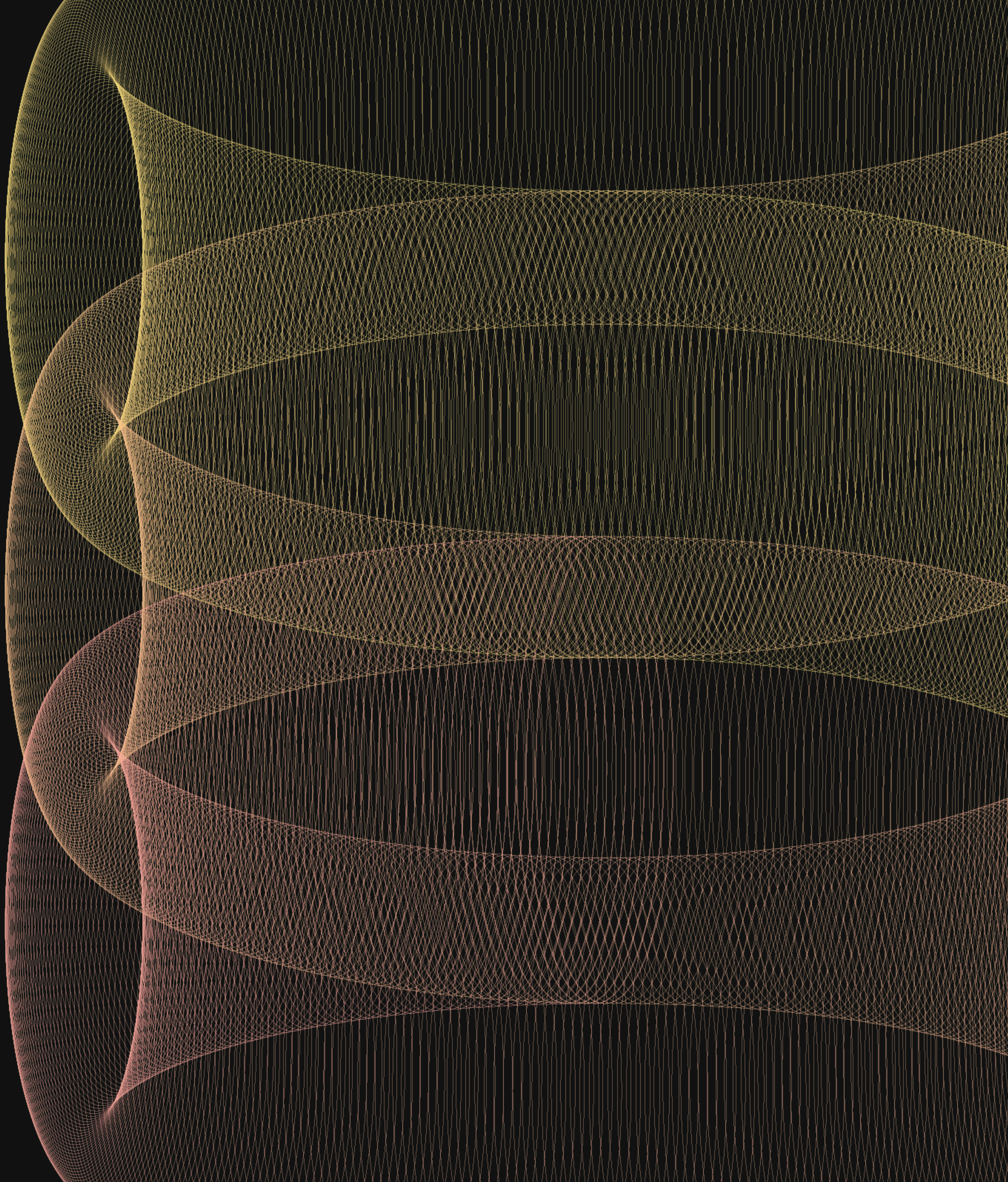
Software Stacks for Networking

A guide for Service Providers

Plume®



Introduction

One of the fundamental choices facing Communication Service Providers (CSPs) is which software to use on the networking devices they supply to their customers' homes and businesses. This decision can have a material impact on the future of the CSP, affecting crucial areas such as

- **networking capabilities that can be supported.**
- **additional services that can be provided.**
- **the effort and time to launch new devices and services.**
- **breadth of device and chipset suppliers that can be selected.**
- **the cost of chips and devices to the CSP.**

In addition to impacting the entire sweep of a CSP's product offering, the selection of the Customer Premises Equipment (CPE) software tends to be a strategic, long-term choice. Major updates to the firmware on devices in the field can be challenging and time-consuming, so a CSP should choose carefully and ensure that the device selected can adapt and evolve. Finally, it can be beneficial to have the same software across as much of a CSP's product portfolio as possible to unify the observability and control of devices and to unify the features, services, and capabilities across all of a CSP's deployments.

This whitepaper outlines the options available regarding software for CPE devices, including the major individual software components. It also looks at the various Software Development Kits (SDKs) that package a number of the individual components into pre-integrated and pre-tested complete solutions. By comparing the advantages and disadvantages of the various approaches, CSPs can choose a path that may provide them with the greatest benefit.

Major updates to the firmware on devices in the field can be challenging and time-consuming, so a CSP should choose carefully, and ensure that their choice can adapt and evolve.

Components

Modern CPE devices include home gateways, routers, access points, extenders, and repeaters. These devices support WiFi and Ethernet networking, broadband connections such as cable, fiber, and DSL, and may also support a range of other networking and connectivity-related services.

Figure 1 shows the main components that go into a modern networking device software stack. This paper walks through each of these elements, from the top of the stack to the bottom.

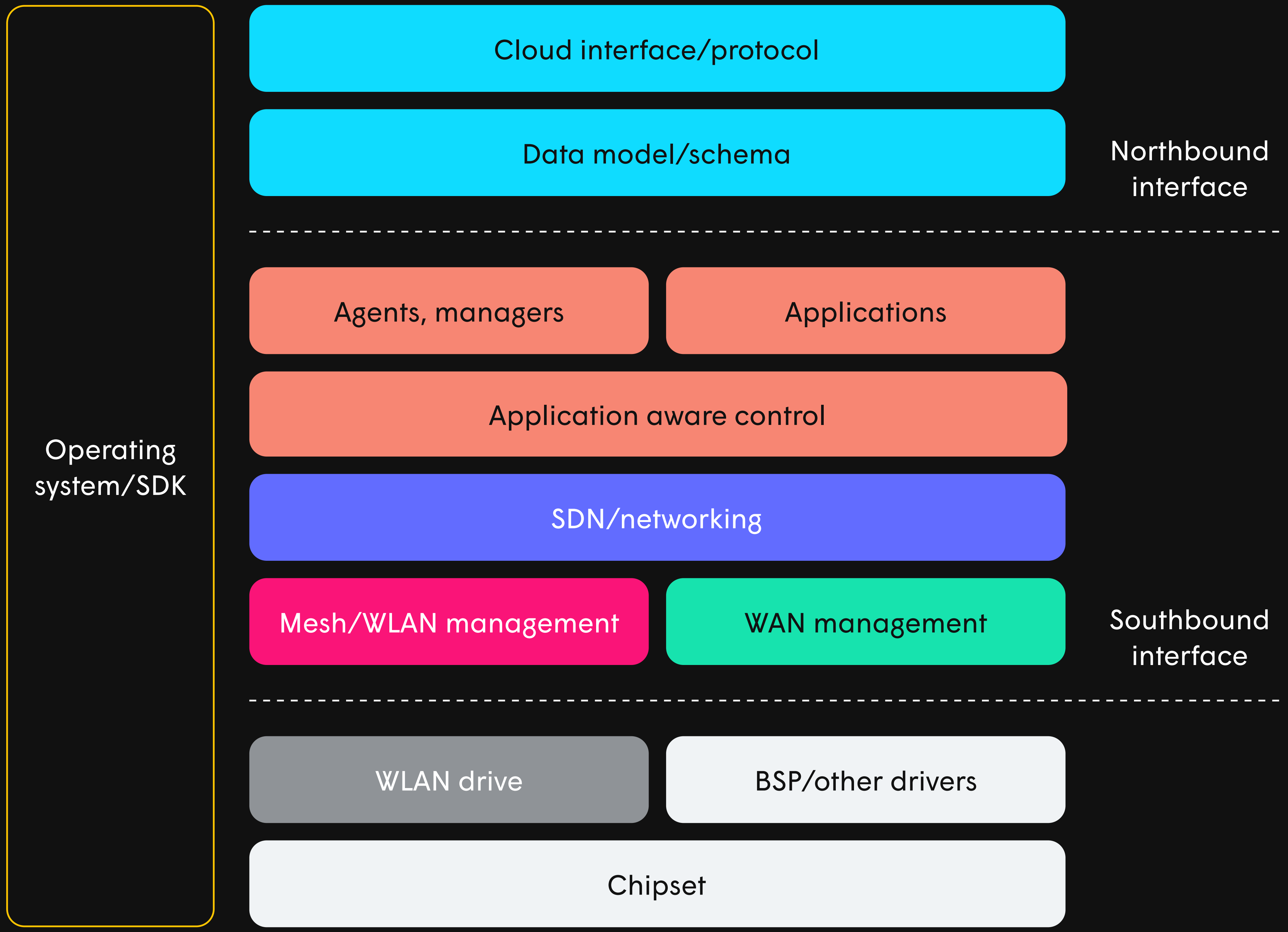


Figure 1

Cloud interface/protocol

Home CPE devices, such as WiFi access points (APs), were historically standalone devices that routed data but did not interact with the cloud for their own management. Modern networking devices generally have a connection to the cloud, which may offer a tremendous number of advantages over systems that cannot be managed from the cloud, such as

- superior observability and monitoring.
- centralized control with a unified, hardware abstracted interface.
- large amounts of compute power and memory in the cloud for new services.
- availability of ready-to-use, sophisticated, machine-learning algorithms in the cloud for advanced insights.
- ease and speed of rolling out upgraded services, at times without major changes to the firmware on devices.

The cloud interface defines the protocol by which the devices will communicate with the cloud. The options that should be considered for this are

TR-369: This is a standard defined by the Broadband Forum, also known as the User Services Platform (USP). Broadband Forum standards have traditionally been adopted primarily by the Telecoms (Telco) industry. This standard includes advanced support for having multiple clouds (potentially from different vendors) control different aspects of the CPE device. As an evolution of the broadly deployed (but often inefficient) TR-069, it has garnered a lot of interest; however, there are disadvantages. TR-369 includes multiple options and modes. For example, the underlying communications protocol can be CoAP, STOMP, WebSockets, or MQTT.[1] This may degrade its ability to provide interoperability between different clouds and networking gear unless a leader emerges with a widely-deployed stack to scale the solution.

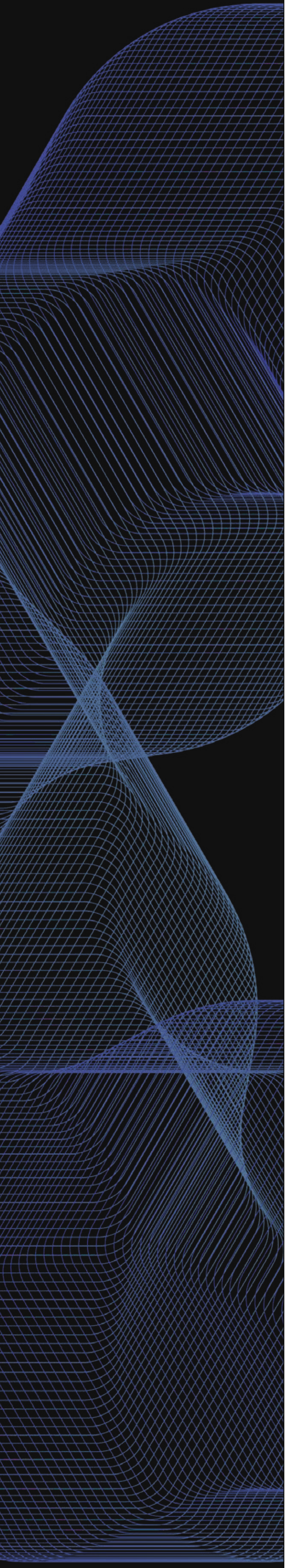
WebPA and WebConfig: These are protocols created by the RDK organization. They are in use primarily by cable operators, although some Telco operators have recently adopted them. WebPA is designed to be a real-time control and measurement interface, while WebConfig was designed for maximum efficiency when downloading large volumes of data to do the initial configuration of networking devices. These protocols are proven at scale, having been deployed in over 80 million devices.[2]

MQTT and OVSDB: MQTT is one of the modes supported in TR-369. It was designed as an efficient way to receive measurements from large numbers of sensors. It, therefore, is a good choice for the uplink from networking devices to the cloud, as they frequently report measurements and statistics regarding the environment they are in and their operation. OVSDB is a type of database, designed to support Software Defined Networking (SDN). Along with natively supporting SDN operation, it creates self-synchronizing copies of the device state locally in the device and in the cloud. Processes in the device, or in the cloud, can modify this state, and the state will be automatically updated in both locations. It is, therefore, highly effective for sending commands and configurations from the cloud to the device. Both of these protocols have been used at scale, having been deployed in over 40 million homes and businesses, and are in use by some of the largest CSPs in the world.[3]

1 TR-369a2 – User Services Platform

2 RDK Surpasses 80 Million Device Deployments

3 Plume powers more than 40 million active residential and small business locations with its cloud-hosted services



Data model/schema

While the cloud interface/protocol defines how the cloud and networking device will communicate, the data model/schema describes the content of those communications. The data model defines what parameters will be exchanged, what those parameters mean, and how they will be formatted so that both the cloud and the devices interpret the parameters the same way. The industry is consolidating on the data model, which is good news for future interoperability between systems. At this time there are two major options:

TR-181: This is a sister standard to TR-069/369 from the Broadband Forum. It has been in use for many years and is constantly being enhanced with new parameters. It is used by both the cable industry as part of the RDK software suite, and by the Telco industry together with TR-069 and TR-369. Unfortunately, like TR-369, there are many versions, options, and competing sets of TR-181 parameters, so it is important to know not just that it is TR-181, but "which" TR-181. Additionally, TR-181 data models do not support native SDN functions, leaving it to vendors to create proprietary models for any real-time networking service.

OVSDB: The OVSDB protocol comes with a native SDN data model that was originally defined for metro-area and data-center communication. It has been extended to support wireless networking and the additional services listed earlier. As described earlier, this solution is deployed in over 40 million homes.

Operating system/core SDK

Most networking equipment uses an operating system that is one or another variant of Linux. What distinguishes them is all the other core functionality that goes with the basic operating system. These "core" functions are represented by the "Operating system/ SDK" block in Figure 1 above. Further confusing things, the term "Software Development Kit" is used to represent two levels of software packages. As networking has evolved, what was once considered a full SDK has been superseded by SDKs that include more components, and often contain one of the older, more narrowly focused core "SDKs" within it. An example core SDK would be OpenWRT. However, in today's networking devices, OpenWRT forms a base package or core, on which more components are added to produce a more complete and ready-to-run CPE device.

With this in mind, there are three types of core SDKs that are reasonable options for CSPs:

OpenWRT: This has historically been used as the base of networking devices used in Telco networks. It is available as open-source and has already been ported to a large number of chipsets and networking devices.

RDK: Reference Design Kit (RDK) has historically been used by cable operators. It is also available as open-source and also has been ported to a large number of chipsets and networking devices. The features and capabilities of OpenWRT and RDK are commensurate. Recently RDK has been gaining adherents in the Telco space as well. RDK has very strong industrial backing from Comcast, while OpenWRT is currently a more grassroots effort.

Chipset proprietary SDKs: The major chipset manufacturers often supply SDKs of their own, specifically for their chipsets. These are typically not available as open-source, but require a license and agreements with the chipset manufacturer. These SDKs may be customized versions of OpenWRT or RDK. The proprietary SDKs can be advantageous because the chipset manufacturers tune the software specifically to the capabilities of their chipset, utilizing more advanced WiFi networking features, or more effective network acceleration techniques. Often the chipset vendors will support new features on their proprietary SDKs before they are supported on the open-source SDKs like OpenWRT and RDK.

Application-aware control: This layer monitors and controls traffic flows in the home based on the specific application type. Until today, networking monitoring and controls were largely application unaware; however, the majority of services dealing with connectivity, security, and quality of experience management require application awareness to be included in the SDK software stack by default.

Agents, managers, and applications: This layer serves as the glue between the cloud interface and the raw networking capabilities in the CPE device. These layers are what determine which features and services can be supported. A given SDK will come with a built-in set, but additional modules can be added to extend the capabilities. Each of the core SDKs (OpenWRT, RDK, proprietary) includes many functions, which is a key area of differentiation between the higher-level SDKs. Therefore, this will be discussed in more detail in the section on high-level SDKs.

SDN/Networking

There are currently two major options in the industry for the networking layer:

- **The standard Linux bridge:** This has the advantage of being supported by the hardware networking accelerators on most chipsets as they are delivered with reference software from the chipset vendor. It has the disadvantage of limited flexibility. Deploying new services on the Linux bridge typically requires firmware updates on the devices, which are slow to deploy, and are prone to failure.
- **Open Virtual Switch (OVS):** OVS is the leading software-defined networking switch. It is flexible and configurable, often allowing the deployment of new services (without having to upgrade the firmware on the device) just by reconfiguring the SDN switch. OVS is required to be added to the Linux kernel in order to use the SoC built-in hardware acceleration with OVS, which is supported by major chipset manufacturers on their new chipsets as part of their standard reference designs.

MESH/WLAN management

Networking devices for the home or small businesses virtually all include WiFi. Increasingly, they support networks with multiple-APs to provide full coverage at high data rates throughout the home or business. And, it is necessary that these systems support wireless backhaul connections between the APs, eliminating the dependency on having wiring in the environment. Such wirelessly connected multi-AP systems are typically called "mesh networks." A variety of mesh network protocols have been used throughout the industry.

- **802.11s:** One of the oldest mesh systems, it is in use in consumer multi-AP systems from Google. Open source is available for this solution. The main disadvantage is that the protocol is quite old, and there has been little work on it over the intervening years[4], resulting in fewer capabilities than other approaches. In particular, it takes a traditional mesh approach in which the backhaul connections are on the same frequency channel, allowing for a fully-interconnected mesh, but making for a high degree of congestion or self-interference on the one backhaul channel. These issues have resulted in a low level of adoption in the industry.
- **OpenSync:** The open-source OpenSync project includes a system for connecting multiple APs over WiFi or Ethernet. It has three primary advantages. First, it supports the richest set of WiFi steering, topology, statistical measurements, and controls among the various mesh systems. Second, it supports a wide variety of SDN services distributed across the APs beyond just WiFi networking. Third, it is deployed in over 40 million homes around the globe.
- **EasyMesh:** EasyMesh is a protocol for WiFi mesh networks that was standardized by the WiFi Alliance. This is both its strength and its weakness.

The WiFi Alliance is a large, strong standards organization that is very well known. Hence, EasyMesh is broadly known as well. However, the actual adoption and deployment of the technology has been modest and is showing a downward trend. Figure 2 shows a graph of the number of devices receiving certification as the standard has evolved from its initial simplest version to more recent versions.[5]

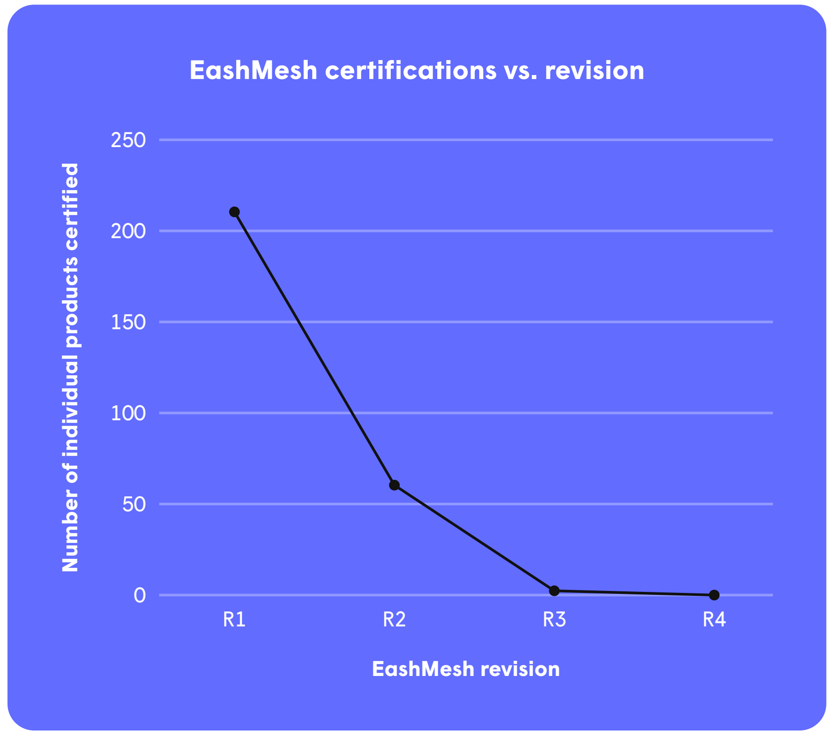


Figure 2

[4] Measurement of the Performance of IEEE 802.11s

[5] Data retrieved from the WiFi Alliance's product finder website

The WiFi Alliance is trying to rectify this, but the solution is creating a new set of problems. Believing the more advanced revisions of the specification to be too complicated for most companies to implement, starting with R4 the Alliance has made most features optional. As with TR-369, when evaluating a vendor with EasyMesh, a CSP must ask which features of EasyMesh are actually included.

The following figure compares the feature sets of Easymesh and OpenSync. It shows the large disparity between the mandatory and optional features of EasyMesh, as well as the overall lack of features in any version of EasyMesh.

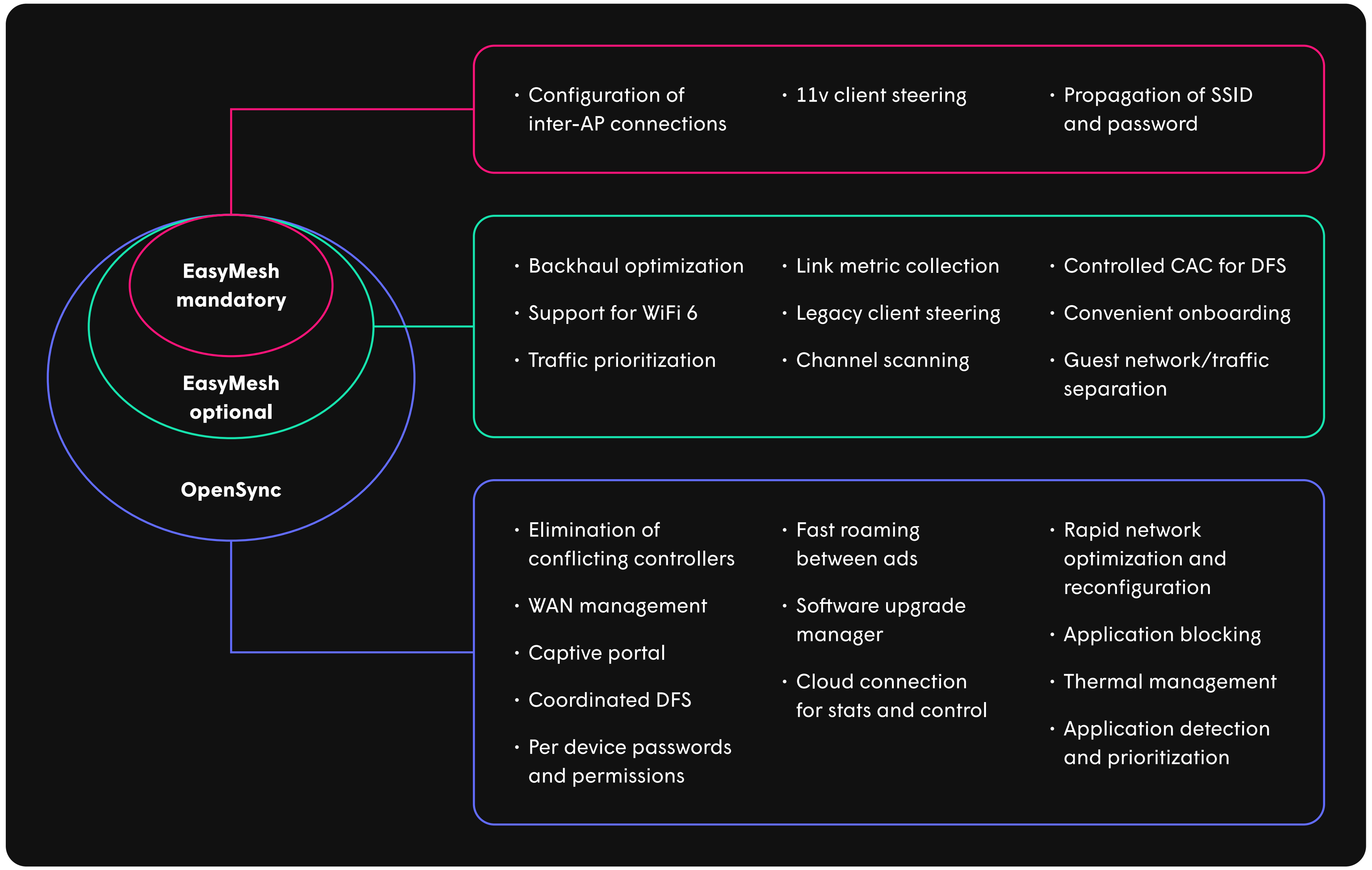


Figure 3

There are several other issues with EasyMesh[6]:

- Only an EasyMesh R1 version is available as open-source[7]. Other implementations have been proprietary.
- EasyMesh was fundamentally designed around local control. Cloud control is not defined or tested for interoperability.
- EasyMesh does not support any services other than WiFi management. It does not include support for cyber-security, parental controls, WiFi motion, etc.

Finally, the basic vision of EasyMesh—equipment that consumers could purchase at retail, bring home, and add to their carrier-supplied network—is both unrealistic and likely undesirable. The more important points of interoperability are at the top of the stack—which talks to the cloud, enabling multiple options for clouds and embedded software—and at the bottom of the stack, enabling the selection of multiple chipsets.

[6] The Advantages of OpenSync over EasyMesh, Plume, May 2022

[7] prplMesh - prpl Foundation

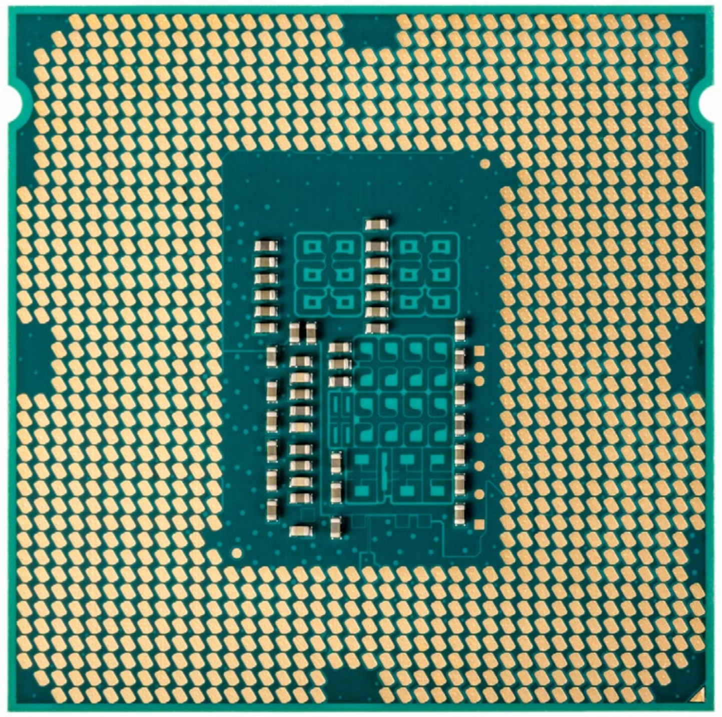
WAN management

WAN management is usually present only in gateway devices and describes the software used to manage the broadband access coming to the premises. Not surprisingly, WAN management methods are split between the Cable and Telco access worlds. TR-069 is generally used to monitor and control DSL and fiber access. WebPA and WebConfig are generally used to monitor and control cable-based access. CSPs have been dealing with WAN management for many years, and many have highly customized systems that they add to a device specifically for their network.

WLAN driver, BSP, and Chipset

Generally, the WLAN driver, board support package (BSP), and chipset all come as a package from the chosen chipset vendor. Chipsets vary in features, performance, and price, and CSPs are generally experienced in selecting a chipset vendor for their products. The WLAN driver deserves special consideration for the way it interacts with the rest of the software stack. Most chipset vendors provide proprietary drivers and management, specifically for their chipset. As discussed earlier, these generally provide the best performance and the most features, and are generally available first after the introduction of a new chipset. However, there is an open-source WiFi management interface called cfg80211, accompanied by tools (hostapd, wpa_suplicant) that is becoming more popular in the industry, and which can be run on chipsets from different vendors. CSPs should do their homework, because not all software stacks support all drivers, and therefore cannot support all chipsets.

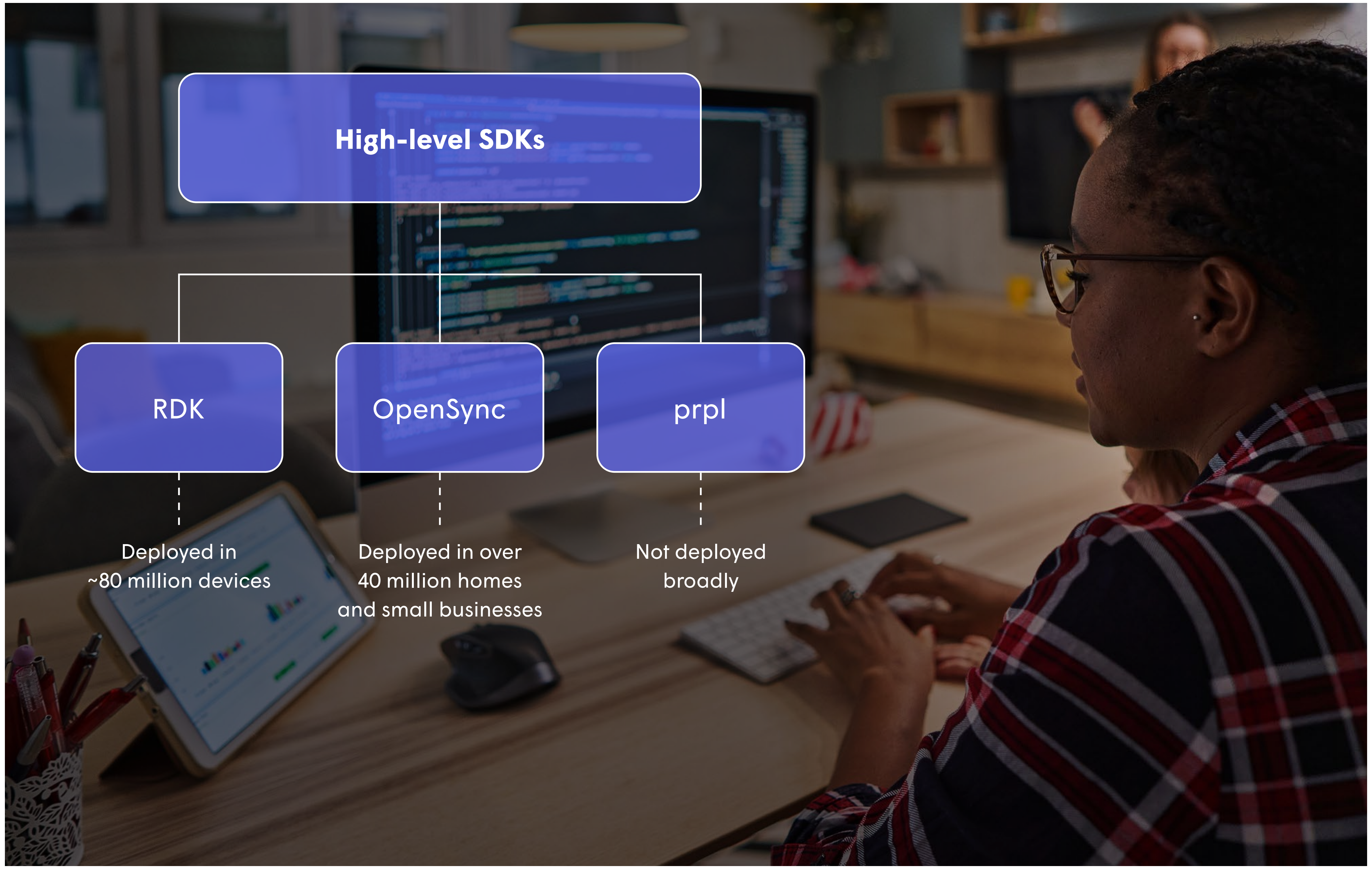
CSPs should do their homework, because not all software stacks support all drivers, and therefore cannot support all chipsets.



Software Development Kits

Hypothetically, a CSP could create its ideal stack by selecting its favorite solution for each component listed in the previous section. However, this would entail significant work to integrate the different components. A more common approach is to select a high-level SDK that pre-integrates a set of components, often including a "core SDK". CSPs typically select one of these high-level SDKs to give themselves a head start and may ask their suppliers to modify any components that do not match the CSPs' respective needs.

The high-level SDKs that are of interest now are RDK, OpenSync, and prpl. RDK is quite complete and widely deployed (~80 million devices total, although many of them are video set-top boxes). OpenSync is similarly complete and deployed in over 40 million homes and small businesses.[8] The third option of interest is prpl. While prpl is not deployed broadly at this time, a number of suppliers and CSPs have expressed interest in this project. Complicating the comparison is that each of these SDKs has a roadmap, making it important to consider both the near-term state of the SDK and the planned status in ~1 year's time (mid-2023 as of this writing).


High-level SDKs
RDK
OpenSync
prpl
Deployed in ~80 million devices
Deployed in over 40 million homes and small businesses
Not deployed broadly

[8] Plume powers more than 40 million active residential and small business locations with its cloud hosted services

Therefore, in the following table, there are two columns for each SDK, one for the current situation as of mid-2022 ("now"), and one for the next major iteration ("future"), roughly mid-2023. The table describes the options for each component that the SDK supports. Cells that are blank indicate that the SDK does not support a given component. It would be up to the CSP or its supplier to add such a component into the software if desired. In cells showing multiple items separated by a comma, either/any of the options are supported. When items are separated with a "+", a single solution that combines the two techniques is supported. In a few cases support for a protocol has been suggested, but what will actually happen is uncertain. These items are highlighted with a question mark.

While the support for various components is a primary consideration, each SDK has other factors that should be considered.

RDK: RDK is a significant open-source effort, being driven strongly by the commercial interests of Comcast and its affiliates. OpenSync is the chosen WiFi, multi-AP networking technology for RDK. It is complete, proven, and broadly adopted. Additionally, there has been discussion about adding an SDN capability to RDK in the future. The main drawback is that it is strongly aligned with cable access and Comcast in particular. This works well for some CSPs, but not for others.

OpenSync: OpenSync is an open-source project (opensync.io) that can provide a complete solution that has been used in both cable-access and DSL/fiber-access deployments. It is unusual in the industry in serving both the Cable and Telco sides of the business. It is supported across all the major chipset vendors and over a dozen different device vendors. NetExperience and Plume provide cloud solutions that can control OpenSync-enabled devices. OpenSync was adopted as the basis for the 1.x series of Telecom Infra Project (TIP) OpenWiFi releases. As part of that effort, TIP offers open-source cloud software that can control OpenSync-capable devices. Multiple companies, including Plume, Comcast, and Iopsys are adding code to OpenSync to support more cloud-facing interfaces. In particular, OpenSync will be aligned with both Cable (WebPA/WebConfig) and Telco (TR-369/TR-181) cloud interfaces, making it interoperable with a range of cloud-control solutions in that time frame.

Component	RDK now	RDK future	OpenSync now	OpenSync future	prpl now	prpl future
Cloud interface	WebPA + WebCfg, TR-069 through CCSP	WebPA + WebCfg	MQTT + OVSDB	MQTT + OVSBD, WebPA/Cfg, TR-369		TR-369, WEbPA/ Cfg?
Data model	TR-181 + WebCfg	"OneWiFi" = TR-181 + WebCfg + OpenSync	OVSDB	Tr-181, OVSDB		TR-181
SDN		OVS	OVS	Linux SDN, OVS		
WLAN mesh	OpenSync in repo	OpenSync in distribution	OpenSync	OpenSync	Easymesh R1	EasyMesh R4
WAN mgt.	WebPA + WebCfg	WebPA + WebCfg	TR-069/181 (ODM supplied)	TR-069/369, TR-181, WebPA + WebCfg	TR-069/369 TR-181	TR-069/369 TR-181
WLAN driver	WiFi HAL	WiFi HAL	WiFi HAL, BSDK, QSDK, cfg80211	WiFi HAL, BSDK, QSDK, cfg80211	cfg80211	Cfg80211, WiFi HAL?
SDK	RDK	RDK	RDK, OpenWRT, QSDK, BSDK	RDK, OPenWRT, WSDK, BSDK prplWRT	prplWRT	prplWRT
Chipset support	Virtually all	Virtually all	Virtually all	Virtually all	QCOM, Maxlinear	QCOM, Maxlinear, others?

prpl: prpl is an open-source effort that has two halves to it. One half is working on prplWRT, a hardened version of OpenWRT that is intended for use by carriers. prplMesh is the half that is working on the higher level SDK for WiFi networking devices. prplMesh has garnered a fair bit of attention through promotion at various standards organizations. However, the organization has been working on it since 2017, and the only thing delivered so far is an implementation of EasyMesh R1, the most basic version of the EasyMesh specification.[9] It is also not broadly available across device manufacturers, and none of the major cloud vendors support it at this time. One of the problems with prplMesh is that it is trying to be all things to all people. Figure 4 shows it supporting virtually all protocols, including OpenSync and RDK. This may be a reason for the slow progress it has made. Given the dominance of prpl by Telco companies, support for Cable protocols like WebPA may be significantly delayed or not happen at all.

The diagram in Figure 4 is from a presentation given to the Wireless Broadband Alliance regarding the architecture of prpl.

[8] prplMesh - prpl Foundation

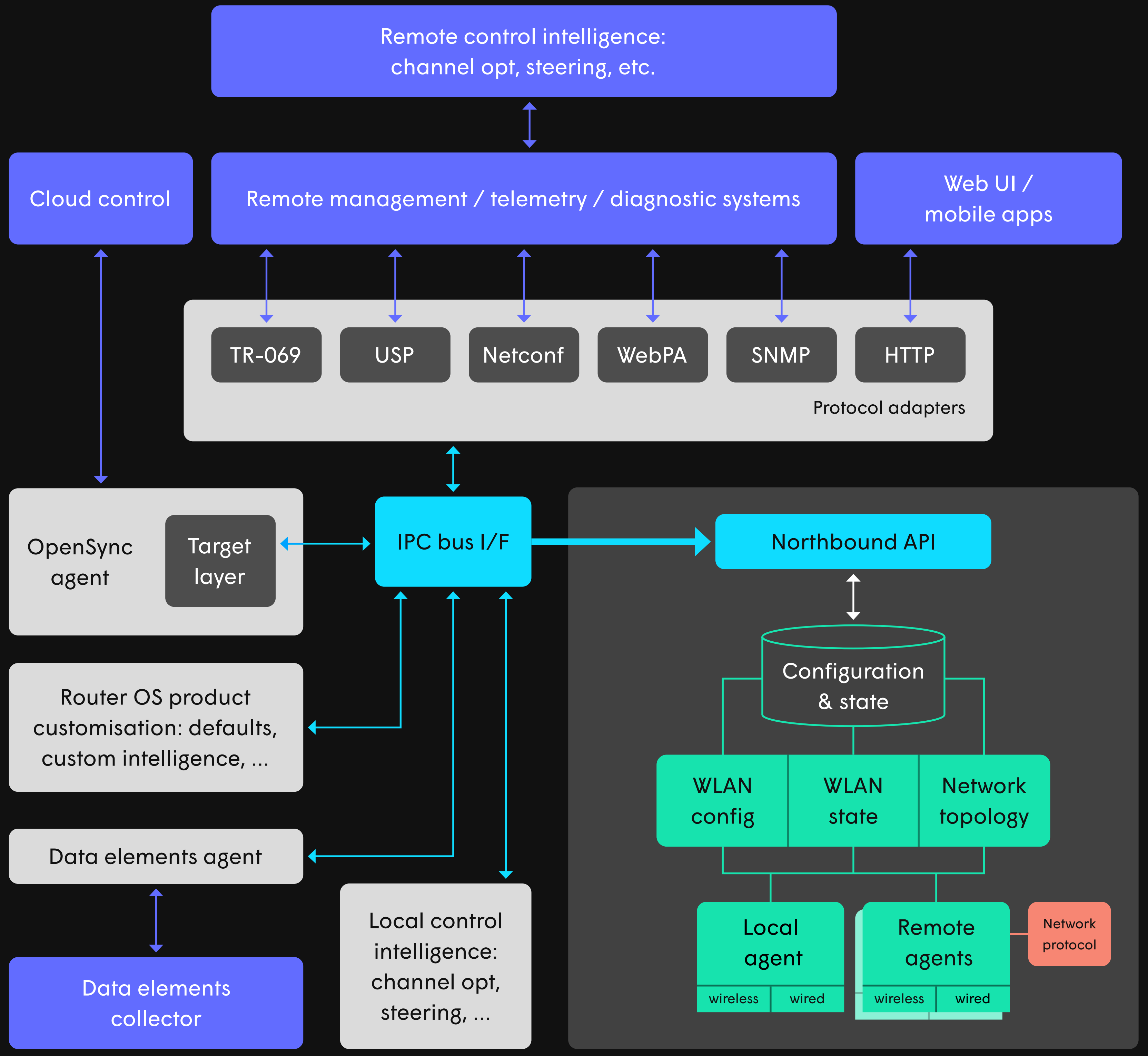


Figure 4

Applications

In this paper, applications refer to services beyond WiFi or network management. These applications are particularly valuable to CSPs as they can be used in a variety of ways to improve CSP's operating metrics. Applications can be

- sold as add-on services, increasing ARPU.
- bundled with higher service tiers, driving higher ARPU through greater adoption of high service tiers.
- included in all plans, creating a multi-services bundle and increasing customer satisfaction, both known to significantly reduce customer churn.

Examples of applications that are becoming popular for CSPs to provide may include

- sophisticated network management (Multi-AP, adaptive).
- advanced device typing (for display in user app and support tools, client steering control, network-wide dashboards, and data mining).
- cyber-security (detecting and blocking viruses, malware, spyware, and remote access).
- parental controls (limiting access to inappropriate content and controlling screen time).
- flow/app control based on policies (for applying prioritization or throughput reservations, gathering customer information, and as an aspect of parental controls, limiting screen time).
- access controls (providing for per-user passwords, and unique per-user access limits to the internet or devices in the home).
- WiFi-based motion detection (in which WiFi signals can be used to create a security alarm system, or a wellness monitoring system).

These applications typically require a combination of cloud software and on-device firmware capability. Figure 5 shows a conceptual view of how all these applications may be constructed. The on-device firmware needs to support four fundamental types of activities: (1) gathering the relevant information that is needed for the application, (2) moving the information to the cloud, (3) receiving controls from the cloud, and (4) enforcing the correct actions once the cloud has made a decision about what to do.

Each application has a specific set of data that needs to be gathered. For example, device typing requires the following information to be gathered on the device and moved to the cloud:

- MAC OUI
- DHCP fingerprint
- Vendor class ID
- Host name
- HTTP user agent signature
- DNS query information
- UPnP exchanges
- Device WiFi capabilities
- Applications running

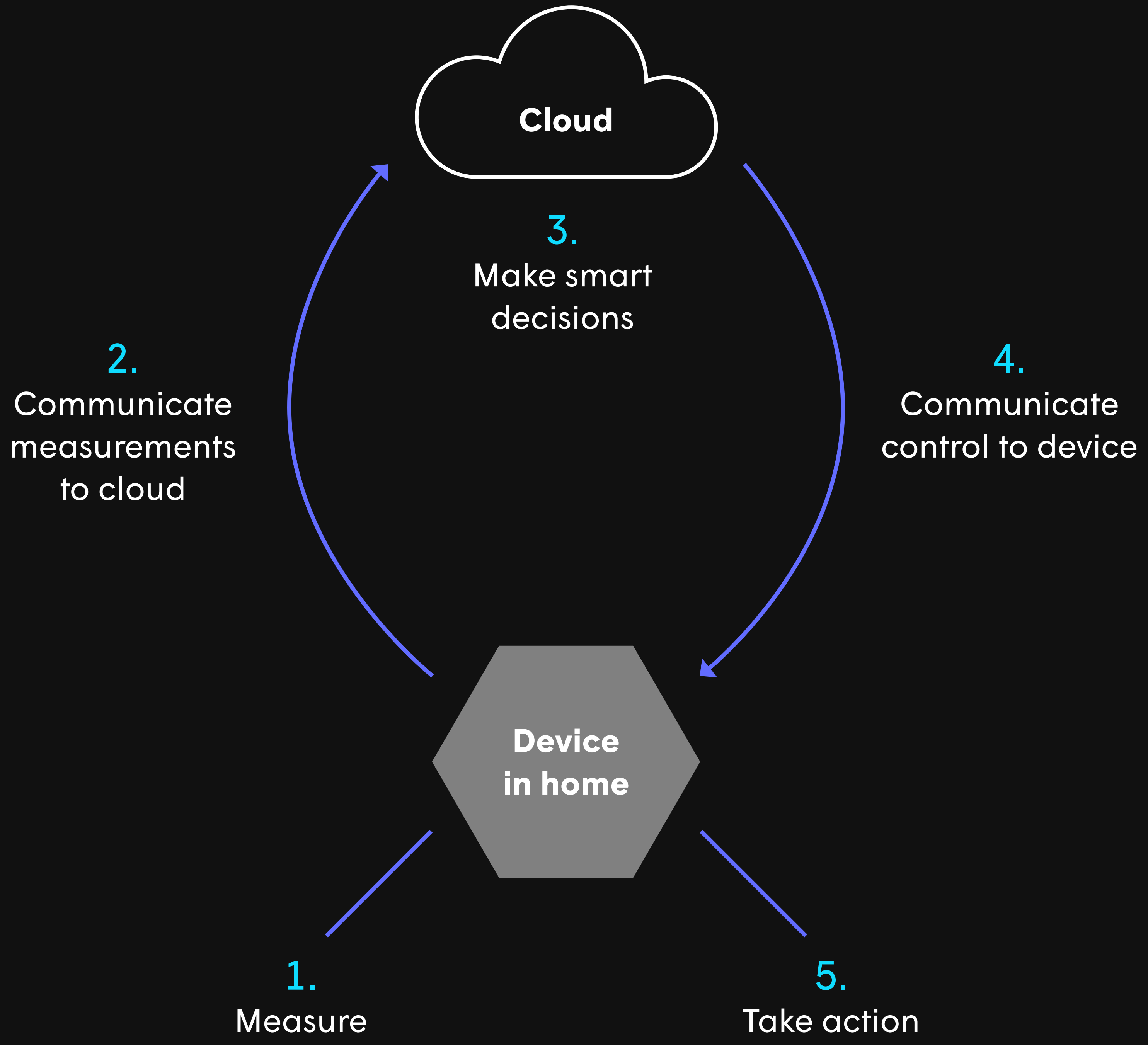
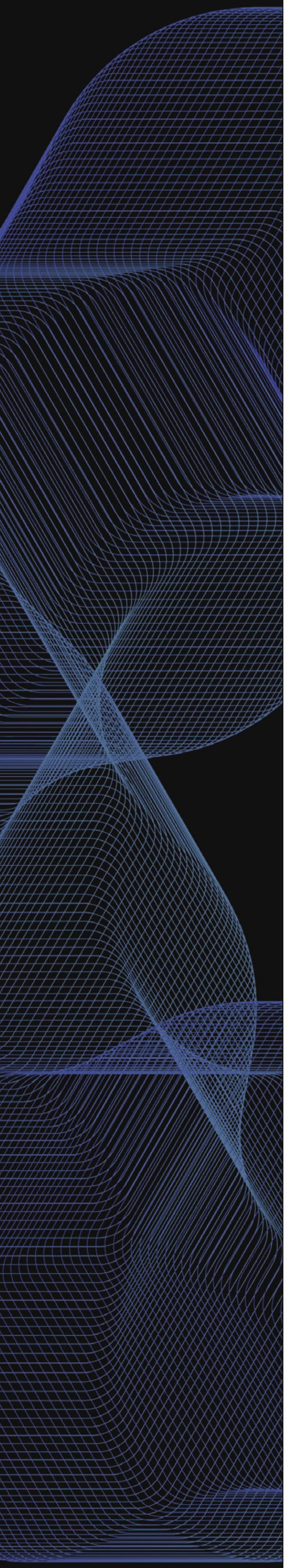

Cloud
3. Make smart decisions
2. Communicate measurements to cloud
4. Communicate control to device
Device in home
1. Measure
5. Take action

Figure 5



In the cloud, sophisticated machine learning algorithms can jointly analyze all of the data, and based on training over millions of devices, determine what type of device is connected to the WiFi or Ethernet network.

Examples of the types of actions that the device firmware needs to take include

- blocking all traffic from or to a particular MAC address.
- blocking specific flows of traffic.
- applying prioritization or throughput reservations to a particular traffic flow, or all traffic from/to a particular MAC address.

Among the high-level SDKs, OpenSync is unusual in having built in all the measurements and controls that are required for device typing, flow identification, cyber-security, parental controls, access controls, and motion detection. It is also unusual in having the necessary fields for all of these measurements in its data model. In the other SDKs, support for these services requires installing an additional specific app onto the networking device in the home, if such an app is available at all. This is far less convenient than having the required functionality already integrated and tested, particularly since multiple apps of this sort can interact with each other in unpredictable ways.

OpenSync uses its SDN capability to perform many of these measurements, as well as to take many of the actions, such as blocking specific traffic flows. An advantage of this approach may be the flexibility provided by the SDN. Often, new services/applications can be rolled out with fewer modifications to the firmware on the device.

Among the high-level SDKs, OpenSync is unique in having built in all the measurements and controls that are required for device typing, flow identification, cyber-security, parental controls, access controls, and motion detection.

Conclusion

CSPs have a number of options for the software that they put on their GWs, routers, APs, extenders, and repeaters. The choice affects the networking and wireless capabilities, additional services, effort, schedule to launch new devices and services, and the selection and cost of hardware vendors. This paper has examined these options in detail, and the good news is that three options—RDK, OpenSync, and prpl—are all converging towards the same end goal, giving CSPs

- complete independence from all vendors at every layer of the stack.
- improved service delivery velocity.
- the ability to mix and match, and deliver tangible services to the end consumers from multiple vendors and innovators.

CSPs may not have the luxury of waiting for the complete standards to be established; the process is necessarily ongoing due to the constant advent of new technologies and solutions. Thus, the selection of one of the options depends on how fast a CSP wishes to go to market with new services and how well its timeframe aligns with the maturity of the options presented. Since CSPs have a real need for the technology now—and all approaches will naturally converge over time while continuing to evolve—CSPs should make a selection now and start implementing a cloud-based, multi-services management system

To learn more, email us at partner@plume.com or visit plume.com today.

Plume®