

Grégoire Marty

Full Stack Dev • Data Engineering & Analytics • AI / LLM / MCP • Production Systems

France → Worldwide (remote) • gregoiremarty@pm.me • gregoiremarty.com

linkedin.com/in/martygregoire • github.com/gregoirembv

PROFILE

Data systems running in production - ingestion pipelines, analytics engines, LLM and MCP integrations - end to end ownership, from the storage layer to the deploy. Metrics is where the constraints came from: a basketball analytics platform covering four leagues off a single VPS. One machine with a ceiling, thirty seconds from push to production, no staging, no second pair of eyes. Four of the libraries that came out of it are published. *Solo means the tooling is the reviewer*. I take on backend, data and AI-integration work for teams who need the thing still standing in six months.

OPEN SOURCE

Four libraries extracted from production incidents • github.com/gregoirembv • MIT • zero-dependency

→ **Packaged and tested:** hatchling, CI matrix on Python 3.11 / 3.12 / 3.13, pytest and ruff on every push. Each one answers a failure I had to fix properly rather than work around.

twinrun • [differential testing](#)

→ Runs two implementations across a grid of inputs and proves not a single value moved. Written to replace the storage layer under hundreds of computed figures: 174/174 and 109/109 identical, zero divergence.

→ Caught a tie-break that silently permuted a ranked list, and a date column that changed type underneath a comparison - both invisible to unit and integration tests.

bytecap • [memory-bounded cache](#)

→ A cache bounded by measured bytes that refuses a load it cannot afford, instead of evicting everything and dying anyway. Written after an unbounded module-level dict killed 71 workers in a loop.

→ The measurement behind it: one parsed object costs 3.54 MB resident against 302 KB pickled - serialized size understates real memory by roughly 12×.

staleguard • [freshness contract for derived data](#)

→ Makes a derived artifact refuse to pretend it is fresh. Written after a refactor moved a consumer without moving its producer: 112 games invisible for four weeks, zero errors in any log, a fully green test suite.

hardwood • [coordinate normalization](#)

→ Turns shot coordinates from any provider and any frame of reference into one canonical half-court, with league-aware geometry - NBA, WNBA, NCAA men and women, FIBA arcs, corners, lanes and restricted areas.

PRODUCTION SYSTEM

Metrics • [hey-metrics.com](#) — [live](#) • 2024 — present

- **Basketball analytics platform** covering WNBA, NBA, NCAAW and NCAAM. Built solo, end to end. Live continuously since 2024, closed beta, 50+ users.
- **MCP layer in production:** servers over Lemon Squeezy, Brevo and GitLab so billing, mailing and deploys are driven in natural language, with every side-effectful action written to an audit table.
- **Columnar migration:** resident memory taken from 5-6 GB down to ~0.25 GB, stored data reduced by a factor of fifty - the answer to a single-machine ceiling had to be a better structure, not another node.
- **Play-by-play engine:** possession counter validated at 99.5% accuracy; every game action indexed and filterable into a video playlist.
- **Analytics engine:** 30+ advanced metrics implemented from the source papers - Four Factors, on/off ratings, TS% and eFG%, usage, win shares, PER, 6-zone shot analytics, league-adjusted values, percentile ranks.
- **Backend & storage:** Python 3.12, FastAPI across 8 route modules, 4 uvicorn workers under systemd; atomic writes, schema versioning, daily warm-up, sub-second cached responses.
- **Infrastructure & observability:** OVH VPS, nginx, GitLab CI/CD - 30 seconds from push to live without staging; 6 alert rules evaluated every 5 minutes off a metrics_history table fed by psutil and journalctl. No third-party monitoring.

OTHER WORK

MMC • [Mon Meilleur CV](#) • 2023 — present

- **AI resume builder** - document parsing, LLM-based enrichment, schema-validated generation tailored to a job description. Python, FastAPI, structured prompt pipelines.

Le Roster • [Sports Media](#) • 2023 — 2025

- **Full-stack build and hosting** of a basketball media outlet - frontend, backend, deployment and ongoing maintenance.

W • [WNBA Podcast](#) • 2024 — present

- **Data analysis and dataviz** for a basketball audience - advanced statistics, custom charts, tactical breakdowns built from the Metrics engine.

SERVICES

Custom MCP server: 1—3 weeks

Data pipeline & analytics engine: 4—8 weeks

Architecture / performance audit: 3—5 days

LLM integration sprint: 2—4 weeks

Web application — frontend & backend: 3—5 weeks

Ongoing retainer: ongoing

EXPERIENCE

Freelance AI & Data Engineer • Independent • 2023 — present

- LLM integration, custom MCP servers, data pipelines and full-stack builds for sports tech, SaaS and media clients.
- Open for engagements since January 2026. Metrics and the four published libraries are the reference: production systems, not portfolio pieces.
- Technical documentation: API references, data model specifications, deployment guides.

Content & Web Coordinator • Le Roster • 2023 — 2025

- Technical ownership of a sports media platform: frontend, backend, hosting, release process.

Founder / Manager • Androuet / Racines / Prunelle • 2018 — 2023

- Cheese shop and grocery store, owner and manager. Found the time to build CRM and HR tools while learning to be efficient at selling goods.

Independent Consultant • Music Industry • 2011 — 2018

- Project management and delivery for labels and artists (France, UK, US). Produced two albums for Ubiquity Records (California). Swiss army knife: building websites, presskits, newsletters, recording and editing music and videos, management of teams, people, careers and projects...

EDUCATION

Anthropology • Master

- Studied cultural movements from the 90's — hip hop, punk, anything edgy and out of the box.

STACK

Languages: Python 3.12 • JavaScript (ES6+) • SQL • Bash

AI: LLM integration • RAG • agents • MCP servers • structured extraction • evaluation & golden datasets

Infra & ops: Linux • nginx • systemd • Docker • VPS • Let's Encrypt • GitLab CI/CD • GitHub Actions • observability & alerting

Spoken: French (native) • English (fluent) - remote, France & international

Backend: FastAPI • uvicorn • REST • OpenAPI • Pydantic • async • SQLite • columnar storage

Frontend: Vanilla JS SPA • custom routing • lazy loading • dataviz & charting • HTML/CSS

Practices: Differential testing • schema versioning • atomic writes • audit logging • pytest • ruff • packaging (hatchling)

Oh, and I love dogs!