|  |  |
| --- | --- |
| Input contribution  Use case | |
| Use Case Title:\* | Semantic discovery in presence of a ``network`` of  M2M service providers (M2MSP) |
| Group Name:\* | RDM |
| Source:\* | INRIA (on behalf of ETSI STF 589) |
| Contact: | INRIA  Luigi Liquori, [Luigi.Liquori@inria.fr](mailto:Luigi.Liquori@inria.fr)  Marie-Agnès Peraldi-Frati |
| Date:\* | 2020-03-31 |
| Abstract:\* | This use case could be considered as either the “use-case zero”, or a “parametric use-case” because it is suitable to be **instantiated in many concrete cases**. It shows the importance of fixing, formalizing and extending:  a) *Formal Graph Topologies* to capture most common scenarios involving networks of M2MSP;  b) *Formal Semantic Discovery Routing Mechanism* (SDRM) to route a Semantic Query between M2MSP with exhaustivity/non exhaustivity constraints and iterative vs. recursive routing modality;  c) *Formal Semantic Discovery Query Language* (SDQL) to express a large type of queries;  d) *Formal Semantic Resolution Query Mechanism* (SRQM) necessary to reduce locally a complex query into a number of simpler queries. |
| Agenda Item:\* |  |
| Work item(s): | See proposed WI in TP-2020-0040 |
| Document(s)  Impacted\* | TR 001 |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Decision requested or recommendation:\* | Include in TR 001 |
| 'Template Version: January 2019 (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.

# Title

Semantic discovery in presence of as \*network\* of M2M service providers (M2MSP).

## Description

This use case could be considered as either the “use-case zero”, or a “parametric use-case” because it is suitable to be instantiated in many concrete cases. It shows the importance of fixing, formalizing and extending:

a) *Formal Graph Topologies* to capture most common scenarios involving networks of M2MSP;

b) *Formal Semantic Discovery Routing Mechanism* (SDRM) to route a Semantic Query between M2MSP with exhaustivity/non exhaustivity constraints and iterative vs. recursive routing modality;

c) *Formal Semantic Discovery Query Language* (SDQL) to express a large type of queries;

d) *Formal Semantic Resolution Query Mechanism* (SRQM) necessary to reduce locally a complex query into a number of simpler queries.

**Note that, AFAIK, points a, b, c, and d, are not actually part of the oneM2M standard.**

## Source

ETSI TR 103 714: “SmartM2M; Study for oneM2M Discovery and Query use cases and requirements”

## Actors

1. 5 M2M Devices (M2MD), also known as Application Entities (**AE**) X of type T1, Y of type T2, Z of type T3, V of type T4, and W of type T5.
2. 2 M2M Application Providers (M2MAP), also known as **Middle Node Common Service Entities** (MN-CSE) P, and Q.

A M2MSP is a service provider having a local database containing information on their M2MD. The local database include location information (where each device is currently located), the device type, etc. We let P and Q have some “Contractual Regulatory and Security Relationship” (CRSR) with A.

1. 4 M2M Service Providers (M2MSP), also known as Infrastructure Node Common Service Entities (IN-CSE) A, B, C, and D.

A M2MSP is a service provider having a database containing information on their M2MD. The local database include location information (where each device is currently located), the device type, etc. We let A,B,C, and D have CRSR each other’s.

In an hypothetical analogy with the Border Gateway Protocol 4 (BGP4, <https://tools.ietf.org/html/rfc4271>), we could envisage (TBD in T2) 3 kinds of CRSR between M2MSP, namely:

CRSR IN {CUSTOMER, PROVIDER, PEER}.

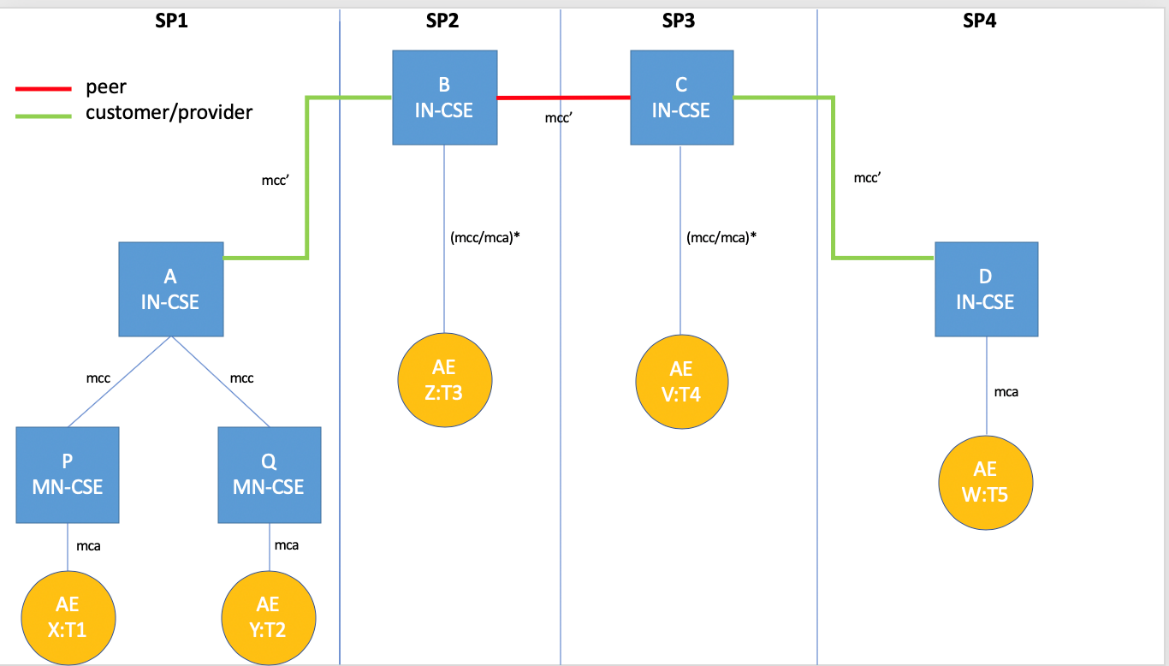
The three CRSR can intuitively be explained as follows:

1. CSE1 **PEER** CSE2: CSE1 and CSE2 mutually shares infrastructure, MN-CSE, and AE for free.
2. CSE1 **PROVIDER** CSE2: CSE1 offers on a contractual basis infrastructure, MN-CSE, and AE (for example CSE2 pays a monthly bill to CSE1 on the basis of the usage of the infrastructure, MN-CSE, and AE of CSE1, and shares security policies of CSE1).

CSE1 **CUSTOMER** of CSE2: CSE1 takes advantage of the infrastructure, MN-CSE, and AE of CSE1 and offers to the latter the usage of MN-CSE and AE registered in its domain

## Pre-conditions

We consider the following topology:



## Triggers

Using a suitable SEMANTIC DISCOVERY QUERY LANGUAGE (SDQL) X:T1 send a semantic discovery query service request SDREQ to MN-CSE P

**SDREQ1 = ?T2|FC2 AND ?T3|FC3 AND ?T4|FC4 AND ?T5|FC4**

That can be intuitively read as follows:

some AE of type T2 registered in ANY (\*) CSE satisfying the filter criteria FC2, AND

some AE of type T3 registered in ANY (\*) CSE satisfying the filter criteria FC3, AND

some AE of type T4 registered in ANY (\*) CSE satisfying the filter criteria FC4, AND

some AE of type T5 registered in ANY (\*) CSE satisfying the filter criteria FC5

We could add in the SDQL also an OR nonterminal and consider queries such as

some AE of type T2 registered in ANY (\*) CSE satisfying the filter criteria FC2, OR

some AE of type T3 registered in ANY (\*) CSE satisfying the filter criteria FC3, OR

some AE of type T4 registered in ANY (\*) CSE satisfying the filter criteria FC4, OR

some AE of type T5 registered in ANY (\*) CSE satisfying the filter criteria FC5

In addition to ANY-prefix we could also consider other restricted semantic routing directives such as

CURRENT = search in the local database

CUSTOMER[N] = search in the databases of N CUSTOMER CSE

PROVIDER[N] = search in the databases of N PROVIDER CSE

PEER[N] = search start on the databases of N PEER CSE

We could add in the SDQL also an OR nonterminal and consider semantic queries in SDQL of the following shape:

SDREQ1’ = (?T2|FC2 OR ?T3|FC3) AND (?T4|FC4 OR ?T5|FC4)]

SDREQ1’’ = (?T2|FC2 AND ?T3|FC3) OR (?T4|FC4 AND ?T5|FC4)]

We could add in the SDQL also a NOT nonterminal and consider semantic queries in SDQL of the following shape:

SDREQ1’’’ = (?T2|FC2 AND ?T3|FC3) OR (?T4|FC4 AND (NOT (?T5|FC4)) ]

Summarizing, the shape of a logical query could be a restricted kind of either a Conjunctive Normal Form (CNF) or a Disjunctive Normal Form (DNF). A CNF/DNF will be translated into a set of simpler unitary Queries by the CSE receiving the query using the SRQM. Complexity of SRQM can be an issue and should be discussed.

## Normal Flow

A possible “trace” of the semantic discovery routing mechanism, inspired to [1,2] proceeds as follows (the real routing mechanism will be described and formally specified in T2 and simulated in T3):

* X sends a semantic discovery request (SDREQ1) to P via an mca-pointer
* P verifies the integrity of SDREQ1 and forward a semantic discovery request SDREQ to A via an mcc-pointer that starts the semantic routing into the network of CSE.
* SDREQ1 is resolved locally in A into four subqueries, namely SDREQ2, SDREQ3, SDREQ4, and SDREQ5, where:

SDREQ2= ?T2|FC2 AND

SDREQ3= ?T3|FC3 AND

SDREQ4= ?T4|FC4 AND

SDREQ5= ?T5|FC5

1. A start lookups in its local database, solving in {SDREQ2,3,4,5} and fail
2. A down-forwards SQREQ to Q via an mcc pointer
3. Q solve the subquery SQREQ2 ?T2|FC2 in its local database returning Y to A
4. A send back Y to P and X
5. A up-forwards SQREQ3 & SQREQ4 & SQREQ5 to B via an mcc’ pointer
6. B solve the SQREQ3 ?T3|FC3 in its local database returning Z to A (and back to P and X)
7. B side-forwards SQREQ4 & SQREQ5 to C via an mcc’ pointer
8. C solve the SQREQ4 ?T4|FC2 in its local database returning V to B (and back to A, P and X)
9. C down-forwards SQREQ5 to D via an mcc’ pointer
10. D solve the SQREQ5 ?T5|FC5 in its local database returning W to C (and back to B, A, P and X)

**Note 1**. The presented semantic routing trace is “recursive” i.e. A waits for B, which waits for C, which waits for D.

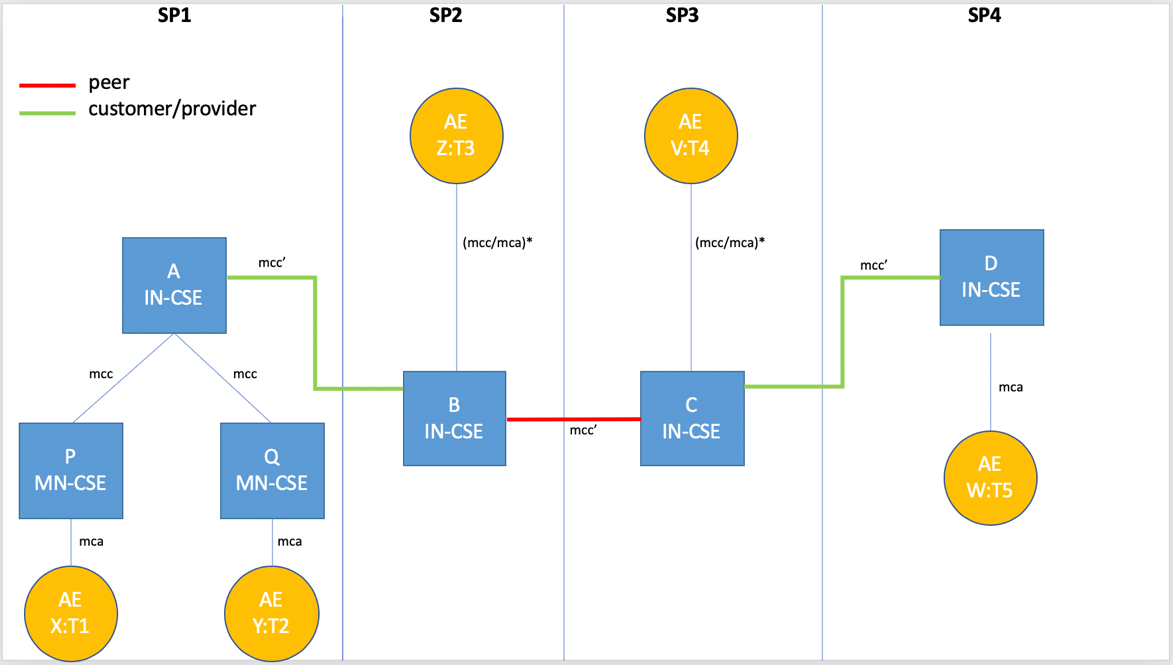
**Note 2.** We can also discriminate if routing is:

1. “Exhaustive”. As example, I am looking for a semantic resource that there exists for sure, and I will explore all the distributed data bases until I will found it!
2. “Non-exhaustive”. As example, I am looking for a semantic resource that there exists for sure, and I will explore the data bases space until I reach a fixed number N of hops!

**Note 3.** When A forwards to B, it follows that A pays a bill to B. When B forwards to C, it follows that B and C have a common agreement and so do not bill each other. When C forwards to D, it follows that D pays a bill to C. In summary, B and C are rewarded for their “routing job”.

## Alternative flow

We consider the following alternative topology:



A possible “trace” of the semantic discovery routing mechanism, inspired to [1,2] proceeds as in Subsection 1.6, excepting for a the following caveat.

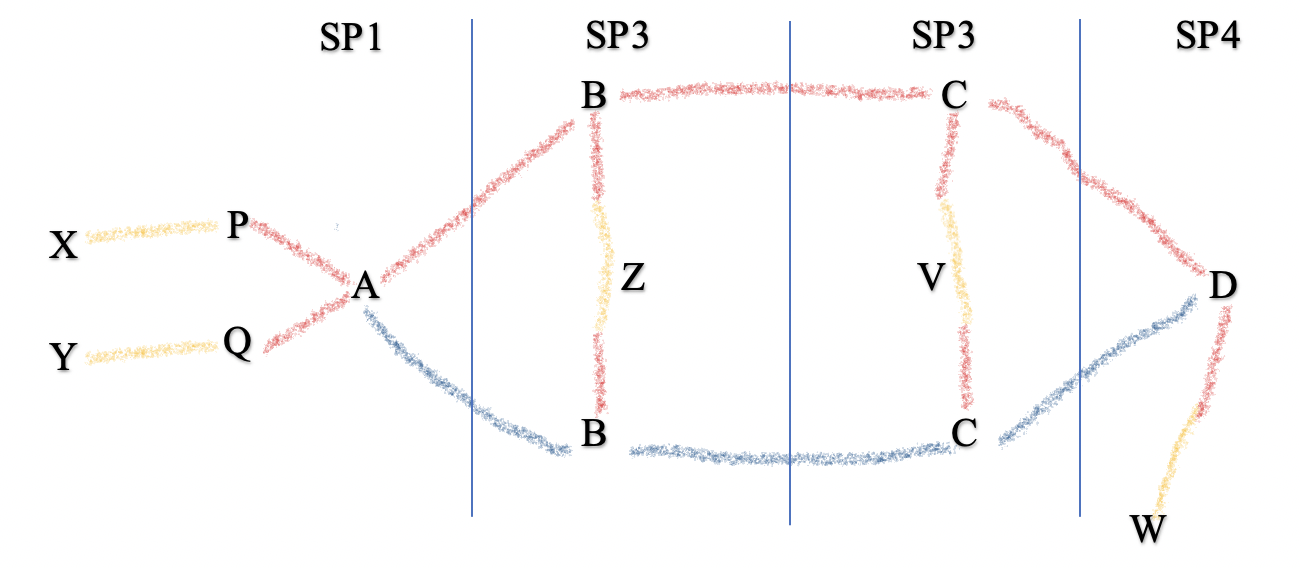
**Caveat**. When A forwards to B, it follows that B pays a bill to A (*I do a favour and I have also to pay for it…*). When B forwards to C, it follows that B and C have a common agreement and, as such, they not charge it each other. When C forwards to D, it follows that C pays a bill to D (*Again, I do a favour and I have also to pay for it…*). In summary, B and C does a job for their providers and, moreover, they have to pay for their “routing job”…

Without standardization alternative traces can happen in practice: because of the distributed nature of the semantic discovery routing, we should try to avoid it. This unfortunate situation **is not new in Internet** and it is referred as VALLEY ROUTING [1]. “Good routing” should guarantee that routing is always “valley preserving” (or “no valley”). Valley routing property is also preserved in the network aware resource discovery protocol of L. *et al*. [2].

## Post-conditions

**X can start to interact with Y, Z, V, and W.**

## High Level Illustrations



## Potential requirements

The following requirements extend the oneM2M requirements TS-0002-V4.6.0.

1. The M2M system SHALL provide a
   1. SEMANTIC DISCOVERY QUERY LANGUAGE (SDQL) equipped with a formal syntax and a
   2. SEMANTIC DISCOVER RESOLUTION QUERY MECHANISM (SDQM) to translate a complex query into a set of unitary ones.
2. The M2M system SHALL provide some TOPOLOGY DIRECTIVES to organize M2MSP.
3. The M2M system SHALL provide mechanisms for CONTRACTUAL RELATIONS between M2MSP.
4. The M2M system SHALL provide a SEMANTIC DISCOVERY ROUTING MECHANISM (SDRM) that can be RECURSIVE or ITERATIVE (TBD in T2) and EXAUSTIVE or NON-EXAUSTIVE (TBD in T2) to route queries from source X (resp. P) to destination Y (resp. Q).
5. The M2M system SHALL provide NETWORK and BRAND neutrality.
6. The M2M systems SHALL provide SECURITY MECHANISMS well adapted to the distributed nature of Semantic Network Discovery.

**References:**

[1] Lixin Gao: On inferring autonomous system relationships in the internet. IEEE/ACM Trans. Netw. 9(6): 733-745 (2001) <https://dl.acm.org/doi/10.1109/90.974527>).

[2] Luigi Liquori, Rossano Gaeta, Matteo Sereno. A Network Aware Resource Discovery Service. EPEW 2019 - 16th European Performance Engineering Workshop, Nov 2019, Milano, Italy, Lecture Notes in Computer Science, Springer Verlag, to appear, 2019.

[3] oneM2M. Use\_Cases\_Collection-V4\_3\_0. See Ex 12.9. Semantics query for device discovery across M2M Service Providers.

[4] oneM2M. REQ-2014-0005R01-Semantics\_query\_for\_device\_discovery\_on\_Inter-M2M\_SP.