



**Open Source
Foundation**

prplMesh

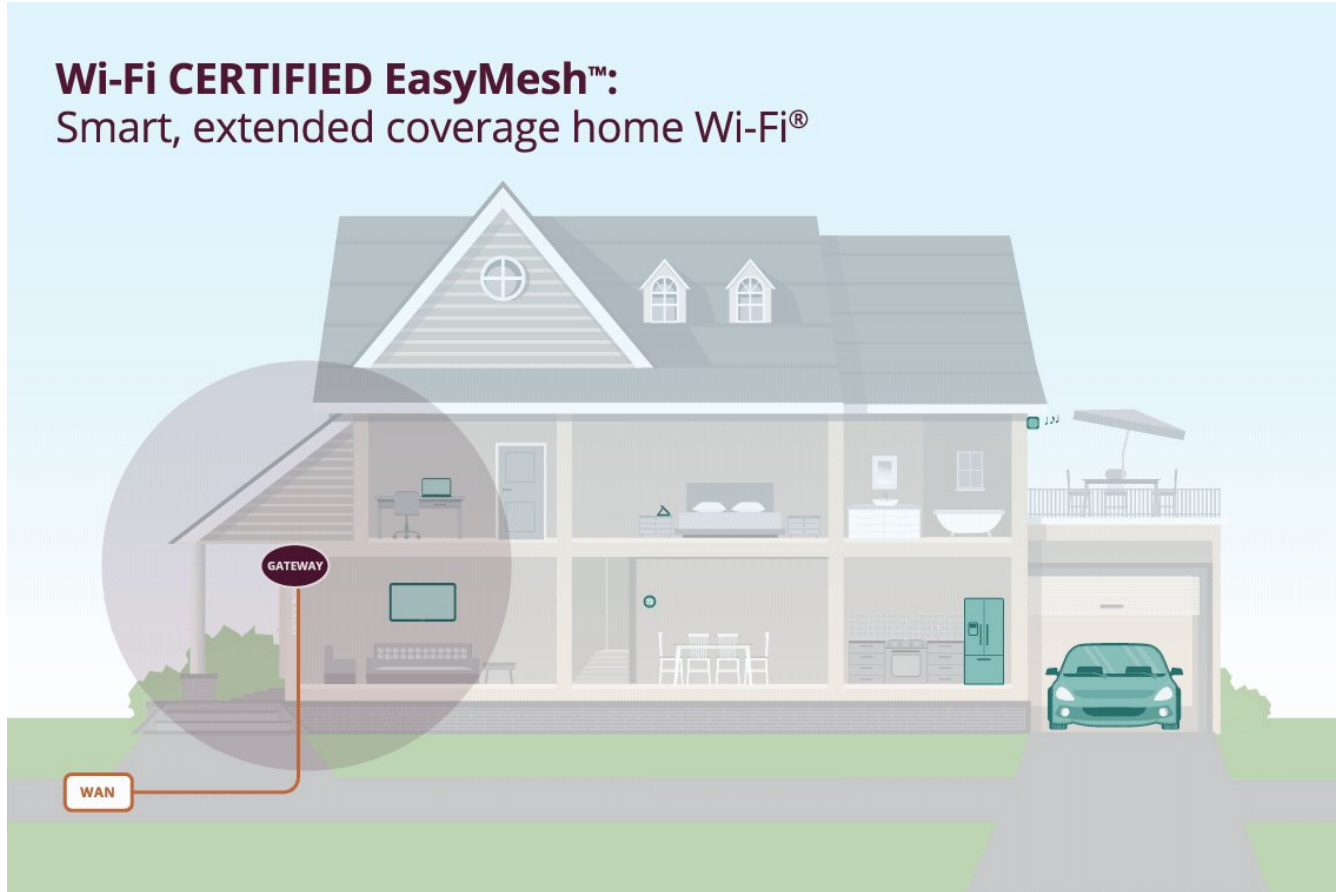
Prpl Summit 2019
Berlin
2019-10-23

CERTIFIED EasyMesh™



Whole-home Network Coverage

Wi-Fi CERTIFIED EasyMesh™:
Smart, extended coverage home Wi-Fi®



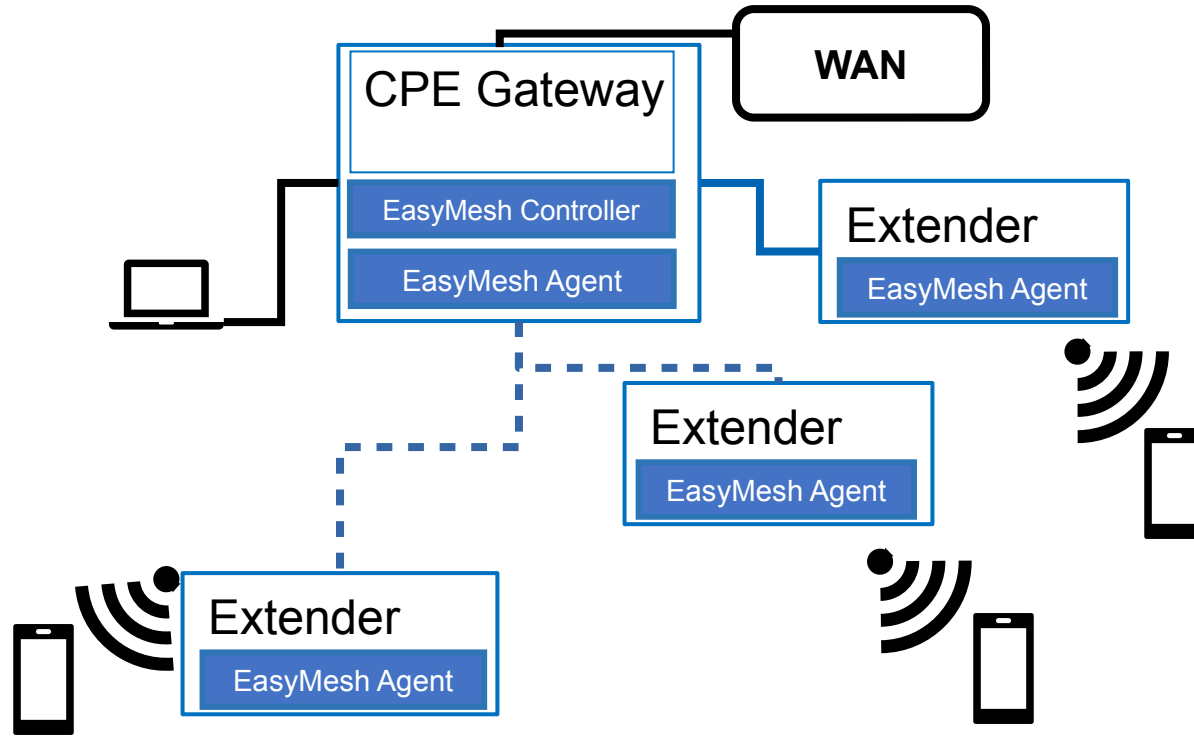
Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

What is EasyMesh?



- **Protocol name: Multi-AP**
 - ▶ Based on IEEE1905.1
- **One Controller**
 - ▶ Master of the mesh
- **Several Agents**
 - ▶ Wireless access points

Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

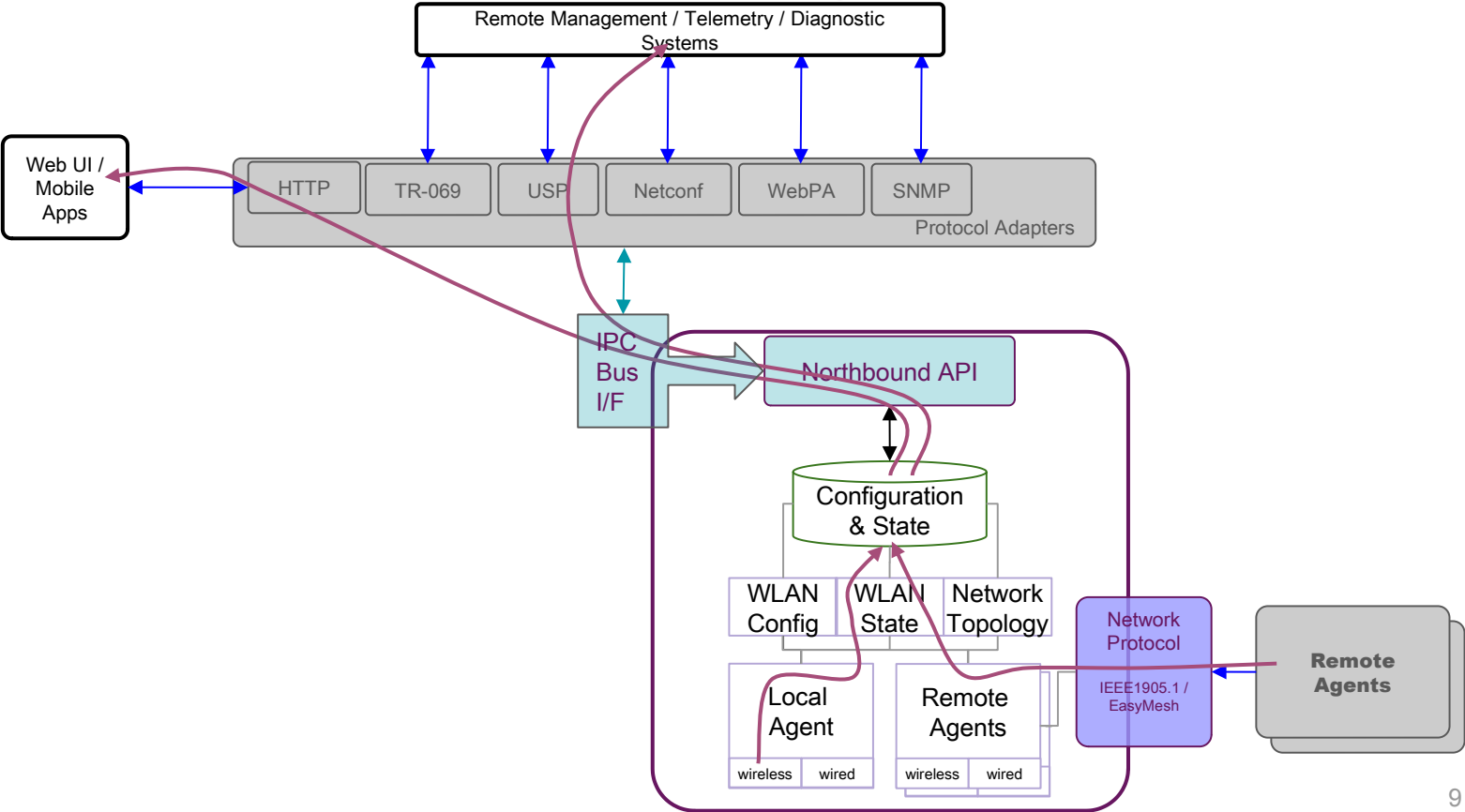
Management API

The Central Configuration Problem

- Stand-alone access point: configuration *through web UI, mobile app, remote management system*
- Distributed network of access points: configuration through Controller, requiring careful synchronisation logic, *through web UI, mobile app, remote management*
- Different API for each? Let configuration client detect the use case and select?

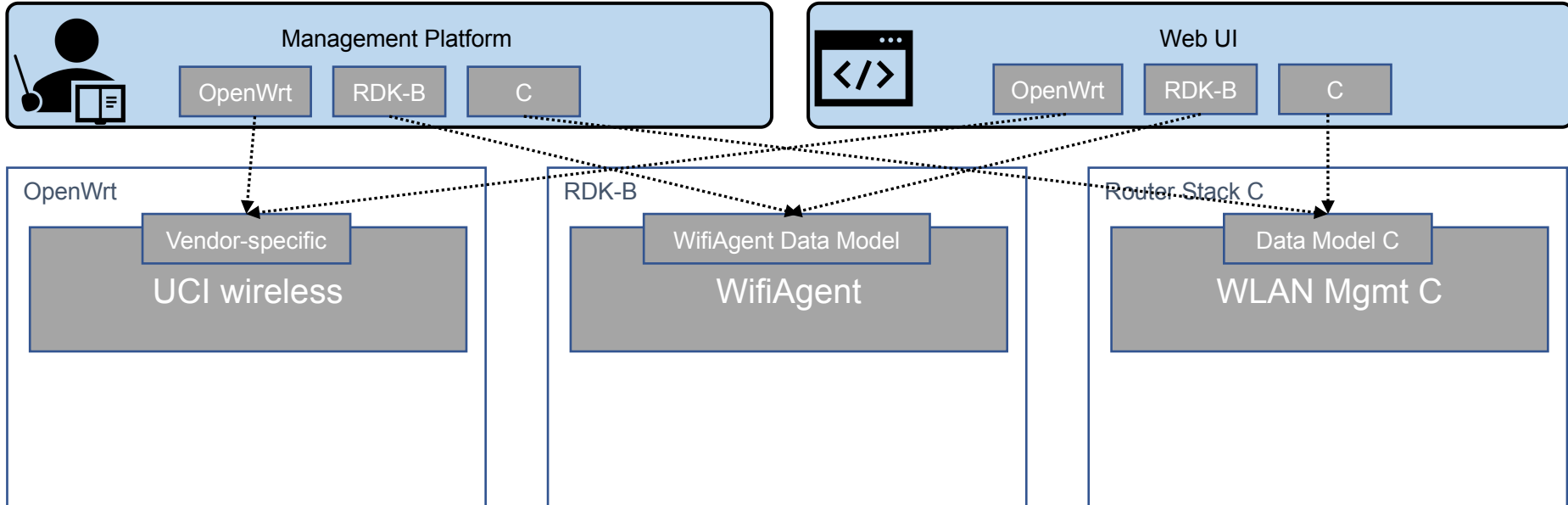
Logical solution: converge to
single API, always through Controller
regardless of whether extenders are present

Central API Access through Controller



Management API

Today: different OSS implementation, different web UI/mobile app per gateway



Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ EasyMesh standard

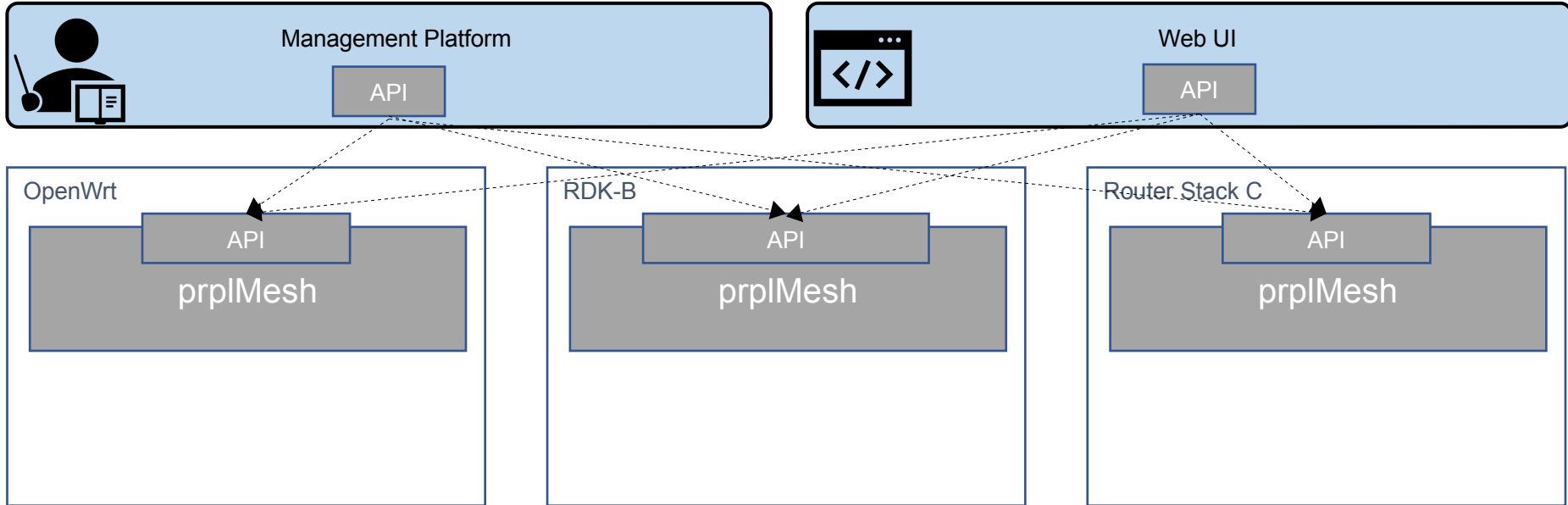
Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

Management API

prplMesh: single, common, standardised API → reuse of OSS / UI development



Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

→ **Standardised management API**

Northbound Interface

Outside World Interface to prplMesh

Control Function API

Evolution of the prpl Foundation [High Level API](#)

Defined by: Broadband Forum OB-MAP project stream, prpl

Defined as:

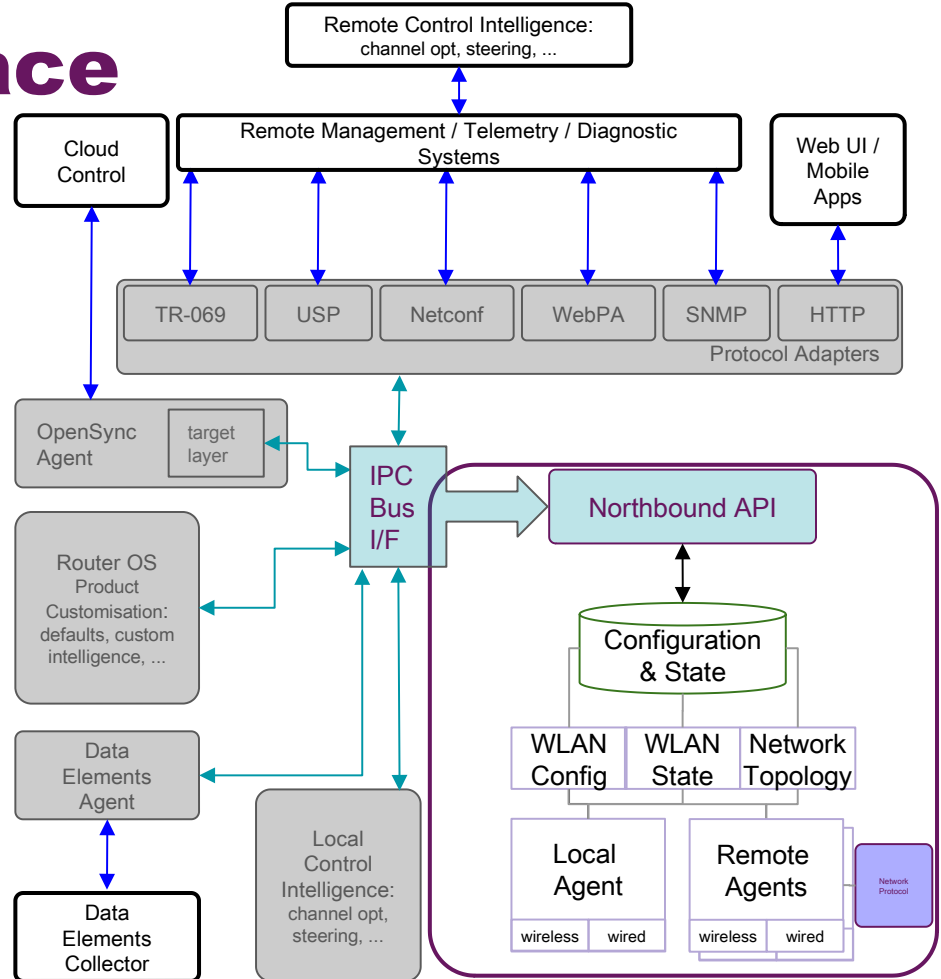
- object/parameter getters and setters
- real time subscription to changes and events
- functions

Access:

- **local**, through the local IPC bus of the router OS
- **over the network**, translated by a protocol adapter

Scope: *control, state, diagnostics, telemetry, topology*

- wireless configuration (HL API, TR-181, RDK-B)
- EasyMesh configuration (EasyMesh, OBMAP)
- wireless state (Data Elements, RDK-B wifihal, OpenSync)
- network topology (IEEE1905.1, EasyMesh, TR-181)
- event logging
- self-organising network control operations (EasyMesh Controller)



Wireless Management Complexity

Evolution of Complexity

Wireless Access Point in 2013:
Best effort service

- IEEE802.11a/b/g/n/ac
- Dual-band, dual-concurrent
- DFS in 5GHz, radar detection
- Per-region regulatory frequency issues
- Single SSID, security mode, credentials
- WPA, WPA-2, WPA-Enterprise

Can everyone keep up?

Wireless Access point in 2019:
Critical infrastructure

- IEEE802.11ax
- OFDMA, scheduling
- Coordinated channel change
- Upcoming 6GHz band
- Wireless backhaul links: concurrent operation as AP and STA
- Distribution of credential configuration, WPA3, EasyConnect
- Hotspot, Agile Multiband (ANQP)
- Band, client steering
- QoS, low latency
- Diagnostics and telemetry collection

Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

→ **Standardised management API**

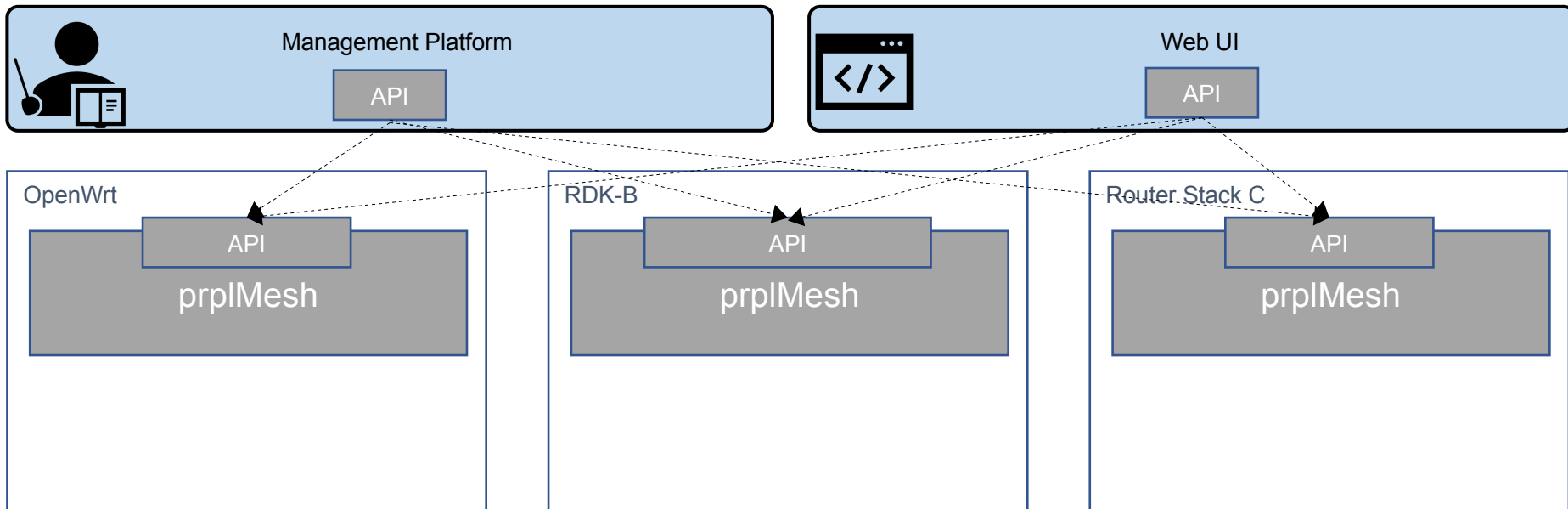
Growing wireless management complexity

Ongoing effort and cost.
Divergent behaviour.

3

Wireless Management

prplMesh: portable, common, open source implementation



Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

→ **Standardised management API**

Growing wireless management complexity

Ongoing effort and cost.
Divergent behaviour.

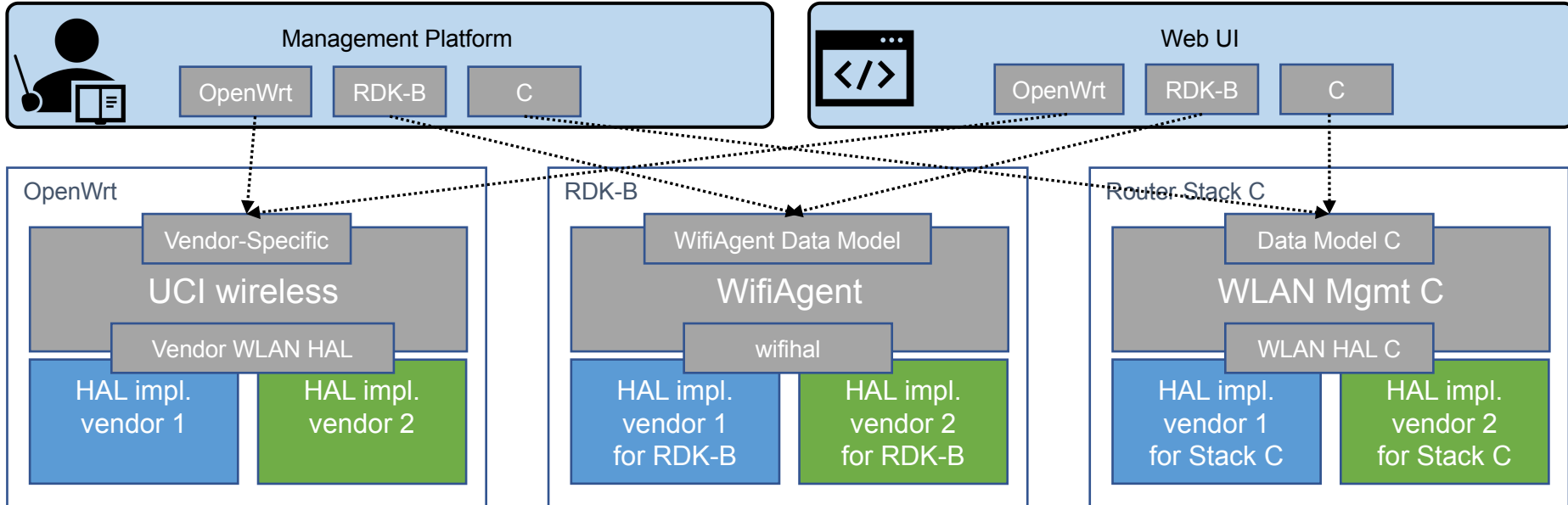
3

→ **Common wireless
management implementation**

Wireless Silicon Interface

Management API

Today: different OSS implementation, different web UI/mobile app per gateway



Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

→ **Standardised management API**

Growing wireless management complexity

Ongoing effort and cost.
Divergent behaviour.

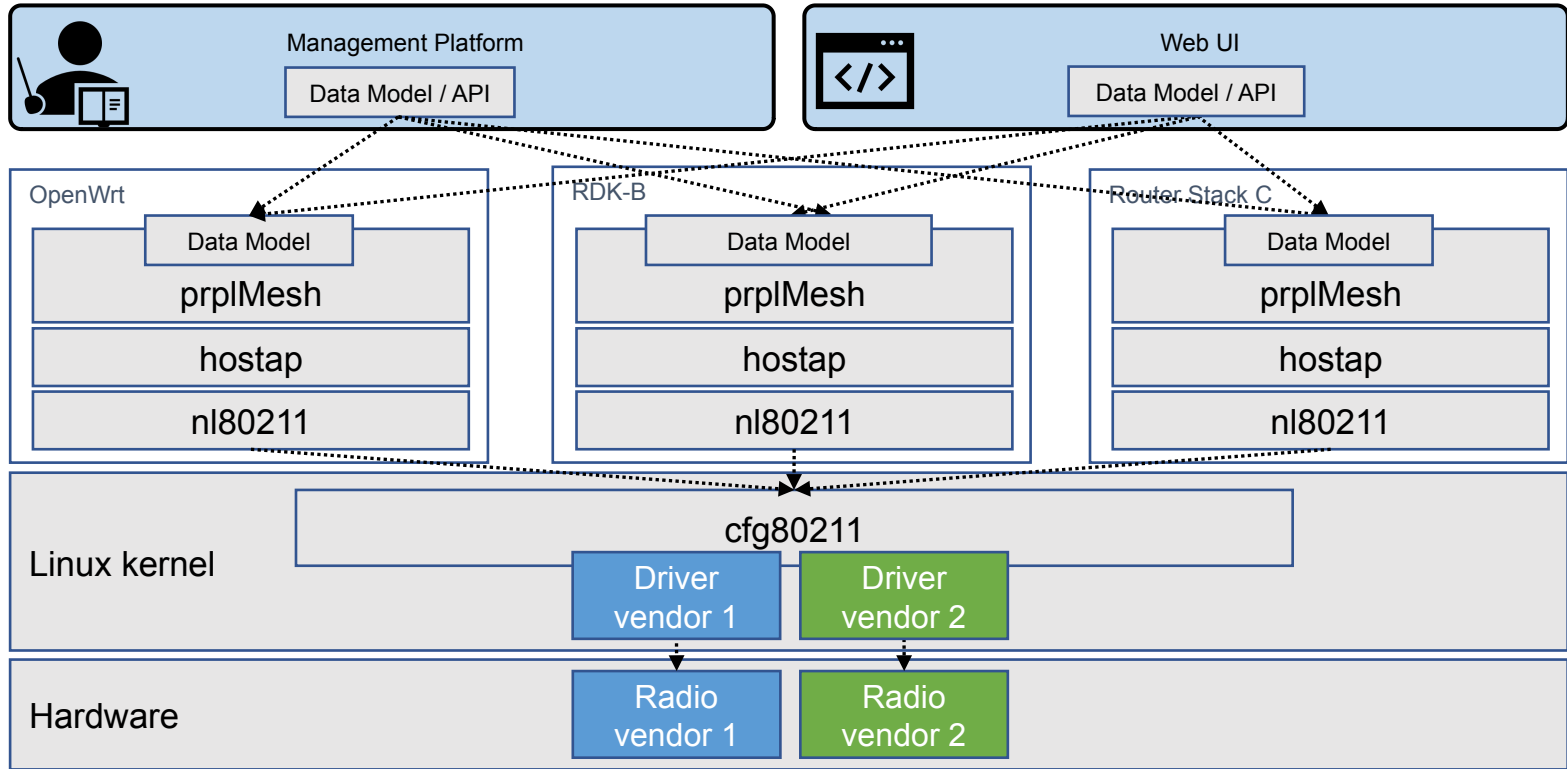
3

→ **Common wireless management implementation**

Different wireless silicon driver interfaces

Portability impediments.
Development cost of HAL implementations.

4



Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

→ **Standardised management API**

Growing wireless management complexity

Ongoing effort and cost.
Divergent behaviour.

3

→ **Common wireless management implementation**

Different wireless silicon driver interfaces

Portability impediments.
Development cost of HAL implementations.

4

→ **Standardised wireless driver interface**

prplMesh

Problem Statement

Whole-home self-organising network

Proliferation of mutually incompatible extender solutions.
Lack of operator control and visibility.

1

→ **EasyMesh standard**

Different management API's

Development cost of customer UI's and OSS.
Difference between single AP and distributed network.

2

→ **Standardised management API**

Growing wireless management complexity

Ongoing effort and cost.
Divergent behaviour.

3

→ **Common wireless management implementation**

Different wireless silicon driver interfaces

Portability impediments.
Development cost of HAL implementations.

4

→ **Standardised wireless driver interface**

Why Invest in prplMesh?

Chip vendors gain scalability: convergence towards fewer SW platforms to support

- precious resources better spent on value-add product differentiation

OEMs gain consistency and commonality between different chip vendors

- making it easier for OEMs to switch SoCs

In turn, ISPs gain consistency and commonality between different OEMs

making it easier to switch OEMs

prplMesh is seeking investors to quickly catch-up and keep-up with proprietary stacks

- contribute funding, or SW resources (we have a skill-set profile)
- coding = Technicolor, Intel, Minim, GlobalLogic, Quantenna, CableLabs
- funders = SoftAtHome, Verizon, Intel, CommScope, Vodafone, SmartRG, Quantenna



prplMesh Project Goals

Wi-Fi CERTIFIED EasyMesh™ Reference Implementation

- Business friendly [BSD 2-clause + patent licence](#)
- Pass EasyMesh (Release 1) certification of Agent and Controller protocol on a reference platform

Platform Independence

- Allow for integration into any router Operating System based on a recent version of the Linux kernel
- Implementations of open source router Operating Systems: OpenWrt and RDK-B

Advance prpl Foundation Carrier Interest Group low-level API recommendations

- Use prpl low-level API for wireless, Ethernet switch, Ethernet port, Ethernet PHY

Common Wireless Management

- Standardised central control of ever more complex wireless management across all router stacks

Reference Platform

- [Turrís Omnia](#) with:
 - Wi-Fi 5: retail Qualcomm wireless radio cards (802.11n+802.11ac) with open source Linux mac80211 device drivers
 - Wi-Fi 6: Intel wav600 wireless radio cards (802.11ax) with production Intel cfg80211 device driver



prplMesh Project Goals

Cooperation with open source communities

- Collaborate with the linux-wireless community to bridge any gaps in cfg80211
- Collaborate with the hostap community, and upstream support for EasyMesh in hostapd and wpa_supplicant
- Integration in OpenWrt and RDK-B

Cooperation with standards bodies

- The Broadband Forum's OB-MAP project will define standard interfaces and define add carrier manageability features

Cooperation with wireless silicon vendors

- Generic, interoperable solution supporting all wireless silicon solutions

Allow for solution vendor differentiation

- Any Controller or Agent decision making capability is optional. It must be possible to add algorithms, outside the prplMesh code, to provide functions such as:
 - Channel optimisation
 - Band Steering
 - Client Steering
 - Backhaul topology optimisation

High Level Architecture

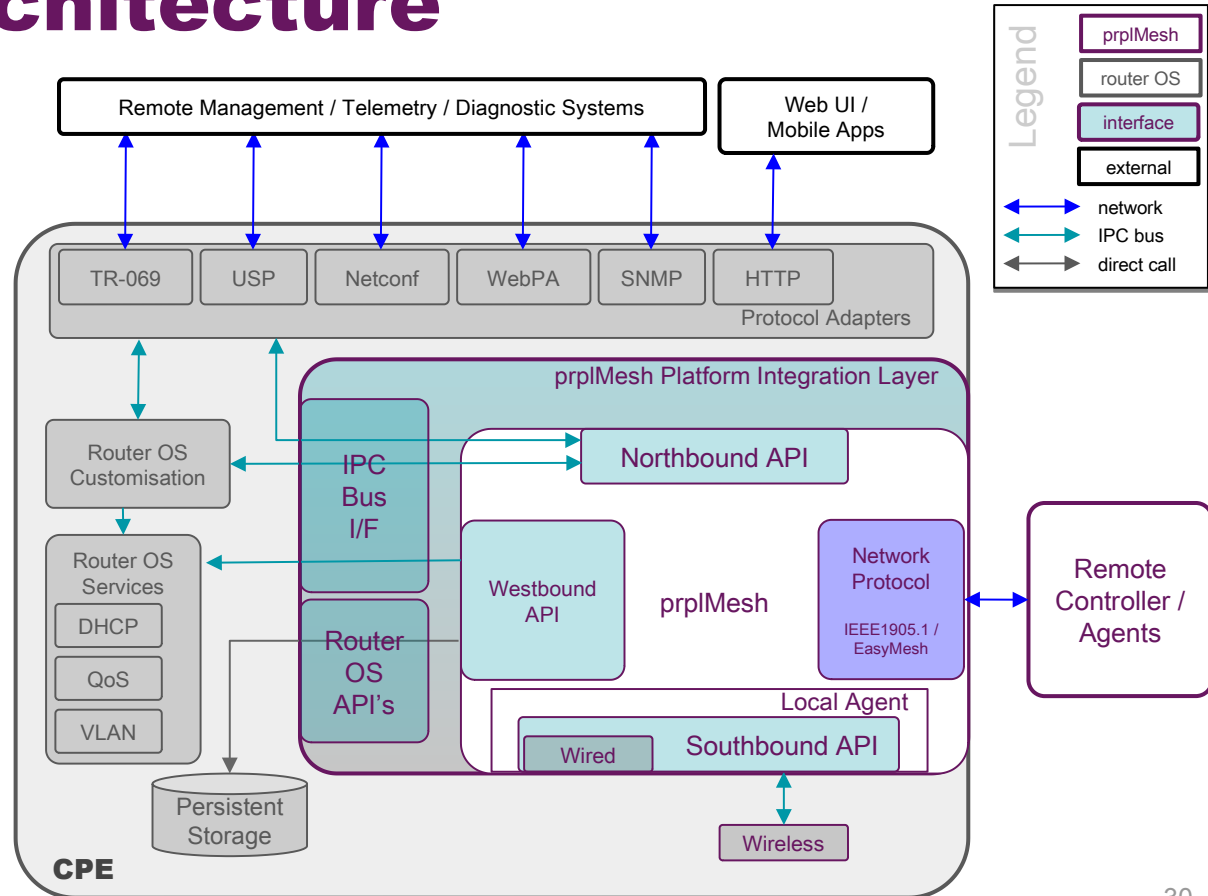
Wireless Management of a router/AP/extender shifts to prplMesh and is controlled through the **northbound API**.

EasyMesh Protocol can be deactivated if not needed.

Other Protocols can be added to support already deployed proprietary solutions.

Local wired network interfaces are passively managed, wireless interfaces fully controlled by prplMesh through the **southbound API**.

Integration into a router OS:
 Software build integration
 IPC bus interface
Westbound API:
 Other services interaction
 Persistent storage



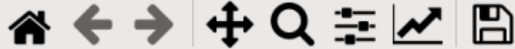
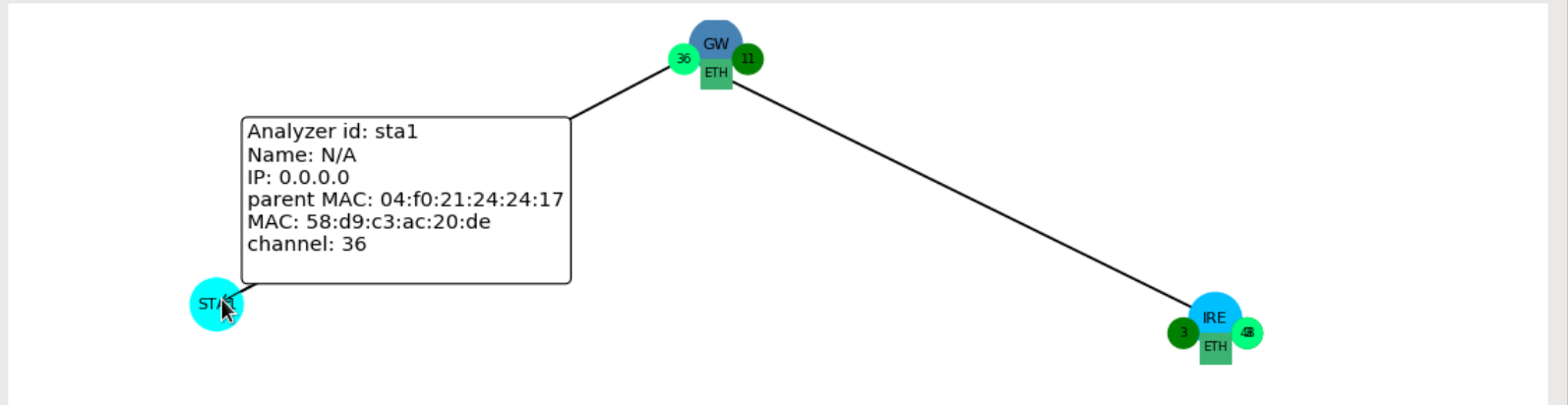
Demonstration

File



File time: 13

ConnectivityMap



x=-544.503 y=-155.44

EventsLogger

```
97.042|BML_EVENT_TYPE_CLIENT_ALLOW_REQ, hostap_mac: 04:f0:21:23:33:5e, sta_mac: 58:d9:c3:ac:20:de, ip: 0.0.0.0  
97.044|BML_EVENT_TYPE_CLIENT_DISALLOW_REQ, hostap_mac: 04:f0:21:23:2f:07, sta_mac: 58:d9:c3:ac:20:de  
97.046|BML_EVENT_TYPE_CLIENT_DISALLOW_REQ, hostap_mac: 04:f0:21:24:0f:99, sta_mac: 58:d9:c3:ac:20:de  
97.048|BML_EVENT_TYPE_BSS_TM_REQ, target bssid: 04:f0:21:23:33:5e, disassoc imminent: 0
```


**Thank
you**



**Open Source
Foundation**