

RDK Gpon Manager

Introduction

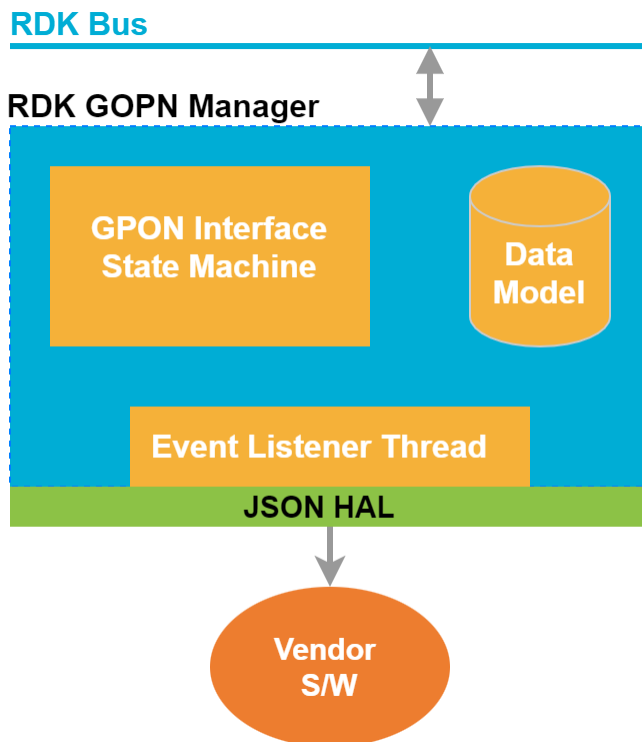
RDK GPON Manager is a component that interfaces directly with the RDK Bus.

It is responsible for:

- Supporting GPON interfaces,
- Storing configuration and statistical data for the ONU,
- Configuring the ONU
- Polling the status of the ONU
- And communicating with other RDK Managers to prepare a GPON interface for use for WAN

Architecture

The following diagram shows how RDK GPON Manager sits in the RDK-B architecture as well as its internal processes:



RDKGPONManager-ModuleDiagram

The RDK GPON Manager component will hold the GPON data model and will run the GPON InterfaceState Machine process.

The GPON Interface State Machine will be used to configure GPON links for WAN. By default, there will initially be 0 instances of the GPON Interface State Machine running on startup, and a new instance will be started for every ONU VEIP interface that is enabled for use.

An Event Listener Thread will be started after initialisation to receive status updates from the ONU.

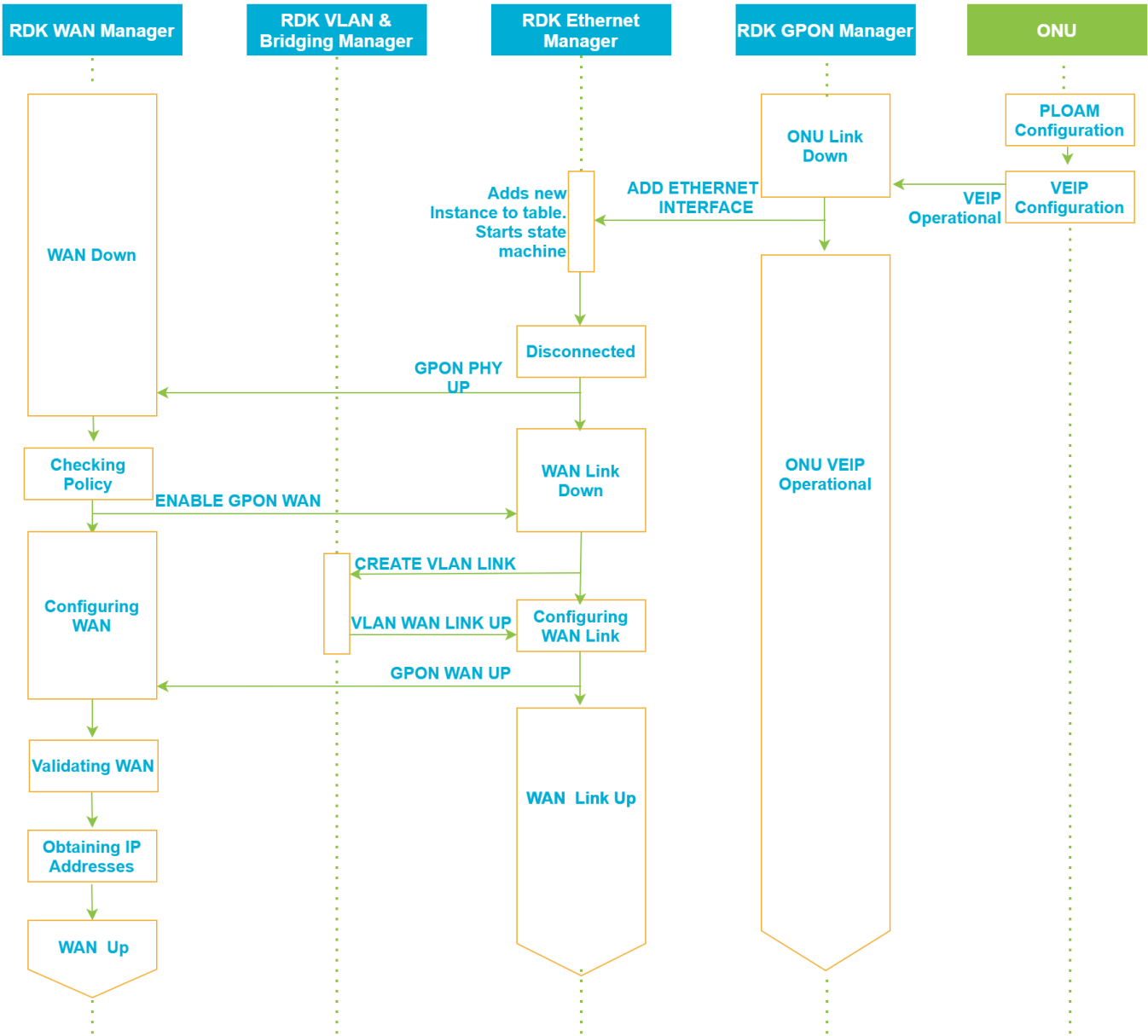
Data models owned by RDK GPON Manager

System Integration

This section describes how RDK GPON Manager integrates with RDK WAN Manager and the rest of the RDK-B system.

GPON Connected

The following diagram shows how RDK GPON Manager will handle setting up a connected GPON interface. Every box represents a state that the corresponding Manager or Module is in, while every arrow represents either a transition between states, or event messages between Managers and Modules.



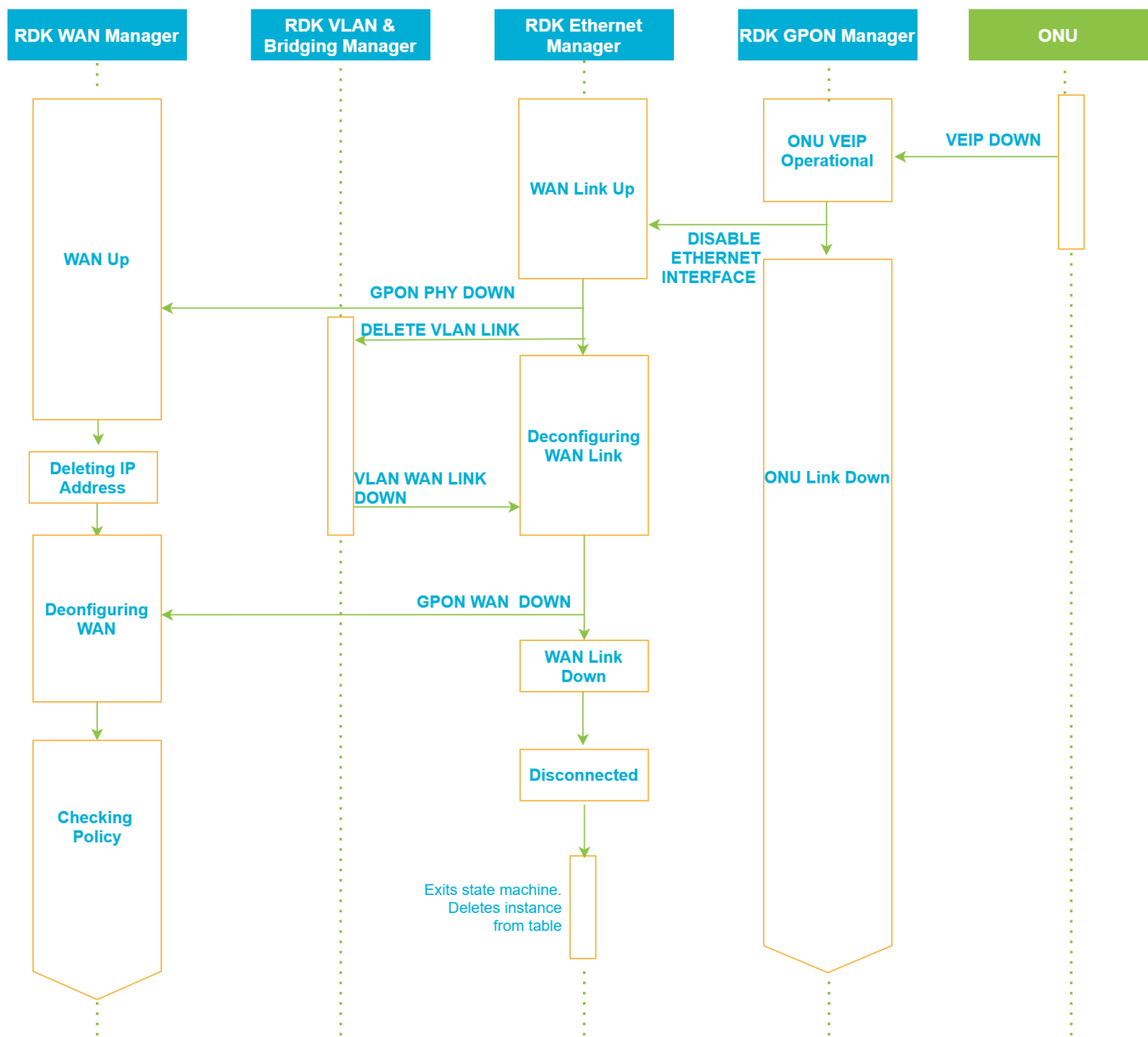
Process-SettingUpGPON

Since the OLT controls the PLOAM and OMCI configuration, the ONU will be expected to configure the GPON connection outside of RDK GPON Manager's control.

RDK GPON Manager will simply monitor the ONU's Registration State as well as the status of the VEIP interface, which is the interface presented to the Gateway. Once the VEIP interface is ready for use, RDK GPON Manager will communicate with RDK Ethernet Manager to configure that VEIP interface for WAN, in a similar way to how a physical Ethernet interface would be used for WANoE. This allows the WAN control mechanism to be generic across all Ethernet interfaces, physical or virtual, and simplifies RDK GPONManager.

GPON Disconnected

The following diagram shows how RDK GPON Manager will deconfigure the GPON WAN link if the ONU's VEIP interface goes down:

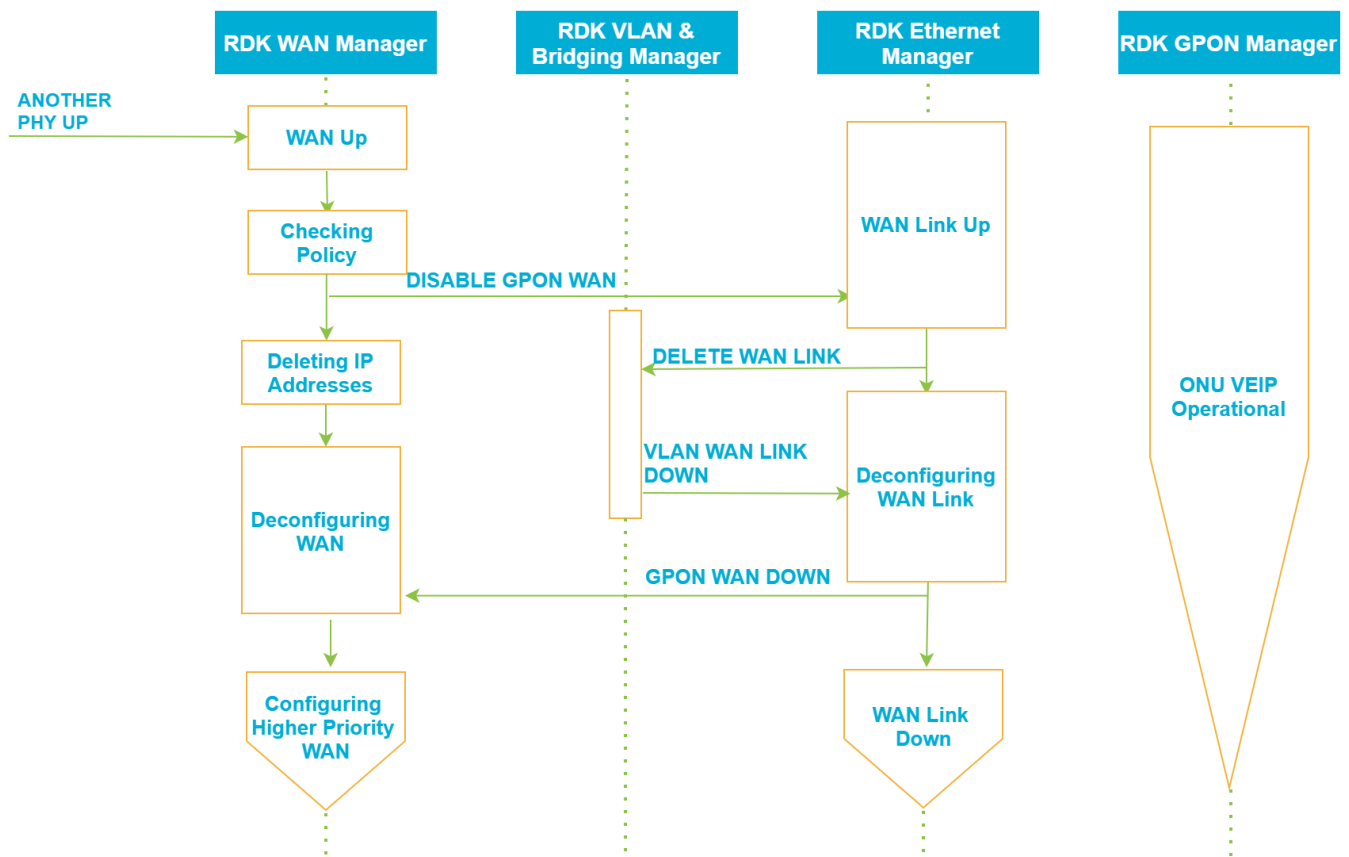


GPONInterfaceDisconnected

In the above scenario, if the ONU's VEIP interface goes down, we must also tear down the Gateway's WANoE connection. RDK GPON Manager will disable the instance of the Device.Ethernet.Interface data model it created in RDK Ethernet Manager, in order to force it to tear down the WAN link. This process is similar to what would happen if a physical Ethernet interface were disconnected, however in this scenario the data object for the VEIP interface will be deleted from the Gateway and the state machine in RDK Ethernet Manager will completely exit.

GPON Disabled for WAN

The following diagram shows how RDK Ethernet Manager will deconfigure the WAN link if it is disabled by RDK WAN Manager:



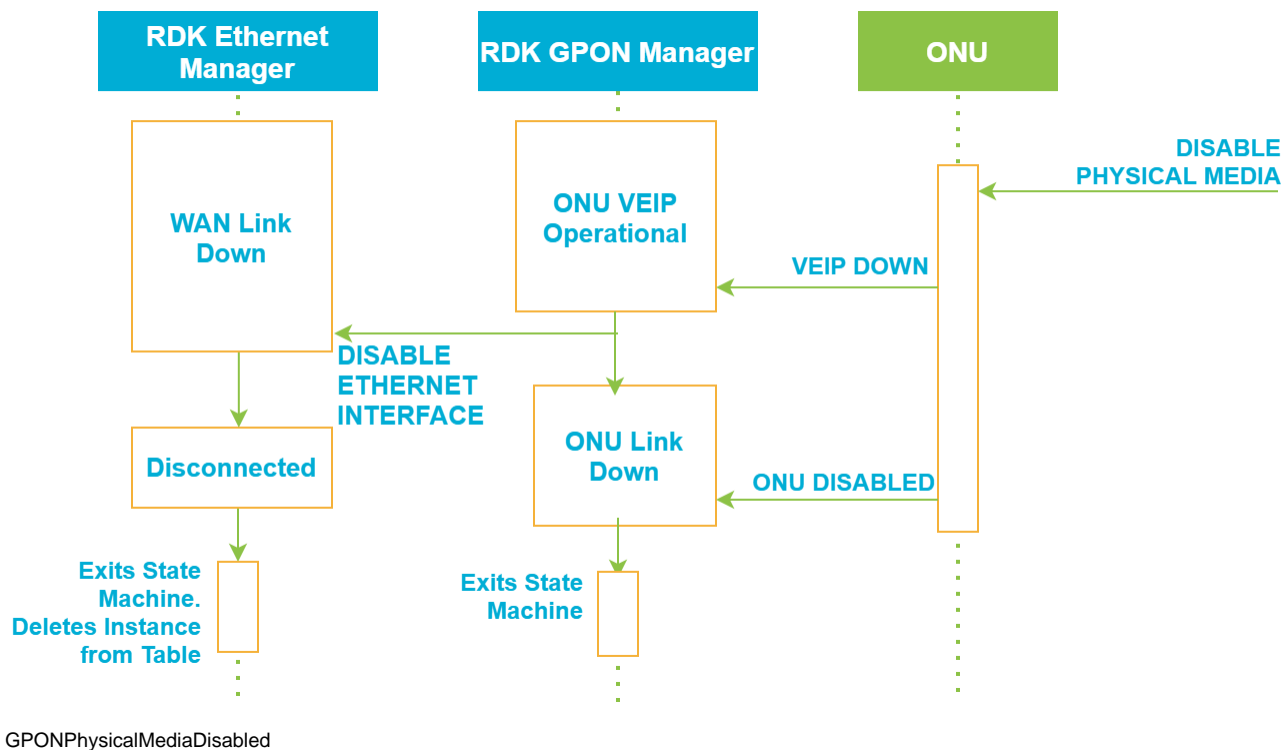
GPONWANDisabled

In this scenario, RDK Ethernet Manager will deconfigure the WAN link and will enter a "WAN Link Down" state, where it will remain while the ONU's VEIP interface is operational. If RDK WAN Manager re-enables the interface for WAN while the ONU's VEIP interface is still operational, then RDK Ethernet Manager will once again progress through the states to configure the WANpoE.

In this scenario, RDK GPON Manager remains in its "ONU VEIP Operational" state. This is because as far as the ONU is aware, its connection to the OLT is still operational, and it is only the WAN VLAN link that would have been configured on top of the VEIP interface (e.g. "veip0.1" or "erouter0") that has been disabled.

ONU Disabled

The following diagram shows how the ONU's physical media can be disabled outright by an external component:



In this scenario, the ONU is expected to tear down its VEIP interface, which will trigger RDK GPON Manager to inform RDK Ethernet Manager to tear down the WANOE connection. Above diagram shows RDK Ethernet Manager receiving the "Disable Ethernet Interface" message while it is already in the "WAN Link Down" state, but it could in fact receive this message at any time. When RDK Ethernet Manager receives this "Disable Ethernet Interface" message, it will exit the state machine and delete the created instance from its *Device.Ethernet.Interface* data object. If the ONU is ever disabled, the state machine in RDK GPON Manager will also exit so as not to consume resources while it is not required.

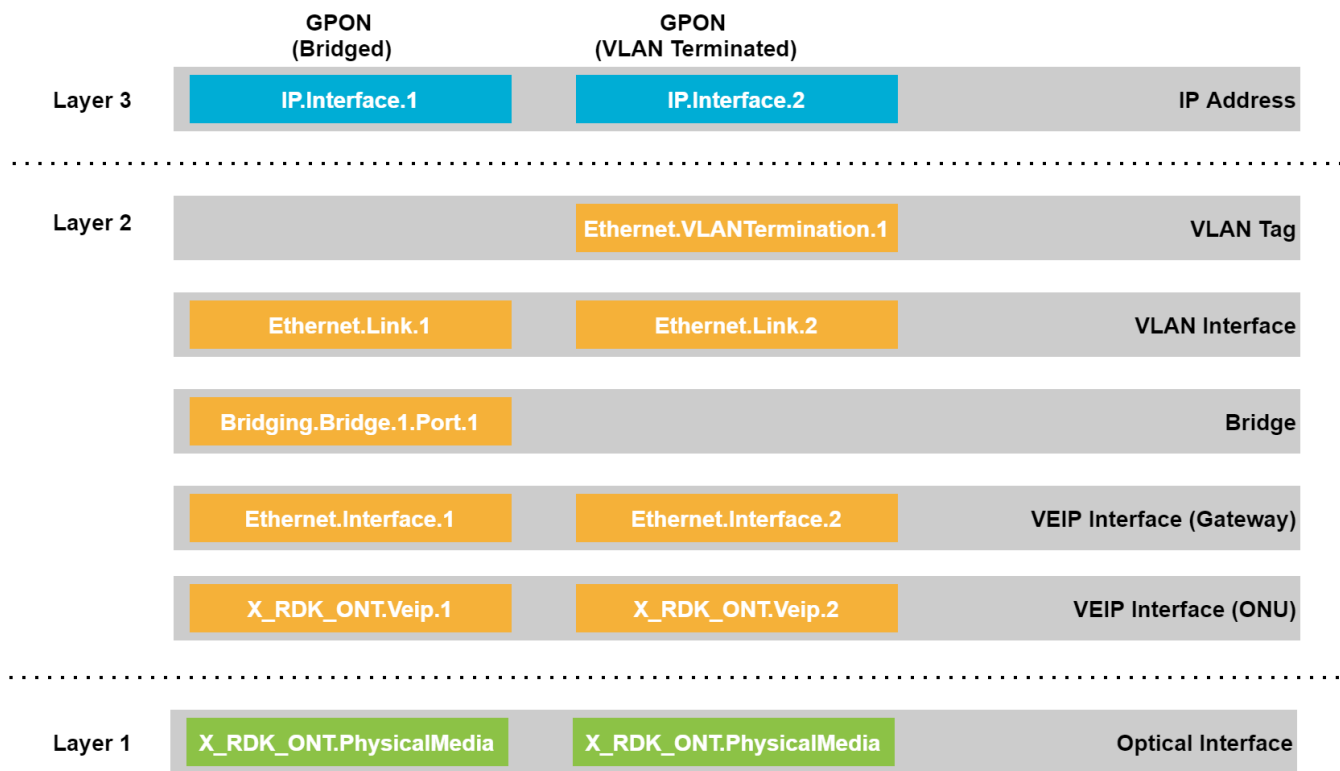
Data Model

RDK GPON Manager will own a custom data model to store data related to modelling an ONU. The following diagram shows the data model objects that will be owned by RDK GPON Manager:



TR-181dataobjectsstoredintheRDKGponManagerdatamodel

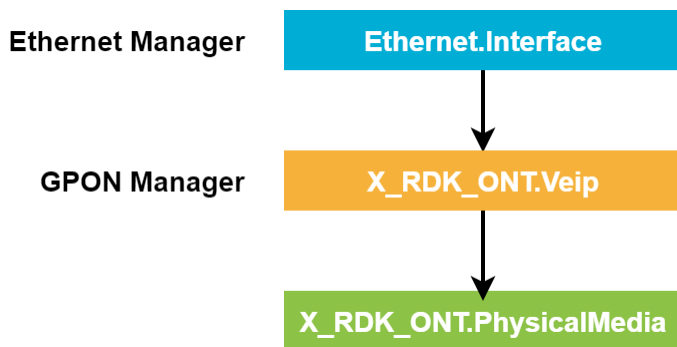
The following diagram shows how the data models across all RDK Managers will stack on top of each other using their *LowerLayers* parameters in order to configure and use GPON for WAN:



Interface Stack for GPON

Since RDK GPON Manager owns only the *Device.X_RDK_ONT* data model, it will have to communicate with other Managers to set up and configure the WAN link.

RDK GPON Manager will be responsible for populating its own *Device.X_RDK_ONT* data model, as well as creating/deleting instances of *Device.Ethernet.Interface* in order to inform RDK Ethernet Manager to use the VEIP interface as an ordinary Ethernet interface, and to configure it for WAN. This relationship is shown in the following diagram:



Data objects to be populated by the RDKGPONManager

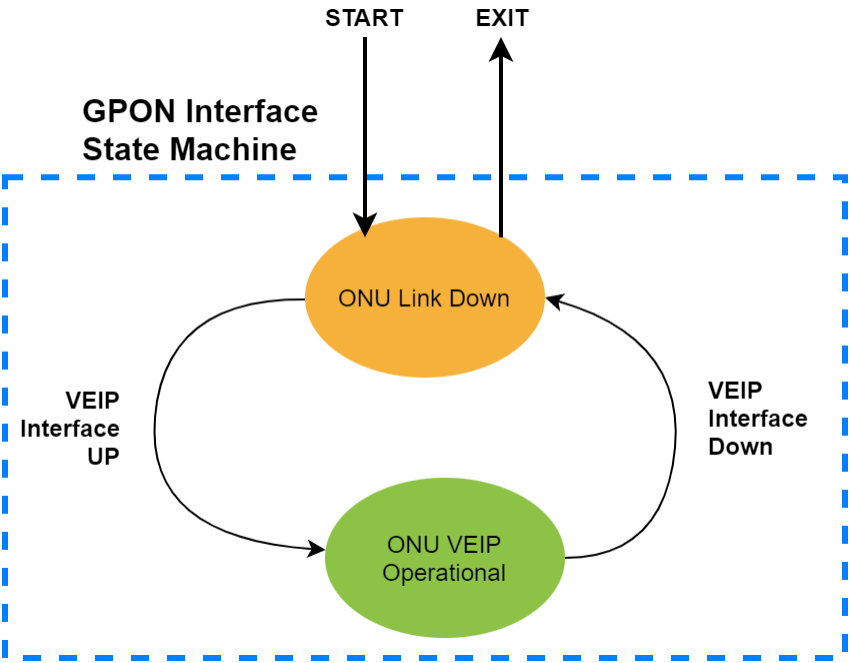
GPON Interface State Machine

The GPON Interface State Machine will run as a threaded process in the RDK GPON Manager. It will be responsible for:

1. Checking the status of the VEIP interface,
2. Triggering the configuration process to create a virtual ethernet interface on the Gateway,
3. And triggering the deconfiguration process for a created virtual ethernet interface.

Multiple instances of GPON Interface State Machine can be started in parallel, with each instance operating on a separate ONU VEIP interface. This allows multiple interfaces to be in different states at any given time while reducing code complexity. This also prevents code duplication by ensuring the same code is deployed in multiple instances.

By utilising a state machine instead of event-driven mechanisms, we can ensure that the correct actions are taken at the right time and in a specified order. When RDK GPON Manager receives an event, it will update the data model, which the state machine will then check to determine the next course of action. This ensures that multiple events only trigger a single action, for if the state machine has entered the correct state, it no longer needs to process a duplicate incoming event. This prevents race conditions and ensures the GPON link is always in the correct state.



GPONInterfaceStateMachine-StateDiagram