Cristian Gonzales

Senior Engineer at Microsoft

https://bit.ly/eey7W https://bit.ly/2McyUX https://bit.ly/2vA649 xcristian.gonzales@gmail.com

Experience

Microsoft Corporation
Senior Software Engineer, Office Data Flywheel September 2026 – Present
Software Engineer, Excel AI April 2025 – September 2026

- Sole, original author, and owner of the Office Data Flywheel dashboard, a self-service Power BI reporting layer built using an internal SQL-like analytics language that unifies Word, Excel, PowerPoint, and Cowork Agent product telemetry derived from massive internal datasets processed through Azure Machine Learning (AML) pipelines.
- Built Plotly-driven visualizations for user intents, document taxonomies, model comparisons, and OfficeJS code-generation diagnostics within the agents, replacing ad hoc query exploration with a prioritized product fix backlog and direct input to model-selection and vendor decisions.
- Drove internal adoption to tens of DAU/WAU, and hundreds of distinct users (accounting for 90% of the cluster's queries), streamlining and compressing analysis from days to minutes; demoed to Satya Nadella (CEO) through PM leadership and set the precedent for Data Flywheel efforts across the business unit, with CVP-level visibility.
- Rated 200/200 for FY26 performance (highest possible), placing in the top 1–5% of engineers at my level across the company.
- Maintained the PROSE research group's evaluation framework for Excel Agent Mode and built Python/C++ 1DS telemetry bindings to plumb Excel Agent evaluation telemetry into internal databases for the purpose of streamlining ingestion and building visualizations, giving researchers a consistent view of agent configuration changes and ship decisions.

Microsoft Corporation
Software Engineer, Office Copilot (Backend) December 2024 – April 2025

- Prototyped an AI-enriched service to serve dynamic, personalized prompts that live within the Copilot chat pane of the Office web document editor, to drive habitual usage of Copilot within the Office web apps, whereby the LLM prompts are contextualized with information about the user from Microsoft Graph and internal Office data sources.
- Core contributor to an experiment, in an effort to further drive habituated Copilot usage, to extend the functionality of the AI rewrite feature within the Word web editor, to better inform the AI-suggested rewrites using previously authored text samples from the user.

Microsoft Corporation
Software Engineer, Office Online (Backend) May 2023 – December 2024

- Changed Discovery, an integral part of the WOPI protocol that powers *all* of Office Online, to pare down Office Online's partner storage integration offering to only the officially supported Office apps and app capabilities for partners, along with mapping the partner test version of Discovery to a more production-ready version–this yielded the net result of not exposing internal prototyping and test datacenters, which can prove to be extremely volatile, to real customers (i.e. an outage internally impacted external customers).
- As part of shipping new, CDN-deployed customer funnels to the flagship Office apps, found at powerpoint., excel., and word.cloud.microsoft, leveraged existing Office Online infrastructure to address a feature gap in an internal offering of Azure Front Door (AFD), a layer-7 aware traffic manager, by building a service on the Office Online stack to absorb failed requests from AFD and proxy them to the CDN to serve the static content in question, improving the reliability of serving the CDN-deployed content by 2%.
- As a byproduct of standing up new web services within Office Online's infrastructure to support Office's effort to bring Office to the cloud.microsoft domain, introduced the usage of the .NET 6+ runtime and Kestrel as a web server, in lieu of .NET Framework and Internet Information Services (IIS), for which there was precedent for, which yielded a ballpark savings of 20% in compute costs.

- Migrated server-to-server authentication flows for critical internal proxy services supporting Office Online, from using app-asserted Bearer tokens to more secure Proof-of-Possession tokens, mitigating the risk of high-privilege access (Bearer) token theft and unauthorized access to protected resources, that of which was a top-down C-suite driven effort to enhance customer trust across Microsoft.

Microsoft Corporation
Software Engineer, Microsoft Loop (Backend) May 2022 – May 2023

- Led the design and implementation of a waitlist management API, using a proprietary NoSQL persistent store, for Loop, a new product in the Office 365 ecosystem, with an upper bound of 200,000 new users at public preview.
- Implemented European Union data boundary (EUDB) -compliant merging logic for Loop user "presences" in cross-boundary "workspaces", and EUDB-compliant routing of requests to datacenters nearest to the client, bringing down the P99 latency times for a single user presence request from 3 seconds to a few hundred milliseconds.
- Implemented server-to-server (S2S) authentication, using Azure Active Directory (AAD) tokens, for Loop and reduced latencies using on-box token caching, bringing down the P99 latency for some scenarios by more than 50%.

Social Finance, Inc. (SoFi)
Senior Software Engineer (Backend) March 2022 – May 2022

- Contributor to a service adapter that served as a read-only caching layer to all microservices across the company so they could meet their SLAs, in an effort to reduce P99 read times for some services in the order of hundreds of milliseconds.
- Prototyped a Debezium listener that listened to data change events from event sources onboarded to our service, for the purpose of dynamically storing them in DynamoDB, regardless of the shape (schema) of the Debezium event payload.
- Introduced code infrastructure to support LocalStack in development workflows to avoid the overhead of redeploying AWS infrastructure to our staging environment, allowing for leaner workflows where redeployments took seconds instead of minutes.

Meredith Corporation (Bizrate Insights)
Software Engineer (Backend) July 2021 – March 2022

- Introduced in-memory caches to microservices (AWS Lambda) to reduce network requests to an internal A/B testing service, reducing response times of microservices of interest by roughly 80% for "hot" Lambda containers– benchmarks in the order of 5 seconds to a few hundred milliseconds.
- Minified, and introduced CDNs (AWS CloudFront) for, static content being served by a core Java Spring microservice, cutting down the time to serve static content by >80% in some cases.
- Architected a management microservice (AWS Lambda) sitting behind a REST controller (AWS API Gateway) to ban users at the network boundary, using AWS Cloud Development Kit and AWS Web Application Firewall (WAF), whereby WAF would examine universally unique identifiers associated with the user.
- Introduced Object Relational Mapping, using SQLAlchemy, in the form of a Data Access Layer and packaged as a Python pip distribution, to replace ad hoc raw database operations being made in Apache Airflow jobs.
- Cultivated a strong engineering culture by championing Test-Driven Development where there was otherwise none, resulting in the turnaround time for features in the Software Development Lifecycle (SDLC) being cut by nearly half, as bugs where caught earlier in the SDLC, and SDETs and developers were able to work in parallel rather than serially.
- Championed DevOps by defining cloud infrastructure-as-code through CloudFormation, allowing for infrastructure to become less anecdotal and helping developer velocity when an engineer would try to understand the surrounding infrastructure of a given part of the greater architecture.

Northrop Grumman
Software Engineer (Infrastructure) August 2020 – July 2021

- Developed a library of dynamic HashiCorp Packer configurations to build DISA and NIST STIG-hardened RHEL and CentOS machine images for VMWare VSphere and Docker, using Gradle (Groovy) and an in-house Ansible library, migrating deployments from PXE-boot to template-based and reducing deployment times by roughly 60%.

- Developed Jenkins pipelines to deploy hardened infrastructure and test in-house Ansible configurations using Docker workflows, allowing the team to save at least 80 developer hours per month, and developed VSphere wrapper APIs, using Java and Gradle, for Jenkins resource management.
- Wrote Ansible playbooks to harden firewalld with respect to NIST STIGs, and tested them by spoofing IP-layer header attributes and creating Martian packets, using Python's Berkeley Sockets API interface and kernel tunables.
- Wrote Ansible roles to configure and harden deployments of Cisco NX-OS Layer-3 switches, VMWare ESXi guests, Linux NetworkManager, and Chrony timeservers.

Northrop Grumman
Software Engineer (Build and Release) July 2019 – August 2020

- Developed Jenkins pipelines to streamline different build processes in Linux, Solaris, and Windows, decreasing the time of the build and release process from four days to two hours, recognized in a company press release.
- Ported static Jenkins web interface logic into a dedicated Groovy DSL codebase to describe job configurations as code, tailored for the needs of different Jenkins pipeline workflows (developer sandboxes, release builds, nightly builds, etc.), where efforts allowed for an estimated 40% decrease of COGS for the configuration management team.
- Automated deployment for builds for ad hoc projects and ported build systems from Apache Ant to Apache Maven.

Northrop Grumman
Software Engineer (Backend) July 2019 – August 2020

- Developed an API to filter JMS messages from a legacy JMS service, using Guice injections to differentiate between different instances of the service and JAXB for object serialization to send messages across the wire.
- Refactored database schemas and their associated JPA mappings, along with protocol buffer definitions, to differentiate between different satellite missions for a server-side event message syslog implementation, where the mission identifier was injected using Guice.
- Developed a JAXB unmarshalling utility so subscribers may filter syslog event messages based on the satellite mission, where subscriptions were made through the unmarshalled data and Guice dependency injections.
- Served as a Scrum Master, leading daily Scrum meetings, retrospectives, and planning.

Northrop Grumman
Software Engineering Intern June 2017 – August 2017

- Prototyped a visualization tool to track satellite health statuses with NASA's GMSEC API and Guice injections.

Education

University of California, Santa Cruz
Computer Science, B.S. June 2016 – June 2019

Stanford School of Engineering: Artificial Intelligence Professional Program
XCS 221: Artificial Intelligence: Principles and Techniques August 2026 – December 2026

Selected through leadership nomination to be part of Microsoft Office Product Group's (OPG) Applied Science Upskilling Program for Engineers, with a capstone project at the conclusion of the course.

Skills

Languages: Bash, C, C#, Go, Java, Python, SQL, TypeScript

Tools and ecosystems: Apache Airflow, AWS, Docker, Git, Hugging Face, LaTeX, PyTorch

Concepts: Applied AI, Build and Release Engineering, Computer Networking, Containerization and Virtualization, Deep Learning, DevOps, Distributed Systems, Experimentation and A/B Testing, Machine Learning, Operating Systems

References available upon request.