# Introduction to
SONiC-OTN RESTCONF and TELEMETRY

Alibaba Cloud
Xin Lei (leixin.lei@alibaba-inc.com)

SONiC-OTN Management Framework Overview

Management Framework consists of the following components:

- REST Server – a HTTP server implemented in Go language supports RESTCONF APIs for YANG data.

- Telemetry Dialout Client – a service implemented in Go language supports gNMI dialout publish

- Translib – a library that adapts requests from REST server to SONiC data providers and vice versa.

- Multi-DB – introduced it the last meeting
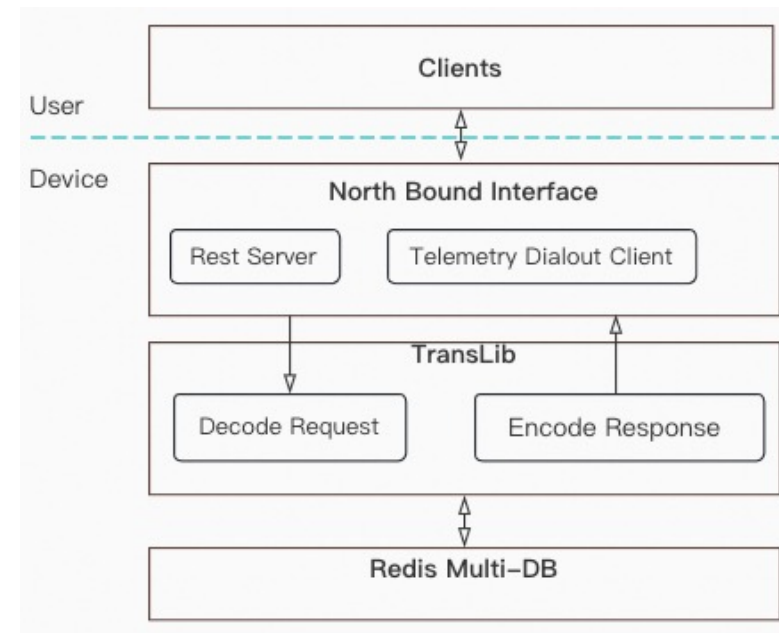
SONiC–OTN Management Framework Abstract Diagram

Management Framework Arch

Translib consists of three parts:

- **Message Conversion**
  - Request Handler:
    The APIs support to convert the incoming path and payload from the request into YGOT structures.
  - YGOT Request Binder:
    Generate a set of Go structures for the given YANG modules at build time and Unmarshall the given request into the Go structure objects.

- **Data Translation**
  - App Modules:
    Translate YGOT structures to DB schema and vice versa.
  - Transformer: A way to translate data automatically.

- **DB Operation:**
  - DB Access Layer: CRUD APIs in Go language
  - CVL: Validate configuration data before it is written to DB
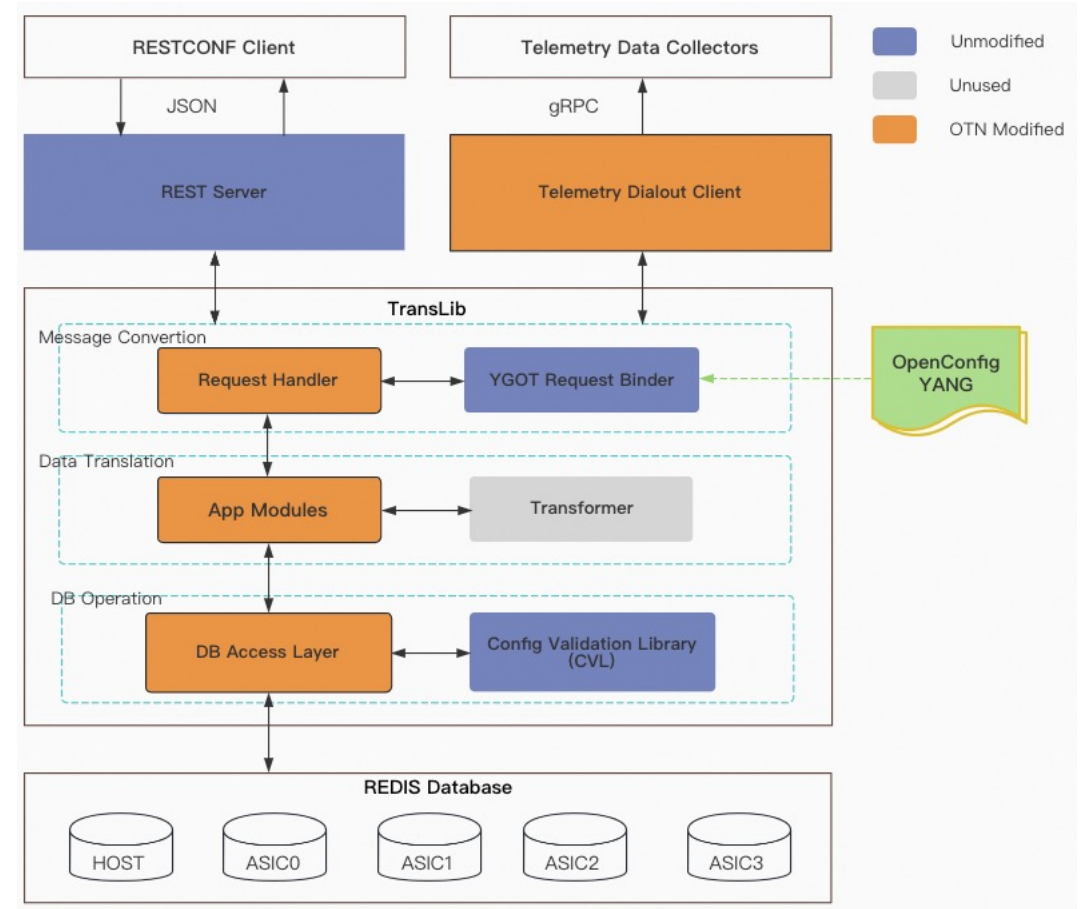
Management Framework architecture
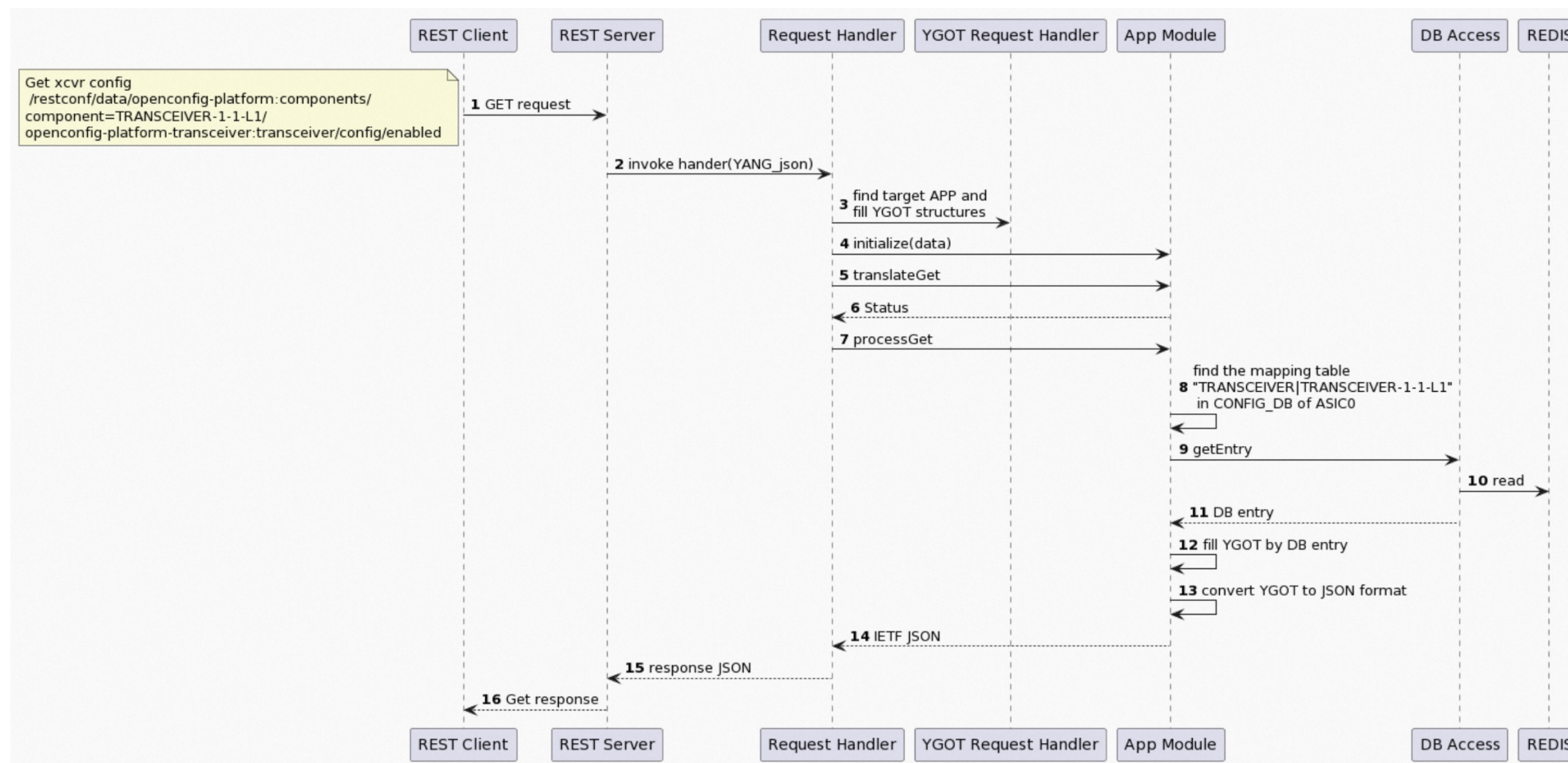
REST Set Flow

The flow of Management Framework for Turning on the laser :
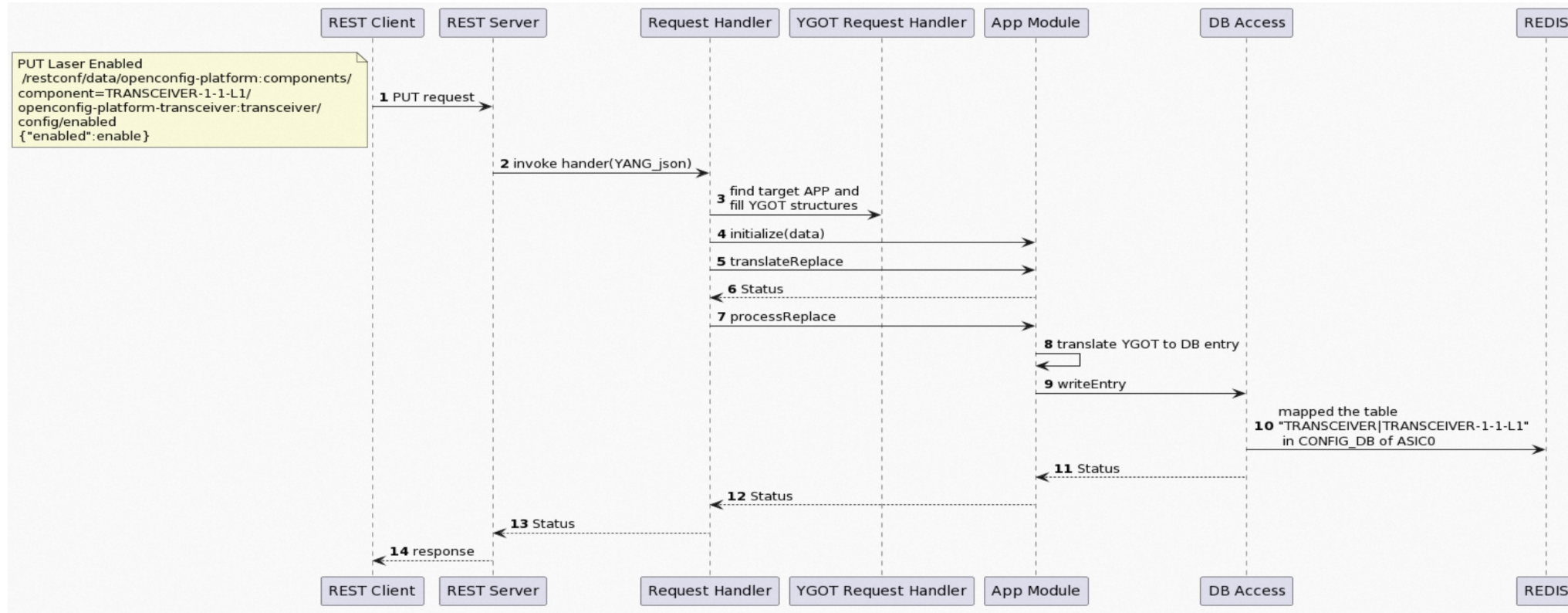
REST Get Flow

The flow of Management Framework for querying the laser value of the transceiver:

Multiple Databases Strategy

- Name different components by a fixed rule

- Derive two new type in DB Access Layer
  - type MDB map[string][*MaxDB*]*DB
  - type NumberDB map[string]*DB

- Extend App Interface with several interfaces

- Bind the new interfaces to TransLib request handlers

```
[ComponentType]-[ChasissID]-[SlotID]-[PortName /Number]
Without slot: CHASSIS-1、CU-1
Without port: FAN-1-7、PSU-1-5、LINECARD-1-2
Port name: OCH-1-3-L1、TRANSCEIVER-1-1-C1、PORT-1-4-OLPPRIIN
Number: AMPLIFIER-1-4-1、APS-1-4-1、OSC-1-4-1
```

```
translateMDBReplace(numDB db.NumberDB)
processMDBReplace(numDB db.NumberDB)
translateMDBGet(mdb db.MDB)
processMDBGet(mdb db.MDB)
// used by Telemetry
translateGetRegex(mdb db.MDB)
processGetRegex(mdb db.MDB)
```
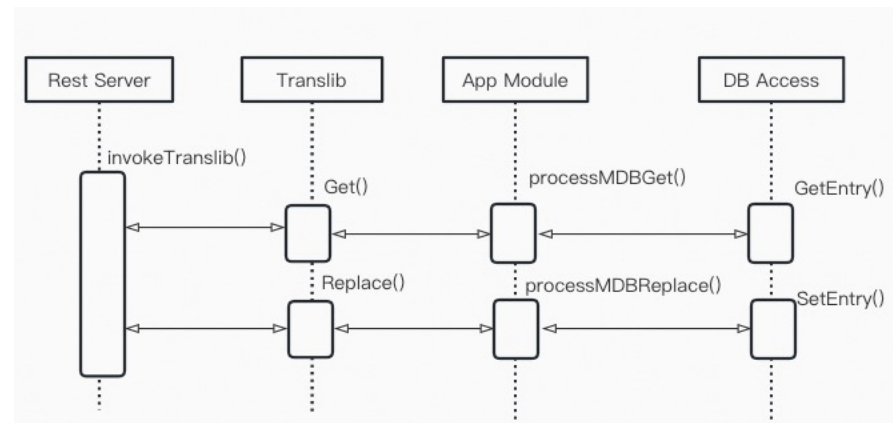
Database Modeling

opencific-optical-amplifier	module	
optical-amplifier	container	
amplifiers	container	
amplifier[name]	list	
name	leaf	
config	container	
name	leaf	
type	leaf	
target-gain	leaf	
min-gain	leaf	
max-gain	leaf	
target-gain-tilt	leaf	
gain-range	leaf	
amp-mode	leaf	
target-output-power	leaf	
max-output-power	leaf	
enabled	leaf	
fiber-type-profile	leaf	
state	container	
name	leaf	
type	leaf	
target-gain	leaf	
min-gain	leaf	
max-gain	leaf	
target-gain-tilt	leaf	
gain-range	leaf	
amp-mode	leaf	
target-output-power	leaf	
max-output-power	leaf	
enabled	leaf	
fiber-type-profile	leaf	
component	leaf	
ingress-port	leaf	
egress-port	leaf	
actual-gain	container	
instant	leaf	
avg	leaf	
min	leaf	
max	leaf	
interval	leaf	
min-time	leaf	
max-time	leaf	
actual-gain-tilt	container	
input-power-total	container	
input-power-c-band	container	
input-power-l-band	container	
output-power-total	container	
output-power-c-band	container	
output-power-l-band	container	
laser-bias-current	container	
optical-return-loss	container	

DATABASE	TABLE	FIELD
CONFIG_DB	AMPLIFIER|AMPLIFIER-1-4-1	name
		type
		target-gain
		min-gain
		max-gain
		target-gain-tilt
		gain-range
		amp-mode
		target-output-power
		max-output-power
		enabled
		fiber-type-profile
STATE_DB	AMPLIFIER|AMPLIFIER-1-4-1	name
		type
		target-gain
		min-gain
		max-gain
		target-gain-tilt
		gain-range
		amp-mode
		target-output-power
		max-output-power
		enabled
		fiber-type-profile
		component
		ingress-port
		egress-port
COUNTERS_DB	AMPLIFIER:AMPLIFIER-1-4-1_ActualGain:15_pm_current	instant
	AMPLIFIER:AMPLIFIER-1-4-1_ActualGain:24_pm_current	avg
	AMPLIFIER:AMPLIFIER-1-4-1_ActualGainTilt:15_pm_current	min
	AMPLIFIER:AMPLIFIER-1-4-1_ActualGainTilt:24_pm_current	max
		interval
		min-time
		max-time
		starttime
		validity

- Leaf node must have a field of the same name in database
- List node must have a table in database
- Container node for a gauge value must have a table in database, and other Container nodes may have a table or not

Object-oriented approach of App Modules

Simplify the implementation of App Modules in two stages:

1. Automatic data translation for CONTAINER nodes ---- (supported)
   - LEAF node is an object named *GoStructField*
   - CONTAINER node is an object named *GoStruct*

2. Automatic data translation for one YANG model ---- (future)
   - Traverse YANG tree by depth-first search
   - Get the mapped DB info of YANG nodes automatically

Supported App Modules

YANG Model Driven Design
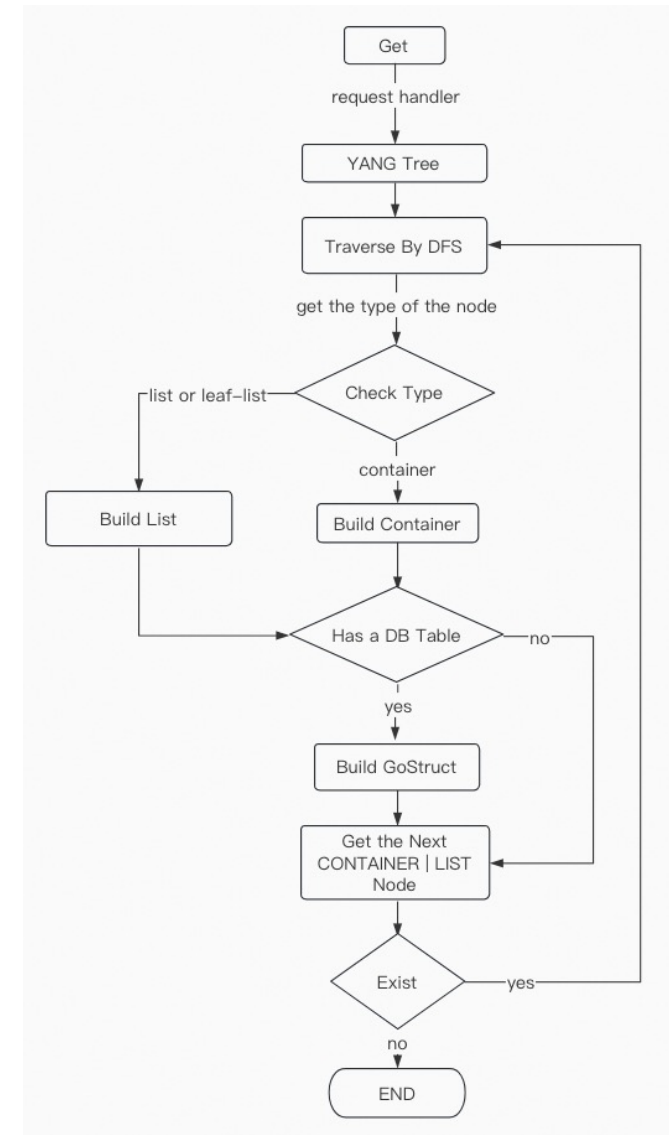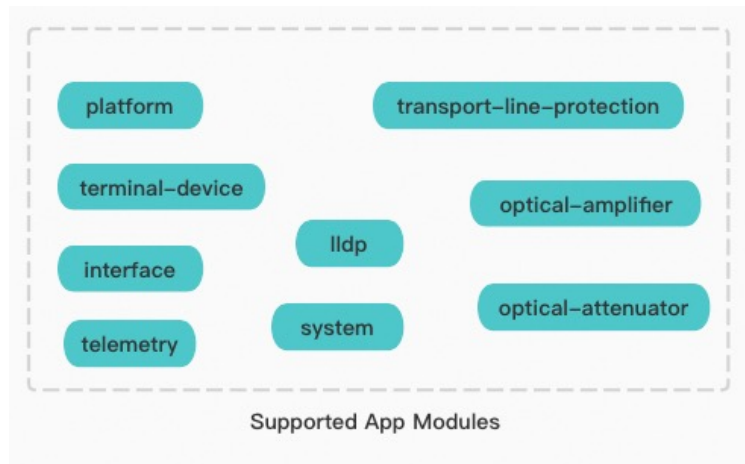
# YANG Model Driven In SONiC can be TRUE ?

Current Situation：
- Step 1: All leaf nodes were built automatically, but the mapping between YANG and DB schema was written manually
- Step 2: A set of buildXXX functions which represent the hierarchy of YANG models were written manually
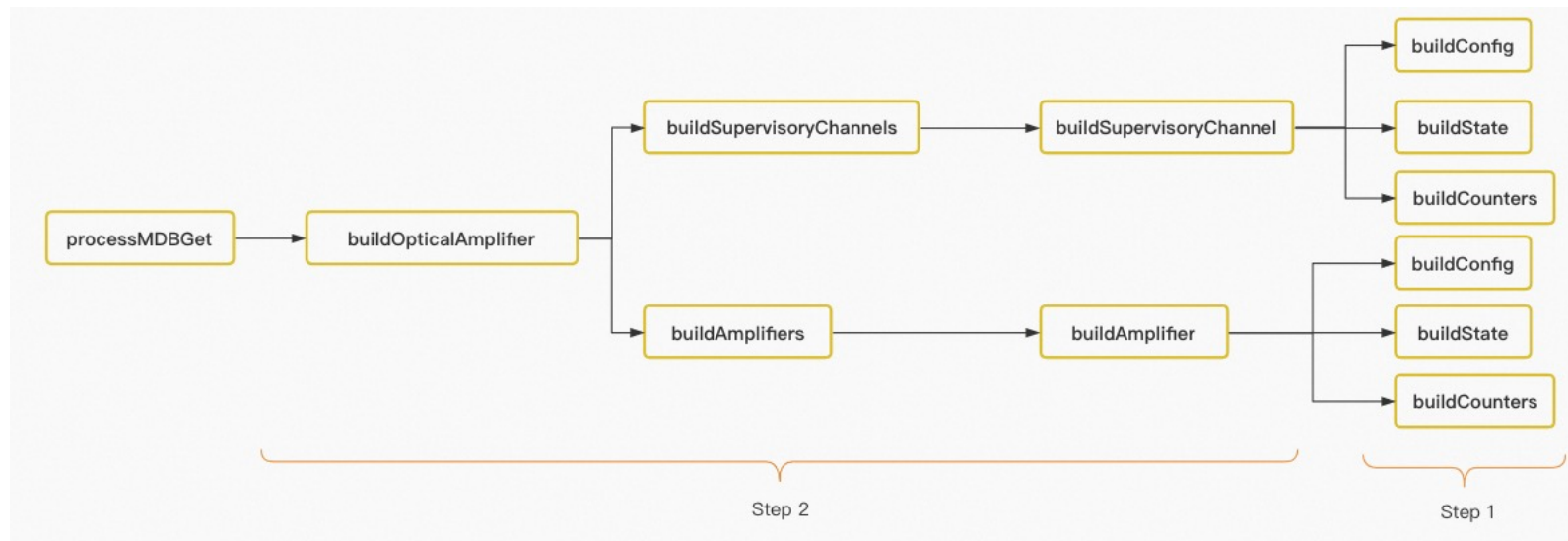
Dial-out Client With OpenConfig YANG

- Data Collectors: receive the gNMI data from Dial-out Client

- Dial-out Client: publish the subscribed data periodically or it was changed

- TransL Data Client: execute subscriptions and translate the JSON data from TransL Utils to gNMI format

- TransL Utils: a set of APIs to invoke Translib, then parse the playload returned by Translib to JSON format

- Subscriptions managed by OpenConfig Telemetry Module
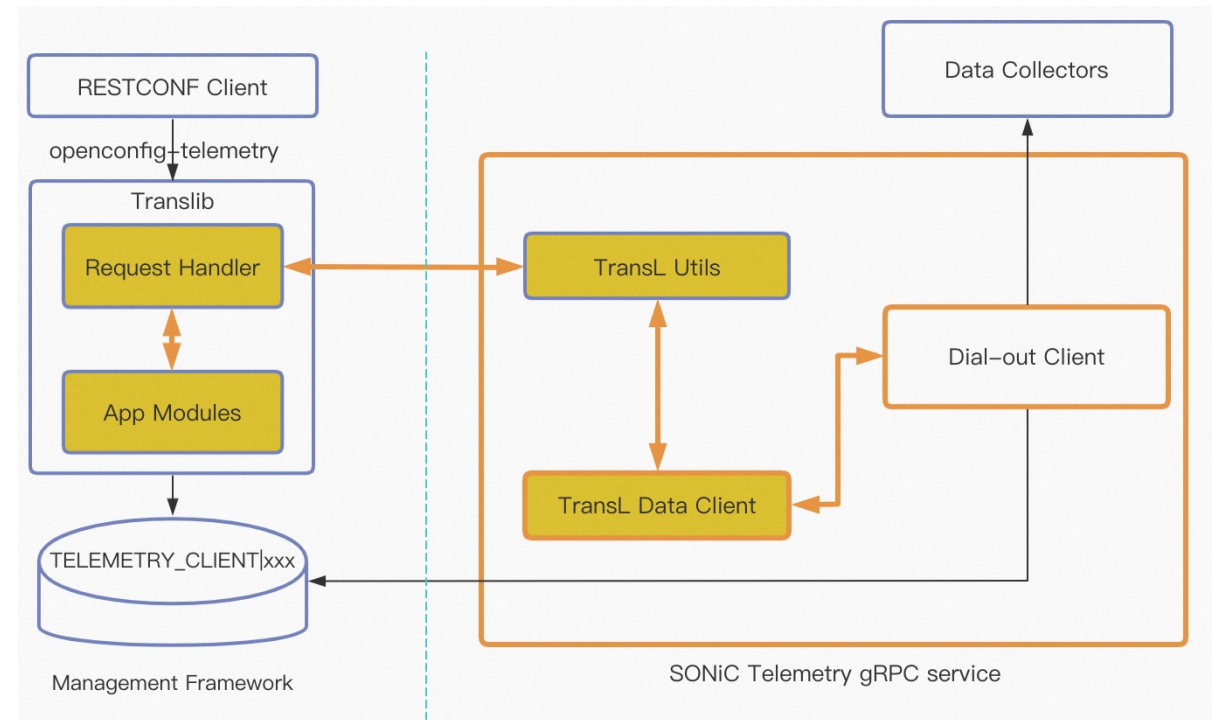
SAMPLE Subscribe-Publish Flow
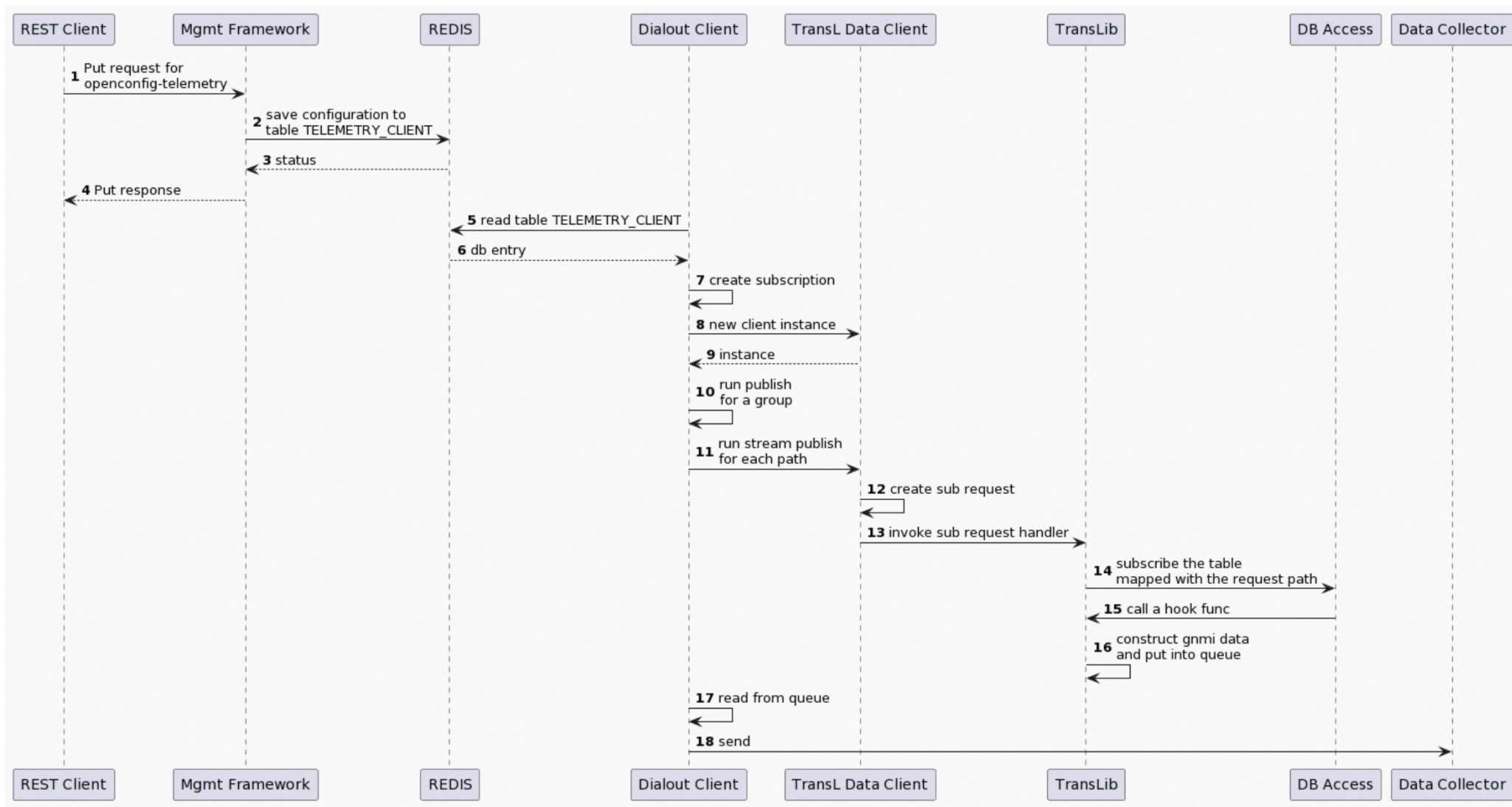
Custom Functionalities in Dial-out Client

- sensor-path without key
- data aggregation for sensor-path without key
- data filtering support to publish PM data only

```
"sensor-group-id": "och_sensor_group",
"config": {
    "sensor-group-id": "och_sensor_group"
},
"sensor-paths": {
    "sensor-path": [
        {
            "path": "/openconfig-platform:components/component/
                    openconfig-terminal-device:optical-channel/state",
            "config": {
                "path": "/openconfig-platform:components/component/
                        openconfig-terminal-device:optical-channel/state"
            }
        }
    ]
}
```

```
▼ fullContent: {}
  ▼ update: {}
      timestamp: "1681795812981484033"
    ▼ prefix: {}
        origin: "TBC04-K-BA-T008"
        target: "OC-YANG"
    ▼ update: []
      ▼ {}
        ▼ path: {}
          ▼ elem: []
            ▼ {}
                name: "openconfig-interfaces:interfaces"
            ▼ {}
                name: "interface"
              ▼ key: {}
                  name: "INTERFACE-1-1-C1"
            ▼ {}
                name: "state"
            ▼ {}
                name: "counters"
        ▼ val: {}
          ▼ jsonVal: {}
              in-broadcast-pkts: 0
              in-multicast-pkts: 0
              in-octets: 1954052689028752
              in-pkts: 2860386613942
              out-broadcast-pkts: 0
              out-multicast-pkts: 0
              out-octets: 394932905060786
              out-pkts: 578111704195
      ▼ {}
        ▼ path: {}
          ▼ elem: []
            ▼ {}
                name: "openconfig-interfaces:interfaces"
            ▼ {}
                name: "interface"
              ▼ key: {}
                  name: "INTERFACE-1-1-C4"
            ▼ {}
                name: "state"
            ▼ {}
                name: "counters"
        ▼ val: {}
          ▼ jsonVal: {}
              in-broadcast-pkts: 0
              in-multicast-pkts: 0
              in-octets: 491299806004930
              in-pkts: 719175790754
              out-broadcast-pkts: 0
              out-multicast-pkts: 0
              out-octets: 407868277406234
              out-pkts: 597046815923
```

```
▼ fullContent: {}
  ▼ update: {}
      timestamp: "1681795812629116160"
    ▼ prefix: {}
        origin: "TBC04-K-BA-T008"
        target: "OC-YANG"
    ▼ update: []
      ▼ {}
        ▼ path: {}
          ▼ elem: []
            ▼ {}
                name: "openconfig-platform:components"
            ▼ {}
                name: "component"
              ▼ key: {}
                  name: "OCH-1-1-L2"
            ▼ {}
                name: "openconfig-terminal-device:optical-channel"
            ▼ {}
                name: "state"
        ▼ val: {}
          ▼ jsonVal: {}
              actual-frequency-offset: 570
            ▼ chromatic-dispersion: {}
                avg: 4
                instant: 4
                interval: 900000000000
                max: 4
                max-time: 16817958008177790
                min: 4
                min-time: 16817958008177790
            ▼ input-power: {}
                avg: -0.81
                instant: -0.83
                interval: 900000000000
                max: -0.8
                max-time: 1681795804492181885
                min: -0.83
                min-time: 1681795809493981383
            ▼ laser-bias-current: {}
                avg: 244
                instant: 244
                interval: 900000000000
                max: 244
                max-time: 16817958008177790
                min: 244
                min-time: 16817958008177790
            ▼ osnr: {}
                avg: 34.46
                instant: 34.1
                interval: 900000000000
                max: 34.6
```

Q & A