Episode Book

EPISODE #5748

What's Actually Inside an eSIM Adapter

PUBLISHED	6 October 2026 at 18:17 UTC
DURATION	24:51
LISTEN	myweirdprompts.com/episode/esim-adapter-euicc-open-source/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
BOOK GENERATED	19 October 2026 at 17:39 UTC

SYNOPSIS

Most eSIM coverage focuses on carriers and roaming revenue, but the real story sits at the bottom of the stack: a commodity secure element, a GSMA-rooted certificate, and an LPA that moves credentials around. This episode traces that stack from silicon to software — what an eUICC actually is, why certified cards cost under 25 euros while consumer adapters sell for multiples of that, and where the genuine lock-in lives (spoiler: it's the certificate, not the plastic). We also walk through the open source LPA landscape, including lpac, OpenEUICC, EasyEUICC, and MiniLPA, and what it takes to provision a profile onto a router or failover device that will never run Android. Plus: the 5ber cautionary tale, and why a vendor advertising its own dispensability might be the healthiest posture in the market.

TRANSCRIPT



Corn

Most coverage of eSIM treats it as a carrier story. Big networks, dumb pipes, who loses the roaming revenue. That framing is backwards.



Herman

The interesting part is at the bottom of the stack. It's a chip and a certificate.



Corn

Which is why Daniel's question this week is the right one to ask. Here's what he wrote in. He wants to know what's actually under the hood of eSIM, because if you're trying to run one on legacy hardware that doesn't support it natively, you're stuck with a physical adapter. And he's long suspected those adapters are overpriced for what they are, since the thing they're really doing is putting a eUICC chip into a SIM form factor.



Herman

He's not wrong.



Corn

He also wants to know the closest thing to a generic eUICC chip you can just buy in a SIM shell, what the most open source options are for writing and managing profiles, and how you get one of these onto a router or a failover device that will never run Android. His instinct is that you provision the profile somewhere else and then move the card over. And his framing at the end, which I like, is that the lock-in isn't the hardware. It's vendors whose main value is locking you into their walled garden, with incentives that don't point where you'd hope.



Herman

The easy-installation pitch is real, for people who don't want to think about any of this.



Corn

Right. So today we're going to trace the stack from the silicon up to the software, and see where the open source path actually leads.



Herman

Start with vocabulary, because the acronyms are where people get lost. eUICC is the chip. It's a physical piece of silicon, a secure element, and it's the thing that holds eSIM profiles. An eSIM profile is not a chip. It's a logical subscription, a bundle of credentials and operator data, that gets installed onto that chip. In a phone, the eUICC is soldered to the board. You never see it. But the same silicon exists in plastic, in the 2FF, 3FF, 4FF form factors, so you can drop it into a regular SIM slot on a device that has no eUICC of its own.



Corn

So the adapter is not converting anything. It's not translating between two systems. It's just a eUICC that happens to be shaped like a SIM card.



Herman

That's the whole trick. And the second layer is RSP, Remote SIM Provisioning. That's the GSMA protocol that delivers a profile over the air. Consumer devices run SGP.22. Machine-to-machine runs SGP.02. There's a newer IoT framework, SGP.32, that's meant to bridge the two worlds.



Corn

And the third piece?



Herman

The LPA. Local Profile Assistant. That's the software on the device that talks to the eUICC on one side and to the operator's server on the other. It's what downloads, enables, disables, deletes a profile. When you tap through a carrier app to install an eSIM, you're driving an LPA without knowing it.



Corn

So we've got a commodity chip, a standardized protocol, and a piece of software that mostly exists to move credentials around. Which raises the question of where the money actually is.



Herman

It's not in the chip. The eUICC is commodity silicon. The gatekeeper is the certificate. Every eUICC that can install a real operator profile carries a GSMA-rooted certificate, and that root of trust is issued under GSMA rules. You cannot self-issue one. That's the moat.



Corn

And that's where the lock-in lives. Not in the plastic.



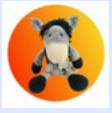
Herman

Exactly where Daniel suspected, though he framed it as the vendor's walled garden. The walled garden is downstream of the certificate.



Corn

Let's do the pricing, because Daniel's thesis is specific. He thinks the adapters are overpriced.



Herman

He's right, and here's the number that makes the case. sysmocom, the German company, sells a GSMA-certified eUICC in SIM card form factor, the sysmoEUICC1-C2G, for twenty-three euros eighty including VAT. That is a production card, with production certificates. It can install profiles from essentially any operator.



Corn

Twenty-three euros for the real thing.



Herman

They also sell a version with test certificates instead of production ones, the C2T, and that one costs ninety-five euros twenty. Which sounds backwards until you realize test certificates are rarer. Fewer of them get issued, so they cost more. And there's a solder-down MFF2 variant, ten chips for two hundred ninety-seven euros fifty, if you're building something rather than slotting something in.



Corn

So the certified silicon, the actual hard part, is under twenty-five euros. Which means when a consumer card costs three times that, you're paying for something else.



Herman

Storage, branding, and the app. The 9eSIM line is the clearest example. The V0 holds four hundred eighty kilobytes, which is around fifteen profiles. The V0 Max is one point one megabytes, about thirty profiles. The V3 is one and a half megabytes, up to fifty profiles, running the Kigen KG operating system.



Corn

And the app is the part they're charging for.



Herman

The app is the convenience layer. The card itself is a standards-compliant eUICC. There's nothing exotic in it.



Corn

Which means the honest description of an eSIM adapter is: a chip you can buy for twenty-odd euros, in a plastic shell, with a logo on it.



Herman

And a support contract you may or may not want.



Corn

So what's actually on the card? If I buy the sysmocom card, what am I holding?



Herman

A secure element with a certificate chain that terminates at a GSMA root. The community maintains a list, the osmocom eUICC Developer Manual has a page called Known eUICC Cards, and it's the canonical reference for which cards carry which root certificates. That page matters more than it sounds like it does, because the official verification path is not friendly.



Corn

We'll come back to that. What does the software side look like? Because the chip is only half the story.



Herman

The LPA layer is where open source has won. The one to know is lpac, from the estkme group. It's written in C, it's cross-platform, it's compatible with SGP.22 version two point two point two, and it's AGPL licensed. Around six hundred seventy-five stars on GitHub, a hundred seventy-two forks. It's in Debian, so you can install it with a single apt command.



Corn

Which is a different universe from sideloading a vendor APK.



Herman

Considerably. And the reason lpac travels so well is its backends. It doesn't assume it's talking to a card through a phone. It has pluggable APDU backends. PCSC for a card reader, MBIM and QMI for talking through a modem, AT commands, gbinder on Android. And on the network side it has pluggable HTTP backends, curl or winhttp. You pick them



Herman (continued)

with environment variables and it just works.



Corn

That's the design decision that makes the whole thing portable.



Herman

It's the reason you can drive an eUICC inside a router from a Linux box without any Android anywhere in the picture.



Corn

What else is out there?



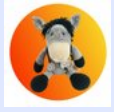
Herman

OpenEUICC, from Peter Cai, is the fully free Android LPA, GPL licensed. It has to be a system app, signed with the platform certificate. EasyEUICC is the sibling that doesn't need root. It's a user app, and it works with removable cards whose ARA-M field has been set to trust its certificate hash. EasyLPAC is a graphical front end for lpac that runs on Windows, Linux and macOS. MiniLPA is a desktop LPA built on PCSC. NekokoLPA is an Android LPA, on the Play Store and on GitHub. There's lpa-gtk, which is a GTK4 front end aimed at Linux phones. euicc-go is a Go library for the RSP side. There's a Rust project, simrs, that simulates SIM and eUICC and parses profile packages. And there's an odd one, OpenRSP, which is a blockchain-flavored attempt at an open provisioning system.



Corn

Blockchain enters the chat.



Herman

It does that.



Corn

So the reader can pick their poison. Phone, desktop, Linux mobile, or a Go library if they're building.



Herman

The interesting thing is the convergence. EasyEUICC works with removable cards because the card vendors ship them with a hash that trusts it. The hash is two A two F A eight seven eight B C seven C three three five four C two C F eight two nine three five A five nine four five A three E D A E four A F A. That's a certificate thumbprint, and it's the entire mechanism by which a card vendor says "this open source app is allowed to manage me."



Corn

Which is a real act of trust. The vendor is giving up the ability to be the only one who can touch the card.



Herman

It's a choice, and it's reversible, which is the point. Now, to Neil Brown's writeup from January of last year, because he documented the practical path better than anyone. He bought a 9eSIM card, a USB reader, and an adapter, about thirty pounds delivered. He provisioned the card using lpac and EasyLPAC, first on Android, then on Debian Linux.



Corn

And the caveat he ran into?



Herman

He could not write profiles to the card while it was in his laptop's WWAN slot. He tried. He couldn't find a way. The workflow that worked was: pull the card, put it in the reader, provision from the machine, put it back.



Corn

So for a router, the same rule applies.



Herman

For most setups, yes. You provision elsewhere and you insert. But there is a path that avoids the round trip, and it's lpac over MBIM or QMI. Those backends are how you talk to an eUICC through the modem itself, from Linux. There used to be a wrapper project, lpac-libmbim-wrapper, that added MBIM support, and then MBIM support was merged upstream and the wrapper's own readme now says it's not needed anymore.



Corn

So the in-situ path exists. It's just less documented than the pull-the-card path.



Herman

Much less documented. And 9eSIM's own card reader page markets the reader for exactly the pull-the-card case. Their copy says it loads new profiles without putting the card in your phone, handy when the card lives in a fixed device, a router, an IoT gateway. They're selling you the friction.



Corn

Which brings us to the thing Daniel actually cares about. The lock-in, and the incentives.



Herman

The cautionary tale is 5ber. 5ber was a consumer removable eUICC, popular, sold with its own app. Its parent company, iFree Group, ran out of funding in late 2024. The app went down. The cloud went down. And here's the part that matters: the cards themselves were fine. The hardware was fine. But the only officially supported way to manage them had disappeared.



Corn

So what happened to people who owned one?



Herman

The cards stayed usable, but only because the community LPA tools could talk to them, and only if you had a reader. The pre-paid download credits people had bought became worthless. 9eSIM wrote a retrospective about it, and their line is sharp. Most physical eSIM cards only work with their maker's app. If the company shuts down, you can't download new profiles on the phone, and the brand premium and any prepaid downloads turn into dead weight.



Corn

Dead weight is the right phrase. The silicon was fine. The business wasn't.



Herman

And that's the argument for open standards in one paragraph. The card didn't die. The permission structure around it did.



Corn

So 5ber is what happens when convenience is the only interface.



Herman

Right. And 9eSIM's counter-pitch is explicit. Their cards work with any open source LPA, NekokoLPA, OpenEUICC, MiniLPA, over the public eUICC standard, with no proprietary cloud in the loop. Their own wording is: if we ever stopped developing our own app tomorrow, your card would keep working with whichever community LPA you prefer.



Corn

A vendor selling convenience while advertising its own dispensability. That's a strange posture.



Herman

It's a bet on the standard outlasting the company. Which is the correct bet, historically.



Corn

But here's where I want to push, because there's a detail in Neil Brown's chip info output that undercuts the whole story slightly. What did his card report?



Herman

The default SM-DP+ address on the card pointed at Kigen infrastructure, at smdp-plus-zero dot eu dot cd dot rsp dot kigen dot com, with the root discovery service at lpa dot ds dot gsma dot com.



Corn

So even the open card, managed by the community LPA, on hardware the vendor says is yours, depends on GSMA and Kigen infrastructure for the actual download of a profile.



Herman

It has to. That's the protocol.



Corn

Then the independence is partial. You don't need the vendor's app. You do still need the GSMA root and the operator-side servers.



Herman

That's the honest boundary. The open source ecosystem reaches the LPA layer. It does not reach the root of trust. You can manage your card however you like, but you cannot conjure a profile out of the air. The profile comes from an operator, through a provisioning server, and that server is part of the GSMA system.



Corn

Which is why Daniel's phrase, the walled garden with dubious incentives, is worth unpacking. The incentive a vendor has is to be the only path to your card. The incentive an operator has is to be the only path to your profile. Two different walls.



Herman

Two different walls, and only one of them has been knocked down by open source. The vendor wall is mostly down. The operator wall is not down at all.



Corn

Then let me put the practical question to you. If I've got a mobile router, no Android, no vendor app, and I want an eSIM profile on it, what does the workflow actually look like, end to end?



Herman

You buy a certified removable eUICC. A sysmocom card, or a 9eSIM, or an ESTKme. You buy a USB reader, and an adapter if you need a smaller form factor. You install lpac on a laptop, or EasyLPAC if you want a graphical front end. You put the card in the reader. You pull an activation code from your operator, feed it to lpac, and it downloads and installs the profile onto the chip. You list the profiles, you enable the one you want, you set a nickname so future-you remembers which is which. Then you pull the card out of the reader and put it in the router.



Corn

And the router never knows any of this happened.



Herman

The router thinks it's a SIM card. That's the entire elegance. The device sees a standard UICC and talks to it over the standard interface. The eUICC inside is doing all the eSIM work, and the device is none the wiser.



Corn

What about updates? Changing operators?



Herman

Same round trip, unless you've got the MBIM or QMI path working through the router's own modem. If you do, you can drive lpac against the card in place. If you don't, you pull the card, which for a router in a cabinet is a physical inconvenience rather than a technical barrier.



Corn

A physical inconvenience is a lot of the real world of infrastructure.



Herman

It's most of it.



Corn

Let me come back to the certificate, because that's the load-bearing claim and I want it precise. What exactly does it let you do that a card without it can't?



Herman

The eUICC presents a certificate chain to the operator's provisioning server. That server validates the chain against the GSMA root. If it checks out, the server releases the profile. If not, the download fails. A card with a test certificate can only talk to test infrastructure. It cannot install a real production profile.



Corn

So a card without production certificates is, for the purpose of actually getting service, a very thin paperweight.



Herman

It's a development tool, which is what the SGP.26 test certificates are for. Developers use them to exercise RSP flows without touching a real operator. The mistake is buying test-certified hardware thinking it's a bargain.



Corn

Has that happened?



Herman

It has a long history in the SIM world. Cards get sold as certified when they aren't, or when the certificate is expired, or when it's a test credential with the GSMA logo printed on the plastic.



Corn

The logo is ink.



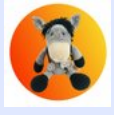
Herman

The certificate is what's checked, and it's checked electronically, at provisioning time, by the operator's server. Nothing about the printing on the card affects whether the download succeeds.



Corn

So how does a buyer check before they've paid?



Herman

The honest answer is that it's harder than it should be. The community list on the osmocom manual is the best starting point, because it maps cards to root certificates based on what people have actually read off the hardware. The GSMA runs a registry of certified products, and you can check an EID against it in principle, but it's not always current, and it's not designed for a consumer standing in a checkout flow.



Corn

Which is a gap. There's no clean verification path for the person buying one card.



Herman

There isn't. You rely on the vendor's reputation and the community's record. Which is uncomfortable, and it's the exact kind of thing the open source ecosystem exists to fill.



Corn

It also means a counterfeit or a mislabeled card can survive in the market for a long time before anyone notices.



Herman

Especially at the cheap end, where the buyer is least equipped to check.



Corn

So the stack is open all the way up to the LPA, and closed at the root. What's the realistic ceiling of the open source approach, then?



Herman

It gets you everything except the issuance of credentials. You can own the card, manage the card, choose your tooling, and never depend on a vendor's app. You cannot issue yourself a GSMA-rooted certificate, and until that changes, provisioning runs through the GSMA system.



Herman

And there's a practical dependency people miss. The card's default discovery service, the thing it uses to find the right provisioning server, is GSMA's, and the European servers for several card lines run on Kigen infrastructure. If Kigen changed something at the top, that would ripple down to a lot of people who thought their setup was fully independent.



Corn

Including the people who bought specifically to avoid that.



Herman

Including them.



Corn

Hold on, let's put a number on the profile side, because the storage math matters for someone deciding what to buy. If the V0 holds around fifteen profiles and the V3 holds up to fifty, what's actually in a profile?



Herman

Credentials, operator-specific data, applets, and the file structure the network expects. The sizes vary by operator, but the practical ranges are what 9eSIM publishes. Fifteen, thirty, fifty. Neil Brown's V3 reported about one and a half megabytes of free non-volatile memory and a profile version of two point three point one.



Corn

Which is a lot of headroom, unless you're juggling operators the way some people do.



Herman

And most people aren't. For the typical use case, a router fails over to a second operator when the primary drops. That's two profiles. The storage question is mostly a marketing axis.



Corn

The silicon doesn't care how many profiles you keep. It cares whether the certificate checks out when you ask for a new one.



Herman

That's the whole game.



Corn

So I want to state the misconception directly, because I think most people carry it. The common belief is that the GSMA logo on a card means the card is certified.



Herman

It means nothing of the kind. The logo is printed on plastic. Certification is a certificate chain inside the chip, validated by the provisioning server at the moment of download. A card can carry the logo and fail to install a real profile, and a card with no logo at all can work perfectly if its certificate chain is valid.



Corn

The only way to know is to check the EID against a registry, or to trust the community list, or to try it and watch the download fail.



Herman

Which is a terrible verification story for a thing you buy with money.



Corn

It is. So let's imagine how this actually goes wrong in practice.



Hilbert

You're right that the certificate is the moat. I'll tell you how the moat gets crossed anyway.



Herman

Go on.



Hilbert

I was buying eUICC cards for a client in the mid eighties, and a supplier offered me a batch of generic cards, printed with the GSMA logo, at a price that made the certified ones look like a mistake. The logo was the whole pitch. He had a sheet of paper that said certified, which meant nothing, and the price, which meant less than nothing. I bought a hundred of them.



Corn

And?



Hilbert

They had test certificates. Every one of them. We tried to provision a real operator profile in front of the client, and the server rejected the chain. The download failed on the first card, and on the second, and on all of them. I had a hundred cards, a client watching, and a supplier who stopped answering the phone. I paid for the lot. The GSMA logo was ink. I knew it was ink. I still bought them.



Herman

So the piece of paper was the whole verification.



Hilbert

The piece of paper, the logo, and the price. Three things that felt like verification and weren't. The only thing that would have told me was reading the EID and checking it against the registry, and even then the registry was behind. My cousin worked at the card bureau, she was the one who told me the cards were test stock, and she said don't quote her, because she wasn't supposed to have looked.



Corn

The verification gap isn't theoretical.



Hilbert

It's the whole market at the bottom. That's why the community list exists, and why it's a list of what people have actually read off the hardware, not what the seller claimed. When the official path is slow, people build a faster one.



Herman

That's the practical version of what we've been circling.



Corn

Which tells me the buyer's real defense is the vendor's reputation, and second, a reader and a tool like lpac, so you can check the chip yourself before the client sees it.



Hilbert

Before the client sees it. That's the rule. You check it in the reader, on your own bench, before it goes anywhere near the thing it's supposed to run.



Corn

The open source tooling isn't just about convenience. It's also the only inspection path most people have.



Corn

The thing I keep coming back to is that the open ecosystem solved the problem it could solve, and the problem it couldn't solve is the one that bites you at the worst moment.



Herman

Which is exactly what Hilbert's story shows. The tooling on the bench is the difference between knowing and hoping.



Corn

The forward-looking question is this. The GSMA registry exists, it's just not built for the person buying one card. If the community list keeps growing, and lpac keeps spreading, there's an argument that verification becomes a commodity too. But that only works if the GSMA registry stays current, or if the community list gets good enough to be trusted the way the official one should be.



Herman

If Kigen changes the discovery infrastructure, the whole thing shakes in ways nobody's modeled.



Corn

Which is where we leave it. Daniel's path works. It's just not free of the root of trust, and it never will be, until someone decides to open it.



Herman

Thanks as always to Hilbert Flumingtop for producing the show.



Corn

This has been My Weird Prompts, the human-AI collaboration podcast. If you want to dig into the technical details, the osmocom eUICC Developer Manual and the lpac GitHub repo are the best starting points. And if you've ever been burned by a certified eUICC that wasn't, we'd love to hear about it. Email us at show at my weird prompts dot com.



Corn

We'll be back soon.