
Why “Cheap” Hardware Is the Most Expensive Mistake You’ll Make

A Look at the Hidden Costs of Uncertified
Modules and Flaky Connectivity

CONTENTS

01	The \$12 Module and the Six-Figure Mistake	03
02	What the Sticker Price Doesn't Include	04
03	Certification, Decoded	06
04	Anatomy of a Flaky Connection	08
05	The Real Bill	10
06	A Framework for Evaluating Hardware	12
07	What Getting It Right Looks Like	14
–	Conclusion	16
A	Certification Quick Reference	17
B	The Hardware Evaluation Checklist	18

Executive summary

01

The module on a bill of materials is rarely the real cost of "going cheap" on connected hardware. The real cost shows up later, in re-certification, field failures, and support burden.

02

Certification is the only mechanism that verifies a device will actually be allowed onto – and stay reliably connected to – a real network anywhere in the world you plan to deploy it.

03

A short evaluation framework lets a buyer catch these risks at the sourcing stage, before they show up as a support ticket eighteen months into a deployment.

01

The \$12 Module and the Six-Figure Mistake

On a unit-cost spreadsheet, an uncertified or gray-market cellular module can look meaningfully cheaper than a certified equivalent. In theory, it's a simple margin decision; in practice, it's closer to a loan against risks you haven't priced yet.

In this eBook, we will examine the things that tend to fill that gap once a device is actually in the field. A specific, traceable set of costs: re-certification, field service, support escalations, and the slower, harder-to-line-item cost of a customer who no longer trusts the product.

Let's be clear at the outset: "cheap" and "inexpensive" are not the same thing. A properly certified module, paired with connectivity designed to hold up outside a lab, can be the inexpensive choice once you account for what happens if and when it fails - even if it's rarely the cheapest number on the initial spreadsheet.

To show this in practice, we'll follow a composite scenario through the next few chapters: a manufacturer of connected industrial equipment, shipping a sensor product to customer sites worldwide. The challenge represents a blend of patterns that show up often enough in cellular IoT deployments to be worth walking through in detail.



02

What the Sticker Price Doesn't Include

A cellular module's list price is one line item in a much longer bill. Getting a working radio into a finished product also means antenna design and tuning, RF filtering, power management tuned to the module's draw characteristics, and PCB real estate set aside for all of it. None of that is optional, and none of it shows up in a module's per-unit quote.

MODULE LIST PRICE

Antenna Design and Tuning
RF Filtering
Power Management
PCB Real Estate

Industry cost breakdowns generally provide the fully loaded cost of a cellular IoT device, once these supporting components and integration work are included, at several multiples of the module's list price. Treat that as a directional pattern rather than a number to quote precisely. It varies a great deal by design, radio technology, and production volume, and different sources land on different multiples.

Here's the part that matters more than the multiplier itself: skipping certification doesn't just save the certification fee. It usually means skipping the engineering rigor that certification exists to verify in the first place, since a device that's never been tested against the standard has no independent check on whether the antenna, power design, and firmware actually hold up. The module price was never the whole story, and certification is a part that all too often gets skipped.

Skipping certification doesn't just save the certification fee. It usually means skipping the engineering rigor that certification exists to verify in the first place.



03

Certification, Decoded

It's easy to hear "get a certified module" as a compliance checkbox and move on. It's worth understanding what each certification actually verifies, because they're not redundant with each other. Each one catches a different kind of failure.

Regional Regulatory Marks

a radio doesn't emit outside its allowed spectrum and won't interfere with other devices or networks nearby

PTCRB and GCF

a device correctly implements the cellular standard it claims to support, tested against real network behavior rather than just a lab bench

Operator-specific Certification

a device type is accepted onto an individual operator's network at scale

Regional regulatory marks (the FCC in the United States, CE and UKCA in the EU and UK, and equivalent marks in other markets) verify that a radio doesn't emit outside its allowed spectrum and won't interfere with other devices or networks nearby. This is essential to legally selling and operating a device in a given market. It is not, on its own, a guarantee the device will actually work well on a live cellular network.

PTCRB and GCF, industry-run certification bodies that verify whether a device correctly implements the cellular standard it claims to support, tested against real network behavior rather than just a lab bench. This is what catches a device that transmits within spec but fails to register, hand off between towers, or hold a session under real-world conditions.

Operator-specific certification sits on top of both of these. Individual network operators around the world often run their own acceptance testing before allowing a device type onto their network at scale, since even a fully standards-compliant device can behave differently against one operator's specific network configuration than another's. For a device meant to work across multiple countries and multiple operators (as most global IoT deployments are), this layer matters more than it might first appear.

The throughline across all three: each layer checks a different failure mode; legal and interference risk, standards compliance, and operator-specific compatibility. Skipping any one of them doesn't remove the risk it's designed to catch. It just moves the discovery of that risk from a certification lab to a live deployment, which is a far more expensive place to find it.



04

Anatomy of a Flaky Connection

Certification is the paperwork you hopefully dealt with before the deployment. This chapter is about what actually goes wrong in the field if that step is skipped, or when connectivity is treated as an afterthought rather than a design requirement from the start.

When it comes to diagnosing a problem, the usual suspects are unglamorous: antenna placement compromised by a tight enclosure, firmware that was never hardened against real network conditions like congestion or marginal signal, and power draw spikes during radio transmission that brown out the rest of the device. None of these show up as a hard failure in a demo. They show up as devices that "mostly" work.

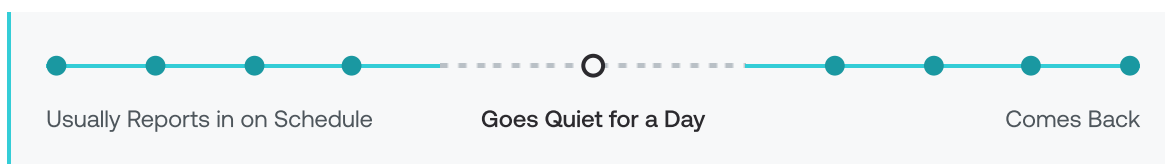
Antenna placement compromised by a tight enclosure

Firmware that was never hardened against real network conditions like congestion or marginal signal

Power draw spikes during radio transmission that brown out the rest of the device

The word “mostly” may prove quite expensive. A five-unit pilot running in an engineer’s office, close to a strong signal, rarely surfaces these problems. They tend to appear once a design reaches real scale, spread across real sites with real interference, real building materials, and real distance from a tower, which is exactly the environment a pilot doesn’t reproduce.

Back to the composite scenario from Chapter 1: picture that equipment manufacturer, several months into a rollout, fielding a slow trickle of support tickets that all sound similar. A device usually reports in on schedule, then goes quiet for a day, then comes back. No two field technicians can quite reproduce it on demand. Support spends real time ruling out the customer's network, the device's placement, and the backend before anyone even confirms whether it's a hardware issue at all. That ambiguity, more than any single failure, is what makes flaky connectivity expensive. It isn't a clean, diagnosable break. It's a slow, ongoing tax on support and engineering time.



05

The Real Bill

Once a certification gap or a connectivity design shortcut surfaces post-deployment, the costs it generates look nothing like the number that made the original module attractive on paper.

Re-certification and redesign is usually the first and most direct cost. Third-party estimates suggest that re-testing after a failed or skipped certification pass adds meaningfully more time and expense than a first-pass certification would have, since it typically requires an actual hardware revision and a second full round of lab time, not just a resubmission. As with the multiplier in Chapter 2, treat any specific figure here as a rough, sourced estimate. Costs vary widely by device complexity and by how much redesign is actually required.

Field service and logistics is the second, and often larger, cost once devices are already deployed. Replacing hardware that's already shipped, whether through a technician visit or a customer-managed swap, is a fundamentally different cost profile than catching the same problem on a bench before a single unit left the factory.

Support burden compounds both of the above. Continuing the composite scenario from Chapter 4, every one of those ambiguous "goes quiet, comes back" tickets consumes support and engineering hours before anyone even confirms root cause, and that diagnostic time scales with the size of the deployment, not with the number of actual failures.

Then there's the cost that never shows up on a spreadsheet at all: **opportunity cost and trust**. Engineering time spent firefighting a hardware problem in the field is time not spent on the next feature or the next market. And a customer whose first experience with a connected product is an unreliable one is a much harder sell on the next generation of it, whatever the eventual fix looks like.

None of this appears on the original module price comparison. It shows up later, spread across a P&L, a support queue, and a renewal conversation, which is exactly why it's easy to underweight at the moment a sourcing decision actually gets made.



06

A Framework for Evaluating Hardware Before You Commit

The previous chapters make the case for why this matters. This one is meant to be used, not just read: a short set of questions worth bringing into any hardware or module sourcing conversation, framed as things a buyer can ask for and verify directly, not take on faith.

01 Ask for documentation, not assurances.

Request the module or device's regulatory ID (an FCC ID or regional equivalent) and confirm explicitly whether the certification covers the module alone or the finished end product it will ship inside. These are genuinely different things, and end-product certification is what actually matters once the module is integrated into your design.

02 Confirm PTCRB and/or GCF listing, and for what specifically.

A listing generally covers particular network bands and technologies. Confirm the listing actually matches the bands and technologies you intend to deploy, in every market you intend to deploy to, rather than assuming a listing is a blanket guarantee.

03 Ask whether operator-specific certification has actually been completed,

and in which markets, for the module or device as you intend to deploy it. A device that's PTCRB or GCF listed but has never been tested against a specific operator's live network still carries real, undiscovered risk in that market.

04 Ask about the chipset vendor's committed firmware support lifecycle.

A certified module built on an end-of-life chipset is a shorter-term fix than it looks like on the day you buy it, since firmware updates and long-term support are part of what keeps a certified device compliant and reliable over its actual field life.

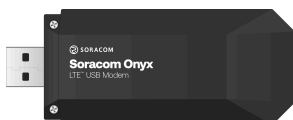
05 Ask for a reference deployment at meaningful scale,
not just a lab demonstration or a single pilot. A module that performs well on a bench and a module that performs well across a few thousand units spread across real sites are not always the same claim.

Five questions, asked before a purchase order is signed, is a small amount of friction compared to the costs described in Chapter 5. That's the actual trade this framework is asking a buyer to make.



07 What Getting It Right Looks Like

It's worth being concrete about what it looks like when certification and connectivity are treated as part of a product's design from the start, rather than bolted on after a field problem. Two examples from Soracom's own hardware line illustrate the pattern, without being the only way to apply it.



Soracom Onyx

FCC PTCRB TELEC CE

Soracom Onyx, a USB cellular modem that is certified as a complete end product, not just as a component, across FCC, PTCRB, TELEC, and CE among other regional marks. The practical benefit of end-product certification specifically: a team building on it isn't starting a certification process of its own from zero.



Soracom LTE-M Button

FCC CE PTCRB GCF

Soracom LTE-M Button, a purpose-built trigger device rather than a general-purpose modem, carries FCC, CE, PTCRB, and GCF certification out of the box, with connectivity built in and configured before it ships. It's a useful example of the same principle applied to a finished product meant to be deployed directly, rather than integrated into a larger design.

This pattern isn't unique to Soracom-branded hardware, and it shouldn't be read that way. It shows up across the certified-module ecosystem more broadly, including through Soracom's own partnerships with module and connectivity technology vendors like Sequans and Kigen, among others, each addressing a different part of the certified-hardware picture rather than competing for the same slot. The point isn't which specific vendor a reader chooses. It's the underlying principle: look for hardware where certification and real-world connectivity were part of the original design brief, verified and documented, not assumed.



CHAPTER 07

The practical benefit of end-product certification specifically: a team building on it isn't starting a certification process of its own from zero.

CONCLUSION

The Cheap Module Is a Loan, Not a Discount

The expensive mistake in the title of this eBook was never really a single bad purchase order. It's treating a risk-transfer decision (i.e. whether a device can reliably get certified, connect, and stay connected) as if it were only a line-item cost decision.

The good news is that the fix isn't complicated. Five simple questions asked before a sourcing decision is finalized rather than after a field failure forces the question can make a world of difference. That's the practical takeaway of this eBook, and it's worth keeping close at hand the next time a hardware evaluation lands on your desk.

Looking to change out your hardware, but uncertain of what to look for? [Contact Soracom's IoT experts](#) to discuss what a properly certified, reliably connected deployment looks like for your fleet.

Certification Quick Reference

CERTIFICATION	WHAT IT ACTUALLY VERIFIES	WHAT HAPPENS IF YOU SKIP IT
Regional regulatory marks (FCC, CE/UKCA, and equivalents)	The radio stays within its allowed spectrum and won't interfere with other devices	The device isn't legally sellable or operable in that market
PTCRB / GCF	The device correctly implements the cellular standard against real network behavior, not just a lab bench	The device may transmit within spec but fail to register, hand off, or hold a connection reliably
Operator-specific certification	The device behaves correctly against a particular network operator's specific configuration	Even a standards-compliant device can be refused activation, or behave unpredictably, on that operator's network

The Hardware Evaluation Checklist

A condensed version of Chapter 6, formatted for quick reference during a sourcing conversation:

- ☐ Regulatory ID requested, and scope confirmed (module-level vs. end-product certification)
- ☐ PTCRB and/or GCF listing confirmed, matched against your actual target bands and technologies
- ☐ Operator-specific certification status confirmed for every market you plan to deploy to
- ☐ Chipset vendor's firmware support lifecycle confirmed in writing
- ☐ Reference deployment at meaningful scale requested and reviewed, not just a lab demo