TECHNICAL EXPLAINER · 2026 · REMOTE ON-DEVICE TESTING

How TestLink™ works, and why it never touches your network

A purpose-built test instrument that puts your device in our lab without the device, or your IT department, going anywhere.

KEY TAKEAWAYS

- TestLink™ is a custom instrument that sits on your bench, cables to your device's interfaces, and reaches ELTON over its own private 5G uplink. It never joins your guest, enterprise, or lab network.
- The connection is a closed loop to the ELTON lab: an always-on VPN tunnel with mutual certificate authentication, TLS 1.3 in transit, and AES-256 for artifacts at rest. Testing over TestLink™ is equivalent to racking the device in our lab.
- There is no firewall change, no IT ticket, and no security review of a network integration, because there is no network integration.
- A full x86 single-board computer runs the harness. A reprogrammable interface board simulates and drives USB, CAN, serial, and other buses. A hardware KVM instruments the device UI remotely.
- Pairing binds a node on the digital twin to the physical port the instrument is cabled to. ELTON builds custom skills and tools from the twin artifacts, and execution results map back to the same node.
- Three delivery models run on the one instrument: a tunnel to our lab today, AI at the edge next, and self-service on your bench. TestLink™ is included in every ELTON subscription.

> **"Plugging in TestLink connects your product to our lab, not to your network."**

The problem it removes

Exploitability is a runtime question. You answer it on the real product, against live interfaces, or you are guessing. That used to mean one of two bad options: ship the device to a test lab and lose it for weeks, or let a vendor's test rig onto your corporate network and spend a quarter in security review before the first packet moves.

TestLink™ deletes both. The device stays on your bench. The instrument arrives in a hardened travel case, powers up, and finds our lab over its own cellular radio. Your network team never gets

a ticket because there is nothing to review. From the first cable to the first executed test case is measured in minutes.

Inside the instrument

TestLink™ is a custom instrument we build, image, and serialize in our Naperville lab. It is not a laptop with a VPN client. The chassis is machined, cooled by a mounted fan behind a UL fire-retardant faceplate, and cabled internally over a managed switch fabric. Each subsystem exists to put the harness in direct contact with a real device interface.

SUBSYSTEM	WHAT IT DOES UNDER TEST
Full x86 single-board computer	Runs the ELTON test harness locally on the instrument. A complete compute node, not a passthrough, so generated tooling executes one cable away from the target.
Reprogrammable interface board	A compute module on a carrier with programmable IO that speaks the device's buses directly: USB device and host roles, CAN, serial (UART), SPI, I2C, and GPIO. It can emulate a peer, inject malformed traffic, or drive an interface that a general-purpose PC cannot reach.
Hardware KVM plane	Keyboard, video, mouse, plus serial and USB, captured at the physical layer. The harness operates the device's own console and UI remotely, as if a tester were standing at it, with no software agent installed on the product.
Managed switch fabric	Isolated wired connectivity between the instrument's subsystems and the device's Ethernet interfaces, kept entirely inside the instrument.
RF capture radio	A dedicated wireless adapter for traffic capture and upstream control on the device's radios.
Cellular gateway	Carries the private 5G uplink to the ELTON lab. The only path off the instrument, and it never reaches your network.
Isolated power delivery	Power paths are hardware-isolated, and charge-only paths carry an inline data blocker, so a power connection cannot become a data connection.

Every serialized component in a unit is recorded against that unit, so the exact hardware that touched your product is traceable for as long as the record matters.

Security controls

The design goal was simple to state: testing over TestLink™ should be no different than a point to point connection to a device sitting in our lab. Everything below follows from that.

CONTROL	IMPLEMENTATION

Transport	Always-on VPN tunnel over private cellular, out of band from any network you operate. TLS 1.3 inside the tunnel. The instrument dials out only; it exposes no inbound listeners.
Mutual authentication	Both ends present certificates. The instrument carries a per-unit client certificate provisioned at build and bound to a hardware-backed key; the lab endpoint is pinned. Neither side talks to anything it cannot verify.
Data at rest	Findings, captures, and execution logs are AES-256 encrypted on the instrument and in the platform. Keys are held in a secure element, not on general storage.
Boot integrity	The harness image is signed and measured at boot. A modified image does not run and does not receive a tunnel.
Fail closed	If the tunnel drops, testing stops and the instrument holds nothing open. There is no local operator path into the device that outlives the session.
Blast radius	There is no integration to threat model on your side. The reach of TestLink™ is the cable you plugged into the device under test, and nothing else.

The twin makes it precise

TestLink™ does not aim a generic scanner down a pipe. When the instrument is paired, the platform knows exactly which node of the digital twin the cable is on: the component behind the port, the protocol it speaks, and the trust it carries on this product. ELTON builds custom skills and tools from those twin artifacts, the architecture, the SBOM, and the documented interfaces, shaped to the specific interface in front of the instrument. A DICOM listener gets DICOM tooling. A CAN bus gets a CAN fuzzer configured for the frames this device actually uses. A vendor service port gets tooling built for that vendor's stack.

The pairing works in both directions. Twin knowledge decides what the instrument tries, and what the instrument observes writes back to the twin: what answered, what was live, how the port behaved under test. The digital model and the physical product stay mapped to each other, so every finding lands on the node it was proven against, and the feedback is real because the hardware is.

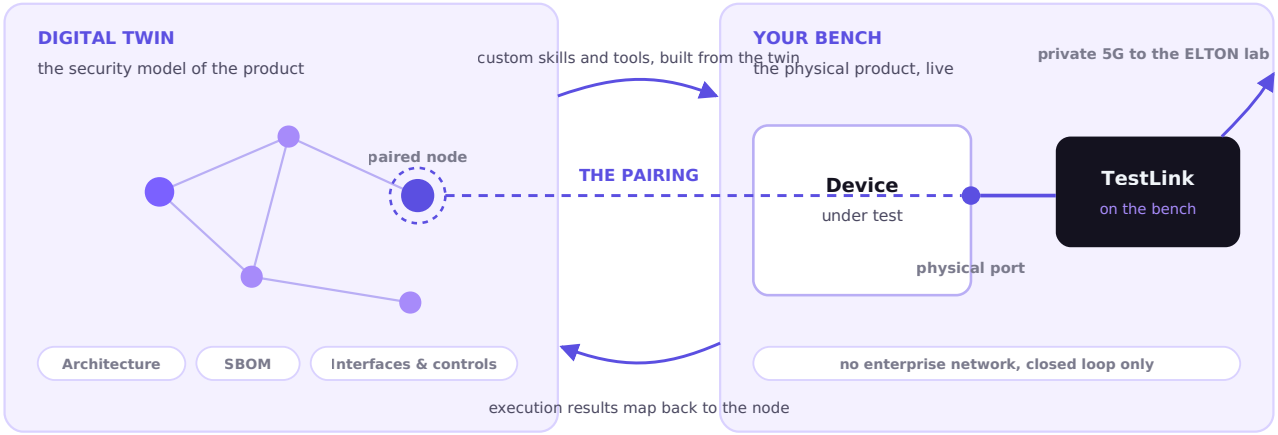

Figure 1. Pairing binds a twin node to the physical port. Custom tools built from the twin artifacts flow to the instrument, and execution results map back to the same node.

Form one: a tunnel to our lab

In its current form, TestLink™ is a remote access point into a closed loop. Cable the instrument to the interface you want tested: USB, CAN, Ethernet, serial, or over the capture radio for wireless. Our team drives the same AI test harness we run in the lab, through the tunnel, against your live device. The harness does not know the difference, and neither does the evidence. Test cases execute on real hardware, results stream into the platform, and every finding lands with the interface it was proven on.

Form two: AI at the edge

The x86 compute plane in the instrument is sized for more than passthrough. The next form runs the pipeline's generated tooling on the instrument itself. AI enumerates what is physically connected through the reprogrammable interface board, generates test cases scoped to that interface, and executes them locally, with the tunnel carrying results and orchestration instead of raw test traffic. Discovery gets faster because the tooling sits one cable away from the target instead of one continent.

Form three: self-service on your bench

The same instrument, driven by your team. Open the product's release view in the platform, point at the node on the digital twin that matches the physical cable, and launch. The pairing step is what makes self-service testing precise: the platform reasons about the port the way an attacker standing at it would, and generates only the cases that make sense there. Your product team tests any hour, any day, and the evidence files itself. This is the model our customers use for release-over-release testing without shipping hardware anywhere.

What this means for a device program

The critical device never leaves your building. Your IT organization stays out of the loop because the loop never enters their network. And the output is executed evidence in the same platform that manages your vulnerabilities, dispositions, and FDA record. One instrument, three ways to use it, all included in the subscription.

About ELTON. ELTON is an AI-native, integrated cybersecurity testing and management service for medical devices. ELTON builds a digital twin of each device, runs testing across the full stack on every release, and proves which findings are exploitable, with every verdict evidenced for regulatory review. ELTON meets FDA premarket (524B) and postmarket cybersecurity testing and management requirements. Talk to us at **eltoncyber.com**.