YANG of THINGS

Side meeting

Alexander Pelov <a@ackl.io>

AD: Benoit Claise <bclaise@cisco.com>

Note Well

Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:

- **The IETF plenary session**
- **The IESG, or any member thereof on behalf of the IESG**
- **Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices**
- **Any IETF working group or portion thereof**
- **Any Birds of a Feather (BOF) session**
- **The IAB or any member thereof on behalf of the IAB**
- **The RFC Editor or the Internet-Drafts function**

All IETF Contributions are subject to the rules of RFC 5378 and RFC 8179.

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice. Please consult RFC 5378 and RFC 8179 for details.

A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.

A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.

Reminder:

Minutes may be taken *
This meeting might not be recorded **
Presence may be logged ***

- Scribe; please contribute online to the minutes at: http://etherpad.tools.ietf.org:9000/p/yot

** Recordings and Minutes are public and may be subject to discovery in the event of litigation.

Remote participation through: https://jitsi.tools.ietf.org/yot

Agenda bashing

10:00 am> Introduction, Why YANG for IoT? [10 min]

10:10 am> The CoMI framework [35 min]

10 min: The CoMI protocol – Michel Veillette, Peter van der Stok, Alexander Pelov

10 min: YANG-to-CBOR – Michel Veillette

10 min: The SID registry – Alexander Pelov

5 min: Schema mount – Michel Veillette

10:45 am> Mapping existing tools to YANG [05 min]

5 min: LWM2M to CoMI - Peter van der Stok, Jaime Jimenez

10:50 am> CoMI for IoT (as of today) [15 min]

5 min: Firmware update over CoMI - Michel Veillette, Alexander Pelov

5 min: Event logger and notification control - Michel Veillette

5 min: LPWAN - Laurent Toutain, Ana Minaburo

11:05 am> Yang , Embedded Systems and Devices [10 min]

5 min: YANG for embedded systems: Andy Bierman

5 min: Manufacturer Usage Description (MUD) - Eliot Lear, Thorsten Dahm

11:15 am> Discussions [45 min]

Why YoT?

Andy Bierman
Michel Veillette
Peter van der Stok
Alexander Pelov <a@ackl.io>

History

- Three major drafts, WG documents @CoRE with WGLC by IETF100
  - CoMI (since January 2017)
  - SID (since October 2016)
  - YANG-CBOR (since April 2016)
- IETF98
  - Semi-formal CoMI meeting
    - Efficient binary encoding ; IoT-centric YANG ; More
- IETF99 is the last IETF without
  - Working CoMI implementation
  - Working SID registry
  - Specialized YANG modules for IoT

History

- Three major drafts, WG documents @CoRE with WGLC by IETF100
  - CoMI (since January 2017)
  - SID (since October 2016)
  - YANG-CBOR (since April 2016)

- IETF98
  - Semi-formal CoMI meeting
    - Efficient binary encoding ; IoT-centric YANG ; More

- IETF99 is the last IETF without
  - Working CoMI implementation
  - Working SID registry
  - Specialized YANG modules for IoT

YoT is THING-Centric !

Routers and others MAY (and will) benefit significantly from our work, but will not be the main concern.

YANG Ecosystem

2003 – NETCONF WG

RFCs 4741, 4742, 4743, 4744

2008 – NETMOD WG

RFCs 6241, 6242, 6243, 6244, 6020, 6021

TODAY

+1300 public YANG models

IETF WG

LIME, L3SM, SUPA, I2NSF

YANG Users

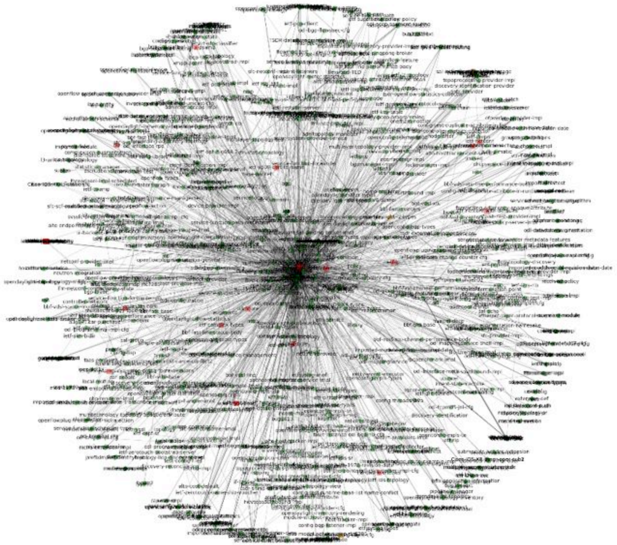
Broadband Forum – IEEE – IETF – ITU-T – Metro Ethernet Forum – OpenConfig – OpenDaylight – OpenNetworkingFoundation

It’s the data model!

Powerful and versatile data modeling language

Standard data modeling language of choice at IETF for management.

YANG



All public YANG modules
April, 2017

It’s the data model!

Powerful and versatile data modeling language

Standard data modeling language of choice at IETF for management.

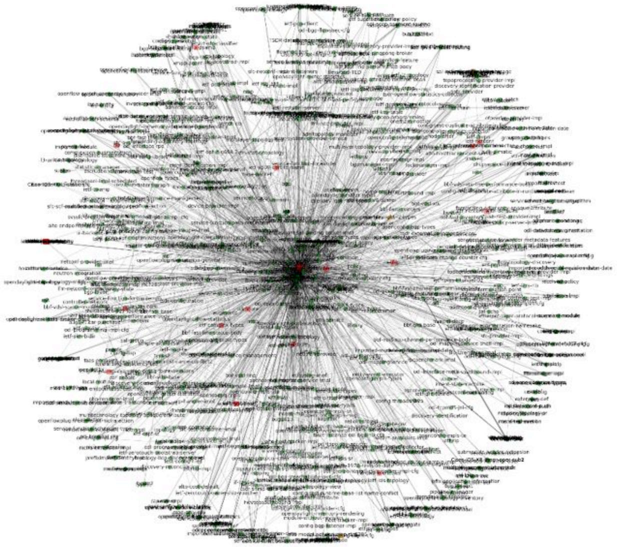
YANG

Modules, sub-modules
Templates
Extending data models
Conditions (if-feature)
Extensible enumerations

Simple data types
- unsigned integer, integer, string, enumeration, bits, binary, empty

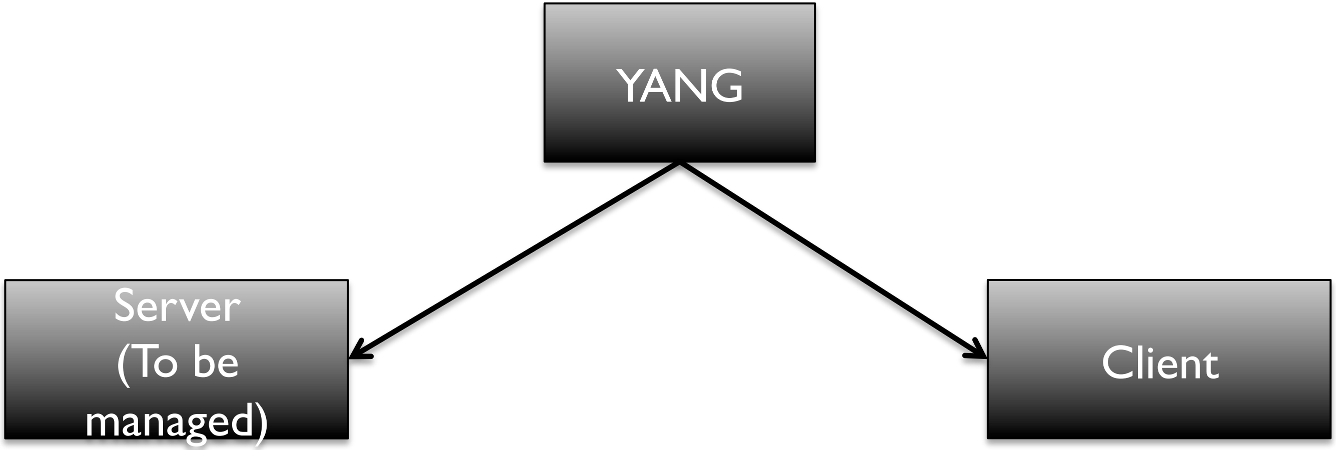
Unions
Labels (identity)
References to labels, data items, etc.
Collections
- Sets, lists

Structures (composite types)

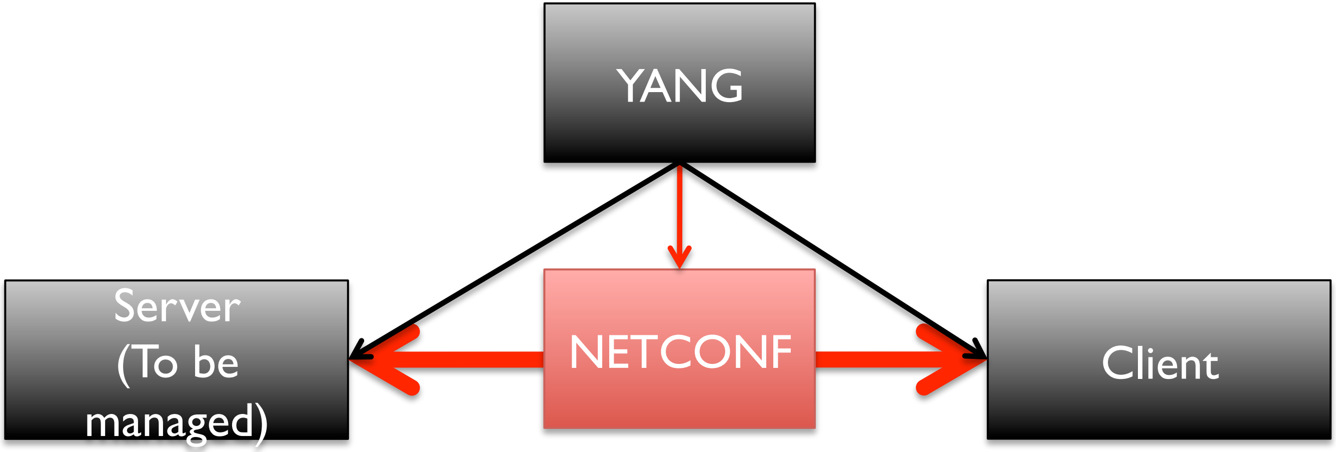


All public YANG modules
April, 2017

Data model as contract

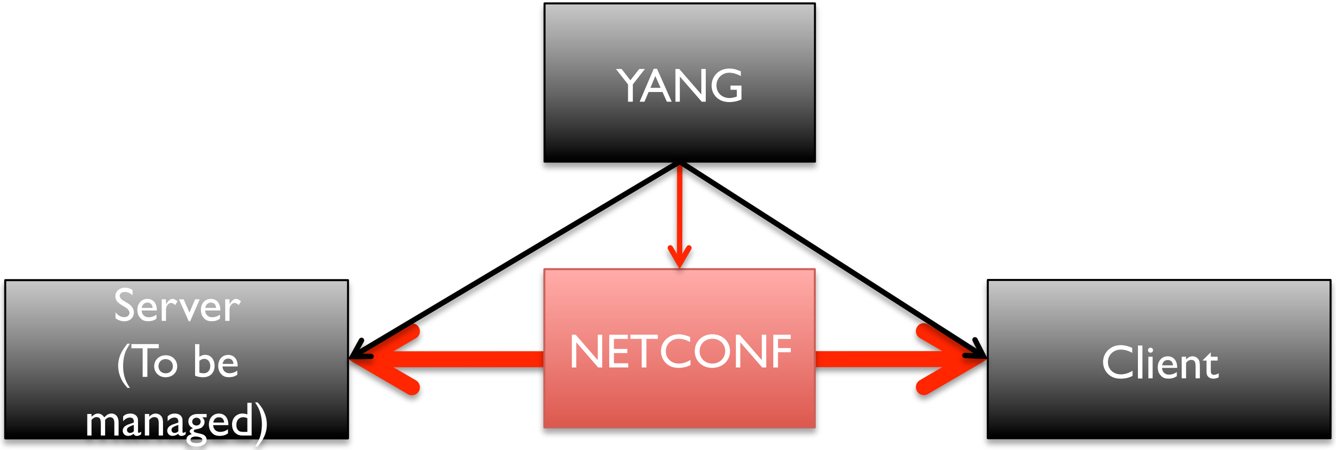


And the interaction model!



Rich interaction model

And the protocol bindings!

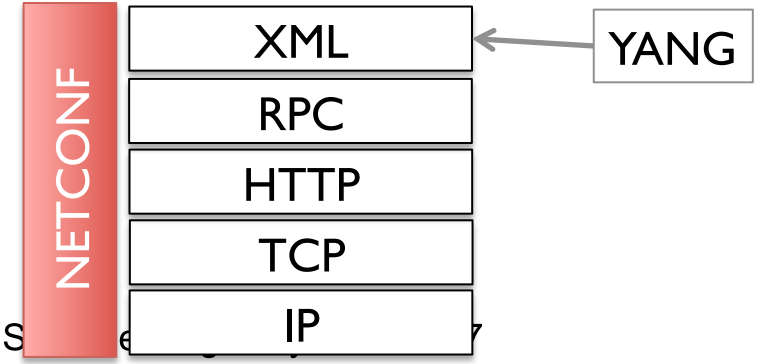


YANG

Server
(To be
managed)

NETCONF

Client



NETCONF

XML

RPC

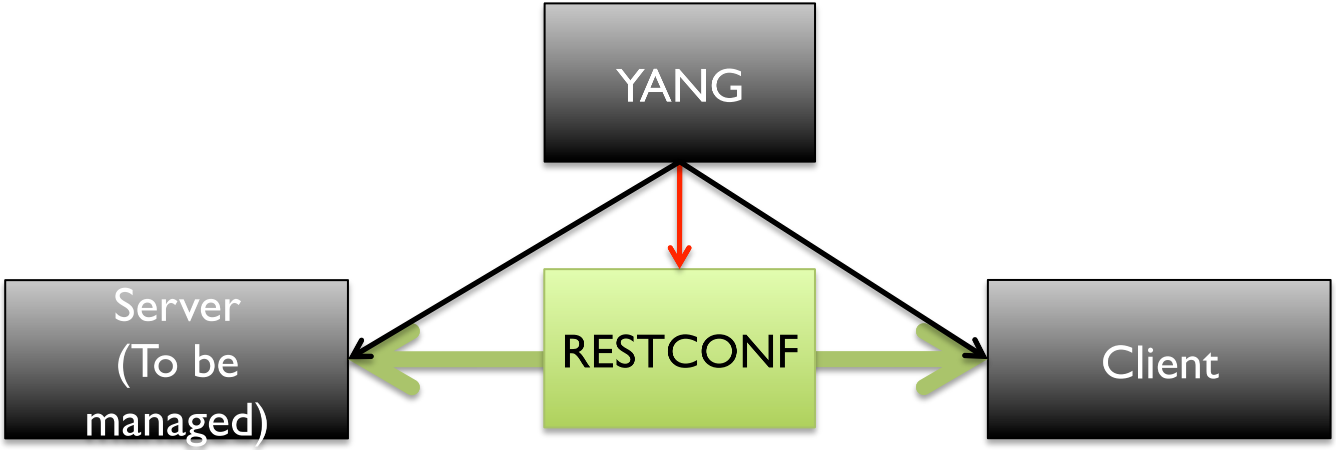
HTTP

TCP

IP

YANG

And the protocol bindings!



NETCONF

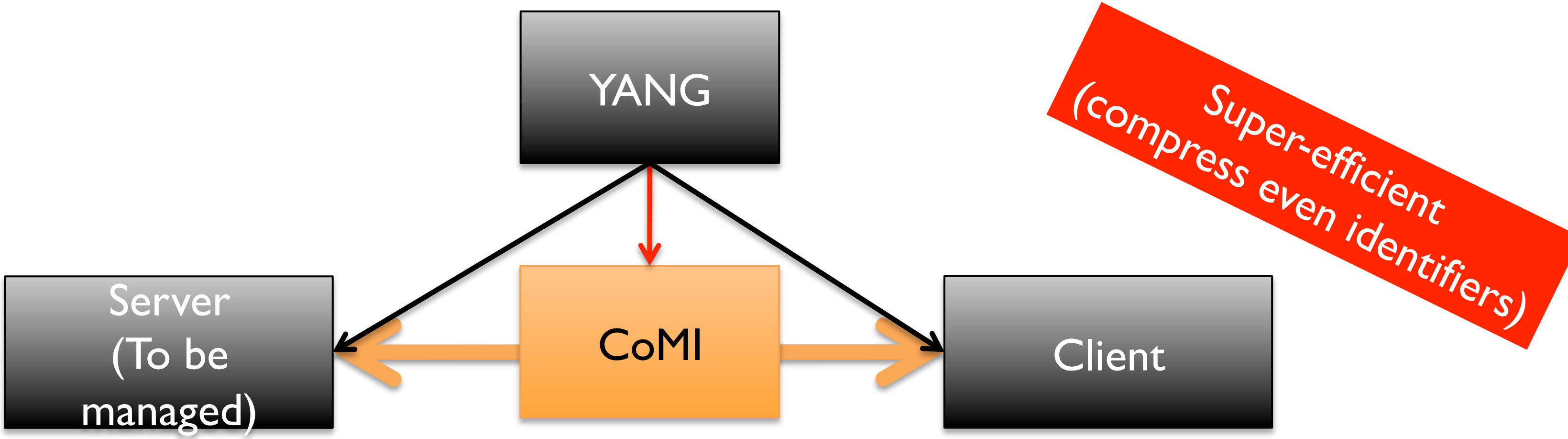
- XML
- RPC
- HTTP
- TCP
- IP

RESTCONF

- **JSON**
- **REST / RPC**
- HTTP
- TCP
- IP

YANG

And the protocol bindings!

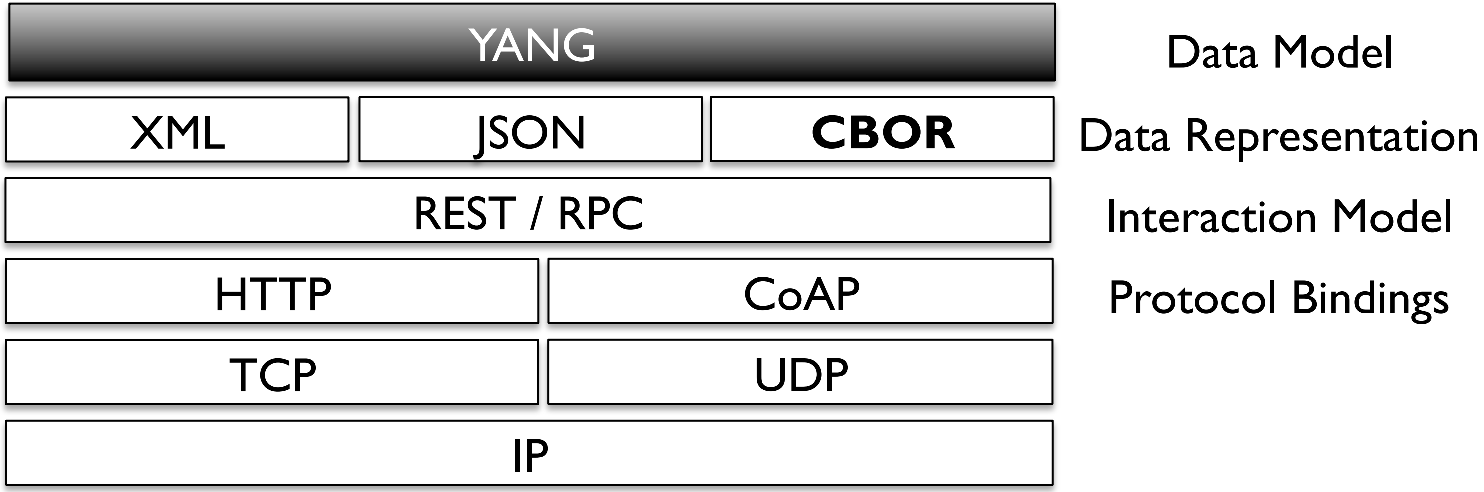


Super-efficient
(compress even identifiers)

NETCONF	RESTCONF	CoMI
XML	JSON	**CBOR**
RPC	REST / RPC	REST / RPC
HTTP	HTTP	**CoAP**
TCP	TCP	**UDP**
IP	IP	IP

YANG → **CBOR**

Which gives: the YANG Stack



YANG — Data Model

XML | JSON | **CBOR** — Data Representation

REST / RPC — Interaction Model

HTTP | CoAP — Protocol Bindings

TCP | UDP

IP

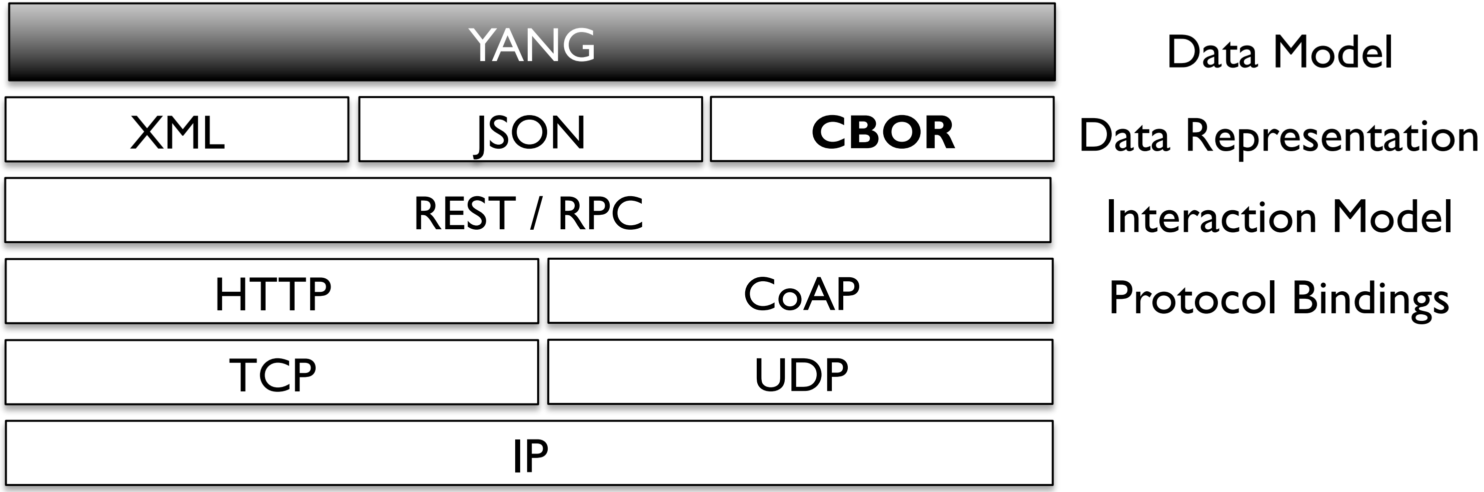
Features (small sample)

Constraints on data

Rich built-in data + Rich extension mechanism

Transactions

Balance between high-level data modeling and low-level bits-on-the-wire encoding



YANG — Data Model

XML | JSON | **CBOR** — Data Representation

REST / RPC — Interaction Model

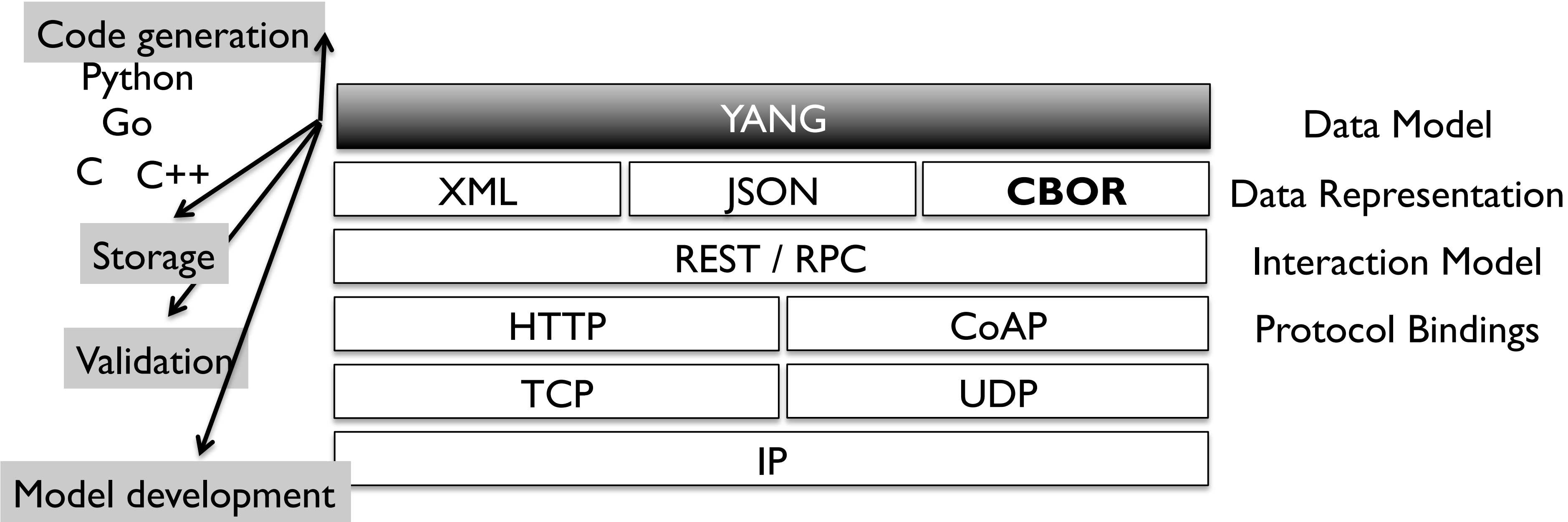
HTTP | CoAP — Protocol Bindings

TCP | UDP

IP

And the ecosystem

+ Tools!



Code generation

Python

Go

C C++

Storage

Validation

Model development

YANG			Data Model
XML	JSON	**CBOR**	Data Representation
REST / RPC			Interaction Model
HTTP	CoAP		Protocol Bindings
TCP	UDP		
IP			

YANG for IoT (YoT)

IETF

6TiSCH

LPWAN

(Side meetings)

YANG models

Manufacturer Usage Description (MUD)

LWM2M – CoMI mapping

RD

Decentralized Registry

SID

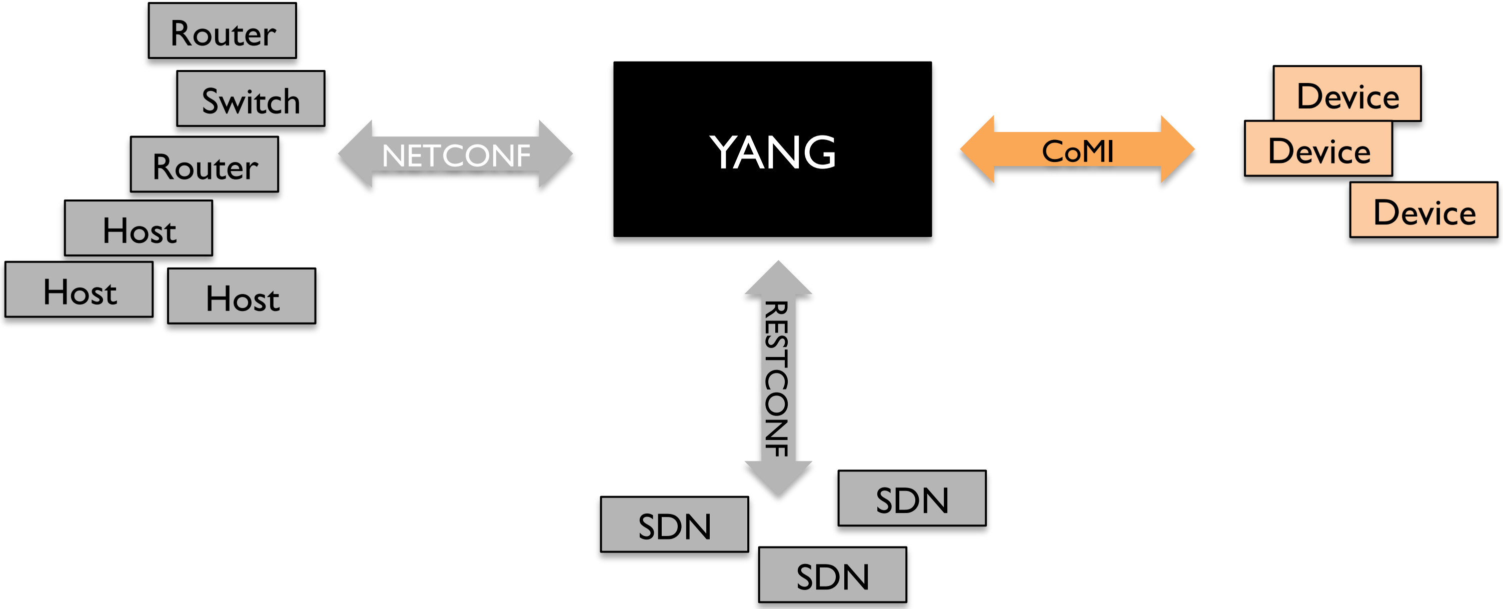
YANG for IoT (YoT)

- CoRE as a WG is not currently focused on network or device management.

- The "Yang of Things" (YOT) non-WG mailing list will discuss:
  - best practices for using YANG-based data modeling for the management of networks with constrained devices and constrained networks.
  - How to best make use of properties of the combination of technologies involved (YANG, CBOR, SID, CoAP, RESTCONF, …).
  - How these tech could be applied outside the COMI focus of interest.

- The YOT mailing list will also be the proper forum to discuss new YANG modules targeting constrained devices and networks.

Bringing YANG to IoT



Side meeting, July 20th, 2017

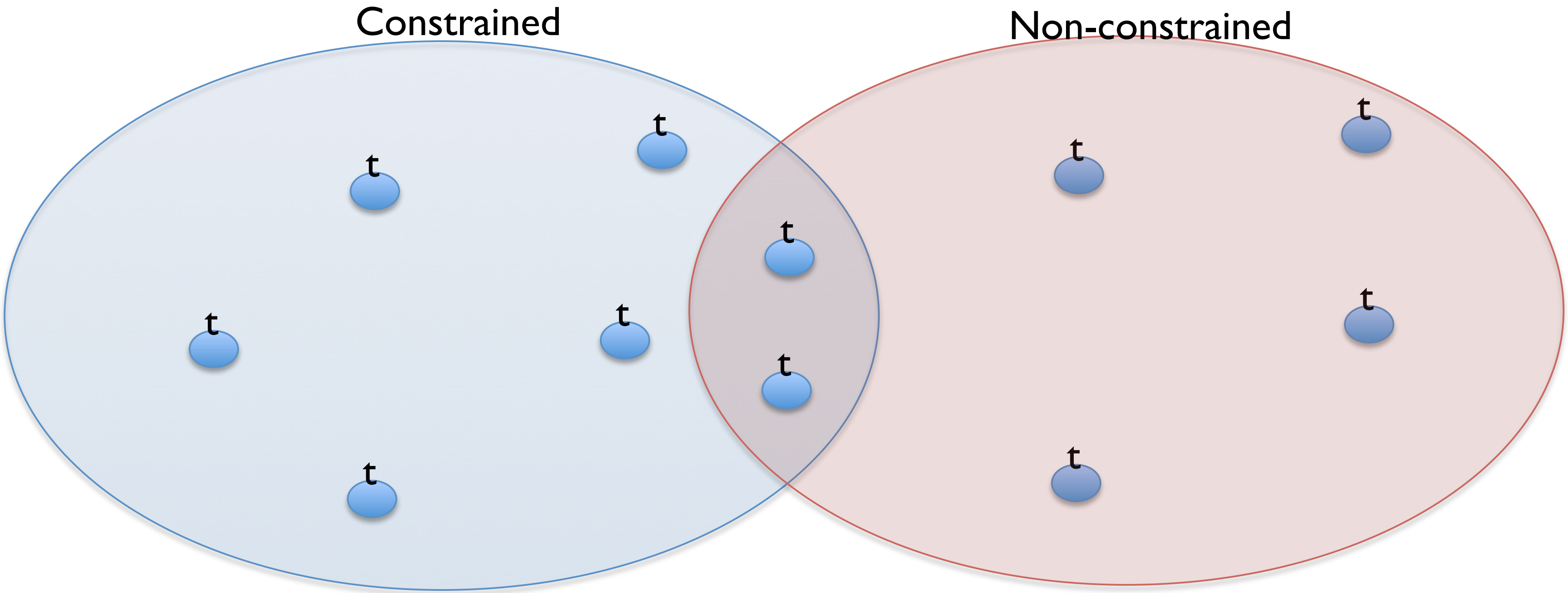
Constrained Management Interface (CoMI)

draft-ietf-core-comi-01

Andy Bierman
Michel Veillette <michel.veillette@trilliantinc.com>
Peter van der Stok
Alexander Pelov

Side meeting, July 20th, 2017

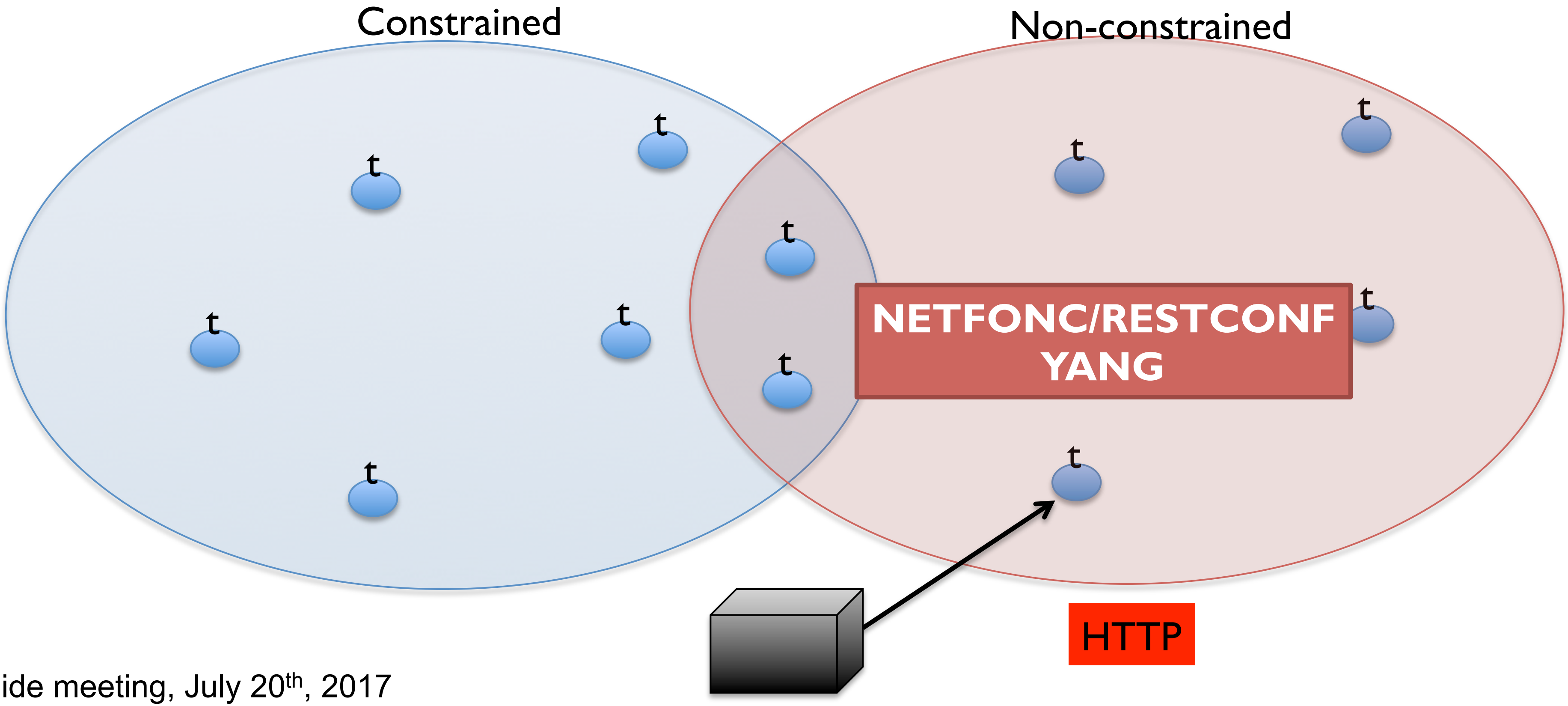
You want to manage things



Constrained

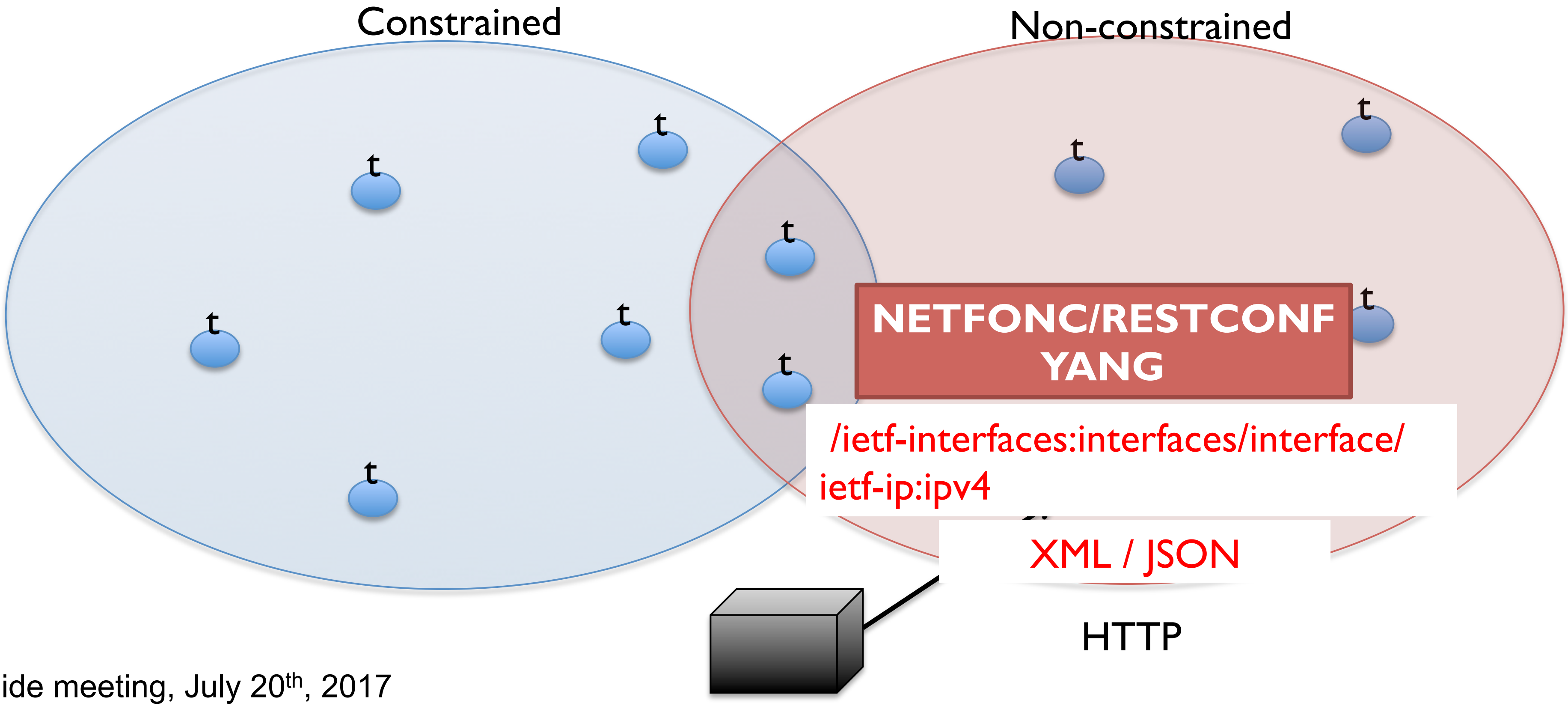
Non-constrained

You want to manage things



Side meeting, July 20th, 2017

You want to manage things



NETFONC/RESTCONF
YANG

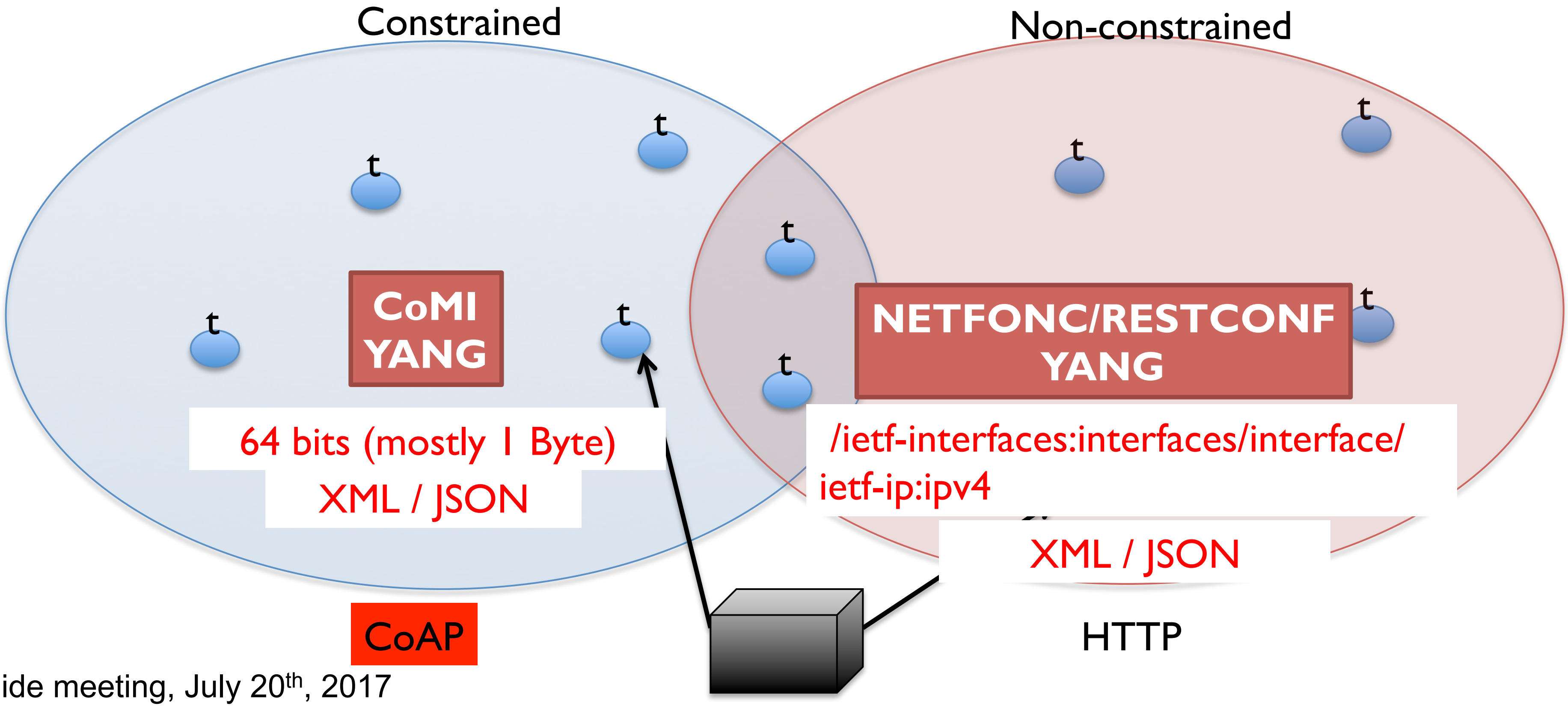
/ietf-interfaces:interfaces/interface/ietf-ip:ipv4

XML / JSON

HTTP

Side meeting, July 20th, 2017

You want to manage things



Protocol overview

```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",               / name  (SID 1537) /
  +1 : "Ethernet adaptor",   / description (SID 1534) /
  +5 : 1179,                 / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                  / enabled (SID 1535) /
 }
]
```

Protocol overview

CoAP

```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",                / name  (SID 1537) /
  +1 : "Ethernet adaptor",    / description (SID 1534) /
  +5 : 1179,                  / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                   / enabled (SID 1535) /
 }
]
```

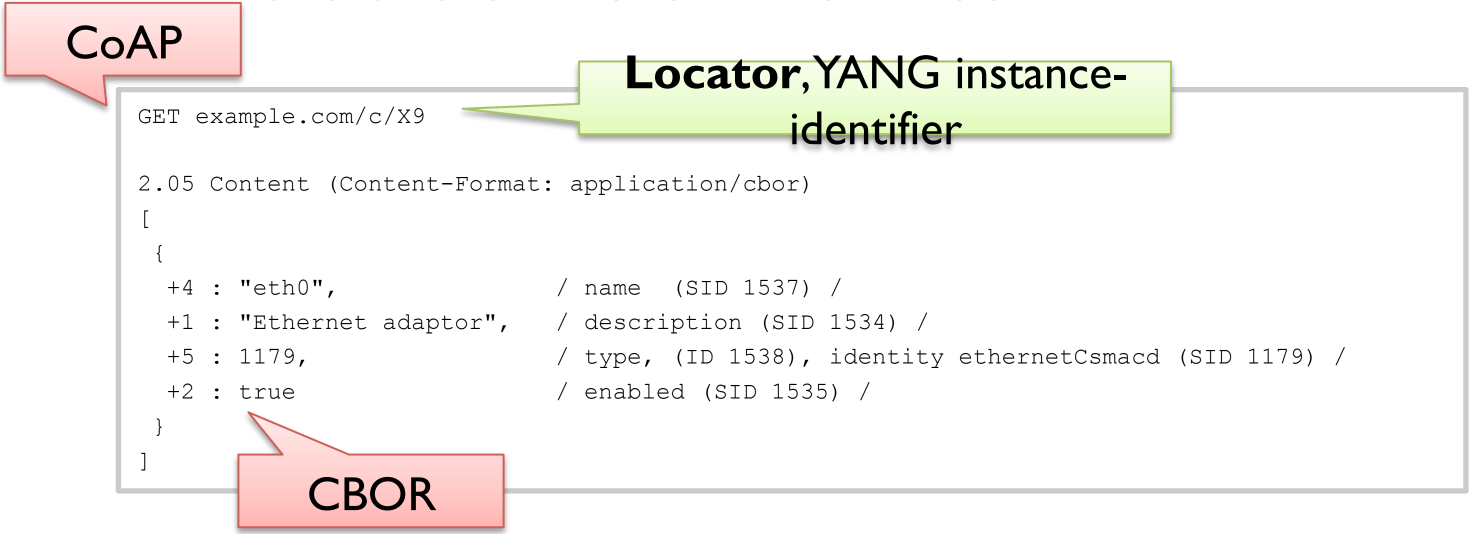
Protocol overview

CoAP

```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",                / name  (SID 1537) /
  +1 : "Ethernet adaptor",    / description (SID 1534) /
  +5 : 1179,                  / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                   / enabled (SID 1535) /
 }
]
```

CBOR

Protocol overview



CoAP

Locator, YANG instance-identifier

```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",                / name  (SID 1537) /
  +1 : "Ethernet adaptor",    / description (SID 1534) /
  +5 : 1179,                  / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                   / enabled (SID 1535) /
 }
]
```

CBOR

Protocol overview

CoAP

```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",                / name  (SID 1537) /
  +1 : "Ethernet adaptor",    / description (SID 1534) /
  +5 : 1179,                  / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                   / enabled (SID 1535) /
 }
]
```

Locator, YANG instance-identifier

Value, defined in draft-ietf-core-yang-cbor

CBOR

Protocol overview

CoAP

Locator, YANG instance-identifier

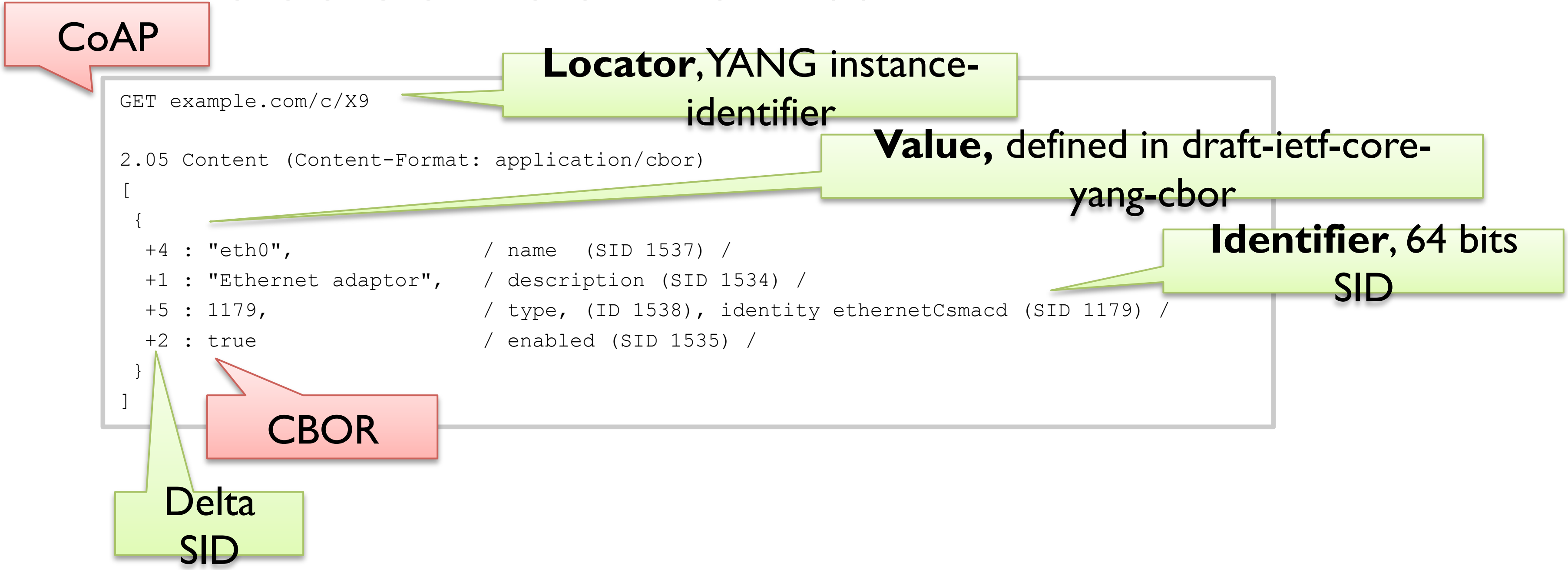
```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",                / name  (SID 1537) /
  +1 : "Ethernet adaptor",    / description (SID 1534) /
  +5 : 1179,                  / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                   / enabled (SID 1535) /
 }
]
```

Value, defined in draft-ietf-core-yang-cbor

Identifier, 64 bits SID

CBOR

Protocol overview



```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",               / name  (SID 1537) /
  +1 : "Ethernet adaptor",   / description (SID 1534) /
  +5 : 1179,                 / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                  / enabled (SID 1535) /
 }
]
```

CoAP

Locator, YANG instance-identifier

Value, defined in draft-ietf-core-yang-cbor

Identifier, 64 bits SID

CBOR

Delta SID

Protocol overview

CoAP

```
GET example.com/c/X9

2.05 Content (Content-Format: application/cbor)
[
 {
  +4 : "eth0",               / name  (SID 1537) /
  +1 : "Ethernet adaptor",   / description (SID 1534) /
  +5 : 1179,                 / type, (ID 1538), identity ethernetCsmacd (SID 1179) /
  +2 : true                  / enabled (SID 1535) /
 }
]
```

Locator, YANG instance-identifier

Value, defined in draft-ietf-core-yang-cbor

Identifier, 64 bits SID

CBOR

Delta SID

```
81                                # array(1)
   a4                             # map(4)
      04                          # unsigned(4)
      64                          # text(4)
         65746830                 # "eth0"
      01                          # unsigned(1)
      70                          # text(16)
         457468657726e65742061646170746f72 # "Ethernet adaptor"
      05                          # unsigned(5)
      19 049b                     # unsigned(1179)
      02                          # unsigned(2)
      f5                          # primitive(21)
```

32 bytes payload

Side meeting, July 20th, 2017

Protocol overview

Method	Resource	Content-Format
GET response	data node	/application/yang-value+cbor
PUT request	data node	/application/yang-value+cbor
POST request	data node	/application/yang-value+cbor
DELETE	data node	na
GET response	datastore	/application/yang-tree+cbor
PUT request	datastore	/application/yang-tree+cbor
POST request	datastore	/application/yang-tree+cbor
FETCH request	datastore	/application/yang-selectors+cbor
FETCH response	datastore	/application/yang-values+cbor
iPATCH request	datastore	/application/yang-patch+cbor
GET response	event stream	/application/yang-tree+cbor
POST request	rpc, action	/application/yang-value+cbor
POST response	rpc, action	/application/yang-value+cbor

Protocol overview

YANG data node	Encoding
uint8, uint16, uint32 and uint64	CBOR unsigned integer
int8, int16, int32 and int64	CBOR unsigned integer / signed integer
decimal64	CBOR unsigned integer / signed integer
string	CBOR text string
boolean	CBOR simple value 'true' or 'false'
enumeration	CBOR unsigned integer
bits	CBOR byte string
binary	CBOR byte string
leafref	As specified by the 'path' YANG statement
identityref	CBOR unsigned integer OR CBOR text string
empty	CBOR simple value 'null'
union	bits, enumeration, identityref and instance-identifier are tag to avoid datatype ambiguities
instance-identifier	CBOR unsigned integer / array OR CBOR text string

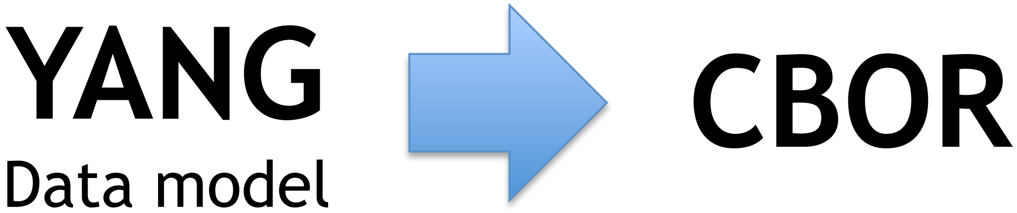
CBOR Encoding of Data Modeled with YANG

draft-ietf-core-yang-cbor-04

Michel Veillette
Alexander Pelov
Abhinav Sumaraju
Randy Turner
Ana Minaburo

Goal

Define the serialization rules to encode YANG data nodes in CBOR



YANG

Data model

CBOR

I-D.ietf-netmod-yang-json performs the same task for JSON. The table of content of both drafts are similar.

What YANG has?

- Simple data types
  - unsigned integer, integer, string, enumeration, bits, binary, empty
- Unions
- Labels (identity)
- References to labels, data items, etc.
- Collections
  - Sets, lists
- Structures (composite types)

What YANG has?

- Simple data types
  - unsigned integer, integer, string, enumeration, bits, binary, empty
- Unions

✓ **CBOR types**

- Labels (identity)
- References to labels, data items, etc.
- Collections
  - Sets, lists
- Structures (composite types)

What YANG has?

- Simple data types
  - unsigned integer, integer, string, enumeration, bits, binary, empty

✓ **CBOR types**

- Unions
- Labels (identity)

✓ **Tagged CBOR types**

- References to labels, data items, etc.
- Collections
  - Sets, lists
- Structures (composite types)

What YANG has?

- Simple data types
  - unsigned integer, integer, string, enumeration, bits, binary, empty
- Unions
- Labels (identity)
- References to labels, data items, etc.
- Collections
  - Sets, lists
- Structures (composite types)

✓ **CBOR types**

✓ **Tagged CBOR types**

✓ **Name / SID**

What YANG has?

- Simple data types
  - unsigned integer, integer, string, enumeration, bits, binary, empty
- Unions

✓ **CBOR types**

- Labels (identity)

✓ **Tagged CBOR types**

- References to labels, data items, etc.
- Collections

✓ **Name / SID**

  - Sets, lists
- Structures (composite types)

✓ **CBOR maps**
CBOR arrays

WGLC coming soon!

YANG Schema Item iDentifier (SID)

draft-ietf-core-sid-01

Andy Bierman

Michel Veillette

Peter van der Stok

Alexander Pelov <a@ackl.io>

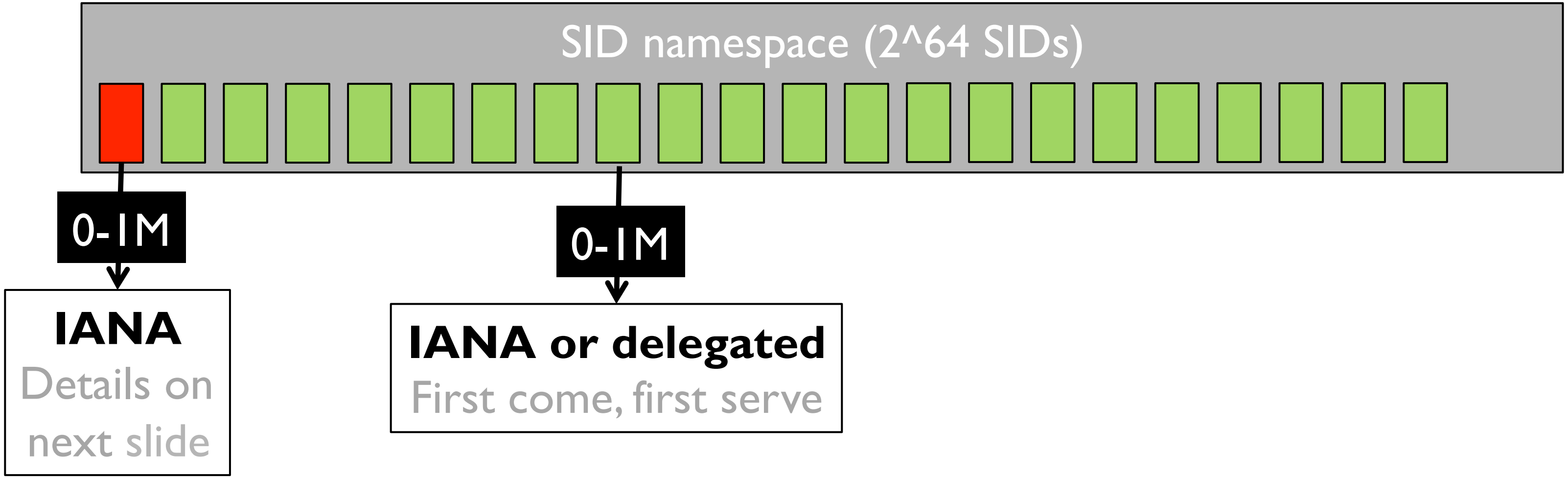
Side meeting, July 20th, 2017

Status and next steps

Four main topics

- SID definition (semantic)
  - 64 bit identifier assigned to all YANG identifiers
- SID file format (.sid)
  - YANG Schema -> JSON format
- SID file lifecycle
  - Range registration, .sid generation, .sid update
- Allocation policies
  - Two-tier allocation system
    - MegaRange (1M SIDs) and Range (~1000 SIDs flexible size)
  - Review allocation policy with IANA

SID Mega-range Registry



IANA

Details on next slide

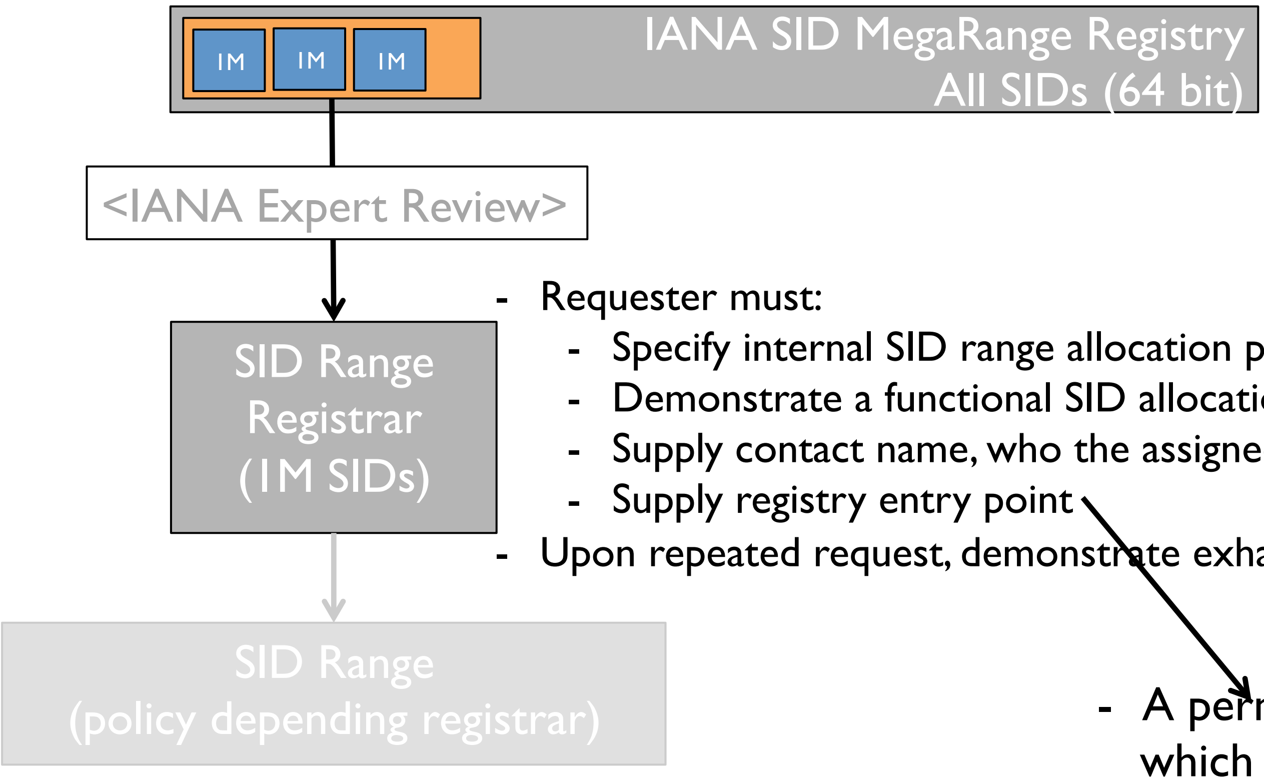
IANA or delegated

First come, first serve

SID Mega-range Registry

Hierarchical Allocation - 0-100M

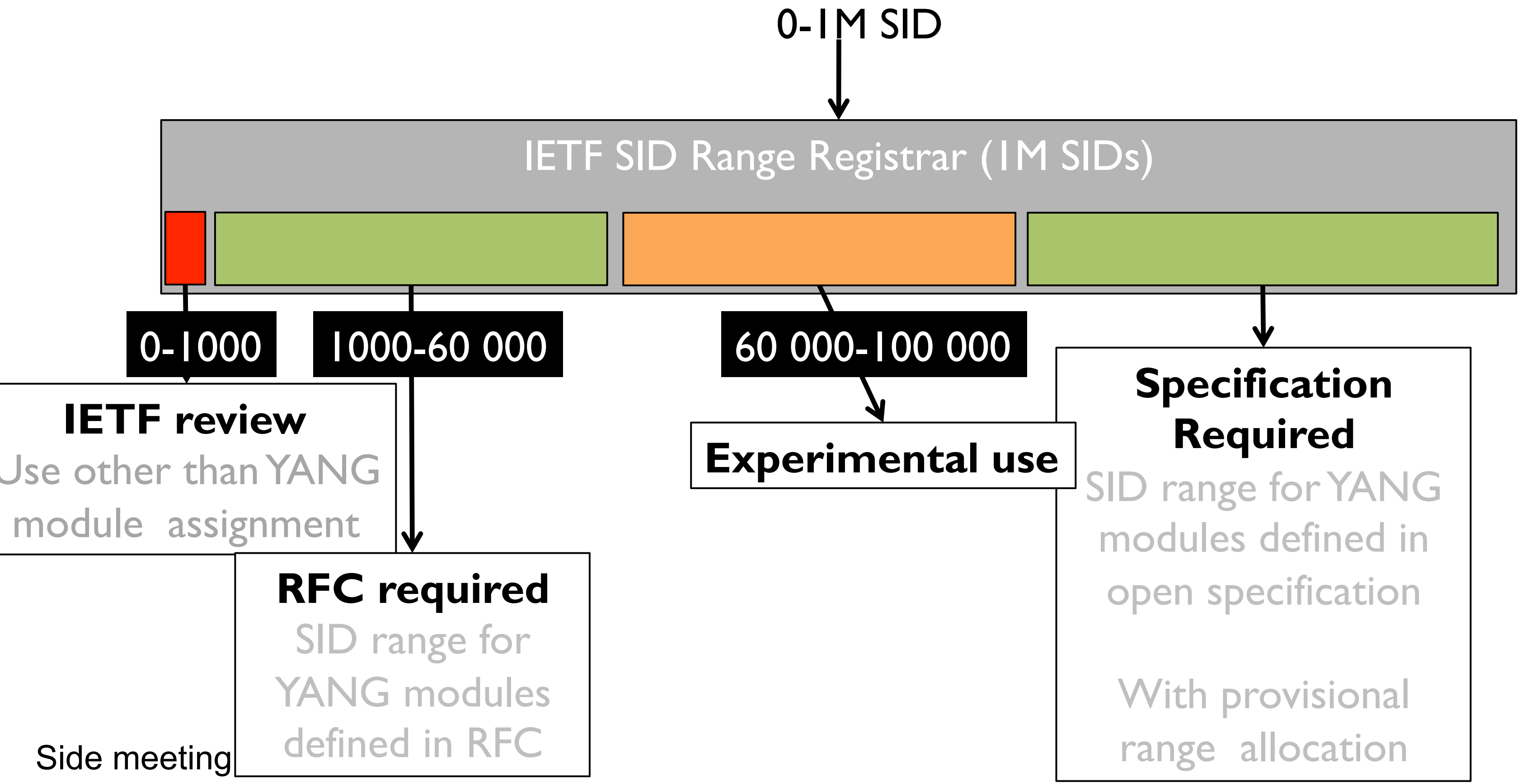
Reserved >100M


IANA SID MegaRange Registry
All SIDs (64 bit)
IM
IM
IM
<IANA Expert Review>
SID Range Registrar (1M SIDs)
SID Range (policy depending registrar)

- Requester must:
  - Specify internal SID range allocation policy
  - Demonstrate a functional SID allocation infrastructure
  - Supply contact name, who the assignee is, change controller
  - Supply registry entry point
- Upon repeated request, demonstrate exhaustion of range

- A permanent HTTP URL which allows to retrieve all YANG definitions

IETF SID Range Registrar



0-1M SID

IETF SID Range Registrar (1M SIDs)

0-1000

IETF review

Use other than YANG module assignment

1000-60 000

RFC required

SID range for YANG modules defined in RFC

60 000-100 000

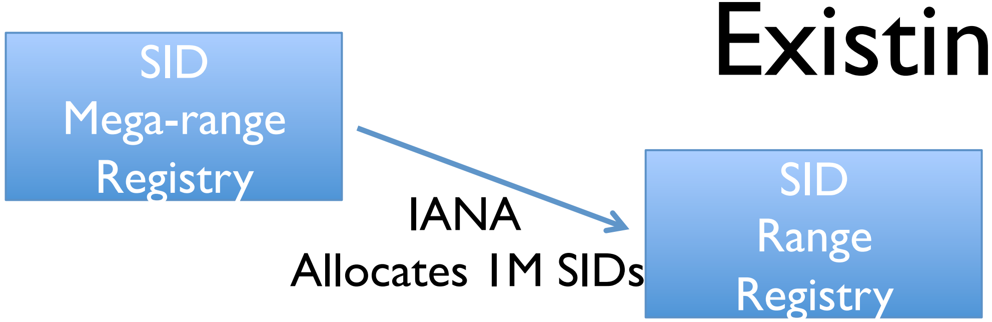
Experimental use

Specification Required

SID range for YANG modules defined in open specification

With provisional range allocation

Registration procedure Existing

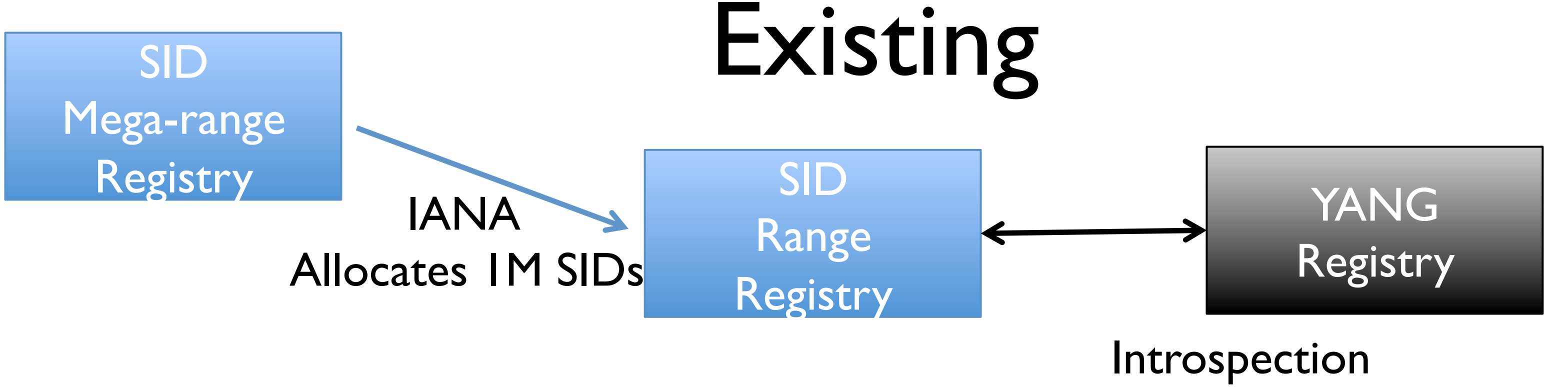


SID Mega-range Registry

IANA Allocates 1M SIDs

SID Range Registry

Registration procedure
Existing



SID Mega-range Registry

IANA
Allocates 1M SIDs

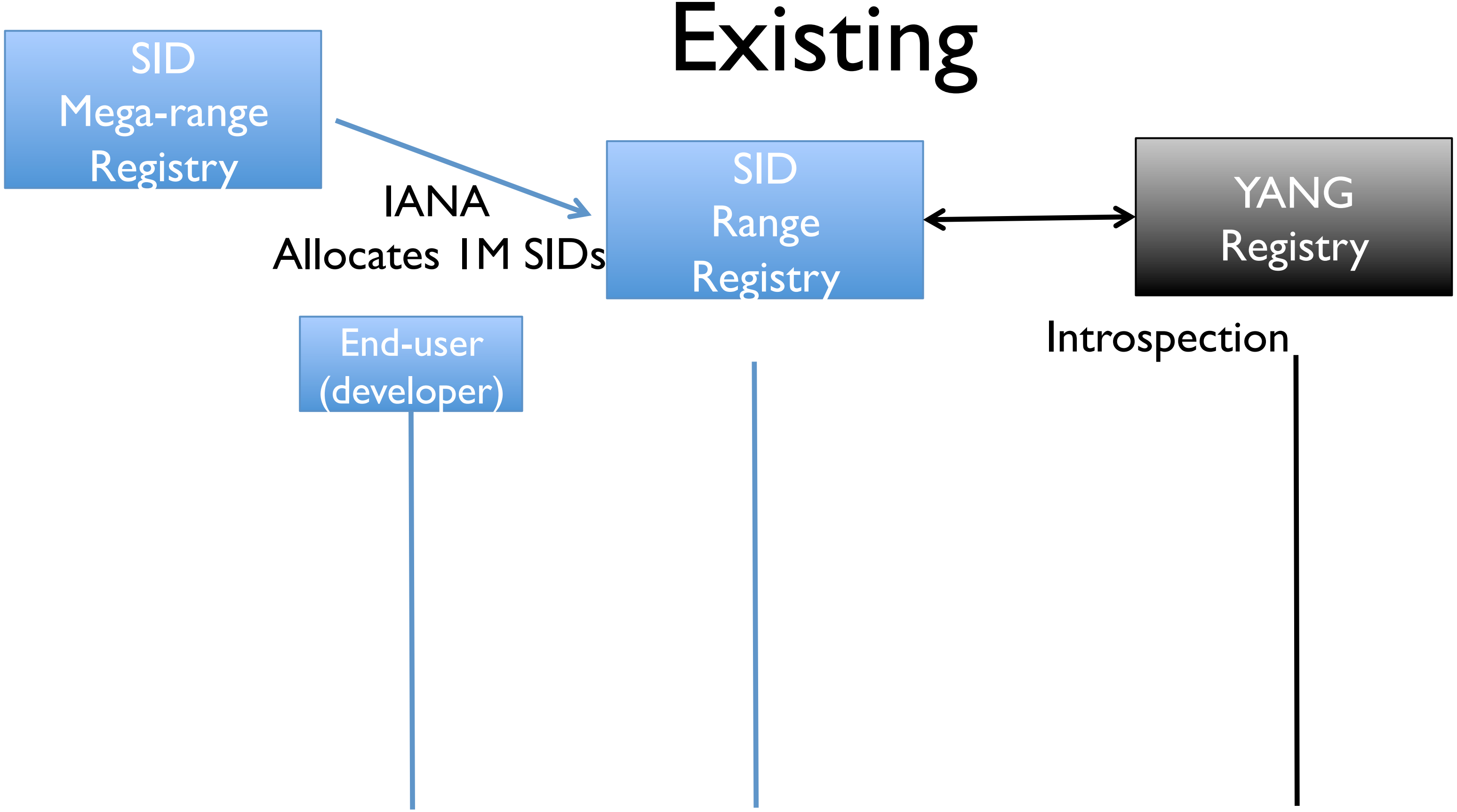
SID Range Registry

YANG Registry

Introspection

Side meeting, July 20th, 2017

Registration procedure Existing



SID Mega-range Registry

IANA
Allocates 1M SIDs

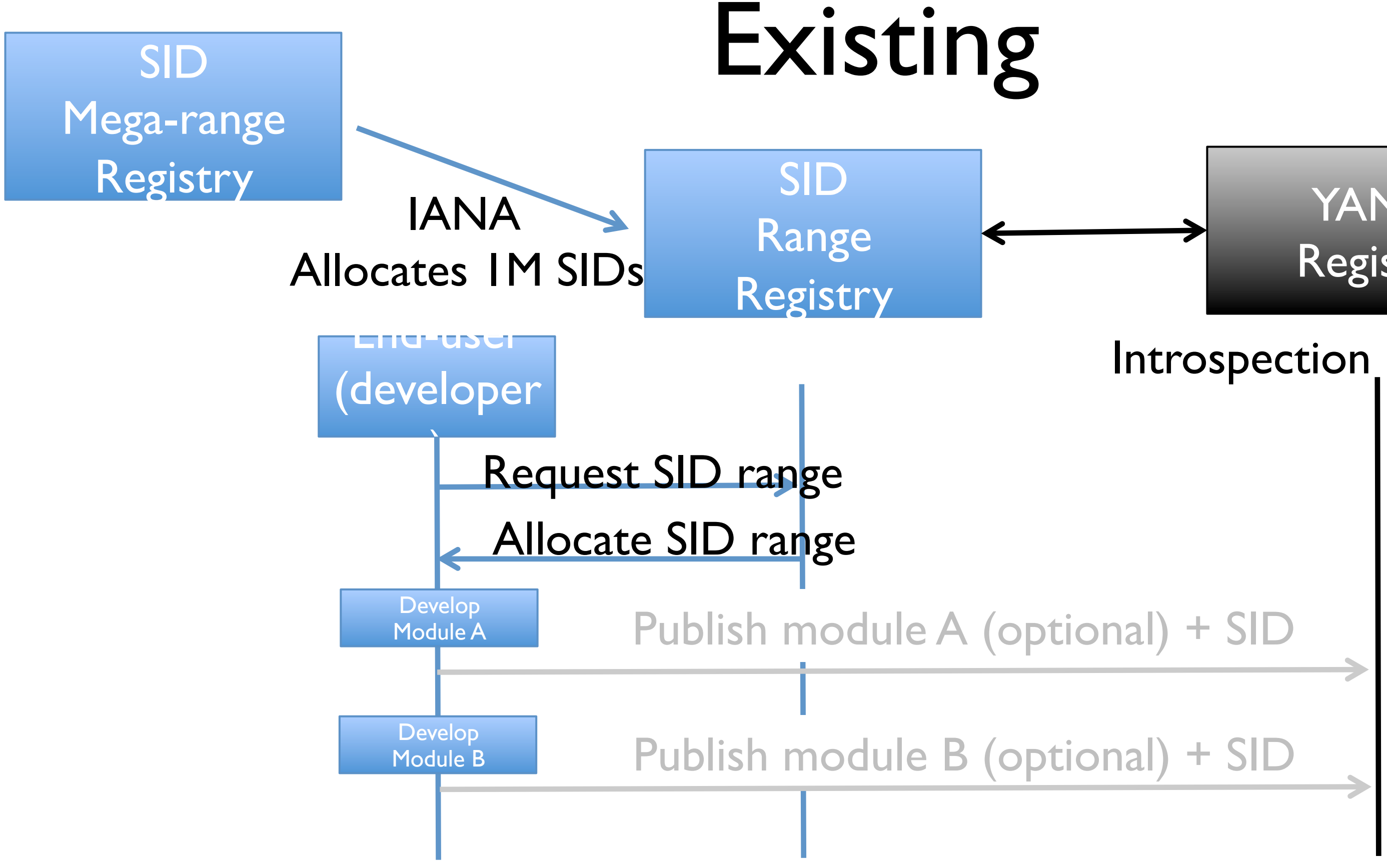
SID Range Registry

YANG Registry

End-user (developer)

Introspection

Registration procedure Existing



SID Mega-range Registry

IANA
Allocates 1M SIDs

SID Range Registry

YANG Registry

End-user
(developer
)

Introspection

Request SID range

Allocate SID range

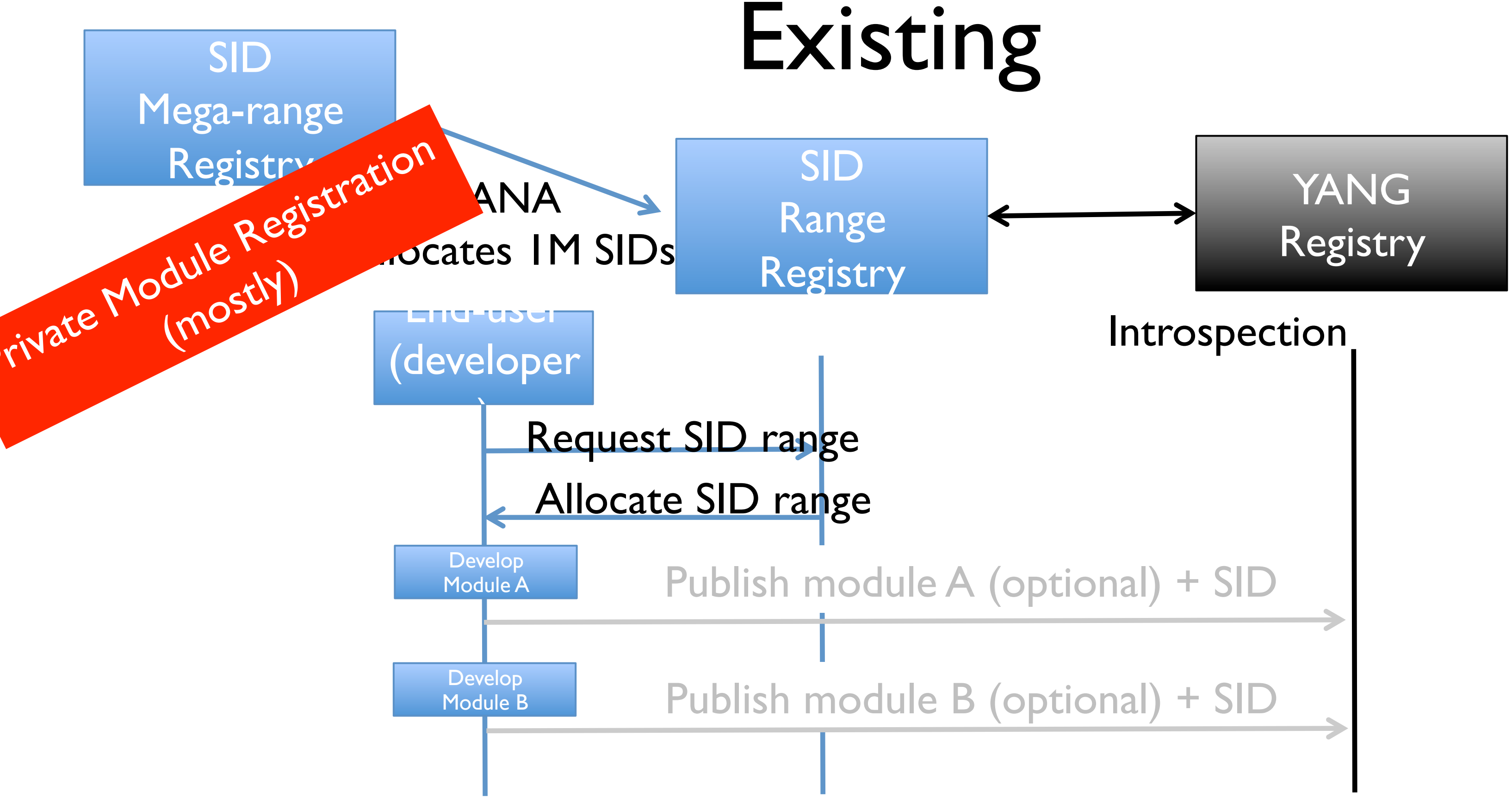
Develop Module A

Publish module A (optional) + SID

Develop Module B

Publish module B (optional) + SID

Registration procedure
Existing



SID Mega-range Registry

ANA

locates 1M SIDs

SID Range Registry

YANG Registry

rivate Module Registration (mostly)

End-user (developer

Introspection

Request SID range

Allocate SID range

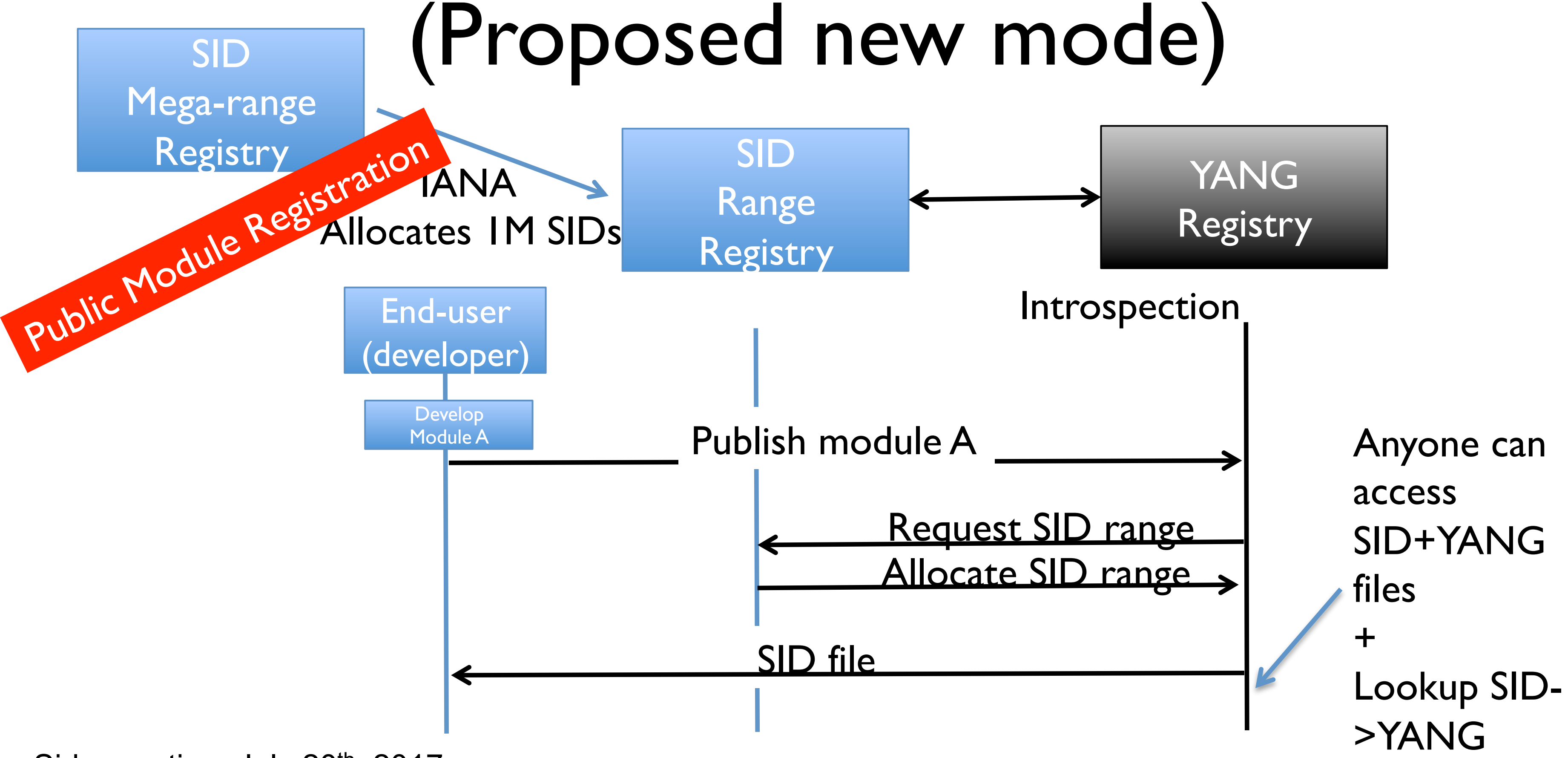
Develop Module A

Publish module A (optional) + SID

Develop Module B

Publish module B (optional) + SID

Registration procedure (Proposed new mode)



SID Mega-range Registry

Public Module Registration

IANA Allocates 1M SIDs

SID Range Registry

YANG Registry

End-user (developer)

Introspection

Develop Module A

Publish module A

Request SID range

Allocate SID range

SID file

Anyone can access SID+YANG files + Lookup SID->YANG

Side meeting, July 20th, 2017

YANG schema identifier (SID)

- Can be auto-generated and auto updated
- ietf-interfaces@2014-05-08.sid

```
{
  "assignment-ranges": [
    { "entry-point": 1500, "size": 100 }
  ],
  "module-name": "ietf-interfaces",
  "module-revision": "2014-05-08",
  "items": [
    { "type": "Module", "label": "ietf-interfaces", "sid": 1500 },
    { "type": "feature", "label": "arbitrary-names", "sid": 1501 },
    { "type": "identity", "label": "/interface-type", "sid": 1504 },
    { "type": "node", "label": "/interfaces-state", "sid": 1506 },
    { "type": "node", "label": "/interfaces-state/interface", "sid": 1507 },
  ...
  ]
}
```

Schema Mount for CoMI

Michel Veillette <michel.veillette@trilliantinc.com>

Side meeting, July 20th, 2017

Schema mount

- A "container" or "list" node becomes a mount point if the "mount-point" extension (defined in the "ietf-yang-schema-mount" module) is used in its definition.
- For example:

```
+--rw network-instances
    +--rw network-instance* [name]
       +--rw name               string
       +--rw enabled?           boolean
       +--rw description?       string
       +--rw (ni-type)?
       +--rw (root-type)?
          +--:(vrf-root)
             +--mp vrf-root?
                +--ro rt:routing-state/
                |  ...
                +--rw rt:routing/
                   ...
```

Schema mount

- New YANG construct
- Should CoMI supports schema mount?

- Possible instance-identifier CBOR encoding

  Normal data node : [sid, key1, …]

  Mounted data node : [[sid parent schema, sid mounted schema, …], key1, …]

- Possible instance-identifier URI encoding

  Normal data node : /c/sid?k="key1,…"

  Mounted data node : /c/sid parent schema/sid mounted schema/…?k="key1,…"

LWM2M-YANG
SWITCH PRESENTATIONS

Firmware Update Over The Air

Michel Veillette michel.veillette@trilliantinc.com

Alexander Pelov

Side meeting, July 20th, 2017

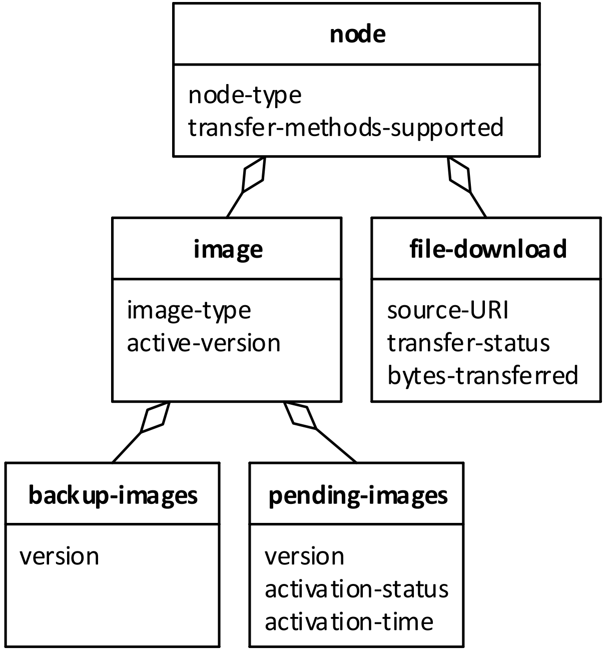
Piece of the puzzle

Coordination
Transfer protocol
Image file
Validation
Introspection
Status

Status

What is currently running on the managed node?

```
+--ro upgrade-status!
  +--ro node-type                         identity-
ref
  +--ro transfer-methods-supported* string
  +--ro image-types*
  |  +--ro image-type                     identity-
ref
  |  +--ro active-version                 string
  |  +--ro backup-images*
  |  |  +--ro version                     string
  |  +--ro pending-images*
  |     +--ro version                     string
  |     +--ro activation-status           enumeration
  |     +--ro activation-time             uint32
  +--ro file-download*
     +--ro source-URI                     string
     +--ro transfer-status                enumeration
     +--ro bytes-transferred              uint32
```



node

node-type
transfer-methods-supported

image

image-type
active-version

file-download

source-URI
transfer-status
bytes-transferred

backup-images

version

pending-images

version
activation-status
activation-time

Introspection

What should be running?

```
+--ro manifest
   +--ro image-file*
      +--ro node-type                           identity-ref
      +--ro firmware-images*
      |  +--ro image-type                       identity-ref
      |  +--ro image-version?                   string
      |  +--ro compatibility*
      |  |  +--ro compatible-version            string
      |  |  +--ro synchronized-upgrade-required? boolean
      |  |  +--ro rollback-supported?           boolean
      |  +--ro image-hash                       binary
      +--ro firmware-signing-certificate*       binary
      +--ro signature?                          binary
```

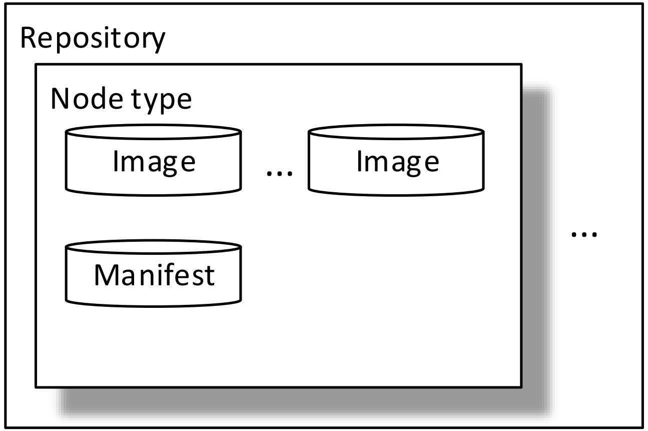


Image file

- Unmodified
  - To avoid any incompatibility and/or complexity
  - File as produced by the tool chain
  - Validation and security (e.g signature) not added to this file

Transfer protocol

- Many need to be supported
  - coap block, coaps block
  - coap+tcp, coap+tls
  - http, https
  - ftp
  - Ipv6 multicast
- Other may be added later without updating the specification.

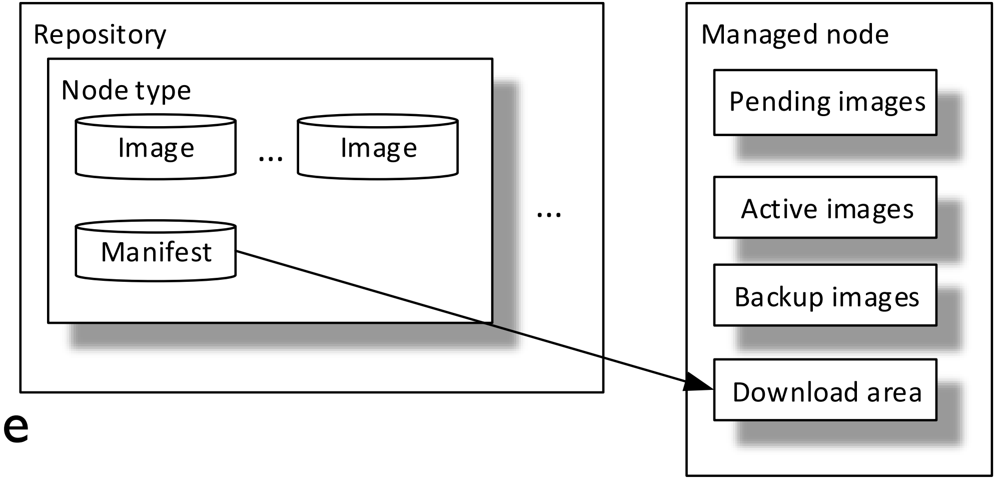
Coordination

```
rpcs:
   +---x initiate-transfer
   |  +---w input
   |     +---w file*
   |        +---w source-URI       string
   |        +---w length           uint64
   +---x fragment
   |  +---w input
   |     +---w byte-offset         uint64
   |     +---w last-fragment       boolean
   |     +---w file-image-fragment binary
   +---x cancel
   +---x activate
      +---w input
         +---w activation-time     uint32
```

```
notifications:
   +---n transfer-status
   |  +--ro source-URI                   string
   |  +--ro status                       enumeration
   +---n activation-result
      +--ro firmware-images*
         +--ro image-type                identity-
ref
         +--ro (result)
            +--:(activation-success)
            |  +--ro current-version     string
            |  +--ro previous-version    string
            |  +--ro success-reason      enumeration
            +--:(activation-failure)
               +--ro version             string
               +--ro failure-reason      enumeration
```

Validation

- How the manager and the managed node known?
  - If the set of images are compatible?
  - If the set of images are valid and authentic?



Manifest information

Is sent to the managed node

for upgrade validation.

Event logger and notifications

Andy Bierman

Michel Veillette <michel.veillette@trilliantinc.com>

Peter van der Stok

Alexander Pelov

Side meeting, July 20th, 2017

Event logger – Why needed

- CoMI don't support access to prior notifications
  (This a requirement is some legislations, some markets)
- Which notifications are available in which event streams or loggers
  Need to be configurable
- A mechanism is required to add notification metadata to streams or loggers
  e.g. timestamp, security, facility, signature, …

Event logger – Data model

```
module: trilliant-event-logger
   +--rw event-loggers* [name]
   |  +--rw name                 uint32
   |  +--rw notifications* [sequence-number]
   |  |  +--rw sequence-number   uint32
   |  |  +--rw timestamp?        timestamp
   |  |  +--rw notification?
   |  +---x get-notifications
   |     +---w input
   |     |  +---w (creteria)?
   |     |     +--:(sequence-number)
   |     |     |  +---w start-sequence-number? uint32
   |     |     |  +---w end-sequence-number?   uint32
   |     |     +--:(timestamp) {timestamp}?
   |     |        +---w start-timestamp?       timestamp
   |     |        +---w end-timestamp?         timestamp
   |     +--ro output
   |        +--ro notifications* [sequence-number]
   |           +--ro sequence-number           uint32
   |           +--ro timestamp?                timestamp
   |           +--ro notification?
```

```
+--rw reporting-control* [handler]
|  +--rw handler         string
|  +--rw notification*   SID
|
|
+--rw meta-info-control* [handler]
   +--rw handler         string
   +--rw meta-info*      SID
```

YANG for Static Context Header Compression (SCHC)

Laurent Toutain - Ana Minaburo

Laurent.Toutain@imt-atlantique.fr
ana@ackl.io

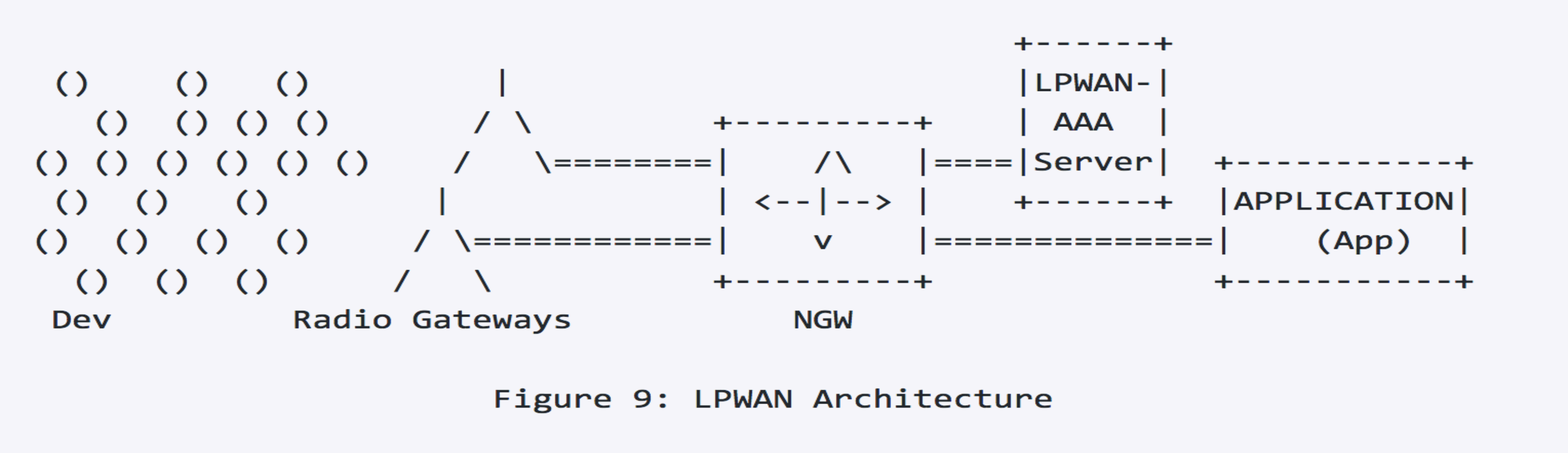
IETF 99 - Prague

draft-toutain-lpwan-yang-static-context-hc-00

Side meeting, July 20th, 2017

Architecture

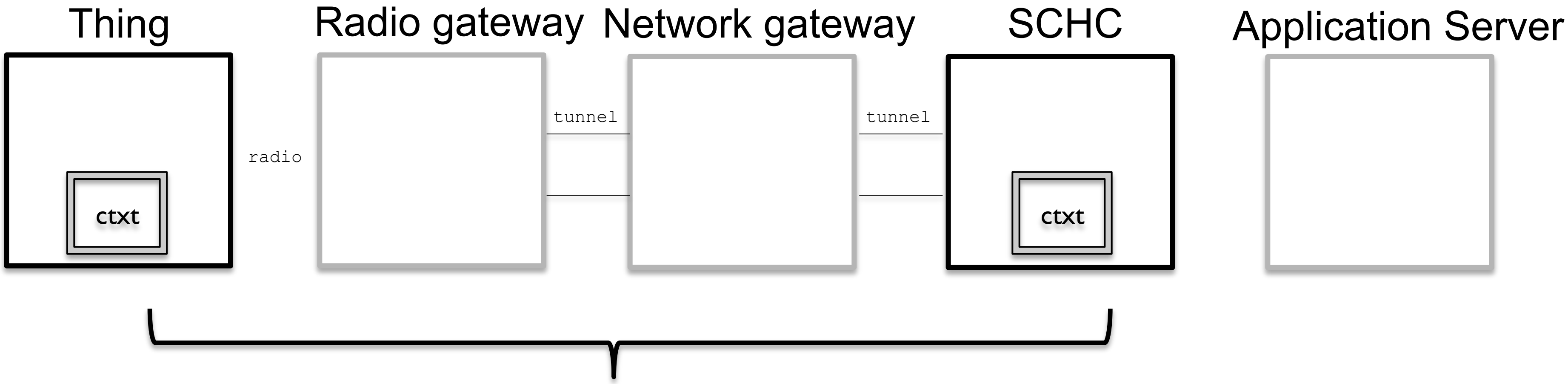
- Define common terminology
  - – Dev, NGW, App



```
                                                 +------+
 ()    ()   ()          |                         |LPWAN-|
   ()  () () ()       / \         +---------+    | AAA  |
() () () () () ()    /   \========|    /\   |====|Server|  +-----------+
 ()  ()   ()        |             | <--|--> |    +------+  |APPLICATION|
()  ()  ()  ()     / \============|    v    |==============|    (App)  |
  ()  ()  ()      /   \           +---------+              +-----------+
 Dev         Radio Gateways           NGW
```

Figure 9: LPWAN Architecture

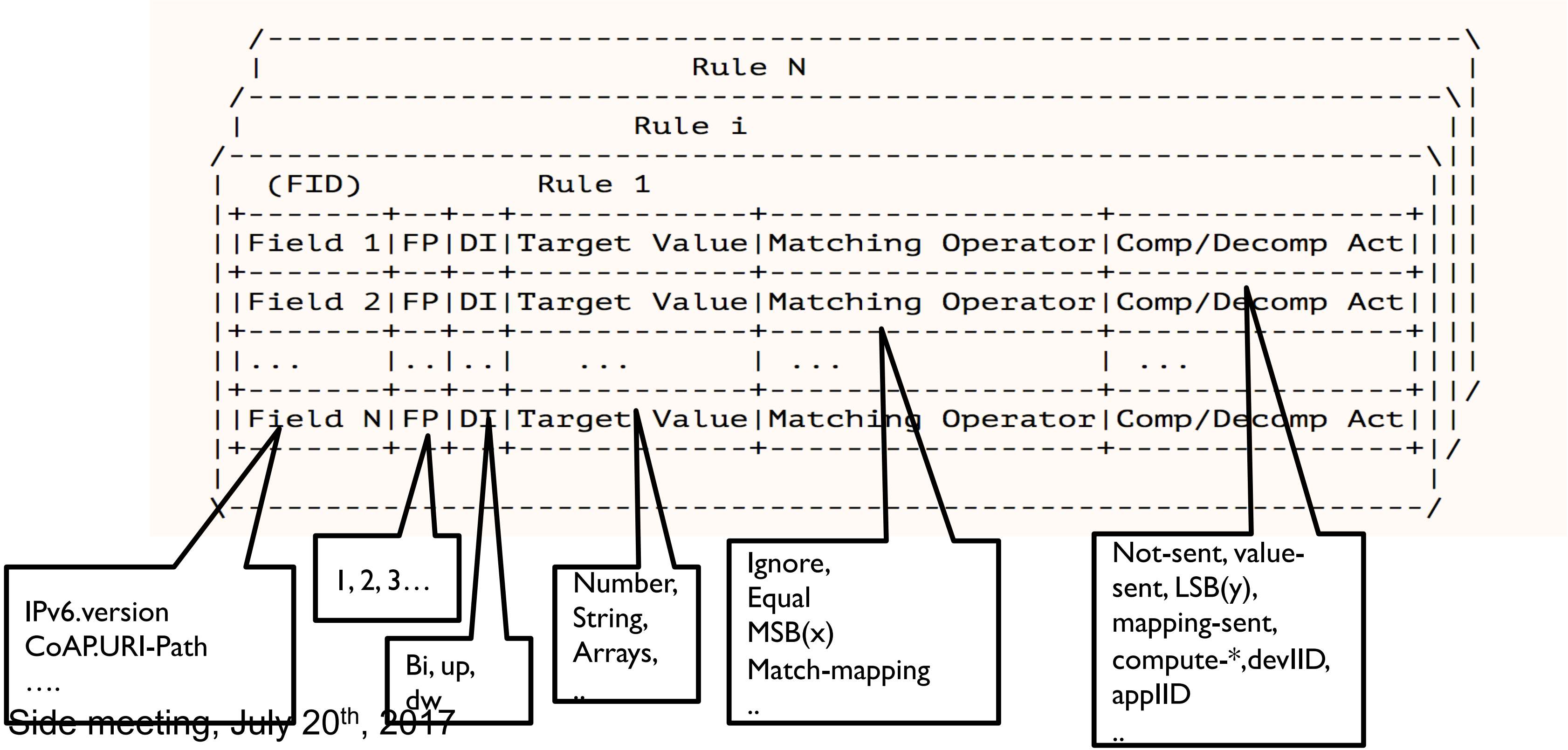
Target Architecture



No IP forwarding

Side meeting, July 20th, 2017

New fields



```
  /-----------------------------------------------------------------\
  |                         Rule N                                  |
 /-----------------------------------------------------------------\|
 |                       Rule i                                    ||
/-----------------------------------------------------------------\||
|  (FID)            Rule 1                                        |||
|+-------+--+--+------------+-----------------+---------------+|||
||Field 1|FP|DI|Target Value|Matching Operator|Comp/Decomp Act||||
|+-------+--+--+------------+-----------------+---------------+|||
||Field 2|FP|DI|Target Value|Matching Operator|Comp/Decomp Act||||
|+-------+--+--+------------+-----------------+---------------+|||
||...    |..|..|   ...      | ...             | ...           ||||
|+-------+--+--+------------+-----------------+---------------+||/
||Field N|FP|DI|Target Value|Matching Operator|Comp/Decomp Act|||
|+-------+--+--+------------+-----------------+---------------+|/
|                                                               |
\---------------------------------------------------------------/
```

IPv6.version
CoAP.URI-Path
….

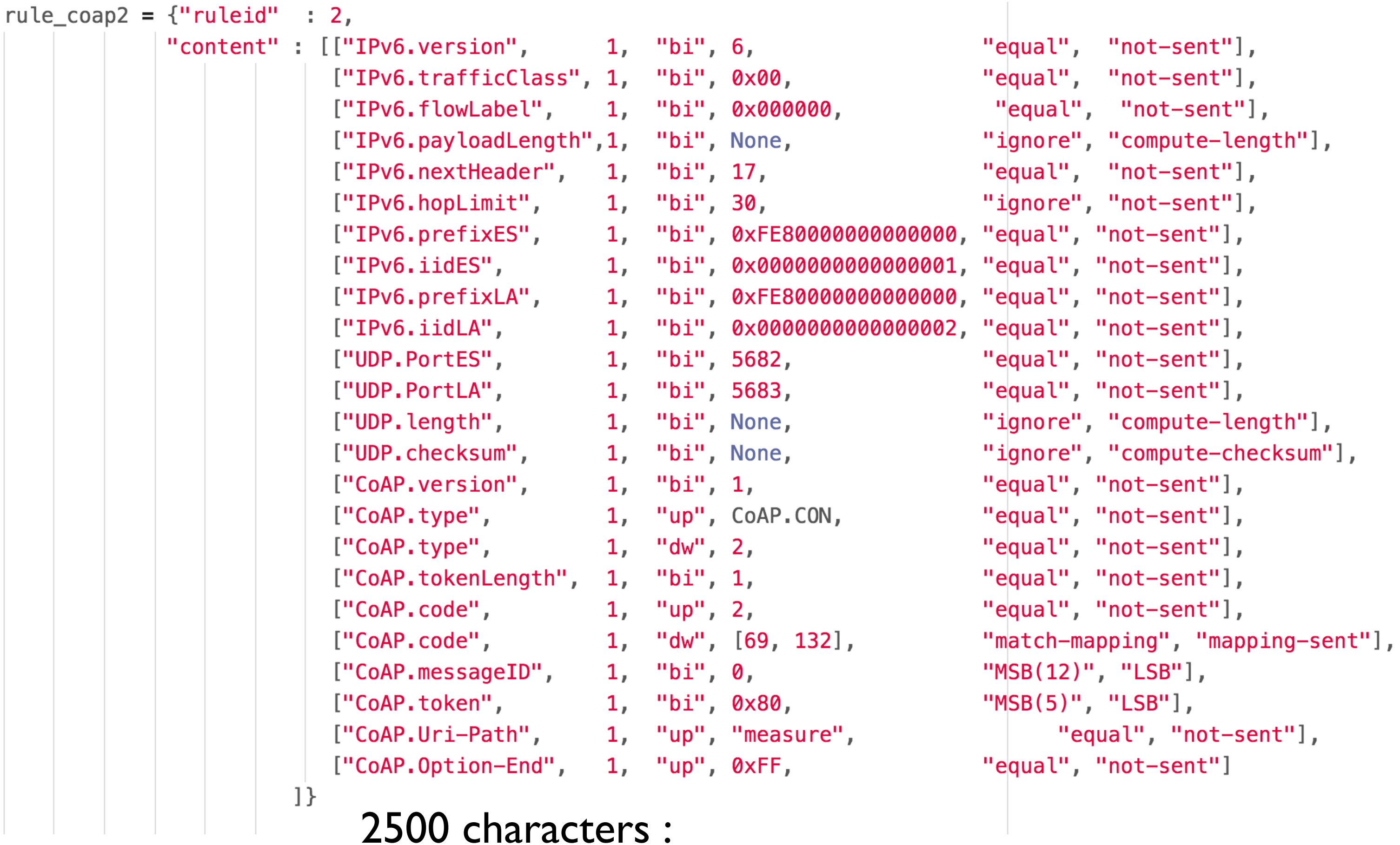
1, 2, 3…

Bi, up, dw

Number,
String,
Arrays,
..

Ignore,
Equal
MSB(x)
Match-mapping
..

Not-sent, value-sent, LSB(y),
mapping-sent,
compute-*,devIID,
appIID
..



```
rule_coap2 = {"ruleid"  : 2,
             "content" : [["IPv6.version",      1,  "bi", 6,                  "equal",  "not-sent"],
                          ["IPv6.trafficClass", 1,  "bi", 0x00,               "equal",  "not-sent"],
                          ["IPv6.flowLabel",    1,  "bi", 0x000000,            "equal",  "not-sent"],
                          ["IPv6.payloadLength",1,  "bi", None,               "ignore", "compute-length"],
                          ["IPv6.nextHeader",   1,  "bi", 17,                 "equal",  "not-sent"],
                          ["IPv6.hopLimit",     1,  "bi", 30,                 "ignore", "not-sent"],
                          ["IPv6.prefixES",     1,  "bi", 0xFE80000000000000, "equal", "not-sent"],
                          ["IPv6.iidES",        1,  "bi", 0x0000000000000001, "equal", "not-sent"],
                          ["IPv6.prefixLA",     1,  "bi", 0xFE80000000000000, "equal", "not-sent"],
                          ["IPv6.iidLA",        1,  "bi", 0x0000000000000002, "equal", "not-sent"],
                          ["UDP.PortES",        1,  "bi", 5682,               "equal", "not-sent"],
                          ["UDP.PortLA",        1,  "bi", 5683,               "equal", "not-sent"],
                          ["UDP.length",        1,  "bi", None,               "ignore", "compute-length"],
                          ["UDP.checksum",      1,  "bi", None,               "ignore", "compute-checksum"],
                          ["CoAP.version",      1,  "bi", 1,                  "equal", "not-sent"],
                          ["CoAP.type",         1,  "up", CoAP.CON,           "equal", "not-sent"],
                          ["CoAP.type",         1,  "dw", 2,                  "equal", "not-sent"],
                          ["CoAP.tokenLength",  1,  "bi", 1,                  "equal", "not-sent"],
                          ["CoAP.code",         1,  "up", 2,                  "equal", "not-sent"],
                          ["CoAP.code",         1,  "dw", [69, 132],          "match-mapping", "mapping-sent"],
                          ["CoAP.messageID",    1,  "bi", 0,                  "MSB(12)", "LSB"],
                          ["CoAP.token",        1,  "bi", 0x80,               "MSB(5)", "LSB"],
                          ["CoAP.Uri-Path",     1,  "up", "measure",                "equal", "not-sent"],
                          ["CoAP.Option-End",   1,  "up", 0xFF,               "equal", "not-sent"]
                       ]}
```

2500 characters :

- sigfox 250 frame,
- LoRaWAN : 200 frames

Why COMI

- Compact and unique representation for:
  - – Well-known Field ID, MO, CDA
- Compact exchanges between SCHC C/D
  - – NGW-> dev: Prefix assignment
  - – Dev-> NGW: setup port numbers, destination addresses,…

ietf-lpwan-schc.yang

```
+__rw compression_context
   +__rw context_rules* [rule_id]
      +__rw rule_id      uint8
      +__rw rule_fields* [name occurence direction]
        +__rw name                              string
        +__rw occurence                              uint8
        +__rw direction                            enumeration
        +__rw target_value?                            lpwan_types
        +__rw matching_operator?                    matching_operator_type
        +__rw matching_operator_parameter?           lpwan_types
        +__rw compression_decompression_function?        compression_decompression_
function_type
        +__rw compression_decompression_function_parameter?  lpwan_types
```

SWITCH PRESENTATION
Embedded

SWITCH PRESENTATION
MUD

Discussion

- -What other points could YANG address for IoT?
- - Interaction models for YANG and IoT
- - Intersection between YANG and IoT communities
- - Getting YANG out of the routers
- - Getting efficient technologies to routers (from the constrained world)
- - Events, notifications, security, …

Next steps?