

Datasheet

Axon Framework

Build your next-generation applications
with AxonIQ technology

Customers are facing more and more complexity every day due to digital transformation, new customers, challenges, products, functionality, legislation, employees, competition, and distribution channels. In many cases this means looking at replacing existing monolithic brown field applications or in green field situations.

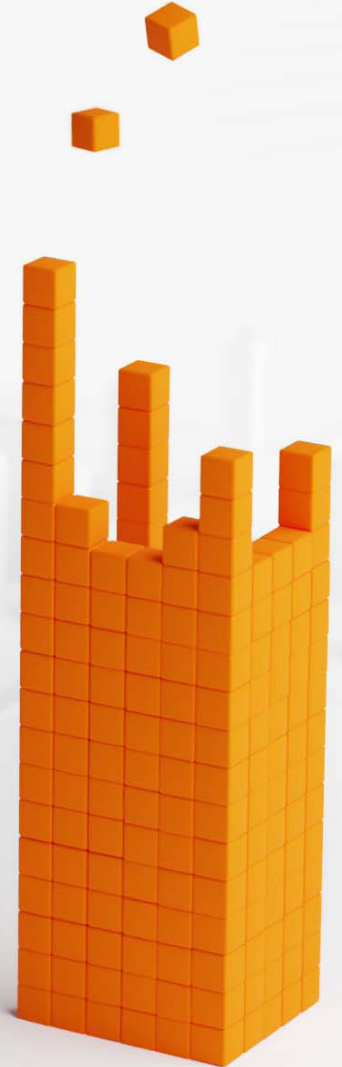
Complex application requirements:

- Application agility
- Timeliness
- Data for analytics and AI
- Scalability
- Auditability

Why Event Driven Architecture (EDA)?

EDA is a modern software architecture and model for application design. And, the core of everything we do at AxonIQ. With an event-driven system, the capture, communication, processing, and persistence of events are the core structure of the solution. This differs from a traditional model.

- Event Driven Microservices (EDM) are services that for example make an action to reduce an 'available stock' number by one when one item is purchased and taken away. The reduction by one here is captured in an event. Making that reduction is what a microservice does by reacting to that event.
- Domain Driven Design (DDD) is used to allow all functions to use the same language that's spoken and used to collaborate and communicate to each other in a way that makes sense to both (business and development). The structure of the software reflects the real world within the company..
- Command Query Responsibility Separation (CQRS) - to enable efficient allocation of resources it's ideal to have a read only version of the changes you've stored. So, requests for information and change requests in progress don't slow each other down.
- Event Sourcing (ES) - is the single source of truth and transparent so the origins of any status can be established, which is particularly useful for auditing in financial scenarios, for example.



Why Axon Framework?

Axon Framework is an open source framework for building modern applications using an event-driven architecture (EDA).

An effective event driven architecture requires domain driven design principles, the transparency of event sourcing, and command query responsibility separation to offer tangible business benefits. However, an effective event-driven architecture is daunting for an in-house developer team to implement from scratch.

Axon Framework was the basis for, and benefits from, the AxonIQ development team's experience to establish which approaches are most effective.

Benefits of the Axon Framework

- No need to build a Framework yourself
- Modern / Well Maintained
- Mission critical
- Open Source
- Distributed
- Agility
- Timeliness
- Transparency

Why AxonIQ

We offer Axon, the best purpose-built event-thinking platform. Axon is an end-to-end development and infrastructure platform that lets you smoothly evolve event-driven systems into microservices.



Developing EDA without the right off the shelf and open source tools, you will spend around 50% of your time on technical challenges. You will need 5-8 man-years to develop the framework & data storage just to get started. Do not reinvent the wheel, focus on the important stuff

- AxonIQ



Build



Axon
Framework



Event Driven
Microservices

DDD

Domain Driven Design



CQRS



Event Sourcing



Run



Axon
Server



Routing



Event Store



Observability



High
Availability

Feature overview

License

- Free & open source (Apache 2.0 License) Commercial developer support options

Messaging / Core

- Full Fledged support for:
 - ▷ Command Messages and Handlers
 - ▷ Event Messages and Handlers
 - ▷ Query Messages and Handlers
 - ▷ Dead-Letter Queue
- Annotation driven configuration of Command, Event and Query Handlers
- Distinct Message Buses for Commands, Events and Queries
- Message Bus implementations for Location Transparency
- Distributed Bus implementations for cost-effective application scaling:
 - ▷ Axon Server (Commands, Events, Queries)
 - ▷ Spring Cloud Discovery (Commands)
 - ▷ JGroups (Commands)
 - ▷ AMQP (Events)
 - ▷ Kafka (Events)
- Event Stream Replays
- Support for redundancy, sharding and high-availability setups
- Quick set up of infrastructure components using Configuration API

Modeling

- Aggregate Modeling
- Complex Business Transactions with Sagas
- Integrated Message Handling on Aggregates and Sagas
- Deadline Scheduling and Handling for Aggregates and Sagas
- Given-When-Then Test Fixtures for Aggregates and Sagas

Event Sourcing

- Dedicated Aggregate Event Sourcing Repository
- Event Store implementations:
 - ▷ Axon Server
 - ▷ JPA
 - ▷ JDBC
 - ▷ MongoDB
- Aggregate Snapshotting
- Upcasting for Event revisions

Miscellaneous

- Application monitoring through:
 - ▷ [Dropwizard Metrics](#)
 - ▷ [Micrometer](#)
- Distributed Tracing through OpenTelemetry
- Serializer configuration
- Thorough integration with Spring for simplified configuration
- Axon Framework Spring Boot Starter
- Added support for Kotlin users
- Added support for Project Reactor users
- Added support for CDI users
- Added support for Multi-Tenancy

Use case examples

New Application Development, Modernizing Software Platforms (Monolith to Microservices), Payments, Administration, Central Register, Marketplaces, E-commerce, Logistics, Crypto Management, Customer Onboarding

Contact us