

Christopher Gorski

Senior Software Engineer | AWS & Azure Specialists

<https://www.linkedin.com/in/christopher-gorski-7a762038a> | chris.dev1472@outlook.com

Work Experience

Senior Software Engineer – Allegro

April.2024 – Present

- Built and operated backend services primarily in C# (selected Go where fit) for real-time pricing, dynamic matching, and auth; used `async/await`, `Channels`, `Pipelines/Span`, and `SocketsHttpHandler/HttpClientFactory`; tuned `Server GC`, `ThreadPool`, and connection pools to cut p99 latency and boost peak throughput under Friday spikes.
- Re-platformed checkout to ASP.NET Core 8 Minimal APIs + MediatR with Kafka Outbox/Saga; hot-path p99 420 → 210 ms @ ~6.5k rps; verified via k6 and Datadog APM.
- Implemented EF Core compiled models for hot aggregates and a Dapper path for fan-out reads; DB reads -38%, ~\$14k/qtr IOPS savings; added read-replica routing for surge windows.
- Designed idempotent command handlers (outbox/inbox) with exactly-once consumption; eliminated duplicate order creation; added saga compensations for payment/stock rollbacks.
- Hardened Kestrel (HTTP/2/3), HPACK dynamic table sizes, response buffering and gzip thresholds; container CPU -22% and k8s requests -18% with no SLO regressions.
- Added Polly policies (hedging/timeouts/retries/jitter) and bulkheads; stopped two cascading failures triggered by upstream tail latencies.
- Observability: end-to-end OTel traces (W3C context), Serilog structured logs, exemplars on Prom metrics; cross-service debug 45 → 12 min.
- Security: OAuth2/OIDC (Auth Code + PKCE), mTLS between services, envelope encryption for card tokens; zero auth incidents in two quarters.
- Frontend: migrated AngularJS → Angular 17 MFE and built React promo-rules editor; LCP p75 3.2 s → 1.9 s, CLS 0.14 → 0.04; added RUM + trace-linking to backend spans.
- Release: blue/green & canary via Argo Rollouts with analysis templates (PromQL + Web Vitals); deploy failure 3.1% → 0.6%, MTTR 27 → 9 min.

Senior Software Engineer – Artisan AI

Mar.2024 – Mar.2025

- Built a low-latency inference gateway in ASP.NET Core + gRPC fronting GPU workers; p95 ~180 ms @ 2.1k rps, 99.9% availability behind Envoy.
- Implemented RAG retrieval (SQL Server + Postgres/pgvector) with Redis Bloom prefilters and re-ranking; search F1 +13%, tail p99 520 → 260 ms.
- Orchestrated async pipelines with Kafka + Hangfire (idempotent outbox/inbox); reprocessing cost -37% while maintaining exactly-once semantics.
- Added guardrails middleware (input sanitization, prompt templates, toxicity classifier); hallucinations -28% and policy violations -62%.
- Deployed models with Triton + KServe, autoscaling via HPA; A100 throughput +41%, per-request unit cost -24%.
- Frontend: React admin + Vue prompt playground with latency charts and feature flags; PR cycle 3.4 → 2.1 days, integration bugs -41%.
- Introduced OpenTelemetry spans for prompt lifecycle; enabled fine-grained SLOs (latency, accuracy, refusal) per tenant.
- Implemented API monetization (rate limits, quota buckets in Redis + policies); prevented 3 major abuse spikes without affecting normal traffic.

DevOps Engineer – AioCare

Oct.2021 – Feb.2024

- Rolled out a Kubernetes platform (3 envs, 8 node pools) with Terraform + Helm + Argo CD, cutting deploy time 45 min → 7 min (-84%) and enabling ~20 prod releases/week; verified in Argo CD history and Jira releases.
- Built progressive delivery (blue/green + canary 5%→25%→100%) on GitHub Actions → Argo Rollouts, reducing failed deploys by ~72% and MTTR 38 min → 11 min; confirmed via PagerDuty + Grafana incident timelines.
- Introduced SLOs (p95 API < 250 ms, error budget 1%/30d) with burn-rate alerts; lifted API uptime to 99.95% over 12 months and cut MTTD 12 min → 3 min; Prometheus/Grafana dashboards as evidence.
- Optimized cloud spend -28% YoY using HPA/VPA, spot nodes, storage lifecycle policies, and right-sizing; environment spin-up shrank 2 days → 2 hours with reusable Terraform modules; validated by Cost Explorer/Billing + CI timestamps.
- Shipped end-to-end observability (OpenTelemetry traces, Prometheus metrics, Loki/ELK logs) and golden dashboards per service, lowering “unknown root cause” postmortems by ~60% and after-hours pages by ~50%; PagerDuty analytics + RCA tags.
- Hardened SDLC for medical data: SBOM + image signing (Syft/Grype, Cosign), OPA/Gatekeeper policies, Vault/KMS secrets; reduced critical CVEs in images by >90% within 2 quarters; security scan trends used for proof.
- Delivered DR runbooks and quarterly game-days achieving RPO ≤ 5 min **and** RTO ≤ 30 min via cross-region backups and infra-as-code restores; results documented in DR drill reports.

Software Engineer – Spyrosoft

Jul.2016 – Sep.2020

- Designed, developed, and maintained enterprise and consumer-facing applications for multiple international clients, such as a fintech dashboard for BNP Paribas, an e-commerce platform for a retail client, and a healthcare management tool for a telemedicine startup; delivered scalable solutions using React, Angular, and Vue on the front end, with FastAPI, Django, Java, and Golang powering high-performance back ends, ensuring seamless integration, security, and excellent user experience.

Education

Master's Degree in Computer Science

Lublin University of Technology

10/2014 - 03/2016

Bachelor's Degree in Computer Science

Lublin University of Technology

05/2010 - 09/2014

Skills

- Programming Languages: C#, T-SQL, TypeScript/JavaScript, Python, Go, SQL
- Frameworks & Libraries: ASP.NET Core 8 (Minimal APIs, MVC), gRPC, SignalR, EF Core, Dapper, MediatR, FluentValidation, Polly, Hangfire/Quartz, Node.js, React, Angular, Vue
- Cloud & Infrastructure: AWS (Lambda, S3, EC2, EKS etc), Azure, Docker, Kubernetes
- Database Technologies: PostgreSQL, MongoDB, MySQL, Firebase, Redis
- Message Queues & Real-Time Systems: RabbitMQ, Kafka, WebSocket, Firebase, SNS
- Security & Compliance: OAuth 2.0, JWT, SSL/TLS, GDPR
- DevOps & CI/CD: GitLab CI/CD, Jenkins, Travis CI, Docker, Kubernetes, AWS Lambda
- Testing Frameworks: JUnit, Mockito, Jest, Cucumber, Postman, Selenium
- Version Control & Collaboration: Git, GitHub, GitLab, Bitbucket, Jira, Confluence
- Monitoring & Logging: Prometheus, Grafana, AWS CloudWatch, Datadog