Data Model

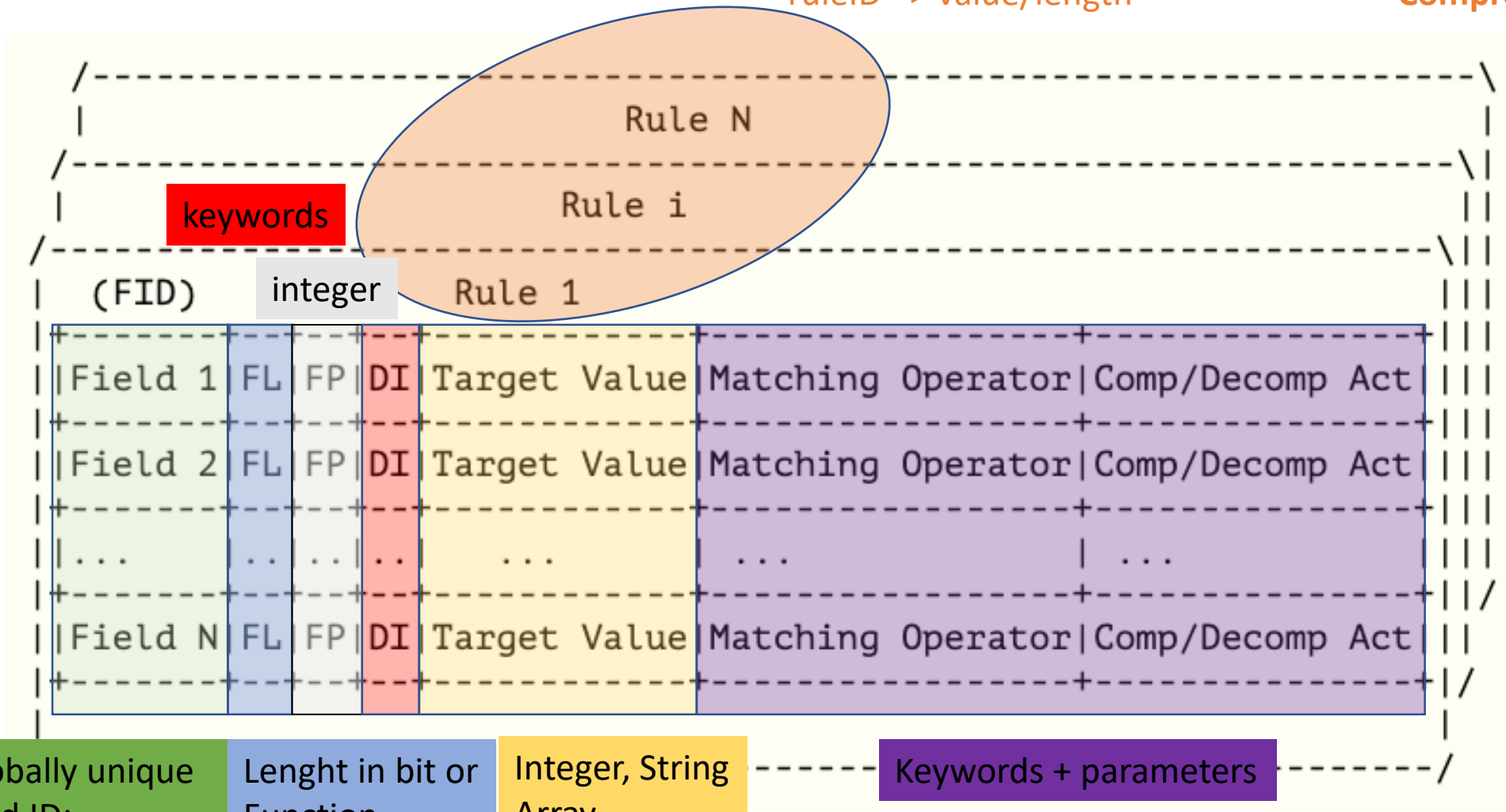
Why a data model

- draft-ietf-lpwan-ipv6-static-context-hc-18 focuses on compression and fragmentation mechanisms.
  - Rules are defined in an “abstract” manner
- Data model to represent the rules:
  - Cover draft-ietf-lpwan-ipv6-static-context-hc-18 and coap
    - Compression
    - Fragmentation
  - Easily extendable:
    - New fields, new MO, new CDA,…
- Yang and COREConf:
  - Study the impact on CBOR of Yang choices.

Fragmentation

ruleID => value/length

Compression



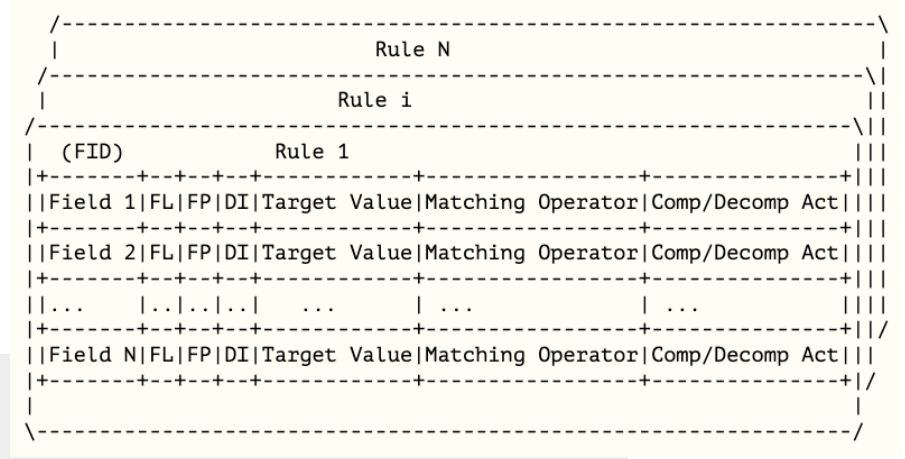
Globally unique field ID: `IPv6.version`

Lenght in bit or Function

Integer, String Array

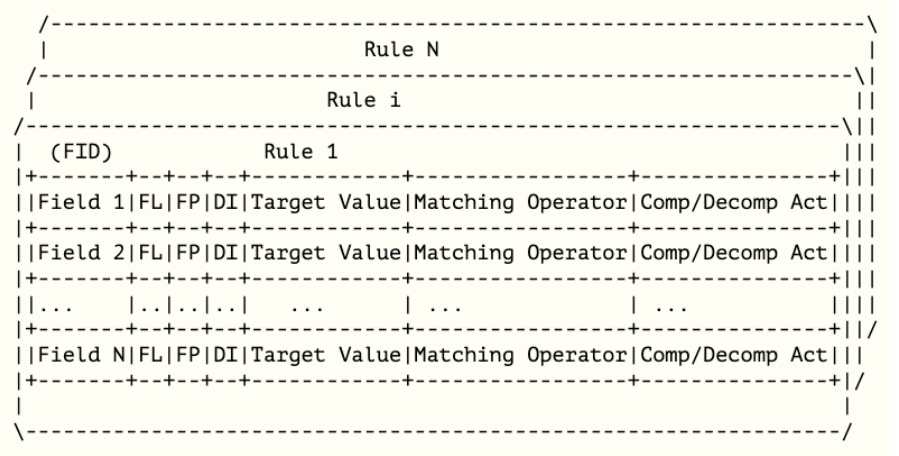
Keywords + parameters

Yang tree



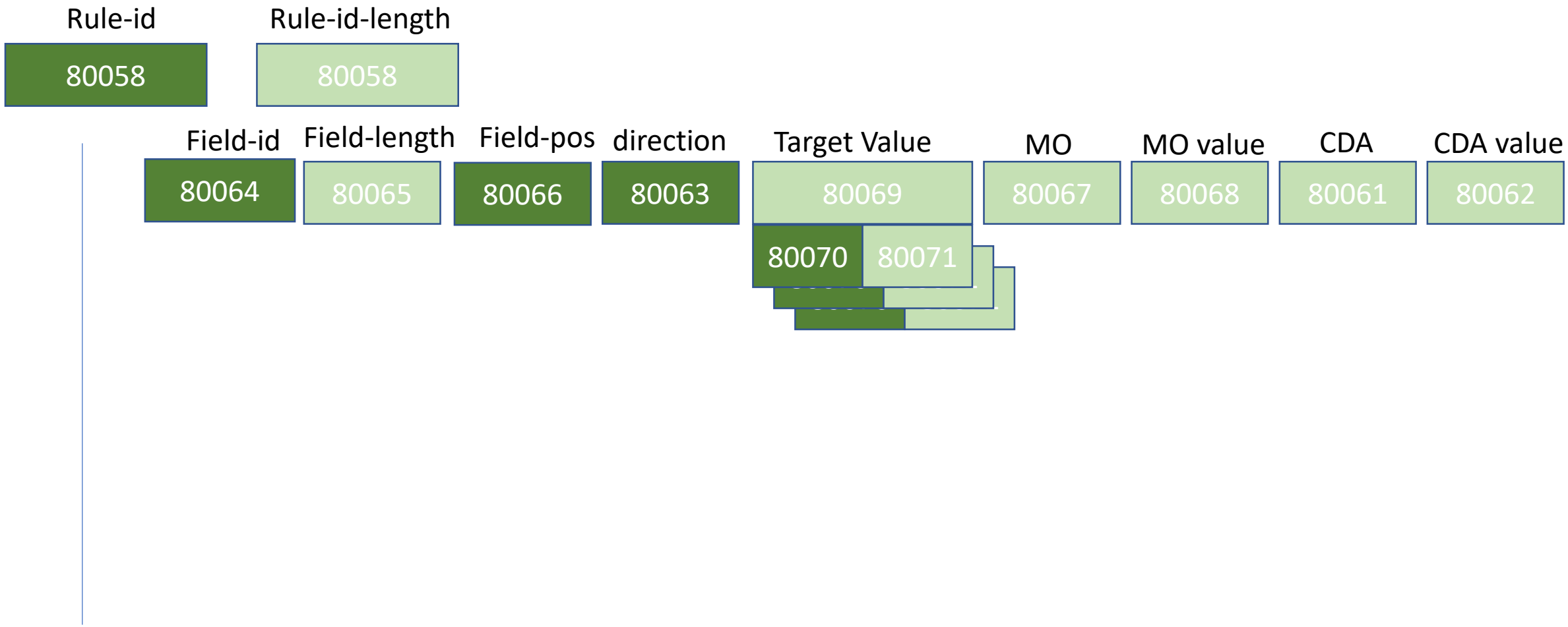

```
+--rw schc-context
   +--rw all-rules* [rule-id]
      +--rw rule-id                uint16
      +--rw rule-id-length?        uint8
      +--rw (rule-nature)?
         +--:(fragmentation)
         |  +--rw dtagsize?        uint8
         |  +--rw wsize?           uint8
         |  +--rw fcnsize?         uint8
         |  +--rw (mode)?
         |     +--:(no-ack)
         |     +--:(ack-always)
         |     +--:(ack-on-error)
         +--:(compression)
            +--rw rule-entries* [field-id field-position direction-indicator]
               +--rw field-id                                  field-id-generic-type
               +--rw field-length?                             field-length-type
               +--rw field-position                            uint8
               +--rw direction-indicator                       direction-indicator-type
               +--rw target-values* [target-value-pos]
               |  +--rw target-value?       target-value-type
               |  +--rw target-value-pos    uint8
               +--rw matching-operator?                        matching-operator-type
               +--rw matching-operator-value?                  target-value-type
               +--rw compression-decompression-action?         compression-decompression-action-type
               +--rw compression-decompression-action-value?   target-value-type
```

SID values

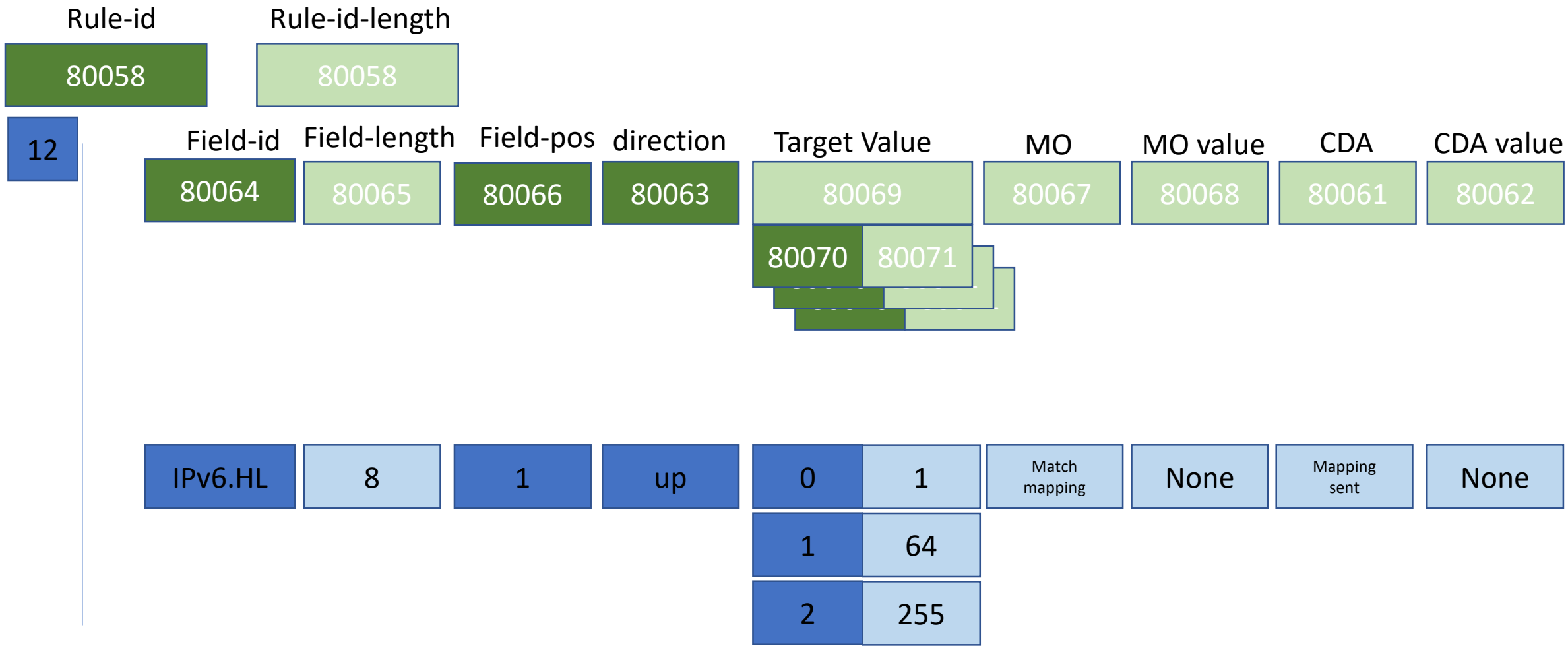



```
80056      node /schc-context
80057      node /schc-context/all-rules
80058      node /schc-context/all-rules/rule-id
80059      node /schc-context/all-rules/rule-id-length
80060      node /schc-context/all-rules/rule-nature/compression/rule-entries
80061      node /schc-context/all-rules/rule-nature/compression/rule-entries/compression-decompression-action
80062      node /schc-context/all-rules/rule-nature/compression/rule-entries/compression-decompression-action-value
80063      node /schc-context/all-rules/rule-nature/compression/rule-entries/direction-indicator
80064      node /schc-context/all-rules/rule-nature/compression/rule-entries/field-id
80065      node /schc-context/all-rules/rule-nature/compression/rule-entries/field-length
80066      node /schc-context/all-rules/rule-nature/compression/rule-entries/field-position
80067      node /schc-context/all-rules/rule-nature/compression/rule-entries/matching-operator
80068      node /schc-context/all-rules/rule-nature/compression/rule-entries/matching-operator-value
80069      node /schc-context/all-rules/rule-nature/compression/rule-entries/target-values
80070      node /schc-context/all-rules/rule-nature/compression/rule-entries/target-values/target-value
80071      node /schc-context/all-rules/rule-nature/compression/rule-entries/target-values/target-value-pos
80072      node /schc-context/all-rules/rule-nature/fragmentation/dtagsize
80073      node /schc-context/all-rules/rule-nature/fragmentation/fcnsize
80074      node /schc-context/all-rules/rule-nature/fragmentation/wsize
```

SID Access


Rule-id
Rule-id-length
80058
80058
Field-id
Field-length
Field-pos
direction
Target Value
MO
MO value
CDA
CDA value
80064
80065
80066
80063
80069
80067
80068
80061
80062
80070
80071

SID Access


Rule-id
Rule-id-length
80058
80058
12
Field-id
Field-length
Field-pos
direction
Target Value
MO
MO value
CDA
CDA value
80064
80065
80066
80063
80069
80067
80068
80061
80062
80070
80071
IPv6.HL
8
1
up
0
1
Match mapping
None
Mapping sent
None
1
64
2
255

How to represent values

```
identity field-id-sid-type  {
         description "Field ID with SID";
}

identity field-id-ipv6-version {
         base field-id-sid-type;
         description "IPv6 version field from RFC8200";
}

identity field-id-ipv6-trafficclass {
         base field-id-sid-type;
         description "IPv6 Traffic Class field from RFC8200";
}

identity field-id-ipv6-flowlabel {
         base field-id-sid-type;
         description "IPv6 Flow Label field from RFC8200";
}
```

How to represent values - sid

```
80021      identity /field-id-sid-type/field-id-coap-option-max-age
80022      identity /field-id-sid-type/field-id-coap-option-no-response
80023      identity /field-id-sid-type/field-id-coap-option-observe
80024      identity /field-id-sid-type/field-id-coap-option-proxy-scheme
80025      identity /field-id-sid-type/field-id-coap-option-proxy-uri
80026      identity /field-id-sid-type/field-id-coap-option-size1
80027      identity /field-id-sid-type/field-id-coap-option-size2
80028      identity /field-id-sid-type/field-id-coap-option-uri-host
80029      identity /field-id-sid-type/field-id-coap-option-uri-path
80030      identity /field-id-sid-type/field-id-coap-option-uri-port
80031      identity /field-id-sid-type/field-id-coap-option-uri-query
80032      identity /field-id-sid-type/field-id-coap-tkl
80033      identity /field-id-sid-type/field-id-coap-token
80034      identity /field-id-sid-type/field-id-coap-type
80035      identity /field-id-sid-type/field-id-coap-version
80036      identity /field-id-sid-type/field-id-ipv6-appiid
80037      identity /field-id-sid-type/field-id-ipv6-appprefix
80038      identity /field-id-sid-type/field-id-ipv6-deviid
80039      identity /field-id-sid-type/field-id-ipv6-devprefix
80040      identity /field-id-sid-type/field-id-ipv6-flowlabel
80041      identity /field-id-sid-type/field-id-ipv6-hoplimit
80042      identity /field-id-sid-type/field-id-ipv6-nextheader
80043      identity /field-id-sid-type/field-id-ipv6-payloadlength
80044      identity /field-id-sid-type/field-id-ipv6-trafficclass
80045      identity /field-id-sid-type/field-id-ipv6-version
80046      identity /field-id-sid-type/field-id-udp-checksum
80047      identity /field-id-sid-type/field-id-udp-length
```

How to represent values - shortcuts

```
typedef field-id-shortcut-type {
      type enumeration {
              enum ipv6-diffserv              { value   0; }
              enum ipv6-flowlabel             { value   1; }
              enum ipv6-nextheader            { value   2; }
              enum ipv6-devprefix             { value   3; }
              enum ipv6-deviid                { value   4; }
              enum ipv6-appprefix             { value   5; }
              enum ipv6-appiid                { value   6; }
              enum udp-devport                { value   7; }
              enum udp-appport                { value   8; }
              enum coap-type                  { value  -1; }
              enum coap-tkl                   { value  -2; }
              enum coap-code                  { value  -3; }
              enum coap-mid                   { value  -4; }
              enum coap-option-observe        { value  -5; }
              enum coap-option-location-path  { value  -6; }
              enum coap-option-uri-path       { value  -7; }
              enum coap-option-content-format { value  -8; }
              enum coap-option-max-age        { value  -9; }
              enum coap-option-uri-query      { value -10; }
              enum coap-option-accept         { value -11; }
              enum coap-option-location-query { value -12; }
              enum coap-option-proxy-uri      { value -13; }
              enum coap-option-size1          { value -14; }
              enum coap-option-no-response    { value -15; }

      }
}
```

How to represent values – COREconf coding

```
typedef field-id-generic-type {
    description "Field ID generic type. This type is used for either
                 predefined field ID or SID defined Field ID. the goal
                 is to have a shorter representation for predefined Field
                 ID with COREconf: predefined Field ID are coded in 1 byte
                 if value is less than 23 and 2 bytes otherwise. SID Field
                 ID are coded with a tag indicating identityref and a CBOR
                 integer, generally on 4 bytes.";
    type union {
       type identityref {
           base field-id-sid-type;
       }
       type field-id-shortcut-type;
    }

}
```

COREconf will use CBOR type or tag to distinguish between union type => No overhead.

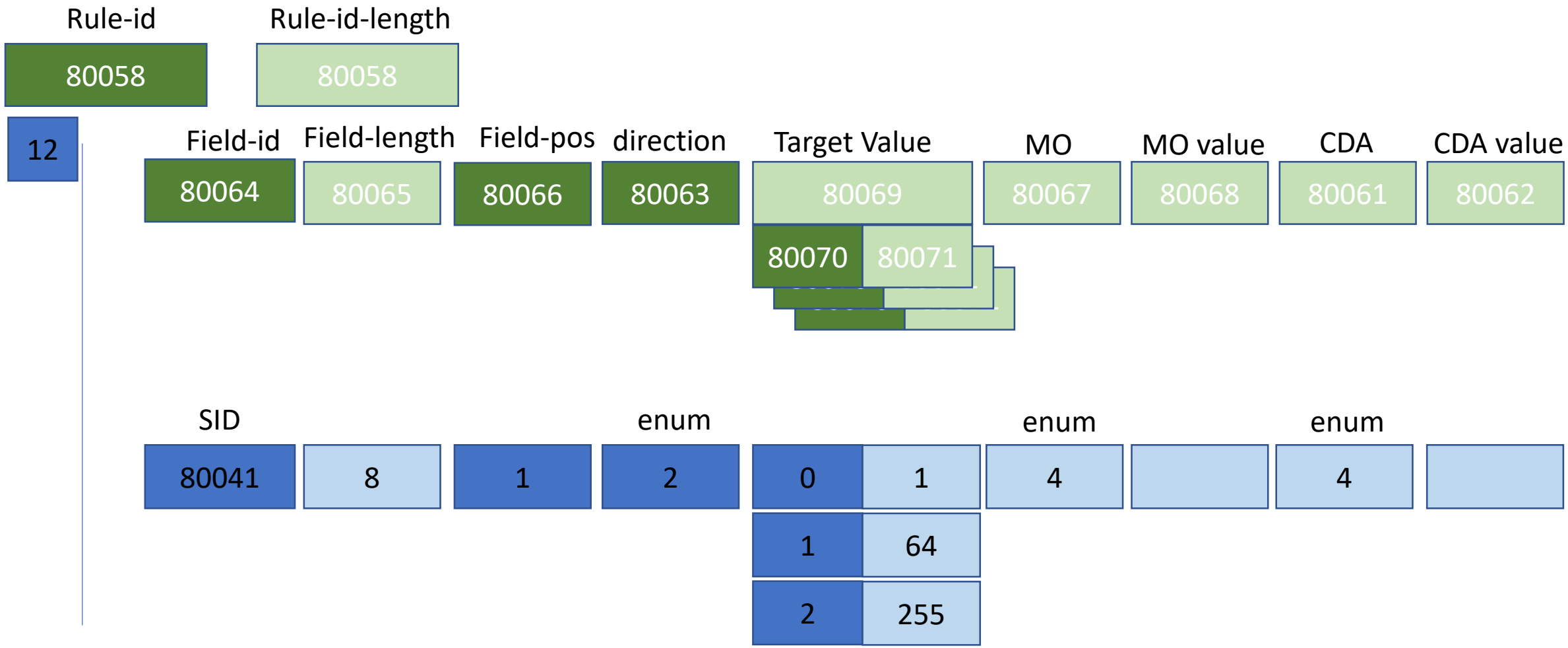
Field Length

```
identity field-length-function-token-length {
         base field-length-sid-type;
         description "residue length in Byte is sent";
}

typedef field-length-function-num-type {
   type enumeration {
        enum variable { value -1;}
        enum tokenlenght { value -2;}
   }
}

typedef field-length-type {
    type union {
        type uint8;
        type field-length-function-num-type;
        type identityref {
            base field-length-sid-type;
        }
    }
}
```

Target Value

```
      typedef target-value-type {
        type union {
          type uint64;
          type string;
         }
      }

grouping target-values-type {
      leaf target-value {
           type target-value-type;
      }
      leaf target-value-pos {
           type uint8;
      }
}
```

SID Access


Rule-id
Rule-id-length
80058
80058
12
Field-id
Field-length
Field-pos
direction
Target Value
MO
MO value
CDA
CDA value
80064
80065
80066
80063
80069
80067
80068
80061
80062
80070
80071
SID
enum
enum
enum
80041
8
1
2
0
1
4
4
1
64
2
255

Next Steps

- Check validity of Yang model
- Study impact of COREconf.