

Don't DIY your event store.

Beyond the prototype: the real cost of building vs. buying event infrastructure.

Event-driven architecture is how enterprises get auditability, real-time responsiveness, and AI-ready data out of their core systems. When the initiative is approved, a familiar proposal follows: *our engineers can build the event store themselves on general-purpose tools*. The proposal usually prices the prototype. It rarely prices the product. An event store is not a database table with an insert-only rule; it is a distributed system that must replicate, order, and serve events reliably, with its own recovery, evolution, scaling, and operational demands. Teams that build one take on that scope permanently, funded from the budget that was meant to deliver business outcomes. The comparison below lays out the full commitment, so the decision is made on total cost of ownership.

Axon Framework is open source under Apache 2.0, so any team can build on it. What follows is about the event store decision only.

CATEGORY	BUILDING IT YOURSELF	WITH THE AXONIQ PLATFORM
DIRECT BUILD COSTS		
Event store engine	Built and hardened from scratch: typically two to four quarters of senior engineers before the first business feature ships.	A purpose-built event store, production ready on day one.
Recovering from errors	Rebuilding a corrupted projection from history depends on replay tooling your team has to build and harden itself.	Rebuild from complete history without losing operations, hardened by 15+ years in production.
Evolving the system	Every change to event modeling is planned by your team with no tested pattern; getting it wrong risks years of stored history.	Tested event-versioning patterns that stay consistent as the platform evolves, so replay never breaks.
LIFETIME OWNERSHIP COSTS		
Platform operations	Scaling, monitoring, tuning, and patching a distributed system, staffed by your team indefinitely as history grows into the billions.	Delivered and supported as part of the platform, kept current by AxonIQ.
Upgrades and migrations	Internal versions, dependency upgrades, and data migrations keep landing on the backlog for as long as the system runs.	Documented upgrade paths and vendor-supported migrations, release after release.
Disaster recovery	Backups, restores, and corruption recovery are yours to design, test, and validate, without vendor support when it goes wrong.	Backup, recovery, and consensus-based high availability, tested and backed by AxonIQ support.
Compliance evidence	Query and access-control tooling has to be built per audit, by the same team that built the store.	A complete, immutable record of business history plus platform-level access control. Audits are answered from the system itself.
Technical debt	Bespoke infrastructure ages into the system nobody wants to touch; every architecture decision afterward works around it.	A product that keeps pace with the ecosystem, so the platform never becomes the reason you cannot move.

PEOPLE AND ORGANIZATION

Morale and retention	Engineers who joined to ship products end up on call for internal plumbing; backfilling a senior departure carries real recruiting and ramp cost.	Engineers work on business problems and build skills in a widely adopted framework, helping retention and recruiting.
Hiring and onboarding	No candidate arrives knowing your internal event store; everyone ramps through the original builders and tribal knowledge.	An established talent pool already works with Axon Framework; documentation and training shorten ramp time for everyone else.
Management overhead	Architecture reviews, planning cycles, prioritization battles, and incident response consume attention that should go to the product.	Infrastructure decisions are Axoniq's to make, not a standing agenda item.
Key-person risk	The system is documented in the heads of the two or three engineers who built it; attrition becomes an outage risk.	A supported product with documentation, training, and a global user community.

OPPORTUNITY COST

Roadmap displacement	Every engineer maintaining infrastructure is one less engineer shipping revenue-generating software.	Engineers stay on revenue-generating work from the first sprint.
Time to production	Commonly a year or more before the platform is trusted with real workloads.	Teams start building business features in week one.
Strategic flexibility	The DIY platform becomes another internal product competing for funding every year.	Infrastructure evolves with vendor investment while your budget stays focused on differentiating the business.

QUESTIONS TO ASK DURING ANY INTERNAL BUILD PROPOSAL

- 01 What if we spend a year building it?
- 02 What if it still doesn't satisfy compliance?
- 03 What business initiatives are delayed while engineering builds infrastructure?
- 04 What's the five-year TCO next to buying?

WHY TEAMS STANDARDIZE ON AXONIQ

A purpose-built event store, the Axoniq Framework, and the operational tooling to run both, delivered and supported as a single platform with a vendor-funded roadmap.

Budget pointed at outcomes

Faster time to value

Predictable TCO

Transferred risk

Organizational flexibility

The question is not whether your organization can build event infrastructure. It's whether you can justify the time, cost, and risk when the business outcomes are needed now.

Organizations choose the Axoniq Platform to avoid that bet: proven event infrastructure that lets teams deliver business value immediately, trusted in production by 96% of the Fortune 100.

EXPLORE THE PLATFORM · AXONIQ.IO

 Or [book a working session](#) to walk through build vs. buy for your architecture.