



Antoine Ducrocq

EBX DEVELOPER · .NET & MICROSERVICES ARCHITECT ·
CI/CD AUTOMATION

cv@hexainnovate.com

hexainnovate.com

linkedin.com/in/ducrocq-antoine

EBX / TIBCO developer and .NET architect, sixteen years of experience on Master Data Management, microservices architecture and DevOps projects. Core skill set: EBX, Java and .NET/C#, in cloud and containerized environments (OpenShift, Azure). Moved from developer to application architect over fourteen years at Matmut; freelance since 2024. Since late 2025, I've been making AI-agent-assisted development repeatable, on a product of my own.

SKILLS

MDM & EBX

TIBCO EBX 6.1 / 6.2

Dun & Bradstreet

Identity resolution

Java triggers

REST API EBX

CLOUD & DEVOPS

Azure DevOps

Artifactory

Harness

Jenkins

Azure Functions

Configuration as code

LANGUAGES

Java

C# / .NET 8

SQL

ARCHITECTURE

Microservices

DDD

Clean Architecture

Event-Driven

CONTAINERS & DATA

OpenShift

Kubernetes

Docker

Elasticsearch

SQL Server

METHODS & TOOLING

Scrum

Scrum Master

Unit & integration testing

Git

PROFESSIONAL EXPERIENCE

HEXAINNOVATE – FREELANCE (MAY 2024 → PRESENT)

EBX Developer & DevOps

Nov. 2024 – Aug. 2026

through Data Verity Corporation

Assignment: AXA XL (US subsidiary) — fully English-speaking working environment

Developer within an international team on the **party master data hub** of an international insurer, on **TIBCO EBX 6.1 then 6.2**, in an architecture exposing the hub to more than a dozen producing and consuming systems, mostly through APIs.

01 MDM / TIBCO EBX

The hub's identity resolution component, end to end.

- Designed and built, jointly with the EBX lead, the entire **Dun & Bradstreet** component: D&B API calls, party data consolidation, matching and reconciliation logic (**DUNS**, **XL Name ID** and **Name-Address** workflows).
- EBX 6.1 → 6.2 migration**: handled the impact on the organization's EBX base image, adaptation work and the test cycle that came with it.
- Developed **Java triggers** and EBX application components; took over that work after the developer in charge of the data model left.
- Resolved a write-side bottleneck caused by high-volume batches of API calls: moved the **Elasticsearch** query path, applied optimizations and revised the rules.

02 Legacy .NET system migration

Rebuilding the API portfolio without the calling systems noticing.

- Overhaul of the **.NET Framework API** portfolio sitting on top of the legacy MDM, under **strict functional parity**: interface contracts and messages left unchanged, with no adaptation required from producing or consuming systems.

- Write APIs hitting the legacy MDM directly: full rewrite in **Java and EBX APIs**, with transposition to the hub's new data model.
- Search and read APIs: migration from .NET Framework to **.NET 8 on Azure Functions**, together with a switch of the data source from the MDM to **Elasticsearch**.
- APIs already backed by Elasticsearch: technical migration from .NET Framework to .NET 8 on Azure Functions.
- Regression testing run jointly with the project test team and the consuming teams, ensuring the migration remained transparent to calling systems.

03 DevOps & platform — Harness, OpenShift, Artifactory

Taking delivery off developer workstations and putting it back into the chain.

- Rebuilt the MDM program's delivery pipeline end to end: around ten **Harness** pipelines covering 4 EBX projects, C# and Java Azure Functions and several cross-cutting components.
- Took over a bare-bones pipeline — deploying the pod only, parameters re-entered by hand, build and Artifactory publication from developer workstations — and turned it into a build → test → artifact → deployment → promotion chain across 4 environments, with no human intervention.
- Declarative modeling of the **OpenShift** topology (ImageStream, secrets, ConfigMap, PVC, Deployment, Service, Route) and centralization of the configuration in a single CI/CD repository covering all EBX projects.
- Designed a **standardized EBX base image** for the organization, bundling its shared modules, published to Artifactory with automatic versioning.
- Removed the **major operational risk** posed by developer-workstation builds and manual entry of version numbers all the way to production: artifacts built in CI, reproducible and traceable to a commit.

Harness · OpenShift · Artifactory · Docker · EBX 6.1 / 6.2 · Java · .NET 8 · Azure Functions · Elasticsearch · Dun & Bradstreet · Azure SQL Server

AI-agent-assisted development

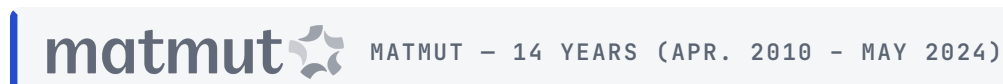
Since late 2025

In-house R&D and product

Getting AI agents to produce software reproducibly, and proving it on a real product rather than a demo.

- First **multi-agent system run locally** — WSL and a personal GPU, self-hosted models.
- **Seven specialized AI agents** — spec, architecture, development, testing, documentation, review, CI. The whole cycle is covered, from framing a feature through to the published release: the assistance doesn't stop at writing code.
- Every change goes through a **four-angle review**: code, architecture, conformance to spec, security.
- **Hexarace**, a C# product under the Hexainnovate brand, in development: SimHub plugins for sim racing. Two modules in progress, public site and changelog, hosting and release pipeline run in-house.

CLaude Code · C# · .NET · AI agents · CI/CD · WSL



Application Architect & DevOps

Dec. 2021 — May 2024

Moved into an architect role on the program set to replace the COBOL mainframe: designing the target platform and raising the bar on engineering practice.

- Designed and built the **microservices platform on OpenShift** meant to replace the group's COBOL mainframe.
- One of the **go-to engineers** for the first development teams building on that platform: guidance, implementation decisions, code reviews.
- Built the **CI/CD pipeline from the ground up** (Azure DevOps, SonarQube, Artifactory, XLDploy).
- Led the **legacy migration** from .NET Framework to .NET 8, from the first assessment through to production.
- Introduced asynchronous, event-driven messaging with **Kafka**.
- Set the **development standards** used by 150 .NET developers.

- Spread engineering practice across the teams: DDD, Clean Architecture, code review.

OpenShift · Azure DevOps · SonarQube · Artifactory · XLDeploy · .NET 8 · Web API · Kafka · DDD · Clean Architecture · SQL Server

Senior Tech Lead & Scrum Master

Jan. 2017 – Dec. 2021

- Technical lead for a **scrum team of about ten developers**: technical decisions, architecture, the quality of what shipped, and the bridge to the business side.
- Ran the **scrum master community of practice** (25 people): cross-team coordination, agile rituals, continuous improvement.
- Contributed to .NET good practice across the whole IT division (150 developers) and improved the shared tooling.

.NET Framework 4.8 · .NET Core · MVC · Web API · SOAP · SQL Server · Scrum

.NET / C# Developer

Apr. 2010 – Mar. 2017

Built and maintained the **.NET/C# applications sitting on top of the COBOL mainframe**: MVC web applications, SOAP web services, performance tuning, unit and integration testing. Day-to-day work on SQL Server and Entity Framework.



CTO & co-founder

Mar. 2013 – Aug. 2015

A crowdfunding platform launched **before the sector was regulated**, designed and run end to end: architecture, development, servers.

- Designed and built the platform **from the ground up**: .NET MVC front end, **REST API**, **SQL Server** database. Built single-handed, later with a junior developer.
- **Ran the infrastructure**: two dedicated servers, one for production, one for the build system (TFS) — backups, security, availability, releases.
- The platform **ran in production for about a year**, carrying **three live funding campaigns**.
- Built to the **regulatory constraints of crowdfunding**, one of the partners holding the status granted by the **AMF**, the French market regulator.

.NET Framework · MVC · REST API · SQL Server · TFS · Dedicated servers

EDUCATION MSc in Software Engineering · Université de Rouen, France — 2003-2009

LANGUAGES French — Native · English — Professional & technical