SmartSwitch – High Availability Overview

Riff Jiang

Outline

- What is HA?
- HA scope and ENI pair placement
- Network setup for HA and traffic forwarding
- Control plane overview and ENI programming model
- HA state machine and operations

HA for ENI

What is HA (High Availability)?

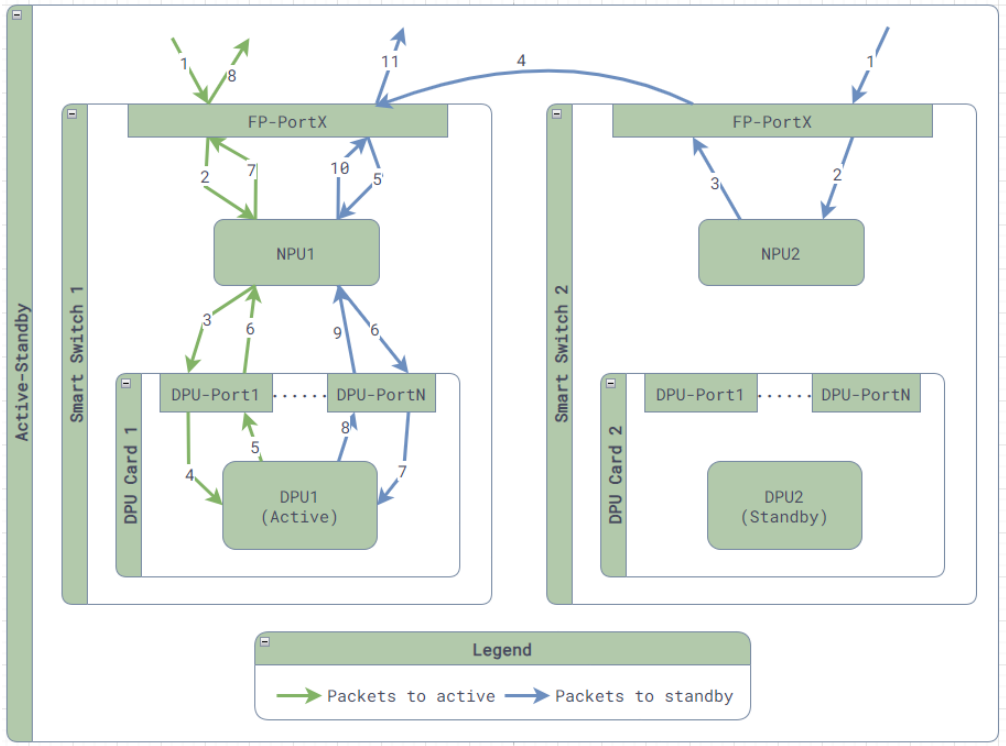
- A single switch / DPU goes down or when a network failure happens, it will not:
  - Kill all the traffic for an ENI (vNIC).
  - Causing all existing flows to be dropped.
- Goals: https://github.com/sonic-net/DASH/blob/main/documentation/high-avail/high-availability-and-scale.md.
  - 0 downtime on planned switchover.
  - <2 sec downtime on unplanned failover to standalone setup for each ENI.
  - Ability to resume connections in the event of both planned and unplanned failover.
  - After both planned and unplanned failover and recovery, the flow on all DPUs will be aligned.
  - …

So, what is HA?

- **Flow HA:** Each ENI will be backed up by 2 DPUs, so flows can be replicated between them, and won't be dropped when one DPU/Switch is having problem.
- **Data path HA:** Handles network failures and reduce the chance of packet drops.

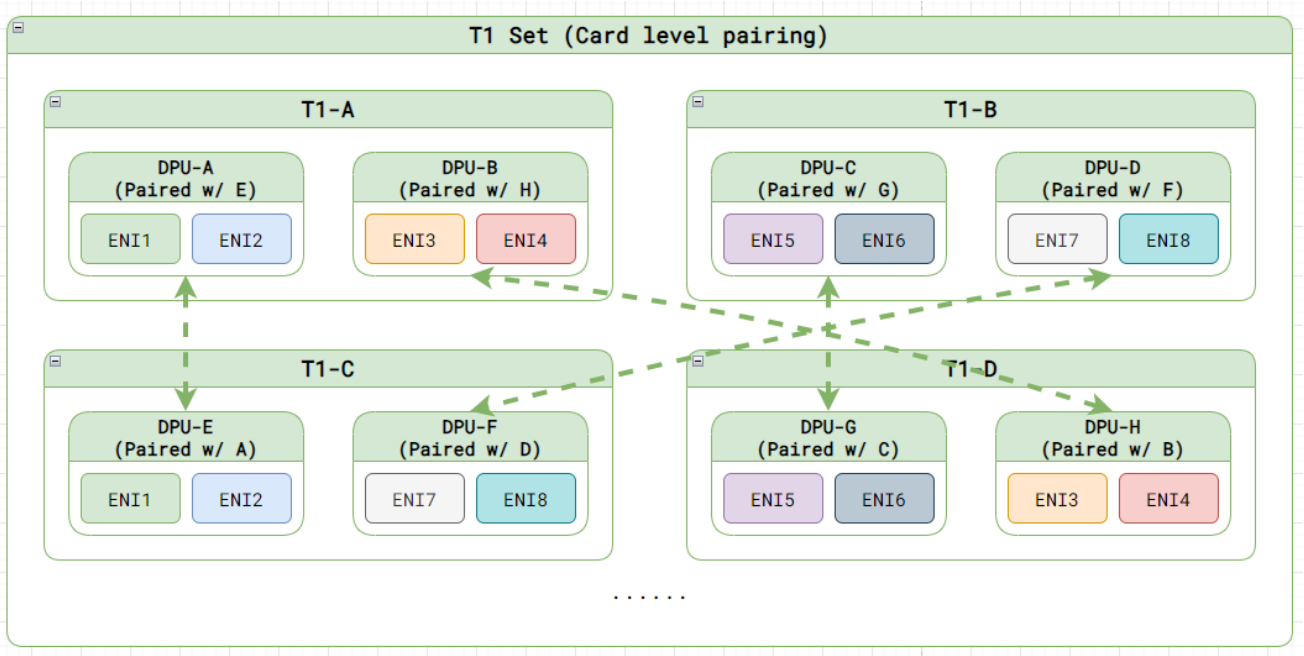
ENI Flow HA – Active/Standby

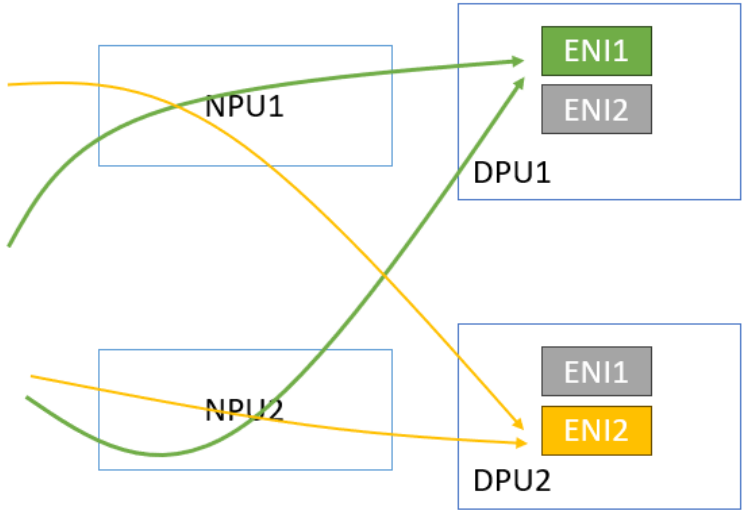
- Each ENI must be placed in 2 DPUs from 2 different switch forming a HA pair.
- 1 DPU serves as active and making flow decisions, while the other DPU serves as standby, acting as a flow storage, only accepting flows replicated from active.
- In steady state, all flows will be inline sync'ed from active to standby. When a DPU rejoins the HA pair, we bulk sync the flows.
- When the active DPU runs into issues, we will failover the active, make the standby the new active, and switch over the traffic to avoid further impact.



ENIs in SmartSwitch

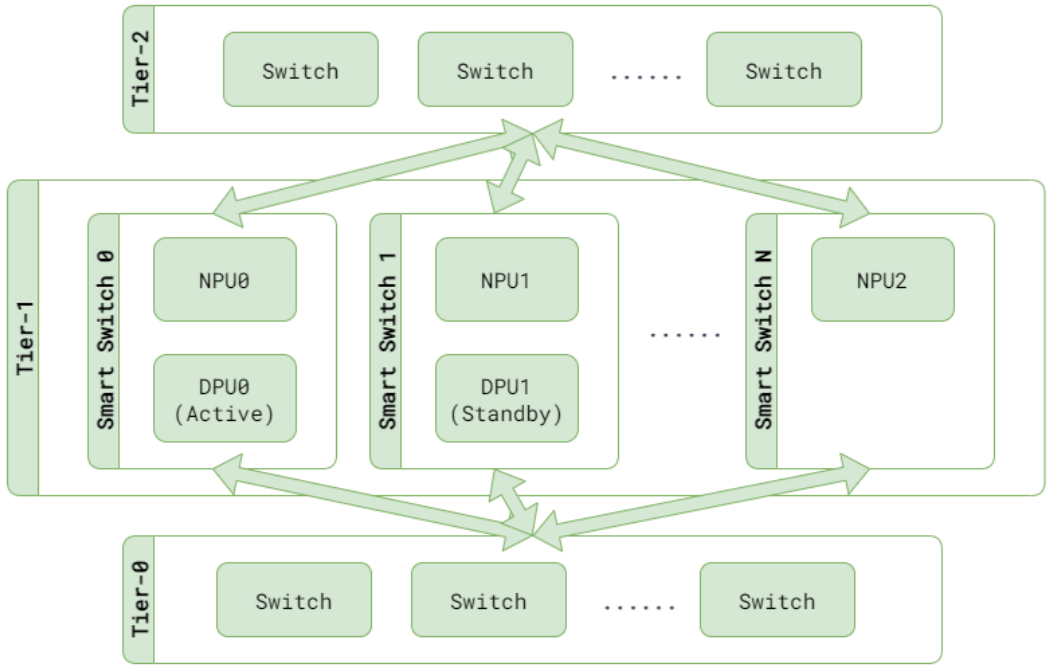
- Card-level ENI pairing
- ENI-level Active/Standby





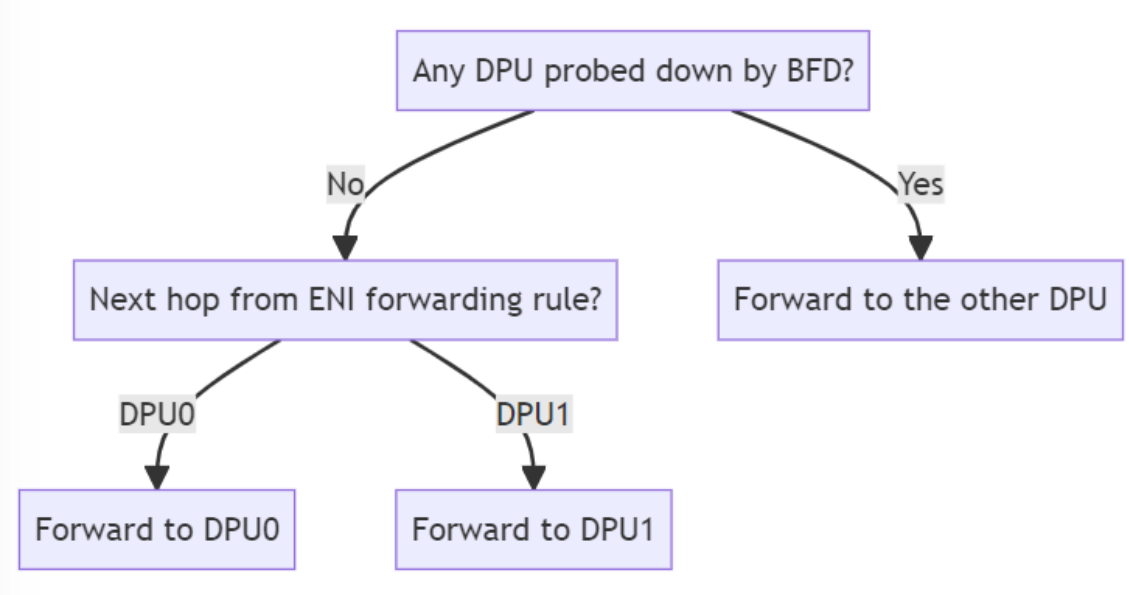
Data Path HA

- All T1s advertise the same VIPs.
- When a T1 receives a packet for an ENI, it forward the packet directly to the active DPU using a VxLan tunnel.
- Handles single switch failure and no more waiting for BGP reconcile.



Controlling Traffic Forwarding

- Card-level NPU-to-DPU probing
  - NPU sending BFD probe to DPU to check if DPU is alive or not.
  - When probed down, NPU will stop forwarding all traffic to this DPU.
  - When probed up, NPU will respect the ENI-level traffic control.
- ENI-level NPU-to-DPU traffic control
  - Explicitly setting up the next hop to the current active side.



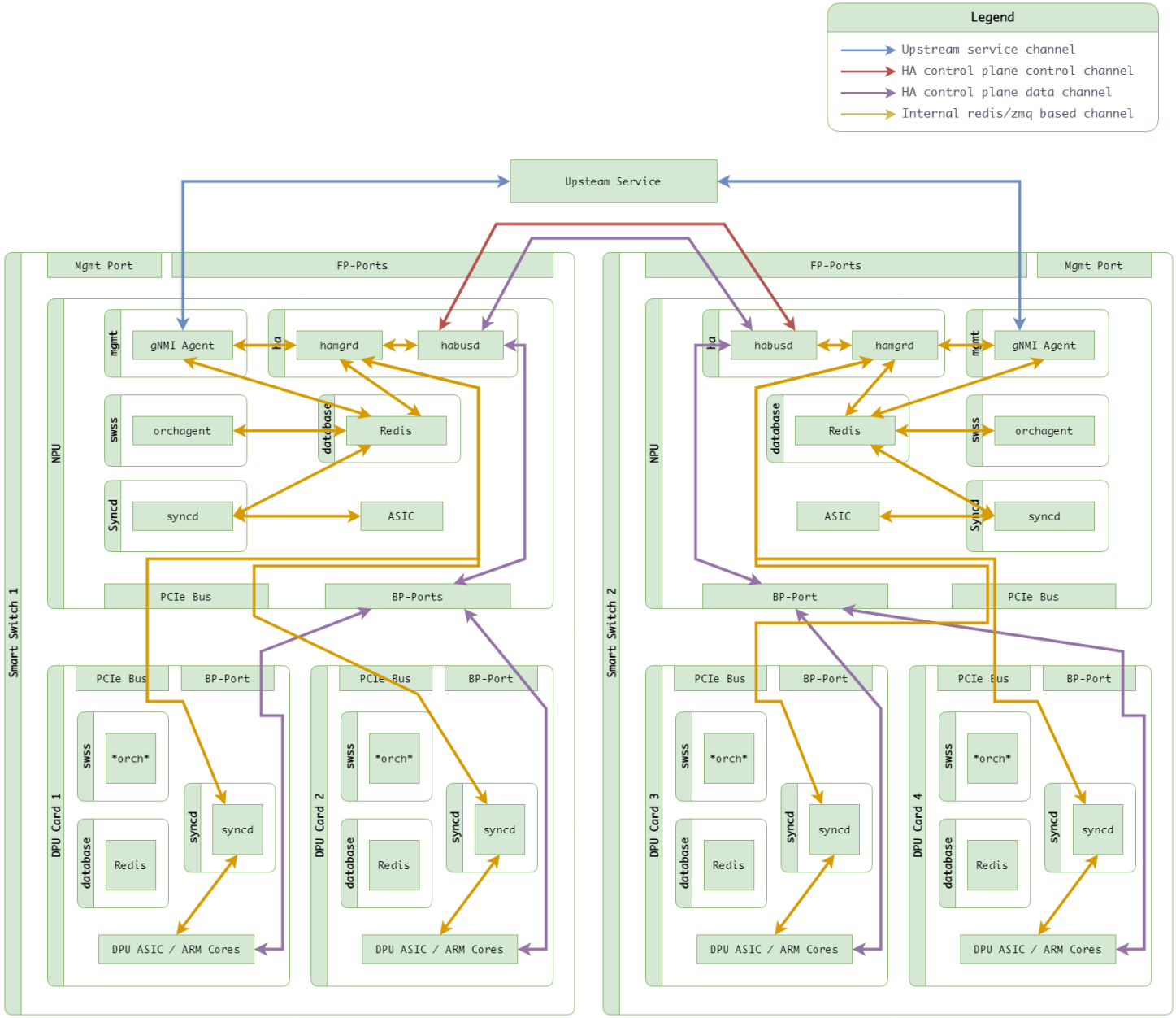
HA Control Plane Overview

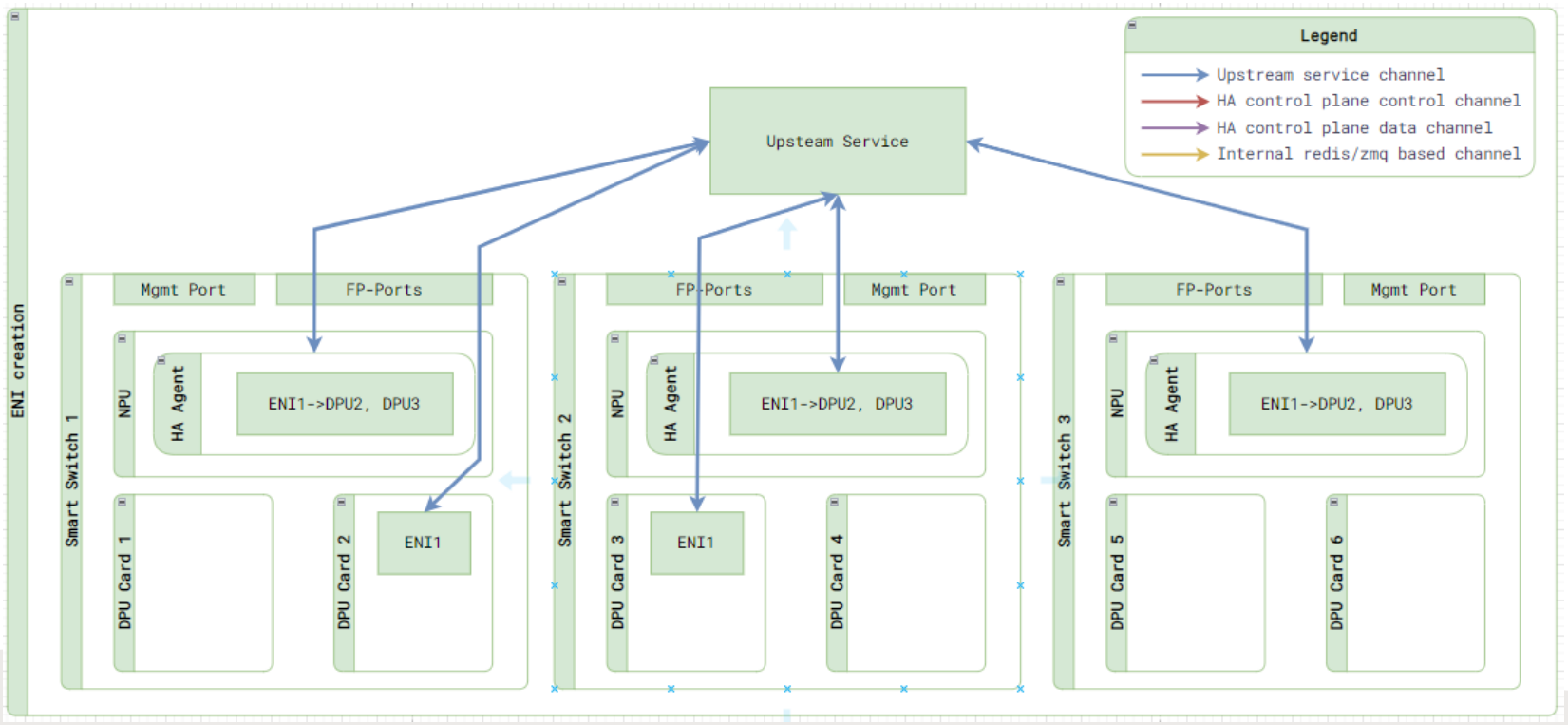
Control Plane Overview

- Upstream Service (SDN controller):
  - Decide ENI placement and pairing.
  - Decide preferred active ENI placement.
  - Decide desired HA states under planned events, such as planned switchover, planned shutdown for upgrades, evening traffic and etc.
  - Triggering HA operations for manual live site mitigations.
- SmartSwitch:
  - Drive the HA state machine transitions.
  - Report every ENI HA state change and reasons, so upstream service knows what is happening and can make decisions for planned events.
  - Handle HA related requests from upstream service.
  - Monitor and handle unplanned events, and trigger defined mitigations, such as driving to standalone setup.

HA communication channels

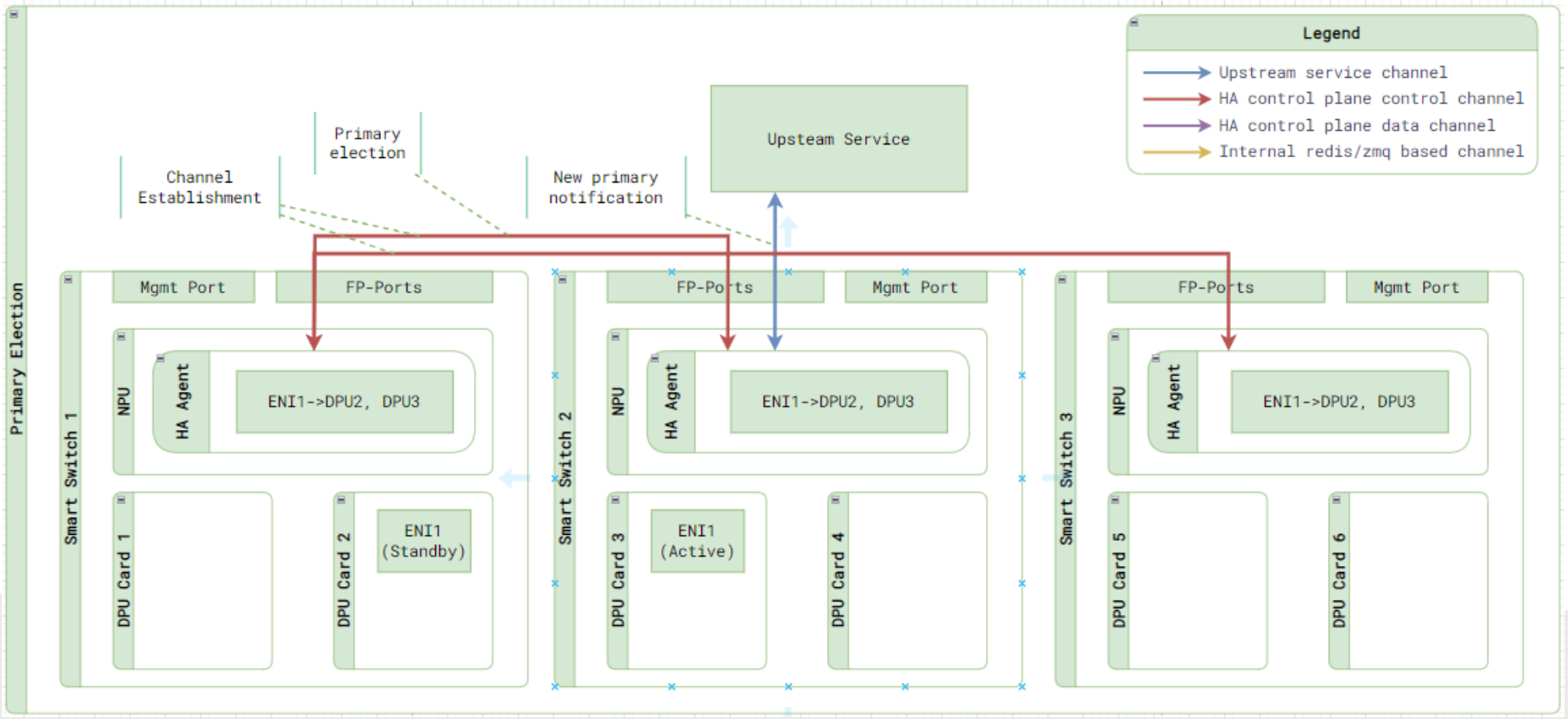
- Upstream Service Channel
  - Goal state programming
- HA Control Plane Channels (gRPC)
  - HA Control Plane Control Channel
    - Driving HA state transitions
    - Updating ENI traffic forwarding rules
    - Guaranteed to work within a bounded time
  - HA Control Plane Data Channel
    - Bulk sync
- Data Plane Channel (Tunnel)
  - Inline sync





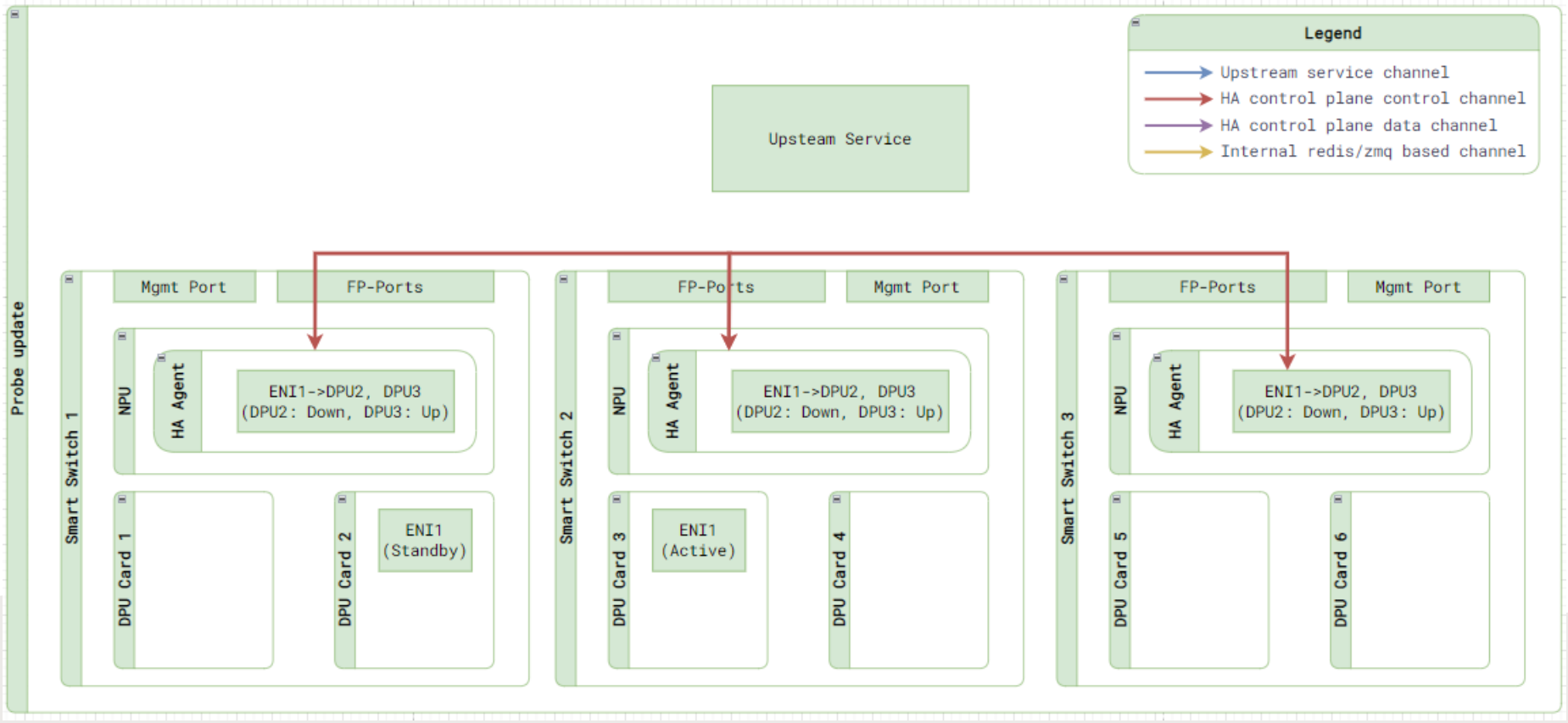
ENI Creation (Step 1)

- Upstream service will first decide where to put the new ENI and form the HA set.
- Upstream service calls northbound interface and programs the following things on each SmartSwitch independently
  - Create the ENI on selected DPUs with its peer information, so we can form a HA set.
  - Program traffic forwarding rules to all the switches that will receive the traffic.



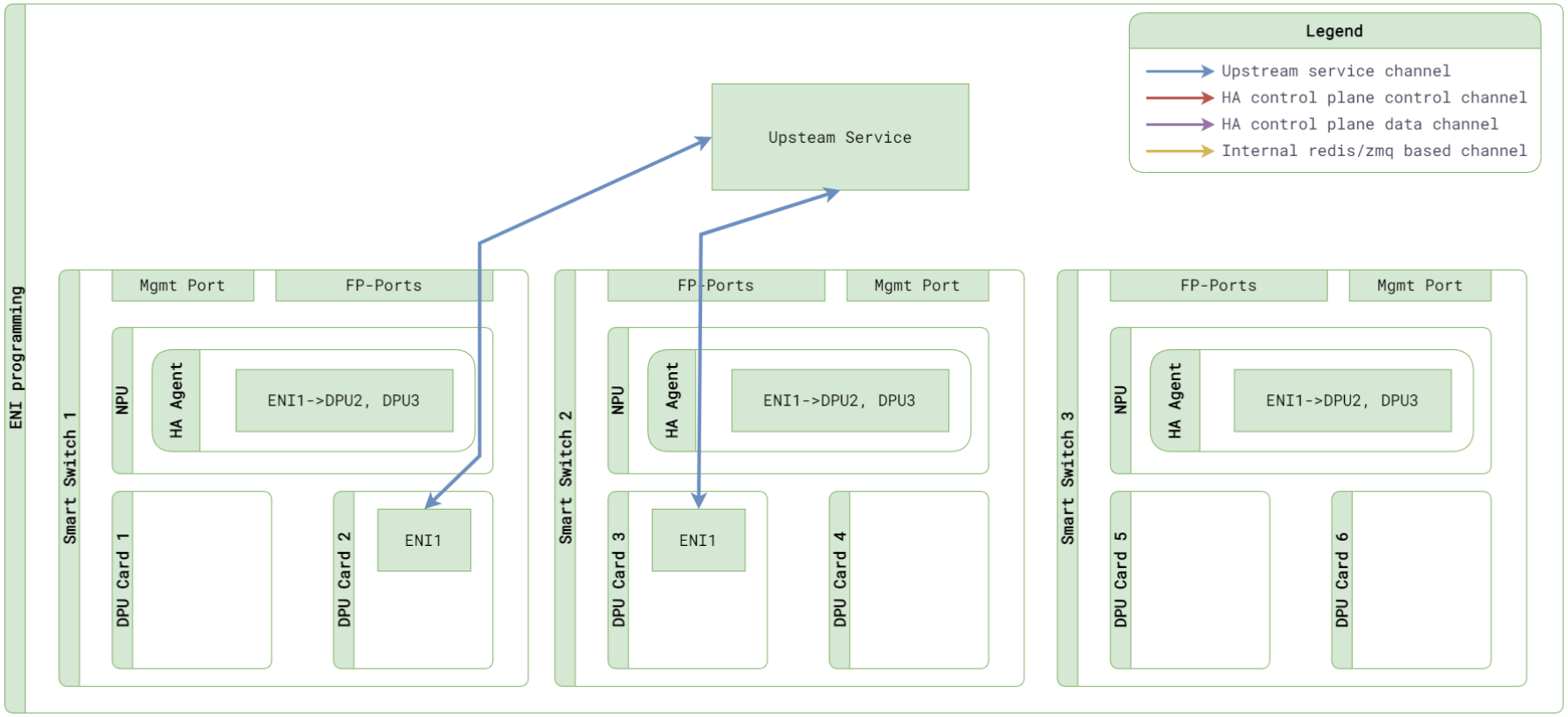
ENI Creation (Step 2)

- Once programming is finished, the 2 ENIs will start forming the HA pair with:
  - Control plane channels created
  - New active elected with the instruction that is specified by upstream service.
- Once the new primary is elected, SmartSwitch will notify the upstream service that the primary is selected.



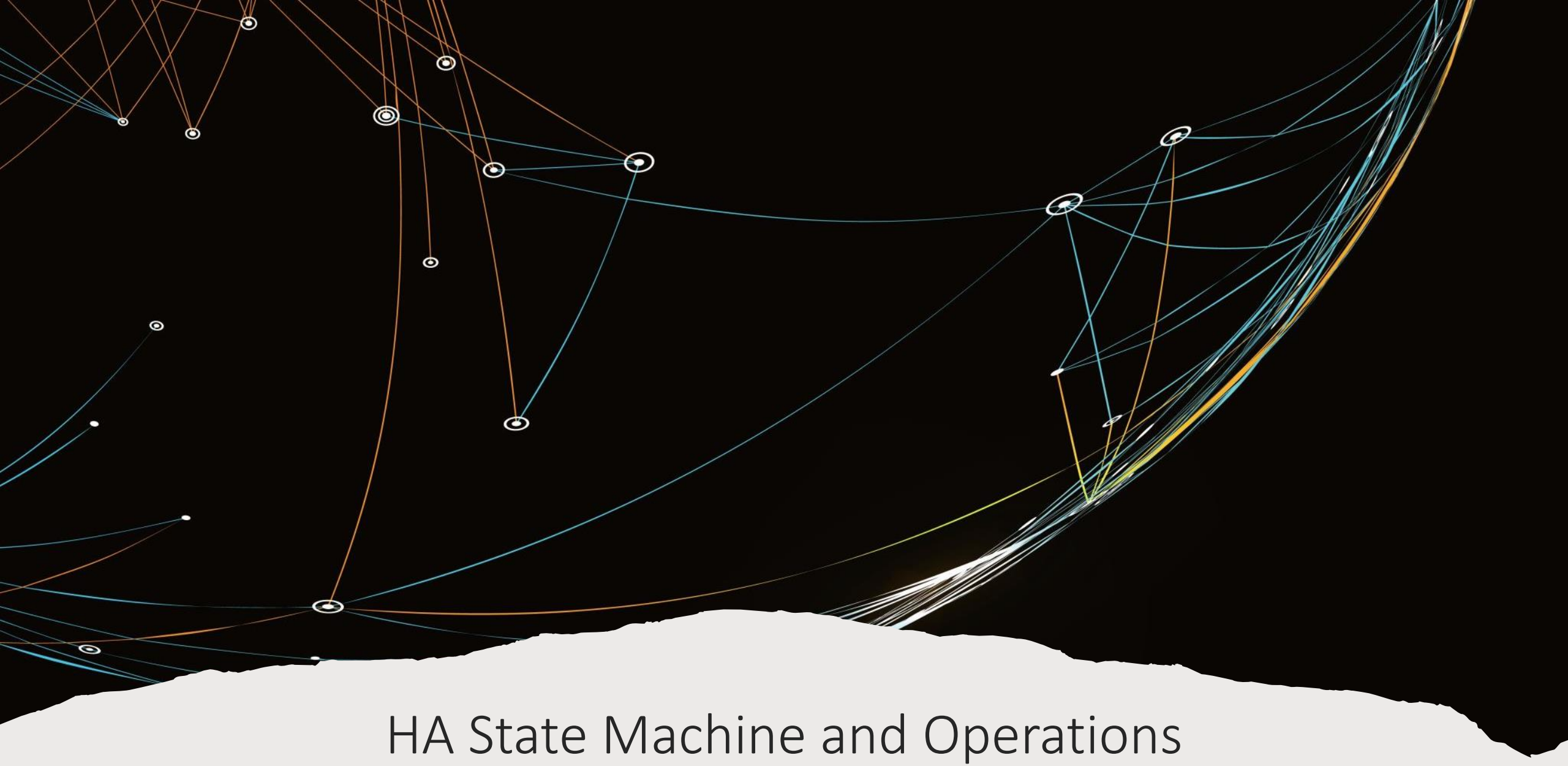
ENI Creation (Step 3)

- The primary election process will also update the probe state of all ENIs, making sure the traffic is forwarded to the right DPU.



ENI Programming

- For all future goal state update, upstream service will program both ENI independently.



HA State Machine and Operations

Planned Events vs Unplanned Events

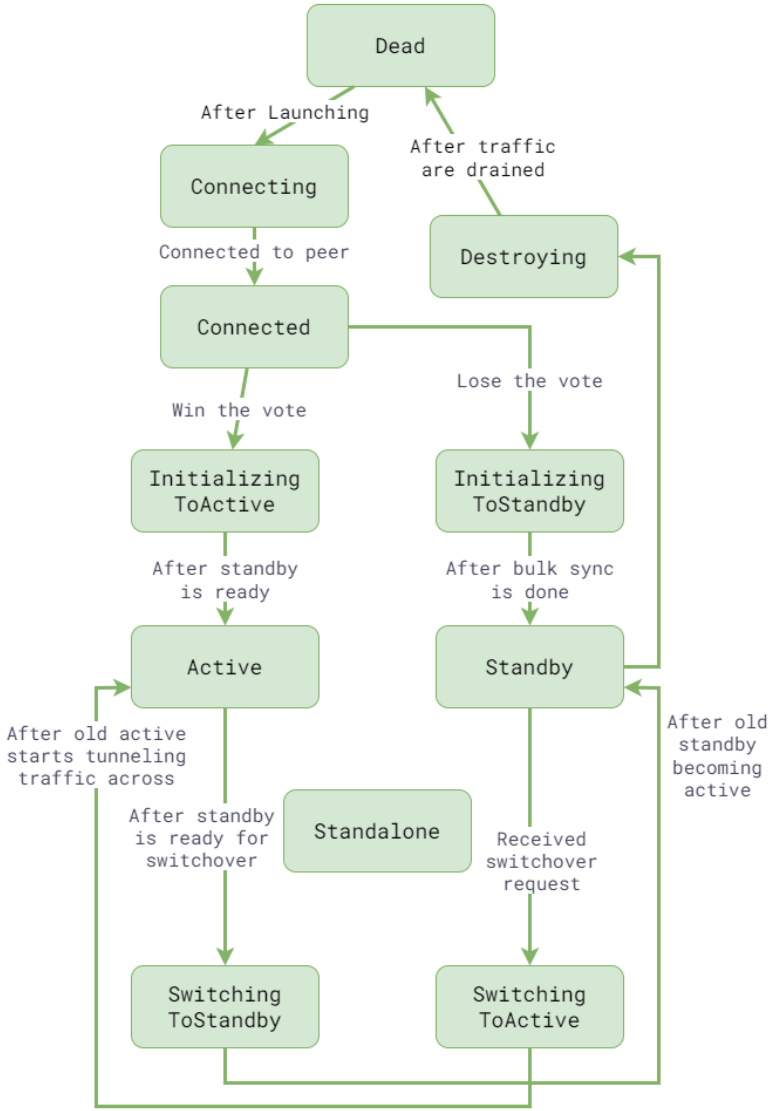
- Planned events:
  - ENI launch, Switchover, Shutdown, Migration, etc.
  - Goal: 0 down time. Avoid bulk sync / flow merge as long as we can.
  - Initiated from and approved by upstream service.
- Unplanned events:
  - Network failure, DPU/NPU/PCIe failure.
  - Goal: <2s failover to standalone setup.
  - Initiated and driven from SmartSwitch w/ predefined config from upstream service.

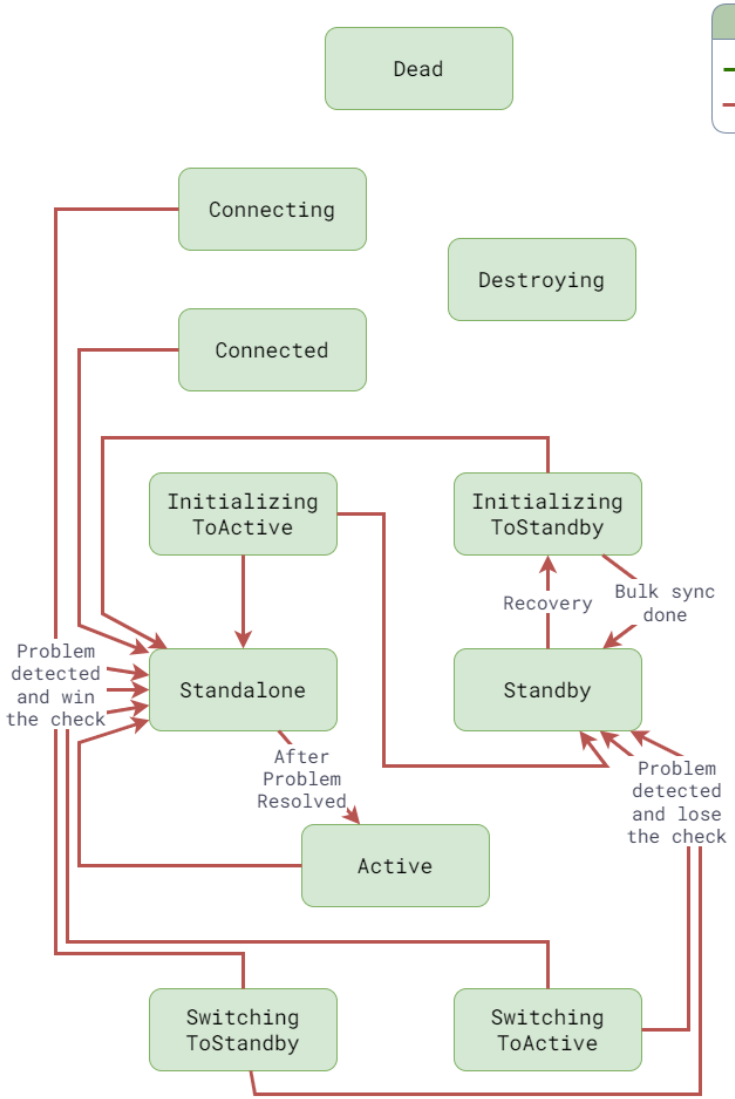
HA State Machine

- Key states: Dead, Active, Standby, Standalone
- Other transition states can be found in appendix.

State	Definition	Receive Traffic from NPU?	Make decision?	Handling old flow?	Respond flow sync?	Init flow sync?	Init Bulk sync?
Dead	HA participant is just getting created, and not connected yet.	No	Drop	Drop	No	No	No
Active	Connected to pair and act as decision maker.	Yes	Yes	Yes	No	Yes	Yes
Standby	Connected to pair and act as backup flow store.	No	Tunneled to pair	Tunneled to pair	Yes	No	No
Standalone	Heartbeat to pair is lost. Acting like a standalone setup.	Yes	Yes	Yes	Yes	No	No

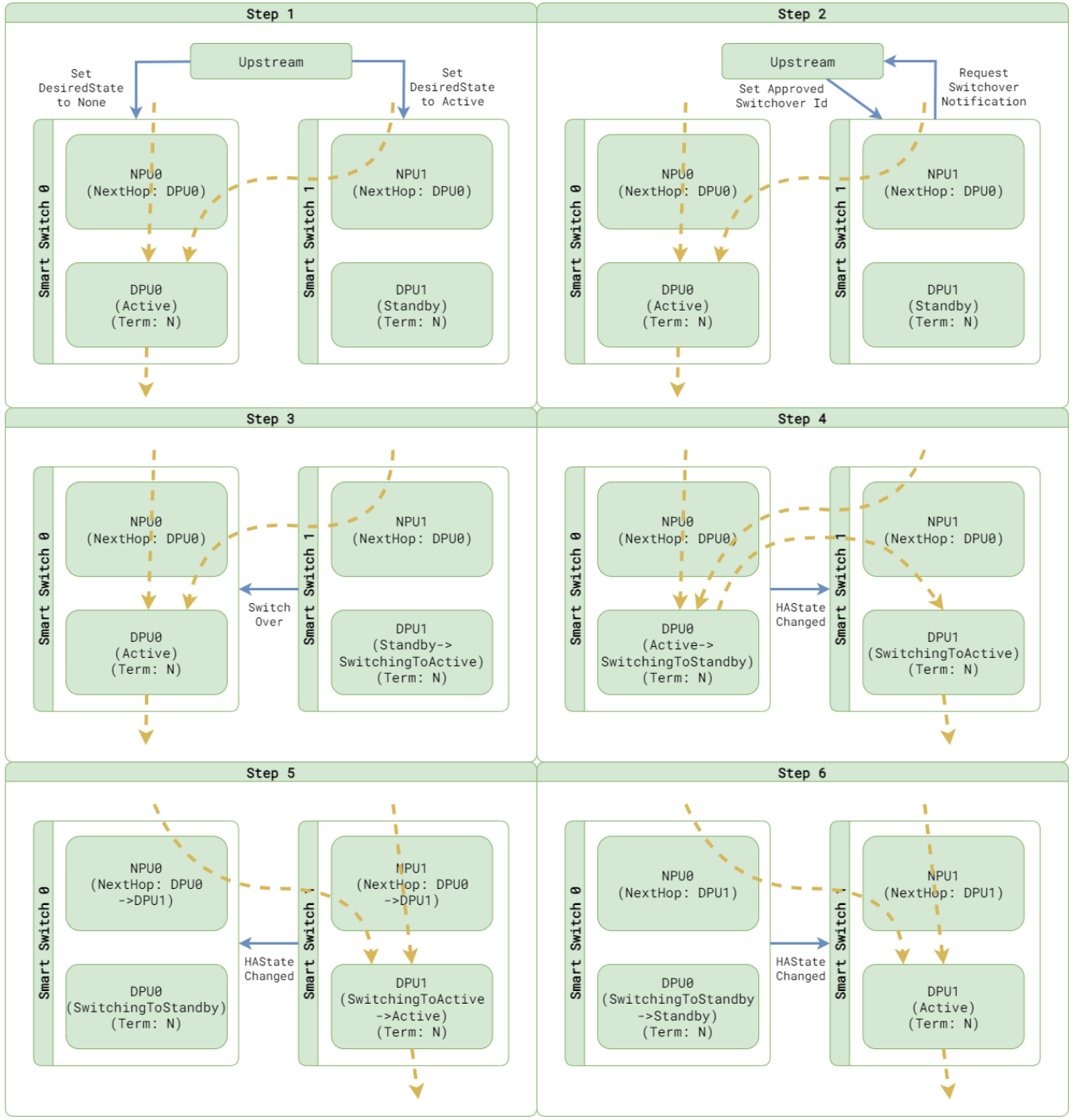
HA State Transition

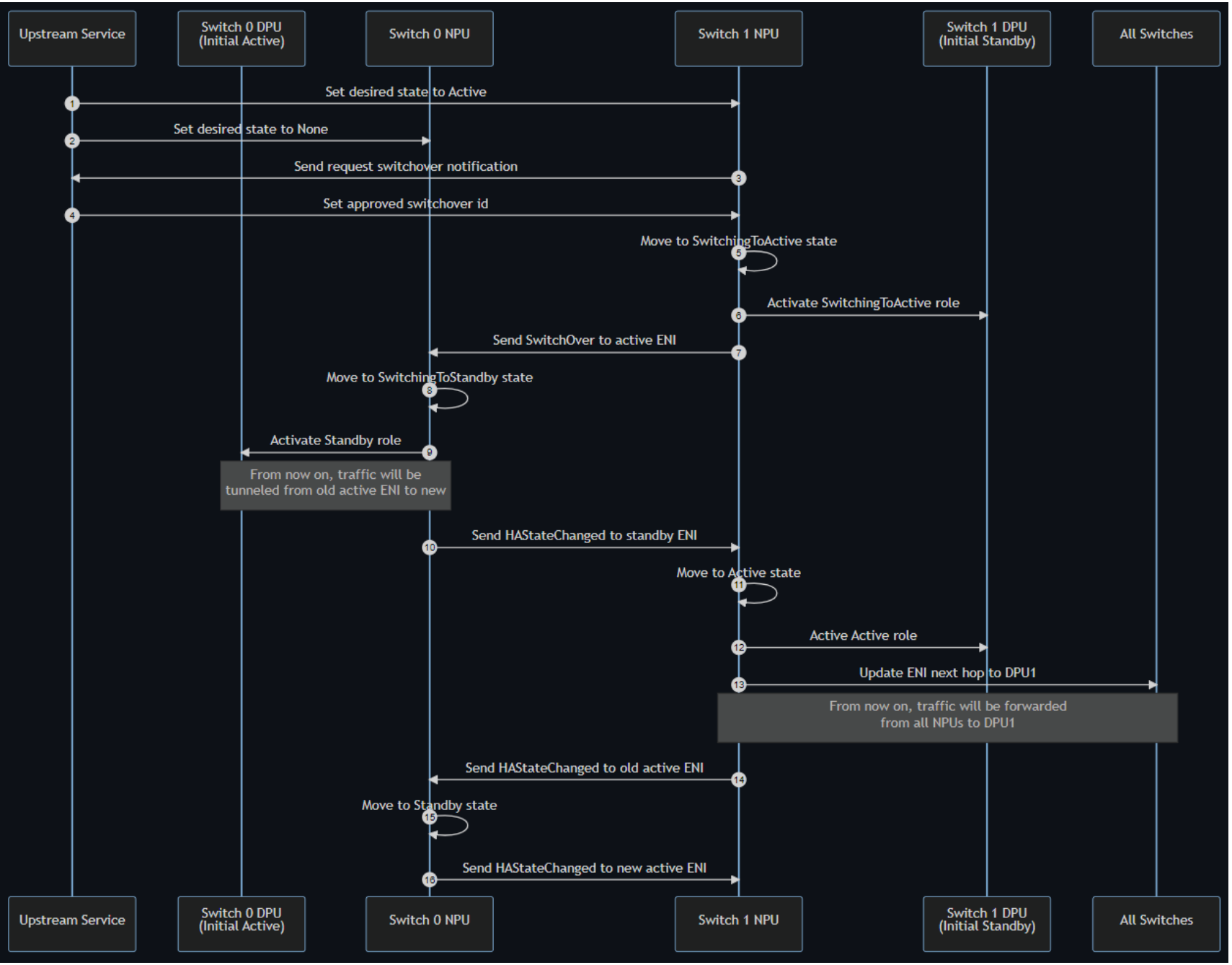




Planned Switchover

- Upstream initiates the operation by setting the desired state.
- SmartSwitch requests upstream for switchover, in which upstream ensures the state is in steady state and policy on both side matches.
- Once approved, SmartSwitch drives the state transition and update the ENI-level traffic control to desired state.
- Switchover starts from a standby node.

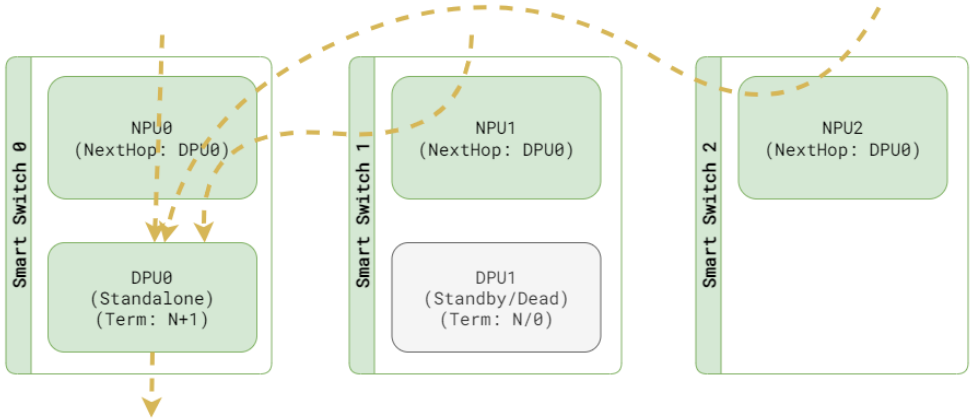




Planned Switchover – Seq. Diagram

Unplanned events vs Standalone setup

- Network interruption will be unavoidable.
- Best Effort: Reduce the chance of dropping packet by stopping flow replication. No solution is going to be guaranteed to be perfect.
- “Standalone-Standby/Dead” pair to avoid 2 deciders / Brain split.
- Card-level standalone setup - All ENIs will failover together. This is due to bulk sync limitation today, but we are going towards ENI level.
- If any unplanned events last long, fire alerts.

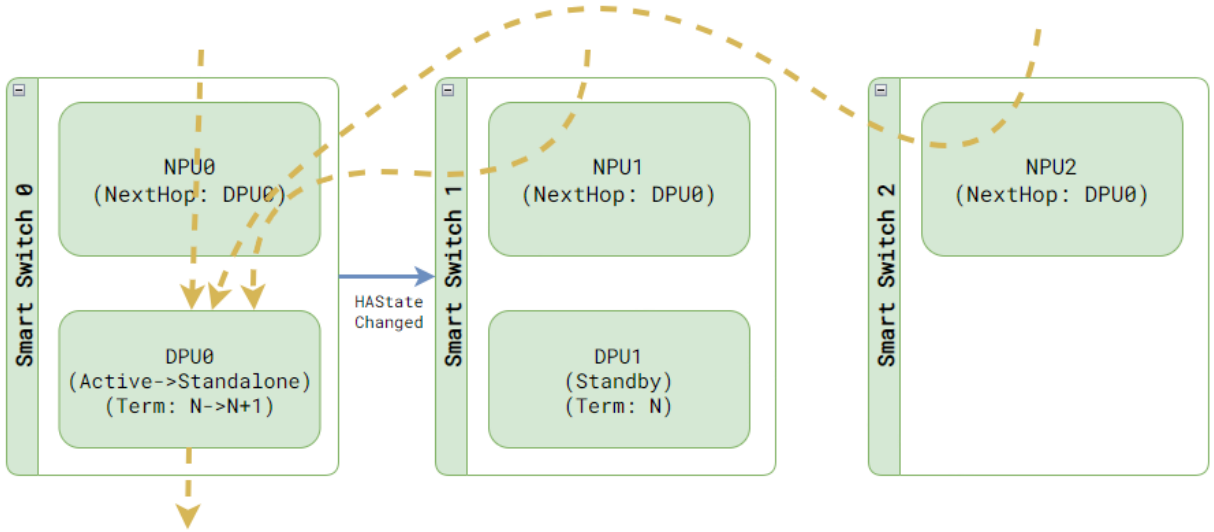


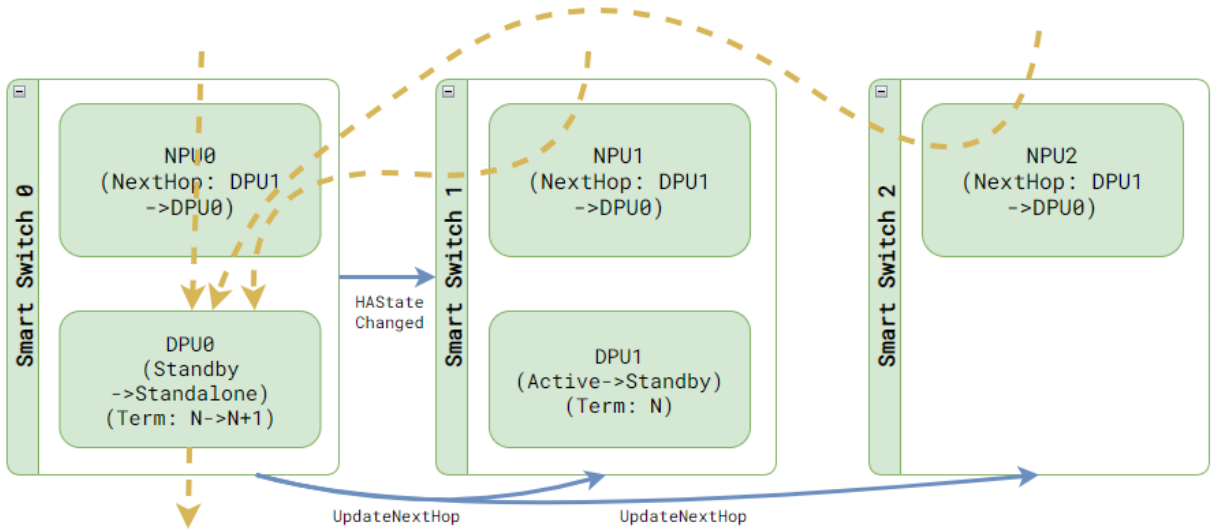
Step 1 - Triggers

Problem	Trigger	Resolve signal
Peer shutdown	Planned shutdown request	HAStateUpdate with Connected state
Peer DPU lost	Peer lost SAI notification	Peer connected SAI notification
Peer DPU dead	HAStateUpdate with dead peer	HAStateUpdate with non-dead peer
High data plane packet drop rate	ENI-level data plane counters	ENI-level data plane counters
Manual Pinned	ENI-level DPU isolation	Isolation removed
Card pinned to standalone	Card pinned to standalone	Pinning removed

Step 2 - Vote

- Merge the signals from ENI level to card level first
- Things to check:
  - Already in standalone state
  - Local DPU health signals (Covers DPU/Switch hardware failures)
  - Send request to pair to start voting
    - Standalone state of the pair
    - Check if any DPU is manual pinned by DRI. If both pinned, cancel vote and alert.
    - Use preferred standalone DPU programmed from upstream (Covers data path failures).
- ENI-level standalone voting is in omitted here. We can find it in full design doc.



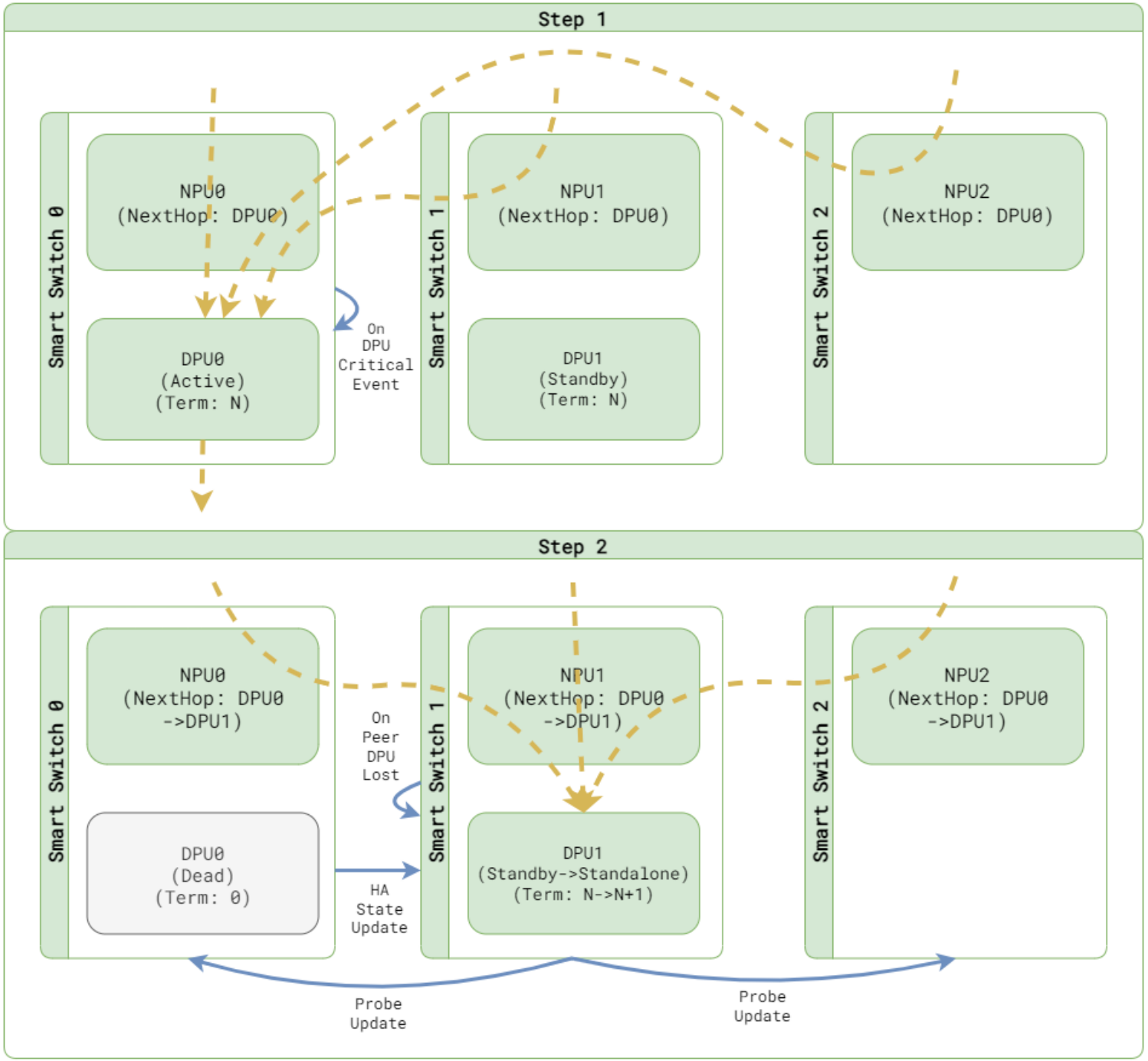


Step 3 – Drive into standalone state

Once vote is done, each ENI's state will be updated and start to drive on its own.

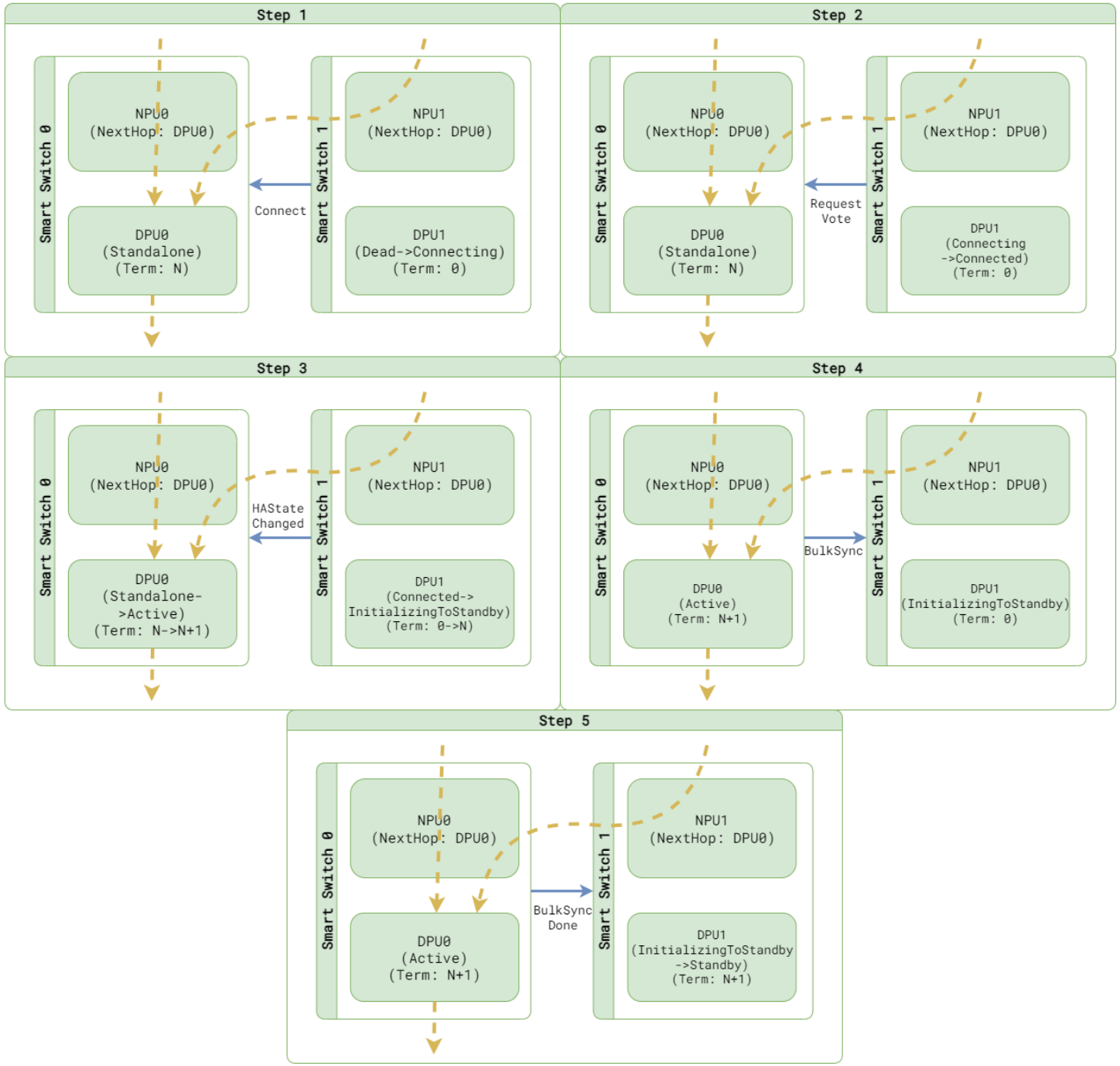
Example – Peer DPU is down

- Using “Peer Lost” SAI notification as trigger.
- Once detected, we will start voting. This will cause the good side to go into standalone state, which stops flow replication.
- Due to card level standalone, other ENIs in standby state will also be moved to standalone state.



Bulk sync is used for bring the node to ready again

Recover from standalone setup



Term Tracking and Primary Election

- Term:
  - Find the DPU that lives the longest and knows the most on the historical flow changes.
  - While a DPU can make flow decision, the term increases whenever the inline sync channel is enabled or disabled.
- Primary election: Whichever with higher term wins.
  - It also need to consider other cases to make the algorithm practical, such as ENI not found, Pinned to standalone, ENI is shutting down.
  - https://github.com/r12f/DASH/blob/user/r12f/ha/documentation/high-avail/smartswitch-ha-hld-proposal.md#83-primary-election

Flow lifetime management

- Active rules all, standby always follows.
  - Inline sync – Connection creation, close or re-simulated.
  - Batched sync – Flow aged out, bulk sync TCP seq number or flow deletion via SAI APIs.
  - Bulk sync – When a node rejoins the HA set.

Bulk sync

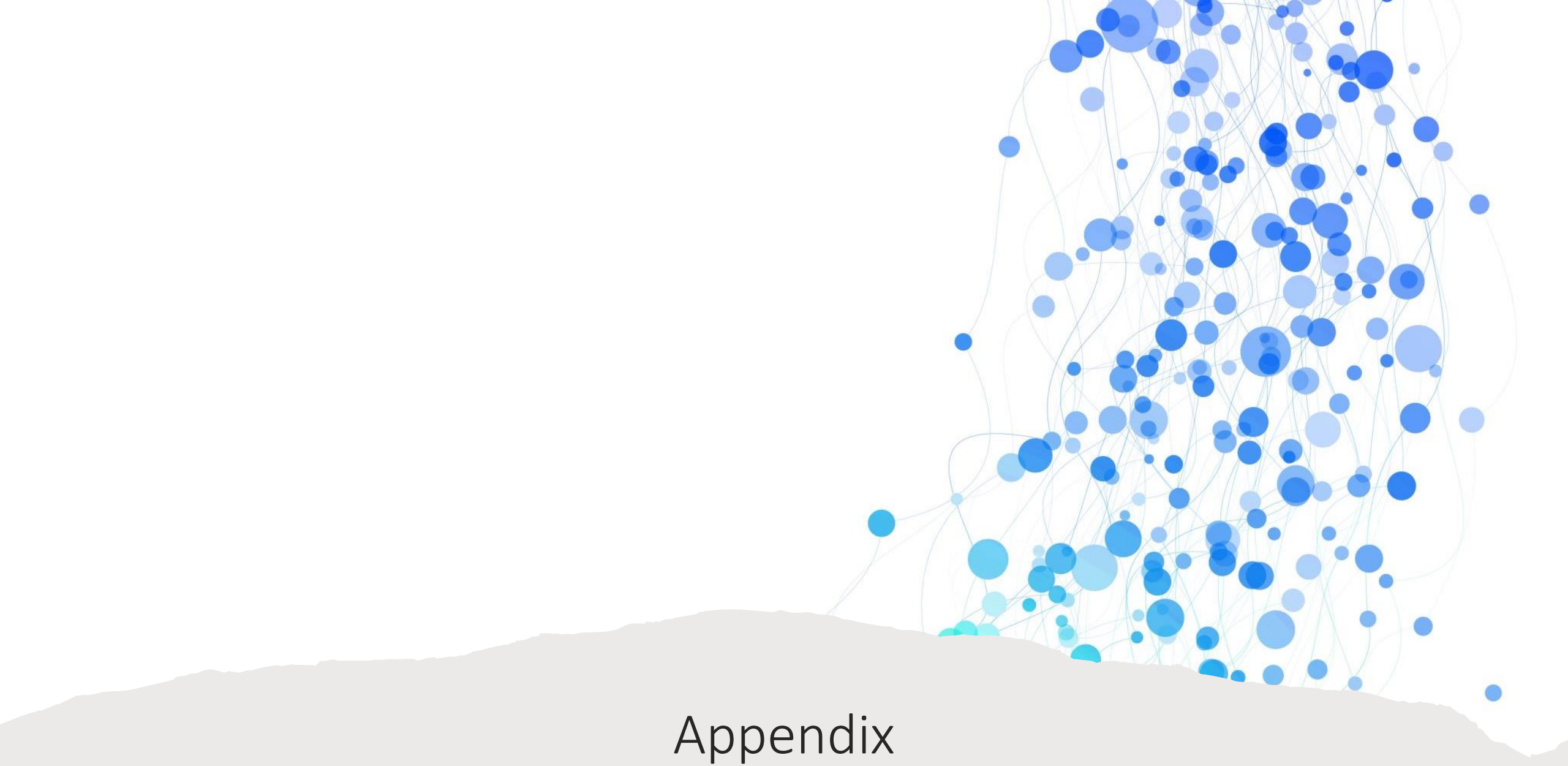
- Perfect sync
  - Whenever a node becomes active, we increase the flow version (color).
  - Sync all flows that has lower flow version than current.
- Range sync (ENI level only)
  - Idea is only sync the flow that is not sync’ed
    - ImpactStart = Whenever inline flow sync channel breaks
    - ImpactEnd = Whenever inline flow sync channel reestablished
  - While a DPU can make flow decision, whenever the inline sync channel is enabled or disabled, we increases the flow version.
  - When inline sync channel is disabled, track all existing flow deletions.
  - Sync only the flows that has the flow version when inline sync channel is disabled.

Having problem talking to upstream?

- Policy will become stale gradually.
- HA will continue to function and handle unplanned events.
- As long as upstream can talk to one side of DPU, we can use pin the reachable side to standalone to make the policy update working again.
- If both sides are down, then policy will not be updated. Everything will freeze as it is. (No planned operations and no policy updates).

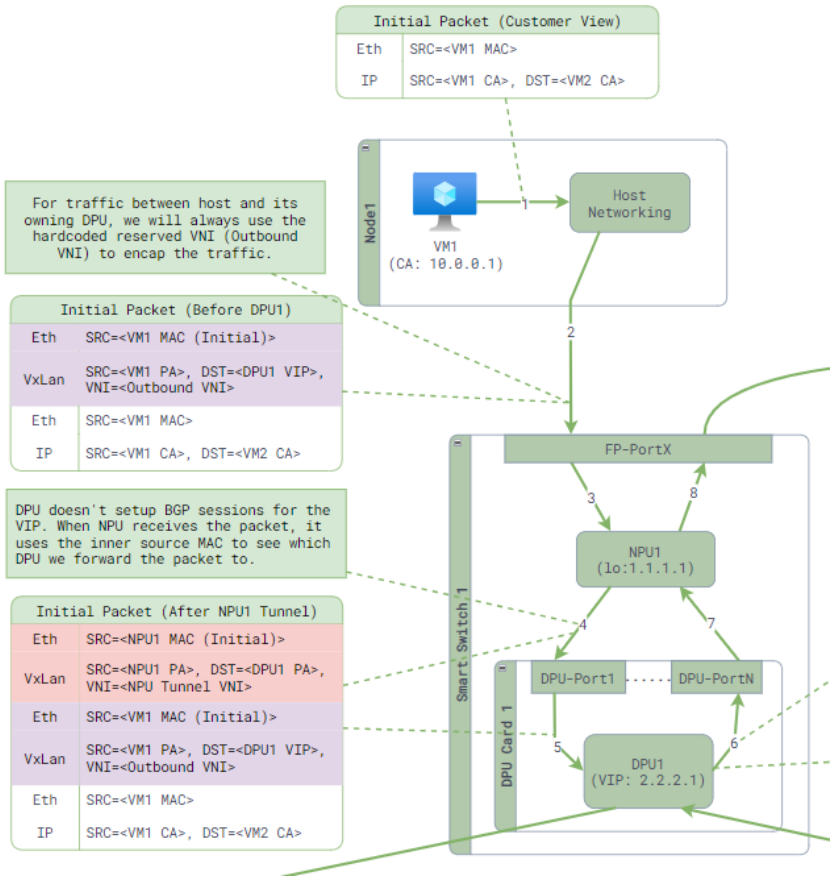
References

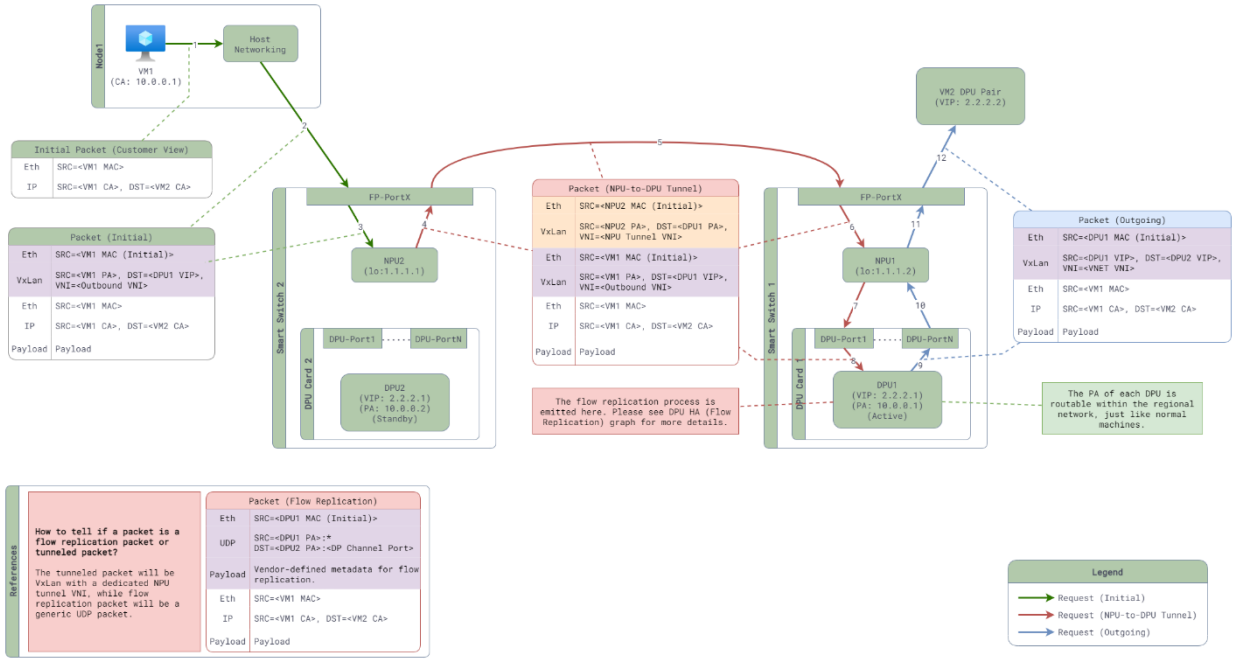
- SmartSwitch HA High Level Design doc: https://github.com/r12f/SONiC/blob/user/r12f/ha/doc/smart-switch/high-availability/smart-switch-ha-hld.md.
- DASH High Level Design doc: https://github.com/sonic-net/DASH/blob/main/documentation/general/dash-high-level-design.md.
- DASH scale design: https://github.com/sonic-net/DASH/blob/main/documentation/general/dash-sonic-hld.md.



Appendix

NPU-to-DPU traffic tunneling





State	Definition	Receive traffic from NPU?	Make decision?	Handling old flow?	Respond flow sync?	Init flow sync?	Init Bulk sync?
Dead	HA participant is just getting created, and not connected yet.	No	Drop	Drop	No	No	No
Connecting	Connecting to its peer.	No	Drop	Drop	No	No	No
Connected	Connected to its peer, but starting primary election to join the HA set.	No	Drop	Drop	No	No	No
InitializingToActive	Connected to pair for the first time, voted to be active.	No	Drop	Drop	No	No	No
InitializingToStandby	Connected to pair for the first time, voted to be standby.	No	Tunneled to pair	Tunneled to pair	Yes	No	No
Destroying	Preparing to be destroyed. Waiting for existing traffic to drain out.	No	Tunneled to pair	Tunneled to pair	No	No	No
Active	Connected to pair and act as decision maker.	Yes	Yes	Yes	No	Yes	Yes
Standby	Connected to pair and act as backup flow store.	No	Tunneled to pair	Tunneled to pair	Yes	No	No
Standalone	Heartbeat to pair is lost. Acting like a standalone setup.	Yes	Yes	Yes	Yes	No	No
SwitchingToActive	Connected and preparing to switch over to active.	No	Yes	Yes	Yes	Yes	No
SwitchingToStandby	Connected, leaving active state and preparing to become standby.	Yes	Tunneled to pair	Tunneled to pair	Yes	No	No