WebPA Client Support on RPI and Emulator

Parodus-WebPA :

WebPA :

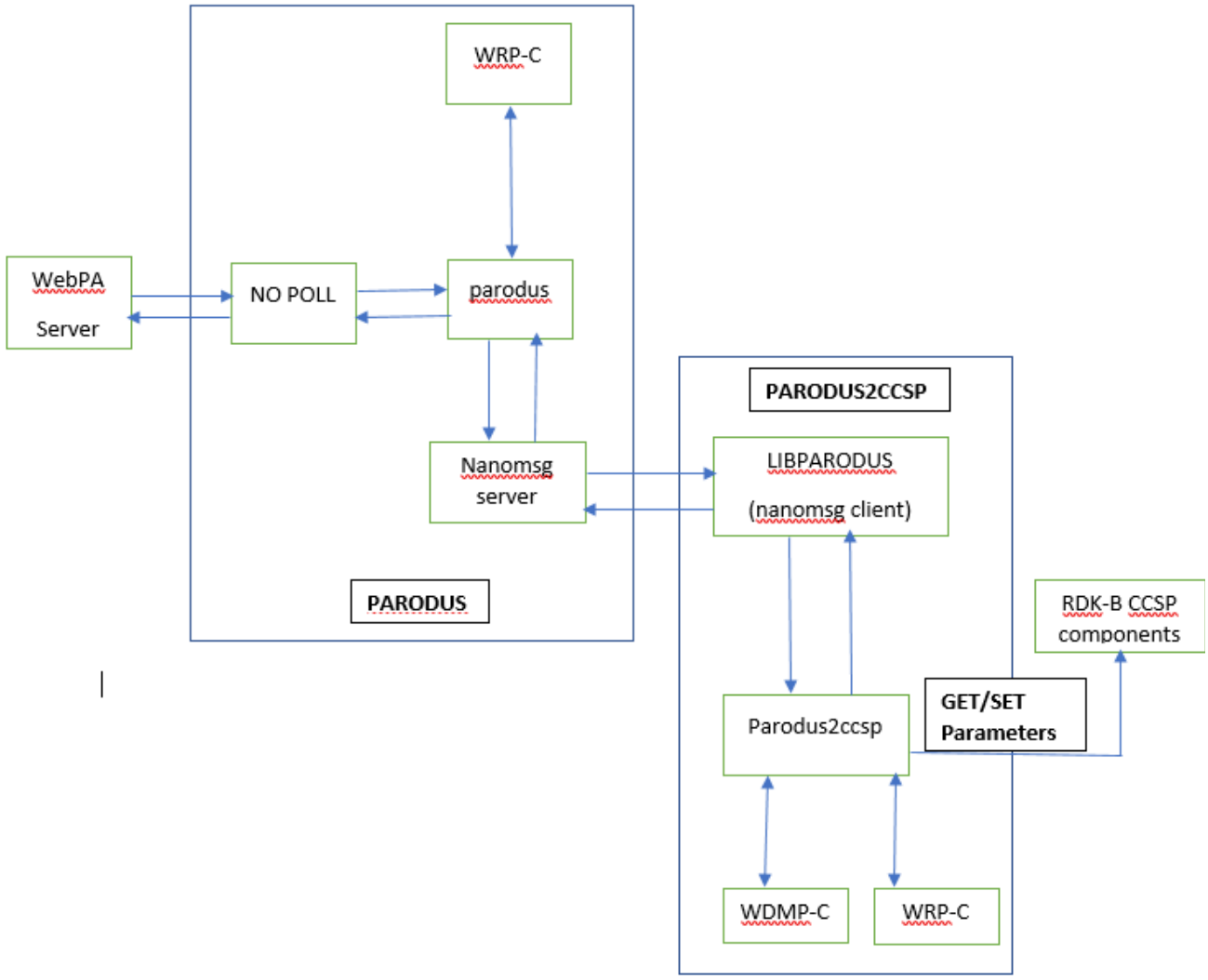
WebPA is the communication channel from Cloud to RDK based home gateway devices. It helps to manage devices from Cloud webpa_swagger. W ebpa client to communicate with parodus in RDK environment.

Parodus :

Parodus is a websocket client which connects to Webpa servers in the cloud & delivers messages to services on the client device.

Parodus-WebPA Structure :

In RDK-B on RPI/Emulator, parodus and parodus2ccsp processes are run on RPI/Emulator.The two process communicate using nanomsg TCP bus as their interprocess communication channel.



How Communication Happens :

Nanomsg is used to communicate with Parodus on the device. This replaces previously defined RPC mechanisms. Message queues are implemented for both incoming WRP messages and outgoing WRP messages from nanomsg. This is same as nopoll incoming and outgoing queue architecture. Nanomsg is used in the following capacities:

- **Nanomsg Server:** Parodus is a nanomsg server, that listens to incoming client requests & provides a mechanism for services on the device to register with Parodus.
- **libparodus/Nanomsg Client:**libparodus is provided as a simple c library for any client that wants to communicate with Parodus. Service registration as well as "fire and forget" use cases are supported.

Nanomsg Registration (provided by libparodus):

This message is used by on device services that wish to register to parodus. During registration service name and port for sending data will be shared as part of this new WRP message registration_message. Registration msg will be sent to nanomsg server, that msg will contain specific URL of that client and service port number.

These registration port and service details are stored in a list in parodus which will be used to push or send messages downstream. Whenever any client gets registered with parodus through libparodus that client will be added to that list. Clients will get deleted from that list when parodus observes that client is killed or dead.

Service Keep Alive Message (provided by libparodus):

To maintain connection with clients parodus sends keep alive messages to clients. It acts like heart beats.

Upstream Communication:

Upstream means sending messages from **Device to Cloud**. Msg_type 4 is used as upstream message. Upstream does not require any registration. Upstream messages will be received as PULL on fixed pre-defined port in Parodus which will be known to all clients without requiring any registration.

Downstream Communication:

Downstream means sending messages from **Cloud to Device**. Parodus processes response coming through webpa server and sends to appropriate client by referring registered client list. Msg_type 3 is used as downstream message. Downstream requires registration, during registration service name and port for sending data will be shared as part of this new WRP message.

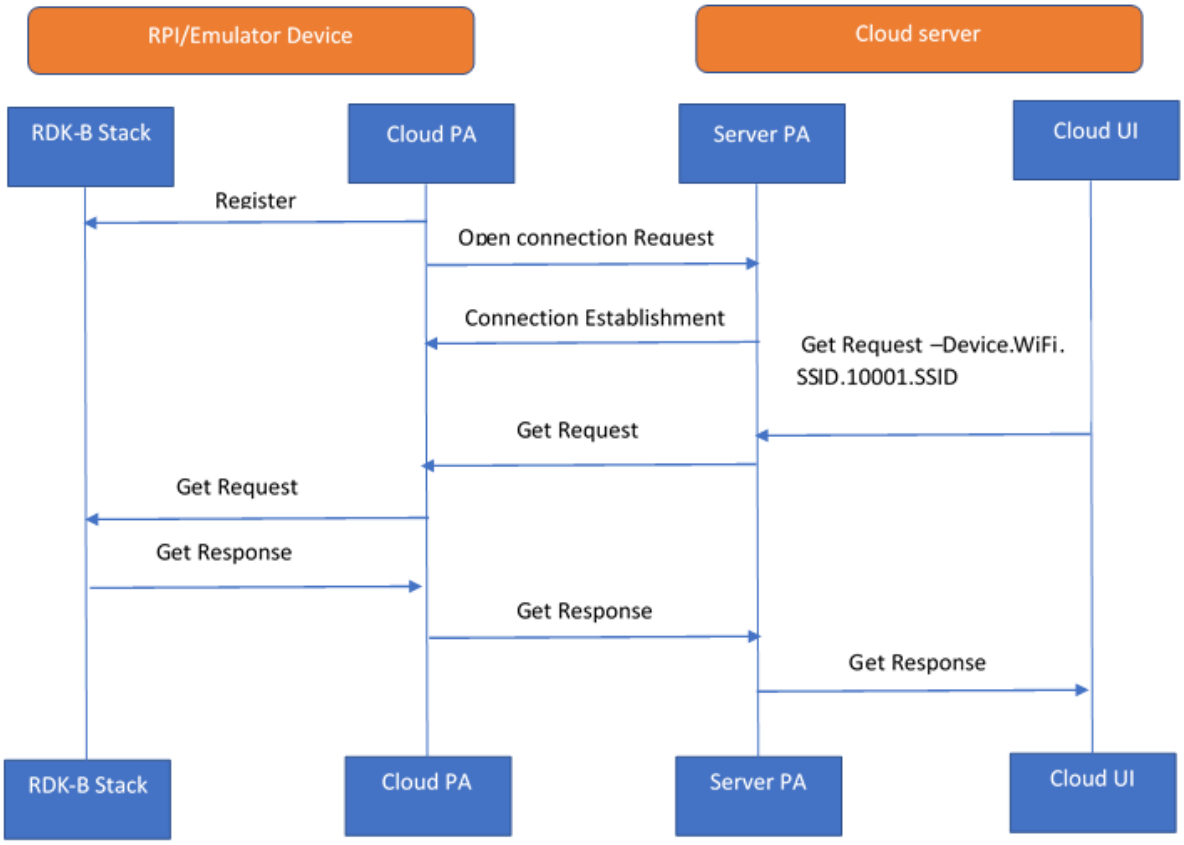
Parodus2ccsp:

This is the RDKB specific client used to handle Ccsp stack related operations like GET and SET of TR181 parameters.

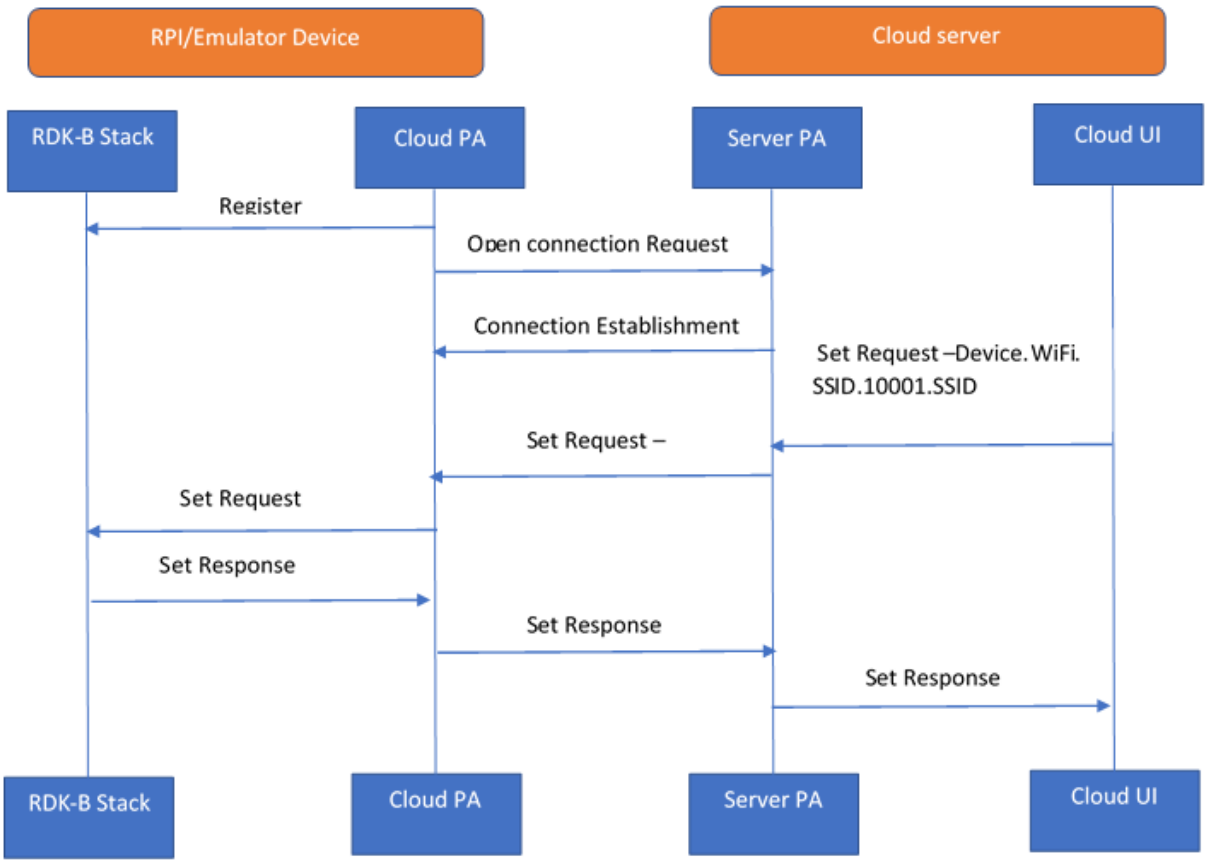
Open Source Libraries Used: Parodus uses below mentioned libraries:

- **WRP-C:** Library to pack/unpack messages in WRP format. It converts msgpack into bytes/String/base64 and vice-versa (ref. WRP-C).
- **WDMP-C:** Webpa DATA Model parser and used to parse GET/SET request/response messages. (ref. WDMP-C)
- **NANOMSG:** for handling TCP socket client and server communication as a bus on 2 different processors.
- **LIBPARODUS:** nanomsg client, provides connectivity to parodus services, acts as parodus client (ref. libparodus).
- **PARODUS**: WebPA client coordinator (ref. parodus).

GET Parameter Sequence Flow of WebPA :



SET Parameter Sequence Flow of WebPA :


RPI/Emulator Device
Cloud server
RDK-B Stack
Cloud PA
Server PA
Cloud UI
Register
Open connection Request
Connection Establishment
Set Request –Device. WiFi.
SSID.10001.SSID
Set Request –
Set Request
Set Response
Set Response
Set Response
RDK-B Stack
Cloud PA
Server PA
Cloud UI