|  |
| --- |
|  |

|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | SDS#46 |
| Title:\* | Logging for GDPR |
| Source:\* | JaeSeung Song, KETI, jssong@sejong.ac.kr  Minbyeong Lee, Hyundai Motors, [minbyeong.lee@hyundai.com](mailto:minbyeong.lee@hyundai.com) |
| Date:\* | 2020-07-07 |
| Input related to\* | WI-0095 oneM2M System Enhancement to Support Privacy Data Protection Regulations (eDPR)  TR-0062 V 0.1.0 |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Impacted other TS/TR(s) |  |
| Decision requested or recommendation:\* | Agree for inclusion in TR-0062 |
| Template Version: January 2017 (Do not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

# Introduction

This contribution provides input about key issue on logging feature for GDPR.

### -----------------------Start of change 1-------------------------------------------

# 8 Proposed Solutions

*Editor’s Note: The section provides solutions to the required functions identified in the previous section.*

## 8.x Solution: Key Issue x – Logging

In GDPR, there are several articles that the processor has to monitor activities on data for various purposes. For example, the processor shall notify the controller without undue delay after becoming aware of a personal data breach. In addition, one of the principles of GDPR is ‘integrity’. This means that the IoT platform playing as the processor should have to keep the data correct. Therefore, IoT platforms should have a logging feature at least recording the following information:

* Tracking access to IoT data: who accessed what and when. If access to data goes without proper access right, the system administrator can track all access to data and thus manifest that only the authorized personnel should be able to read the data.
* Tracking data modifications: one of the principles of GDPR is “integrity”. The IoT platform should have to keep the data correct, therefore any modification should be logged.
* Logging GDPR-specific activities: e.g. when the data subject invokes their rights.
* Logging consent: – date, time, IP address, etc. Then any consent related activities, e.g., consent withdrawal, and the history of the consent of the data subject can be logged.

In order to support logging in oneM2M system, a resource that can support following information has to be defined:

* Enable/disable logging
* What to log
* When to log
* Types of log
* Format of log

Such information can be modeled into a resource called [*logMgtRule*]. The[*logMgtRule*] resource shall be used to define log rules and events that trigger logging. The [*logMgtRule*] resource shall contain the child resource specified in table below. The [*logMgtRule*] resource shall contain the attributes specified in the table below.

Table x: Attributes of [logMgtRule] resource

| Attributes of *[logMgtRule]* | Multiplicity | RW/  RO/  WO | Description |
| --- | --- | --- | --- |
| *logStart* | 1 | RW | When to start this log record |
| *logEnd* | 1 | RW | Wned to end this log record |
| *logCriteria* | 1 | RW | This si a property to provide which information should be logged.  For example, if all the operations on the resource have to be logged, CRUDN have to be mentioned in this property. |
| *logFormat* | 0..1 (L) | RW | This is a property to provide what kinds of log information have to be stored under which format.  Default format could be <event time, Origin, operation, target resource, results>  Additionally, ip address of Origin, binding protocols, etc. can be looged.  Each item can be separated using a delimiter such as ‘;’. |
| *logLevel* | 1 | RW | Level of log information.  Example values coule be store all information, store only successful events, store only failed events. |
| *logResourceIDs* | 0..1 (L) | RW | A list of resource IDs to be logged. |
| *logStorage* | 1 | RW | A reference to a resource that actual log records are stored. |

The [*logMgtRule*] resource can be created by an IoT application managing logging. In this case, the logging application indicates target resources to be logged, as well as other properties of the [*logMgtRule*] resource. If a user wants to record log information for a specific application, the application can be created with an indication activating the logging feature. In this case, existing [*logMgtRule*] has to be referred as a referencing log management rule to be used. The following two properties can be used to indicate log indication and referecing a log management rule:

* *logIndication*: This is a property to indicate a resource with this property is a subject for system log
* *eventLogID:* Which Log rules will be followed. A URI of referencing <logMgtRule> resource has to be added

The following figure shows the high-level resource structure of the proposed logging mechanism.



Figure x. Logging resource structure

The following figure shows procedures that a logging application creates a logging management rule to oneM2M platform and an application uses the created logging rule to record any activities on it.



Figure x2. Procedure showing how log information can be created and started

* Step 1-2:   
  A log management resource *< logMgtRule1>* was created at the IN-CSE by a log management application. Note that the *< logMgtRule >* can also be provisioned. When this resource is created, the application also have to create a resource to store actual log records. In this case, the <logStorage1> is created.   
  For this specific use case, the *< logMgtRule1>* can be set as following: The logStorage attribute refers the address of < *logMgtRule*1> resource. The logFormat attribute is configured time;originator;operation;taget;status. The logResourceIDs attribute is configured AE#1. The logLevel attribute is configured to all request messages. The logCriteria attribute is configured CRUDN.
* Step 3-5:   
  In this example, human body sensor application creates AE1 to IN-CSE with logIndication. The application also refer < *logMgtRule*1> as the logging rule to follow. The IN-CSE then add AE#1 to the logResourceID of < *logMgtRule*1> to start log for AE#1.
* Step 6-7:   
  AE2 tries to read AE#1 resource to show the value to its user. When IN-CSE receives such request, it performs the operation. Then IN-CSE checks whether this message has to be recorded into its log resources. If AE#1 is subject to be loged, IN-CSE takes necessary information, which entity tries to read, when this message was received, which binding is used, what was the result of the request and stores the collected information to a proper resource. In this case, <logStorage1>/<AE#1> is the place to record the processed request.

### -----------------------End of change 1-------------------------------------------