Multi LAN subsystem of RDK-B stack

☒ **Draft**
☐ Baselined for Estimates

Author	
Authorized by	
Version	0.3
Date	05-09-2014

Table of Contents

List of Tables

1. INTRODUCTION

This document covers the architecture details of Multi LAN subsystem which is part of the RDK-B code base used for XB3. Description includes the ARM specific implementation details followed in the architecture.

1.1 Document Scope and Objectives

Scope of this document is to explain the Multi LAN subsystem of RDK-B stack.

1.2 Revision History

This revision history only documents versions that are distributed to multiple team members.

Version	Release Date	Description
0.1		Initial draft version
0.2	05-02-2014	Added LAN initialization
0.3	05-09-2014	Added VLAN and GRE initialization.

Table 1 Revision History

1.3 References

#	Name	Owner/Author	Version	Date
1				
2				

Table 2 Reference Documents

1.4 Legend

#	Notation	Description
1		
2		

Table 3 Legend

1.5 Stakeholder and Review Approval / Sign off

Group	Name	Role

Table 4 Stakeholder List

1.6 Definitions and Acronyms

ACRONYMS

Acronym	Description

Table 5 Acronyms

DEFINITIONS

Term	Description

Table 6 Definitions

1.7 Assumptions and Risks

1.8 Dependencies on other projects / features

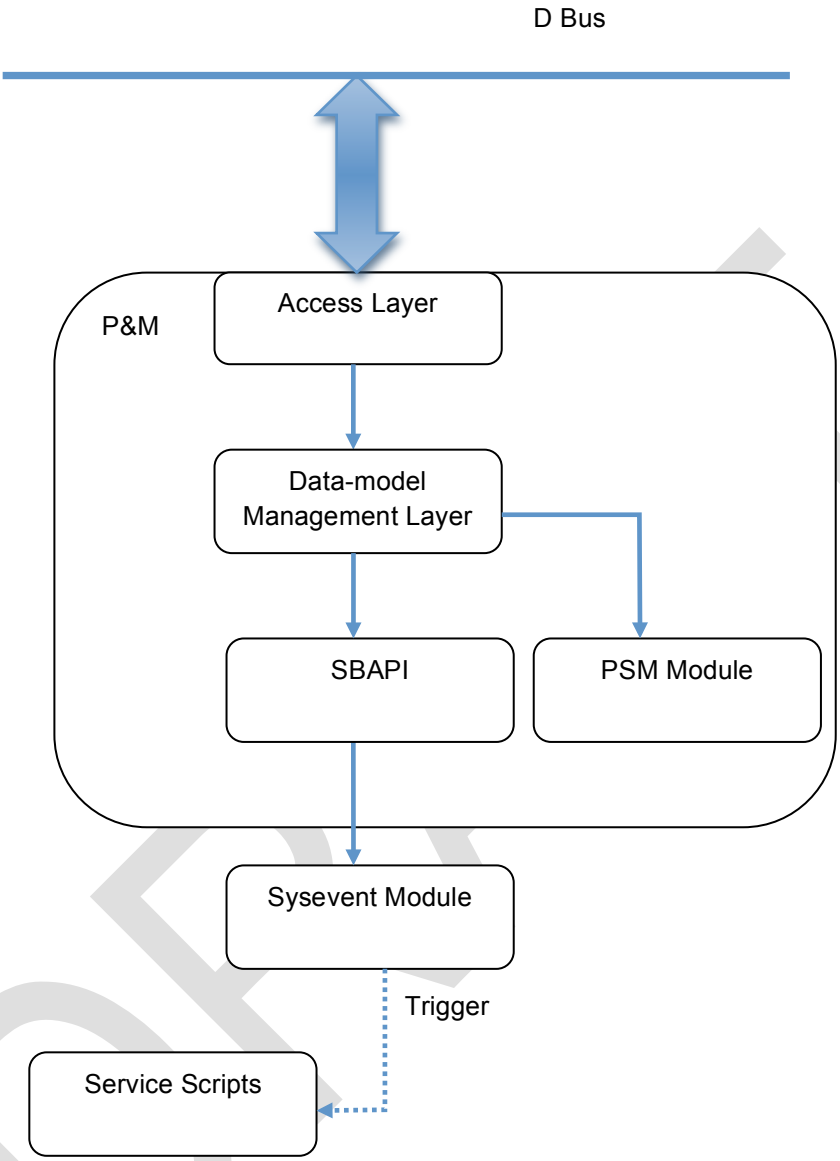
2. Architecture of Multi LAN subsystem

Multi LAN subsystem is part of Provisioning and Management (P&M) and Wifi Provisioning and Management components.
This sub-system supports all the parameters, which are defined in the TR-181 data-model.

Figure 1 shows the architecture diagram of P&M module in the gateway stack. Here, Access Layer takes care of calls to/from D-Bus. It provides all API's for get/set parameters based on request from various other modules in the gateway.

Data-model Management Layer (DML) loads all data model access APIs through a pre-defined XML file TR181-CISCO_MODEL-CISCO_CUSTOMER.XML (Currently using dpc3939b and Comcast as model and customer name respectively). Complete data-model implementation is a part of shared library called libtr181.so.

After loading the DML at initialization, it interacts with the PSM module for any configuration changes, and then eventually Service Scripts takes action accordingly.



1. Architectural view of Multi LAN sub-system

3. Initialization sequence of Multi LAN

Multi LAN initialization is carried out by different layers, which are part of P&M module of the home gateway. There is separate initialization API calls for each layer as shown in below diagram (Figure 3).

During gateway boot-up, the XML file which has all information of the objects, parameters and the APIs will be registered to the common component library. In case of Multi LAN the XML file used to populate the object related information is TR181-dpc3939-comcast.XML. Once the component specific library is loaded, the APIs are registered and then the component specific initialization happens.

During the process of component specific data-model initialization P&M gets the component specific plugin initialization command from CCSP data-model library, corresponding back end manager is called to initialize the Multi LAN subsystem.

Multiple elements are involved in Multi-LAN configuration IP IF, bridges, Ethernet Port, Wi-Fi SSID's, MoCA IF, VLAN,DHCP pool etc. So, Initialization of Multi-LAN involves init for each component.

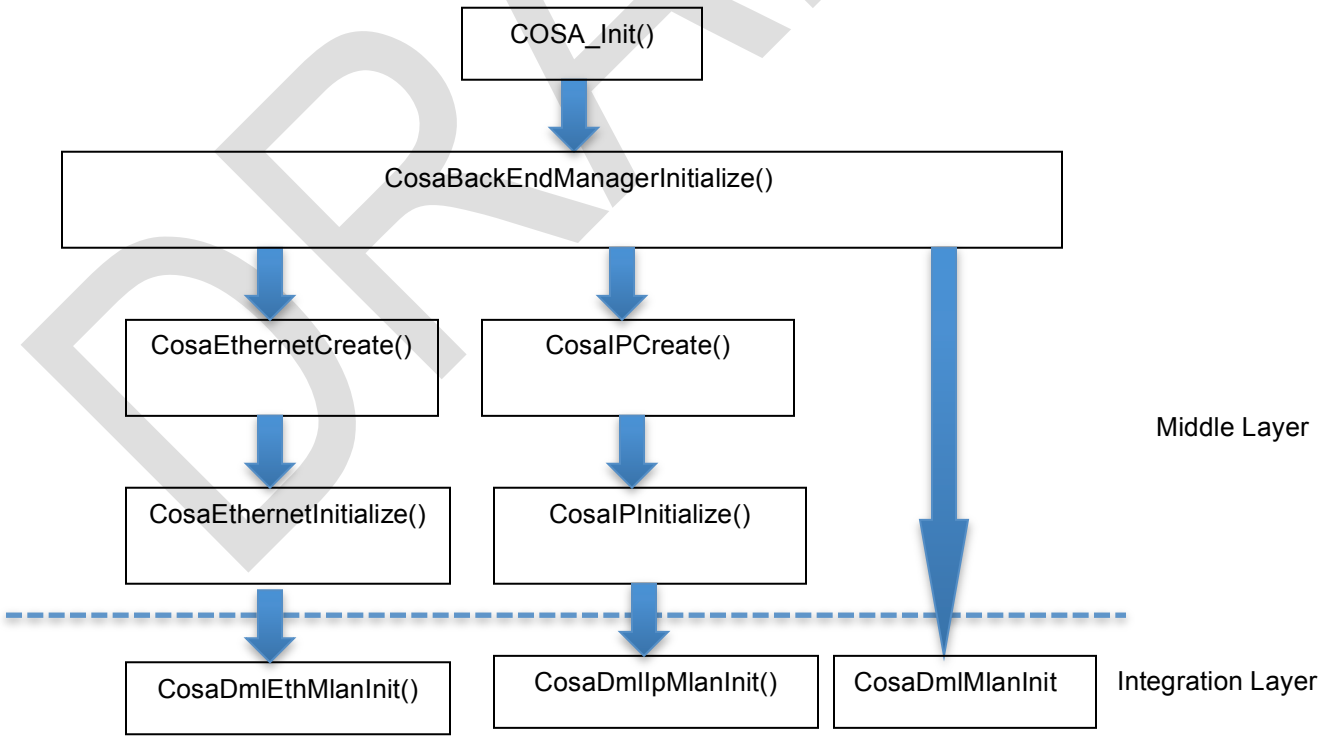


Figure 2: Multi LAN initialization

4. Code flow for Get/Set parameter

During initialization phase of gateway, all the necessary APIs related to an object mentioned in the data-model file will be registered with the data-model management system. So if a query comes for a particular parameter of an object, the registered API for that particular parameter type will be called so that the information can be retrieved from the P&M module.

Once the query comes from UI/SNMP/TR69, the PA accesses corresponding D-BUS API. Later the object is identified from the registry and the corresponding API call specific to the parameter being queried is accessed. API to be accessed is determined by the data-model XML which is used for the library. Then the parameter retrieval process end up in retrieving the value for the parameter from a centralized database maintained by P&M sub-system as shown below (Figure 4).

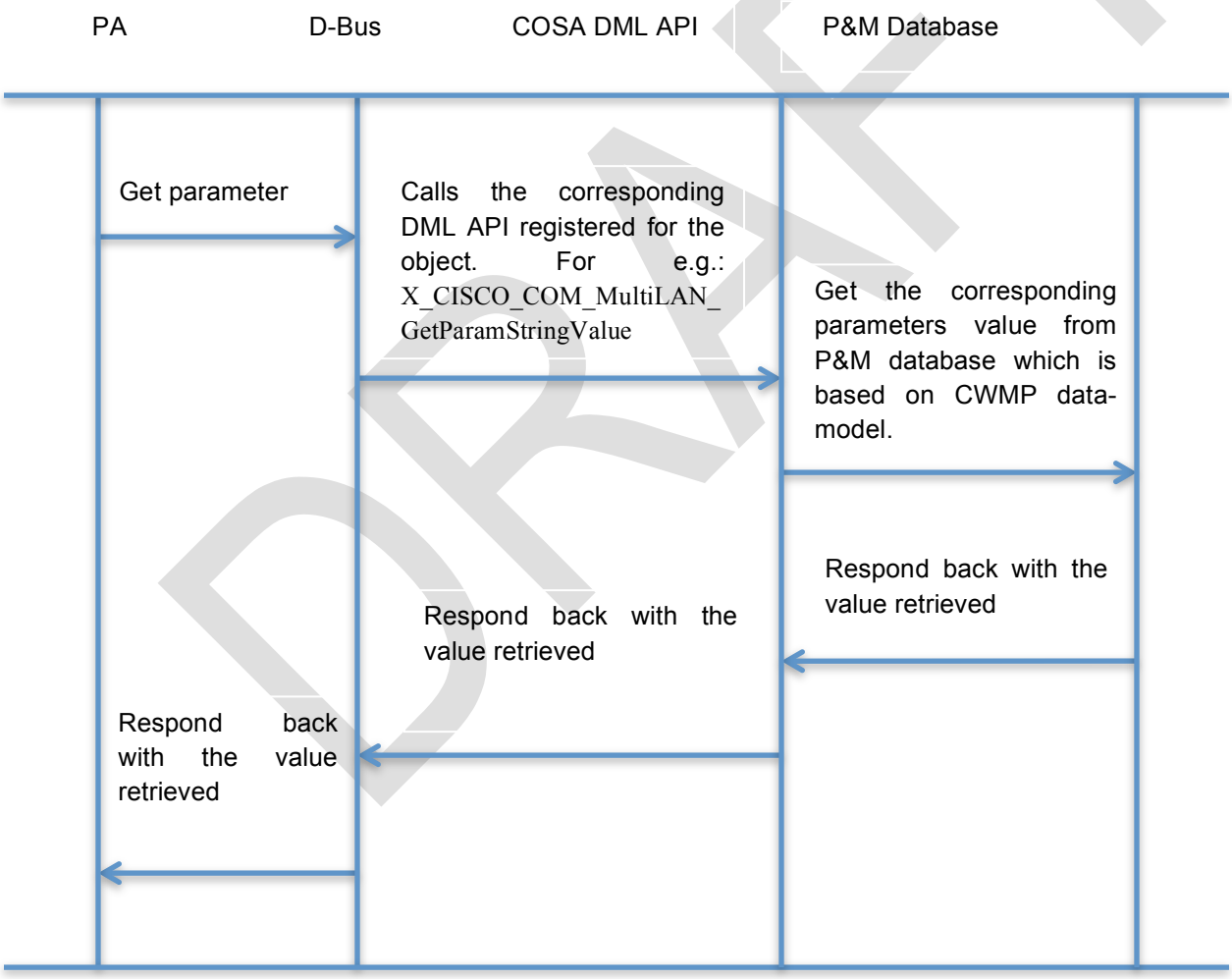


Figure 3: Get API implementation for Multi LAN subsystem

Similarly, In case of set function call, P&M subsystem is updated first with the modified values inside SBAPI layer and a system command is executed which interact with sysevent daemon to make the required configuration changes with the values already saved in P&M subsystem as shown below (Figure 5).

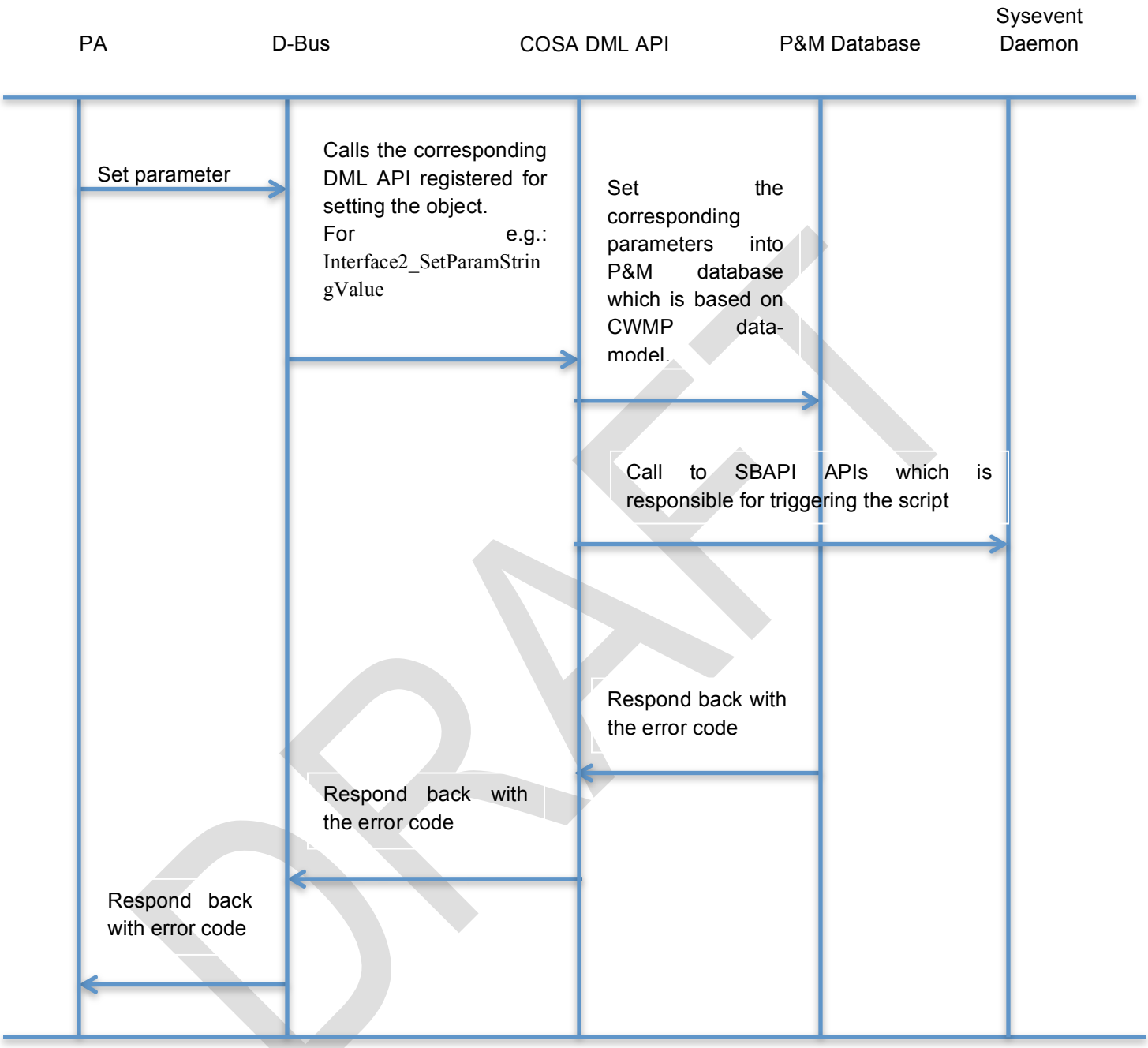


Figure 4: Set API implementation for Multi LAN subsystem

5. SOC dependencies in RDK-B implementation

All set function calls for any parameter reaches SBAPI layer. Code for this layer is available as part of the RDK-B stack at ccsp/CcspPandM/source/TR-181/integration_src.shared/.

SBAPI layer interacts with sysevent daemon (syseventd) using its client (sysevent) for any configuration changes. Sources for sysevent module is present at ccsp/Utopia/source/sysevent/

For e.g. – If doing any configuration changes in DHCP config file and need to start the DHCP server then below system command will be executed which will talk to the syseventd daemon running on the ARM side.

```
sysevent set dhcp_server-restart
```

Once the above command is executed, syseventd daemon performs the required operation to restart the dhcp server. But again that operation can be performed in either of these ways like executing Linux command, script or separate dedicated binary to perform that specific operation. Clearly with this implementation, the parts of RDK-B stack which is actually doing interaction with SOC already been abstracted and created as a part of sysevent daemon and its associated scripts.

6. LAN Configuration

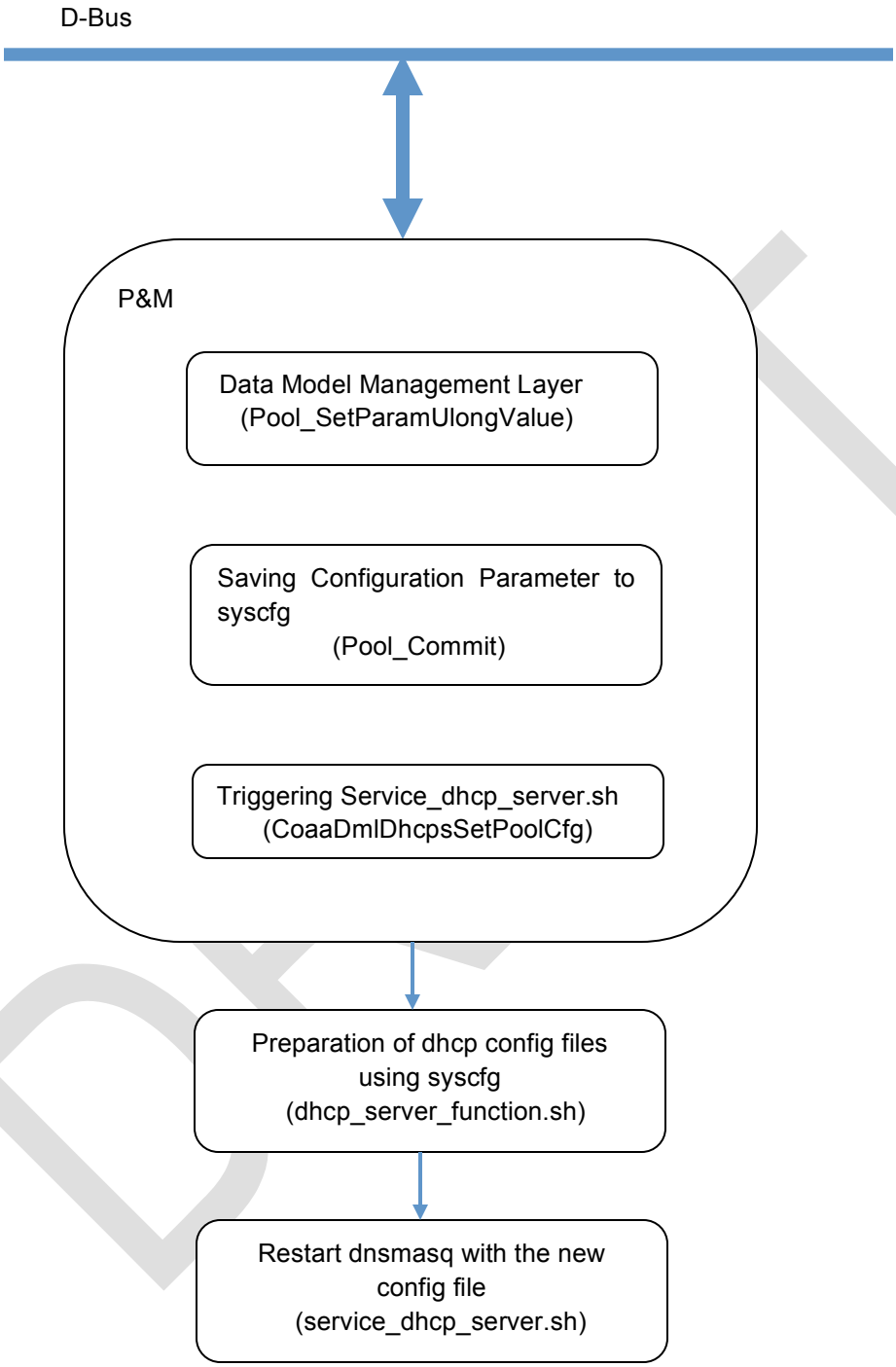
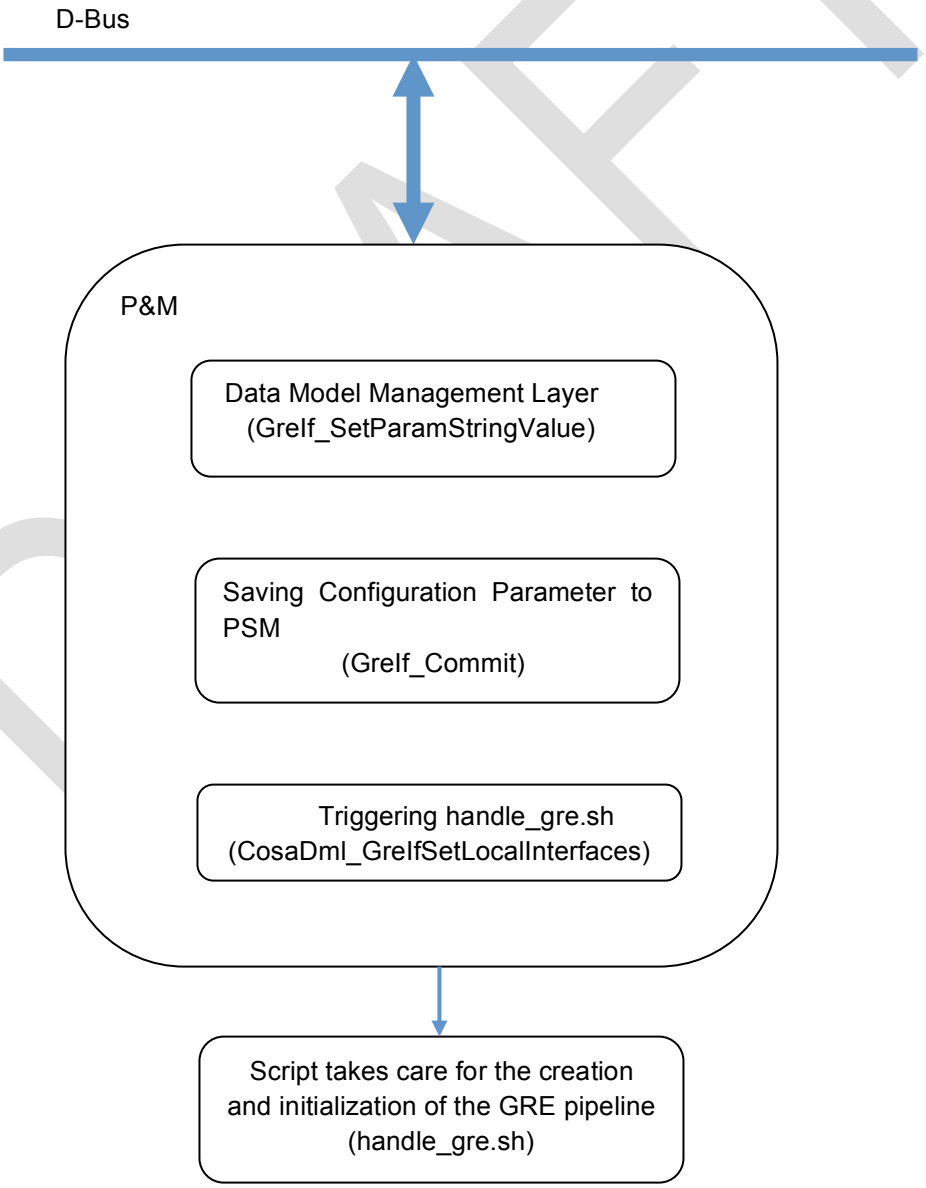


Figure 5: LAN Configuration Flow

- Whenever get/set happens for local ip configuration parameters from the UI, it prepares a message and put it on D-bus.

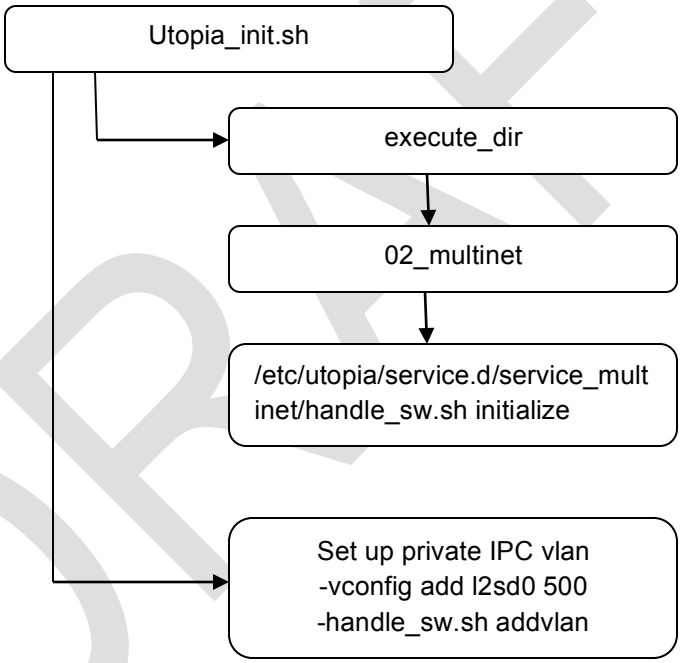
- CcspPandM component receives the message, parse data model xml (TR181-dpc3939b-comcast.XML) and call APIs setting configuration values for DHCP.
- Once set happens for the requested fields, commit method gets called which save all parameters into syscfg.db and trigger an event (sysevent set dhcp_server-restart) to start the dhcp service script.
- It prepare the configuration file (Inside prepare_dhcp_conf function in dhcp_server_function.sh file) using syscfg.
- Dnsmasq is the service which can be used as DNS and DHCP service independently. So, dnsmasq is started with the new modified configuration file. Complete flow is shown in the above figure (Figure 6) with corresponding APIs call.
- Once the service get started, can see the new configuration comes into effect

7. GRE Provisioning



- GRE (Generic Routing Encapsulation) tunneling provides a mechanism to transport packets of one protocol within another protocol.
- The tunnels behave as virtual point-to-point links that have two endpoints identified by the tunnel source and tunnel destination addresses at each endpoint.
- Whenever get/set happens, CcspPandM component receives the message using D-bus, parse data model xml (TR181-dpc3939b-comcast.XML) and call APIs setting configuration values for GRE.
- Once commit method gets called, it triggers the handle_gre.sh which does the creation and initialization for the GRE tunnel.
- Configuration values also get saved in PSM module.

8. VLAN Provisioning



- 02_multinet executes handle.sh script with initialize option which enables vlan.
- vconfig adds the vlan.
- handle_sw.sh gets called with addvlan option which adds the virtual interface and configures it.
- Dependencies : libswctl.so. Command to enable VLAN comes from this library during the registration phase.