

Antoine Boucher

SOFTWARE ENGINEER — PLATFORM AND COMPUTER GRAPHICS

+14384918831 | ✉ antoine@antoineboucher.info | 🏠 antoineboucher.info | 📱 antoinebou12 | 📺 antoineboucher12

Hiring team

April 20, 2026

ORGANIZATION OR CHANNEL WHERE YOU RECEIVED THIS DOCUMENT

Professional narrative and guide to antoineboucher.info

Dear reader,

Why this document exists

This letter is intentionally longer than a one-page pitch. It is meant to stand on its own when you do not yet have my CV in hand, and to explain how I work, what I have shipped, and how the public site antoineboucher.info is organized so you can verify claims quickly. You can treat it as a narrative companion to the PDF resume linked from the English and French landing pages.

I am a software engineer with a master's trajectory in information technology engineering (ÉTS), combining platform engineering (Kubernetes, Terraform, GitLab CI/CD, GCP), backend services in Java, Kotlin, and Python, and a parallel thread in interactive graphics (OpenGL, C++, Blender) tied to research. I also invest time in teaching assistantships, open-source maintenance, and a homelab used to rehearse networking, automation, and self-hosted services the same way others rehearse scales on an instrument.

If you are comparing candidates on paper alone, remember that paper hides cadence. This letter exposes cadence: how I describe trade-offs, how I sequence information, and how much context I assume you already carry about Montreal's university ecosystem, regulated insurers, and research institutes. When we eventually speak, you can probe any paragraph here and ask for receipts—logs, merge requests, syllabi, or student anonymized rubrics where policy allows.

Microcopy and labels you will actually see

The static pages reuse Bootstrap 3-era styling on purpose: it keeps the CV readable on older corporate laptops and avoids chasing front-end fashion when the goal is legibility. Section titles are chosen to match recruiter mental models: **Work Experience** not **Chronology**, **Core Skills** not **Buzzwords**. Icons precede titles so that scanning with peripheral vision still clusters sections correctly. The French equivalents use the same iconography with translated nouns so bilingual reviewers can toggle languages without re-learning layout. Contact mirrors the PDF header: phone, email, LinkedIn slug, GitHub handle, and the canonical domain, each as a clickable anchor. Nothing on the page auto-plays video, spawns pop-up, or asks for cookies; it is a document pretending to be a web page, which is the whole aesthetic.

Career arc in plain language

Research and secure delivery (IMC2). At IMC2—the Institut multidisciplinaire en cybersécurité et cyberrésilience—I operated as a permanent part-time research assistant through early 2026. The role blended platform ownership with product-minded engineering: GitLab CI/CD and runners for research services on GCP, containerized workloads, Terraform-managed rollouts, and GKE networking concerns that matter when microservices actually talk to each other in a cluster. I hardened Java microservices with JUnit and Playwright, connected Android builds into the same promotion path, and contributed to analyst-facing tooling that automated cybersecurity risk analysis across more than a thousand risks and well over a hundred threat categories, with weighted scoring and

an Angular triage interface. That mix—DevSecOps rigor, quantitative prioritization, and human-facing UX—is the through-line I want you to remember.

Teaching DevSecOps (Polytechnique Montréal). As a teaching assistant for LOG8100 DevSecOps, I maintained the GitLab workspace for roughly twenty concurrent teams each term: repositories, CI/CD templates, runners, and workflows. The job was half pedagogy and half reliability engineering for a classroom at scale. I coached Docker and Kubernetes builds under explicit security goals, used OWASP-aligned examples to connect lecture theory to exploitable scenarios, and refreshed lab artifacts so students could resubmit with clearer expectations.

Teaching data, mobile, and integrator tracks (ÉTS). Across ÉTS courses and integrator projects, I logged a large number of contact hours spanning mobile and UX, distributed databases, and integrator delivery. I authored SQL, MongoDB, Kafka, and PostgreSQL exercises with rubrics, unblocked teams on architecture and backend design, and sustained Azure-backed course infrastructure while coaching Scrum. That experience sharpened my ability to translate fuzzy student requirements into testable outcomes—a skill that transfers directly to mentoring juniors and partnering with product owners.

Industry before research intensity. Earlier roles include cloud development at IONODES (Auth0 subscription tiers for organization-scoped IoT access, Sentry and Serilog instrumentation, Azure DevOps cadence), Kotlin and Spring microservices at Intact (campaign APIs, ELK observability), full-stack and web contracts (Django, Vue, CommerceBuild, SEO-oriented commerce), and internships spanning Python toolchains, Selenium automation, and computer vision lab infrastructure. Each stop added a layer: regulated environments, on-call diagnosability, and the politics of shipping when stakeholders disagree on scope.

Graphics and research identity. My graduate work couples tactile sensing metaphors with interactive graphics: friction and texture coordinates that evolve as surfaces wear. That is not a bullet on a sprint board; it is a reminder that I am comfortable with simulation, GPU-adjacent thinking, and papers as deliverables, not only tickets.

Signals I optimize for when joining a team

Definition of done. I care whether a change is observable, measurable, and reversible. Observable means metrics or logs move in a way humans can interpret without SSH folklore. Measurable means acceptance criteria existed before the merge request, even when the product manager was busy. Reversible means feature flags, database migrations with expand-contract patterns, or Terraform plans that can roll back without heroics. Those three words sound generic until you watch a team ship without them; then they become the difference between a weekend outage and a Tuesday deploy.

Collaboration contracts. I default to writing decisions in issues or merge requests so that async teammates inherit context. I treat code review as a teaching channel, not a gate: I comment on naming, invariants, and failure domains, and I accept the same scrutiny in return. In classrooms I learned to calibrate feedback so a student leaves with one next action, not ten anxieties; the same habit applies to juniors on a production service.

Risk posture. Cybersecurity research taught me to speak in threat models, not vibes. Teaching DevSecOps taught me to translate CWE entries into lab exercises students can exploit safely. Industry taught me that auditors and customers read postmortems. I am not looking for a culture that pretends incidents never happen; I am looking for one that narrates them honestly and funds prevention proportionally.

Open source, homelab, and assistant-facing tooling

Public repositories are part of my professional identity, not a hobby footnote. Projects such as `uml-mcp` sit at the intersection of developer experience and AI tooling: a Model Context Protocol (MCP) surface that lets LLM-based coding assistants call diagram-oriented capabilities so architecture views can be produced or refined from repository context or natural language. Related work includes diagram plugins and PlantUML-facing utilities. Separately, I operate a homelab (virtualization, WireGuard and Tailscale

paths, Home Assistant automations, media stacks) because production literacy is not only cloud APIs—it is also understanding failure modes when DNS, certificates, and routing disagree.

Why MCP matters in one paragraph. Assistants without tools hallucinate file layouts; assistants with tools still hallucinate if the tools are vague. MCP standardizes how a host advertises capabilities, how arguments are validated, and how errors return to the model loop. A diagram server under MCP is therefore not a gimmick: it is a contract that says “here are the drawing primitives you may call, here are the failures you must surface to the user, and here is how outputs map back to documentation workflows.” That is the engineering ethos behind `uml-mcp`, not novelty chasing.

Homelab as rehearsal space. Proxmox clusters, Jellyfin stacks, and Renpho metrics piped into Home Assistant are not resume padding. They are where I test backup restores, TLS renewal, and network partitions without risking a grant-funded dataset. When a student asks whether WireGuard is “easy,” I can answer from packet captures and from the emotional memory of fixing a typo in `AllowedIPs` at midnight.

The portfolio site as an interface

The static bilingual site at `antoineboucher.info` is deliberately simple on the surface: a single-column narrative with a particle background, a photo header, and progressive disclosure so readers are not overwhelmed on first paint. Think of it as an interactive CV with affordances borrowed from HTML rather than from a PDF reader.

Top navigation and language toggle. A compact nav bar sits above the hero. The **Blog** pill sends you to the Hugo-powered journal under `/blog/`, where longer write-ups (infrastructure notes, workshops, data explorations) live with their own theme, navigation, and build pipeline described elsewhere in this repository’s documentation. The language control is a paired link: **EN** opens `index-en.html`; **FR** opens `index-fr.html`. Both files mirror structure; they differ in copy and in which PDF the download button targets.

Hero and the Download CV control. The hero block centers an avatar, your name and title, and a single primary call to action. On the English page the control reads **Download CV** and points at `cv-en/resume.pdf` with the HTML `download` attribute. On the French page it reads **Télécharger le CV** and points at `cv-fr/resume.pdf`, also with `download`. Practically, that attribute encourages browsers to treat the navigation as a file transfer with a suggested filename instead of only rendering the PDF inside the tab—a small UX detail that matters on mobile devices and when recruiters file artifacts quickly. There is no separate commerce-style “Add to cart” button on these pages; the main interactive vocabulary is navigation, disclosure toggles, and outbound links.

Collapsible sections (About through Interests). The body uses nested `<details>` elements. At the section level, each panel has a summary row with an icon and a title. Opening a panel reveals list items or nested job blocks; closing it hides noise. On the English page, the outer sections appear in this order: **About** (three high-level bullets: cloud-native focus, open-source footprint, master’s program), **Work Experience** (each employer in its own nested disclosure with highlights and a technical-skills line), **Leadership & Community** (student societies and conference volunteering), **Selected Projects** (curated GitHub and PyPI links with one-line descriptors), **Contact** (email, LinkedIn, site, GitHub), **Education** (ÉTS master’s and bachelor’s with thesis and club notes), **Core Skills** (grouped categories aligned with the LaTeX CV), **Conferences** (for example Graphquon entries), **Certifications & Programs** (Credly and Sertifier-linked credentials), **Publications** (PDFs and rendered markdown where applicable), and **Interests** (short humanizing bullets). The French page preserves the same skeleton with translated labels (for example **À propos**, **Expérience professionnelle**, **Formation**, **Compétences clés**).

Nested job disclosures. Inside Work Experience, each role is another `<details>` block. The summary row shows organization, date span, and title; expanding reveals narrative paragraphs plus a “Highlights” list mirroring the strongest bullets from the CV. This pattern lets a recruiter skim titles first, then drill into IMC2, Polytechnique, ÉTS, IONODES, Intact, and earlier internships without scrolling through an endless wall of text.

Link clusters worth knowing. The Selected Projects and open-source lists intentionally duplicate high-signal GitHub URLs:

`uml-mcp`, diagram tooling, homelab automation, Home Assistant integrations, NuGet and PyPI packages where download counts are part of the story, and teaching-adjacent tooling. Publications link to local `papers/` assets where mirrored PDFs exist. Certifications link out to Credly and Sertifier so third parties can verify badges without trusting this domain alone.

What is not on the landing page. The Hugo blog has its own information architecture (posts, tags, PaperMod-derived navigation). A separate `linktree/` directory in the repository backs an optional link hub. Neither is exhaustively described here, but the Blog pill is the bridge from the static CV surface to those deeper artifacts.

Blog and build pipeline (reader's map). When you click **Blog**, you leave the Bootstrap-era static CV and enter a Hugo site whose CSS is built with npm in the theme directory, then published under a `baseURL` that matches the custom domain so Subresource Integrity and absolute permalinks do not fight each other. Posts mix English and French content depending on topic—workshop recaps, cloud experiments, economics-flavored data studies—and each page inherits the PaperMod navigation patterns (menu button on small screens, dark-mode toggle, tag lists). You do not need to understand Hugo to read the posts, but knowing the pipeline explains why some assets live under `/blog/` while the CV stays at the root.

Accessibility and progressive disclosure. Outer `<details>` panels default to **open** on first load so search engines and skimming humans see the skeleton immediately. Nested job panels start collapsed so the page height stays tractable on laptops. Icons come from Font Awesome via CDN on the static pages; that is a trade-off (external dependency) chosen for visual consistency with minimal custom CSS. Keyboard users can traverse summaries; there is no hidden JavaScript-only navigation on the landing page itself beyond the particle background script.

Downloads beyond the hero button. The hero Download CV is the only control explicitly labeled as a download, but other anchors point to PDFs under `papers/` and to rendered markdown where a long report reads better in HTML first. Those behave like normal links: the browser may inline PDFs depending on settings. If you need an offline bundle, use the hero PDF for the CV and follow publication links individually—I have not shipped a monolithic “download everything” zip because it would go stale weekly.

Leadership, conferences, and credentials as rendered online

The **Leadership & Community** disclosure mirrors student-society work you would otherwise discover only through side conversation: organizing teams, running infrastructure for clubs, and supporting events where software is the enabler rather than the headline. The **Conferences** block highlights recurring participation such as Graphquon, where I have both built tooling (for example ChatGPT-oriented plugins supporting attendees) and presented research-adjacent material; treat those rows as evidence that I can speak to graphics and security communities without treating conferences as vacation. **Certifications & Programs** links out to Credly for AWS Certified Cloud Practitioner and to Sertifier for Québec's scientific entrepreneurship program, which matters if your procurement rules care about formal cloud literacy or startup-method exposure beyond a résumé adjective.

Publications and long-form artifacts. The **Publications** panel mixes PDFs hosted under `papers/` with rendered markdown for reports that benefit from syntax highlighting. Each entry is intentionally stable URL-wise so that citations in older blog posts do not rot immediately. If you are evaluating research engineering, read at least one artifact end-to-end and notice whether figures, equations, and references survive the PDF pipeline; that is the same discipline I bring when generating architecture diagrams from code.

Interests as a sanity check. The final disclosure is short on purpose. It signals that I maintain hobbies outside screens because burnout is a delivery risk. It is not filler; it is a reminder that sustainable teams schedule recovery.

How to use this material in a hiring loop

If you are screening quickly, start with the Download CV control on the language you prefer, then cross-check one employer block on the site against the same section in the PDF. If you are evaluating systems thinking, read the IMC2 highlights and ask me how I would simplify the risk-scoring model for a smaller tenant. If you are evaluating teaching and communication, scan the Polytechnique

and ÉTS disclosures and ask for a fifteen-minute explanation of how I triaged a failing student pipeline. If you are evaluating open source maturity, pick [uml-mcp](#) or another repository and inspect issues, releases, and documentation tone.

Closing

I build platforms other engineers can run with confidence, invest in teaching and documentation so teams avoid repeated mistakes, and keep enough work in public that my standards are easy to verify. Thank you for your time. The two-page resume on [antoineboucher.info](#) is the quickest summary; this document adds structured context when that helps your evaluation.

How this fits your process. This companion is longer than a typical one-page cover letter so you can judge fit beyond keywords: it connects experience, open-source practice, and how the public site is organized in one place. For initial screening, the resume is usually enough. When you move to interviews or a hiring panel wants depth, this PDF is the supporting brief—use it to target questions on the sections that matter for your role, without duplicating every bullet from the CV.

Respectfully,

Antoine Boucher

Attachment: Curriculum vitae (PDF) via [antoineboucher.info](#) — [Download CV](#) / [Télécharger le CV](#)