

Converge

THE TRUSTWORTHY AGENT RUNTIME

The Problem: Agent Drift

AI agents are transforming business automation. But they have a fatal flaw:

- **Non-deterministic** — Same input, different outputs
- **No guarantees** — Workflows "vibe" but don't enforce rules
- **Unauditable** — "Why did it do that?" has no answer
- **Fragile** — Silent failures, partial updates, stuck workflows

Enterprises can't trust agents with critical business processes.

The Market Opportunity

Segment	Pain Point	Spend
Business Automation	Zapier/Make don't guarantee correctness	\$13B by 2030
AI Agent Platforms	No trust layer for LLM orchestration	\$28B by 2028
SMB Operations	12+ SaaS tools, zero coherence	\$200B annually

The gap: **No platform guarantees AI agents behave correctly.**

Today: The SaaS Spaghetti

Every SMB manages a fragmented stack:

Category	Tools	Problem
CRM	HubSpot, Salesforce, Pipedrive	Data silos
Finance	Xero, QuickBooks, Stripe, Brex	Manual reconciliation
HR	Gusto, Rippling, BambooHR	Policy gaps
Delivery	Asana, Monday, Notion, Jira	Status chaos
Marketing	Mailchimp, Intercom, Hootsuite	Channel fragmentation
Comms	Slack, Teams, Gmail, Calendly	Context lost
Automation	Zapier, Make, n8n	Brittle glue

15-20 tools. Zero guarantees. Constant firefighting.

Tomorrow: Declare Truths, Let Converge Work

Instead of wiring tools together, **declare what must be true:**

```
Truth: Get paid for delivered work
Given work is marked complete
Then invoice is issued within 24 hours
And payment is tracked
And overdue triggers escalation
```

```
Truth: No lead left behind
Given a qualified lead exists
Then owner is assigned within 4 hours
And first contact happens within 24 hours
```

Converge handles the orchestration. You own the business logic.

The Solution

MATHEMATICAL GUARANTEES FOR AGENT BEHAVIOR

Converge: The Trustworthy Runtime

A formally-grounded execution engine where:

- **Agents propose** — LLMs suggest, never decide
- **Engine validates** — Invariants enforce business rules
- **Context converges** — Every run reaches a stable, explainable state

"Stop agent drift. Converge to an explainable result."

The 9 Axioms

Mathematical foundations that make Converge trustworthy:

#	Axiom	Formula
1	Monotonicity	$\text{ctx} \subseteq \text{step}(\text{ctx})$
2	Determinism	$\text{step}(\text{ctx}) = \text{step}(\text{ctx})$
3	Idempotency	$\text{agent}(\text{ctx}) = \text{agent}(\text{agent}(\text{ctx}))$
4	Commutativity	$a(b(\text{ctx})) = b(a(\text{ctx}))$
5	Termination	$\exists n: \text{step}^n(\text{ctx}) = \text{step}^{n+1}(\text{ctx})$

The 9 Axioms (continued)

#	Axiom	Formula
6	Consistency	$\neg \exists(f, \neg f) \in \text{ctx}$
7	Starvation Freedom	$\text{enabled}(a) \Rightarrow \diamond \text{runs}(a)$
8	Confluence	$\text{ctx}_1 \cup \text{ctx}_2 \rightarrow \text{ctx}^*$
9	Observability	$\forall \text{effect}: \text{logged}(\text{effect})$

These aren't guidelines — they're **guarantees enforced by the type system**.

From Math to Business Value

Axiom	Business Guarantee
Monotonicity	Audit trail integrity — nothing lost
Determinism	Reproducible decisions — debug anything
Idempotency	Safe retries — no double-charges
Termination	Guaranteed completion — no stuck workflows
Consistency	No contradictions — ledgers balance
Observability	Full explainability — compliance ready

How It Works

```
Root Intent
  ↓
Fan-out (agents propose)
  ↓
Validate + enforce invariants
  ↓
Serial commit (append-only Context)
  ↓
Repeat until fixed point
  ↓
Explainable result + audit trail
```

Every step logged. Every decision traceable.

The Product Stack

Layer	What It Is	Revenue Model
Platform	Core runtime engine	Open source
Packs	Reusable business truth modules	Subscription
Blueprints	Curated compositions	Marketplace
Apps	Deployed tenant experiences	Enterprise

Packs: Install Business Truth

Pre-built modules for common operations:

- Money — Invoice, collect, reconcile, close books
- Customers — Capture, qualify, engage, close, handoff
- Delivery — Scope, track, complete, learn
- People — Hire, onboard, pay, offboard
- Trust — Access, audit, compliance

Each Pack inherits all 9 axioms automatically.

Competitive Landscape

Approach	Problem	Converge Difference
Zapier/Make	No guarantees, brittle chains	Axiomatic correctness
LangChain/AutoGPT	Agent drift, no convergence	Fixed-point execution
Custom code	Expensive, hard to audit	Declarative Truths
Enterprise BPM	Heavyweight, no AI-native	LLM-first, lightweight

We're the **trust layer** the AI stack is missing.

Go-To-Market: Builder-First

Phase 1: Builders as Channel (Now)

Win startups and agencies who build for clients.

They become our distribution to end customers.

Phase 2: Partner Ecosystem (6-12 months)

Builders deploy Converge to their customers.

We support with Packs, training, and co-marketing.

Phase 3: Platform Network (12-24 months)

Partner-built Packs in marketplace.

Converge becomes the standard runtime.

Traction

- **Core engine:** Proven convergence semantics in Rust
- **5 Packs:** Money, Customers, Delivery, People, Trust
- **Toolchain:** Truth-to-Rust compiler, validation suite
- **Early design partners:** [Placeholder]

Partner Strategy: Win or Lose

Partners are not a nice-to-have — they're **the strategy**.

Partner Type	Role	Value
Startups/Builders	Early adopters, feedback	Channel to their customers
System Integrators	Implementation	Enterprise credibility
Vertical ISVs	Domain Packs	Market expansion

Builders who trust Converge become our distribution channel.

Community-Led Growth

Marketing through **education**, not ads:

- **Meetups** — Local builder communities
- **Hackathons** — Build Packs, win prizes
- **Rewards** — Contributor recognition program
- **Certification** — Converge Builder credential
- **Content** — Tutorials, case studies, live streams

Goal: Builders recommend Converge to their clients.

The Ask

Raising: \$X seed round

Use of funds:

Category	Allocation	Focus
Community & Marketing	40%	Education, meetups, hackathons, brand
Engineering	35%	Core team, Pack development
Partnerships	25%	Builder program, integrator network

Goal: 50+ active builders, 10 partner-deployed customers

Converge

STOP DRIFT. START CONVERGING.

Contact

Kenneth Pernyer — [@pernyer](#) · Twitter · GitHub

Aprio One AB — [aprio.one](#)

Hey SH — [hey.sh](#)

Converge — [converge.zone](#)

Packages: [crates.io/converge-core](#) · [hex.pm/converge_ledger](#)