GAL-2 TECHNOLOGIES LLC

GAL-2 Time Contract Architecture

Application-boundary time governance for systems that already depend on UTC, GNSS, PTP, NTP, chrony, grandmasters, hardware clocks, or operating-system time.

Core claim

Your timing stack delivers time. GAL-2 governs whether software should consume it before committing application state.

Version posture: RC5.8 / 1.2.0-rc.3 - public technical preview and enterprise pilot candidate

Public architecture whitepaper
June 2026

Contact: ftorres@gal-2.com
globalalignmentlayer.com

Executive summary

Primary positioning

GAL-2 is not a new clock and does not replace UTC, GNSS, PTP, NTP, chrony, grandmasters, hardware clocks, or operating-system time. It adds an application-facing Time Contract before delivered time becomes committed application state.

Modern timing infrastructure can deliver excellent reference time to hosts, networks, and distributed systems. Applications still face a separate decision: whether a time value should be consumed before a state change is committed.

GAL-2 addresses that application boundary. It exposes a local Time Contract that returns a policy-derived consumption decision, a validity window, contract mode, monotonic sequence, and source lineage. The purpose is to make time consumption explicit, bounded, observable, and fail-closed when policy can no longer justify safe use.

The public status is intentionally bounded: **RC5.8 / 1.2.0-rc.3 is a public technical preview and enterprise pilot candidate, not final production certification.**

Document structure

#	Section	Purpose
1	Problem boundary	Time delivery vs. application time consumption.
2	Architecture	Existing timing stack → GAL-2 daemon → /contract → application decision.
3	Contract semantics	Fields, source lineage, and application behavior.
4	Endpoint and security boundary	Local endpoint scope, recommended hardening, and non-claims.
5	Policy states	Contract modes and protected application outcomes.
6	Evidence and claim boundaries	Public evidence, limits, and references.
7	IXOYE witness boundary	Advisory-only witness role and non-governance boundary.
8	Evaluation path	How a customer evaluator or design partner can characterize the boundary.

1. The problem: time delivery is not time consumption

Reference systems answer an essential question: *what time is it?* This is the domain of UTC, GNSS, PTP, NTP, chrony, grandmasters, time appliances, hardware clocks, and operating-system time.

Applications have another question: *should this delivered time be consumed before committing state?* That question becomes critical for databases, workflows, audit records, trading systems, ledgers, automation, and distributed services that convert timestamps into durable state.

Timing can become stale, degraded, ambiguous, discontinuous, outside policy, or in recovery. A host may continue to return a timestamp while the application has no explicit local policy for whether that timestamp should be consumed.

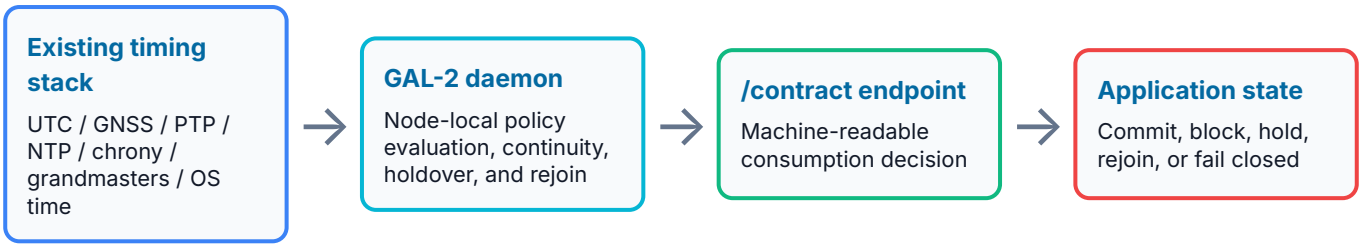
GAL-2 boundary

GAL-2 does not claim the reference layer is wrong. It adds a decision boundary for software that must decide whether delivered time is safe to consume before protected state changes.

Design principle

> **Precision protects the reference. GAL-2 protects the consumer.**

2. Architecture: a local contract before protected commits



Keep the timing stack. Add GAL-2 where delivered time becomes application state.

GAL-2 is deployed as a node-local governance layer beside the application. The existing timing stack remains in place. GAL-2 exposes a local contract endpoint that protected applications can check before time-sensitive commits.

Primary components

Component	Role
Existing timing stack	UTC, GNSS, PTP, NTP, chrony, grandmasters, hardware clocks, time appliances, and OS time continue to provide reference timing.
GAL-2 API	Provides governed upstream time state to the local evaluator model.
GAL-2 daemon	Runs node-local and evaluates freshness, policy, continuity, holdover, rejoin, and state transitions.
Local /contract endpoint	Returns a machine-readable consumption decision to applications.
Protected application path	Application checks the Time Contract before protected time-sensitive state changes.

3. The Time Contract decision object

The Time Contract is not only a timestamp. It tells software whether governed time is safe to consume, why, for how long, and under what source lineage.

Representative live `/contract` response:

```
{
  "safe_to_consume": true,
  "mode": "LIVE",
  "reason": "fresh_api_sync",
  "valid_until": "2026-06-24T16:12:41.847438Z",
  "monotonic_sequence": 12006,
  "source_lineage": [
    "gal2_api",
    "gal2_daemon_rc3_base",
    "rc4_72h_holdover_policy",
    "rc5_ixoye_witness_contract"
  ]
}
```

Field	Meaning
safe_to_consume	Policy-derived decision indicating whether the application may consume governed time for protected actions.
mode	Contract state such as LIVE, WARMING, HOLDOVER, REJOIN, or FAIL_CLOSED.
valid_until	Validity window for the current contract response.
monotonic_sequence	Observable sequence associated with served contracts; used to detect backward sequence behavior in evidence.
source_lineage	Visible lineage describing the governed state path.
reason	Human-readable reason for the current decision state.

Lineage boundary: Presence in `source_lineage` records the observed contract path, including advisory witness surfaces. It does not imply governance authority. IXOYE does not decide `safe_to_consume`.

4. Endpoint and security boundary

The initial evaluator surface is a local application-facing endpoint. Security posture must be explicit: GAL-2 governs time consumption policy; it is not a replacement for host hardening.

Boundary	Statement
Endpoint scope	The local Time Contract endpoint is intended for node-local protected application paths.
Recommended hardening	Recommended: harden local endpoint (Unix domain socket / local ACLs). Restrict process identity and local access.
Host compromise	Host compromise is out-of-scope for contract guarantees. GAL-2 is not a host security product.
Application bypass	If an application bypasses `/contract` and reads raw system time directly, that path is outside GAL-2 protection.

Product boundary: GAL-2 makes time consumption policy observable before protected application state changes.

5. Policy states and application outcomes

`safe_to_consume` is derived from policy and observed state. The application does not silently fall back to raw system time when the contract is no longer safe.

Mode	Meaning	Application outcome
LIVE	Fresh upstream state within policy.	Application may consume governed time if other application checks pass.
WARMING	Local daemon not ready or insufficient trusted state.	Application should wait or use a non-protected path.
HOLDOVER	Bounded continuity under declared policy and uncertainty budget.	Application may continue only while the contract remains consumable.
REJOIN	Controlled recovery after degraded or disconnected state.	Application should consume only if policy permits rejoin conditions.
FAIL_CLOSED	Unsafe or expired policy window.	Protected actions should not commit unsafe time as state.

Boundary rule: The clock does not stop. The protected application path stops unsafe time consumption from becoming committed state.

6. Evidence and claim boundaries

The evidence strategy separates long-duration continuity artifacts from Time Contract policy-boundary artifacts. This prevents overclaiming and makes evaluation easier for timing experts.

Artifact	What it supports
RC4 120-hour Time Contract characterization	Declared policy boundary, documented FAIL_CLOSED boundary row, 14,279 contract samples, 0 observed gal2_time backward steps, 0 observed monotonic_sequence backward steps, clean LIVE_RESTORED recovery.
Solstice Run 7-Day dataset	Earlier long-duration timing stability and continuity evidence. Useful as historical support, not the primary Time Contract boundary artifact.
Red Light Test	Application behavior: raw path may continue while contract path blocks unsafe commits.
RC5.1 Public Evaluator Bundle	Evaluator packaging and distribution artifacts for the Time Contract preview path.

Claim boundary: No metrology claims; not a UTC replacement; not a production SLA. RC5.8 inherits RC4-derived 72h policy language; it does not claim a new 72h wall-clock run of final RC5.8 artifacts.

Public references

Reference	URL
RC4 120-hour Time Contract characterization	https://doi.org/10.5281/zenodo.20582981
Solstice Run 7-Day dataset	https://doi.org/10.5281/zenodo.18018704
Red Light Test video	https://youtu.be/Dn8VCpW_NAs
RC5.1 Public Evaluator Bundle	https://doi.org/10.5281/zenodo.20646973

7. IXOYE witness boundary

IXOYE may appear in source lineage or dashboard views as an out-of-band witness layer. Its role is intentionally limited.

IXOYE is	IXOYE is not
Advisory-only out-of-band observer.	Not a source of time.
A coherence/witness view that can support operational awareness.	Not a fallback path.
Separate from the policy decision.	Not the authority that governs `safe_to_consume`.
Useful for observation and future attestation work.	Not required for the application to consume the Time Contract.

Operational rule: GAL-2 governs. IXOYE observes. IXOYE is advisory-only and does not decide `safe_to_consume`.

8. Evaluation path

The recommended next step is not broad adoption. It is one controlled customer-evaluator or design-partner characterization beside an existing timing stack.

Evaluation setup

- Existing timing stack: PTP, NTP, GNSS, chrony, grandmaster, time appliance, or equivalent host timing infrastructure.
- GAL-2 daemon running node-local beside one representative application or workflow.
- Application or harness checks `/contract` before protected time-sensitive state changes.
- Controlled impairment scenario designed and documented in advance.

Success criteria

- Capture `/contract` logs and mode transitions.
- Verify `monotonic_sequence` does not step backward across the run.
- Observe expected LIVE, HOLDOVER, REJOIN, or FAIL_CLOSED behavior under declared policy.
- Produce a bounded evidence package with raw logs, timestamps, hashes, and scope boundaries.
- Document what is proven, what is not proven, and what remains future work.

Evaluation ask: One design partner or customer evaluator to run GAL-2 Time Contract alongside an existing timing stack and evaluate application-boundary governance under controlled impairment.

Conclusion

GAL-2 does not replace what existing timing stacks already do. It adds a contract at the application boundary where delivered time becomes committed state.

The initial public posture is intentionally bounded: RC5.8 / 1.2.0-rc.3 is a public technical preview and enterprise pilot candidate. The goal is external characterization, not an unbounded production certification claim.

> **Core closing statement:** Raw time keeps going. GAL-2 knows when to stop.

Contact

Francisco Edgardo Torres Alvarado
Founder, GAL-2 Technologies LLC
ftorres@gal-2.com
globalalignmentlayer.com