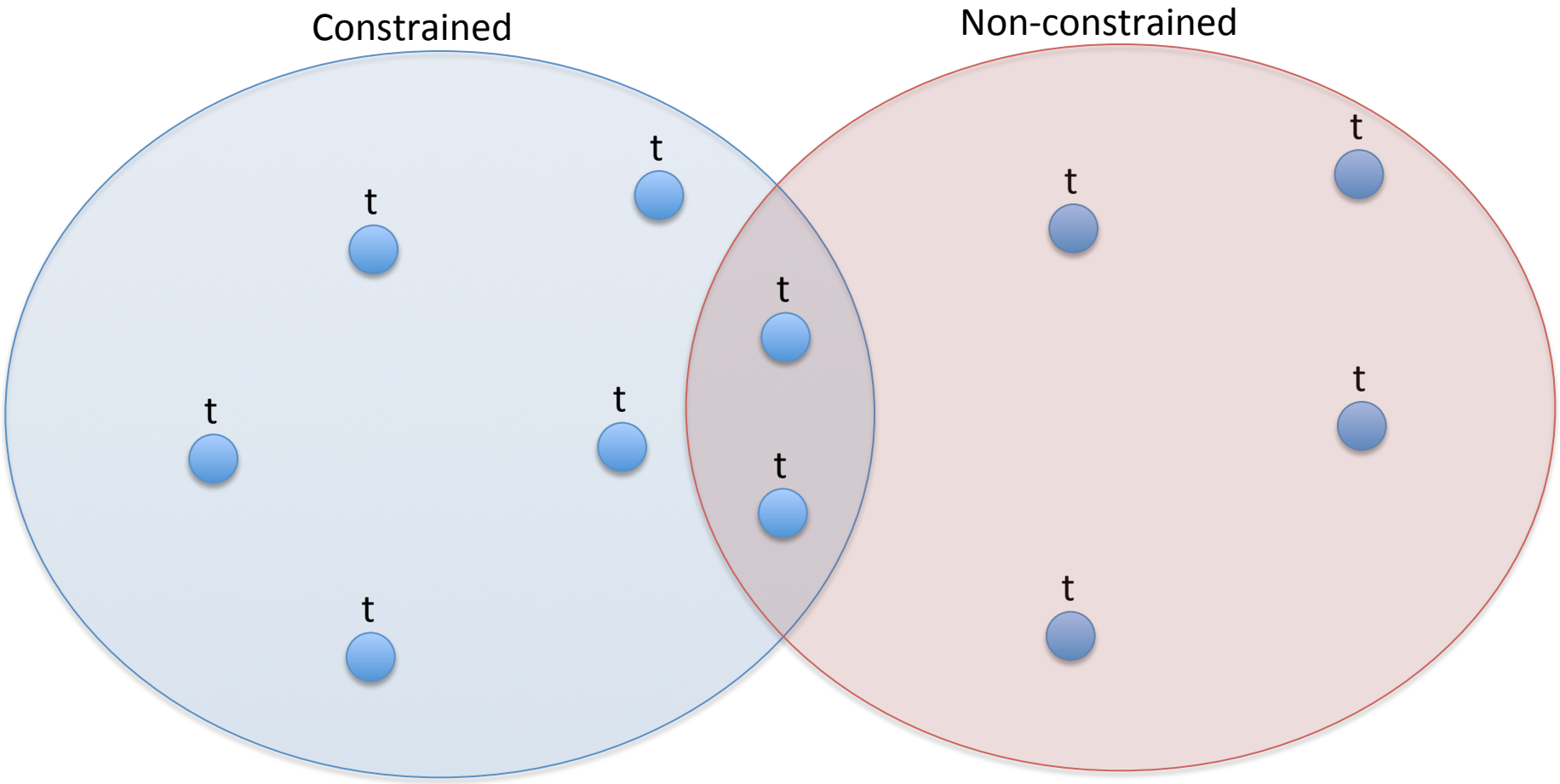
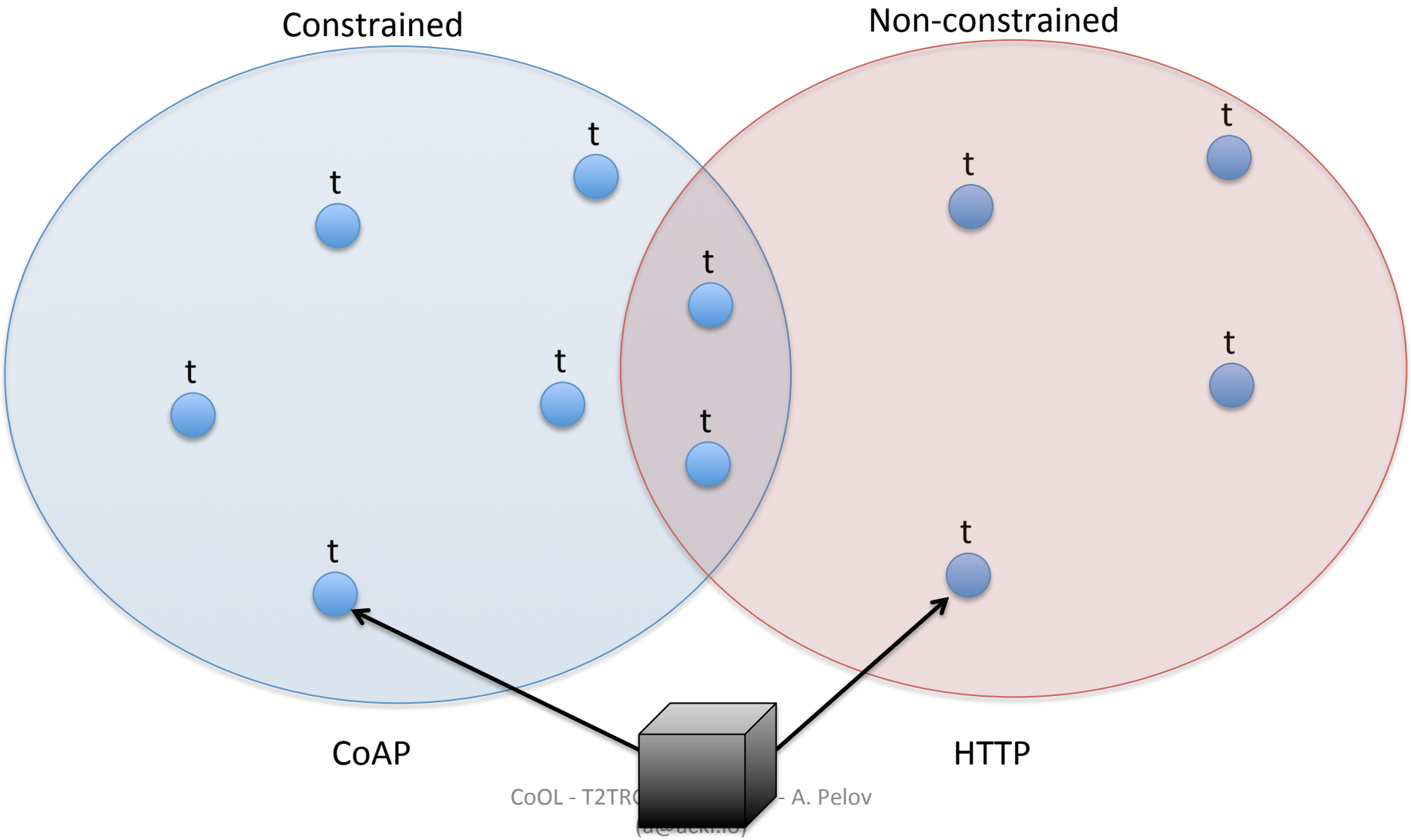
Constrained Objects Language CoOL

Michel Veillette
Alexander Pelov (a@ackl.io)
Abhinav Somaraju
Randy Turner

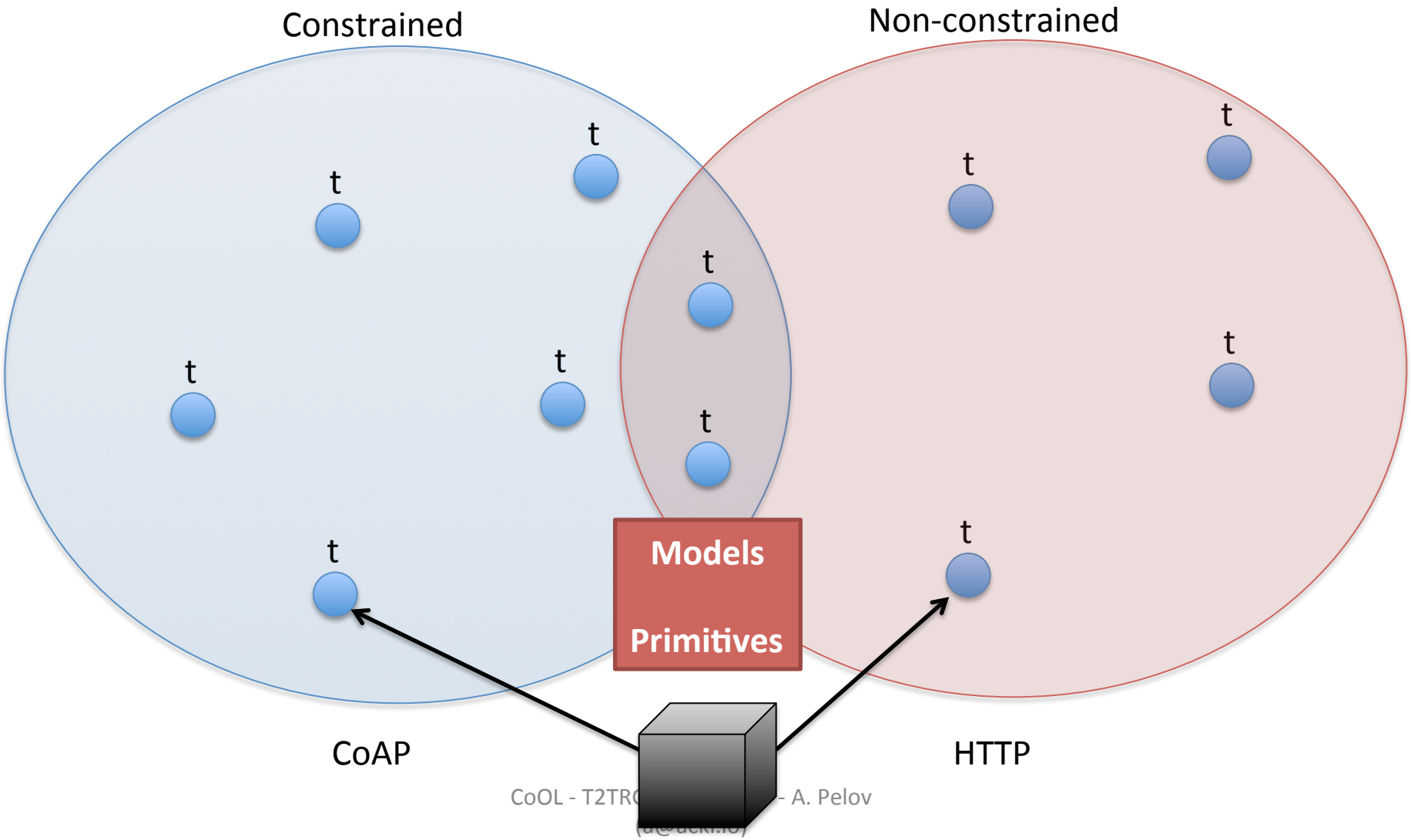
You want to manage things



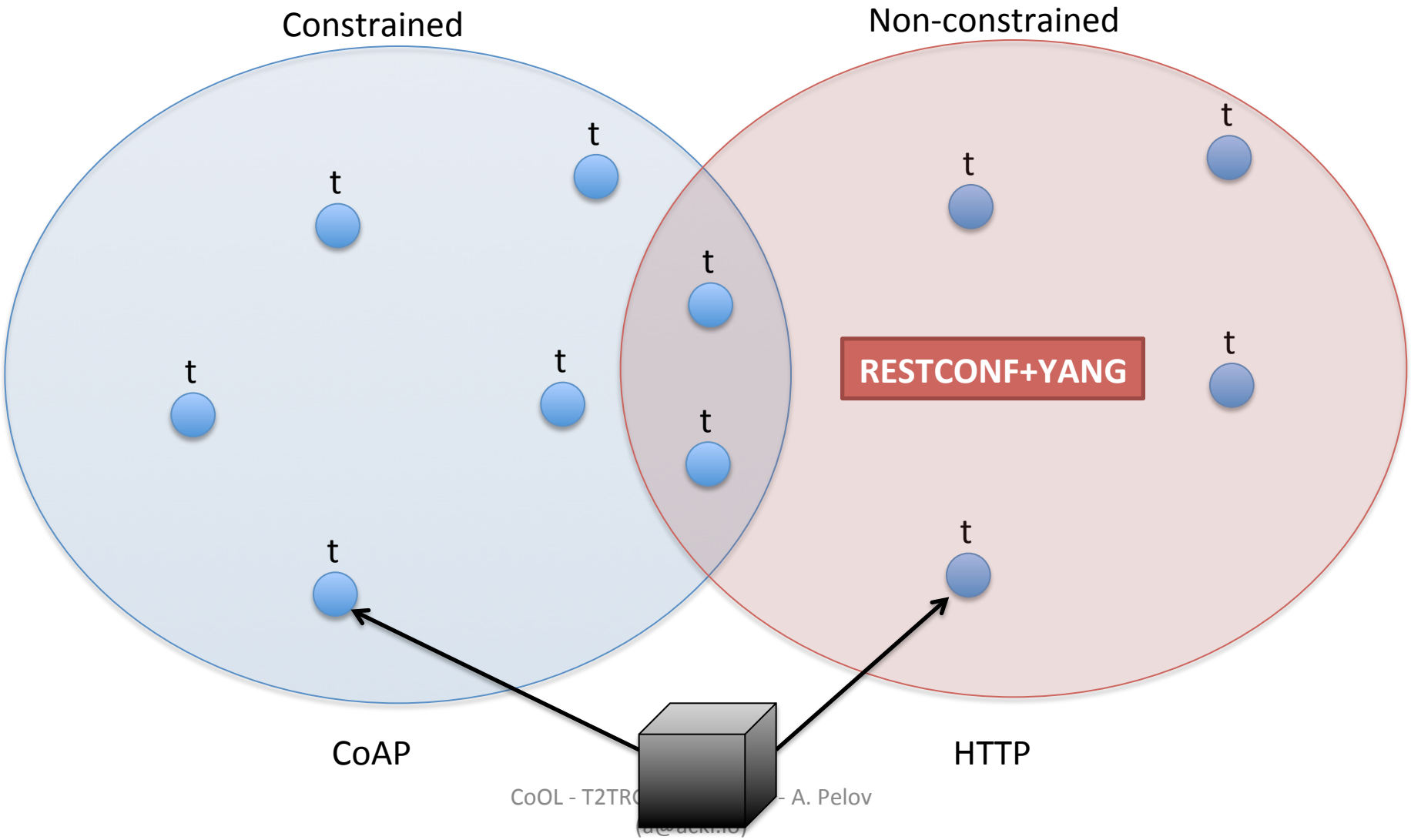
You want to manage things



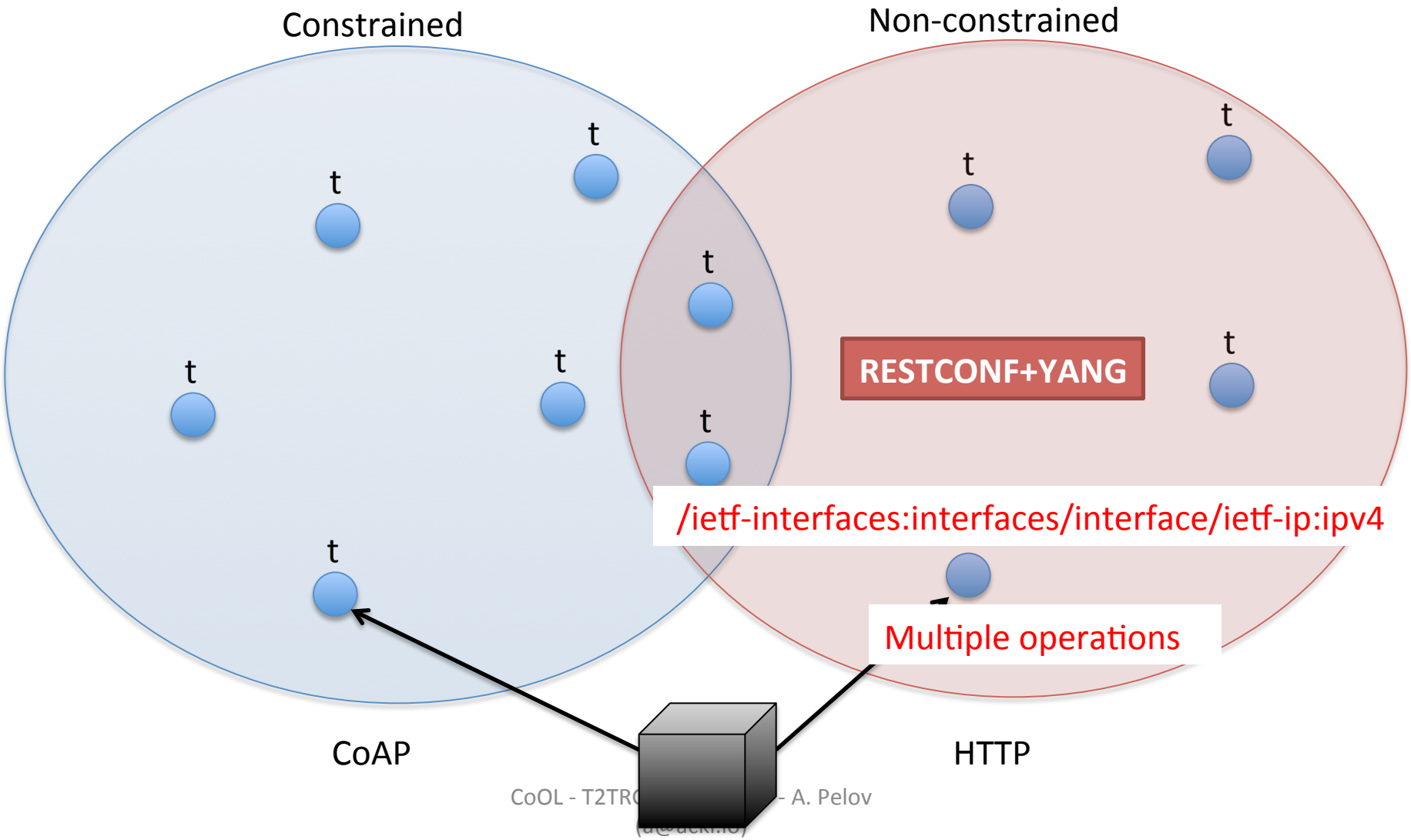
You want to manage things



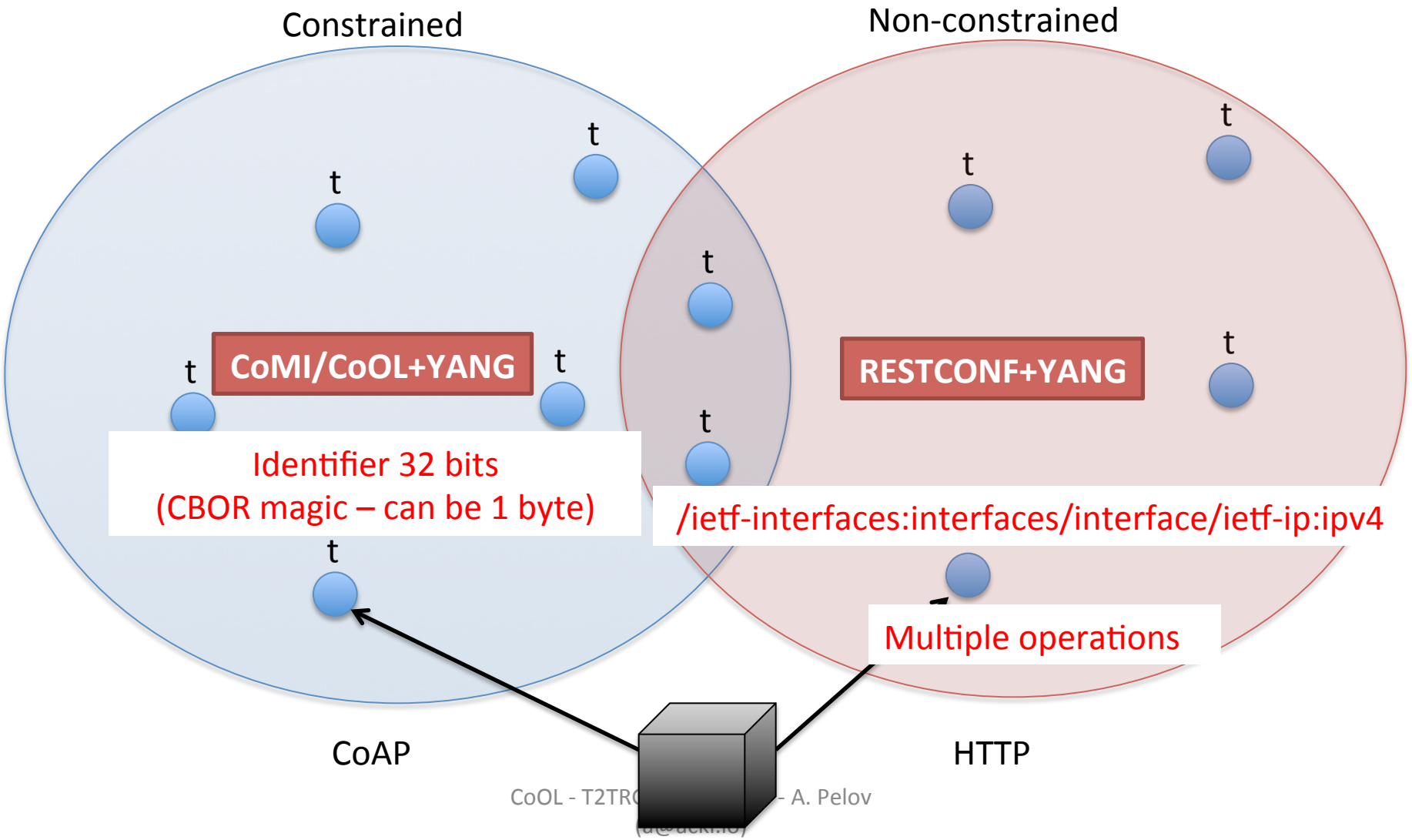
You want to manage things



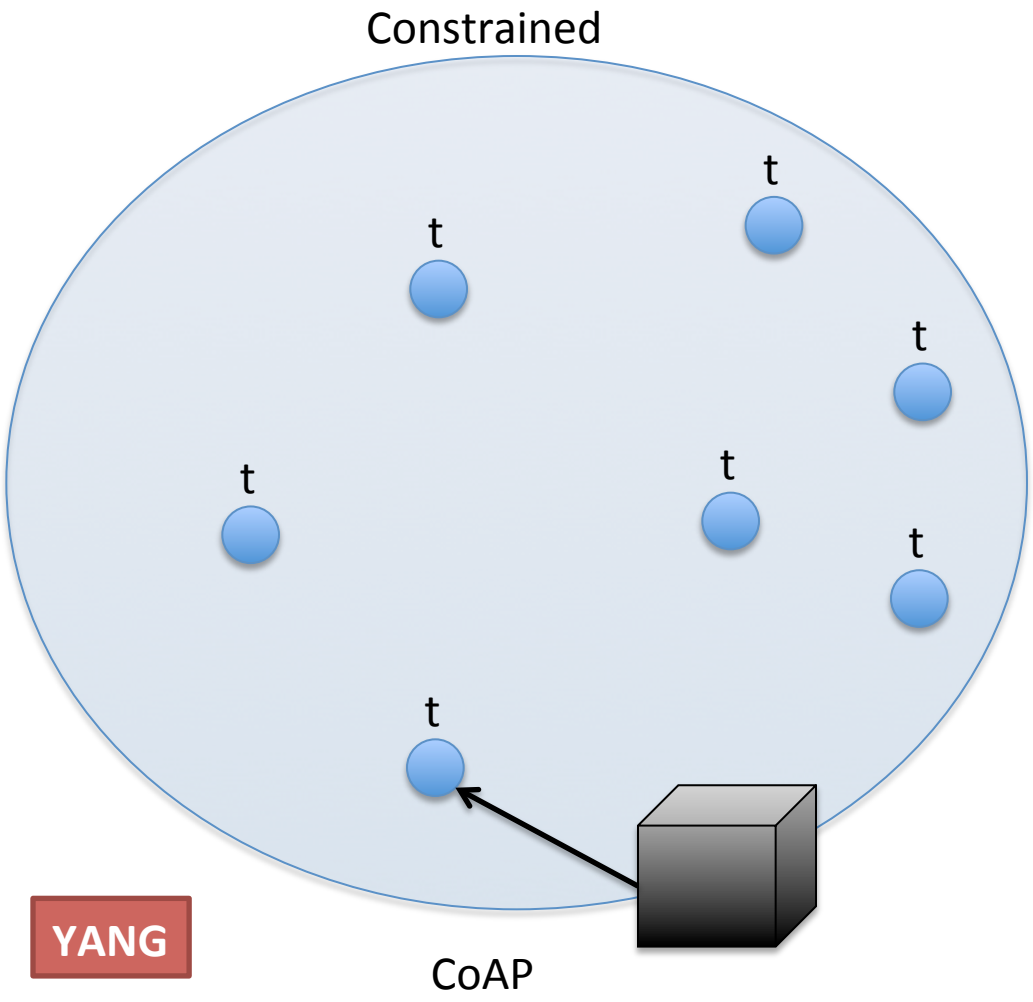
You want to manage things



You want to manage things



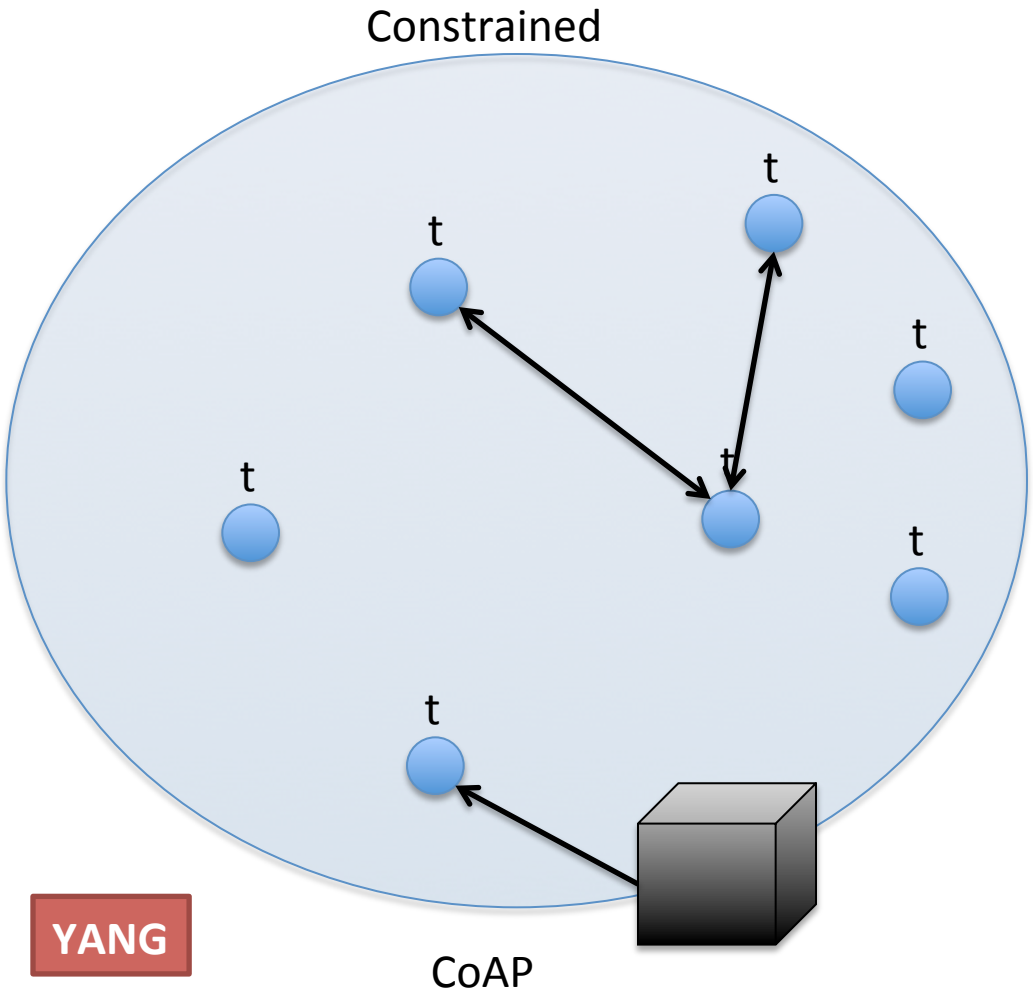
You want to manage things



I want to manage LPWAN (LR-WAN)

- 10 000 devices per antenna
- 50 kbps max (can be 270 bps)
- 1-10% duty cycle

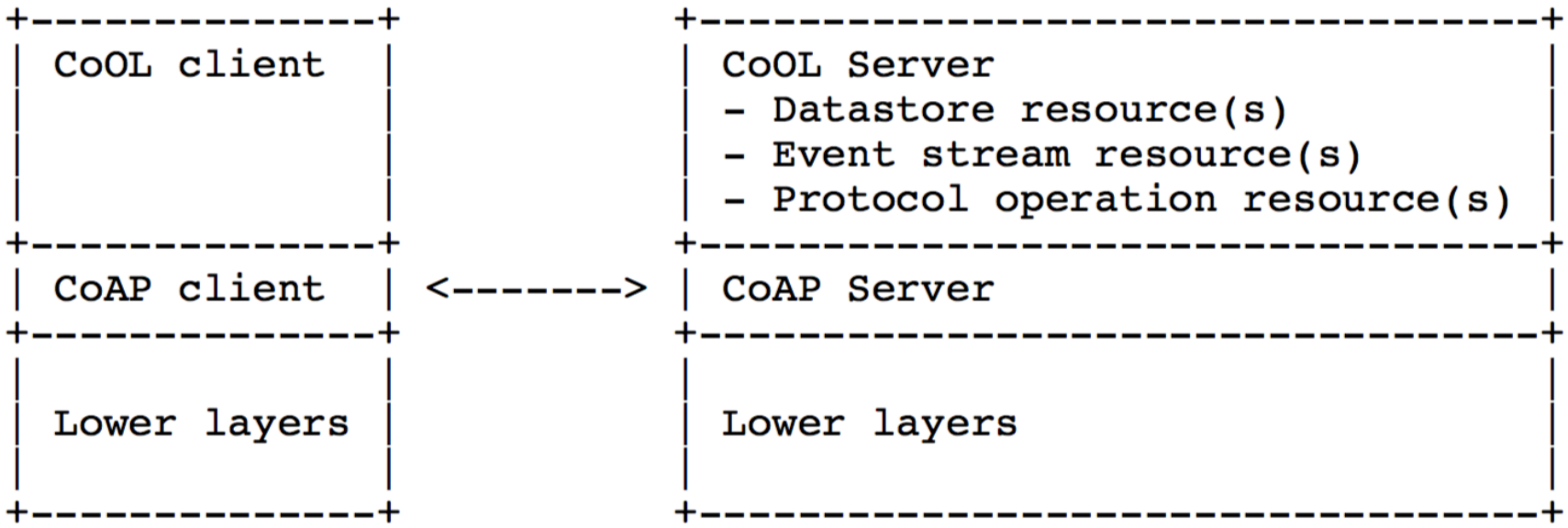
You want to manage things



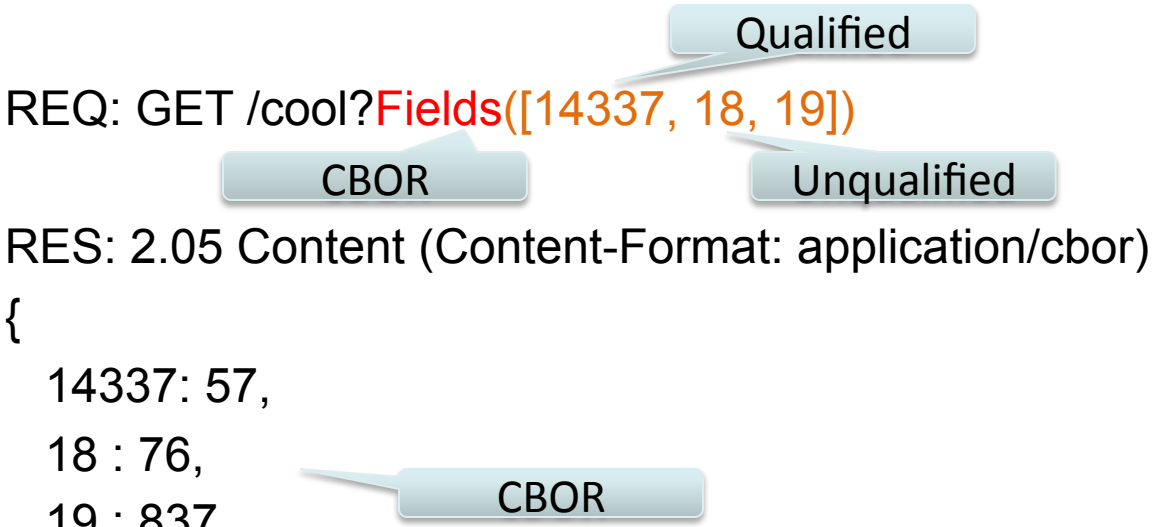
I want to manage LPWAN (LR-WAN)
- 10 000 devices per antenna
- 50 kbps max (can be 270 bps)
- 1-10% duty cycle

T2T management
- Thread / ZigBee / Other

Architecture



- Perform on a single resource (e.g. GET /cool)
- *"Fields"* option contains the list of nodes selected, encoded using a CBOR array



CoMI vs CoOL - Identifiers

	CoMI	**CoOL**
30 bits	Unmanaged **Hash(long identifier)**	Managed **Module ID (20 bit) + Node ID (10 b)** -> NID is LSB

CoMI vs CoOL - Identifiers

	CoMI	CoOL
30 bits	Unmanaged **Hash(long identifier)** Collisions - Re-hashing - Handling thousands of nodes puts strain on client (learn re-hashes and hashes for each dev) - Dynamic module loading – rehashing in a retrieved tree can be painful	Managed **Module ID (20 bit) + Node ID (10 b)** -> NID is LSB Automatically allocated (can be manual) Central repository for modules - Need IANA action - Distributed scenarios (t2t!) - Lots of magic (*module discovery*,…) ID of DATA, EVENTS and RPCs !

CoMI vs CoOL - Identifiers

	CoMI	CoOL
30 bits	Unmanaged **Hash(long identifier)** Collisions - Re-hashing - Handling thousands of nodes puts strain on client (learn re-hashes and hashes for each dev) - Dynamic module loading – rehashing in a retrieved tree can be painful	Managed **Module ID (20 bit) + Node ID (10 b)** -> NID is LSB Automatically allocated (can be manual) Central repository for modules - Need IANA action - Distributed scenarios (t2t!) - Lots of magic (*module discovery*,…) ID of DATA, EVENTS and RPCs !
URI	BASE64 mapping 30 bits -> 5 URI-safe characters	CoAP option CBOR encoded (1-5 bytes, mostly 3 bytes)
Body	CBOR encoded (5 bytes)	CBOR encoded (1-5 bytes, mostly 3 bytes)
Collection of nodes	Not supported	CBOR array Optimized

Conclusion CoOL

- Managed IDs
  - Data nodes, notification streams, protocol operations (RPC)
- RESTful collections
  - Use of binary option -> optimal representation
- Explicit PATCH (vs diff-like in CoMI)
- Use CoOL to manage applications
- Next steps
  - Use of deterministic multimaps vs maps
  - Multicast for application management
    - E.g. turn all lights on a controller

BACKUP SLIDES

URIs

CoMI

/mg/XXXXX
/mg/stream

CoOL

- Default endpoint + CoAP « Fields » option
  /cool + Fields(ID)
  /cool/rpc + Fields(ID)
  /cool/stream + Fields(ID) (optional)

- Endpoint 0, 1, … (optional)
  /cool/ep0, /cool/ep1, …
  /cool/ep0/rpc, /cool/ep1/rpc, …
  /cool/ep0/stream, /cool/ep0/stream …

Nodes, collections and filtering

CoMI

/mg/XXXXX?***keys=****keyValue1,keyValue2,…*&***select=****YYYYY(value1, value2, value3, …)*

CoOL

- Fields:

  CBOR integer = single data node

  0x11023

  CBOR array = more than one data nodes

  [0x11023, 0x24, 0x25, 0x26]

  -> if an item is an array, then – key + search

  [0x11023, 0x24, **[0x25, [keyValue1, keyValue2,…], [value1, value2, …]]**, 0x26]

Example: Part of a YANG module

```
typedef date-and-time {
    type string {
      pattern ’\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?(Z|[\+\-]\d{2}:\d{2})’;
    }
}

container clock {
    leaf current-datetime {
      type date-and-time;
    }

    leaf boot-datetime {
         type date-and-time;
    }
}
```

CoMI

/ietf-system:system-state/clock (0x021ca491 and CDKSQ)
/ietf-system:system-state/clock/current-datetime (0x047c468b)
/ietf-system:system-state/clock/boot-datetime (0x1fb5f4f8)

REQ: GET example.com/mg/CDKSQ
(Content-Format: application/cbor)

RES: 2.05 Content (Content-Format: application/cbor)

```
{
  0x021ca491 : {
     0x047c468b : "2014-10-26T12:16:51Z",
     0x1fb5f4f8 : "2014-10-21T03:00:00Z"
   }
}
```

CoOL

ietf-system module ID=0x44
/ietf-system:system-state/clock (DNID=0x23, FQDNID=0x11023)
/ietf-system:system-state/clock/current-datetime (DNID=0x24)
/ietf-system:system-state/clock/boot-datetime (DNID=0x25)

REQ: GET example.com/cool *Fields(0x11023)*
(Content-Format: application/cbor)

RES: 2.05 Content (Content-Format: application/cbor)

```
{
  0x11023 : {
    0x24 : "2015-10-02T14:47:24Z-05:00",
    0x25: "2015-09-15T09:12:58Z-05:00"
  }
}
```

CoOL – two data nodes

ietf-system module ID=0x44
/ietf-system:system/clock (DNID=0x05, FQDNID=0x11005)
/ietf-system:system/clock/timezone/timezone-utc-offset/timezone-utc-offset (DNID=0x07)
/ietf-system:system-state/clock/current-datetime (DNID=0x24, FQDNID=0x11024)

REQ: **GET /cool *Fields([0x11005, 55])***

RES: 2.05 Content (Content-Format: application/cbor)

```
{
   0x11005: {
     0x7 : 540
   },
   0x11024 : "2015-10-08T14:10:08Z09:00”
}
```

CoOL – filtering + search

CoAP request:
GET /cool *Fields([69635, [66562, ["eth0"], [5, 6]])*

CoAP response:
2.05 Content Content-Format(application/cbor)
{
69635 : "datatracker.ietf.org",
66562 : {
5 : "iana-interface-type.ethernetCsmacd",
6 : true
}
}

OR

2.05 Content Content-Format(application/cbor)
{
69635 : "datatracker.ietf.org",
66562 : undefined
}

URI question

- How do you modify a collection of objects
  - PATCH in CoMI
    - Someday FETCH
    - Cannot do PUT/POST/DELETE on a collection
  - PATCH in CoOL
    - FETCH

- FETCH + PATCH = REST ?

CoMI vs CoOL PATCH

- CoMI = run a diff
  - New entries are added, existing are updated, null-ed are deleted

- CoOL= YANG module
  - patch-request + modification type + data node + value
    - Delete, merge, replace, remove, insert-first, insert-last, insert-before, insert-after
    - List entry
    - List value

CoOL PATCH vs CoMI PATCH

CoAP request:
PATCH /cool Content-Format(app/cbor)

```
{
  1028 : [
    {
      5 : 0, 6 : 4, 7 : [69642, [null]]
    },
    {
      5 : 1, 6 : 0, 8: {
        69642 : {
          11 : "NTP Pool server 2",
          12 : {
            13 : "2.pool.ntp.org"
          }
    } }
    }, {
      5 : 2, 6 : 3, 8: {69641 : true }
    } ]
}
```

```
{
     "B":  {
      { "key1" : "author1",
       "key2" : "book2"}:
       { null : null},
      { "key1" : "author5"} :
        {"counter1" : 4444},
      { "key1" : "newauthor",
       "key2" : "newbook"}:
      { "col1" : 1,
        "counter1" : 1}
      },
    "book" : {
      "title" : "favoured",
      "author": {"familyName" : null},
      "tags" : [ "example"],
      "phoneNumber" : "+01-123-456-7890"
    }
}
```

CoOL protocol operations (aka RPC)

module ietf-system: module ID 68
rpc set-current-datetime: protocol operation ID 1
input parameter current-datetime: input parameter ID 1

CoAP request:
POST /cool/rpc/68/1 Content-Format(application/cbor)
{
  1 : "2015-10-08T14:10:08Z09:00”
}

CoAP response:
2.05 Content
…

Questions

- How do we identify items?
  - Data nodes, event streams, protocol operations (RPC)
  - Managed vs Unmanaged
- How do we identify a collection of items?
  - Query parameters, FETCH method
    - How do we PUT/POST/DELETE them? PATCH all?
  - CoAP option
- How do we PATCH?
  - Diff vs explicit operations
- How do we represent requests/responses?
  - Death of the JSON object (CBOR map) as a main carrier
- How do we manage applications on devices?

But wait, there is more…

- Application management
  - Endpoints
  - Application multicast
    - Do not confuse with network multicast
- Deterministic MultiMaps instead of objects
  - (or Ordered Pairs)