Distributed Mobility Management (DMM) WG

DMM Work Item:
Forwarding Path & Signaling Management (FPSM)

draft-ietf-dmm-fpc-cpdp-03.txt

M. Liebsch, S. Matsushima, S. Gundavelli, D. Moses, L. Bertz

IETF96, Berlin

2016-07-21

What is this work about..?

❑ Enable the separation of a mobility gateway's Control-Plane function from its Data-Plane function

❑ Enable distributed deployment of Control- and Data-Plane functions

❑ Functional architecture

❑ Operational aspects and data model which apply in between the functional elements

Mobility Control-Plane

(API)

FPC Client

FPC Agent

DPN Configuration API
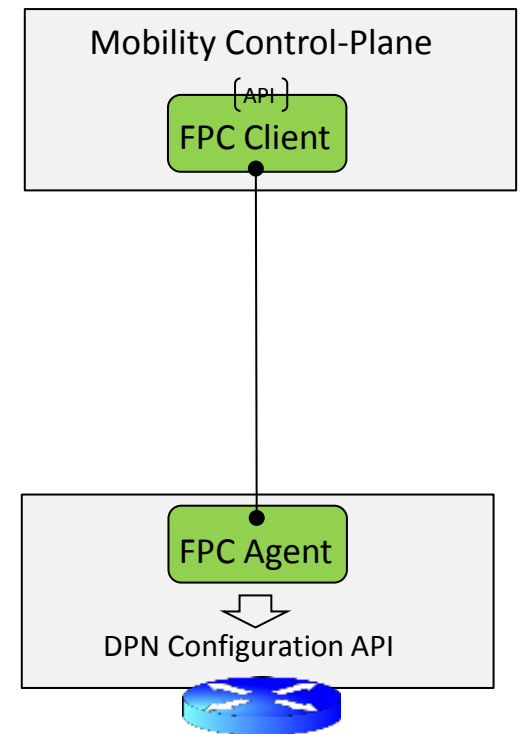
Progress since last IETF95 meeting

❑ No update of draft version 03 ..

❑ Draft did not look sound having two operational modes and different data models specified

❑ Some features, such as session representation, not well captured

❑ First ideas of a common data model came up at IETF95

❑ Team made progress in the design of a common data model
   ❑ Evolution of version 3 data model
   ❑ Enables different levels of abstraction in between Data- and Control-Plane
   ❑ Enables great level of implementation- and operational flexibility

Status at last meeting per draft version 03

❑ Support of two operational modes (Mode I and Mode II)

❑ YANG models and tree as well as operational specification for Mode I

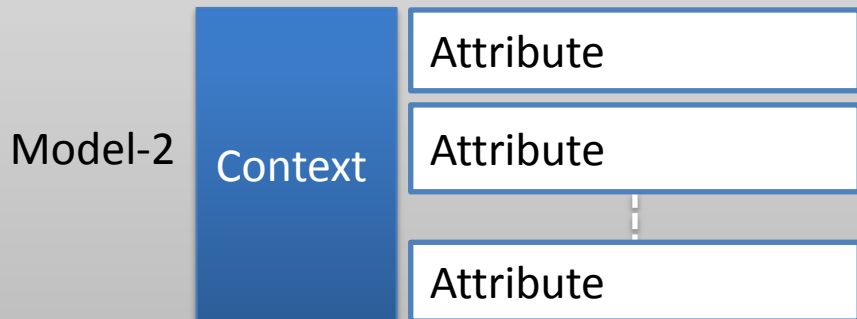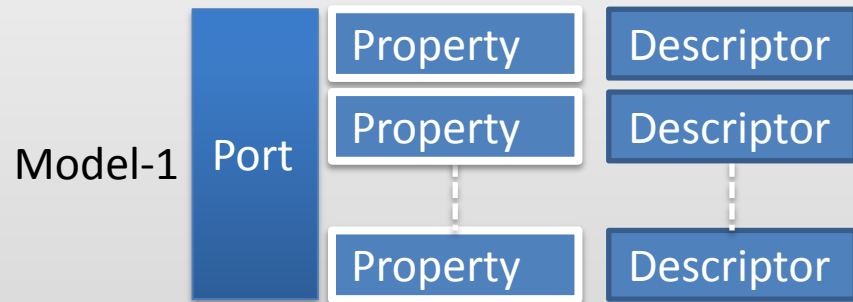   ❑ Rules semantic based on ports, traffic selectors and traffic treatment actions (properties)
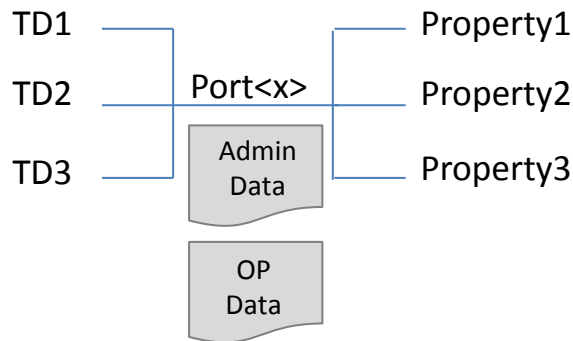
❑ Different operation and no data model for Mode II
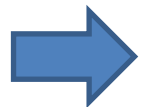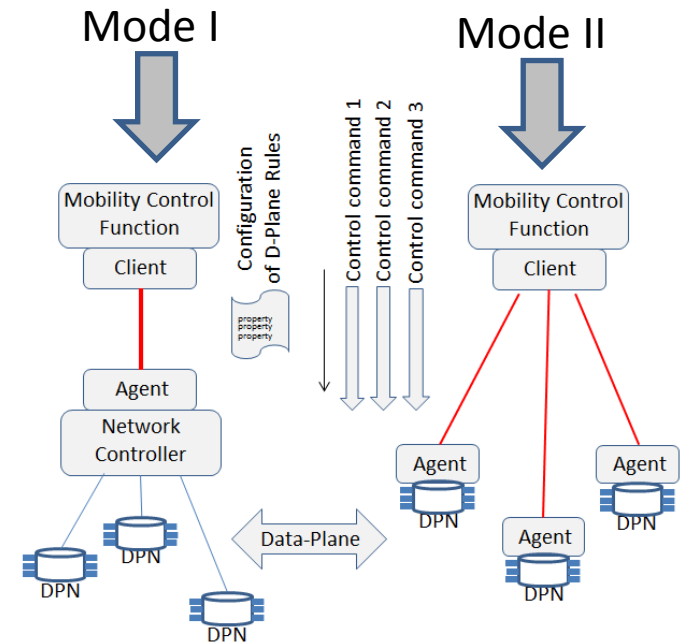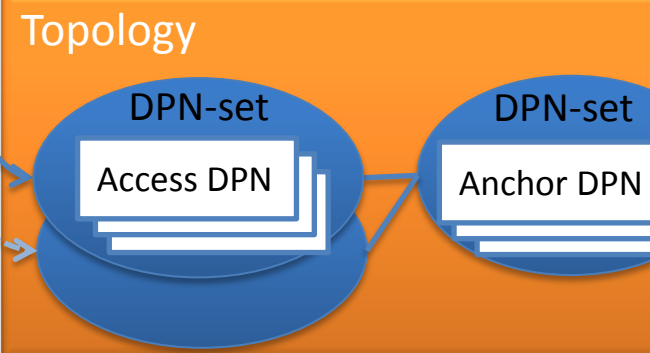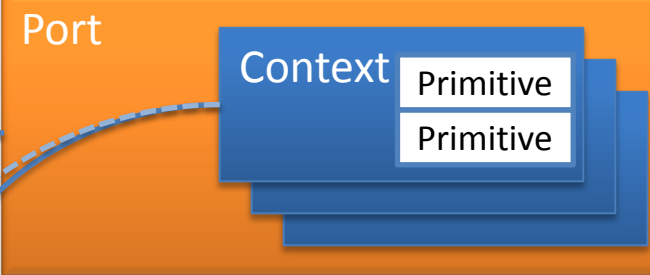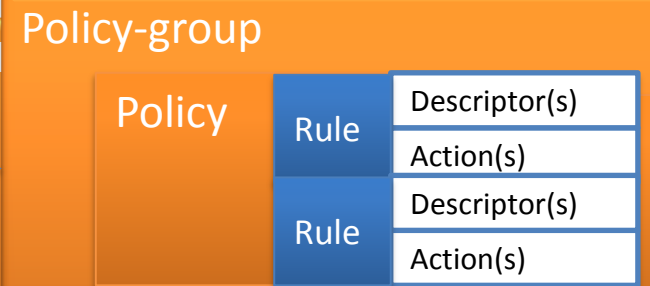

Recognized the need to harmonize data model and operation in the view of both modes

Concept for the evolved data model

For your reference

Version 03 – Figure 15 – Single Call Example

```
                              +-----------+
+------+ +------+      +-----+ |  FPC      |       | FPC    | | Anchor  |
|MAG-C1| |MAG-C2|      |LMA-C| | Client|           | Agent  | |   DPN   |
+------+ +------+      +-----+------+                +-------+ +---------+
[MN attach]  |                |                        |          |
  |------------PBU----->|                              |          |
  |        |            |----(1)-PRT_ADD----------->|             |
  |        |            |       [PRT_ID,PROP_TUN,    |--tun1 up->|
  |<-----------PBA------|              PROP_QOS,      |--tc qos-->|
  |        |            |        TS_CONTAINER(HNP)]  |-route add>|
  | [Edge]-===================================================-|
  | [DPN1|  |            |                            |          |
  |        |            |                             |          |
  |    [MN handover]    |                             |          |
  |        |---PBU ---->|                             |          |
  |        |            |---------PROP_MOD--------->|             |
  |        |<--PBA------|     [PRT_ID,PROP_TUN]      |-tun1 mod->|
  |        |            |                            |           |
  |        | [Edge]-===================================================-|
  |        | [DPN2]     |                            |           |
```

JSON Representation (Examples)

Example 1 – Does not share with others
```
{
 "context-id": "1",
 "delegating-ip-prefixes" : [ <hnp1> ],
  DL Tunnel Information,
  QoS Information
}
```

This is similar to V03 Figure 15's Information representation but w/o IDs.

Example of RestCONF JSON based on model – NOT final!

Example 2 – Context Sharing (Single/Multi Transactions)
```
[ {
 "context-id": "1",
 "delegating-ip-prefixes" : [ <ipaddress1> ],
  DL Tunnel Information,
  Base Charging-Information,
  Other Charging-Information
},
{
  "context-id": "2",
  "parent-context": "1",
  DL Tunnel Information,
  Other Charging-Information
} ]
```

Here, information required by the Context (determined by technology type) is implicitly shared, I.e. delegating-ip-prefixes & *Base Charging-Information* are inherited via the parent-context.

JSON Representation – Common Data

Example – Sharable Data
(Single/Multi Transactions)
```
{
 "context-id": "1",
 "delegating-ip-prefixes" : [ <ipaddress1> ],
   Tunnel Information,
 "ports": [ "1" ]
}
[ { "port-id": "1",
   "policy-groups" : [ "1" ] } ],
[ {
      "policy-id": "1",
      "rules": [
           "order" : "1",
           "actions": [
                { "action-id": "1",
                  "action-type": "qos",
                  QoS Information... },
                Other Common Data
           ]
      ]
  } ]
```

```
{
 "context-id": "2",
 "delegating-ip-prefixes" : [ <ipaddress2> ],
 "ports": [ "1" ]
}
```

Example of RESTConf JSON based on model – NOT final!

Structure

Version 03

- Ports

- Properties

- Descriptors / Forwarding Rules

Converged Model

- Now Called Rules

- Context Properties or Actions (as appropriate)

New Concepts

- Contexts - Hierarchy with fate & attribute sharing

- Ports (formerly non-virtual form of ports) – Contains Policy Groups

Structures Used for

- Policy Groups* – Explicit Groups of Policies

- Policies* – A Group of Rules (multi-valued form of the QoS-Template & Descriptors in the V03 Config Tree)

- Rule* – Descriptor(s) + Action(s)

Next Steps

❑ Description of new data model and operational aspects

❑ Define and describe updated YANG data model and RPCs

❑ Adopt suitable structure in draft revision 4

❑ Plan for an early update to solicit feedback (September)