM) with the respective Data Collection Event ID and the requirement for data collection. This message includes the Data Collection event ID and/or the analytics event ID, the ADAES ID, the time validity and area of the request, the required confidence level etc.

5. The Data Producer(s) sends a subscription response as an ACK to the ADAES.

6. The ADAES based on subscription, may receive offline stats/data on the edge DN load based on the analytics/data collection event ID from the data producer or from A-ADRF (see clause 5.3.4). Such offline data can be per EDN or per DNAI or per EAS/EES load statistics and edge computational resource utilization stats for a given time and area of interest. One example can be the load in terms of number of EAS or EES connections for a given area or time window, or the average edge computational resource usage or usage ratio based on the EDN total resource availability, EDN overload/high load indication events, probability of EAS/EES unavailability due to high load, etc.

7. The Data Producers at the edge start collecting data. Such data can be measurements or analytics based on the data source/producer, as follows:

- from OAM or EAS/ASP (for EAS load info): Per EAS/EES computational resource load, number of connections per EES/EAS

- from SEALDD server / N6 endpoint: N6 load / SEALDD server load

- from 5GC / NWDAF: DN performance analytics

- from OAM / MDAS: UPF load analytics (per DNAI)

- from MEC platform services (e.g., RNIS): per cell radio conditions / load for all cells within EDN coverage

NOTE: How the ADAES obtains the EAS load information from EAS/ASP is up to implementation.

8. The Data Producer send the data to the ADAES (based on step 7 measurements or analytics), where the data correspond to the data collection ID or the analytics event ID for which the ADAES subscribed. Such data can be provided one time or periodically or based on a threshold (e.g., load >X%).

9. The ADAES derives edge analytics on EDN / DNAI load or per EES/EAS load, based on the analytics ID and type of request. The analytics are derived based on the performance analytics received per DN or load analytics per DNAI/UPF; as well as considering measurements on the computational or RAN resource load or number of connections for the EES/EASs which are active at the EDN. Such analytics can be stats or prediction for a given area/time and based on the event type for a given network configuration. Such analytics can be for example a predicted load indication for the EDN or for an EES or EAS (e.g. 50% load or medium /high load), a predictive load in terms of number of EAS or EES connections for a given area or time window, or the predicted computational resource usage or usage ratio based on the EDN total resource availability, EDN overload/high load indication events, probability of EAS/EES unavailability due to high load.

10**.** The ADAES sends the edge analytics to the consumer, based on the request and the derived analytics in step 9. Such analytics indicate a prediction of the EDN load considering inputs from both 5GS as well as from edge platform services. Such prediction can also be in form of a recommendation for triggering an EAS relocation to a different platform.

### 6.4.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #3.

- Inputs: edge platform load data, EAS/EES load data, DN performance data, UPF load analytics.

- List of Data Sources

- Data Source #1 information: OAM / MDAS

- Data required from Data Source #1: UPF load analytics

- Data Source #2 information: 5GC / NWDAF

- Data required from Data Source #2: DN performance analytics

- Data Source #3 information: SEALDD server

- Data required from Data Source #3: N6 load measurements, SEALDD computational resource load

- Data Source #4 information: EES

- Data required from Data Source #4: EES computational resource load or number of connections of EES

- Data Source #5 information: EAS

- Data required from Data Source #5: EAS computational resource load or number of connections of EAS

- Data Source #6 information: RNIS

- Data required from Data Source #6: per cell average radio conditions and resource utilization (for all cells within edge coverage)

- Output: stats or predictions on the EDN load conditions, EES or EAS load stats/predictions, recommendation for EAS relocation trigger (due to expected high load or edge resources).

### 6.4.3 Solution evaluation

This solution addresses Key Issue #2 and introduces edge data analytics to predict the load of an edge platform or an edge service. Such analytics can improve edge support services by allowing the pro-active edge service operation changes to deal with possible edge overload scenarios.

This solution is feasible and doesn't introduce any dependency to 3GPP network systems. For the interaction to 5GC, ADAE server acts as AF, whereas for the interaction to OAM, ADAE server can be seen as a trusted 3rd party MDA service consumer (for consuming UPF load analytics). For the data collection related to ETSI MEC service like RNIS, this is only possible if the EDN has deployed such service, and any interaction between ADAE server and RNIS can be either up to ECSP implementation or by re-using EDGE-3 interface (RNIS acting as EAS).

## 6.5 Solution #4: Support for performance analytics for UE-to-UE sessions

### 6.5.1 Solution description

This solution addresses Key Issue #1.

This solution introduces application layer analytics to predict the performance of an application session among two or more VAL UEs within a service or group. Such prediction relates to application QoS attributes prediction for a given time horizon and area. This can be requested by the VAL server during the session, or the VAL server can subscribe to receive predicted application QoS downgrade indication for an ongoing session. Such analytics will help improving the application service experience and allow the VAL layer to pro-actively adapt to predicted application QoS changes.

Figure 6.5.1-1 illustrates the procedure where the VAL session performance analytics are performed based on data collected from the ongoing VAL sessions.

Pre-conditions:

1. ADAECs are connected to ADAES



Figure 6.5.1-1: ADAES support for VAL session performance analytics

1. The consumer of the ADAES analytics service sends a subscription request to ADAES and provides the analytics event ID e.g. "VAL UE to UE session prediction", the target VAL UE ID or group of UE IDs, the VAL session / service ID, the time validity and area of the request, the required confidence level, exposure level for providing UE to UE analytics. Such request can also include whether the analytics notification shall be periodic or based on an expected application QoS change (in that case also the thresholds can be provided at the request)

2. The ADAES sends a subscription response as an ACK to the consumer.

3. The ADAES selects the corresponding ADAEC #1 of the VAL UE 1 where the session performance analytics need to be performed. Such UE can be for example a capable and authorized UE from the involved VAL UEs within the service or group, e.g. a group lead.

4a. The ADAES sends a subscription request to the ADAEC #1 with the analytics event ID and the configuration of the reporting required (e.g., periodic, based on threshold or event). Such request also includes the application QoS attributes to be analyzed (latency, jitter, PER,..)

4b. The ADAEC #1 sends a subscription response to ADAES

A session starts between the VAL UE #1 and a VAL UE #2 (or more VAL UEs).

5. The ADAEC #1 (after being aware from the VAL client that the session started) sends a notification to ADAES that a session started, and it could be possible to provide real-time data analytics for VAL UE to UE session performance in the target area.

6. The ADAEC #1 starts collecting data from the corresponding VAL UE(s) based on subscription. Such data can be about the latency, throughput, jitter, QoE measurements, PQI load, etc. The data can be collected by ADAEC #1 from other ADAECs via ADAE-C interface, or from the VAL clients (VAL client to VAL client interaction is out of scope).

7. The ADAEC either detects or predicts an application QoS change (depending on the authorization of ADAEC to perform analytics). Such change can be for example an application QoS downgrade related to the UE-to-UE session latency, or the PER/channel losses higher than a predefined threshold, for a given time horizon with a certain confidence level.

8. The ADAEC sends the data or the analytics (depending on whether ADAEC provides a prediction) to the ADAES.

9. The ADAES based on the received notification, confirms/verifies the analytics received or provides analytics (in case that data were reported) for the UE-to-UE session. Such analytics can be about predicting the application QoS change for the UE-to-UE session.

13**.** The ADAES sends the analytics to the consumer.

### 6.5.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #x

- Inputs: per UE-to-UE session performance measurements (application QoS measurements such as latency, channel losses, data rate, jitter)

- List of Data Sources

- Data Source information: VAL UE #1, VAL UE #2

- Data required from Data Source: application QoS measurements

- Output: Predicted application QoS metric change (downgrade or upgrade)

### 6.5.3 Solution evaluation

This solution addresses Key Issue #1 and introduces application layer analytics to predict the performance of an application session among two or more VAL UEs. Such solution enables the VAL layer to pro-actively adapt to predicted application QoS changes for VAL UE-to-UE sessions.

This solution is technically viable and doesn't introduce any impact on 5GS.

## 6.6 Solution #5: Service experience to support application performance analytics

### 6.6.1 Solution description

This solution addresses the open issues stated in Key Issue #1, specifically on what data to be collected. This solution supplements the existing solution #1 in clause 6.2 with service experience information.

In some scenarios, where data from Application servers (like VAL server) is not available (overload or any other reasons) or doesn't show the quality of service experience at the UE side, the ADAE server may need to rely on alternate information sources like the application clients (like VAL clients) that provide the visibility on application service status. ADAE server can use this information from the clients alone, for the predictions and share with the consumer of the analytics. This solution supports a mechanism for the ADAE client to send service experience report to the ADAE server. ADAE server upon receiving the service experience information from the UE side entities can use it for predictions of application performance analytics. The service experience information from VAL client may include end-to-end response time, connection bandwidth, request rate, VAL server availability, etc.

This solution describes three procedures in more detail:

- Pull service experience information

- Push service experience information

- Service experience information based on triggers configured by ADAE server at ADAE client.

#### 6.6.1.1 Pull service experience information

Figure 6.6.1.1-1 illustrates the procedure where the ADAE server pulls the service experience information from the ADAE client.



Figure 6.6.1.1-1: Pull service experience informtion from UE

The procedure can be initiated by the ADAE server upon receiving a service experience from an ADAE client, to fetch service experience information other ADAE clients or upon receiving VAL server performance analytics request from application service provider (application server) or any other event that requires the ADAE server to determine the service experience data.

1. The ADAE server sends Pull service experience request to the ADAE client. The request contains identity of the specific VAL server and VAL service ID, for which the service experience report is required, as mentioned in Table 6.6.1.1-1.

Table 6.6.1.1-1: Pull service experience information request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| VAL server Identity | M | Identity of the VAL server for which the service experience information is requested. |
| VAL service ID | O | Identity of the VAL service. |

2. Upon receiving the Pull service experience request from the ADAE server, the ADAE client may take user consent to send the report if the user consent is not available already.

3. The ADAE client sends the Pull service experience response to the ADAE server. The response contains service experience report, as specified in Table 6.6.1.1-2. The ADAE client receives the service experience related information from the VAL client.

Table 6.6.1.1-2: Pull service experience information request response

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| Result | M | Indicates whether the report is available or not |
| VAL UE ID | M | Identity of the VAL UE |
| VAL service ID (NOTE) | O | Identity of the VAL service. |
| VAL Server Id | M | Identify the VAL server for which the service experience report is sent |
| Timestamp (NOTE) | O | Time stamp of the collected report |
| VAL service experience report (NOTE) | O | Information related to VAL service experience. It may include end-to-end response time, connection bandwidth, request rate, VAL server availability, etc. |
| NOTE: These IEs are included only if the result is success. | | |

4. The ADAE server uses the service experience report for derivation of VAL server performance analytics.

#### 6.6.1.2 Push service experience information

Figure 6.6.1.2-1 illustrates the procedure where the ADAE client pushes the service experience information to the ADAE server.



Figure 6.6.1.2-1: Push service experience informtion from UE

The ADAE client determines the service experience information based on information received from the VAL client. The service experience information includes application specific performance measurements like end-to-end response time, connection bandwidth, request rate, server availability time, etc. On request VAL client or any other trigger conditions, the ADAE client sends the service experience report about a VAL server to the ADAE server.

1. The ADAE client sends Push service experience request to the ADAE server. The request contains service experience report about a VAL server and includes the information elements as specified in Table 6.6.1.2-1.

Table 6.6.1.2-1: Push service experience information request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| VAL UE ID | M | Identity of the VAL UE |
| VAL service ID | O | Identity of the VAL service. |
| VAL Server Id | M | Identify the VAL server for which the service experience report is sent |
| Timestamp | O | Time stamp of the collected report |
| VAL service experience report | O | Information related to VAL service experience. It may include end-to-end response time, connection bandwidth, request rate, VAL server availability, etc. |

2. Upon receiving the Push service experience request from the ADAE client, the ADAE server uses the service experience report for derivation of VAL server performance analytics.

3. The ADAE server sends Push service experience response to the ADAE client.

The ADAE server may take further actions based on the analysis of the report as shared by the ADAE client. A service experience information from certain UEs, can trigger the ADAE server to fetch further service experience information other UEs. Use the service experience information report from other UEs, to determine/predict analytics.

- If most of the UE side entities report a similar service experience, then it could be the application server problem across globally.

- If only some UEs report a bad service experience, the problem could be localized among a group of UEs.

- If the bad service experience from only a UE, the problem is localized to the UE.

#### 6.6.1.3 Service experience information based on triggers

Figure 6.6.1.3-1 illustrates procedure for the ADAE server to configure triggers to the ADAE client to send the service experience report. The procedure can be initiated by the ADAE server upon receiving VAL server performance analytics request from application service provider (application server).



Figure 6.6.1.3-1: Configure service experience report trigger

1. The ADAE server sends Configure service experience report trigger request to the ADAE client. The request contains identity of the specific VAL server(s) for which the service experience report is required. The request includes the information elements as specified in Table 6.6.1.3-1.

Table 6.6.1.3-1: Configure service experience report trigger request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| VAL Server specific criteria | M | Identify the list of VAL servers for which the service experience report is requested  List of VAL server specific criteria |
| > VAL Server Id | M | Identity of the VAL server |
| > Triggering Criteria | M | Information about the triggers on which the service experience is to be reported for the VAL server |
| Common Triggering criteria | O | Information about the triggers (applicable to all VAL servers) on which the service experience is fetched |
| Service experience measurement to monitor | O | Information about the service experience measurements which needs to be fetched and included in the report. If not present, by default end-to-end response time is measured. |
| Notification Target Address | O | The Notification target address (e.g. URL) where the notifications destined for the ASM Server should be sent to. |

2. Upon receiving the Configure service experience report trigger request from the ADAE server, the ADAE client stores the triggering criteria for sending service experience report and may take user consent to send the report if the user consent is not available already.

3. The ADAE client sends the Configure service experience report trigger response to the ASM server, indicating the result of the request.

### 6.6.2 Solution evaluation

This solution addresses Key Issue #1 and the introduces a mechanism for the ADAE server to use the service experience information from the UE side entities via ADAE clients to determine the application performance analytics. The solution proposes three mechanisms, ADAE server pulls the service experience information from the ADAE, ADAE server pulls the service experience information from ADAE client based on certain triggers configured on the ADAE client by the ADAE server, and ADAE client pushes to ADAE server.

## 6.7 Solution #6: Support for slice related application data analytics

### 6.7.1 Solution description

This solution addresses Key Issue #5.

This solution introduces application layer analytics to provide insight on the performance of the VAL applications when using a given network slice (from a list of subscribed slices for the VAL customer). Such solution provides an analytics service to a consumer who can be either the VAL server (for helping to identify what slice it will use for its applications) or for other consumers such as SEAL NSCE to support on providing analytics (since NSCE doesn't contain an analytics engine for providing analytics on top of NWDAF/MDAS).

Figure 6.7.1-1 illustrates the procedure where the VAL server performance analytics are performed based on data collected from the ongoing VAL sessions as well as data from the DN (VAL server, DN database or networking stack at DN).

Pre-conditions:

1. ADAEC is connected to ADAES



Figure 6.7.1-1: ADAES support for slice-related performance analytics

1. The consumer of the ADAES analytics service sends a subscription request to ADAES and provides the analytics event ID e.g. "slice perf prediction", the target S-NSSAI, DNN, NSI ID, the time validity and area of the request, the required confidence level, whether offline and/or online analytics are needed etc.

2. The ADAES sends a subscription response as an ACK to the consumer.

3. The ADAES subscribes to the Data Sources with the respective Data Collection Event ID and the requirement for data collection related to the request slice(s). Such requests can be towards:

- OAM for providing PM data related to the requested slice / NSI. Alternatively, if the interaction to OAM happens via NSCE layer, such subscription can be performed to NSCE (where ADAES is acting as VAL server).

- NWDAF for providing slice related analytics for the given area and time horizon (indicated in step 1). Such analytics can be the slice load level related network data analytics, or the service experience related network data analytics for a given slice

4. The ADAES based on subscription, receives PM data notification from OAM or from NSCE (via OAM APIs or NSCE-S APIs)

5. The ADAES based on subscription, receives the requested NWDAF analytics outputs. Such analytics can be:

- network slice or NSI statistics or predictions (clause 6.3.3 of TS 23.288)

- per slice instance service experience stats or predictions (clause 6.4.3 of TS 23.288)

6. The ADAES can also provide analytics on the VAL session performance based on Solution #1 procedure 6.1.2 and filters the analytics only for the sessions which are connected to that requested slice for the area of interest.

7. The ADAES abstracts or correlates the data/analytics from steps 4-6 and provides analytics on the slice or NSI performance for the the target VAL application/server. For example, such analytics can be about the min/average/max predicted RTT / end to end latency for the VAL application/server if this server uses a given slice/NSI (or for a list of given slices) within an area of interest.

8**.** The ADAES sends the analytics to the consumer.

### 6.7.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #x

- Inputs: per slice measurements and analytics, application session performance analytics, historical data on slice information

- List of Data Sources:

- Data Source #1 information: OAM or NSCE

- Data required from Data Source #1: PM data for a given NSI

- Data Source #2 information: NWDAF

- Data required from Data Source #2: slice load analytics for NSI/S-NSSAI, service experience for NSI/S-NSSAI

- Data Source #3 information: VAL UEs (based on Solution #1)

- Data required from Data Source #3: application QoS measurements for an application session connected to requested slice

- Output: Statistics or prediction for the VAL application QoS for one or more requested S-NSSAIs/NSIs

### 6.7.3 Solution evaluation

This solution introduces application layer analytics to provide insight on the performance of the VAL applications when using a given network slice (from a list of subscribed slices for the VAL customer). This solution is technically viable and does not have any dependency to other slice related analytics since it targets the application performance for a target slice or NSI and not the network slice related performance.

## 6.8 Solution #7: Slice configuration recommendation

### 6.8.1 Solution description

This solution addresses Key Issue #6.

This solution provides a procedure for network slice configuration recommendation based on collected network slice performance and analytics and historical network slice status and network performance. The consumer can be either the VAL server or other consumers such as SEAL NSCE.

Figure 6.8.1-1 illustrates the procedure for network slice configuration recommendation.

Pre-conditions:

1. The ADAES is registered and capable of interacting with 5GS to collected network slice data.

2. The ADAES is registered and capable of interacting with NSCE to collected network slice performance and analytic.



Figure 6.8.1-1: ADAES support for network slice configuration recommendation

1. The consumer of the ADAES sends a subscription request to ADAES and provides the target S-NSSAI, DNN, slice requirement, area of the interest, interest time period of the historical data (e.g. last year), the required confidence level, whether offline and/or online analytic are needed etc.

2. The ADAES sends a subscription response to the consumer.

3. The ADAES subscribes to the Data Sources with the respective Data Collection Event ID and the requirement for data collection related to the request slice(s). Such requests can be sent to SEAL NSCE, OAM, NWDAF or the combination of them.

4. Based on subscription, the ADAES (acting as VAL server) may receive performance and analytics data from SEAL NSCE (e.g. QoE metrics, as defined in the TS 23.435 clause 9.4.2).

5. Based on subscription, the ADAES may receive Network slice / NSI related performance data from OAM as defined in TS 28.552 [12].

6. Based on subscription, the ADAES may receives Network slice related Observed Service experience statistics, Load level information of a Network Slice defined from NWDAF as defined in TS 23.288[18]

7. If the data is collected from multiple sources, the ADAES combines or correlates the data/analytics from steps 3-5, and stores the data into data repository if needed.

8. Collect the historical data from data repository and analyze the network slice usage pattern. When the amount of stored historical data does not cover the required interest time period of the historical data, ADAES analyze the slice usage pattern based on the existing stored historical data.

9. The ADAES provides network slice configuration recommendation based on the slice requirement, slice performance and derived slice usage pattern from step 8.

8**.** The ADAES sends the network slice configuration recommendation to the consumer. The recommendation may be related to parameters in the slice serviceProfile if the consumer is the SEAL NSCE. Or the recommendation may be related to slice resource /functional configuration (e.g. slice capacity, coverage) if the consumer is the management system.

### 6.8.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #7

- Inputs: per slice measurements and analytics, historical data on slice information

- List of Data Sources:

- Data Source #1 information: SEAL NSCE

- Data required from Data Source #1: performance and analytics data for a given S-NSSAI

- Data Source #2 information: OAM

- Data required from Data Source #2: PM data for a given S-NSSAI

- Data Source #3 information: NWDAF

- Data required from Data Source #3: slice load analytics for S-NSSAI, service experience for S-NSSAI

- Data Source #4 information: A-ADRF

- Data required from Data Source #4: historical slice load analytics and service experience for S-NSSAI.

- Output: Statistics for the network slice configuration recommendation for one or more requested S-NSSAIs. The recommendation may be related to parameters in the slice serviceProfile per S-NSSAI if the consumer is the SEAL NSCE. Or the recommendation may be related to slice resource /functional configuration per S-NSSAI / NSI (e.g. slice capacity, coverage) if the consumer is the management system.

### 6.8.3 Solution evaluation

This solution is technically viable and does not have any dependency to other slice related analytics since it targets the slice configuration recommendation for a target S-NSSAI.

## 6.9 Solution #8: support for location accuracy analytics

### 6.9.1 Solution description

This solution addresses Key Issue #7.

This solution introduces application layer analytics to allow a VAL server to be notified based on analytics whether the accuracy of a location can be met for a given application and optionally for a given UE/group route. For example, a VAL server may request the ADAE server to provide analytics whether the accuracy of a location for the UEs within a VAL application is predicted to be sustainable or is expected to downgrade in a specific area or for an expected route from location A to location B.

Figure 6.9.1-1 illustrates the procedure for location accuracy analytics enablement solution.

Pre-conditions:

1. ADAES is connected to A-ADRF

2. ADAES has discovered SEAL LMS or FLS



Figure 6.9.1 -1: Location accuracy analytics procedure

1. The VAL server makes a request to ADAE server for location accuracy prediction/stats, including an analytics event ID (e.g. "location accuracy prediction" or "location accuracy sustainability"), an analytics request type (if not identified specifically at the event ID) which can be the location accuracy prediction for a given location X and/or for a given UE/app. The request may include also the target area, a target VAL application, or a VAL UE, group of UEs or the VAL service, time of day, accuracy threshold and requirements. If the VAL UEs are provided by the VAL server, this request may also include the expected route or a set of waypoints for the UEs of the VAL application.

2. The ADAE server sends a subscription response as an ACK to the VAL server.

3. The ADAE server discovers and maps the Data Sources with the respective analytics event ID for collecting location data for the corresponding VAL UEs or VAL service or area.

4a/4b. The ADAE server subscribes for NWDAF UE mobility analytics per VAL UE (for all the VAL UEs) and gets notification on the per UE location/mobility analytics based on TS 23.288 clause 6.7.2. Such analytics may be requested for a list of waypoints per UE route (if indicated at step 1).

5a/5b. The ADAE server requests and receives from SEAL LMS location reports for the respective VAL UEs or location reports from all VAL UEs within the requested area. Such request may also indicate the required location accuracy (requires enhancements to SEAL LMS procedures). In case where FLS is deployed, it can be also possible to request and receive fused location reports (combined location reports from 3gpp/n3gpp sources) if the ADAE server identifies that higher location accuracy is required.

6. The ADAE server may also request and receive location accuracy historical analytics /data from A-ADRF for the corresponding VAL UEs or VAL service area.

7. The ADAE server abstracts or correlates the data/analytics from steps 4-6 and provides analytics on the location accuracy for the the target VAL application. Depending on the event ID in step 1, the ADAE server can indicate whether the location accuracy is sustainable or is predicted to be downgraded or can be upgraded and become more granular (e.g. from meter to decimetre).

8. The ADAE server sends the analytics to the consumer.

### 6.9.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #8

- Inputs: per VAL UE or area location reports and UE mobility analytics

- List of Data Sources:

- Data Source #1 information: NWDAF (Nnwdaf\_AnalyticsInfo service via N33, as specified in TS 23.288 [2])

- Data required from Data Source #1: UE mobility analytics

- Data Source #2 information: SEAL LMS (SS\_LocationMonitoring API, SS\_LocationAreaMonitoring API as specified in TS 23.434 [5])

- Data required from Data Source #2: UE location reports and achieved accuracy, Location reports from all UEs within a given area

- Data Source #3 information: FLF

- Data required from Data Source #3: fused location reports per VAL UE (via FLS-2 as discussed in TR 23.700-96 [8])

- Data Source #4 information: A-ADRF

- Data required from Data Source #4: historical location accuracy statistics for target VAL service area or VAL UE

- Output: a predictive location accuracy sustainability or change indication, a predictive location accuracy sustainability or change indication for a route of the UE with the application.

### 6.9.3 Solution evaluation

This solution addresses Key Issue #7 and introduces location accuracy analytics to predict possible downgrade of location accuracy for a given VAL application. Such solution enables the VAL layer to pro-actively adapt to predicted location accuracy changes for the VAL service (e.g. to change the distance between VAL UEs or trigger the change of VAL UE speed if location accuracy is changed).

This solution is technically viable and doesn't introduce any impact on 5GS. This solution requires enhancements to SEAL LMS and to FLS for reporting the accuracy as part of the location reporting.

## 6.10 Solution #9: support for service API analytics

### 6.10.1 Solution description

This solution addresses Key Issue #8.

This solution introduces service API analytics to allow a VAL server or any other consumer (e.g. API provider) to be notified on the predicted /statistic availability and service level for the requested service API analytics. Such analytics may allow the API provider to perform actions to avoid service API invocation failures or other actions like throttling/rate limitations. Also, such analytics will support the VAL server to identify if/when to perform an API invocation request based on the API expected status at the given area and time horizon.

Figure 6.10.1-1 illustrates the procedure for service API analytics enablement solution.

Pre-conditions:

1. ADAES is registered to CCF



Figure 6.10.1 -1: Service API analytics procedure

1. The subscribing entity (CAPIF entity, VAL server / API invoker, API provider) sends an event subscription request to the ADAE server to receive analytics for one or more service APIs. The event subscription request includes the information elements as specified in Table 6.10.1-1

Table 6.10.1-1: Event subscription request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| Identity information | M | The information to determine the identity of the subscribing entity |
| Service API information | M | The service API name or type |
| Analytics event ID and criteria | M | The event criteria include event type information relevant to the prediction or stats on the number of failure API invocations, API availability, frequency and occurrence of API version changes, API location changes for the target API, etc |
| Time Validity and horizon | O | Time validity of the request and time horizon for the predictions |
| Area of interest | O | Geographical or topological area for which the subscription applies |
| Notification reception information | O | The information of the subscribing entity for receiving the notifications for the event. |

2.Upon receiving the event subscription request from the subscribing entity, the ADAE server checks for the relevant authorization for the event subscription. If the authorization is successful, the ADAE server stores the subscription information.

3. The ADAE server sends an event subscription response indicating successful subscription

4. Upon sending the subscription response, the ADAE server requests to collect API logs to be used to derive analytics and triggers API invocation log pull request towards the CAPIF core function. The API invocation log fetch request indicates the API (or list of APIs) for which logs are required. Based on the ADAE server deployment, this can be performed via CAPIF\_Logging\_API\_Invocation API as specified in 3GPP TS 23.222 [9]. The message may include the IEs as specified in Table 6.10.1-2.

Table 6.10.1-2: API invocation log fetch request

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| Service API log requestor information | M | Identity information of the originated application querying service API log request |
| ADAES ID / Security credentials | M | Identity information of the ADAES |
| Service ID or UE ID | M | Identity of the application service or UE for which the API invocations apply |
| Target API information / List of API information | M | Information on target API or list of target APIs |
| >Query information | O | List of query filters such as invoker's ID and IP address, service API name and version, input parameters, and invocation result |
| > API aggregation abstraction flag | O | What type of aggregation or abstraction/filtering needs to be applied |
| Reporting format configuration | O | The logs reporting configuration (frequency, periodicity etc) |
| Area of validity | O | The geographical area for which the request applies |
| Time or validity | O | The time of validity for the request |
| Exposure level requirement | O | The level of exposure requirement (e.g. permissions on the logs like read/write/delete..) for the logs to be exposed |

5. The CCF authorizes the request and fetches the API logs from the storage unit. CCF then sends the requested information to the ADAE server via a API invocation log fetch response message. The message may include the IEs as specified in Table 6.10.1.-3.

Table 6.10.1-3: API invocation log fetch response

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| Result | M | Identity information of the originated application querying service API log request |
| Service ID or UE ID | M | Identity of the application service or UE for which the API invocations apply |
| Target API information / List of API information | M | Information on target API or list of target APIs |
| >API Logs Collection Event | O | The API logs based on the subscription event. This includes for example the number of failure API invocations, API availability, frequency and occurrence of API version changes, API location changes for the target API, API throttling events, number of API invocations for a given area and time etc |
| > reporting info | O | The time and area for which the reporting applies |

6. The ADAES may also request and receive service API historical analytics /data from A-ADRF for the corresponding service APIs.

7. The ADAE servers authorizes and anonymizes the API logs (if not performed by CCF) and abstracts based on exposure level. The exposure level can be known based on pre-configuration by the OAM or based on the subscription and type of invoker. The ADAE server then derives analytics on the target service API(s) based on the logs received from the CCF. Such analytics are predictions/stats for the API status based on the analytics event.

8. The ADAE server sends the analytics as event notifications to all the subscribing entities that have subscribed for the event matching the criteria. If a notification reception information is available as part of the subscribing entity event subscription, then the notification reception information is used by the ADAE server to send event notifications to the subscribing entity.

The notification includes the following parameters: the analytics event ID, the service API name and/or type, stats or predictions based on abstracted or anonymized API logs (for example number of failure API invocations, API availability, frequency and occurrence of API version changes, API location changes for the target API, API throttling events, number of API invocations for a given area and time, API load statistics for a given edge network, etc), the time and area where the reporting occurs and is valid.

### 6.10.2 Corresponding Analytics API

This subclause provides a summary on the corresponding Analytics API for solution #9.

- Inputs: service API logs

- List of Data Sources:

- Data Source #1 information: CAPIF CCF (via CAPIF\_Logging\_API\_Invocation API)

- Data required from Data Source #1: Service API logs for requested APIs

- Data Source #2 information: A-ADRF

- Data required from Data Source #2: historical data / statistics on service API availability and service level

- Output: stats or predictions for service API(s). For example, the failure rate of API invocations, API predicted availability, frequency and occurrence of API version changes, API location changes for the target API, API throttling events, number of API invocations for a given area and time, API load statistics for a given edge data network (for edge provided service APIs).

### 6.10.3 Solution evaluation

This solution addresses Key Issue #8 and introduces service API analytics to provide stats or predict possible downgrade of performance and availability of a service API. This solution is technically viable and doesn't introduce any impact on 5GS.

# 7 Deployment scenarios

## 7.1 General

This clause provides the different deployment models for ADAE services. There could be three deployment options:

- ADAES can be deployed at a centralized cloud platform, and collects data from multiple EDNs

- ADAES can be deployed at the edge platform

- Hierarchical ADAES deployment, where multiple ADAE services are deployed in edge or central clouds (e.g. in hierarchical arch). Such deployment allows for local-global analytics for system wide optimization

## 7.2 Deployment model #1: Cloud-deployed ADAES

In this deployment, as shown in Figure 7.2-1, the ADAES is centrally located and can provide analytics services to the application and edge services (EAS/EES, VAL server, SEAL services).

The statistics/predictions that the ADAES provides are applicable to the ADAE server service area, which can be provided for the entire PLMN.



Figure 7.2-1 cloud deployed ADAES

## 7.3 Deployment model #2 Edge-deployed ADAES

In this deployment, as shown in Figure 7.3-1, the ADAES is located at the EDN and provides analytics services to the EAS or other edge native applications at the edge platform. ADAES can be deployed by the ECSP or the MNO to provide analytics for the application or edge parameters.

The statistics/predictions that the edge deployed ADAES are applicable to the ADAE server service areas (as shown in the example in Fig 7.2-2), which are equivalent to the EDN service areas. Such analytics can be about the edge load or the EAS performance and can be provided to consumers within EDN.

In this deployment the interaction between edge deployed ADAES is possible for exchanging edge/application analytics for application mobility scenarios or for cases when ADAES #1 and #2 service areas have overlapping coverage.



Figure 7.3-1 edge deployed ADAES

## 7.4 Deployment model #3: Hierarchical ADAES deployment

In this deployment, multiple ADAESs can be located at different EDNs/DNs and can be deployed by the same ADAE provider. Such hierarchical deployments allow the local – global analytics derivation (which may be needed for improving the analytics confidence level). The centrally deployed ADAES can also act as ADAE analytics aggregator entity and configures the edge deployed ADAES to derive analytics on different sub-areas.

One example is the use of analytics for the EDN#1 or EDN#2 load which will help predicting the VAL server performance at a centrally located ADAES. Such deployment is also applicable for ML-based analytics methods, like supervised learning, where the centrally located ADAES acts as ML model training entity, and the edge located ADAESs can act as ML model inference entities (using edge data to improve the prediction accuracy).

The statistics/predictions that the edge deployed ADAES correspond to the ADAE server service areas (as shown in the example in Fig 7.4-1), which is equivalent to the EDN service areas. The central ADAE server covers all PLMN area and is used to coordinate or jointly perform analytics with the distributed ADAES. Such analytics services can be provided to consumers at the central DN, like the VAL servers or SEAL services or even at the PLMN side (e.g. NWDAF consuming service experience analytics).



Figure 7.4-1 hierarchical deployment of ADAES

# 8 Overall evaluation

## 8.1 General

The following clauses contain an overall evaluation of the solutions presented in this technical report, their applicability to the identified key issues and possible dependencies to other groups. This clause also includes a summary of the inputs/outputs and corresponding analytics APIs to be considered for the normative phase.

## 8.2 Solution evaluations

### 8.2.1 General

All the key issues and solutions specified in this technical report are listed in Table 8.2.1-1. This table includes the mapping of the key issues to the solutions and corresponding solution evaluations.

Table 8.2.1-1 Key issue and solutions

|  |  |  |
| --- | --- | --- |
| Key issues | Solution | Dependency on other working groups |
| Key issue #1: Support for application performance analytics | Solution #1: Support for application performance analytics |  |
| Solution #2: Data Analytics Enablement |
| Solution #4: Support for performance analytics for UE-to-UE sessions |
| Solution #5: Service experience to support application performance analytics |
| Key issue #2: Support for edge analytics enablement | Solution #2: Data Analytics Enablement |  |
| Solution #3: Support for edge load analytics |  |
| Key issue #3: Support for data collection for application layer analytics | Solution #2: Data Analytics Enablement |  |
| Key issue #4: Key Issue on interactions with SEAL services | Solution #2: Data Analytics Enablement |  |
| Solution #6: Support for slice related application data analytics |  |
| Solution #7: Slice configuration recommendation |  |
| Solution #8: Location accuracy analytics |  |
| Key issue #5: Support for slice-related application data analytics | Solution #6: Support for slice related application data analytics |  |
| Key issue #6: Support for slice configuration recommendation enablement | Solution #7: Slice configuration recommendation |  |
| Key issue #7: support for location accuracy analytics | Solution #8: Location accuracy analytics |  |
| Key issue #8: Support for service API capability analytics | Solution #9 : Service API analytics |  |

More specifically,

- For Key issue #1 (Support for application performance analytics), Solution #1 and Solution #4 provide analytics capabilities for application performance targeting the UE-to-network and UE-to-UE sessions respectively. Solution #2 provides a generic mechanism which can be used for data analytics enablement and can be adopted by Solutions #1 and #4; whereas Solution #5 provides methods for collecting/notifying service experience data from the UE and can be seen as further elaboration of the collection/notification means based on Solution #1. There is no identified conflict among the solutions, and the solutions can complement each other.

- For Key issue #2 (Support for edge analytics enablement), Solution #3 provides analytics functionality for edge load, whereas Solution #2 provides a generic mechanism which can be used for edge load analytics enablement and can be adopted by Solution #3 (whereas it is not mandatory). Solution #2 may require enhancements in eEDGEAPP to allow the collection of the EES load information.

- For Key issue #3 (Support for data collection for application layer analytics), Solution #2 provides a generic mechanism for data collection enablement to be used for ADAE analytics derivation.

- For Key issue #4 (Key Issue on interactions with SEAL services), solutions #2, #6, #7, #8 have interactions with other SEAL services. Solution #2 provides a generic mechanism for data collection enablement to be used for ADAE analytics derivation. Additionally, Solution #6 and #7 interact with SEAL NSCE server for analytics derivation. Finally Solution #8 collects location reports from SEAL LMS and may require location enhancements of SEAL LMS. There is no identified conflict among the solutions, and the solutions can complement each other.

- For Key issue #5 (Support for slice-related application data analytics), Solution #6 provides slice-related application analytics and is optionally interacting with SEAL NSCE service (no enhancements are needed at SEAL NSCE).

- For Key issue #6 (Support for slice configuration recommendation enablement), Solution #7 provides a support capability for slice configuration recommendation and is interacting with SEAL NSCE service (no enhancements are needed at SEAL NSCE).

- For Key issue #7 (support for location accuracy), Solution #8 provides location accuracy analytics functionality and is optionally interacting with SEAL LMS or FLF service (enhancements are needed at SEAL LMS if used as data source).

- For Key issue #8 (Support for service API capability analytics), Solution #9 provides a support capability for service API analytics.

### 8.2.2 ADAE analytics services

Table 8.2.2-1 provides an overview of the analytics services which are provided by ADAE layer, based on the individual solutions.

Table 8.2.2-1 ADAE analytics services

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Analytics Event | Solution | Inputs | Data Collection Sources | Analytics Outputs | Type of analytics |
| VAL server performance analytics | Sol #1, Sol #5 | per VAL server performance measurements, historical data/stats for VAL server performance, network / KPI monitoring from 5GS | 1. VAL UE  2. VAL Server  3. OAM  4. 5GC (NWDAF, NEF) | Analytics on application QoS metrics per VAL server | Prediction, statistics |
| VAL UE/session performance analytics | Sol #1, Sol #5 | per VAL session performance measurements | 1. VAL UE  2. VAL Server | Analytics on application QoS metrics per VAL session | Prediction |
| VAL UE-to-UE session performance analytics | Sol #4 | per UE-to-UE session performance measurements | 1. VAL UEs | Analytics on application QoS metric change for UE-to-UE session | Prediction |
| edge load analytics | Sol #3 | edge platform load data, EAS/EES load data, DN performance analytics, UPF load analytics | 1. OAM / MDAS  2. 5GC / NWDAF  3. SEALDD server  4. EES  5. EAS  5. MEP / RNIS | 1. stats / predictions on the EDN load conditions,  2. EES or EAS load stats/predictions, 3. recommendation for pro-active EAS relocation trigger | Prediction, statistics |
| Slice related performance analytics | Sol #6 | per slice measurements and analytics, application session performance analytics, historical data on slice information | 1. OAM or NSCE  2. 5GC / NWDAF  3. VAL UEs | Statistics / prediction for the VAL application QoS for one or more requested S-NSSAIs/NSIs | Prediction, statistics |
| Location accuracy analytics | Sol #8 | UE mobility analytics,  UE location reports and achieved accuracy,  historical location accuracy statistics for target VAL service area or VAL UE | 1. SEAL LMS / FLS  2. 5GC / NWDAF  3. A-ADRF | a predictive location accuracy sustainability or change indication | Prediction |
| Service API analytics | Sol #9 | Service API logs for requested APIs, historical data / statistics on service API availability and service level | 1. CCF  2. A-ADRF | stats / predictions for service API(s) | Prediction, statistics |
| Slice configuration recommendation | Sol #7 | per slice measurements and analytics, historical data on slice information | 1. SEAL NSCE  2. OAM  3. NWDAF  4. A-ADRF | Statistics for the network slice configuration recommendation for one or more requested S-NSSAIs | Statistics |

# 9 Conclusions

## 9.1 General conclusions

This technical report fulfills the objectives of the study on application architecture for enabling application data analytics. The report includes the following:

1. Definition of terms and abbreviations used in the study (clause 3);

2. Key issues identified by the study (clause 4);

3. Architectural requirements and detailed application architecture for enabling Application Data Analytics (clause 5);

4. Individual solutions addressing the key issues (clause 6);

5. Deployment scenarios (clause 7); and

6. Overall evaluations of all the solutions (clause 8);

## 9.2 General conclusions for normative work

For normative work in 3GPP Rel-18, it is recommended to define：

1. Terms and abbreviations, the definition of terms and abbreviations captured in clause 3 will be reused.

2. Requirements on ADAE, the architectural requirements identified in clause 4 will be used as baseline architectural requirements; such requirements include also per functionality-imposed requirements as well as requirements for the internal ADAE architecture.

4. Application architecture for enabling Application Data Analytics Enablement, the architectures as specified in clause 5.3 will be used as baseline architecture.

5. Deployment scenarios will be considered as captured in clause. Additional deployment models and their implications on the solutions will be considered.

6. The definition of ADAE analytics services, data sources and corresponding APIs as captured in clause 8.2.2 based on the concluded solutions (see clause 9.3).

## 9.2 Conclusions of solutions

The study concludes with ADAE functionality, following solution considerations for the normative work:

1. Following individual solutions, corresponding to the key issues, will be considered as candidate solutions:

a. for Key issue #1 (Support for application performance analytics):

i. Solution #1 (Support for application performance analytics)

ii. Solution #2 (Data Analytics Enablement)

iii. Solution #4 (Support for performance analytics for UE-to-UE sessions)

iv. Solution #5 (Service experience to support application performance analytics)

b. for Key issue #2 (Support for edge analytics enablement):

i. Solution #2 (Data Analytics Enablement)

ii. Solution #3 (Support for edge load analytics)

c. for Key issue #3 (Support for data collection for application layer analytics):

i. Solution #2 (Data Analytics Enablement)

d. for Key issue #4 (Key Issue on interactions with SEAL services):

i. Solution #2 (Data Analytics Enablement)

e. for Key issue #5 (Support for slice-related application data analytics):

i. Solution #6 (Support for slice related application data analytics)

f. for Key issue #6 (Support for slice configuration recommendation enablement):

i. Solution #7 (Slice configuration recommendation)

g. for Key issue #7 (Support for location accuracy analytics):

i. Solution #8 (Location accuracy analytics)

h. for Key issue #8 (Support for service API capability analytics):

i. Solution #9 (Service API analytics)

Annex A (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2022-02 | SA6#47-e |  |  |  |  | TR skeleton as approved by SA6 in S6-220082. | 0.0.0 |
| 2022-02 | SA6#47-e |  |  |  |  | Implementation of the following pCRs approved by SA6:  S6-220083, S6-220355, S6-220356, S6-220357, S6-220359, S6-220360, S6-220474. | 0.1.0 |
| 2022-04 | SA6#48-e |  |  |  |  | Implementation of the following pCRs approved by SA6:  S6-220925, S6-220644, S6-220819, S6-220972 | 0.2.0 |
| 2022-05 | SA6#49-e |  |  |  |  | Implementation of the following pCRs approved by SA6:  S6-221154, S6-221349, S6-221301, S6-221350, S6-221351, S6-221204, S6-221161, S6-221474, S6-221352, S6-221156 | 0.3.0 |
| 2022-07 | SA6#49-bis-e |  |  |  |  | Implementation of the following pCRs approved by SA6:  S6-221824, S6-221694, S6-221777, S6-221992, S6-221857, S6-221858, S6-221859, S6-221713, S6-221860, S6-221861 | 0.4.0 |
| 2022-09 | SA6#50-e |  |  |  |  | Implementation of the following pCRs approved by SA6:  S6-222164, S6-222166, S6-222167, S6-222420, S6-222426 | 0.5.0 |
| 2022-09 | SA#97-e | SP-220909 |  |  |  | Presentation for approval at SA#97-e | 1.0.0 |
| 2022-09 | SA#97-e |  |  |  |  | MCC editorial update for publication after TSG SA approval (SA#97) | 18.0.0 |
| 2022-12 | SA#98-e | SP-221245 | 0001 | 1 | B | Missing evaluations and EN resolution | 18.1.0 |