**3GPPSA4 #120-e S4-220998**

**E-meeting, 17-26 August 2022**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.0* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26**.**857** | **CR** | pseudo | **rev** | **-** | **Current version:** | **0.3.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | **[FS\_5G\_MSE] Update on 4.2** | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Tencent | | | | | | | | | |
| ***Source to TSG:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_5G\_MSE | | | | |  | ***Date:*** | | | 8/03/2022 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **B** |  | | | | | ***Release:*** | | | 18 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) Rel-12 (Release 12)* *Rel-13 (Release 13) Rel-14 (Release 14) Rel-15 (Release 15) Rel-16 (Release 16)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | This document proposes minor updates and clarifications on clause 4.4.1 | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | * Editorial improvements * Making MSE configuration API an optional feature for MSE SDKs * A table defining the MSE requirements for specification, MSE SDK abstract, MSE SDK instantiation, and MSE service * A new clause on the benefits of this approach. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  |  | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  |  | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  |  | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

CHANGE

### 4.4.1 MSE framework proposal #1

#### 4.4.1.1 Architecture

Figure 4.4.1.1-1 shows a possible framework for Media Service Enablers. The MSE framework consists of two parts: the *MSE specification* (on the left of the figure) and the *MSE implementation* (on the right).

**MSE Specification**

(Platform-dependent) **MSE SDK**

Media interfaces

Control interfaces

Configuration API

**MSE SDK abstraction**

Media interfaces

Control interfaces

Configuration API Abstraction

**MSE Service**

Media interfaces

Control interfaces

Configuration API

*Specification*

*Implementation*

MSE Description Document

Media specification

MSE Configuration API

Service API

*Platform-independent*

*Platform-dependent*

**(c)**

**(b)**

**(d)**

**(a)**

Figure 4.4.1.1-1. Media Service Enablers Framework

#### 4.4.1.2 MSE Specification

An MSE Specification (a) defines:

1. *Media aspects*

a. Functional description of the MSE including the mandatory and optional features.

b. The control interfaces such as provisioning, authentication that is used by the application, and other functions to interact with this MSE.

c. The media interfaces that includes all inputs and outputs format and protocols.

d. Network interface including system and radio network.

e. Event, notifications, reporting, and monitoring.

f. Error handling.

2. *MSE Configuration*

a. An *MSE Description Document (MDD)* that describes an implementation’s functional support in a standardised way, including:

1. Functions supported by an MSE implementation and their configuration parameters.

2. Optionally the performance/cost metrics for the different features/options.

b. An *MSE Configuration API (MCA)* abstraction for:

1. Optionally retrieving the MSE Description Document.

2. Configuring the MSE instantiation.

3. Optionally retrieving the state and status of the MSE instantiation.

c. A service API for the MSE Configuration API.

Media aspects (1) are usually covered by SA4 specifications. However, the MSE Configuration (2) is absent from current SA4 specifications and is what the MSE Specification adds. The value of this is that, for any SDK or service that is conforming to the MSE specification, a description of the features and their configuration parameters can be described using a standard document format. Furthermore, this description can be retrieved through the configuration API if supported by the implementation. Additionally, the external function or service can set a specific configuration for running that SDK. Furthermore, the state and status of the running SDK can be retrieved at any time.

The language and syntax of the MSE Description Document and the general framework of the MSE Configuration API can be defined uniformly for all SA4 Media Service Enabler specifications and only specific codepoints are defined in that specification. An external function or application understanding the MSE Description Document syntax, as well as supporting the MSE Configuration API, can retrieve the information from an MSE implementation. If it recognizes the MSE Specification identifier, it can parse and process the MSE Description Document and its configuration parameters.

An example of an MSE Description Document can be found in ISO/IEC 23090-8 [ISO-23090-8]. The function description document is a JSON document that describes the functionalities and features that a function provides as well as its configuration parameters.

#### 4.4.1.3 MSE implementation

An MSE implementation may consist of up to three aspects:

1. The MSE SDK abstraction (c), an abstract SDK definition intended to be realized as a Software Development Kit, which includes the followings:

i. Media aspects conforming to the MSE specification.

ii. MSE Description Document and MSE Configuration API.

2. The MSE SDK instantiation (d) which is an SDK implementation in a specific environment and conforms to the following:

i. Media aspects conforming to the MSE Specification.

ii. MSE Description Document and a specific implementation of the MSE Configuration API.

3. The MSE service (b) which is the MSE implementation as a service, i.e with APIs that are platform-independent (such as web-based APIs) and conforms to the following:

i. Media aspects conforming to the MSE Specification.

ii. MSE Description Document and a platform-independent implementation of the MSE Configuration API.

As shown in Figure 4.4.1.1-1, while the MSE SDK abstraction and the MSE Service are platform-independent, the MSE SDK is an instantiation of the MSE SDK abstraction for a specific platform/environment.

An MSE Specification does not required to include all three aspects. For instance, if an MSE is only intended to be realized as a software development kit, then its specification would include specifications for the SDK abstraction and one or more SDK instantiation.

Note that in the cases of MSE SDK abstract SDK (c) and MSK SDK (d), the MDD may not be retrievable through the MSE configuration APIs. In these cases, MDD is a side document, describing the features supported by the SDK.

Table 4.4.1.3-1 summarizes the above features.

Table 4.4.1.3-1. Summary of MSE features for various components of Figure 4.4.1.1-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Specification  (a) | MSE service  (b) | MSE SDK abstract  (c) | MSE SDK  (d) |
| MSE Description Document (MDD) | Describes the mandatory and optional features in a standard way. | Describes features implemented and configuration options using MDD.  This document can be retrieved using MCA. | Describes features implemented and configuration options using MDD.  This might be a side document. | Describes features implemented and configuration options using MDD.  This might be a side document. |
| *MSE Configuration API (MCA)* | The API abstraction describing how to configure the MSE. | A platform-independent API for MSE configuration. | An abstract API for MSE configuration. | An API instance for MSE configuration. |

#### 4.4.1.3 Example

As shown in figure 4.4.1.1-1, the MSE Specification can be deployed in two different ways: as an SDK for running on devices or as a microservice running on an Application Server. To demonstrate converting an existing 3GPP specification to an MSE specification, we use the 5GMS Media Session Handler defined in TS 26.501 [?], shown in figure 4.4.1.3-1.



Figure 4.4.1.3-1. Media Session Handler as defined in 26.501



Figure 4.4.1.3-2. Media Session Handler as MSE SDK abstraction, MSE SDK instantiations, and MSE service

The MSE Specification for the Media Session Handler (MSH) shown in Figure 4.4.1.4-2 describes the following:

1. Media aspects:

a. Functional description of:

i. Service Access Information.

ii. Consumption Reporting.

iii. Metrics Reporting.

iv. Dynamic policies.

v. Network Assistance.

b. M5d, M6d, M7d API definitions:

i. M5d as is already defined.

ii. M6d and M7d as abstract APIs.

iii. M6d and M7d as service APIs.

2. MSE Configuration

a. An MSE Description Document which describes:

i. An identifier that shows this MSE conforms to (1).

ii. Optional features of (1a) and (1b) with their configuration parameters.

iii. Optionally the performance/cost metrics for the different features/options.

b. Abstract API definitions for:

i. Retrieving the MSE Description Document (2a).

ii. Configuring the MSE instantiation.

iii. Retrieving the state and status of the MSE instantiation.

c. A service API for the abstract API (2b). and MSE SDK implementation of the above specification for Android should support the following:

3. Media aspects conforming to (1), including a specific implementation of the M6d and M7d service APIs.

4. The MSE Description Document (2a) and a specific implementation of the abstract APIs (2b).

The MSE Description Document describes the features implemented by the MSE. The abstract APIs allow an external Android process to retrieve this document and configure the SDK with a set of configurable parameters that are described in the MSE Description Document. They also allow it to interrogate the state and status of the running SDK.

#### 4.4.1.4 Benefits

The benefits of the above approach are the following:

1. The MSE specification defines all mandatory and optional features in a single document, the MDD, with references to the specific relevant clause(s).

2. The MSE specification also optionally defines the MSE Configuration APIs for managing and retrieving information from an implementation.

3. An implementer can use the MSE specification’s MDD as a feature checklist.

4. An implementer can use the MSE Configuration API to implement the API for MSE services.

5. The SDK instantiation of an MSE specification includes a side MDD describing the features supported by the SDK and the optional configurations it may have.

6. The MSE service instantiation of an MSE specification includes an MSE configuration API conforming to the one defined in the MSE specification that can be used for retrieving and configuring the service.

7. The MSE service instantiation provides an MDD (as a side or as part of retrieval through MSE configuration API) that provides the supported features of the MSE service instantiation.

END OF CHANGES