Warm reboot extension to PFC storm watchdog

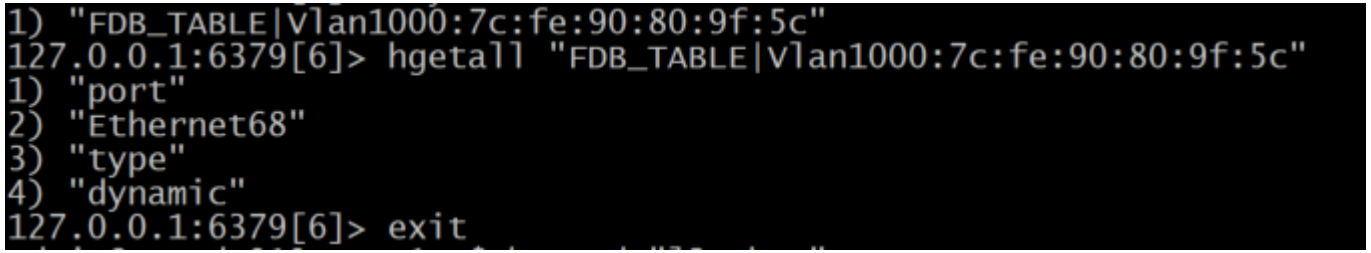
Content

- Problem statement
- Design
- Demo

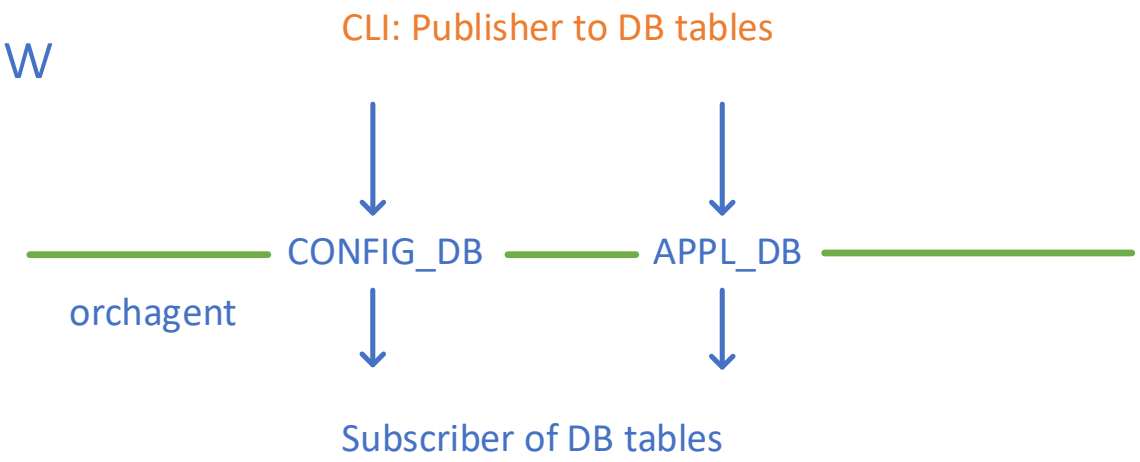
Warm reboot problem for a subsystem

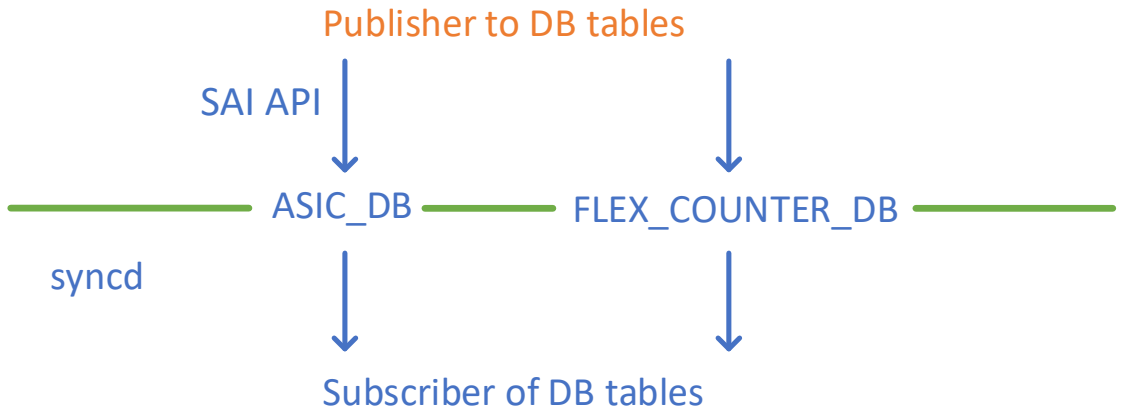
- warm-reboot = cold reboot + run-time generated states

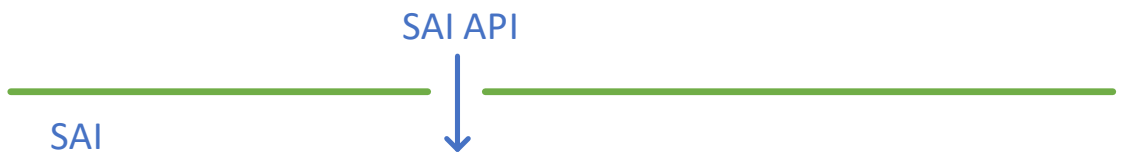
- Run-time generated states require warm-reboot handling
  - Recorded in DBs (CONFIG_DB, APPL_DB, and/or STATE_DB)
  - E.g., FDB/MAC table in STATE_DB (DB #6)



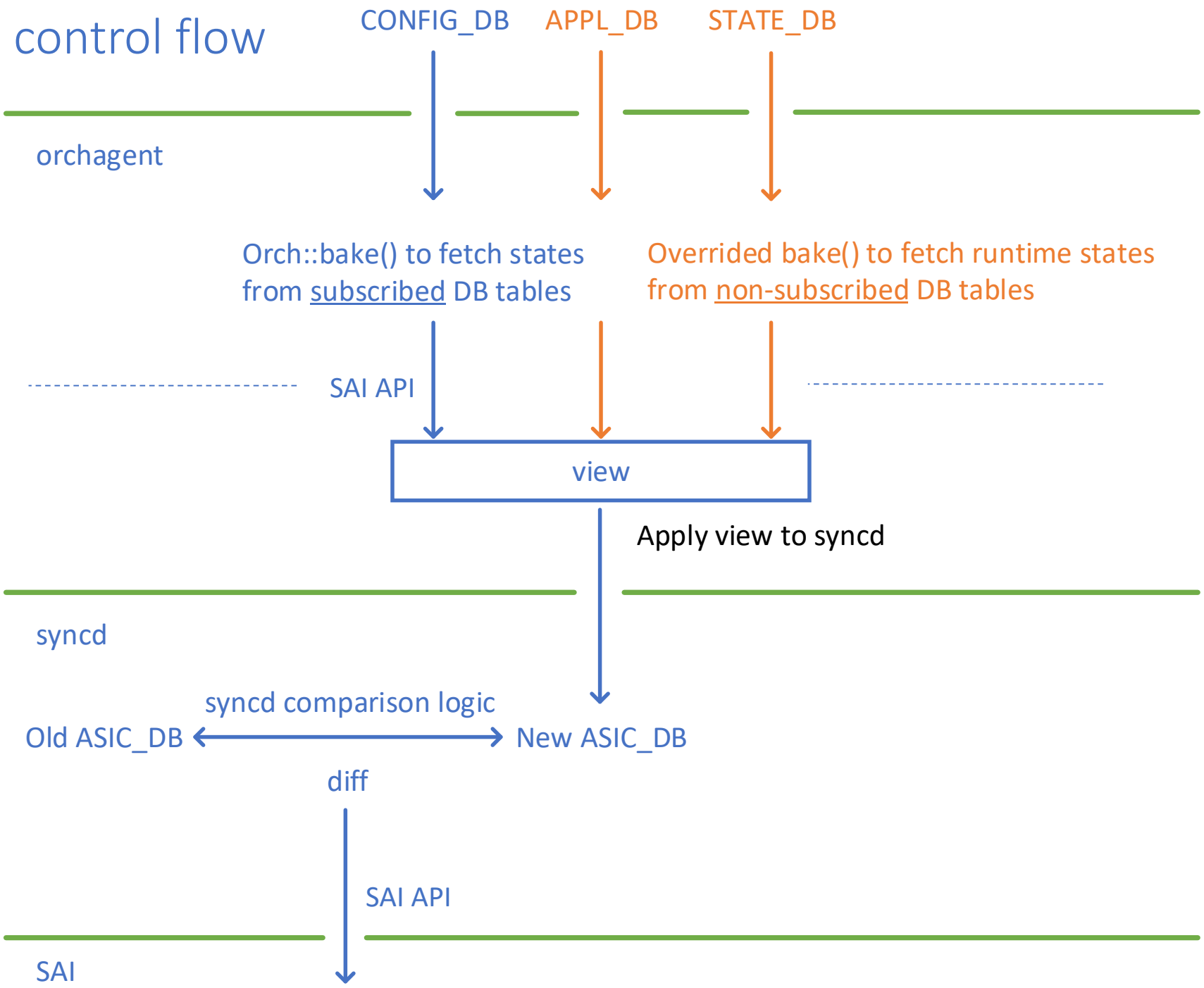
Cold-reboot control flow





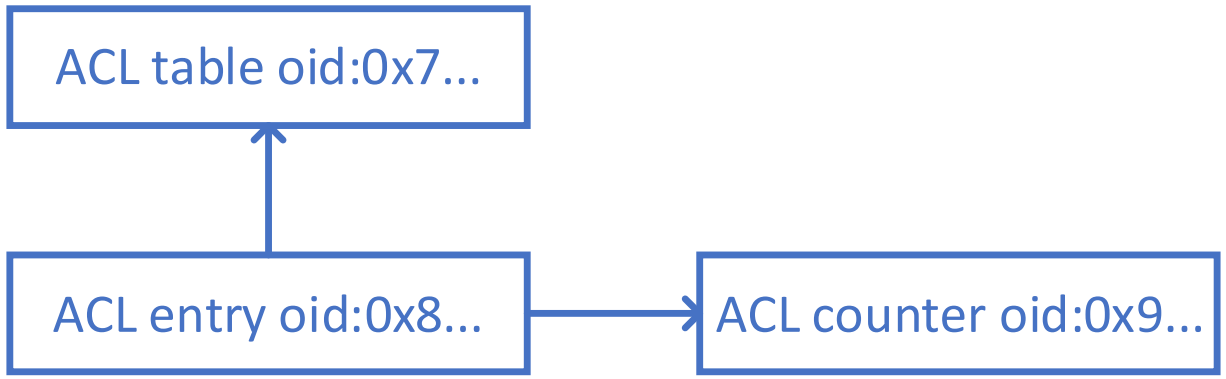


Warm-reboot control flow



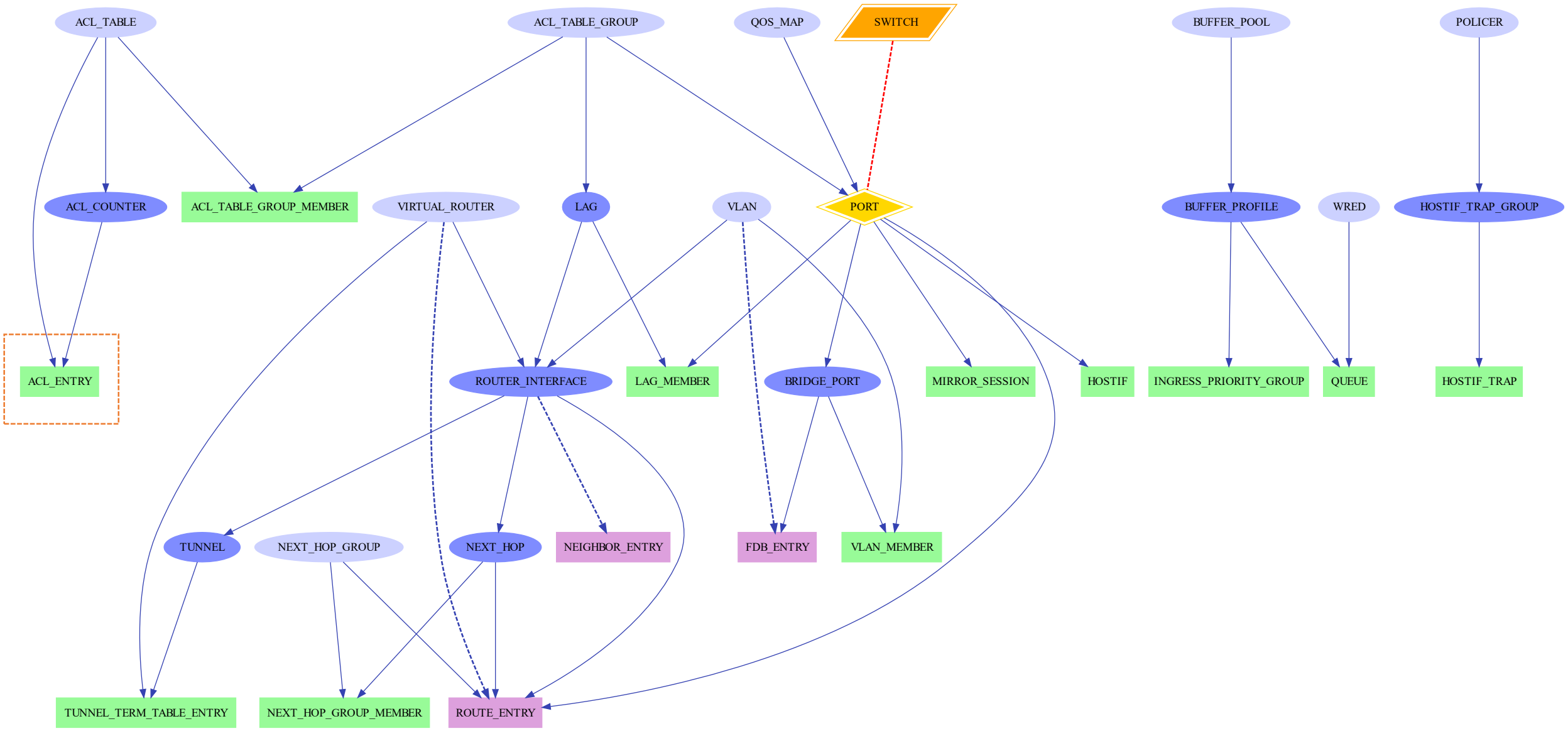
What is a view?

- View
  - Association between objects
- Objects
  - Representation of SAI/hardware entities
  - Identified uniquely by an OID




```
52 +    "ASIC_STATE:SAI_OBJECT_TYPE_ACL_ENTRY:oid:0x80000000006b5": {
53 +        "type": "hash",
54 +        "value": {
55 +            "SAI_ACL_ENTRY_ATTR_ACTION_COUNTER": "oid:0x90000000006b4",
56 +            "SAI_ACL_ENTRY_ATTR_ACTION_PACKET_ACTION": "SAI_PACKET_ACTION_DROP",
57 +            "SAI_ACL_ENTRY_ATTR_ADMIN_STATE": "true",
58 +            "SAI_ACL_ENTRY_ATTR_FIELD_TC": "4&mask:0xff",
59 +            "SAI_ACL_ENTRY_ATTR_PRIORITY": "999",
60 +            "SAI_ACL_ENTRY_ATTR_TABLE_ID": "oid:0x70000000006b2"
61 +        }
62 +    },
```

A bigger view

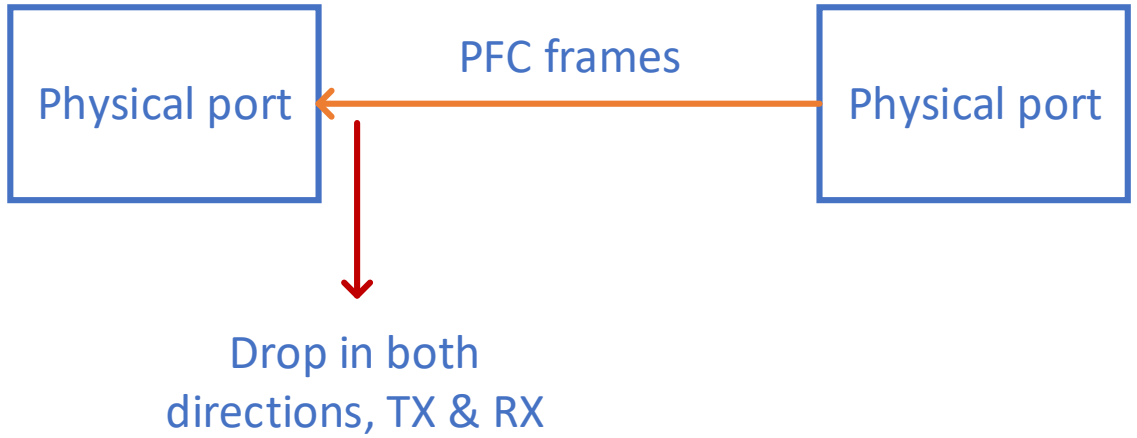


Cold-reboot vs. warm-reboot

- Cold-reboot
  - Listener to DB table states
- Warm-reboot
  - Fetch states from DB tables

PFC watchdog as an example

- Detection time
  - Time to deem the occurrence of a PFC storm



- Restoration time
  - Time to deem the dismiss of a PFC storm

PFC watchdog control flow internals

config_db.json

```
"Ethernet4": {
      "action": "drop",
      "detection_time": "200",
      "restoration_time": "200"
}
```

Cold start

Run time

sonic-cfggen:
Publisher of CONFIG_DB PFC_WD table

orchagent

PfcWdOrch:
Subscriber of CONFIG_DB PFC_WD table

PfcWdOrch::doTask()

PfcWdOrch:
Publisher to FLEX_COUNTER_DB
· FLEX_COUNTER_GROUP_TABLE:PFC_WD table

· FLEX_COUNTER_TABLE:PFC_WD:<port_id> table
· FLEX_COUNTER_TABLE:PFC_WD:<queue_id> table

PfcWdOrch:
Subscriber of COUNTERS_DB
PFC_WD_ACTION event channel

Install/Remove drop action

Start/Stop pfc wd stats update timer

syncd

Publisher of COUNTERS_DB
PFC_WD_ACTION event channel

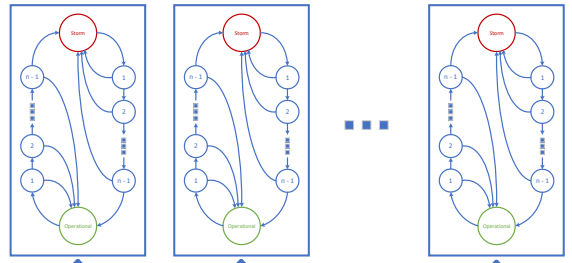
Subscriber of FLEX_COUNTER_DB
· FLEX_COUNTER_GROUP_TABLE:PFC_WD table

· FLEX_COUNTER_TABLE:PFC_WD:<port_id> table
· FLEX_COUNTER_TABLE:PFC_WD:<queue_id> table

PFC_WD flex counter thread

Ethernet4, queue3

Ethernet4, queue4



COUNTERS_DB

COUNTERS:oid:<port_id> table
COUNTERS:oid:<queue_id> table

SAI

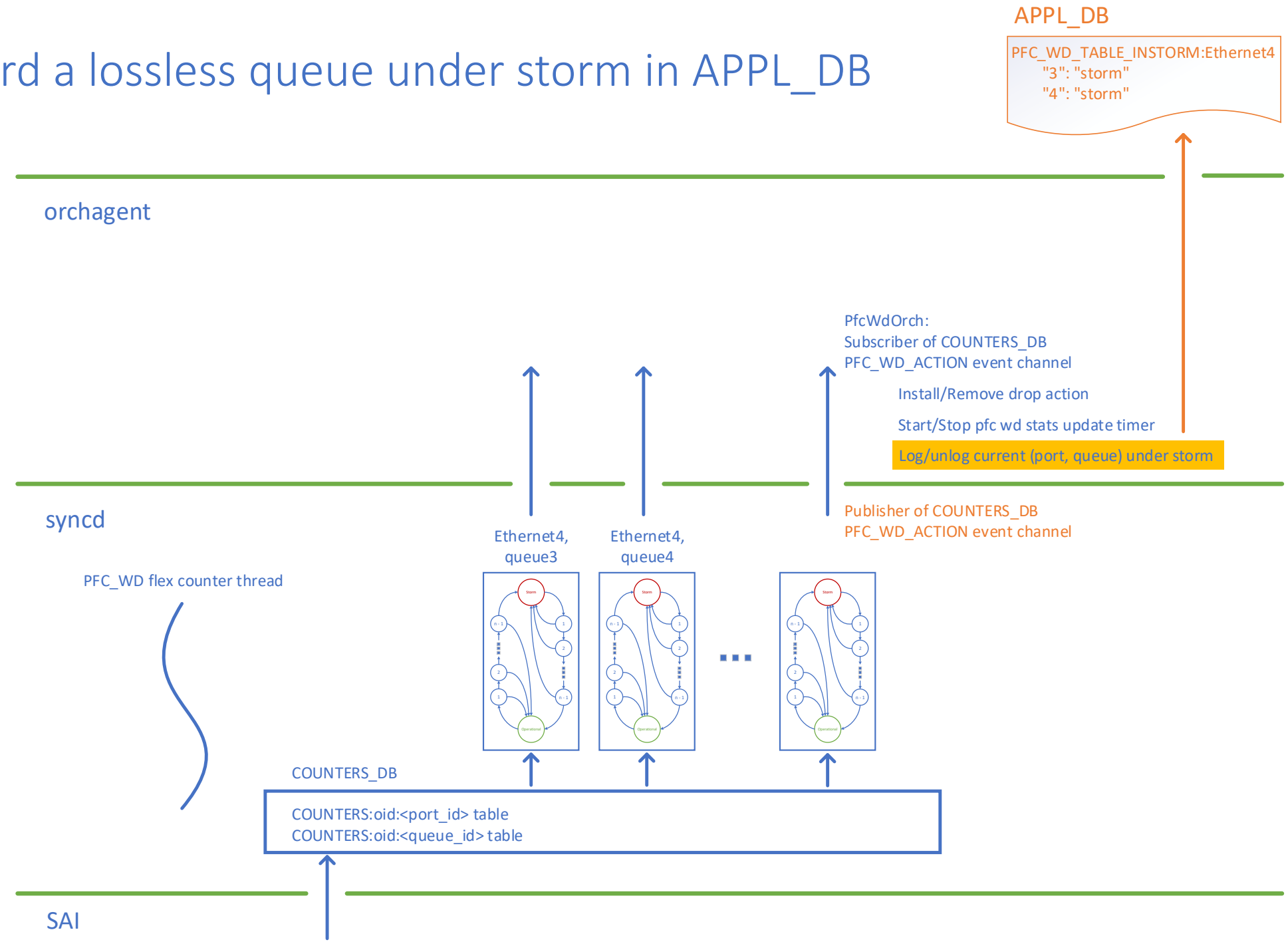
Run-time state for PFC watchdog

- Drop action for a lossless queue under storm
- State of PFC watchdog state machine per lossless queue

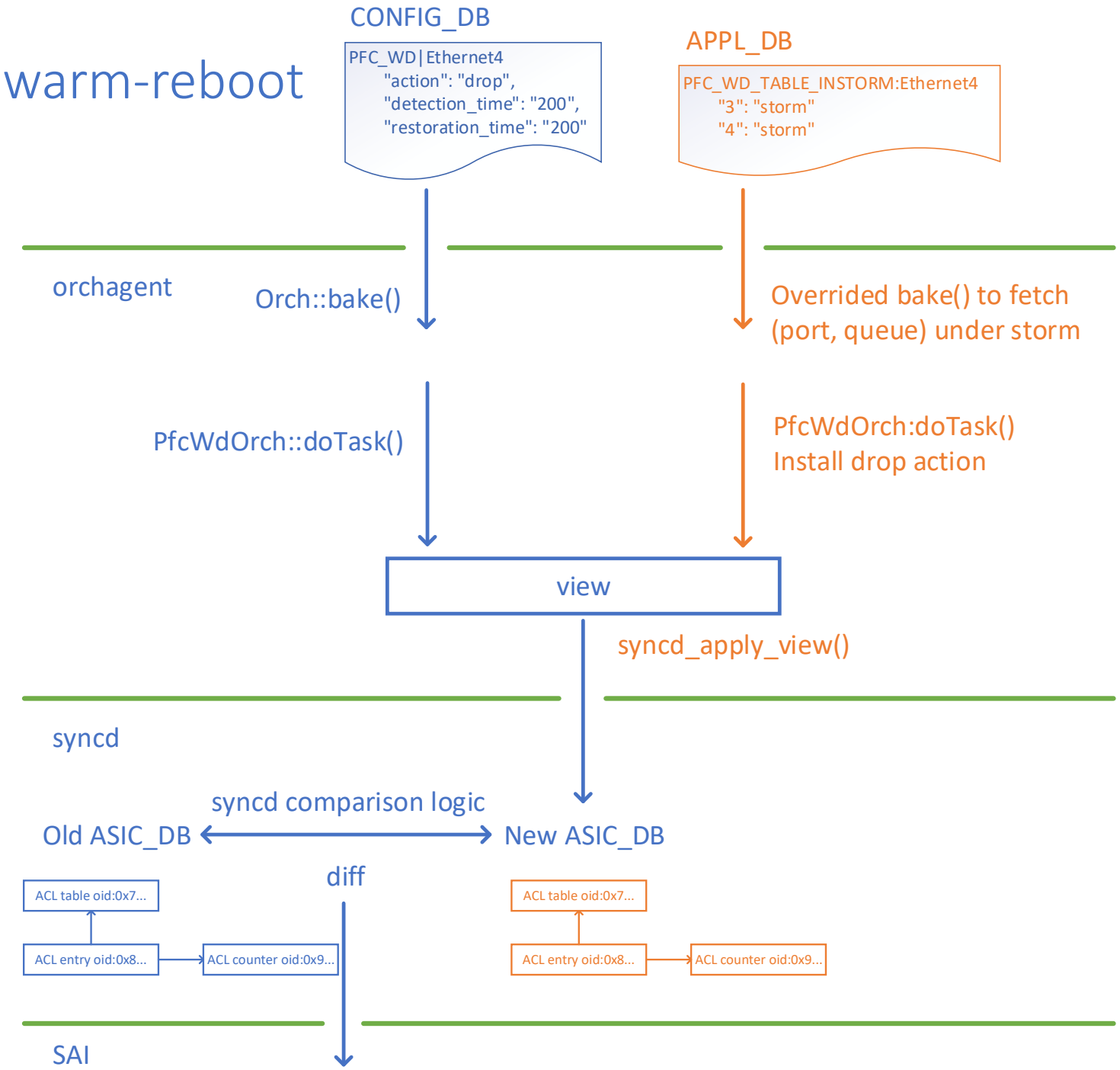
Warm-reboot for PFC watchdog

- Drop action for a lossless queue under storm
  - Record a lossless queue under storm in APPL_DB
  - Install drop action in warm-reboot

1) Record a lossless queue under storm in APPL_DB

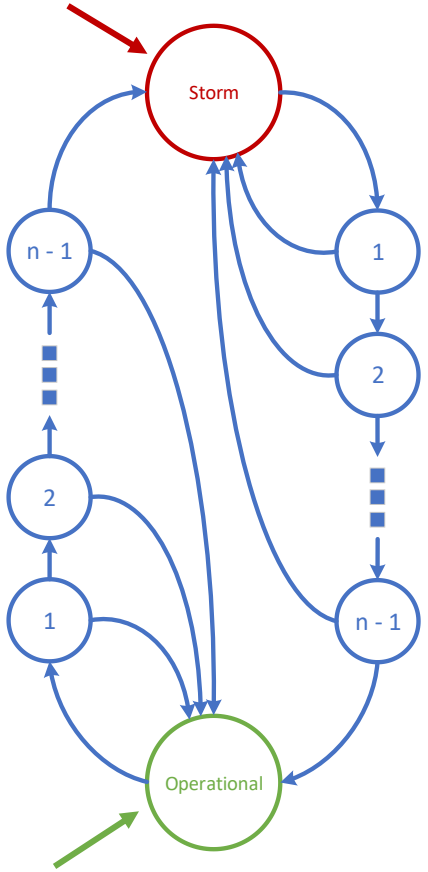


2) Install drop action in warm-reboot

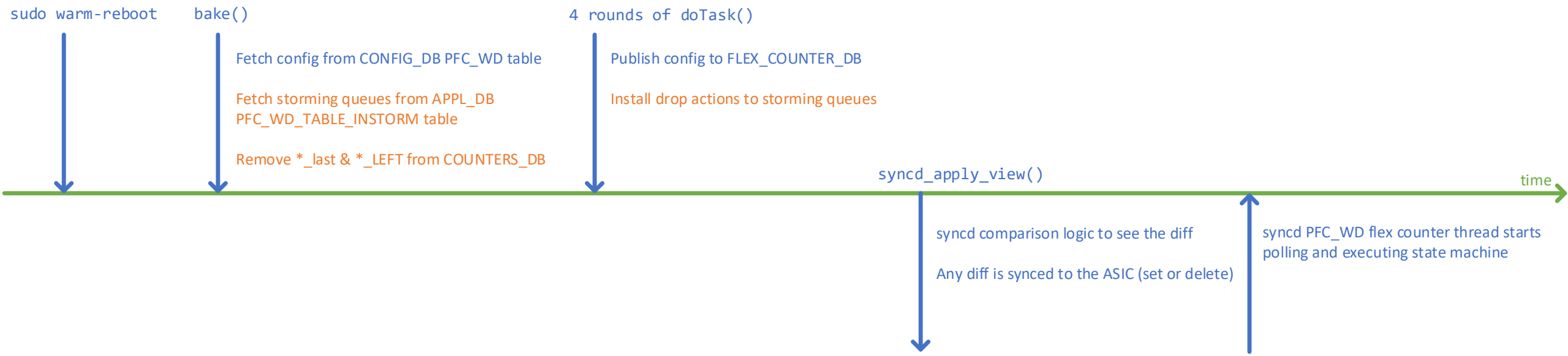


Warm-reboot for PFC watchdog

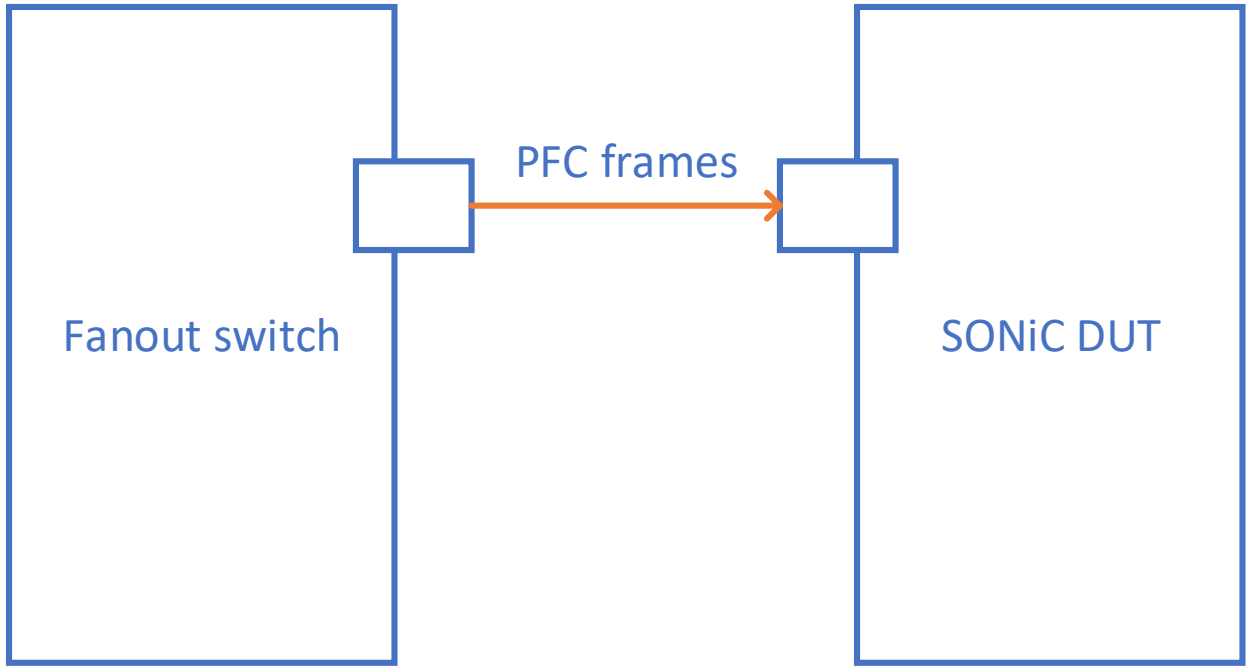
- State of PFC watchdog state machine per lossless queue
  - Kick-start state machine from a clean state---either operational or storm
    - Remove all *_last
    - Remove all *_LEFT



PFC watchdog warm-reboot flow

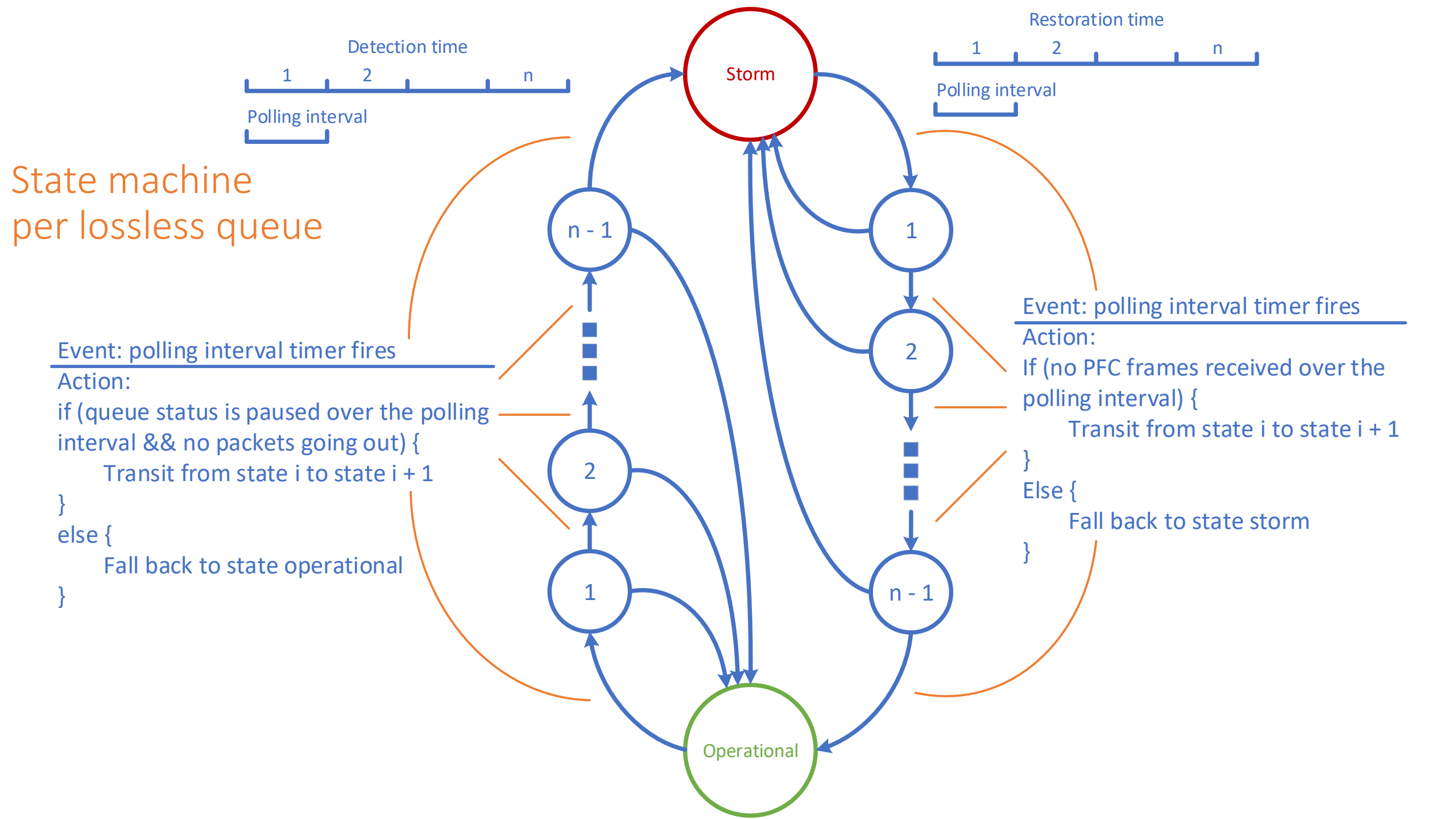


Demo



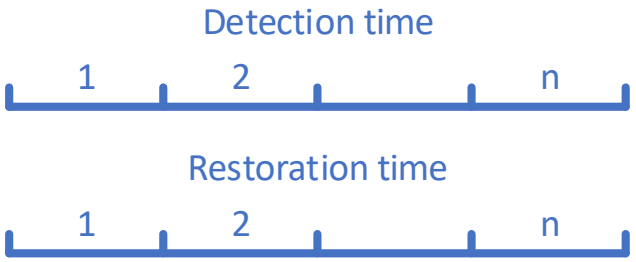
- orchagent is alive
- Storming queue starts in storm status with drop action installed

State machine per lossless queue



State machine implementation

- *_last
  - Lines 19 & 20
  - Lines 21 & 22
- *_LEFT
  - Lines 23 & 24
  - Lines 41 & 42
- PFC_WD_STATUS
  - Lines 11 & 12




```
admin@str-a7050-acs-1:~$ redis-cli -n 2 hgetall "COUNTERS:oid:0x150000000001bf"
 1) "PFC_WD_DETECTION_TIME"
 2) "200000"
 3) "PFC_WD_RESTORATION_TIME"
 4) "200000"
 5) "PFC_WD_ACTION"
 6) "drop"
 7) "PFC_WD_QUEUE_STATS_DEADLOCK_DETECTED"
 8) "2"
 9) "PFC_WD_QUEUE_STATS_DEADLOCK_RESTORED"
10) "1"
11) "PFC_WD_STATUS"
12) "stormed"
13) "SAI_QUEUE_STAT_PACKETS"
14) "0"
15) "SAI_QUEUE_STAT_CURR_OCCUPANCY_BYTES"
16) "0"
17) "SAI_QUEUE_ATTR_PAUSE_STATUS"
18) "true"
19) "SAI_QUEUE_ATTR_PAUSE_STATUS_last"
20) "true"
21) "SAI_QUEUE_STAT_PACKETS_last"
22) "0"
23) "PFC_WD_DETECTION_TIME_LEFT"
24) "200000"
25) "PFC_WD_QUEUE_STATS_TX_PACKETS"
26) "0"
27) "PFC_WD_QUEUE_STATS_TX_DROPPED_PACKETS"
28) "0"
29) "PFC_WD_QUEUE_STATS_RX_PACKETS"
30) "0"
31) "PFC_WD_QUEUE_STATS_RX_DROPPED_PACKETS"
32) "0"
33) "PFC_WD_QUEUE_STATS_TX_PACKETS_LAST"
34) "0"
35) "PFC_WD_QUEUE_STATS_TX_DROPPED_PACKETS_LAST"
36) "0"
37) "PFC_WD_QUEUE_STATS_RX_PACKETS_LAST"
38) "0"
39) "PFC_WD_QUEUE_STATS_RX_DROPPED_PACKETS_LAST"
40) "0"
41) "PFC_WD_RESTORATION_TIME_LEFT"
42) "200000"
43) "SAI_QUEUE_STAT_BYTES"
44) "0"
45) "SAI_QUEUE_STAT_DROPPED_PACKETS"
46) "0"
47) "SAI_QUEUE_STAT_DROPPED_BYTES"
```

Warm-reboot for PFC watchdog

- State of PFC watchdog state machine per lossless queue
  - Kick-start state machine from a clean state