

Making the Case for Axoniq Framework

A practical guide for teams running Axon Framework 4 on how to reduce risk, save engineering time, and get your managers on board.

CONTENTS

01	Executive Summary	03
02	Where Axon Framework 4 Teams Stand Today	04
03	What Is Axoniq Framework?	05
04	What Axoniq Framework Solves	05
05	The Axoniq Platform: What the License Buys	08
06	The Foundation You Inherit: Axon Framework 5	09
07	A Migration Designed to Be Estimated	10
08	Making the Case to Your Manager	11
09	Manager FAQ and Response Scripts	12
10	Recommended Next Steps	14

WHO THIS IS FOR

Engineers who already understand why the move matters and need to make the case upward, plus the engineering leaders who will approve it.

Executive Summary

Axoniq has built event-driven infrastructure for real-time critical systems for 14 years, since before “agentic” was a word. Its open source Axon Framework has been in production since 2009 with more than 90 million downloads, and 96% of the Fortune 100 rely on its event-driven architecture, from retail at scale to regulated banking.

2009

In production since

90M+

Framework downloads

96%Of the Fortune 100

Axon Framework 4 reached end of life on **June 30, 2026**. Teams still running it are operating without active maintenance, and every quarter on an unmaintained framework adds security, compliance, and hiring risk that compounds quietly until it becomes urgent.

For production systems, the path forward is Axoniq Framework: the commercial offering built on the Axon Framework 5 foundation. It packages the components production deployments depend on: Axon Server connectivity, distributed messaging, integration test tooling, and field-level data protection. It also comes with the Axoniq Platform, an operations layer that gives teams handler-level performance insight, event processor management with automatic scaling, dead-letter queue management, alerting, and automatic Axon Server clustering.

Underneath all of it sits the most significant framework evolution in the product's history: an async-native core, Dynamic Consistency Boundaries, and domain code free of framework annotations. Taken together, this is the moment event sourcing gets dramatically simpler. The pattern has always offered capabilities nothing else can match: a complete, immutable history of everything the business has done. What it demanded in return was ceremony, specialist knowledge, and homegrown operational tooling. Axoniq Framework closes that gap.

This paper is written for engineers who understand why the move matters and need to make the case upward. It translates Axoniq Framework capabilities into outcomes your manager cares about: reduced risk, lower operating cost, predictable scaling, and a codebase engineers want to work in. It also explains why migration is a managed, well-documented process rather than a rewrite.

It is rare that the safe choice and the exciting one are the same choice. This is one of those times.

Where Axon Framework 4 Teams Stand Today

Axon Framework 4's end of life reflects a deliberate choice about where Axoniq invests its development effort, not a lapse in support. Enterprise features that require significant ongoing maintenance are moving to the commercial Axoniq 5 Framework, while the core framework remains open source and production-capable, an open core model built to keep development sustainable.

You can stay on 4.x, upgrade to Axon Framework 5 open source, or add Axoniq Framework for enterprise features. But for teams staying on 4.x, end of life is not an abstract milestone: it means no further releases, security patches, or compatibility updates as the Java ecosystem moves forward around your application. For engineering leaders, that translates into three concrete considerations.

Security	Unmaintained dependencies increasingly require additional review and explanation, especially in regulated industries.
----------	---

Operations	Workarounds accumulate as surrounding infrastructure evolves, adding maintenance effort without adding business value.
------------	--

Talent	Developers want to build on current technology, and modern platforms make it easier to attract and retain engineers.
--------	--

If your system connects to Axon Server, distributes commands and queries across instances, or handles regulated data, Axoniq Framework is the tier built for you. The rest of this paper explains what it includes, what the license buys beyond the code, and how to turn that into a proposal your manager can approve.

Three paths, one decision

Stay on 4.x and absorb the risk; move to Axon Framework 5 open source and keep building the operational tooling yourself; or adopt Axoniq Framework and get the production components and the platform with it.

What Is Axoniq Framework?

Axoniq Framework is the commercial artifact, published under [io.axoniq.framework](#), that extends the open-source Axon Framework 5 with the production components enterprise systems rely on. It is a product built directly on top of Axon Framework 5. The Axoniq Framework BOM includes everything in the open-source BOM, and the Axoniq Spring Boot Starter is a drop-in replacement for the open-source starter, which means adoption is a matter of swapping dependencies rather than rewriting code.

It also represents something bigger than a packaging decision. Event sourcing earned its reputation as the architecture for systems that cannot afford to lose history, but it also earned a second reputation alongside it as powerful but demanding. Axoniq Framework is the clearest statement yet that the second reputation is obsolete, because every component in it exists to remove a piece of friction that event-sourced teams used to absorb as the cost of doing business.

Production use requires a license obtained through the Axoniq Platform, and this is where the commercial tier earns its keep, since the license is attached to an operations platform rather than just a legal permission.

TWO DETAILS FOR THE BUDGET CONVERSATION

- During evaluation, Axoniq Framework modules start without credentials and simply log a warning, so teams can validate the entire migration before making any commercial commitment.
- Smaller organizations may be eligible for complimentary access to Axoniq Framework components, which is worth raising with the Axoniq team early.

What Axoniq Framework Solves

The strongest case for Axoniq Framework is not its feature list. It is the set of problems your organization is living with right now. Each problem below appears differently depending on the role: engineers experience it in daily development, managers own the operational impact, and the business ultimately carries the cost.

Production depends on a component nobody supports

Axon Server is AxonIQ's dedicated event store and message routing infrastructure, purpose-built to handle the event streaming, distribution, and querying that Axon Framework applications rely on. The connection between your application and Axon Server is the single most critical link in the system: it is where your events live and where your commands and queries get routed through.

The team feels this at 2 a.m., when an incident touches the connection layer and there is no one to escalate to, just source code and a forum. The manager feels it the next morning, explaining an outage to leadership with no vendor accountability and no answer to the question of who is responsible for the fix. The business feels it as unquantified exposure: revenue and reputation resting on a component with no support contract behind it, a fact that surfaces at the worst possible moments (audits, due diligence, and postmortems).

The power is in the combination: the framework cleanly models the business, and Axon Server guarantees the resulting event history is complete, tamper-proof, and instantly queryable.

When growth outpaces your architecture

Every successful product eventually outgrows a single instance, and in most systems that moment triggers a scaling project. Your team feels it as stolen sprints: capacity work displaces the roadmap for a quarter while routing, coordination, and consistency get reworked by hand. Your manager feels it as a difficult tradeoff between delivering new capabilities and investing in the architecture required to support growth. The business feels it when demand arrives and the system requires additional engineering work before it can respond.

WHAT AXONIQ FRAMEWORK CHANGES

The distributed messaging module, with `DistributedCommandBus` and `DistributedQueryBus`, routes commands and queries across instances as a built-in capability. Scaling becomes an operational adjustment: add instances, and the routing follows. The quarter-long capacity project becomes a configuration change, and growth becomes purely good news.

Green builds that still break in production

Teams that test against mocks are testing a story about their infrastructure, not the infrastructure itself. The team feels it as on-call fatigue and a creeping distrust of their own green builds, because the defects that hurt are exactly the ones mocks cannot catch. The manager feels it in incident reviews that keep tracing back to integration behavior, and in the erosion of confidence every time a clean release breaks in production. The business feels it where it hurts most: customers discover the defects first.

WHAT AXONIQ FRAMEWORK CHANGES

The Axon Server Testcontainer runs integration tests against a real Axon Server instance in the pipeline. Defects surface where they are cheapest (before release), and a green build starts meaning what everyone assumed it meant. It is one of the least expensive reliability investments available, and it arrives packaged and maintained.

Immutable events meet the right to erasure

Event sourcing's greatest strength, an immutable record of everything, collides directly with data protection obligations such as erasure requests. The team feels it as an assignment no application developer should get: build production cryptography in-house, and get it right. The manager feels it as owning compliance responsibility in an area that few application teams are equipped to implement and maintain themselves. The business feels it as regulatory exposure, and as additional scrutiny during security reviews when data protection depends on internally maintained capabilities.

WHAT AXONIQ FRAMEWORK CHANGES

The data protection module provides field-level cryptographic protection of event data through annotations, with support for multiple crypto engines and modern Java records. Compliance obligations are met with a supported product capability, security reviews get a product name instead of an apology, and for finance, healthcare, or any GDPR-scoped business, this problem alone can justify the license conversation.

The Event Store is the Foundation for AI

Agentic infrastructure needs durable, queryable context to reason from safely. The event store already provides exactly that: a complete, immutable, structured record of every decision the system has made. No separate data pipeline is required. Teams investing in Axoniq Framework today are investing in the event store as the infrastructure the AI systems will run on tomorrow.

The Axoniq Platform: What the License Buys Beyond the Code

The Axoniq Framework license runs through the Axoniq Platform, and the platform is where the day-two value lives. Every capability here operates on the same asset the framework already gives you.

For teams that have created their own toolbox to deal with the operational complexity of event-sourced applications (replays of event processors, scaling with load, performance monitoring and debugging), this is the section that lands hardest: the operational work that used to be the challenge of running event sourcing in production is now a product. It is the base of operations for Axon Framework and Axon Server, designed around the questions operations and engineering teams actually ask.

Framework metrics and handler-level insight

See the health and performance of any part of the application, down to individual message handlers, so bottlenecks are found with data instead of guesswork.

Event processor management

Start, stop, and reset event processors from the platform, and scale them automatically as load changes.

Dead-letter queue management

Retry or delete dead-lettered items without hand-built tooling, turning error recovery into a routine operation.

Monitoring and alerting

Set up alerts for both application and Axon Server health so the team hears about problems before customers do.

Automatic clustering and license rotation

Axon Server cluster formation is handled for you, and license files rotate automatically, removing two recurring sources of operational toil.

Development Agent

Build Axon Framework 5 applications from a prompt in a structured way, and explore shared public projects, which accelerates onboarding and prototyping.

A useful detail for the budget conversation: many platform features are available on the Individual plan free of charge, with no credit card required. Teams can experience the operational model before committing, which lowers the perceived risk of the whole proposal.

The management translation: most organizations end up building fragments of this themselves (dashboards, runbooks, in-house tooling, and scaling procedures). The platform replaces that invisible internal project with a supported product, and the cost of the license should be compared against the engineering time currently spent maintaining those fragments.

Once a team operates through the platform day to day, migrating fully onto Axoniq Framework is a continuation of how they already work, not a separate adoption decision.

The operational work that used to be the challenge of running event sourcing in production is now a product.

06 / THE FOUNDATION

The Foundation You Inherit: Axon Framework 5

Everything in Axoniq Framework stands on Axon Framework 5, the most significant evolution in the framework's history, and those gains come along automatically. The async-native core replaces ThreadLocal context handling with an explicit [ProcessingContext](#), eliminating whole categories of concurrency bugs and working cleanly with imperative Java, Project Reactor, Kotlin coroutines, and virtual threads.

Dynamic Consistency Boundaries (DCB) free system design from the fixed aggregate, letting teams define consistency regions that match the business domain and ship in parallel without coordinating through shared aggregates. Domain entities can be plain Java objects with zero framework annotations, so domain tests become ordinary unit tests and business logic stays readable by any Java developer.

A redesigned Configuration API supports annotations, builders, or pure Java, with or without Spring. And a rebuilt extension ecosystem, including PostgreSQL event storage and enterprise-grade Apache Kafka distribution, ships as first-class citizens of the new architecture rather than afterthoughts.

For the manager's conversation, the point is simple: adopting Axoniq Framework is not choosing between commercial components and framework modernization; it is both in one move. An immutable event history is the foundation agentic infrastructure needs to operate on this data safely; that is what makes this migration AI-readiness work, not just a framework upgrade.

When domain classes are plain Java, tests are ordinary tests, configuration fits the project, and operations come from a platform, event sourcing stops being a specialist discipline and starts being simply how the team builds.

07 / MIGRATION

A Migration Designed to Be Estimated

The Axoniq engineering team publishes a comprehensive migration guide covering the full journey from Axon Framework 4, including a dedicated path for moving onto the Axoniq Framework modules.

OpenRewrite-based recipes mechanically rewrite the predictable parts of an Axon Framework 4 codebase. The documentation is honest about the boundaries: the recipes do not guarantee a compiling project, and the remaining compile errors, together with markers left in the source, become a guided checklist for the manual steps. Axoniq also publishes early migration skills for AI coding agents in the same public agent-skills repository.

Documented migration paths cover each component individually: messages, aggregates, projectors and event processors, the event store, snapshotting, test fixtures, serializers, dead-letter queues, interceptors, configuration, and sequencing policies. Teams can scope component by component and sequence the work incrementally.

THE RISK THAT ISN'T THERE

Code conversion touches only source and build files. Event store data and token tables are never modified, which removes the scariest category of migration risk entirely.

The final step onto Axoniq Framework itself is the lightest of all: dependency swaps and package renames, with field names and method signatures unchanged.

Making the Case to Your Manager

Technical arguments rarely win budgets on their own. The framing on the next page maps each capability onto the outcome language that engineering and business leaders respond to.

RISK

Axon Framework 4 is end of life. Axoniq Framework puts the most critical components of the system, including the connection to the event store, on a commercially maintained and licensed footing. The move is the risk-reduction option, not the risky one.

OPERATIONS

The Axoniq Platform replaces homegrown dashboards, in-house tools and scaling runbooks with supported product capabilities, including automatic processor scaling and Axon Server clustering. Compare the license against the engineering hours currently spent on that invisible internal tooling.

COMPLIANCE

The data protection module gives regulated businesses a supported, field-level answer to data protection obligations in an event-sourced system, replacing what would otherwise be an in-house cryptography project.

SCALE

Distributed messaging makes horizontal scaling an operational adjustment rather than an architecture initiative, so capacity keeps pace with the business without heroics.

COST

Testing and non-production environments run on the Individual plan, and monthly pricing options are available for flexibility.

VELOCITY

The Axon Framework 5 foundation brings async-native concurrency, Dynamic Consistency Boundaries, and framework-free domain code, so teams ship in parallel with fewer defects and faster tests. That same permanent, structured event history is what AI agents need to operate on production data safely.

ONE PRACTICAL SUGGESTION

Anchor the conversation on the cost of doing nothing.

The migration has a bounded, estimable cost. Remaining on an unmaintained framework creates an ongoing maintenance burden that grows over time and may eventually require work under less favorable conditions.

Manager FAQ and Response Scripts

These are the questions internal champions hear most often. Each response is written so you can use it nearly verbatim. Adapt the specifics to your system, but keep the structure: acknowledge the concern, answer it with a fact, and end on the decision you are asking for.

“Why take on a framework migration now? We have a full roadmap already.”

The timing is not ours to choose. Axon Framework 4 reached end of life on June 30, 2026, which means we are already accumulating unpatched risk on the framework that runs our core system. The migration has a bounded, estimable cost that I can scope by running the automated conversion recipes against one service. Staying put has an unbounded cost that grows every quarter and will eventually be paid under worse conditions, probably during an audit finding or a forced dependency upgrade. I am asking to schedule bounded work now instead of unbounded work later.

“Why does this cost money? Axon Framework has always been free.”

The open-source framework is still free, and everything we would build on remains open source. What is commercial is the production tier: the Axon Server connector, distributed messaging, the integration test container, and the Axoniq Platform that operates it all. Those are exactly the components our uptime depends on, and the license puts them on a maintained, supported footing instead of a community dependency. The honest comparison is not license versus free. It is license versus the engineering hours we currently spend on homegrown dashboards, retry scripts, and scaling runbooks that the platform replaces.

“How do I know the migration will not blow up like these projects usually do?”

Three reasons this one is different. First, the mechanical work is automated: OpenRewrite recipes rewrite the predictable parts of the codebase, and what remains becomes a countable checklist of compile errors and source markers, so the estimate is based on evidence rather than optimism. Second, the migration guide documents a path for every component individually, so we can sequence the work incrementally instead of committing to a big-bang cutover. Third, and most importantly, code conversion never touches event store data or token tables. Our data, the part that actually cannot be recovered from a bad migration, is not in scope at all.

“Can we just use the open-source Axon Framework 5 and skip the license?”

For some systems, yes, and I would tell you if ours were one of them. Ours is not. We rely on Axon Server connectivity and distributed command and query routing, and in version 5 those components live in Axoniq Framework. The good news is that the pricing model respects how we want to buy: the modules run without credentials during evaluation, many platform features are free on the Individual plan, and smaller organizations may qualify for complimentary access. We can validate the entire migration and see the platform working on our own system before spending anything.

“Are we locking ourselves into a vendor?”

Your business history is always portable to another tool. Moving to a different framework isn't necessarily “free”, it takes real work and time, like any migration. But because your complete history lives in the event store, you have an objective source of truth throughout the process. You simply stream the events into whatever format the new tool requires, whenever you need to.

“Do we have the skills for this, or will we depend on one or two experts?”

This move reduces our key-person risk rather than adding to it. In event sourcing, getting the boundaries wrong used to be a real problem. Axon Framework 5 makes this far more flexible, so mistakes are much easier to correct. The Axoniq Platform adds a Development Agent for building version 5 applications in a structured way, which shortens onboarding for new team members. Any competent Java developer can be productive in this codebase, a claim we cannot make about our current one.

“What do you need from me, specifically?”

Two things, neither of which costs money yet. First, approval for a short timeboxed spike: run the conversion recipes against one representative service and stand up the free Individual plan, so we return with a grounded estimate and working evidence instead of a slide. Second, a conversation with the Axoniq team about licensing, including whether we qualify for complimentary access. After the spike, I will bring you a scoped proposal with real numbers, and you can make the investment decision with evidence in hand.

STRUCTURE OF A GOOD ANSWER

Acknowledge the concern. Answer it with a fact. End on the decision you are asking for.

Recommended Next Steps

-
- 01 Start with the migration guide's prerequisites section to confirm Java, Spring Boot, and Axon Server version alignment for your environment.
 - 02 Run the OpenRewrite recipes against a branch of one representative service to get a grounded effort estimate: the remaining compile errors and source markers give you a concrete, countable scope.
 - 03 In parallel, sign up for the Axoniq Platform Individual plan (free, no credit card) and run the Axoniq Framework modules in evaluation mode against that service. That gives you working evidence for the proposal: a migrated service, real platform metrics, and a validated dependency swap.
 - 04 If you are a smaller organization, ask about complimentary access before assuming the commercial tier is out of reach.
-

The full migration guide is available at docs.axoniq.io under the Axon Framework Reference, and the platform at platform.axoniq.io. Teams not ready to commit to the full migration this quarter can ask about long-term support (LTS), which keeps AF4 patched while the migration proceeds on a slower timeline, at a cost that grows over the long term.

TALK TO US

The teams that move now get more than safety.

They get the version of event sourcing everyone signed up for in the first place. To talk through your migration path or licensing questions, reach out to the Axoniq team.

Talk to the Axoniq Team →