Layer1Labs Silicon

LAYER1LABS SILICON INC. · DOCUMENT

ChronoFabric Technical Brief

Deterministic orchestration enforced in hardware: what ChronoFabric is, how it is delivered, and the behavior measured on the beta platform.

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

This brief is for hardware evaluators. It states what ChronoFabric is, the problem it answers, and what has been measured on the hardware it runs on today — each figure with the run it came from, because a timing figure without its configuration is not a result an evaluator can use.

ChronoFabric is the hardware tier of the Layer1Labs architecture. The same programs and the same compiled specifications that run in the ChronoHive software engine are enforced on FPGA and hardware endpoints, with the same determinism guarantees. It is available now as soft IP — a licensable hardware IP block — targeting Kria FPGA platforms, and it is complete and ready for beta customers.

See it live: ChronoFabric, as published — `https://layer1labs.ai/chronofabric/`

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

Contents

1 The problem: timing that software cannot hold **3**

2 What ChronoFabric is **3**

3 Measured on hardware **4**
3.1 Delivery: every scheduled packet, delivered 4
3.2 Departure precision: packets leave at the exact tick 4
3.3 Under full CPU load, the tick still holds 4
3.4 Adaptive, and still exact 4
3.5 Two boards, one tick 4
3.6 Tighter budgets: equivalence at higher-rate cadences 5
3.7 Verified in simulation at the same budgets 5
3.8 Sustained over 24 hours 5

4 The beta platform and delivery **6**

5 The staged roadmap **6**

6 Evaluating ChronoFabric **6**

7 Limits, stated plainly **6**

Colophon **7**

Notices **8**

1 The problem: timing that software cannot hold

Packet systems live or die on departure timing. A packet that leaves late is not slightly worse than one that leaves on time; in a synchronized system it can miss its slot entirely, and the work it carried has to be done again or done without. The usual way to schedule departures — software running on a general purpose processor — has two weaknesses that no amount of tuning fully removes.

First, every scheduling decision costs processor time on the critical path: the processor has to notice the packet, decide when it leaves, and issue it, for every packet, forever. Second, software timing is at the mercy of everything else the processor is doing. Load the processor and the departures wander — typically into the millisecond range, thousands of times wider than the nanosecond precision the schedule was written to.

ChronoFabric answers both weaknesses by moving enforcement into hardware. The schedule is decided once, when the program is compiled, and the hardware issues each packet at its programmed tick. The hardware scheduler issues no per-packet work to the CPU at all.

The setting this matters most is the one Layer1Labs builds for: large-scale AI infrastructure, where synchronization waste across accelerators is a real, recurring cost. The outcome classes are the ones operators already plan around — less idle time across the fleet, smoother infrastructure-level energy demand, and more useful work per watt and per facility.

2 What ChronoFabric is

ChronoFabric is a deterministic orchestration architecture delivered as soft IP. *Deterministic* means the same program with the same inputs makes the same decisions, at the same ticks, on every run. *Soft IP* means it is delivered as a licensable hardware IP block that the customer integrates into their own FPGA designs — not as a finished board or a boxed product.

Programs are authored once, in Lingua Franca, and compiled by chronoc into a single constraint specification (.cspec). That specification is the deployment contract for both tiers of the architecture: the ChronoHive engine runs it in software, and ChronoFabric runs the identical specification on hardware.

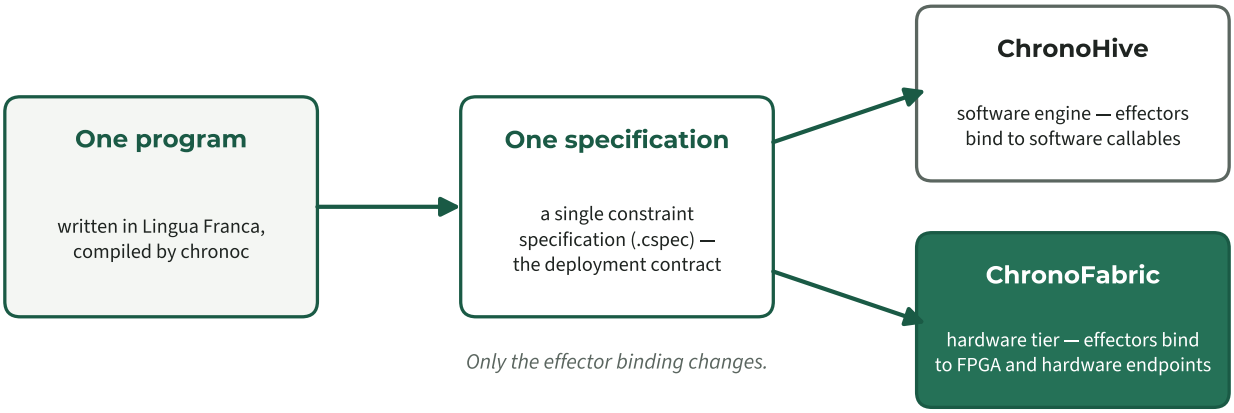


Figure 1: One program, one specification, two tiers. Only the effector binding changes between the software engine and the hardware tier. Source: the architecture as published at layer1labs.ai/chronofabric/.

The one thing that changes between tiers is the *effector binding*. 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 admission control (which work is allowed to proceed), the sequencing (the order work happens), and the determinism guarantees carry over from the software tier unchanged.

3 Measured on hardware

Every measurement in this section was made on Kria FPGA evaluation boards (KR260 and KV260) over 10-gigabit Ethernet links — the beta platform described in Section 4 — with one exception, stated where it appears: the timing-budget equivalence runs described below used a single board's native 1-gigabit connection. Production-scale fabrics (100G and 400G) and ASIC implementations are the roadmap, not a measurement, and no figure here comes from them. One subsection below reports no board figure at all: the gate design's verification in simulation at the higher-rate budgets, labeled there as what it is.

3.1 Delivery: every scheduled packet, delivered

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.

3.2 Departure precision: packets leave at the exact tick

Across 2,000-packet runs, the median (P50) and 99th percentile (P99) departure intervals were exactly the programmed 1 ms — 1,000,000 ns — with jitter (standard deviation) as low as 61 ns: roughly 160 times tighter than the same boards scheduling from the CPU.

3.3 Under full CPU load, the tick still holds

With all four CPU cores held at 100% load, ChronoFabric's scheduled departures still landed on the programmed tick — median and P99 exact — while CPU-scheduled departures on the same boards degraded to millisecond-scale jitter. This is the direct consequence of the design in Section 1: the hardware scheduler issues no per-packet work to the CPU, so CPU load has nothing to disturb.

3.4 Adaptive, and still exact

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. Adaptation, in other words, did not cost a single departure.

3.5 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.

© 2026 Layer1Labs Silicon Inc.

3.6 Tighter budgets: equivalence at higher-rate cadences

The runs above all work at a 1-millisecond cadence. The timing mechanism was also validated against the much tighter budgets that faster links would impose. On a single Kria KR260 board, over its native 1-gigabit connection, the per-packet timing 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, no measured spread against the target schedule. This validates the timing mechanism at those budgets; it is not a measurement of data transport at those line rates, and no figure in this brief comes from a 100G, 400G, 800G, or 1.6T link. The full result set, with every level's basis, is carried in the companion document, *ChronoFabric Measured Results*.

3.7 Verified in simulation at the same budgets

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; the full treatment, with every budget's basis, is carried in the companion document, *ChronoFabric Measured Results*.

3.8 Sustained over 24 hours

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 — the real-time stack (a PREEMPT_RT kernel with isolated CPUs) with the hardware profile — passed with a worst-case scheduling latency of 22 microseconds on the isolated core, 99.99% of cycles at or under 12 microseconds. The real-time software configuration on its own also passed: worst case 33 microseconds, 99.99% at or under 15 microseconds. 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. All three runs are on that one board; the figures are not platform-wide.

Result	Figure	Basis
Packets delivered	21,000+, 0 missed, 100%	Two boards, 10 Gb/s link
Endurance run	10,000 packets, continuous	Included in the runs above
Departure interval, P50 / P99	exactly 1,000,000 ns	2,000-packet runs
Departure jitter (StdDev)	as low as 61 ns	Same runs
Adaptive run	10,048 of 10,048; 53 adjustments	Single run, 0 misses
Cross-board spread	0 ns	Two boards, burst fire
Timing-budget equivalence	100% on target, 0 stale	One board, 1 Gb/s; budgets 10,000–620 ns
24-hour, full hardware profile	22 μs (99.99% $\leq$ 12 μs)	One KV260, 864M cycles
24-hour, real-time software only	33 μs (99.99% $\leq$ 15 μs)	One KV260, 864M cycles
24-hour, stock software	904 μs (99.99% $\leq$ 25 μs)	One KV260, 864M cycles

Table 1: The measured results as published, each with the run it came from. Source: layer1labs.ai/chronofabric/, "Measured on Hardware."

© 2026 Layer1Labs Silicon Inc.

4 The beta platform and delivery

ChronoFabric begins on FPGA evaluation platforms — reconfigurable hardware that lets beta customers validate hardware enforcement in their own environments. The measurements above were made on that platform: Kria FPGA evaluation boards (KR260 and KV260) with 10 Gb/s SFP+ links — the timing-budget equivalence runs excepted, which used one board's native 1-gigabit connection, as stated in Section 3.

Delivery today is as soft IP targeting Kria FPGA platforms: the customer licenses the IP block and integrates it into their own designs on those platforms. 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. Beta participation is arranged directly with the Layer1Labs team; there is no public download.

5 The staged roadmap

The published hardware strategy has three stages. Only the first exists today, and only the first has measurements.

1. **FPGA evaluation platforms** — available now, as soft IP targeting Kria platforms. This is the stage every figure in this brief comes from.
2. **Production hardware** — dedicated devices that carry the fabric into production deployments. Roadmap.
3. **Embedded chiplets and merchant silicon** — the long-term path: deterministic orchestration shipping inside the platform itself, including production-scale 100G/400G fabrics and ASIC implementations. Roadmap.

Nothing in stage two or three has been measured, and nothing in this brief should be read as a result for them.

6 Evaluating ChronoFabric

ChronoFabric, like ChronoHive, is proprietary technology of Layer1Labs Silicon Inc., offered for commercial licensing by written agreement. To join the beta or discuss a technical evaluation, write to **info@layer1labs.ai**. For partnerships and licensing, write to **licensing@layer1labs.ai**.

7 Limits, stated plainly

- Every measurement in this brief comes from the beta platform — Kria evaluation boards over 10-gigabit links. Production-scale fabrics (100G/400G) and ASIC implementations are roadmap and unmeasured — at those rates, the gate design is verified in simulation, not measured.
- The timing-budget equivalence runs were made on one board over its native 1-gigabit connection. They validate the timing mechanism at budgets compressed to higher-rate cadences; they are not measurements of data transport at 100G, 400G, 800G, or 1.6T line rates.
- The 24-hour sustained results are for three software configurations on one board. They are not platform-wide figures, and this brief does not present them as such.

© 2026 Layer1Labs Silicon Inc.

- The outcome classes named in Section 1 — throughput recovery, power avoidance, CapEx efficiency — are the benefits the architecture is designed to deliver. They are not measurements, and no fleet-level figure for them is published.
- This brief describes the architecture, the delivery model, and the published measurements. It does not describe the implementation, and it establishes nothing about stages two and three of the roadmap beyond their published intent.

Colophon

Sources: the product description and measurements as published at layer1labs.ai/chronofabric/ and the technical answers at layer1labs.ai/faq/, read on October 26, 2026. The timing-budget equivalence results and the three-configuration 24-hour results are from the program's completed experiment records (runs of April to October 2026); the simulation verification is from the program's verification report of June 2026. Every figure in this brief carries its configuration basis stated alongside it. Built with the Layer1Labs document template family (LuaLaTeX); the figure is script-generated. Version 1.0 is the first published edition of this brief.

© 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.