Introduction to southbound components in AONOS

Alibaba Cloud and Accelink

Xiaodong Gui (guixiaodong.gxd@alibaba-inc.com)
Weitang Zheng (zhengweitang.zwt@alibaba-inc.com)
Ziye Chen (ziye.chen@accelink.com)
Xiaosheng You (xiaosheng.you@accelink.com)

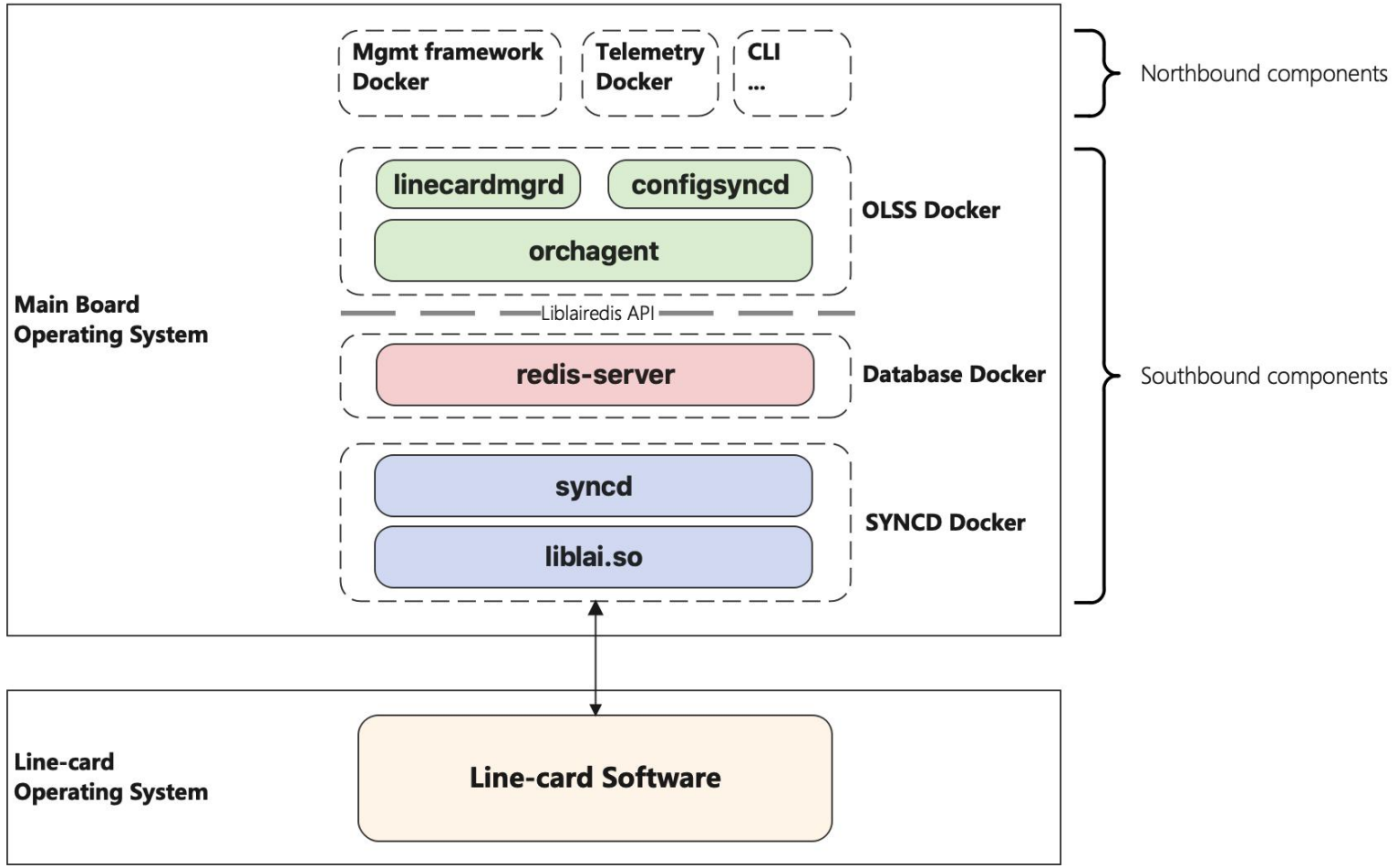
AONOS: Another Optical Network Operating System

Contents:

- AONOS Architecture
- Line-card Abstraction Interface (LAI)
- OLSS, SyncD and Line-card DB in AONOS
- How to enable tx-laser of a transceiver
- How to warm-reboot a line-card
- Performance Monitoring -- by Accelink
- Alarm notification -- by Accelink

AONOS Architecture

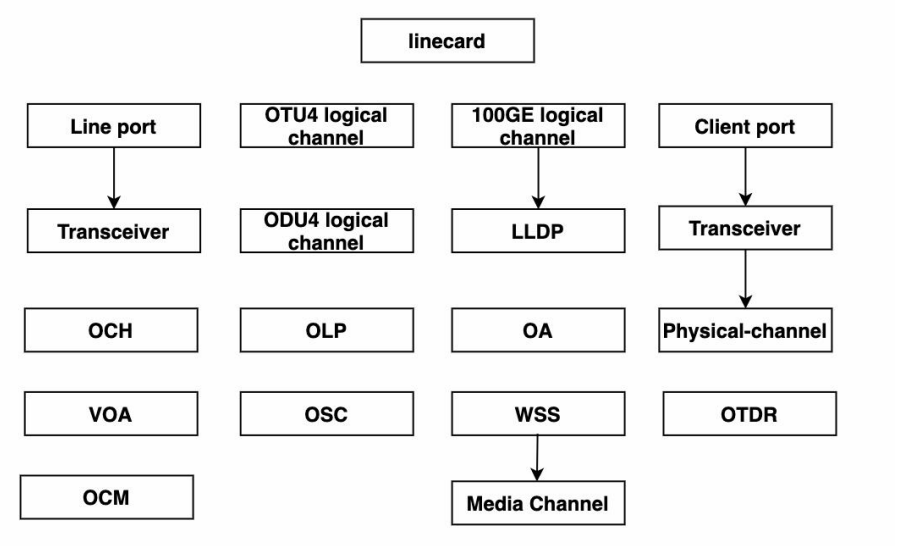
- Northbound Components
  - Restconf (OpenConfig)
  - Telemetry (OpenConfig)
  - Klish-style CLI
- Southbound Components
  - Optical Line-card State Service (OLSS)
  - SyncD (LAI)
- Line-card Database
  - a redis server which is inclusive to a line-card



AONOS architecture

Line-card Abstraction Interface (LAI)

- CRUD APIs (Create/Read/Update/Delete)
- object oriented, a set of attributes and statistics is pre-defined
- references OpenConfig model
- optical terminal system and optical line system are supported



LAI object model and hierarchy

```
     ɛr Create linecard

  @param[out] linecard_id The Linecard Object ID
* @param[in] attr_count Number of attributes
* @param[in] attr_list Array of attributes to set during
*
* @return #LAI_STATUS_SUCCESS on success, failure status
*/
ypedef lai_status_t (*lai_create_linecard_fn)(
      _Out_ lai_object_id_t *linecard_id,
      _In_ uint32_t attr_count,
      _In_ const lai_attribute_t *attr_list);
```

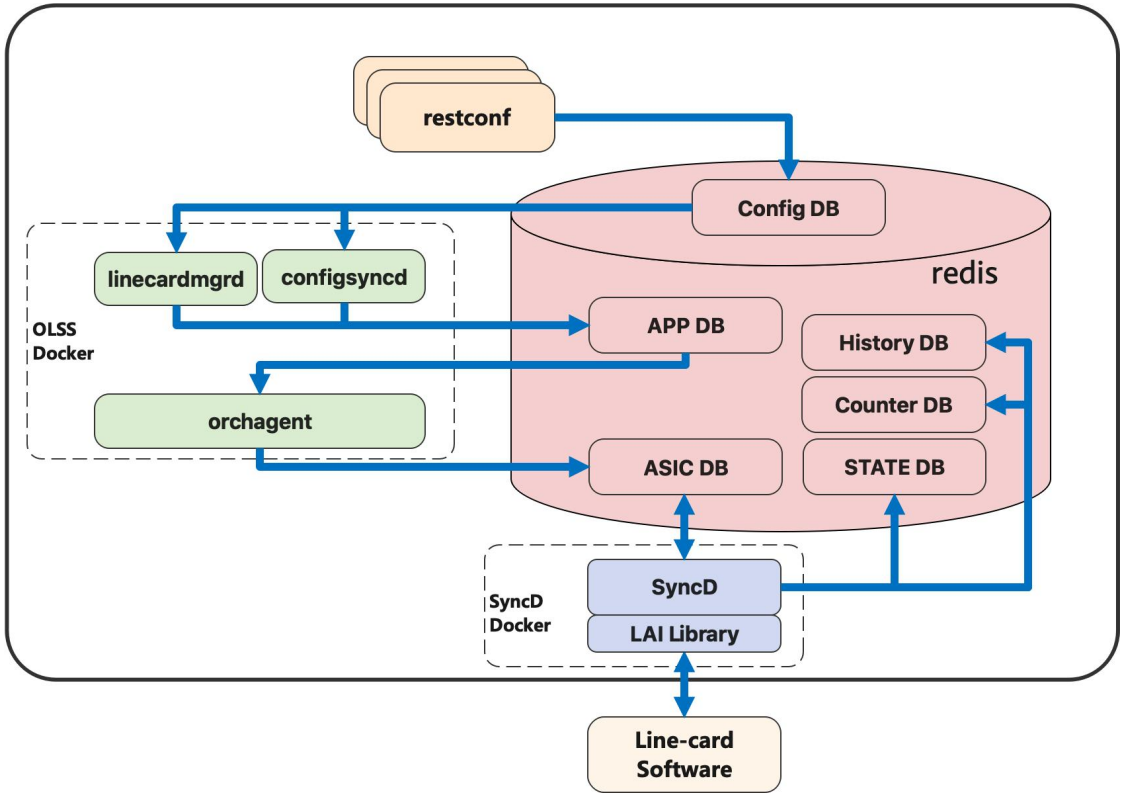
Line-card creation function declaration

OLSS, SyncD and Line-card DB in AONOS

- OLSS Docker
  - **linecardmgrd:** sync line-card configuration
  - **configsyncd:** sync other configuration
  - **orchagent:** translation between APP and LAI objects.

- Syncd Docker
  - **SyncD:** sync LAI objects between main board and line-card.

- Line-card Database Docker
  - **Config DB**: persist configuration
  - **APP DB:** persist App objects
  - **ASIC DB (LAI DB):** persist LAI objects
  - **State DB:** persist state info, current alarm
  - **Counter DB:** persist PM result
  - **History DB:** persist historical PM result and alarm

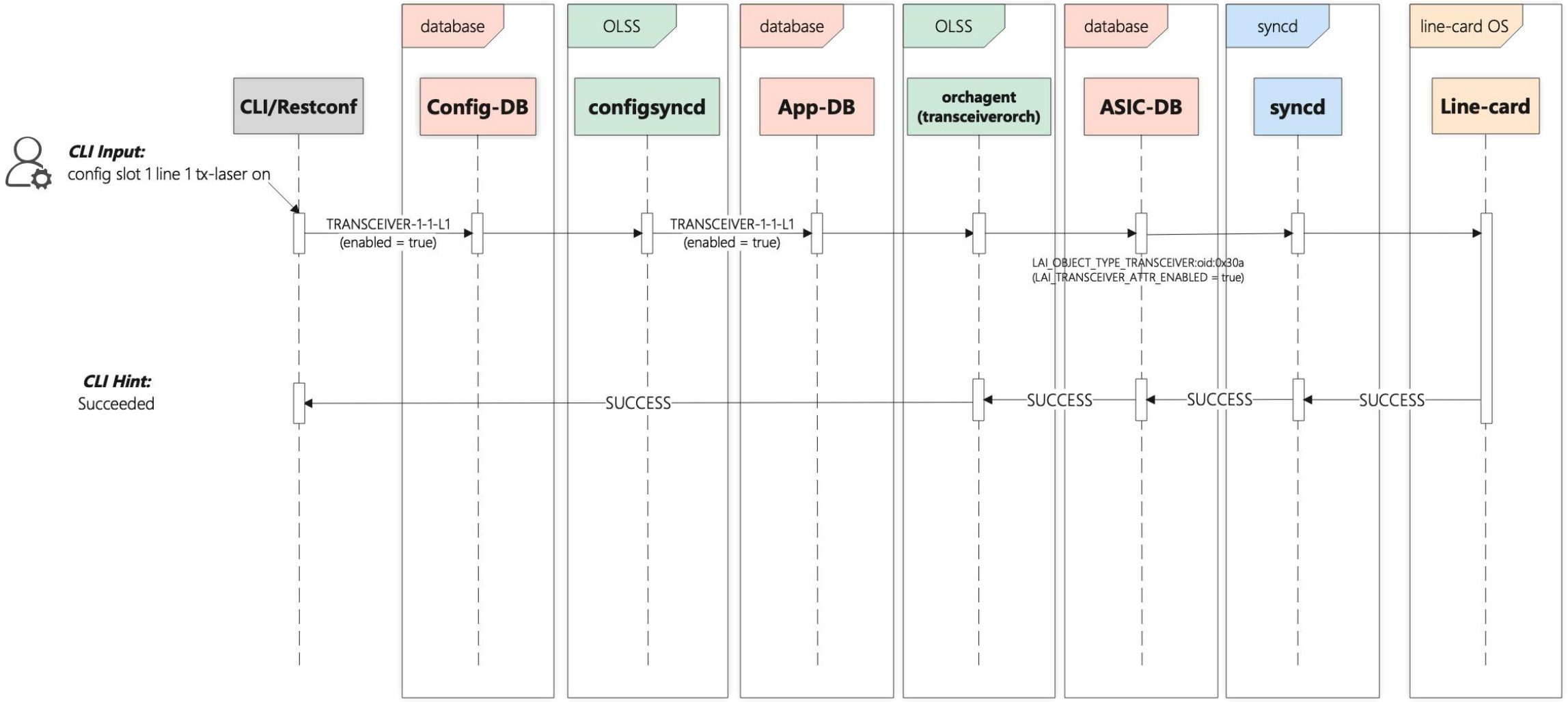


A deep insight into southbound components

Acronyms:

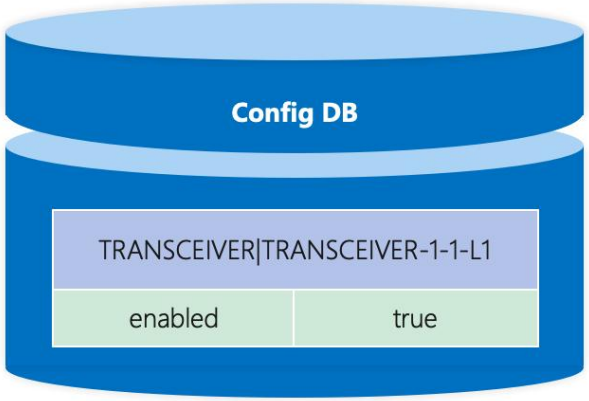
OLSS: Optical Line-card State Service
linecardmgrd: Line-card Management Daemon
configsyncd: Configuration Sync Daemon
orchagent: Orchestration Agent
syncd: Sync Daemon
PM: Performance Monitoring

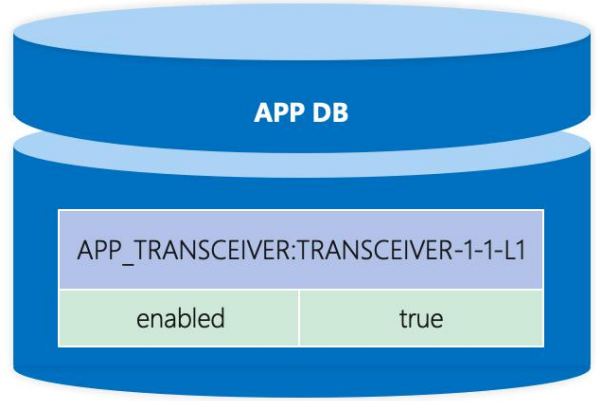
How to enable tx-laser of a transceiver

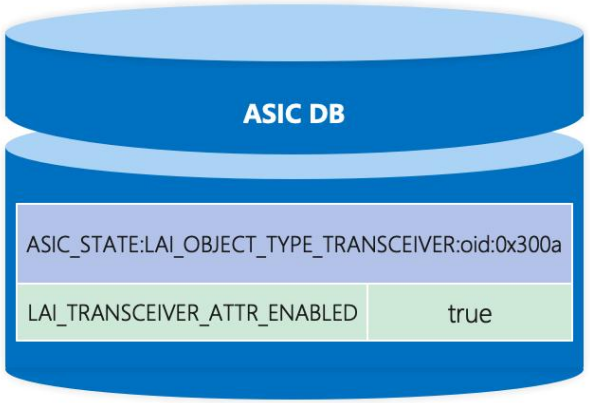


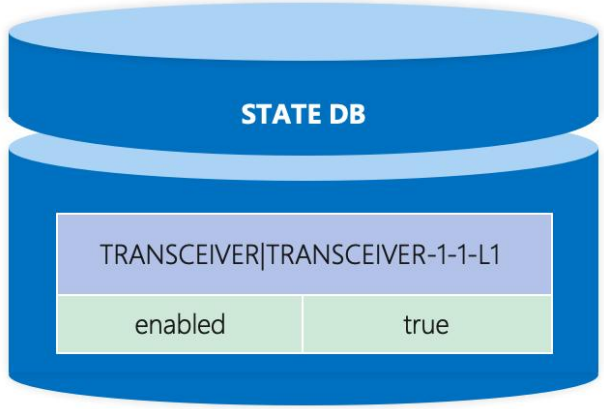
How to store the config and state of tx-laser

- Config DB
  - to store the user configuration of tx-laser
  - no loss after being saved to configuration files
- APP DB
  - be same with Config Table in Config DB
- ASIC DB
  - LAI oriented
  - be used to reconfig line-card
- STATE DB
  - to store the real state of tx-laser



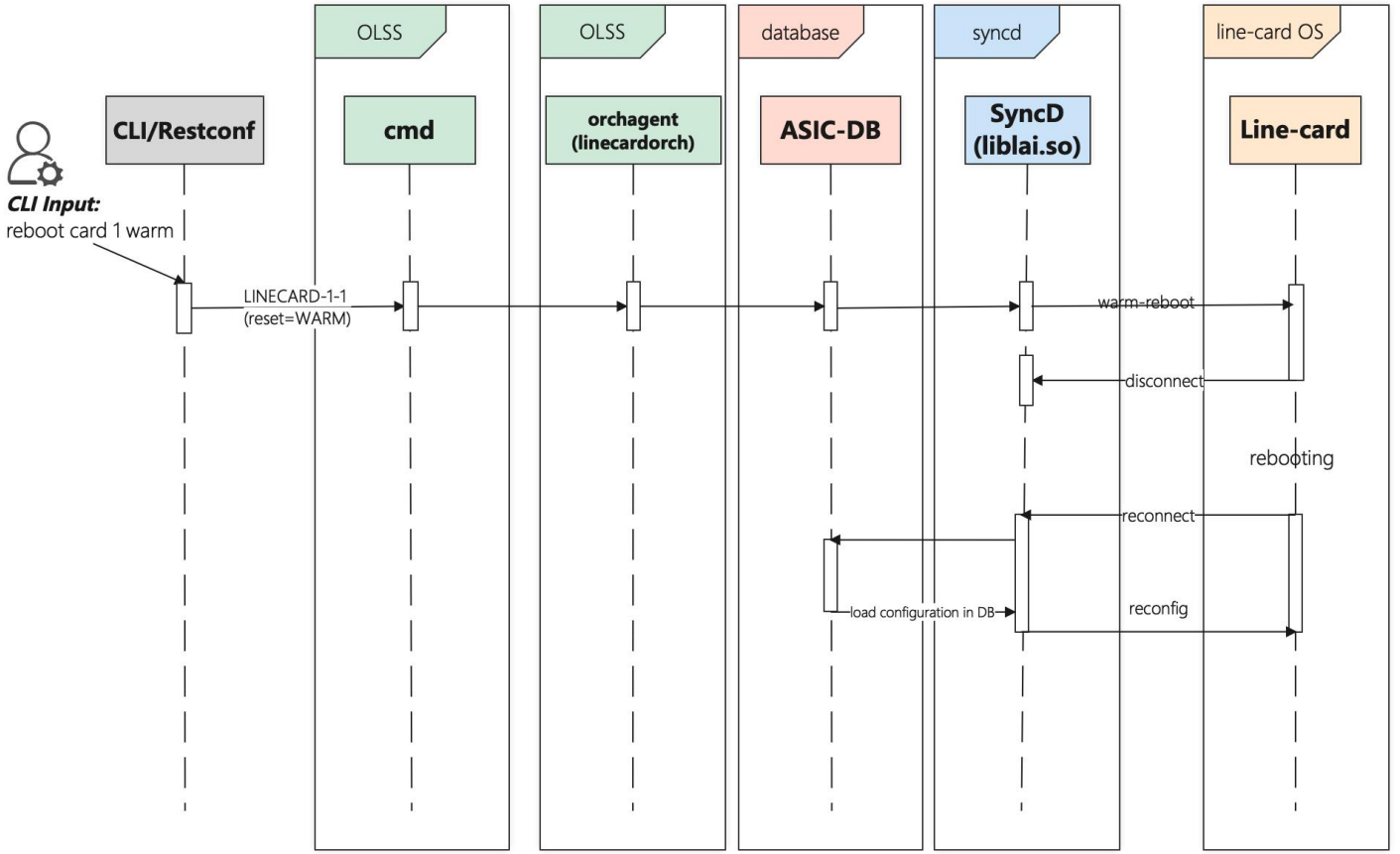






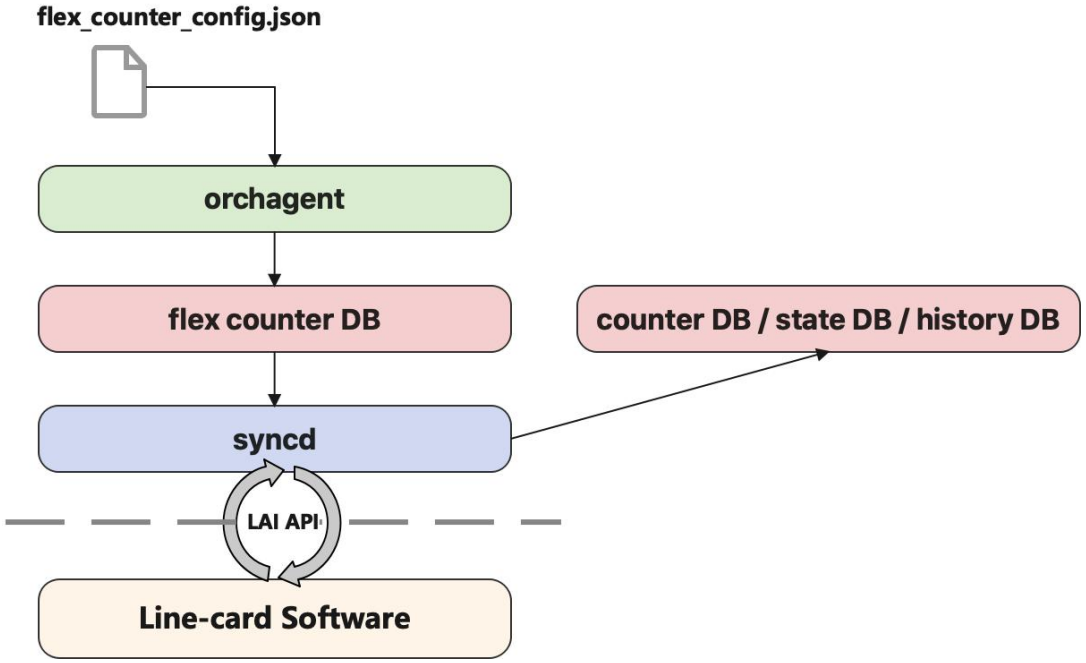
How to warm-reboot a line-card

- User instruction is processed by CLI/restconf.
- The warm-reboot instruction is conveyed to orchagent through redis channel instead of config DB.
- Orchagent invokes lairedis set API to notify SyncD.
- SyncD invokes LAI set API when receiving the instruction from orchangent.
- Line-card OS reboots.
- When line-card recovers, SyncD loads all LAI objects in ASIC DB and reconfigs line-card.

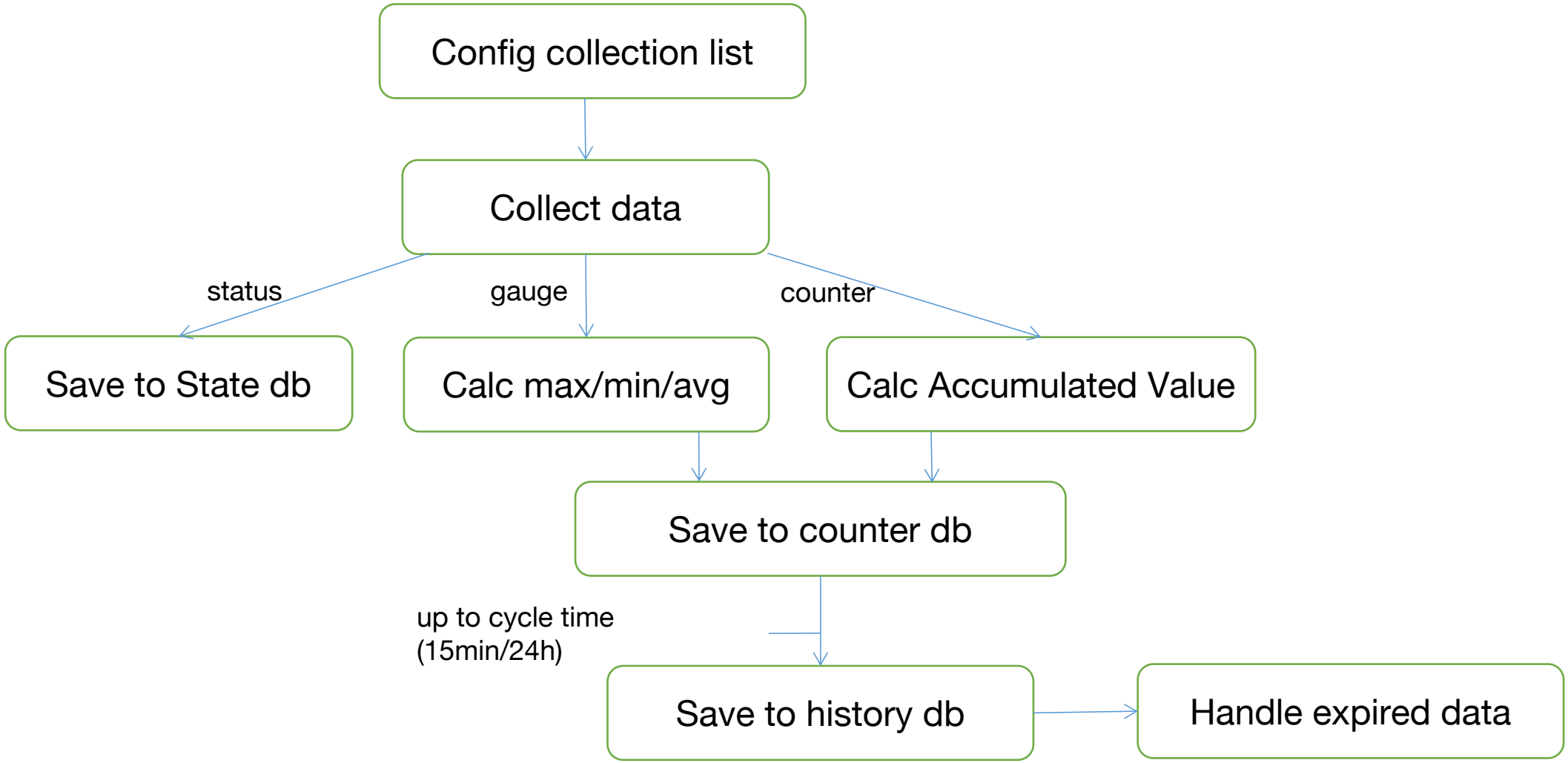


Performance Monitoring

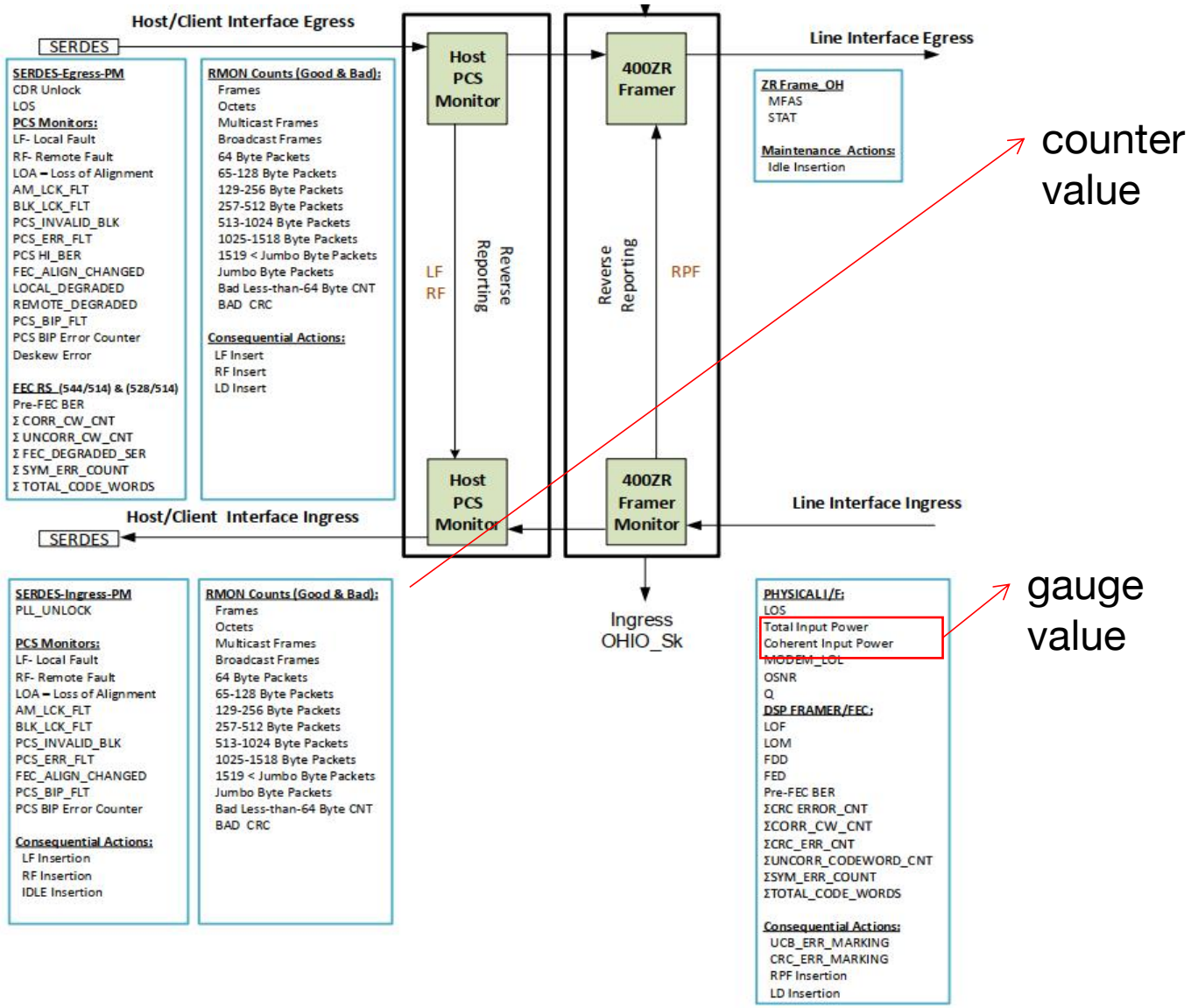
- PM is base on SONiC flex-counter functionality
- Three flex-counter groups are predefined:
  1. 1S_STAT_STATUS
  2. 1S_STAT_GAUGE (min/max/avg)
  3. 1S_STAT_COUNTER
- PM parameters (statistic/attribute ids) are predefined in <flex_counter_config.json>
- SyncD gets/calculates/stores all PM parameters per second.



How does syncd handle PM data



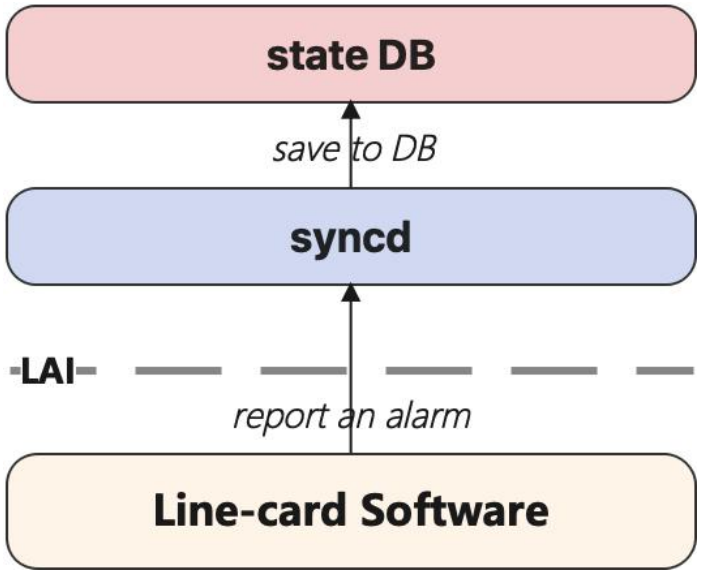
Examples of PM data




```
127.0.0.1:5000[10]> HGETALL
"TRANSCEIVER:TRANSCEIVER-1-1-
L1_InputPower:15_pm_history_1679269500000000000"
 1) "starttime"
 2) "1679269500000000000"
 3) "instant"
 4) "-9.87"
 5) "avg"
 6) "-9.88"
 7) "min"
 8) "-9.95"
 9) "max"
10) "-9.81"
11) "interval"
12) "900000000000"
13) "min-time"
14) "1679269502035000000"
15) "max-time"
16) "1679269762161000000"
17) "validity"
18) "complete"
```

Alarm notification

- When an alarm occurs (or disappears) on the line-card, lai library calls the callback function to store detail info of this alarm.
- Alarms are actively pushed by the line-card
- Three functions are implemented:
  1. real-time alarm(occurs): state db
  2. history alarm(disappears): history db
  3. event alarm(event): history db



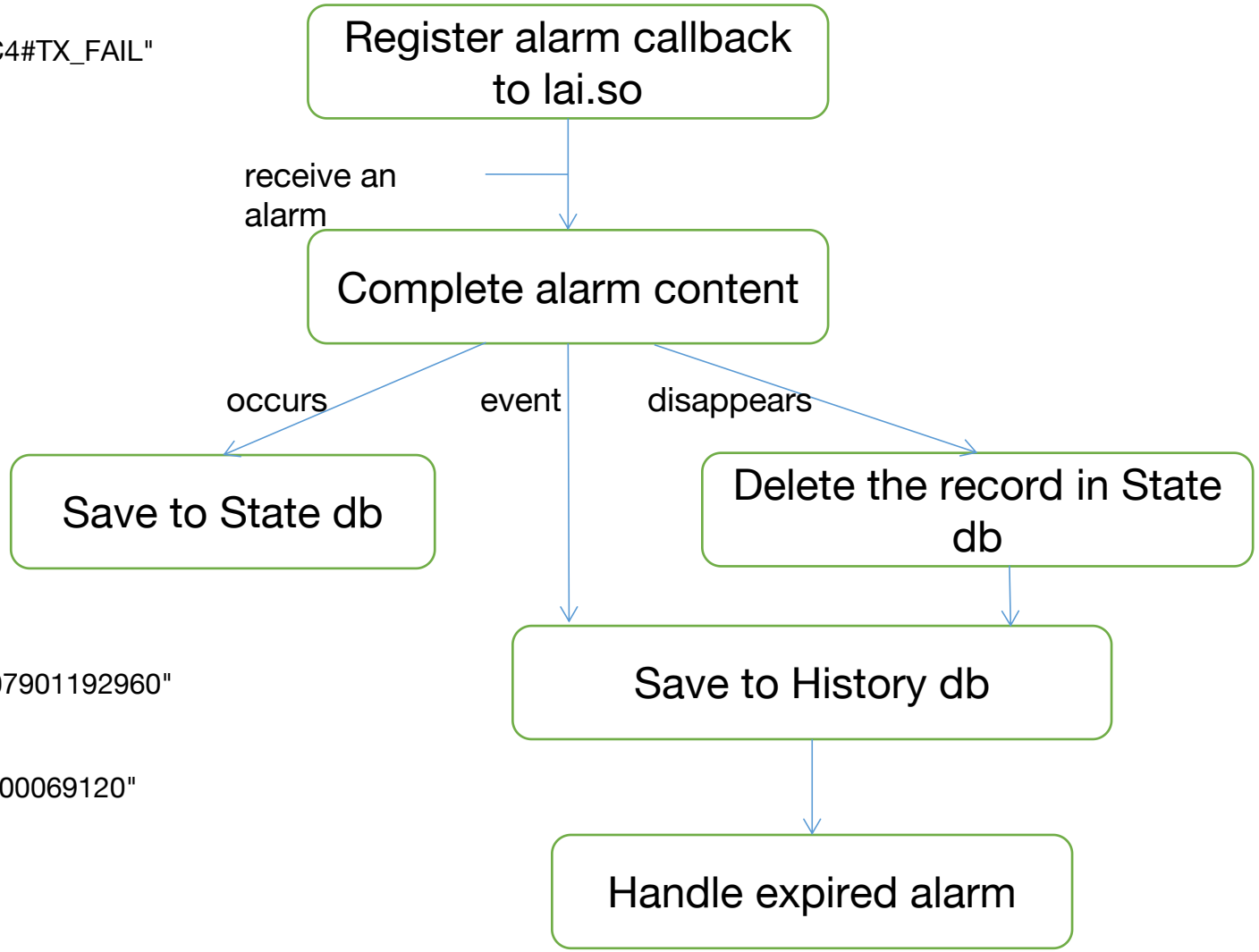
How does syncd handle after receiving an alarm

- State db

```
127.0.0.1:5000[6]> hgetall "CURALARM|PORT-1-1-C4#TX_FAIL"
 1) "id"
 2) "PORT-1-1-C4#TX_FAIL"
 3) "time-created"
 4) "1678928273571887104"
 5) "resource"
 6) "PORT-1-1-C4"
 7) "text"
 8) "#Tx Bias Low Alarm#TX-FAIL"
 9) "severity"
10) "NOT_ALARMED"
11) "type-id"
12) "TX_FAIL"
```

- History db

```
127.0.0.1:5000[10]> keys *
 1)"HISEVENT:PORT-1-1-C1#PORT INIT#1663135707901192960"
 2)"HISALARM:PORT-1-1-C1#OTN OPU
CSF#1663220810560164096"
 3)"HISALARM:PORT-1-1-C1#RX L05#1663135691600069120"
 4)"HISALARM:PORT-1-1-C1#OTN OPU
CSF#1663135691600150016"
 5)"HISALARM:PORT-1-1-C1#GE RX LOSS 0F
SYNC#1663220806253305088
```



Github repositories

sonic-lairedis:
https://github.com/zhengweitang-zwt/sonic-lairedis
sonic-olss:
https://github.com/zhengweitang-zwt/sonic-olss
sonic-swss-common:
https://github.com/zhengweitang-zwt/sonic-swss-common

Q & A