Episode Book

EPISODE #5525

Why Does an eSIM Adapter Need a Card Reader?

PUBLISHED	24 September 2026 at 14:26 UTC
DURATION	24:54
LISTEN	myweirdprompts.com/episode/esim-adapter-card-reader-explained/
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

A listener wrote in with three questions about his 9eSIM adapter: why the company sells a physical USB reader when his phone can already talk to the card, whether he's paying a performance penalty versus a native eSIM phone, and whether two adapters can run in a dual-SIM handset. This episode works through all three. The reader turns out to be a platform workaround, not a technical necessity — iOS and HarmonyOS NEXT forbid third-party apps from writing to a physical SIM, older Android can't reach the card at all, and fixed devices like routers have no UI to run an app on. Along the way: the OEM bugs that get blamed on the card, the OnePlus RIL limitation that has no userspace fix, and an honest admission that nobody has actually benchmarked adapter versus native eSIM latency.

TRANSCRIPT



Corn

I want to start with the part of Daniel's message that made me put my leaf down mid-chew.



Herman

Which part?



Corn

The part where he says he uses one of these adapters, has no complaints, and still can't work out why the company sells a physical reader.



Herman

Ah. Yes. That's a good question, actually.



Corn

It's a great question. Here's how he put it. He's got a 9eSIM adapter sitting in a OnePlus, it works, he's happy with it, but he's got three question marks he wants ironed out. First one. The company sells a physical reader, a little USB thing. But you can also talk to the card through the phone itself, through ADB, through the app. So what does the reader actually do that the phone can't? And he says, and I'm quoting him here, "it doesn't really make sense to me that you need a physical hardware reader writer when the device the eSIM exists within can both read and write."



Herman

Right.



Corn

Second question. Compatibility and performance. He's had no trouble on the OnePlus, but his device has no native eSIM, so he's got nothing to benchmark against. He wants to know if he's paying some latency penalty versus someone on a proper eSIM phone. And then the bigger version of that, are there capabilities native eSIM delivers that an adapter just can't?



Herman

That's the one I want to get to.



Corn

And third. Can you run two of these adapters in a dual physical SIM handset? He wants it kept Android-focused, so that's the lane we're staying in.



Herman

Three good questions, and the first one has a satisfying answer, because Daniel's intuition is correct.



Corn

Let's start with the question that sounds almost too obvious to ask. Why does the reader exist at all?



Herman

So first, let's define the thing, because "physical to eSIM adapter" covers a few products that aren't identical. What we're talking about is a removable eUICC in a SIM card form factor. It's a real smart card. It's got a chip, it's got storage, it's got an operating system on it, and it sits in your physical SIM tray. 9eSIM is one brand. There's also eSIM.me, 5ber, Switch by Plan B, Xesim. They're all the same category.



Corn

And mechanically it's just a SIM.



Herman

Mechanically it's just a SIM. Your modem talks to it exactly the way it talks to a normal SIM card, because it is one. The difference is what's on the chip. Instead of a single carrier profile burned in at the factory, it holds a little file system of downloadable profiles, and an app on your phone can write new ones to it.



Corn

How does the app talk to it?



Herman

Over OMAPI. Open Mobile API. It's been in AOSP since Android 9. The app opens a logical channel to the card and sends it APDU commands, which is the ISO-7816-4 command set that smart cards have used forever. Or you go through TelephonyManager. Either way, you're talking to a card in a slot.



Corn

So from Android's point of view, nothing unusual is happening.



Herman

Nothing at all. And this is the fact the whole episode hangs off. Mishaal Rahman wrote the canonical explainer on this over at Esper back in July, and there's one line in it that says everything. "Android treats the eSIM profiles you download as if they're any other physical SIMs inserted into the device."



Corn

Which means the phone doesn't know it's looking at an eSIM.



Herman

The phone has no idea. There's no eSIM anywhere in its mental model. It sees a SIM card in a slot with a profile on it. That's it. And every quirk we're about to talk about, every limitation, every odd behaviour, flows out of that one fact. The adapter is a bridge. It's pretending to be something it isn't, and it's very good at it, but the seams show.



Corn

So if Android can do everything the reader does, why does the reader exist?



Herman

Because Android isn't the only platform, and because the person using the card isn't always the person provisioning it. Let's take the first one. The 9eSIM reader is fifteen dollars. It's USB-C. It's a standard PC/SC device, CCID class, so no driver install on anything. Windows, macOS, Linux, Android, iOS, it just shows up.



Corn

Fifteen dollars and no drivers. That's a well-behaved piece of hardware.



Herman

It is. And here's why it exists. From their own FAQ, and this is close to verbatim: "iOS does not let third-party apps write directly to a physical SIM, so on iPhone you'll need an external USB card reader."



Corn

Ah.



Herman

HarmonyOS NEXT has the same restriction, versions five and six. So on an iPhone, or on a Huawei running HarmonyOS NEXT, the phone physically cannot write to the card. The reader isn't a convenience there, it's the only path. You pull the card out, put it in the reader, plug the reader into the phone or a laptop, write the profile, put the card back.



Corn

That's a different workflow. You're doing card surgery every time you want to change a profile.



Herman

Every time. And there's a second group. Older Android. There's an OzBargain user, Russ, who put it plainly: "Android versions earlier than Android 11 don't allow communication between the app and the eSIM adapter." So on Android 10 and below, you need the reader. In this case you'll need the card reader to add, delete or select the eSIMs.



Corn

So the reader is a compatibility shim for old phones.



Herman

Partly. Then there's the third group, and this is the one I find most interesting. Fixed devices. Routers, IoT gateways, trackers. The 9eSIM product page lists it as a stated use: loads new eSIM profiles without putting the card in your phone, handy when the card lives in a fixed device. You can't run an app on a travel router. There's no screen, there's no Android, there's no OMAPI stack. So you pull the card, write it on the reader, put it back.



Corn

And the fourth group is anyone doing this at volume.



Herman

Bulk provisioning. If you're a company issuing two hundred of these to field staff, you're not going to sit there pairing phones one at a time. You're going to have a reader on a desk and a stack of cards.



Corn

So the reader is a platform workaround, not a technical necessity.



Herman

That's exactly what it is. Daniel's intuition is correct for Android. On Android 10 and up, the phone can do everything the reader does. The 9eSIM tool guide says it outright, the Android apps work over OMAPI or a PC/SC reader, either one. The reader exists because other platforms forbid what Android permits, and because some devices have no UI to run an app on.



Corn

There's something a bit funny about that. The reader is a workaround for platforms that won't let you touch your own hardware.



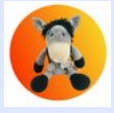
Herman

It's a workaround for a policy, not a limitation. Which is a very different kind of thing.



Corn

So that's question one answered. Now the second half of it, which is the provisioning quirks. Because Daniel mentions running into those, and I think people assume they're the adapter's fault.



Herman

They're almost never the adapter's fault. This is the part where I'd push people to look at their phone vendor before they blame the card. Let's do the documented cases, because they're specific and they're all recent.



Corn

Go.



Herman

OpenEUICC issue three four nine. Filed the eighteenth of August. Samsung invalidates the OMAPI logical channel during profile switching. What that looks like from the user's side is a stuck loading screen, and in the logs you get iccCloseLogicalChannel, invalid argument. The channel that the app opened to talk to the card gets yanked out from under it mid-operation.



Corn

By Samsung.



Herman

By Samsung. There's a fix, PR three five zero, merged on the twenty-second of August. But the maintainers describe it as not fully resolved. So it's patched, not solved.



Corn

What else.



Herman

Issue three five two, filed the twenty-fifth of August. This one's nastier. An empty SIM slot on a dual SIM phone causes com.android.se to spin at a hundred percent of one CPU core. The user measured sixty percent of total battery drain, about seventeen percent an hour. The phone was cooking. Battery temp went from twenty-six point nine to thirty-four degrees.



Corn

An empty slot.



Herman

Not the adapter, not a profile, a hole where a card isn't. And the maintainer, PeterCxy, his response was blunt. "Known bug, not something EasyEUICC can fix. Ask Samsung. It is not EasyEUICC that's re-probing anything."



Corn

Ask Samsung. That's the whole support ticket in three words.



Herman

It's a vendor bug. The adapter is sitting there doing nothing and the phone is burning itself alive looking for it.



Corn

Now, Daniel's on a OnePlus. Is there anything specific to him?



Herman

There is, and this is the one I'd actually want him to test. The osmocom eUICC manual, which is the reference documentation for this whole world, says this about OnePlus: "Some OnePlus smartphones cannot access the standard ISD-R AID. There is no userspace solution to this problem, which is a RIL limitation. Solution: if the card side provides the modified ISD-R AID, then it can be accessed."



Corn

Slow down. What's an AID, and what's ISD-R?



Herman

An AID is an application identifier. It's how you address a specific application on a smart card. ISD-R is the issuer security domain for the root, it's the application that manages profiles on an eUICC. So the phone needs to reach that application to do anything useful. On some OnePlus devices, the radio layer won't let it reach the standard address for it.



Corn

And there's no software fix.



Herman

There's no software fix. It's in the RIL, the radio interface layer, which is below anything an app can touch. The only workaround is if the card itself exposes the application at a different address, a modified one, that the phone will accept.



Corn

So the fix lives on the card, not the phone.



Herman

The fix lives on the card. Which means Daniel, if his 9eSIM is working fine on his OnePlus, his card is probably already exposing the modified address. He could check. But the fact that he has no complaints is itself the answer. He's on the good side of that particular lottery.



Corn

And Xiaomi?



Herman

Xiaomi and HyperOS on Snapdragon will seize the ISD-R logical channel, which results in unavailability. Same shape of problem, different vendor. The phone grabs the channel and won't let go, so the app can't get in.



Corn

So to summarise the quirks. Samsung invalidates channels, Xiaomi seizes them, OnePlus sometimes can't reach them, and none of it is the card's fault.



Herman

None of it. The card is a passive smart card. It answers when it's asked. If the phone won't ask properly, or asks and then hangs up mid-sentence, that's the phone.



Corn

I find that weirdly reassuring and also slightly insulting, as a piece of consumer advice. The answer to "why is my adapter misbehaving" is "your phone vendor made a decision."



Herman

It's the most common answer in Android troubleshooting and it never stops being annoying.



Corn

So the reader is a platform workaround, and the quirks are usually the OEM. But Daniel had two more questions. Performance, and whether two adapters can coexist.



Herman

Let's take performance, and I want to be honest about the negative finding first, because it's the honest answer.



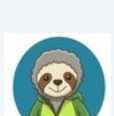
Corn

Meaning what.



Herman

Meaning nobody has measured it. I went looking for a direct benchmark, adapter versus native eSIM, latency or throughput, and there isn't one. Not on the web, not on Hacker News, nowhere. This is a hobbyist topic and nobody has put a stopwatch on it.



Corn

So what do we actually know?



Herman

We know the architecture, and the architecture says there shouldn't be a difference. The APDU path is the same ISO-7816-4 command set either way. The adapter is a real smart card in a real SIM slot. The modem talks to it identically to a physical SIM, because from the modem's side it is one. There's no translation layer, no proxy, no extra hop. The commands go down the same wire to a chip in the same place.



Corn

So the only difference is what's on the chip.



Herman

What's on the chip, and how it was provisioned. And that's the key distinction. There's one performance anecdote out there and it's instructive. OzBargain reviewer Russ noted that webpages loaded slower than normal. But he attributed it himself to the eSIM provider's internet breakout being in Mauritius or Singapore. Not the adapter.



Corn

Routing.



Herman

Routing. The traffic was taking a longer physical path because of where the carrier's gateway sits. That's a property of the network you bought, not the plastic in your SIM tray. And this matters, because I can absolutely see someone buying an adapter, noticing their pages load slower, and blaming the adapter, when the actual cause is that their data is going out through Singapore.



Corn

That's a bad afternoon for the adapter's reputation.



Herman

Entirely undeserved. Daniel's "performance seems fine" is exactly what the architecture predicts. He's not imagining it. There is no overhead to notice.



Corn

So where is there a real difference?



Herman

Profile switching. That's the one place adapters are slower, and it's not close. From the 9eSIM FAQ, first time setup on a new card takes twelve to eighteen minutes. After that, switching between profiles takes under thirty seconds, and you do it through the SIM Toolkit menu.



Corn

Twelve to eighteen minutes.



Herman

First time. That's the initial provisioning, writing the first profile and getting the card into a state where it's useful. After that it's under thirty seconds a switch.



Corn

And native eSIM switching?



Herman

Near instant. You tap a profile in Settings and it's live. So there's a real, measurable gap. But it's a setup cost, not a runtime cost. You pay it when you're configuring, not when you're using.



Corn

Which is the right way round, honestly. I'd rather wait eighteen minutes once than have every page load be slow forever.



Herman

And that's exactly the trade. The adapter is slow to set up and identical in use. Native eSIM is instant to set up and identical in use.



Corn

Now the capabilities question. What does native eSIM give you that the adapter can't?



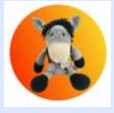
Herman

This is where it gets structural. The big one is Multiple Enabled Profiles. MEP. Android 13 introduced it, and what it does is let a single eUICC run more than one profile at the same time. So one chip, two active lines.



Corn

And adapters can't do that.



Herman

Adapters can't do that. The GSMA spec, SGP.21 version three point zero, says it directly. Support of Multiple Enabled Profiles on a removable eUICC is, and I'm quoting, "for further study."



Corn

For further study. That's spec language for "no."



Herman

That's spec language for "we have not decided to do this and we may never." So MEP is off the table for removable eUICCs. Which means if you want dual SIM with an adapter, you need either a built-in eSIM alongside it, or a second physical slot with something in it.



Corn

So each adapter is one line. One line at a time.



Herman

One profile active at a time per adapter. You can store fifteen or thirty or fifty profiles on the card, depending on the model, but only one of them is live.



Corn

That's the ceiling, then.



Herman

That's the structural ceiling. And there's a second gap that's less obvious but shows up in daily use. Settings integration. Adapter profiles do not appear under Downloaded SIMs in Android Settings. They're treated as ordinary physical SIMs, because that's what the phone thinks they are. Native eSIM integrates with Google's SIM Manager, it shows up in the right place, the carrier provisioning flows know how to talk to it.



Corn

So the adapter works, but it's in the wrong drawer.



Herman

It's in the wrong drawer, and some carrier flows assume the native eSIM path exists. So you occasionally hit a provisioning step that doesn't know what to do with you.



Corn

And the third gap?



Herman

EID whitelisting. This is the underreported one and I think it's the one that would actually bite a listener. Every eUICC has an EID, a unique identifier. Some carriers blacklist specific adapter EIDs.



Corn

Blacklist them.



Herman

There's an OzBargain user, FancyRabbit, who reported that Telstra blacklists 9eSIM's EID prefix, eight nine zero four four zero four five, but not eight nine zero eight six zero three zero, which is an Eastcompeace card.



Corn

So the same product, different chip inside, one works on Telstra and one doesn't.



Herman

Same adapter brand, different eUICC vendor underneath, and the carrier has decided one of them is acceptable and the other isn't. And you'd never know until you tried. There's no warning on the box.



Corn

That's a silent limitation on which networks you can roam onto.



Herman

Completely silent. And for a traveller, which is the main use case for these things, that's the trap. You buy the card, you buy the profile, you land, and it doesn't attach, and the reason is a prefix on an identifier you've never heard of.



Corn

So the adapter isn't broken, it's been disinvited.



Herman

It's been disinvited, by a carrier, for reasons that are theirs and not published.



Corn

Right. Third question. Two adapters in a dual physical SIM handset.



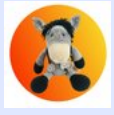
Herman

It can work. There's an OzBargain user, zeemlofuggedxyz, who says it plainly. My phone has dual SIM, both physical nano SIM, and now I use two eSIM adapters.



Corn

So it's possible.



Herman

It's possible and it's fragile. Let's do the failure cases, because they're more useful than the success case. There's a report from the iodéOS community on the eleventh of September. Pixel 8 Pro, dual adapter setup. It worked on first activation. Then he rebooted, and it never reconnected to the mobile network again.



Corn

Reboot broke it.



Herman

Worked on first activation, broke on reboot, and the same hardware worked fine on CalyxOS. Same phone, same adapters, different ROM, different result.



Corn

That's a software stack problem, not a hardware one.



Herman

Entirely software. And then there's the Oppo A74 case. The reviewer toggled a slot and got the message "No eSIM. Disable eSIM to use SIM 2." Which is a strange thing to be told by a phone that doesn't have eSIM.



Corn

That's a phone arguing with itself.



Herman

FancyRabbit's explanation is that it's a bug in Android where the phone bans the ICCID of the profile after you disable it. So the phone blacklists the card it just told you to remove.



Corn

And the fix?



Herman

There's a workaround. You clear the telephony provider's state with an ADB command, pm clear com dot android dot providers dot telephony. That resets the phone's idea of what cards it's seen, and the profile becomes usable again.



Corn

So the recovery path is a command line.



Herman

The recovery path is a command line, which tells you what tier of user this configuration is for.



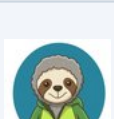
Corn

And the fundamental limit here is still MEP, isn't it. Even if both adapters work, you've got two separate single-profile SIMs.



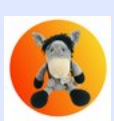
Herman

Two separate single-profile SIMs. Each adapter is one line. So you get dual SIM, but you get it the old way, two physical cards in two slots, and the phone has no idea either of them is an eSIM.



Corn

So it works, but you're running the fragile configuration.



Herman

You're running the configuration where a reboot might orphan you and a toggle might ban your own card. It's not unsupported because it's impossible. It's unsupported because it's held together with vendor bugs.



Corn

Which brings me to the thing I keep circling. All of this points the same direction.



Herman

Say it.



Corn

The adapter is frozen out of where Android is going. MEP is explicitly for further study on removable eUICCs. Adapter profiles don't appear as eSIMs in Settings. The Settings integration, the SIM Manager integration, the multi-profile support, all of that is being built for native eSIM and the adapter is standing outside the window.



Herman

That's the honest summary. Adapters are a bridge technology. They work well right now, they solve a real problem for people with phones that don't have eSIM, and Daniel's experience is typical. But they're structurally capped. No MEP, no Settings integration, EID blacklisting, fragile dual-adapter support. They're the present for people who need them. They're not the future.



Corn

There's a man at the mixing desk who has been very patient.



Hilbert

The reader isn't redundant. You keep saying it is.



Corn

Go on.



Hilbert

You're right that the phone can do it. That's not the point. The reader was never for the person holding the phone.



Corn

Who was it for?



Hilbert

The person behind the counter. I did a stint at a small MVNO, a SIM provisioning desk. Drawer full of blank cards, card writer on the desk, thermal printer next to it that spat out the ICCID labels. Smell of that printer. You'd take a card out of the drawer, put it in the writer, burn the profile on, peel the label, stick it on the card, hand it over.



Corn

And the customer's phone was never involved.



Hilbert

Never in the loop. It never touched the card until the card was already done. That's why the reader still exists. It's a provisioning tool. It's not a user tool. The fifteen dollar one is the same idea, it's just USB-C instead of a rack unit in a back office.



Herman

So it's a back-office tool that escaped into the consumer market.



Hilbert

That's what it is. We had a box of cards that were pre-provisioned. Profiles already burned on. Whole point was the customer never had to do anything. Walked in, walked out, phone worked.



Corn

That's the same workflow, just with the customer doing the provisioning themselves.



Hilbert

Same workflow. Smaller scale. Cheaper hardware. There's a phone call I'm expecting, I'm not taking it in here.



Corn

That's a good correction. The reader was never for the user.



Herman

It reframes the whole first question, actually. Daniel asked why the reader exists when the phone can do it, and the answer is that the reader was never competing with the phone. It was doing a job the phone was never in the room for.



Corn

Let's pull this together, because there's a lot sitting on the table. If you take one thing from this, it's that the adapter is a very good impression of a SIM card, and every limitation you'll hit comes from the fact that the phone believes the impression.



Herman

The corollary. When something goes wrong, look at the phone vendor first. Samsung invalidating channels, Xiaomi seizing them, OnePlus refusing to reach the right address. The card is answering the phone. It's the phone that's misbehaving.



Corn

Which leaves two open questions I don't think anyone can answer yet. Will MEP ever come to removable eUICCs, or is the adapter category permanently capped at one line per card? And will carriers keep blacklisting EIDs as native eSIM becomes standard, or does that fade out once the adapter market is small enough not to matter?



Herman

Both of those are open. I'd guess MEP stays for further study for a long time, because nobody's lobbying the GSMA on behalf of a fifteen dollar card.



Corn

If you're on the fence about buying one, here's the honest version. On Android, the reader is probably not for you. And the quirks you hit are probably your OEM, not the adapter.



Herman

Thanks to Hilbert Flumingtop for producing, and for the correction.



Corn

This has been My Weird Prompts. If you want to tell us where we got it wrong, email us at show at my weird prompts dot com.



Herman

We'll be back soon.