

RDK WAN Manager

Introduction

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

It is responsible for:

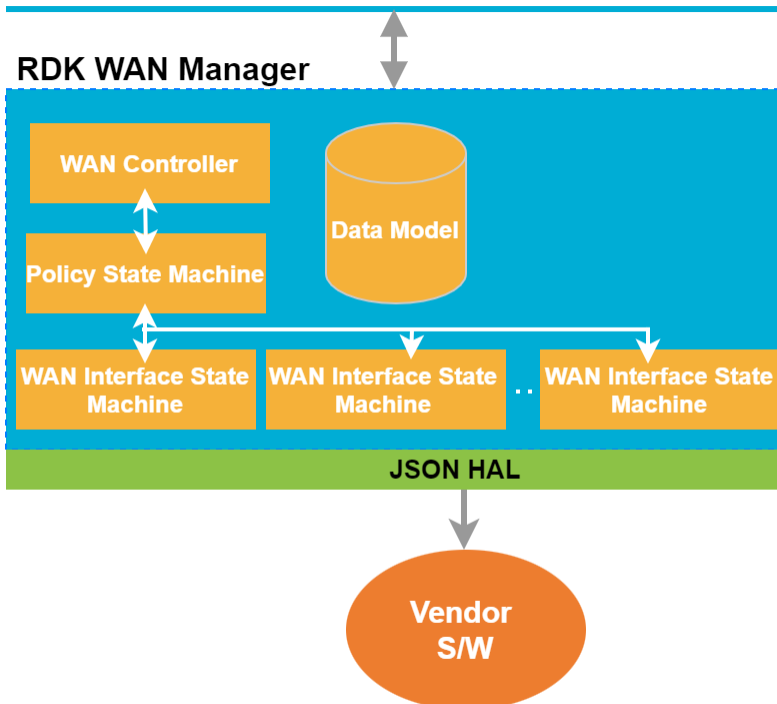
- Storing the WAN configuration of the CPE (e.g. which physical interfaces can be used for WAN),
- Determining which physical interfaces should be used for WAN during runtime
- Triggering Layer 2 configuration for chosen WAN interfaces,
- Triggering Layer 3 protocols (e.g. DHCP, PPP) to establish IP addresses for configured WAN interfaces.

Architecture

The RDK WAN Manager component consists of 3 separate modules:

- An Event Handler for the RDK Bus
- A Data Model,
- Policy State Machines (only 1 will be running at any time),
- WAN Interface State Machines (with 0 or more instances running in parallel).

RDK Bus



The WAN Controller process will handle non-Interface specific activities, such as initialising data models from persistent storage and handling data model changes during runtime.

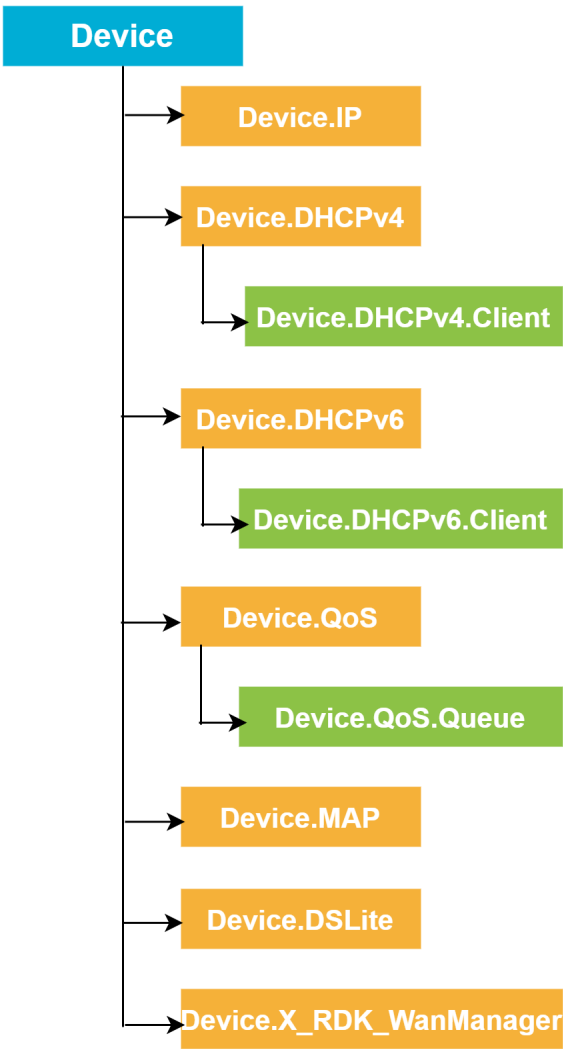
The WAN Controller will also spawn one Policy State Machine based on the WAN Policy configured in the data models. The appropriate State Machine will be launched for the configured Policy in order to manage WAN interfaces.

The Policy State Machine will determine which physical interfaces to use for WAN, and will run instances of the WAN Interface State Machine in order to configure WAN links and obtain IP addresses for each WAN interface. By default, there will initially be 0 instances of the WAN Interface State Machine running on startup, and a new instance will be started for every physical interface being used for WAN.

Data Model

RDK WAN Manager will own standard Layer 3 TR-181 data objects (with additional custom RDK parameters),as well as a custom data model to store data related to configuring WAN interfaces.

The following diagram shows the data model objects that will be owned by RDKWANManager:



The following diagram shows the TR-181 data objects that will need to be populated byRDKWANManager and how they should be stacked on top of other data objects owned by other RDK Managers using the *LowerLayers* parameter:

