**3GPP TSG-SA2 Meeting #164 *S2-2407800r01***

**Maastricht, Netherlands, 19th Aug 2024 - 23rd Aug 2024**

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  | **23.288** | **CR** |  **1134** | **rev** | **-** | **Current version:** | **18.6.0** |  |
|  |
| *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 | **X** |

|  |
| --- |
|  |
| ***Title:***  | General training procedure for Vertical Federated Learning between NWDAF(s) and AF(s) |
|  |  |
| ***Source to WG:*** | China Mobile, CATT, ZTE, OPPO |
| ***Source to TSG:*** | SA2 |
|  |  |
| ***Work item code:*** | AIML\_CN |  | ***Date:*** | 2024-08-07 |
|  |  |  |  |  |
| ***Category:*** | **B** |  | ***Release:*** | Rel-19 |
|  | *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 canbe 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-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | Based on conclusions for KI#2: 5GC Support for Vertical Federated Learning in clause 8.2 of TR23.700-84, this CR aims to specify the general training procedure for vertical federated learning between AF(s) and NWDAF(s).  |
|  |  |
| ***Summary of change:*** | New clause 6.2X is added to specify the general training procedure for vertical federated learning between AF and NWDAFs. |
|  |  |
| ***Consequences if not approved:*** | VFL training procedure not specified. |
|  |  |
| ***Clauses affected:*** | 6.2X.Y.Z (new) |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

>>>>BEGINNING OF CHANGES<<<<

### 6.2X.Y.Z General training procedure for Vertical Federated Learning between NWDAF(s) and AF(s)

Editor’s Note: Whether one general procedure or two separate procedures for NWDAF/AF as server will be discussed and decided in future meeting when the procedures are stable.

#### 6.2X.Y.Z.1 NWDAF as server initiated VFL training

The figure 6.2X.Y.Z-1 below shows general training procedure for Vertical Federated Learning between NWDAF(s) and AF(s).



Figure 6.2X.Y.Z-1: General training procedure for VFL between NWDAF(s) and AF(s)

This below procedure aims for NWDAF initiated Vertical Federated Learning between NWDAF(s) and AF(s).

Editor’s Note: Further extensions are needed to show the interaction between consumer and VFL server. For example, how the consumer sends a subscription request to VFL server.

Editor’s Note: How the VFL server and client register to the network, how to discover VFL server or VFL client is FFS.

Editor’s Note: How to maintain a Vertical Federation Learning process including dynamical reselection, addition, or removal of FL Client NWDAF(s) is FFS.

Editor's Note: Further extensions are needed to show when any of the VFL participants are untrusted AF(s). In this case the procedure below will contain a NEF, and whether the service operations going via NEF is using the existing or new service operation are FFS.

1. VFL server selects the VFL clients that participate in VFL as described in the clause X.

2. To start VFL training, VFL server sends a request to the selected VFL clients. The request optionally includes initial ML model. The request also includes sample information, e.g. UE ID(s) whose corresponding data is used for the VFL model training, and VFL model correlation ID, which identifies the VFL model training process.

Editor’s Note: How to support to share the initial model from sever to client is FFS.

3. [Optional] Each VFL client collects its local data by using the current mechanism if the VFL client has not local data available already.

4. During VFL training procedure, each VFL client further trains the ML model optionally based on the initial ML Model provided by the VFL server and its own data, and reports the local ML model training information (e.g. intermediate training result) to the VFL server. The report also includes VFL model correlation ID, and intermediate training result identification information, which identifies the intermediate result in the VFL training service, e.g. time stamp.

Editor’s Note: Whether or not to support that VFL clients provide intermediate results (e.g., gradient information, loss information) to other VFL clients as instructed by the VFL server is FFS.

5. The VFL server computes the backward local ML model training information (e.g. gradient information or loss information) based on all the local ML model training information and label. The loss information may include loss function, the value of loss, the type of loss function or type of loss. The backward local ML model training information is used for updating the local ML model. Different backward local ML model training information may be computed for different VFL clients, respectively.

 The VFL server may also compute the global ML model metric (e.g. ML model accuracy) based on all the all the local ML model training information and the label.

 The VFL server may also compute contribution weights for each VFL client taking into account the local ML model metric of the VFL client, importance of the feature supported by the VFL client, number of times, intermediate results are provided by the VFL client. The contribution weights are used during inference.

6. [Optional] The VFL Server may terminate the current VFL training process. The VFL server sends VFL training termination message to VFL Client if it decides to terminate the VFL training process.

The VFL training termination decision may be made as follows:

Based on the consumer request, the VFL server sends VFL status report to update the ML model metric to the consumer.

The consumer decides whether the current model can fulfil the requirement, e.g. ML model metric is satisfactory for the consumer and determines to stop or continue the training process. The consumer continues the training process or stops the training process.

Based on the subscription request sent from the consumer, the VFL server updates or terminates the current VFL training process.

 If the VFL server received a request to stop the Federated Training process, steps 7 and 8 are skipped.

7. If the VFL procedure continues, VFL server sends the backward local ML model training information (e.g. gradient information, loss information) to the VFL clients for next round of VFL.

8. Each VFL client computes gradient of its local model and updates its local ML model based on backward local ML model training information distributed by the VFL server at step 7.

NOTE 1: The steps 4-8 should be repeated until the training termination condition (e.g. maximum number of iterations, or the result of loss function is lower than a threshold) is reached.

NOTE 2: If untrusted AF is involved in VFL Clients, the message between 5GC NF and the untrusted AF is via NEF.

NOTE 3: After VFL model training is terminated, VFL server may collect trained ML model from VFL clients. This aims to support VFL model inference with participants that have not involved in VFL model training procedure.

#### 6.2X.Y.Z.2 AF as server initiated VFL training

>>>>END OF CHANGES<<<<