Episode Book

EPISODE #5544

Open Source Wi-Fi Controllers: Why Mixing AP Vendors Is So Hard

PUBLISHED	25 September 2026 at 09:49 UTC
DURATION	26:25
LISTEN	myweirdprompts.com/episode/open-source-wifi-ap-vendors/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:37 UTC

SYNOPSIS

Can any open source controller manage access points from different vendors? Daniel wrote in with a UniFi container, an old TP-Link AP, and six questions. We walk through what controller software actually does — adoption, VLANs, 802.11k/v/r roaming, RADIUS — and why vendors bundle hardware, firmware, and management into one locked stack. Then we survey what open source has actually built: OpenWISP, OpenSOHO, DAWN, TIP OpenWiFi, FreeSDN, and WiMoVE. The punchline: every one of them needs OpenWrt flashed or an open protocol on the AP. Stock UniFi does neither.

TRANSCRIPT



Corn

Here's a take you'll hear from every network forum on the internet: if you want real Wi-Fi, buy one brand, buy the controller, and never think about it again. Lock yourself in. It's the price of a network that just works.



Herman

And that advice is not wrong. It's just incomplete.



Corn

Right, because Daniel wrote in this week with a very specific itch. He runs a UniFi container for management, has an old TP-Link access point sitting around, and got to wondering whether any open source network manager can mix and match APs from different vendors. He suspects it isn't ideal, but he sees a case for decoupling access points from vendor-managed software entirely.



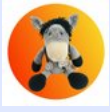
Herman

And then he asked the harder version of the question.



Corn

He did. What does the software side of AP control actually do in practice? Why do most people just use the vendor controller? Are there open source managers that let you mix vendors? What projects have actually sprung up to do this? Why is integrating different vendors' firmware so tricky? And is there a real case for wanting to decouple at all.



Herman

That's six questions in one paragraph.



Corn

Daniel doesn't do single questions.



Herman

No, he does not. Which is good, because the answer to the first one turns out to explain most of the others.



Corn

So let's start with what that controller software is actually doing when you're not looking at it.



Herman

The job description is longer than people think. Adoption and provisioning of access points, pushing configuration out, so SSIDs, VLANs, radio channels, transmit power. Coordinating roaming across the 802.11k, v and r standards. Firmware lifecycle management. Telemetry and monitoring. And RADIUS or 802.1X authentication for anything with a login.



Corn

The roaming piece is the one everybody actually cares about, and it's the one with the most mythology around it.



Herman

It is. The most explicit public enumeration of what a controller does on that front is in the TIP OpenWiFi documentation. Per-radio control of 802.11r Fast BSS Transition, 802.11k Radio Resource Management, 802.11v Network Assisted Roaming, plus client-steering thresholds under a component called uSteer, and channel provisioning under uChannel.



Corn

So that's the feature list. That's what you're buying when you buy into a vendor ecosystem.



Herman

And here's the tension underneath Daniel's whole question. Users want one SSID, seamless roaming, radio control across mixed hardware. Vendors want you to buy the hardware and the management software and the firmware as one bundle. Those two things are in direct conflict, and the vendors have spent fifteen years making sure their side wins.



Corn

Which is the arc here. What the controller does, why vendors lock it down, what open source has actually built, and then the part that I suspect is the real answer to Daniel's question.



Herman

Which is that the hard part isn't the software.



Corn

The hard part is the firmware.



Herman

There's a line from an explainer on 80211networks.com that puts the vendor model about as bluntly as it can be put. Traditional enterprise Wi-Fi is a vertically integrated stack. You buy access points from Vendor A, you run Vendor A's controller, Vendor A's management software, Vendor A's firmware. Their words: this is vendor lock-in by design.



Herman

It's accurate. Switching vendors doesn't mean swapping access points. It means ripping out the entire stack. The controller, the management layer, the configuration you've built up over years, the monitoring history, the integrations. All of it goes.



Corn

And the market data backs up how thoroughly that model has won. The managed service providers, the people who do this for a living across hundreds of client sites.



Herman

Seventy-two percent of MSPs rely on vendor-managed systems. Seventy-five percent say they rely heavily on vendor-specific management. Sixty-five percent say their current Wi-Fi platforms are too costly and too complex to maintain. And MSPs are spending somewhere between two hundred thousand and five hundred thousand dollars a year on software customization.



Corn

That last number is the tell. Half a million dollars a year to make vendor software fit the way they actually work.



Herman

Because the vendor software doesn't fit. It fits the vendor's idea of how a network should be run.



Corn

So why does anyone accept this? And I don't mean that rhetorically, because for most home users the answer is good.



Herman

The convenience is real. You plug in the access points, the controller adopts them, you get one SSID, roaming that mostly works, a polished web interface, a mobile app. Near-zero configuration. That's not inertia. That's a product that does what it says on the box.



Corn

The lock-in is the price of not having to think about it.



Herman

And for a lot of people that's a completely rational trade. The problem only shows up when you want to do something the vendor didn't plan for. Like Daniel, with an old TP-Link AP in a drawer and a UniFi controller already running.



Corn

So what actually exists? Because I went in expecting the answer to be nothing, and the answer is not nothing.



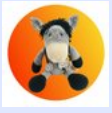
Herman

The answer is not nothing. Start with the most mature option, OpenWISP. It's a network and WiFi controller written in Python on Django. Provisioning, configuration management, firmware upgrades, an x509 public key infrastructure for certificates, RADIUS. Seven hundred and forty stars on GitHub, two hundred and eighty-seven forks, over a thousand commits. Latest release is 1.2.2, dated March of this year.



Corn

And the target hardware?



Herman

Mainly OpenWrt, but designed to work on other systems as well. That qualifier matters enormously and we'll come back to it.



Corn

What's the module list look like?



Herman

It's a proper ecosystem. openwisp-monitoring, openwisp-firmware-upgrader, openwisp-radius, openwisp-network-topology, openwisp-ipam. This is not a hobby script. This is a project that has been built out for years by people who take it seriously.



Corn

But there's a scale problem.



Herman

There's a scale problem. OpenWISP's own documentation says it may not be the best fit for very small networks, fewer than twenty devices. Which is a remarkable thing for a project to admit about itself.



Corn

And that admission created a project.



Herman

It did. OpenSOHO. Announced on the OpenWrt forum in August last year by a developer called rubenbe. Single Go binary. Targets two to twenty devices. And his description of it is about as charming as project descriptions get. Similar to Omada and the Ubiquiti controllers, but way simpler.



Corn

He said the quiet part out loud.



Herman

He was explicit about the lineage too. It's inspired by OpenWISP, but aims for networks which are too small to be maintained with OpenWISP. It reuses OpenWISP's daemons on the access points themselves, so it's not a fork, it's a lighter front end on the same foundation.



Corn

So you've got this fork in the road already, before we've even gotten to the interesting projects.



Herman

Enterprise-grade versus homelab-grade. OpenWISP is the proper tool that's too heavy for a house. OpenSOHO exists because the proper tool is too heavy for a house. That's a tension that runs through this entire space.



Corn

What's the third one?



Herman

DAWN. Decentralized WiFi Controller. And the architecture is the interesting part. It's peer-to-peer. There is no master node. Every access point runs a daemon, they find each other over multicast DNS, Bonjour, and they coordinate directly.



Corn

No controller at all.



Herman

Four hundred and sixty-eight stars, eighty-five forks, GPL-2.0, commits through August of this year. It does band-steering and load-balancing using 802.11k and 802.11v.



Corn

Give me the concrete version of what band-steering looks like.



Herman

A device is sitting on access point one at minus sixty-five dB. It won't roam until it's past minus seventy. But access point two can see it at minus fifty-five. DAWN notices the mismatch and tells the device to switch.



Corn

And the device can just say no.



Herman

The device can just say no. And we're going to come back to that, because it's the single most important fact in this entire episode.



Corn

Keep going. What's the enterprise answer?



Herman

TIP OpenWiFi. Telecom Infra Project, founded by Meta in 2016, five hundred plus member companies. This is the disaggregation play. It separates the access point hardware from the management software using a protocol called uCentral, which is JSON over WebSocket.



Corn

So the AP speaks a standard protocol and any controller that speaks the same protocol can manage it.



Herman

That's the idea. Certified vendors include Edgecore, Actiontec, CyberTan, Asterfusion. The APs run OpenWrt-based firmware, and a common cloud controller manages all of them. Over a hundred thousand deployed devices globally.



Corn

That's not a hobby project either.



Herman

That's a serious industry effort. And it's the closest thing to the Android model that anyone has actually shipped.



Corn

Which brings us to FreeSDN, which I think is the one Daniel will find most interesting.



Herman

FreeSDN is the closest thing to true mix-and-match. Version 26.09.0, AGPL licensed, thirteen vendor adapters including TP-Link Omada, Ubiquiti UniFi, MikroTik, and OpenWrt. Requires four gigabytes of RAM minimum.



Corn

And the architecture is the clever bit.



Herman

It is. And I want to quote their own description because it's precise. It does not replace your vendors' native controllers. It sits in front of them and normalizes their APIs.



Corn

So it's a translation layer.



Herman

It's a translation layer. You keep running your UniFi controller and your Omada controller, and FreeSDN talks to both of them and presents one interface. Which is a different approach from everything else we've talked about.



Corn

And the catch?



Herman

Their own disclaimer. FreeSDN is under active development and is currently intended for evaluation, testing, and homelab use. It is not yet recommended for production.



Corn

That's the honest-project problem in one sentence.



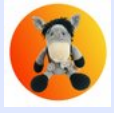
Herman

It is. And there's one more worth mentioning, more for the academic angle than anything practical. WiMoVE. It uses BGP EVPN and VXLAN overlays on OpenWrt access points to handle mobility. Their claim is that it allows for using commodity access points running OpenWrt for large-scale Wi-Fi deployments, even from different vendors.



Corn

Twelve stars on GitHub.



Herman

Twelve stars. MIT licensed, created in 2023, tested on a ZyXEL NWA50AX and a Linksys WRT1900ACS. It's a research project. But the approach is interesting because it treats mobility as a routing problem rather than a wireless problem.



Corn

So that's the survey. Six projects, all real, all doing something. Now the part I actually want to get into, which is why none of them fully solve Daniel's problem.



Herman

Because the firmware is closed.



Corn

Say more.



Herman

Every single project we just listed requires one of two things. Either you flash OpenWrt onto the access point, or the vendor ships firmware that speaks an open protocol. There is no third option.



Corn

And UniFi access points don't do either.



Herman

UniFi access points don't do either. You cannot make a stock UniFi AP talk to an OpenWISP controller. The firmware is closed and it speaks a proprietary protocol. Full stop.



Corn

So when Daniel asks whether he can manage his UniFi gear and his old TP-Link AP under one open controller, the honest answer is no.



Herman

And I want to be precise about this, because I went looking for a counterexample and I couldn't find one. No single open source controller manages stock, unmodified access points from arbitrary vendors. FreeSDN is the closest, and it works by wrapping each vendor's proprietary API rather than replacing the firmware. And it's pre-production.



Corn

So mix-and-match in practice means either flash everything with OpenWrt, or buy new certified hardware.



Herman

Which is where the Android analogy starts to strain. The comparison from 80211networks.com is that OpenWiFi is to access points what Android is to phones. A standardized software stack running on hardware from many manufacturers. And that's a fair description of what OpenWiFi is.



Corn

But Android runs on phones you already own.



Herman

OpenWiFi only works with certified hardware. You cannot retrofit it onto the UniFi APs you already have mounted on your ceiling. So mix-and-match in practice means buy new certified hardware. Which is a much weaker pitch than it sounds when you first hear the phrase.



Corn

Hm.



Herman

There's a second problem, and it's the one I keep coming back to. Roaming is a client decision. Not a controller decision.



Corn

Explain that, because I think most people assume the opposite.



Herman

Most people absolutely assume the opposite. There's an OpenWrt forum contributor called psherman who put it as plainly as it can be put. Roaming is client side decision. You do not need any AP side instrumentation for clients to roam. For any roaming you have to set same name, plus password, plus DHCP subnet for your whole coverage area.



Corn

That's it. That's the whole requirement.



Herman

Everything else is a nudge. And DAWN's own README says the same thing from the other direction. It is ultimately for a client device to decide which AP it will connect to, so it may ignore DAWN's proposal. Especially if it is using a significantly different algorithm to DAWN.



Corn

So all the vendor seamless roaming magic is...



Herman

Nudging. 802.11k tells the client what other access points exist. 802.11v suggests it might want to move. 802.11r makes the handoff faster once the client has decided. But the decision is the client's, and a client that wants to stay on the access point in the far bedroom will stay there until the signal dies.



Corn

Which means a better controller does not solve roaming.



Herman

A better controller does not solve roaming. It improves the odds. That's the honest version.



Corn

And 802.11r, the fast handoff piece, is the one that's supposed to make this seamless.



Herman

And it's broken in a lot of OpenWrt and WPA3 combinations. There's an open issue on the OpenWrt repository, number 9181, titled broken 802.11r roaming if 802.11w is enabled, which covers WPA3 and WPA2 with 802.11w. There's a forum thread where clients get locked out after a failed roam between a Banana Pi BPI-R4 and a GL-MT6000. There's another where a user mapped the entire house to read dBm levels and still couldn't get roaming working across OpenWrt and Zyxel gear.



Corn

He mapped the house.



Herman

He mapped the entire house. And it still didn't work.



Corn

There's something almost noble about that.



Herman

There's something very OpenWrt about that. And the configuration burden historically was real. 802.11r used to require manually listing r0kh and r1kh RADIUS keys across every single access point. Newer builds added a setting to auto-generate them, but that's a recent convenience on top of years of pain.



Corn

So even within OpenWrt, where you control the firmware, roaming is fragile.



Herman

Even within OpenWrt. Which is worth sitting with, because OpenWrt is the best case. It's the case where you have full control, open firmware, open tooling, and a community that has been working on this for a decade. And it's still fragile.



Corn

So let's get to the honest-project problem properly, because I think it's the most revealing thing in all of this.



Herman

FreeSDN labels its own OpenWrt support as preview. Speaks ubus and UCI, not yet audited. Its UniFi adapter is REST only, with writes described as staged and dual-gated.



Corn

Staged and dual-gated.



Herman

Even a purpose-built multi-vendor tool, built by people who clearly understand the problem, treats each vendor as a bespoke integration. Because that's what it is. Every vendor is its own little world with its own API, its own quirks, its own failure modes.



Corn

Compare that to the vendor marketing, which is all seamless and unified and effortless.



Herman

The gap between those two registers is the whole story of this space.



Corn

So is there a case for decoupling? Because Daniel thinks there is, and I think he's right.



Herman

He's right, and the case is stronger than the technical situation suggests. Start with cost and control. There's a line from a piece on goingo.au that I think is the best single sentence about this whole topic. With a locked cloud system, the access points only work while the vendor's subscription is paid, and you cannot move them to another platform. With open WiFi, you own the hardware outright. Meraki is the extreme case. Annual per-device licensing. Stop paying and the hardware you bought becomes a paperweight. It's not a hypothetical, it's a business model.



Corn

And then there's vendor risk, which is the slower version of the same problem.



Herman

If Vendor A raises licence fees, changes their cloud architecture, acquires and sunsets a product line, your options are limited. Pay up, or rip out and replace the hardware. And the second option is expensive enough that most people pay up.



Corn

That's the lock working as designed.



Herman

And there's a line from goingo.au that puts the philosophical version of it better than I can. The difference between open and locked WiFi is not a feature. It is who owns the network you sell.



Corn

Which is a very different argument from the technical one. It's not about whether open source does it better. It's about who holds the keys.



Herman

Right. And the technical situation is worse than the argument. That's the frustrating part. The case for decoupling is strong, and the practical path is narrow.



Corn

Narrow but not closed, though.



Herman

Narrow but not closed. If you're willing to flash OpenWrt, OpenWISP and OpenSOHO and DAWN are all real options. If you're willing to buy certified hardware, OpenWiFi is a real option. If you're willing to run pre-production software, FreeSDN is a real option. Each one requires giving something up.



Corn

And the philosophical fork between OpenWISP and OpenSOHO is interesting on its own. Enterprise-grade versus homelab-grade.



Herman

It mirrors the tension everywhere in open source. The proper tool is built for scale and it's too heavy for a house. So somebody builds a lighter version, and now you have two projects, and the lighter one doesn't do everything the heavy one does, and the heavy one doesn't fit where the light one lives.



Corn

The eternal fork.



Herman

And DAWN is the one I'd actually watch, because the architecture is different. No master node. Every access point is a peer. It sidesteps the what if the controller dies problem entirely, because there is no controller to die.



Corn

Is peer-to-peer the future of home WLAN management?



Herman

I don't know. Honestly. It's elegant and it removes a single point of failure, but it also means no central place to look at the whole network, and no central place to enforce policy. For a house, that's fine. For anything bigger, I'm not sure.



Corn

I was going to say. The thing I keep circling back to is that the controller was never the hard part.



Herman

The controller was never the hard part. The controller is software. Software can be rewritten. The hard part is that the radio firmware in the access point is the vendor's, it's closed, and no amount of open source management software changes that.



Corn

So Daniel's old TP-Link AP and his UniFi gear are never going to live under one open roof.



Herman

Not as they are. Flash the TP-Link with OpenWrt and it joins the OpenWISP or OpenSOHO world. Leave it stock and it lives in TP-Link's world. The firmware decides which world you're in.



Corn

And the UniFi stays in UniFi's world regardless.



Herman

The UniFi stays in UniFi's world regardless, because you can't flash it and it doesn't speak an open protocol. That's the wall.



Corn

The answer to Daniel's question is a qualified no with a lot of interesting qualifications.



Herman

That's the honest summary. No single open manager handles stock hardware from arbitrary vendors. But there are six real projects doing real work, and the reasons they can't fully solve it are more interesting than the answer itself.



Corn

There's one more thing I want to get on the table before we wrap, and it's about the gap between the technical problem and the actual problem.



Herman

Go ahead.



Corn

Because I think you can spend a year reading about OpenWISP and DAWN and OpenWiFi and come away thinking the obstacle is technical. And I'm not sure that's where the obstacle actually lives.



Hilbert

The closet had eleven access points in it and four of them worked.



Corn

Four out of eleven.



Hilbert

This was a regional ISP, rural wireless installs, and procurement bought whatever was on sale that quarter. Three vendors by the time I got there. Two of them had been discontinued. I spent about three weeks trying to get all of them broadcasting the same SSID and handing off cleanly.



Herman

And did they?



Hilbert

Eventually. Same SSID, same password, same subnet. That was the whole trick. Clients roamed when they felt like it. We stopped trying to make them.



Corn

The controller wasn't doing the roaming.



Hilbert

The controller was doing the billing integration and the truck roll scheduling. That's what it was for. The wireless part we handled by putting up so many access points that nobody was ever far from one. Over-provision it and the roaming question mostly stops mattering.



Herman

That's a real answer, actually. Density instead of coordination.



Hilbert

It worked. It was expensive and it worked. And then the owner standardized on one vendor the following spring, and it wasn't because the technology was better.



Corn

What was it?



Hilbert

His brother-in-law got a reseller deal.



Corn

Of course.



Hilbert

The whole fleet went one way in about four months. Nobody asked me. I found out when the first pallet arrived.



Herman

The lock-in decision was downstream of a family relationship.



Hilbert

Most of them are. You'd be surprised how few of these decisions get made in a room with an engineer in it. It's procurement, it's relationships, it's whoever the owner had lunch with. The technical argument is usually made afterward, to justify what somebody already decided.



Corn

Which means the open source projects are solving a problem a lot of deployments don't experience as a problem.



Hilbert

They experience it as a bill. The bill is the problem. And if the bill is fine, the lock isn't a lock. It's just how the network is.



Herman

That reframes the whole thing. The lock is only a lock if you want out.



Hilbert

And most people don't want out. They want the network to work and the invoice to be predictable. The vendor sells both of those things.



Corn

The market for OpenWISP is people who specifically want out.



Hilbert

It's a small market. It's a real market. It's just small.



Herman

Which is why the projects are small.



Hilbert

Why the ones that get big get big by selling to people who want to buy, not to people who want to leave. That's a different business.



Corn

The brother-in-law deal is going to stay with me.



Hilbert

It stayed with me too. I still have the pallet paperwork somewhere. They don't make those forms for anteaters, by the way. The signature line is about this wide.



Herman

The practical lesson is that vendor lock-in is often not a technical decision at all.



Corn

It's procurement. It's relationships. It's inertia.



Herman

The open source projects are technically solving a problem that many organizations have already decided not to have.



Corn

Which is a much more useful way to think about Daniel's question than a feature comparison.



Herman

It is. Because the feature comparison tells you what's possible. Hilbert's story tells you what actually happens.



Corn

Where does that leave the future of this? Because I don't think it's settled.



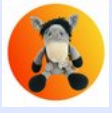
Herman

Two paths, and I don't know which one wins. One is peer-to-peer, DAWN-style, no controller at all, every access point a peer. The other is disaggregated, OpenWiFi-style, where the hardware and software are separated by a standard protocol and you buy certified hardware from whoever.



Corn

The third path is the status quo, which is winning by default.



Herman

The trend toward software controllers, the UniFi container, the Omada software controller, that makes the management layer more portable. But the firmware layer is still the lock. You can move the controller to a container. You can't move the firmware out of the vendor's hands.



Corn

The Android analogy might eventually hold.



Herman

It might. But only if vendors ship open firmware, and there's very little incentive for them to do so. The firmware is the moat. Everything else is downstream of it.



Corn

The controller was never the hard part.



Herman

The hard part is that the radio firmware belongs to the vendor, and no amount of open source management software changes that.



Corn

That's the answer to Daniel's question. A qualified no, with six real projects and one very good reason.



Herman

A pallet of discontinued access points in a closet somewhere.



Corn

Thanks to Hilbert Flumingtop for producing. This has been My Weird Prompts. If you want to send us a prompt, email us at show at my weird prompts dot com. We'll be back soon.