Layer1Labs Silicon

LAYER1LABS SILICON INC. · WHITEPAPER

# ChronoFabric Architecture Overview

The hardware tier of one architecture: the same program, the same compiled specification, enforced on FPGA fabric instead of in software.

Version 1.0 · October 26, 2026 · Layer1Labs Silicon Inc.

ChronoFabric is the hardware tier of the Layer1Labs architecture. It takes the programs that run in the ChronoHive software engine and enforces them on FPGA and hardware endpoints — the same deterministic guarantees, delivered at hardware speed. This paper describes that architecture from the hardware side: the contract both tiers share, the one thing that changes when a program moves from software to fabric, what has been measured on the beta platform, and what has not.

The evidence, in brief: on the beta platform — Kria FPGA evaluation boards (KR260/KV260) linked by 10-gigabit Ethernet — ChronoFabric delivered more than 21,000 scheduled packets with zero missed departures and zero dropped, a 100% delivery rate, with departure timing exact to the programmed tick. Those measurements are stated in full, with their run bases, in the companion document *ChronoFabric Measured Results*; production-scale fabrics (100G/400G, ASIC) remain the roadmap, not a measurement.

**See it live:** Layer1Labs — `https://layer1labs.ai`

© 2026 Layer1Labs Silicon Inc. All rights reserved. See the Notices at the end of this document.

# Contents

**1 The problem: software coordination has a ceiling** **3**

**2 One architecture, two tiers** **3**

**3 The compiled specification is the contract** **3**

**4 Enforcement in hardware** **4**

**5 Delivery: soft IP, ready for beta customers** **5**

**6 The staged roadmap** **6**

**7 Limits** **6**

**Notices** **8**

# 1 The problem: software coordination has a ceiling

AI training clusters stall expensive accelerators on coordination: checkpoint writes and data loading contend for the same storage bandwidth, and nothing arbitrates which work gets the bandwidth, or when. The Layer1Labs architecture answers that in software first — a program's coordination policy is compiled once and enforced deterministically by the ChronoHive engine (the subject of the companion whitepaper, *ChronoHive Architecture Overview*).

But software coordination has a ceiling: every decision costs processor time on the critical path. A software scheduler shares the machine it is scheduling, so its own timing degrades exactly when the machine is busiest. ChronoFabric exists to remove that ceiling. It is a deterministic orchestration architecture designed to unlock additional AI compute capacity by reducing synchronization waste across large-scale distributed accelerator systems — by enforcing the same deterministic schedules in hardware, where the scheduling work costs the host processor nothing. The outcome classes are the ones operators already plan around:

- **Throughput recovery** — less idle time across the fleet.
- **Power avoidance** — smoother infrastructure-level energy demand.
- **CapEx efficiency** — more useful work per watt, and per facility.

# 2 One architecture, two tiers

Programs are authored once, in Lingua Franca — a programming language built for deterministic, timed systems, whose programs declare what happens, and exactly when, as timed reactors (self-contained components that react to events). A program declares `target Chrono`. Our compiler, chronoc, compiles the program into a single constraint specification (.cspec, format CSP1) — one self-contained artifact carrying the whole program: its operations, dependencies, capacities, and schedule.

That specification is the deployment contract for both tiers (Figure 1):

- the **ChronoHive engine** runs it in software; and
- **ChronoFabric** runs the identical specification on hardware.

Only the effector binding changes. An *effector* is a connection point where a program acts on the outside world — a storage write, a network send, a hardware action. In software, effectors bind to software callables (ordinary program functions). On ChronoFabric, the same effectors bind to FPGA and hardware endpoints — the physical interfaces of the device.

Nothing else about the program changes. The architecture's three guarantees — **admission** (which work is allowed to proceed), **sequencing** (the order work happens), and **determinism** — carry over from the software tier unchanged. A team that has validated a program on ChronoHive is not re-validating a different program on ChronoFabric; it is re-binding the same one.

# 3 The compiled specification is the contract

The load-bearing idea of the architecture is that the policy is a compiled artifact, not a runtime behavior. Because the program is compiled ahead of time into one specification, the decisions that govern a run are made once, in the artifact, and enforced — by the engine in software, by the fabric in hardware — instead of being improvised by a heuristic scheduler while the run is in flight.

© 2026 Layer1Labs Silicon Inc.

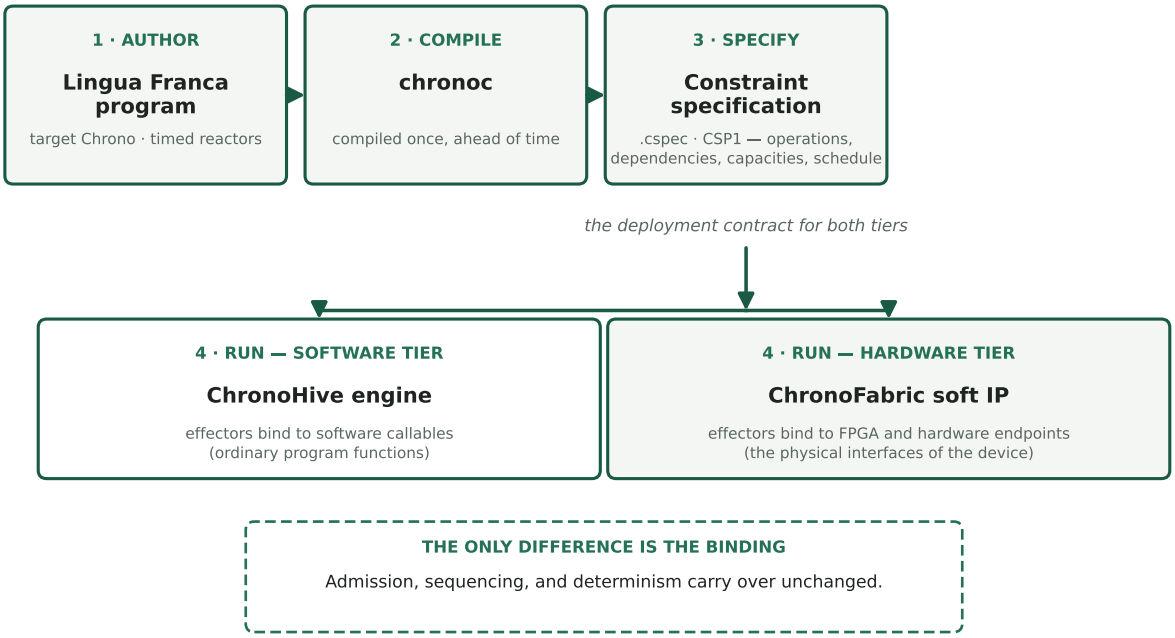



**Figure 1:** One program, two tiers. A Lingua Franca program is compiled once into a single constraint specification, which is the deployment contract for both tiers; only the effector binding differs. Source: layer1labs.ai/chronofabric/ and layer1labs.ai/chronohive/, "How It Fits."

That is what makes determinism possible on both tiers. The same program with the same inputs makes the same decisions, bit-for-bit, on every run, so any run can be replayed exactly — for debugging, for regression testing, or for audit. Layer1Labs holds a formal proof of the engine's determinism: its decisions are exact by construction, and that exactness is achieved with a state small enough to enforce in hardware — the property that lets the same guarantee cross to the fabric. The proof itself is unpublished; it is available to qualified partners and investors under NDA.

The specification being the contract also fixes the boundary of what a hardware customer receives: not a scheduler to tune, but the same compiled program their software tier already runs, with its admission, sequencing, and determinism guarantees stated once, in one artifact.

## 4 Enforcement in hardware

On ChronoFabric, the effectors of the compiled specification bind to FPGA and hardware endpoints, and the fabric issues the scheduled work itself. The published record makes the difference concrete: the hardware scheduler issues no per-packet work to the CPU at all. With all four CPU cores of the beta boards held at 100% load, ChronoFabric's scheduled departures still landed on the programmed tick, while departures scheduled from the CPU on the same boards degraded to millisecond-scale jitter. The scheduler is therefore saturation-proof in the specific sense that matters for orchestration: loading the host does not move the schedule.

Admission and sequencing read the same way in hardware terms. Admission is the fabric allowing only the work the compiled policy admits, in the order the policy specifies; sequencing is departures leaving at the exact programmed tick. The evidence below is stated the way the measurements were run; the full

© 2026 Layer1Labs Silicon Inc.

treatment, with every figure's basis, is the companion document *ChronoFabric Measured Results*.

- **Delivery.** In validation runs between two Kria FPGA boards over a 10-gigabit Ethernet link, ChronoFabric delivered more than 21,000 scheduled packets with zero missed departures and zero dropped — a 100% delivery rate, including a continuous 10,000-packet endurance run.
- **Departure precision.** Across 2,000-packet runs, the median and 99th-percentile departure interval were exactly the programmed 1 ms (1,000,000 ns), with jitter (standard deviation) as low as 61 nanoseconds — roughly $160\times$ tighter than the same boards scheduling from the CPU.
- **Adaptation without misses.** In a 10,048-packet run, the fabric's adaptive scheduler adjusted its own timing 53 times in response to on-chip constraint signals — and still delivered every packet: 10,048 of 10,048, zero misses.
- **Two boards, one tick.** In cross-board tests, two ChronoFabric boards fired their packet bursts at exactly the same hardware tick — a measured collective departure spread of 0 ns.
- **Timing at tighter budgets.** The timing mechanism was also validated against the tighter per-packet budgets that faster links would impose. On a single KR260 board over its native 1-gigabit connection, the budget was compressed to the cadence equivalents of 100-gigabit (10,000 ns), 400-gigabit (2,500 ns), 800-gigabit (1,250 ns), and 1.6-terabit (620 ns) links — 5,000 packets at each level, in quiet conditions and under full CPU stress — and at every level the schedule held exactly: 100% of packets on target, none stale. This validates the timing mechanism at those budgets; it is not a measurement of data transport at those line rates.
- **Verified in simulation.** The same budgets were also verified in simulation, against the gate design itself. In a cocotb regression suite — cocotb is the standard open-source Python framework for verifying chip designs, here driving the gate's design description in the GHDL simulator — the gate held its precision at every simulated link-rate budget from 1 gigabit to the 1.6-terabit equivalent: departure spread of zero at all rates, and zero missed gates at the tightest, 620-nanosecond budget. The suite's formal verification report (June 2026) records 45 of 46 tests passing, one skipped, and none failing. That is verification of the design in simulation — not a physical link at those rates.
- **Sustained, not just sampled.** The 24-hour run was completed for all three software configurations on a Kria KV260 board — 864 million timing cycles in each run. The full hardware profile passed with a worst-case scheduling latency of 22 microseconds (99.99% of cycles at or under 12 microseconds); the real-time software configuration on its own also passed, at a worst case of 33 microseconds (99.99% at or under 15 microseconds); and the stock configuration, measured for contrast, reached a worst case of 904 microseconds (99.99% at or under 25 microseconds) — the gap the real-time configurations close.

Every figure above was measured on the beta platform: Kria FPGA evaluation boards (KR260/KV260) over 10-gigabit links — with two exceptions, stated in their items: the timing-budget equivalence runs used a single board's native 1-gigabit connection, and the simulation item verified the design in a simulator, not on a board. Production-scale fabrics (100G/400G, ASIC) remain the roadmap, not a measurement.

## 5 Delivery: soft IP, ready for beta customers

ChronoFabric is delivered today as soft IP — a licensable hardware IP block — targeting Kria FPGA platforms. It is complete and ready for beta customers. There is no hardware purchase from Layer1Labs and

no separate product to integrate: the deliverable is the IP block that enforces, on the customer's Kria platform, the same compiled specifications the software tier runs.

The beta is open to organizations building or operating large-scale AI infrastructure — teams for whom synchronization waste across accelerators is a real cost, and who want deterministic enforcement evaluated on their own workloads. Participation is arranged directly with our team; there is no public download and no form to fill in. ChronoFabric, like ChronoHive, is proprietary technology of Layer1Labs Silicon Inc., offered for commercial licensing by written agreement.

- Beta and technical questions: info@layer1labs.ai
- Partnerships and licensing: licensing@layer1labs.ai

## 6 The staged roadmap

The hardware strategy is staged, and the stages are not equally established. Only the first stage is measured.

1. **FPGA evaluation platforms** — where ChronoFabric is today. Reconfigurable hardware that lets beta customers validate hardware enforcement in their own environments, delivered as soft IP targeting Kria platforms. Every measurement in this paper is from this stage.
2. **Production hardware** — dedicated devices that carry the fabric into production deployments.
3. **Embedded chiplets and merchant silicon** — the long-term path, so deterministic orchestration ships inside the platform itself.

Stages 2 and 3 — including 100G/400G fabrics and ASIC — are future offerings on this roadmap. They are not part of the measurements above, and no figure from the beta platform should be read as a figure for them.

## 7 Limits

What this architecture paper does not establish, collected in one place:

- **The measurements are beta-platform measurements.** All of them were run on Kria FPGA evaluation boards (KR260/KV260), between two boards or on one; the links were 10-gigabit Ethernet, except the timing-budget equivalence runs, which used one board's native 1-gigabit connection and validate the timing mechanism at compressed budgets only — they are not measurements at 100G, 400G, 800G, or 1.6T line rates. The gate design is additionally verified in simulation at those budgets; that verification is not a measurement either. Nothing here measures a production-scale fabric.
- **Each figure carries its own basis.** The delivery, precision, adaptation, and sustained-run figures come from different runs on the beta platform; they are stated with their bases and are not merged into a single platform claim.
- **The roadmap is a roadmap.** Production hardware, embedded chiplets, merchant silicon, and 100G/400G fabrics are planned stages. They carry no measurements in this paper or anywhere in our published materials.

© 2026 Layer1Labs Silicon Inc.

- **The software tier's results are not repeated here.** ChronoHive's measured training results have their own envelope (single-node, at most 2 GPUs) and are carried in full by the companion documents *ChronoHive Architecture Overview* and *ChronoHive Measured Results*. This paper claims only the published two-tier statement: the same programs, the same specifications, the guarantees carried over unchanged.
- **The determinism proof is unpublished.** It is available to qualified partners and investors under NDA; this paper relies on the measured record, not on the proof.

© 2026 Layer1Labs Silicon Inc.

# Notices

**Copyright.** © 2026 Layer1Labs Silicon Inc. All rights reserved.

**Sharing.** This document may be shared in its complete, unmodified form, with attribution to Layer1Labs Silicon Inc. No other reproduction, modification, adaptation, or derivative use — in whole or in part — is permitted without the prior written permission of Layer1Labs Silicon Inc.

**No license.** Nothing in this document grants, by implication or otherwise, any license or right under any patent, copyright, trademark, or other intellectual property of Layer1Labs Silicon Inc. Publication of this document is a disclosure of information only.

**Patents.** The technology described in this document is the subject of pending patent applications, including 9 provisional patent applications filed with the U.S. Patent and Trademark Office. No statement in this document is a representation as to the scope, validity, or enforceability of any application or of any patent that may issue from one.

**Trademarks.** Layer1Labs, ChronoHive, and ChronoFabric are trademarks of Layer1Labs Silicon Inc. Other names mentioned in this document may be trademarks of their respective owners; their use here is for identification only and claims no affiliation or endorsement. In particular, Lingua Franca is an open-source research project of its own community.

**Measurements and basis.** Figures in this document are drawn from the project's measurement records and are cited where they appear; where a figure is modeled rather than measured, it is labeled as modeled. This document is provided for information only, on an "as is" basis, without warranty of any kind, express or implied, including as to accuracy, completeness, or fitness for a particular purpose. It describes a demonstration system; it is not a specification, an offer, or a forecast of results in any other setting.

**Current version.** Documents in this family are revised in place under version numbers. The current version of this document is the one published at layer1labs.ai; earlier versions are superseded.