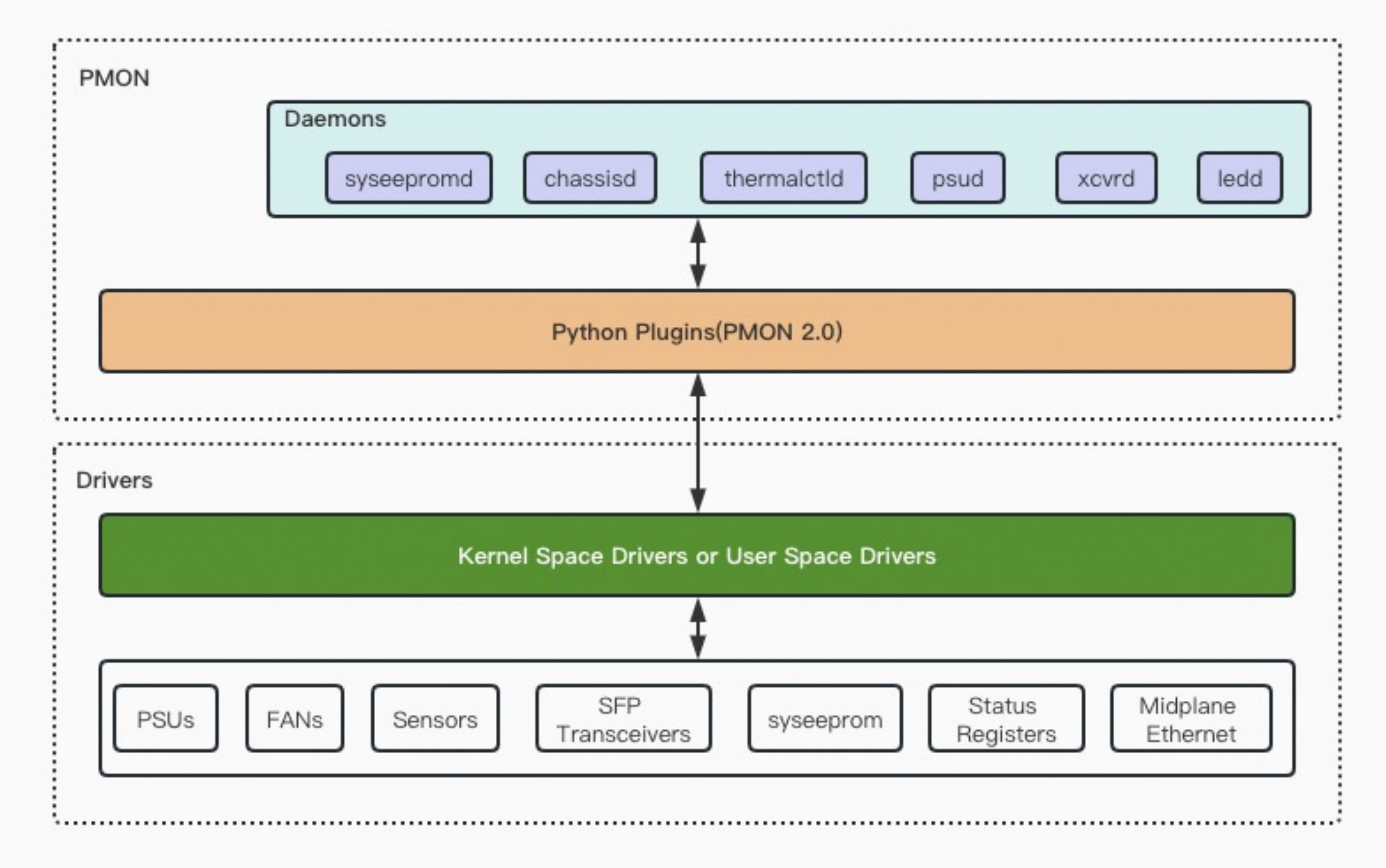
Introduction to SONiC-OTN Platform Monitor

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

SONiC`s Native PMON

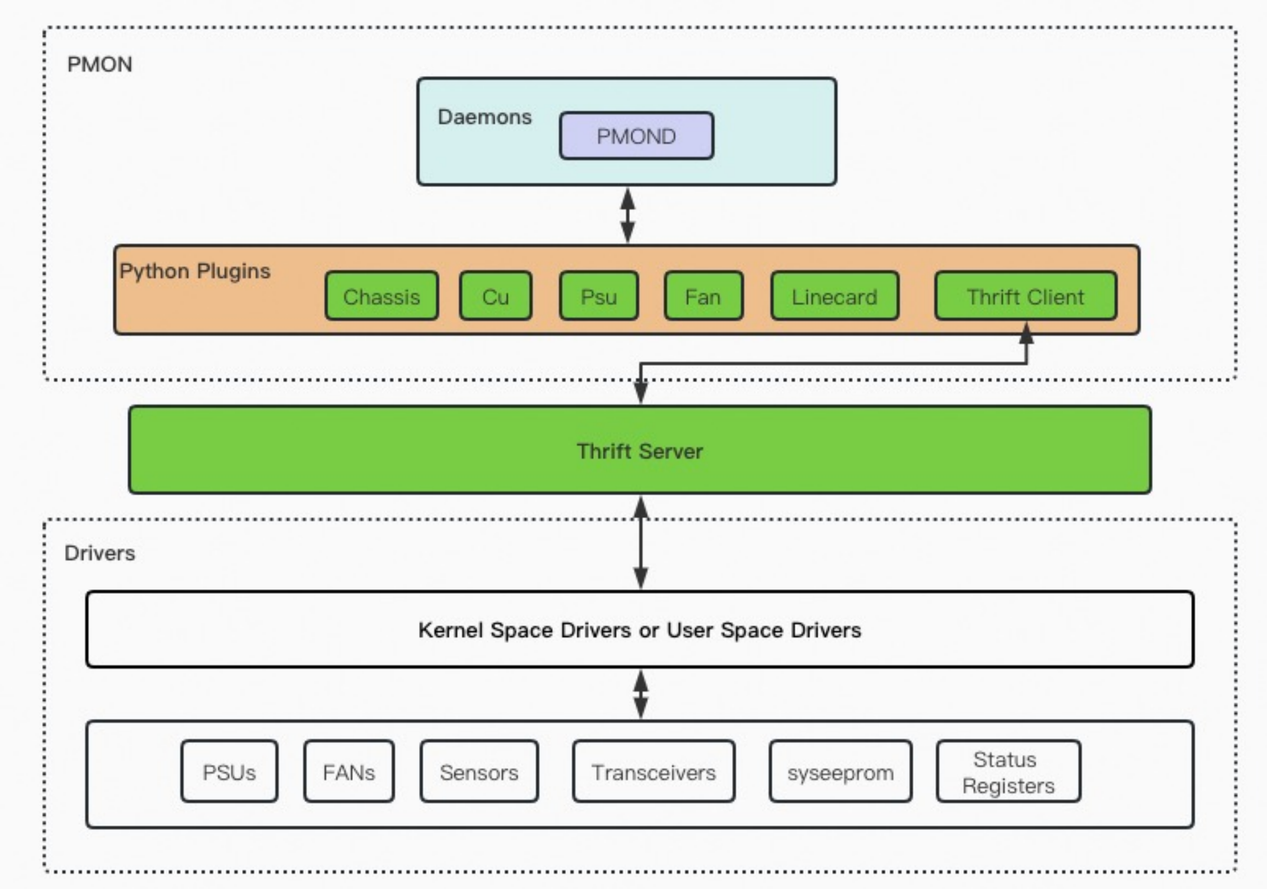
- Designed for the multi-chassis devices
- Every line-card must have a CPU
- Insufficient support for Openconfig YANG models
- Strong dependencies of driver implementation
- Database modeling with high compexity



SONiC PMON architecture

SONiC-OTN`s PMON

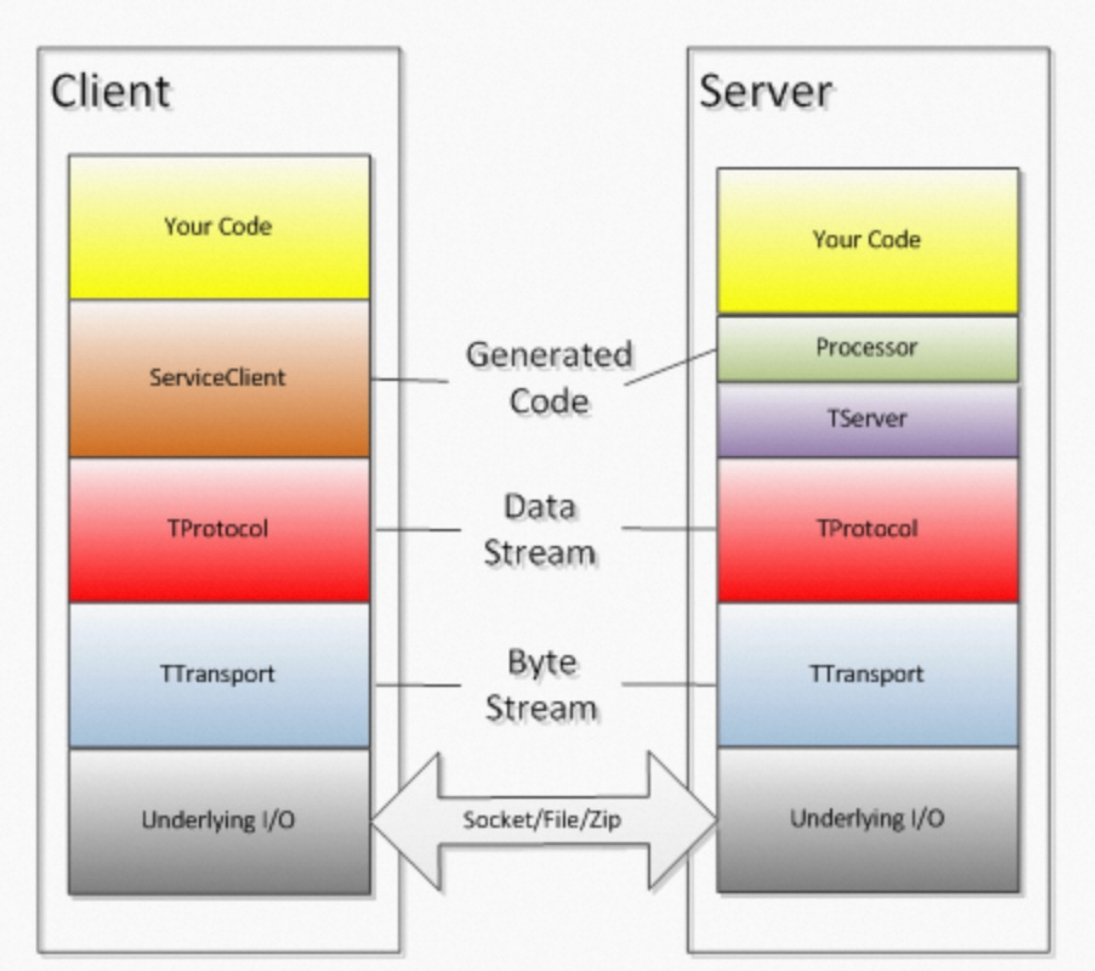
- Add an adaptation layer by thrift
- Free the dependency of driver implementation
- Sufficient support for Openconfig YANG models
- One daemon with 5 threads
- Transceivers are managed by LAI



SONiC-OTN PMON architecture

Thrift Introduction

- An interface description language(IDL)
- Capabilities like data transport, data serialization, and application level processing.
- A code generation engine to build RPC clients and servers.
- Work efficiently and seamlessly between different languages

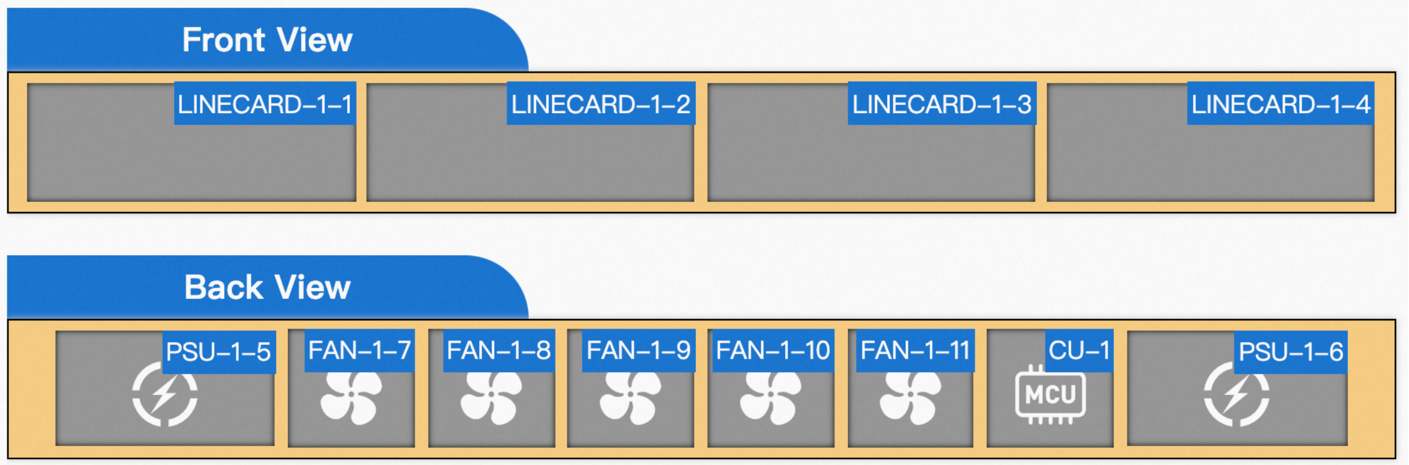


Thrift protocol stack

Equipment Modeling

- objects

- 1 chassis
- 1 or 2 control units
- 1 or more line-cards
- 1 or more fans
- 1 or more PSUs



A whitebox OTN equipment image

- functionalities

- All peripherals update state data and counter data periodically
- Query for inventory information, versions, temperature, power and so on
- LEDs management
- Fans management
- Power management
- Cold or warm reboot

Thrift API

- Each hardware peripheral has a unique type

```
enum periph_type {
    CHASSIS,
    LINECARD,
    CU,
    FAN,
    PSU,
    UNKNOWN
}
```

- Peripheral related thrift RPCs

```
bool periph_presence(1: periph_type type, 2: i8 id);
string get_periph_version(1: periph_type type, 2: i8 id);
i32 get_periph_temperature(1: periph_type type, 2: i8 id);
periph_eeprom get_periph_eeprom(1: periph_type type, 2: i8
id);
```

```
>>> c.pltfm_mgr.periph_presence(periph_type.FAN, 6)
False
>>> c.pltfm_mgr.periph_presence(periph_type.CU, 1)
True
>>> c.pltfm_mgr.periph_presence(periph_type.LINECARD, 1)
True
>>> c.pltfm_mgr.periph_presence(periph_type.FAN, 7)
True
>>> c.pltfm_mgr.get_periph_eeprom(periph_type.CHASSIS, 1)
periph_eeprom(type='CHASSIS', model_name='OBX-1141T',
pn='1141TAC000', sn='AC0405173001', label='', hw_ver='1.1',
sw_ver='0.0', mfg_date='2022-03-03',
mac_addr='ac:ce:57:d8:d0:fc')
>>> c.pltfm_mgr.get_periph_eeprom(periph_type.FAN, 7)
periph_eeprom(type='FAN', model_name='', pn='FANAC000',
sn='AC042315901A', label='', hw_ver='1.1', sw_ver='0.0',
mfg_date='2022-07-19', mac_addr='00:00:00:00:00:00')
>>> c.pltfm_mgr.get_periph_temperature(periph_type.FAN, 7)
5110
>>>
```

Some examples

Thrift API

- Fan speed structure

```
struct
fan_speed {
1: i32 front;
2: i32 behind;
}
```

```
enum led_color {
RED,
GREEN,
YELLOW,
ORANGE,
NONE
}
```

```
enum led_flash_type {
RED_YELLOW_GREEN,
NONE
}
```

- Fan related thrift RPCs

```
fan_speed get_fan_speed(1: i8 id);

ret_code set_fan_speed(1: i8 id, 2: i32 speed);

ret_code set_fan_speed_rate(1: i8 id, 2: i32 speed_rate);  # to be extended
```

Thrift API

- PSU information structure

```
struct psu_info {
1: i32 abs;
2: i32 ambient_temp;
3: i32 primary_temp;
4: i32 secondary_temp;
5: i32 vout;
6: i32 vin;
7: i32 iout;
8: i32 iin;
9: i32 pout;
10: i32 pin;
11: i32 fan;
12: i32 capacity;
}
```

- PSU related thrift RPCs

```
psu_info get_psu_info(1: i8 id);
```

Thrift API

- Reboot related structure

```
enum reboot_type {
POWER,
COLD,
SOFT,
ABNORMAL,
DOG,
BUTTON,
}
```

```
enum power_ctl_type {
OFF,
ON
}
```

- Reboot related thrift RPCs

```
reboot_type get_reboot_type();

ret_code periph_reboot(1: periph_type type, 2: i8 id, 3: reboot_type reboot_type);

ret_code set_power_control(1: i8 slot_id, 2: power_ctl_type type);
```

Thrift API

- LED is for CU, FAN, PSU

```
enum led_type {
CU,
FAN,
PSU,
UNKNOWN
}
```

```
enum led_color {
RED,
GREEN,
YELLOW,
ORANGE,
NONE
}
```

```
enum led_flash_type {
RED_YELLOW_GREEN,
NONE
}
```

- LED related thrift RPCs

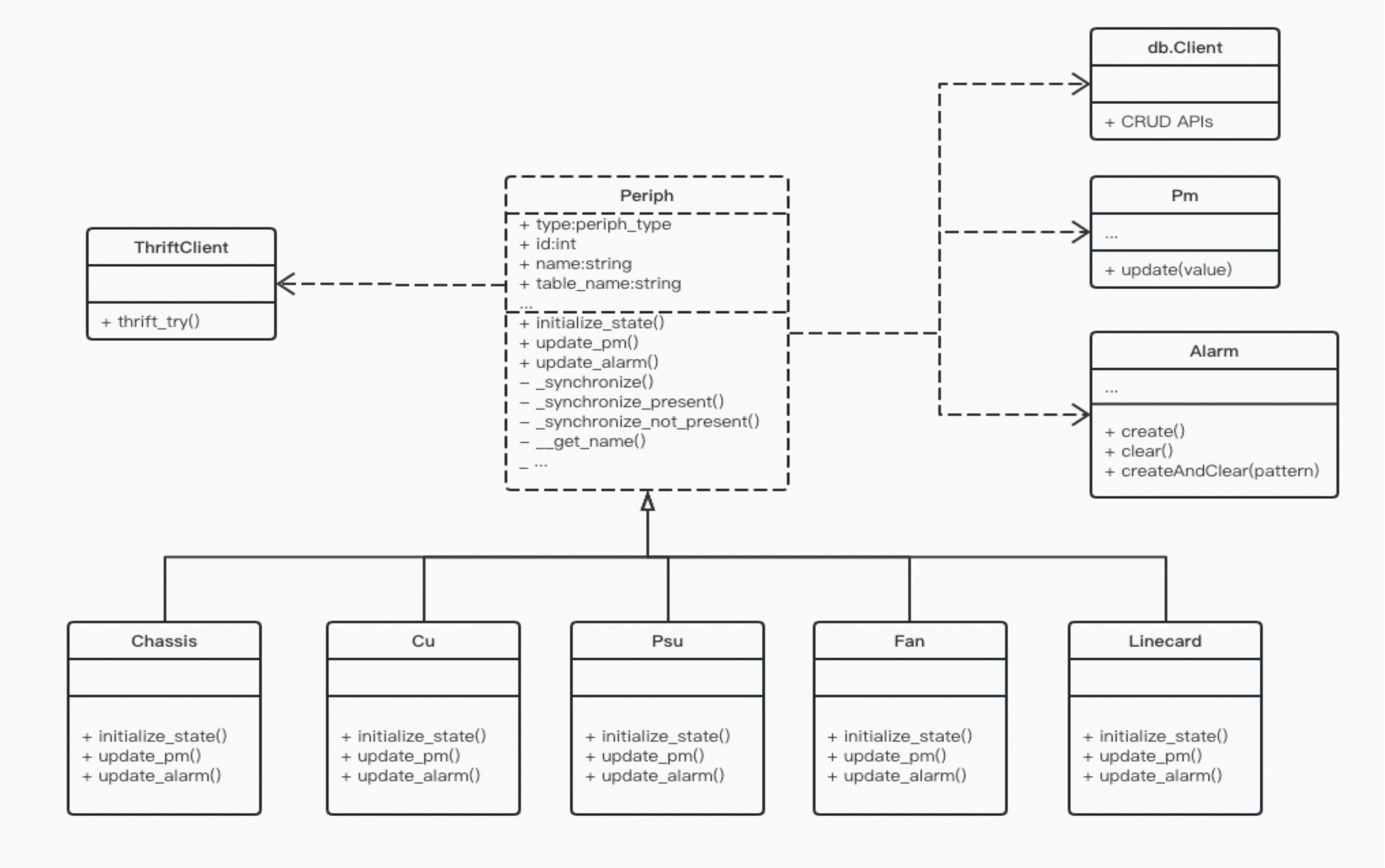
```
ret_code set_led_flash(1: led_type type, 2: i8 id, 4: led_flash_type flash_type);

ret_code set_led_color(1: led_type type, 2: i8 id, 3: led_color color);
```

- Complete thrift definitions

  device.thrift

PMON Classes



sonic-otn-platform-common source code

To Be Done

- Future Items in the next month

- A thrift mock server named devmock
- Reboot functionality
- Fan speed control functionality
- counters and alarms addition

Q & A