The Unauthenticated Interconnect

Why disaggregated silicon needs trust architected into the die-to-die boundary, not bolted on after

Dana Xiadani Founder & Principal Architect, Living Cipher LLC

The industry has made its decision. Custom silicon has moved from monolithic SoCs to disaggregated, multi-die designs. The economics are settled: better yield on smaller dies, the right process node for each function, IP reuse across products, and a faster path from architecture to deployment. Heterogeneous integration is no longer the future. It is the production reality of AI accelerators, custom XPUs, and high-assurance systems shipping now.

But disaggregation changes the trust model, and most of the industry has not caught up to what it changed.

The assumption that no longer holds

A monolithic SoC is fabricated as one die, at one foundry, in one process. Whatever trust you needed, you could reason about a single, contiguous piece of silicon. The boundary of the chip was the boundary of trust.

Disaggregation breaks that boundary into pieces. A modern package now contains multiple dies, often from different vendors, foundries, and process nodes, connected by a die-to-die interconnect. Each die arrives with its own provenance. The interconnect standards that make this possible, the die-to-die PHYs and protocols, were designed to move data fast and cheaply across that boundary. They were not designed to answer a different question:

> *Should this die trust the die on the other side of the link, and on what evidence?*

Today, in most designs, the answer is assumed. Dies are bonded into a package and trusted by construction, because they are in the same package. That assumption was defensible when the chip was one die. It is not defensible when the package is an assembly of separately manufactured parts of mixed provenance.

The silence is the tell

Almost no one is talking about this. Search the discourse around multi-die and chiplet design and you find an enormous, sophisticated conversation about die-to-die bandwidth, about UCIe conformance, about power delivery, thermal, and signal integrity, about known-good-die test and yield. All of it necessary. Almost none of it about authentication. The interconnect is being engineered to a remarkable standard as a *data* channel, and left almost entirely unexamined as a *trust* channel.

That silence is not because the problem is solved. It is because the industry still treats multi-die as a packaging and performance achievement, and has not yet reframed it as a security boundary. The reframing is coming, the way it came for secure boot, for firmware attestation, for supply-chain provenance, each an afterthought until it was suddenly a requirement. The die-to-die boundary is on that same trajectory, and it is earlier on the curve than most of the field realizes.

This is not to say the standards are silent. The dominant die-to-die standard now defines optional, higher-layer hooks for authentication and integrity, and individual vendors have built security engines above them. But optional and higher-layer is the whole problem. What a package needs is not a security feature that some dies may negotiate in software, but an admission decision rooted in hardware and made before the link is allowed to carry anything. A hook that an attacker's software can decline is not a boundary.

Connectivity is not authenticity

A high-bandwidth, standards-compliant die-to-die link will faithfully carry data between two dies. It will do so whether or not the die on the other end is the die you think it is, whether or not its firmware is what it was when it was qualified, and whether or not it has been substituted, modified, or replayed between fabrication and final assembly. The link does its job perfectly. The trust question is out of its scope.

Make it concrete. A die is substituted between fabrication and final assembly, or its firmware is altered after it was qualified. It speaks the interconnect protocol correctly, so it trains the link, passes the cyclic redundancy checks, and satisfies every conformance test the interface applies, because those tests were built to catch corrupted data, not a dishonest source. It is now inside the trust domain of the package, communicating with dies that were never asked to verify it and have no means to. The error-detection machinery did its job perfectly. It was never the machinery that could have caught this.

This is the gap. As multi-die systems scale, and the supply chain for dies grows longer and more heterogeneous, the unauthenticated interconnect becomes the soft center of an otherwise hardened system. A system can have post-

quantum-secure external communications and a fully attested boot chain, and still have dies inside the package trusting each other on no evidence at all.

What trust at the boundary requires

The principle is simple to state and demanding to build correctly: **a die should not be admitted to the trust domain until it has proven, cryptographically and at the hardware root, that it is the specific authorized die, in the expected state, at this moment.** Not trusted because it is in the package. Trusted because it presented evidence, and the evidence was verified in hardware before the high-speed link was allowed to carry anything that mattered. Concretely, that means provenance (a die-unique, hardware-rooted identity that cannot be cloned or transferred), integrity (a measurement of die state checkable against a known-good reference), freshness (evidence bound to this session, so a captured exchange cannot be replayed), enforcement in fabric (a fail-closed admission decision made in hardware, not in software the attacker already owns), and a quantum-resistant cryptographic binding, because silicon shipping now will still be in the field when the threat model has moved.

The approach is practical, not theoretical

None of this requires a research breakthrough. The primitives it depends on are buildable today with standard methods: a hardware-rooted die identity, a measured-integrity check against a known-good reference, freshness-bound attestation evidence, and a fail-closed enable that gates the interconnect on the verdict, all verifiable at the register-transfer level and checkable against published cryptographic standards. A security primitive that merely runs is not the same as one that is correct, and that distinction, verifying the implementation against the standard's own reference behavior rather than trusting that it works, is where the engineering rigor belongs.

The quantum-resistant binding can be demonstrated in silicon. A post-quantum signature datapath (ML-DSA, FIPS 204) can be authored in fabric and taken through synthesis, place-and-route, and post-route timing closure on a production FPGA-SoC, closing at the worst-case process-voltage-temperature corner. Worst-case-corner timing is the number that means something; typical-case figures describe a chip that cannot be built to spec. And there is real craft in reaching that corner: the dominant arithmetic in such a datapath does not close on its own, and the difference between a datapath that closes and one that sprawls lives in implementation decisions that a monolithic figure never shows.

Proving the design on FPGA before committing to an ASIC tapeout is not a limitation; it is the order the work is meant to be done in, because a tapeout is a one-way, expensive door and the design that goes through it should already have earned the trip. A trust architecture proven on a monolithic device, where the discipline hardens a single die, extends to the multi-die case, where it authenticates trust across the boundary between dies of mixed provenance.

One idea, stated plainly

At root this is a single principle: authenticate the signal at the hardware root, upstream, with cryptographic proof, before anything downstream trusts it. In the multi-die case the signal is a die’s identity and evidence crossing an interconnect, and the post-quantum cryptography belongs in the fabric because the authentication has to be bound at the root, not asserted in software an attacker may already hold. The trust material should never cross an exposed bus as a recoverable value; that is a property to enforce in the architecture, not to promise in a policy. The principle is general, and this is one application of it. But the die-to-die boundary is where it has to be made real first, because that is where the silicon is shipping now.

The claim, reduced to what cannot be argued

The die-to-die boundary is a trust boundary. In most disaggregated designs today it is unauthenticated, and the field has barely begun to name it. It is an architectural gap that will have to be closed, deliberately, in hardware, and closing it correctly is a specialized discipline, not a feature to bolt on at the end. The sooner the industry treats the interconnect as a security boundary rather than a packaging achievement, the less expensive that reckoning will be.

Dana Xiadani · Founder & Principal Architect, Living Cipher LLC · dana@livingcipher.org

Author's note: The worst-case-corner timing closure described here is substantiated by tool-generated timing reports, available to qualified reviewers on request.