**SA WG2 Meeting #152E S2-220xyz**

**17 – 26 August 2022, e-meeting**

**Source: Lenovo**

**Title: KI#8: Evaluation and conclusion**

**Document for: Approval**

**Agenda Item: 9.xx**

**Work Item / Release: FS\_eNA\_PH3 / Rel-18**

*Abstract of the contribution:*

# 1 Discussion

7 solutions for Key Issue 8 are currently agreed in the TR23.700-81

Table 1 – Solutions for Key Issue 8

|  |  |
| --- | --- |
|  | **Key Issues** |
| **Solutions** | **8** | |
| **21** | X | |
| **22** | X | |
| **23** | X | |
| **24** | X | |
| **51** | X | |
| **52** | X | |
| **53** | X | |

The following table provides an evaluation of each solution.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Solution Evaluation** | | |
| **Solutions** | **Evaluation** | **Impacts** | **Editor’s Note to be addressed** |
| **21** | **Main Principle of Solution**  NWDAF registers its capability for FL operation (client or server) in NRF.  NWDAF supporting FL server operator selects local NWDAF (supporting local FL) from the NRF. NWDAF with FL server capability aggregates local models trained by NWDAF supporting FL client capability | . NWDAF containing MTLF:  - register the "FL capability" in NF profile to NRF.  For "FL Server" NWDAF:  - support "FL Client" NWDAF discovery procedure.  - support to send FL request to "FL Client" NWDAF.  - support to aggregate the interim learning results from "FL Clients" NWDAF.  - support a new service "Nnwdaf\_MLModelTraining".  For "FL Client" NWDAF:  - support local ML training based on "FL Server" NWDAF request.  - support to report the trained interim ML model to "FL Server" NWDAF.  - support a new service "Nnwdaf\_MLModelTraining". | None |
| **22** | **Main Principle of Solution**  NWDAF registers its capability for FL operation (client or server) in NRF.  NWDAF supporting FL server operator selects local NWDAF (supporting local FL) from the NRF. NWDAF with FL server capability aggregates local models trained by NWDAF supporting FL client capability | NWDAF:  - Support creation of federated learning group with acting as NRF service consumer to discover and select NWDAFs having required capability to perform federated learning.  - Support to share ML model with other NWDAFs and, especially for Central NWDAF, aggregate ML models provided by other NWDAFs.  NRF:  - Support new parameters in NF profile provided by NWDAF.  - Support new discovery parameters related to federated learning group creation. | Editor's note: The aspects of model sharing between Central NWDAF and Local NWDAF should be aligned with in key issue #5.  Editor's note: How the central NWDAF decides to initiate an FL round i.e. trigger conditions to initiate an FL round is FFS.  Editor's note: It is FFS how the NWDAF 1 performs subsequent operations for federated learning when it contains only AnLF. |
| **23** | **Main Principle of Solution**  NWDAF registers its capability for FL operation (client or server) in NRF.  NWDAF supporting FL server operator selects local NWDAF (supporting local FL) from the NRF. NWDAF with FL server capability aggregates local models trained by NWDAF supporting FL client capability.  NWDAF determines model requires FL if:  - Data (corresponding to an Event ID) cannot be collected directly from an NF (based on local configuration)  - The request for analytics information is for a DNAI where federated learning is required.  - The request for analytics information is for a service area where federated learning is required. | - Discovery and selection of local MTLF for model training using federated learning. |  |
| **24** | **Main Principle of Solution**  NWDAF registers its capability for FL operation (client or server) in NRF.  NWDAF supporting FL server operator selects local NWDAF (supporting local FL) from the NRF. NWDAF with FL server capability aggregates local models trained by NWDAF supporting FL client capability.  NWDAF determines FL is required based on the Analytic ID. It is assumed an Analytics Id is preconfigured for a type of Federated Learning. | - Registration into NRF;  - New Data Analytics (i.e. new Analytics ID) for Federated Learning, which can be configured by operator:  - As a server NWDAF:  - Discover request to NRF with the analytics id corresponding to a federated learning  - Client NWDAF instances discovery;  - Initial Federated Learning parameters determination and distribution;  - Collection of local ML model information from the Client NWDAF instances;  - Aggregates local model information to update ML model, e.g. for gradient calculation, reporting to the Server NWDAF.  - Update the training status to Consumer (AF/NF/OAM) periodically or dynamically.  - As a Client NWDAF,  - Receive aggregated model information (updated ML model) from Server NWDAF and perform local ML model update.  NRF:  - Server NWDAF and Client NWDAF registration and discovery. | Editor's note: The type of actions that the consumer can provide is FFS. |
| **51** | **Main Principle of Solution**  NWDAF registers its capability for FL operation (client or server) in NRF.  NWDAF supporting FL server operator selects local NWDAF (supporting local FL) from the NRF. NWDAF with FL server capability aggregates local models trained by NWDAF supporting FL client capability.  Additional exchanges between Server NWDAF and Client NWDAF are proposed to allow negotiation of which clients can be part of the FL learning group. | NWDAF:  - Register FL procedure information into NRF dynamically during the Federated Learning processes.  - Selection of Client NWDAF(s).  - Monitoring Client NWDAF status updates.  - Dynamic discovery of new Client NWDAF(s).  NRF:  - Server NWDAF and Client NWDAF(s) registration for the Federated Learning procedure.  - Registers FL procedure information dynamically during the Federated Learning processes.  - Receive and notify information about Client NWDAF(s) updates to Server NWDAF. | Editor's note: It is FFS if server NWDAF registers/updates client NWDAFs info into NRF. |
| **52** | **Main Principle of Solution**  Additionally proposes the NWDAF MTLF and AnLF MTLF to exchange infomration on the accuracy of the ML model | None specified. | Editor's note: The aspects related to trained ML model(s) sharing should be aligned with Key Issue #5.  Editor's note: The Accuracy metric used depends on the conclusion of KI#1. |
| **53** | **Main Principle of Solution**  .  Solution proposes registration and discovery of multiple NWDAF (containing MTLF) with capability of Horizontal Federated Learning and also provides an horizontal dederated learning training procedure. | MTLF with FL client:  - register the "FL capability, Serving Area, Supported Time Period for FL" in NF profile to NRF.  - support local ML training based on instruction from the MTLF acting as coordinator.  - support to report the trained interim ML model to NWDAF(MTLF) acting as coordinator.  - support MTLF with FL Server discovery procedure.  MTLF with FL server acting as coordinator:  - register the "FL capability, Serving Area, Supported Time Period for FL" in NF profile to NRF.  - support MTLF with FL client discovery procedure.  - support to send FL request to MTLF with FL client.  - support to aggregate the interim ML model from MTLF with FL client. | Editor's note: Whether and how to standardize the interaction between the MTLF1 with FL server acting as coordinator and the MTLF with FL client is FFS.. |

The following are proposed by solutions addressing KI#8:

**Discovery and Selection of NWDAF MTLF supporting FL**

- NWDAF registers to NRF, FL capability as follows:

- FL aggregation (FL server) capability

- Local model training (FL client) capability

- NWDAF with FL server capability discovers NWDAF with FL client capability from the NRF.

**Determining an ML model requires FL**

- NWDAF (FL server) determines an ML model requires FL training based on the following:

- Based on the analytic ID requested by an analytics consumer. It is assumed an Analytics Id is preconfigured for a type of Federated Learning (Solution 21, 24)

Data (corresponding to an Event ID) cannot be collected directly from an NF (Solution 23, 53)

- The request for analytics information is for a DNAI where federated learning is required. (Solution 23)

- The request for analytics information is for a service area where federated learning is required. (Solution 21,23, 53)

**General Procedures for ML model training using FL**

- FL server and FL clients negotiate participation for training an ML model using FL (Solution 22, 51, 53). During negotiation FL server and FL client may exchange one or more of the following information

- Interoperability in terms of being able to run and share ML Model (Solutions 22, 51)

- Availability (Solution 51)

- Prior test to check microcomputation and communication capabilities, i.e., use same FL framework/librarires (Solution 51)

- Support of FL Horizontal capability (Solution 53)

- When the FL server initiates ML model training to FL clients the request includes:

- An indication to enable ML model training (Solution 21)

- A specific FL ID that corresponds to the Analytic ID (Solution 21)

- Analytic ID corresponding to the ML model required (Solution 22, 23, 51)

- ML model information (all solutions)

- Data type list (Solution 23, 24)

- Area of Interest (Solution 53)

- Max response time window (Solution 24)

- Some solutions propose the FL server to monitor the FL progress / status and inform the client FL

- Periodically or when a predefined accuracy level is reached (Solution 24)

- Request client FL or via NRF (Solution 51)

- MTLF with FL server capability send notification to AnLF regarding training accuracy per Model ID ( Solution 52)

As an interim conclusion it is proposed to agree the following:

**Discovery and Selection of NWDAF MTLF supporting FL**

- NWDAF registers to NRF its FL capability (FL server or FL client)

- NWDAF with FL server capability discovers NWDAF with FL client capability from the NRF.

**Determining an ML model requires FL**

- NWDAF (FL server) determines an ML model requires FL training based on the Analytic ID, Service Area/DNAI or if data cannot be collected directly from an NF.

**General Procedures for ML model training using FL**

- NWDAF (FL server) subscribes from NWDAF (FL clients) to train local models including the following parameters:

- Analytic ID correspinding to the ML model required

- ML model information

- Data type list corresponding to the local data required to locally train the ML model.

# 2 Proposal

The following is proposed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* First change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 7 Overall Evaluation

Editor's note: This clause will provide evaluation of different solutions.

Evaluation of solutions for KI#8:

The solutions provide procedures for discovery and selection of NWDAF MTLF supporting FL, determining if an ML model requires training using FL and general procedures to support model training using FL.

**Discovery and Selection of NWDAF MTLF supporting FL**

- NWDAF registers to NRF, FL capability as follows:

- FL aggregation (FL server) capability

- Local model training (FL client) capability

- NWDAF with FL server capability discovers NWDAF with FL client capability from the NRF.

**Determining an ML model requires FL**

- NWDAF (FL server) determines an ML model requires FL training based on the following:

- Based on the analytic ID requested by an analytics consumer. It is assumed an Analytics Id is preconfigured for a type of Federated Learning (Solution 21, 24)

Data (corresponding to an Event ID) cannot be collected directly from an NF (Solution 23, 53)

- The request for analytics information is for a DNAI where federated learning is required. (Solution 23)

- The request for analytics information is for a service area where federated learning is required. (Solution 21,23, 53)

**General Procedures for ML model training using FL**

- FL server and FL clients negotiate participation for training an ML model using FL (Solution 22, 51, 53). During negotiation FL server and FL client may exchange one or more of the following information

- Interoperability in terms of being able to run and share ML Model (Solutions 22, 51)

- Availability (Solution 51)

- Prior test to check microcomputation and communication capabilities, i.e., use same FL framework/librarires (Solution 51)

- Support of FL Horizontal capability (Solution 53)

- When the FL server initiates ML model training to FL clients the request includes:

- An indication to enable ML model training (Solution 21)

- A specific FL ID that corresponds to the Analytic ID (Solution 21)

- Analytic ID corresponding to the ML model required (Solution 22, 23, 51)

- ML model information (all solutions)

- Data type list (Solution 23, 24)

- Area of Interest (Solution 53)

- Max response time window (Solution 24)

- Some solutions propose the FL server to monitor the FL progress / status and inform the client FL

- Periodically or when a predefined accuracy level is reached (Solution 24)

- Request client FL or via NRF (Solution 51)

- MTLF with FL server capability send notification to AnLF regarding training accuracy per Model ID ( Solution 52)

# 8 Conclusions

Editor's note: This clause will list conclusions that have been agreed during the course of the study item activities.

Interim Conclusions for KI#8:

**Discovery and Selection of NWDAF MTLF supporting FL**

- NWDAF registers to NRF its FL capability (FL server or FL client)

- NWDAF with FL server capability discovers NWDAF with FL client capability from the NRF.

**Determining an ML model requires FL**

- NWDAF (FL server) determines an ML model requires FL training based on the Analytic ID, Service Area/DNAI or if data cannot be collected directly from an NF.

**General Procedures for ML model training using FL**

- NWDAF (FL server) subscribes from NWDAF (FL clients) to train local models including the following parameters:

- Analytic ID correspinding to the ML model required

- ML model information

- Data type list corresponding to the local data required to locally train the ML model.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*