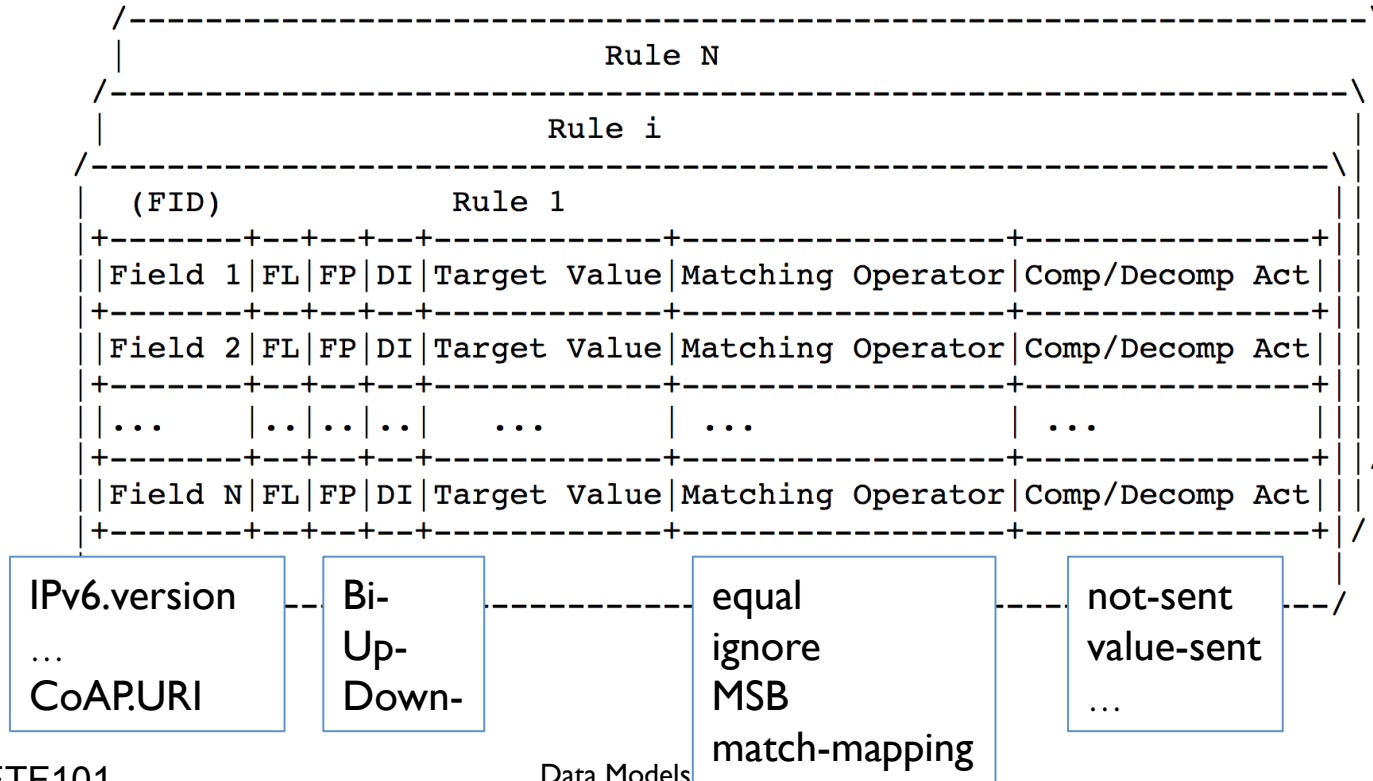


Data Models for SCHC

Authors:

Alexander Pelov <a@ackl.io>

Why?



```

rule_coap0 = {"ruleid" : 0,
"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, "bi", 0, "equal", "not-sent"],
  ["CoAP.tokenLength", 1, "bi", 1, "equal", "not-sent"],
  ["CoAP.code", 1, "bi", 2, "equal", "not-sent"],
  ["CoAP.messageID", 1, "bi", 1, "MSB(4)", "LSB"],
  ["CoAP.token", 1, "bi", 0x01, "MSB(4)", "LSB"],
  ["CoAP.Uri-Path", 1, "up", "foo", "equal", "not-sent"],
  ["CoAP.Uri-Path", 2, "up", "bar", "ignore", "value-sent"],
}]

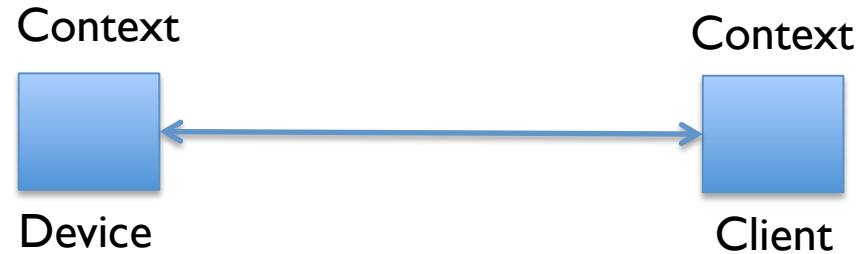
```

```

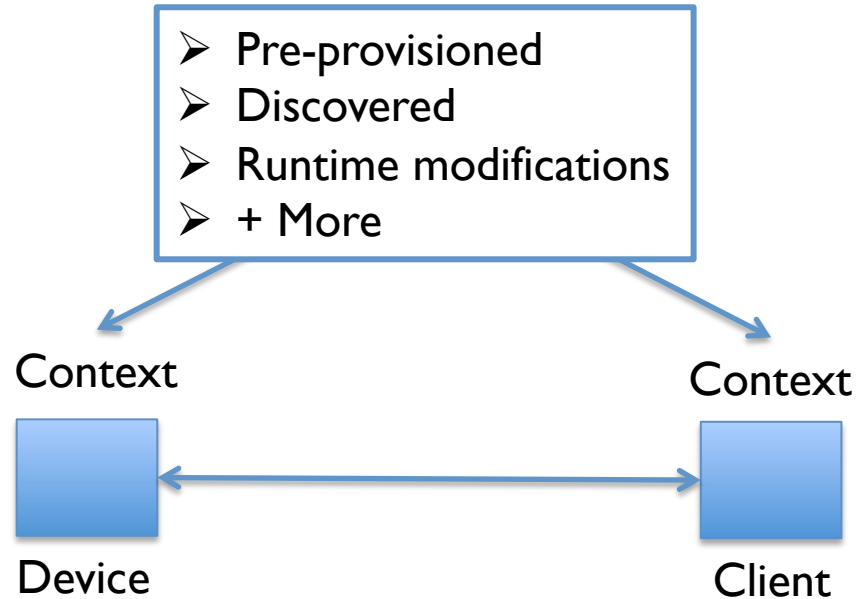
  ]}

```

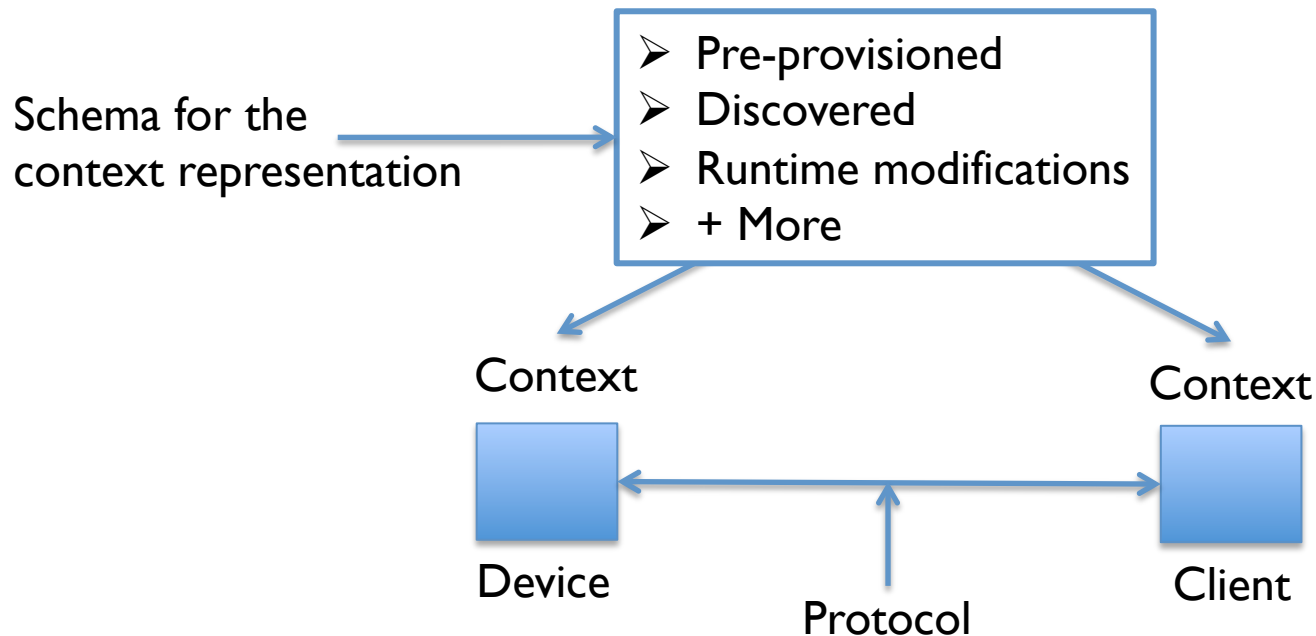
How will this work?



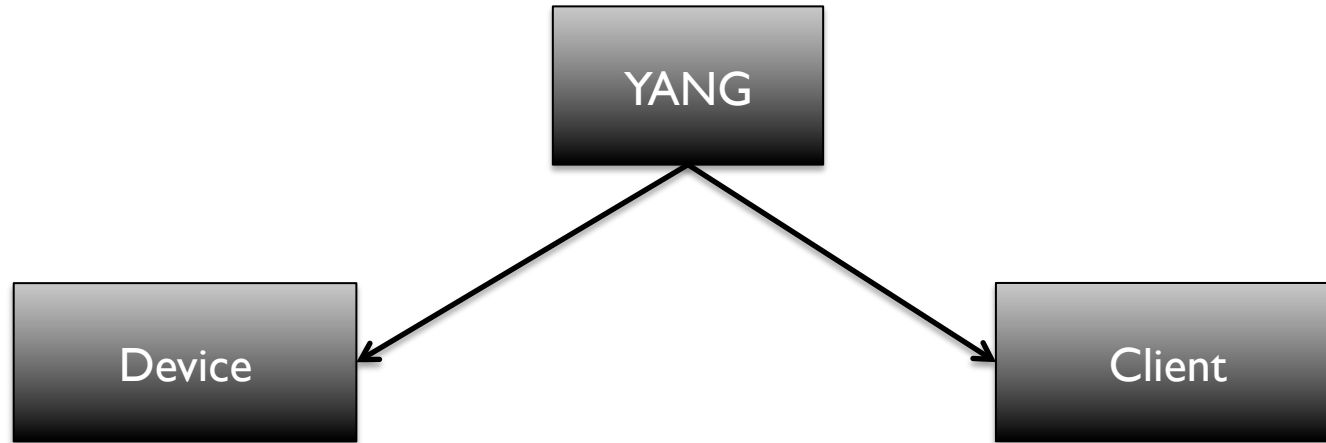
How will this work?



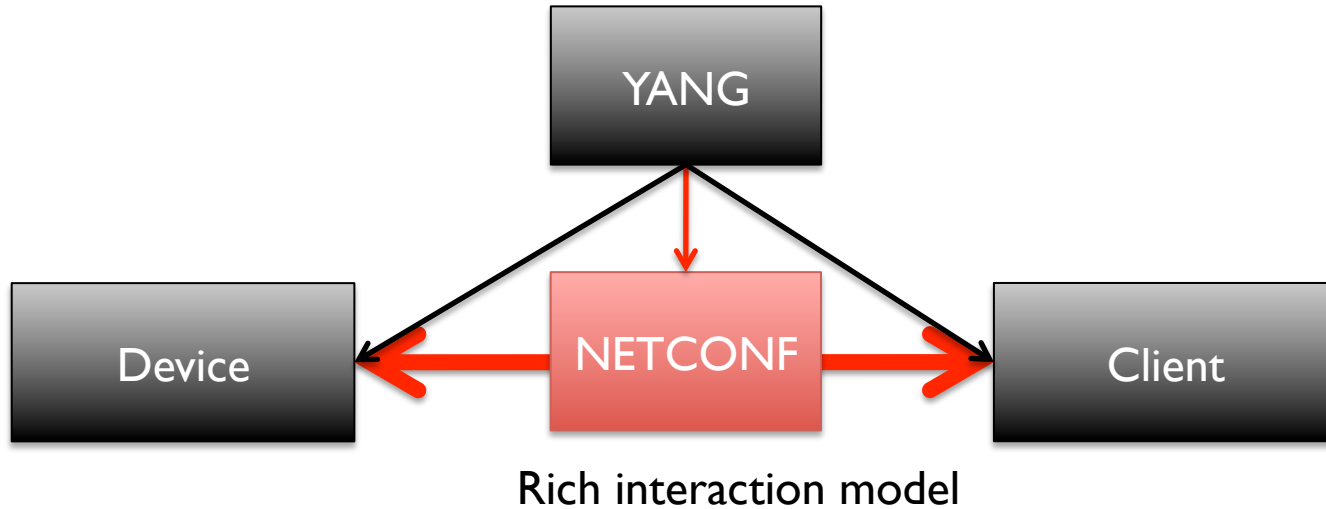
How will this work?



Data model as contract

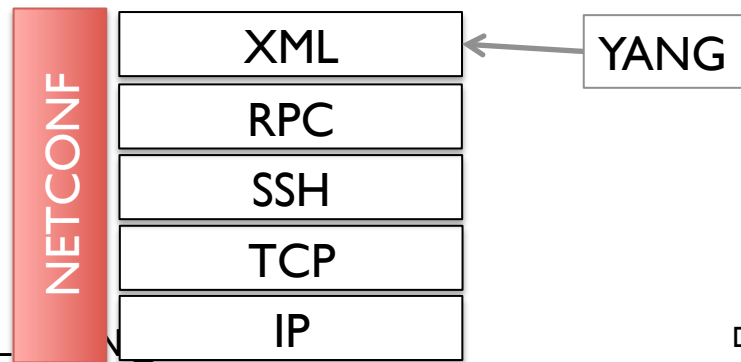
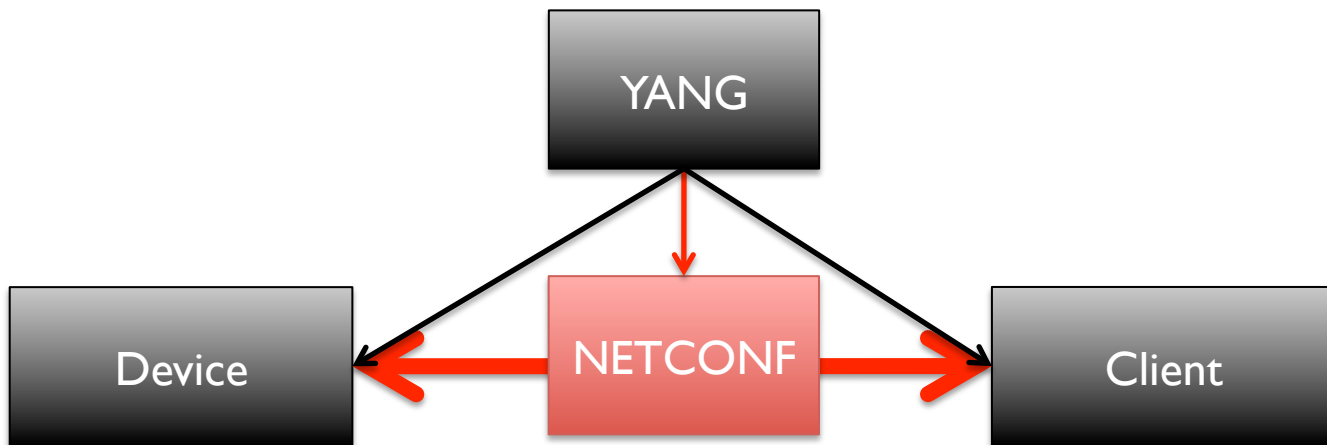


And the interaction model!



And the protocol bindings!

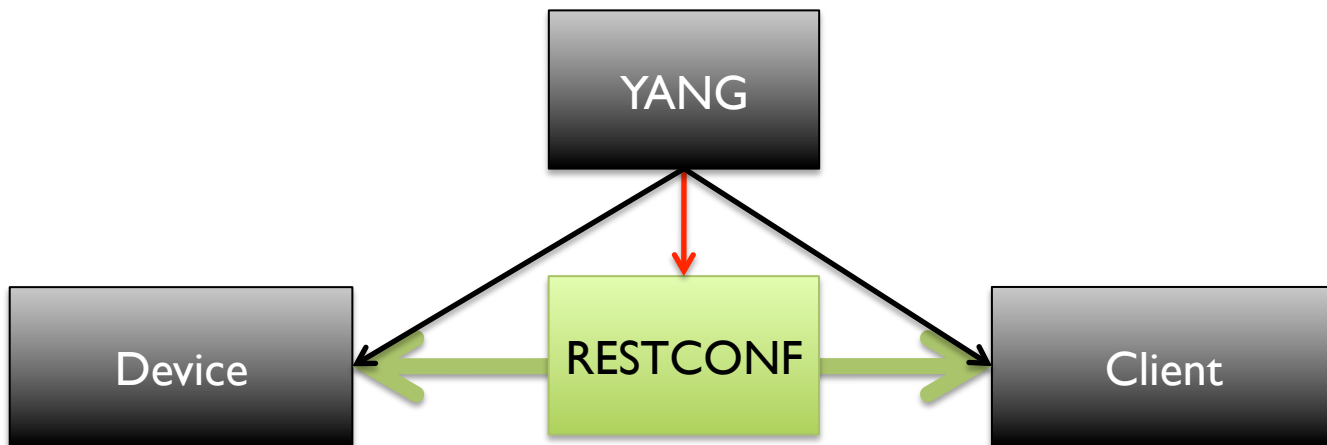
((LPWAN))



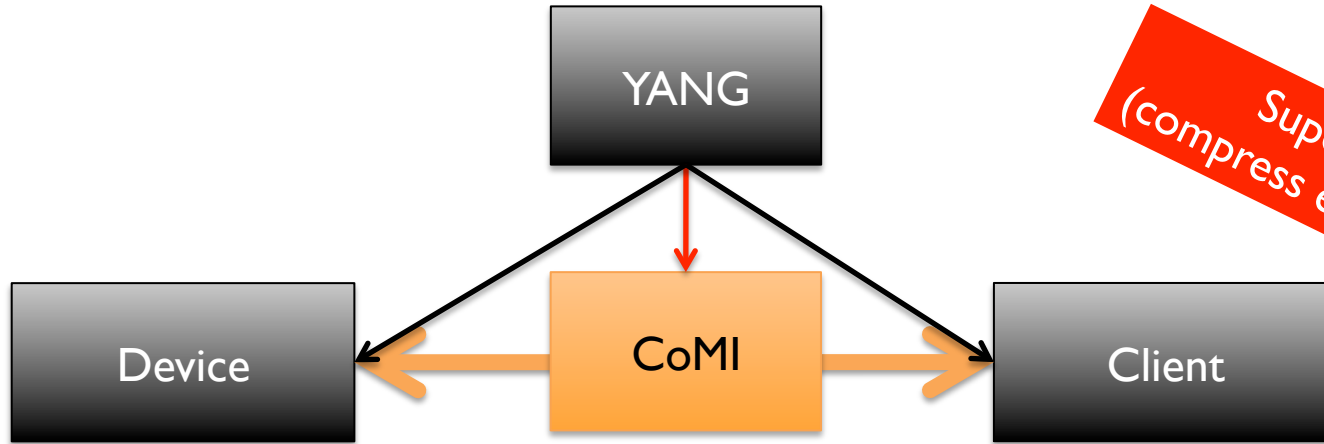
Data Models for SCHC

And the protocol bindings!

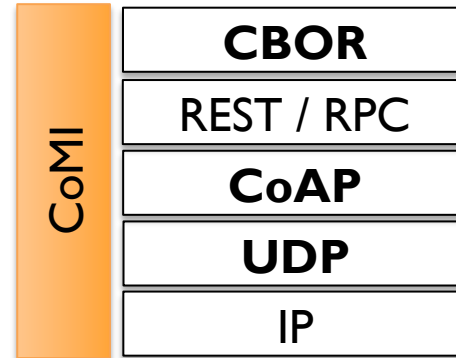
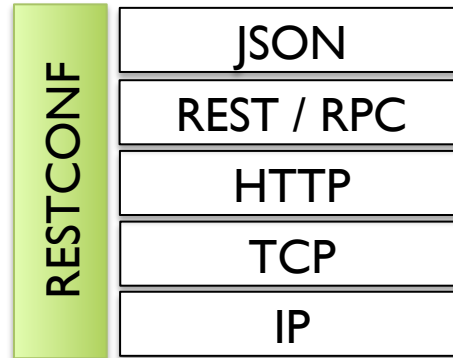
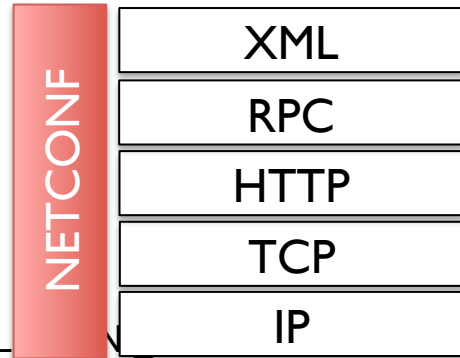
((LPWAN))



And the protocol bindings!



Super-efficient
(compress even identifiers)



Why YANG+CoMI?

- Compact and unique representation for:
 - Well-known Field ID, MO, CDA
- Compact exchanges between SCHC C/D
 - Prefix assignment
 - Setup port numbers, destination addresses,...
 - Compresses well with SCHC!
- Protocol bindings defined

Not starting from scratch...

draft-toutain-lpwan-yang-static-context-hc-00 (expired Jan 2017)

```
module: ietf-lpwan-schc
  +--rw context* [rule-id]
    +--rw rule-id      uint8
    +--rw rule-fields* [field-name field-position direction]
      +--rw field-name      field-name-type
      +--rw field-size?     uint8
      +--rw field-position  uint8
      +--rw direction       direction-type
      +--rw target-value?   lpwan-types
      +--rw matching-operator? matching-operator-type
      +--rw matching-operator-parameter? lpwan-types
      +--rw compression-decompression-action? compression-decompression-action-type
      +--rw compression-decompression-action-parameter? lpwan-types
```

Alternatives?

- Raw JSON / CBOR / Binary encoding
 - No Schema / CDDL
 - Define our own protocol for the exchanges
- Others ?

Thanks !