

# LISP YANG model

## (-05 update)

IETF100, Singapore

# NMDA guidelines

- Network Management Datastore Architecture (NMDA) guidelines are documented in *draft-dsdt-nmda-guidelines*
- Addresses “OpState problem” which has been discussed at IETF: duplication of nodes in separate configuration and operational containers
- New <operational> datastore for operational data added in *draft-ietf-netmod-revised-datastores*. Contains data/state and configuration which is in use by the system
- Metadata annotation indicates the origin of the data in <operational> datastore. E.g. intended or learned.

# LISP YANG updates for NMDA guidelines

- Removed duplicate containers across the models
- Single map-cache on (P)ITR (for both static and dynamic)
- Single mapping database on MS (for both static and dynamic)
- Removed references to “state/config/cfg/etc” from descriptions

# Site-ID/xTR-ID

- Defined 'xtr-id-type'
  - type binary {length "16";}
- Site-ID and xTR-ID added to 'ietf-lisp'
- For role = itr, etr, pitr or petr:
  - Only one xTR-ID/Site-ID per LISP router-instance across all its LISP roles
- For role = ms or mr:
  - Handles multiple sites
  - Matches site-id in registrations/requests to its configuration

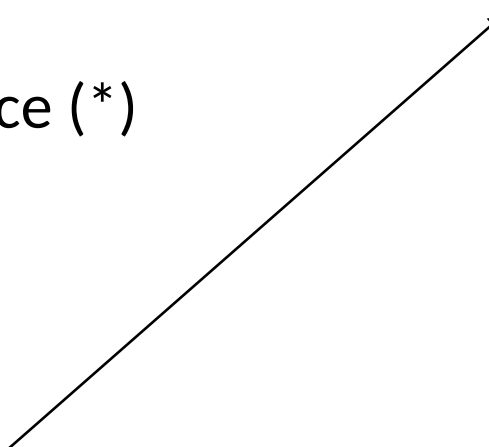
# Map Server

- Authentication key now per Site-ID
- Site-ID/xTR-ID added to mapping records
- Removed 'registered-mappings'
- VNI is now the parent for mappings

# (P)xTR merged models

- PITR and PETR models removed
  - Merged with ITR and ETR
  - Using 'when' statement for containers only relevant for ITR/ETR
  - 'presence' statement on 'itr' container enables both ITR and PITR operation
    - Same with ETR

# LISP hierarchy

- lisp-router (1)
    - lisp-router-instance (\*)
      - (p)etr (1)
      - (p)itr (1)
      - ddt-node
      - map-resolver (1)
      - map-server (1)
  - map-server (1)
    - Site-ID (\*)
      - Auth key (1)
    - VNI (\*)
      - Mapping (\*)
        - EID (1)
        - Site-ID/xTR-ID (1)
- 

# Minor

- Added creation timestamp to mappings
- PeTR per ITR
  - Not per mapping within the ITR
- Version updated to 2017-07-01
  - Except for lisp-address-types.yang



# Next Steps

- More operational data needed, e.g. counters?
- Compare with LISP MIB
- Comments/feedback from LISP WG

# Module ietf-lisp

```
module: ietf-lisp
+--rw lisp
+--rw locator-sets
| +--rw locator-set* [locator-set-name]
| | +--rw locator-set-name string
| | +--rw (locator-type)?
| | | +--:(local-interface)
| | | | +--rw interface* [interface-ref]
| | | | | +--rw interface-ref if:interface-ref
| | | | | +--rw priority? uint8
| | | | | +--rw weight? uint8
| | | | | +--rw multicast-priority? uint8
| | | | | +--rw multicast-weight? uint8
| | | +--:(general-locator)
| | | | +--rw locator* [id]
| | | | | +--rw id string
| | | | | +--rw locator-address
| | | | | | +--rw address-type lisp-address-family-ref
| | | | | | +--rw virtual-network-id? instance-id-type
| | | | | | +--rw (address)?
| | | | | +--rw priority? uint8
| | | | | +--rw weight? uint8
| | | | | +--rw multicast-priority? uint8
| | | | | +--rw multicast-weight? uint8
+--rw lisp-router-instances
+--rw lisp-router-instance* [lisp-router-instance-id]
+--rw lisp-router-instance-id int32
+--rw lisp-role* [lisp-role-type]
| +--rw lisp-role-type lisp-role-ref
+--rw lisp-router-id
+--rw site-id? uint64
+--rw xtr-id? lisp:xtr-id-type
```

# Module ietf-lisp-itr

```
module: ietf-lisp-itr
augment /lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance:
  +--rw itr!
    +--rw rloc-probing!
      | +--rw interval?          uint16
      | +--rw retries?          uint8
      | +--rw retries-interval? uint16
    +--rw itr-rlocs? -> /lisp:lisp/locator-sets/locator-set/locator-set-name
    +--rw map-resolvers
      | +--rw map-resolver*      inet:ip-address
    +--rw proxy-etr
      | +--rw proxy-etr-address* inet:ip-address
    +--rw map-cache
      +--rw virtual-network* [vni]
        +--rw vni                lcaf:instance-id-type
        +--rw mappings
          +--rw mapping* [id]
            +--rw id              eid-id
            +--rw eid
              | +--rw address-type          lisp-address-family-ref
              | +--rw virtual-network-id?   instance-id-type
              | +--rw (address)?
            +--rw time-to-live?          uint32
            +--rw creation-time?         yang:date-and-time
            +--rw authoritative?        bits
            +--rw static?                boolean
            +--rw (locator-list)?
              +--:(negative-mapping)
                | +--rw map-reply-action?  map-reply-action
              +--:(positive-mapping)
                +--rw rlocs
                  +--rw locator* [id]
                    +--rw id              string
                    +--rw locator-address
                      | +--rw address-type          lisp-address-family-ref
                      | +--rw virtual-network-id?   instance-id-type
                      | +--rw (address)?
                    +--rw priority?        uint8
                    +--rw weight?          uint8
                    +--rw multicast-priority? uint8
                    +--rw multicast-weight? uint8
```

# Module ietf-lisp-etr

```
module: ietf-lisp-etr
augment /lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance:
  +--rw etr!
    +--rw map-servers
      | +--rw map-server* [ms-address]
      |   +--rw ms-address      inet:ip-address
      |   +--rw auth-key?      string
      |   +--rw auth-key-type?  lisp:auth-key-type
    +--rw local-eids
      +--rw virtual-network* [vni]
        +--rw vni      lcaf:instance-id-type
        +--rw eids
          +--rw local-eid* [id]
            +--rw id          lisp:eid-id
            +--rw eid-address
              | +--rw address-type      lisp-address-family-ref
              | +--rw virtual-network-id?  instance-id-type
              | +--rw (address)?
            +--rw rlocs? -> /lisp:lisp/locator-sets/locator-set/locator-set-name
            +--rw record-ttl?          uint32
            +--rw want-map-notify?     boolean
            +--rw proxy-reply?         boolean
            +--rw registration-interval? uint16
```

# Module ietf-lisp-mapserver

```
module: ietf-lisp-mapserver
augment /lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance:
  +--rw map-server!
  +--rw sites
  | +--rw site* [site-id]
  | | +--rw site-id      uint64
  | | +--rw auth-key
  | | | +--rw auth-key-value?  string
  | | | +--rw auth-key-type*   lisp:auth-key-type
  +--rw virtual-network-ids
  | +--rw virtual-network-identifier* [vni]
  | | +--rw vni          lcaf:instance-id-type
  | | +--rw mappings
  | | | +--rw mapping* [eid-id]
  | | | | +--rw eid-id          lisp:eid-id
  | | | | +--rw eid-address
  | | | | | +--rw address-type      lisp-address-family-ref
  | | | | | +--rw virtual-network-id?  instance-id-type
  | | | | | +--rw (address)?
  | | | | +--rw site-id*          uint64
  | | | | +--rw more-specifics-accepted?  boolean
  | | | | +--rw mapping-expiration-timeout?  int16
  | | | | +--rw mapping-records
  | | | | | +--rw mapping-record* [xtr-id]
  | | | | | | +--rw xtr-id          lisp:xtr-id-type
  | | | | | | +--rw site-id?      uint64
  | | | | | | +--rw eid
  | | | | | | | +--rw address-type      lisp-address-family-ref
  | | | | | | | +--rw virtual-network-id?  instance-id-type
  | | | | | | | +--rw (address)?
  | | | | | +--rw time-to-live?      uint32
  | | | | | +--rw creation-time?     yang:date-and-time
  | | | | | +--rw authoritative?    bits
  | | | | | +--rw static?           boolean
  | | | | | +--rw (locator-list)?
  | | | | | | +--:(negative-mapping)
  | | | | | | | +--rw map-reply-action?  map-reply-action
  | | | | | | +--:(positive-mapping)
  | | | | | | +--rw rlocs
  | | | | | | | +--rw locator* [id] unter32
```

# Module ietf-lisp-mapserver (contd)

```
module: ietf-lisp-mapserver
  augment /lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance:
    +--rw map-server!
  <cut>
    |
    |
    |         +--rw rlocs
    |         |
    |         |         +--rw locator* [id]
    |         |         |
    |         |         |         +--rw id                string
    |         |         |         +--rw locator-address
    |         |         |         | +--rw address-type      lisp-address-family-ref
    |         |         |         | +--rw virtual-network-id? instance-id-type
    |         |         |         | +--rw (address)?
    |         |         |         +--rw priority?          uint8
    |         |         |         +--rw weight?            uint8
    |         |         |         +--rw multicast-priority? uint8
    |         |         |         +--rw multicast-weight?   uint8
    |         |         +--ro counters
    |         |         | +--ro map-registers-in?          yang:counter32
    |         |         | +--ro map-registers-in-auth-failed? yang:counter32
    |         |         | +--ro map-notify-records-out?     yang:counter32
    |         |         | +--ro proxy-reply-records-out?    yang:counter32
    |         |         | +--ro map-requests-forwarded-out? yang:counter32
    |         +--rw mapping-system-type? lisp:mapping-system-ref
    |         +--ro summary
    |         | +--ro number-configured-sites? uint32
    |         | +--ro number-registered-sites? uint32
    |         | +--ro af-datum
    |         | | +--ro af-data* [address-type]
    |         | | | +--ro address-type      lcaf:lisp-address-family-ref
    |         | | | +--ro number-configured-eids? uint32
    |         | | | +--ro number-registered-eids? uint32
    |         +--ro counters
    |         | +--ro map-registers-in?          yang:counter32
    |         | +--ro map-registers-in-auth-failed? yang:counter32
    |         | +--ro map-notify-records-out?     yang:counter32
    |         | +--ro proxy-reply-records-out?    yang:counter32
    |         | +--ro map-requests-forwarded-out? yang:counter32
```

# Module ietf-lisp-mapresolver

```
module: ietf-lisp-mapresolver
augment /lisp:lisp/lisp:lisp-router-instances/lisp:lisp-router-instance:
  +--rw map-resolver!
    +--rw mapping-system-type?    lisp:mapping-system-ref
    +--rw ms-address?             inet:ip-address
```