Geekfest@OMF: Building a Grassroots Enterprise Engineering Community

Author: Mike Hall (Staff Software Engineer) · **Canonical Archive:** https://www.just3ws.com/

A four-part field guide documenting how Obtiva's Tuesday study tradition inspired a weekly enterprise peer-learning forum, grew into an OpenTelemetry Community of Practice across six lanes, and transitioned cleanly to funded full-time stewardship.

Geekfest@OMF, Part 1: Lineage, Charter, and Zero-Friction Design

This is **Part 1** of a four-part series documenting **Geekfest@OMF**, the internal peer-learning community I started at OneMain Financial in November 2022. This series draws on my session logs, planning notes, and public archive records while excluding proprietary company details and colleague names.

Why Start an Internal Peer Forum?

Large engineering organizations rarely lack smart people. They lack low-stakes rooms where engineers from different squads can learn together.

Daily delivery work pulls teams into narrow lanes. Engineers spend months shipping features inside one service while losing sight of adjacent stacks, shared tools, and emerging practices. Formal corporate tech talks rarely fix that gap. They often require polished slide decks, approval reviews, and top-down agendas that raise the bar too high for everyday practitioners.

Operating as a **Staff Software Engineer** (serving as Software Architecture Lead for the Acquisition Lane and later Associate Director, Staff Engineer, a senior IC role with no direct reports), I wanted a weekly room with almost zero friction. People needed a place to share what interested them, test rough ideas, and reconnect as craftspeople without waiting on management mandates.

Credit Where It Is Due: The Obtiva Lineage

I did not invent Geekfest. Dave Hoover, Kevin Taylor, and the team at **Obtiva** (later acquired by Groupon) pioneered the original Tuesday Geekfest in Chicago.

When I worked at Obtiva, Tuesday Geekfest was the highlight of my week. Developers owned the room. Over time, it grew into a trusted space where people rehearsed conference and user-group talks, shared open-source experiments, and ran hands-on micro-hack sessions. Format mattered less than showing up and participating. That same peer-enablement mindset later shaped how I approached engineering onboarding at Groupon and community organizing in my broader career history.

If you want the deeper history of that tradition, watch The Apprenticeship Move: Dave Hoover on GeekFest and Developer Mentorship or the archived UGtastic YouTube recording with Dave Hoover (9m 41s, synced to the public archive).

From Open Tech Talks to Geekfest@OMF

In mid-2022, I ran several informal sessions called *Open Tech Talks*. They covered microservices, federated GraphQL, event-driven architecture, and Git katas.

Those early sessions worked, but the name caused confusion with formal corporate tech talks. On **November 1, 2022**, I relaunched the group as **Geekfest@OMF** to honor Obtiva's Tuesday tradition and signal a different kind of room.

The charter stayed short and plain:

> Let's share what interests us, what we are looking at over the horizon, or something useful around the corner. We grow as we learn, and by sharing, we grow together.

The ground rules were simple:

- **Fixed weekly cadence:** Tuesdays from 12:15 to 1:00 PM Central. Short enough to fit a lunch break, consistent enough to build a habit.
- **Open attendance:** Anyone could drop in through a standing meeting link. No mandatory invite lists, no manager sign-off.
- **No slide tax:** A live terminal, a code walkthrough, a shared diagram, or an open discussion counted just as much as a formal deck.
- **Recorded by default:** Busy engineers could catch up later when delivery deadlines conflicted with Tuesday lunch.

Letting Peers Shape the Backlog

A solo organizer can carry a forum for a month or two. After that, guessing what people want to learn becomes exhausting.

Early on, I ran a short peer interest survey with two questions:

1. **What subjects do you want to learn about?** Responses spanned programming languages (Ruby, Go, JavaScript), core design concepts (SOLID, domain-driven design), relational and NoSQL databases, runtimes, networking, shell scripting, DevOps tooling, event-driven systems, and security.

2. **Which formats help you learn best?** Peers consistently ranked interactive demos, live coding, and informal walkthroughs alongside or ahead of traditional slide presentations.

That survey did two things at once. It gave me a backlog grounded in real curiosity, and it gave quiet engineers permission to suggest topics they wanted to see without volunteering to lecture on day one.

Designing for Organizer Sustainability

Many grassroots meetups burn out after their first quarter. The organizer scrambles every Monday night for a topic, misses two weeks, and quietly cancels the series.

Two habits kept Geekfest@OMF running for 18 months across 49+ recorded sessions:

- **Scheduled retrospectives:** The initial schedule only ran through late March 2023. On **March 28, 2023**, we paused to celebrate what we had covered, review survey results, and plan the next stretch together.
- **Deliberate breaks:** In **July 2023**, I called a planned summer break from formal presentations while leaving the Tuesday call open for casual conversation. Naming the pause prevented guilt and kept the habit intact.

Those small structural choices turned a personal experiment into a reliable weekly space. In Part 2, I break down the actual curriculum we covered and why mixing fundamentals, polyglot demos, and shared hygiene kept attendance strong.

Geekfest@OMF, Part 2: A Practical Curriculum for Working Engineers

This is **Part 2** of a four-part series documenting **Geekfest@OMF**. In Part 1, I covered the Obtiva lineage and zero-friction design. Here I look at the actual curriculum across 18 months and 49+ recorded sessions, and how those Tuesdays intersected with real production architecture work on the Acquisition Lane.

Balance Variety With Depth

A single-topic lunch group quickly shrinks to one squad. A purely random group feels scattered and forgettable.

To keep engineers returning from across the organization, Geekfest@OMF rotated across four practical tracks. Every session focused on something a working developer could try that same afternoon.

Track 1: Reading the Real System

Many debugging habits rely on guessing or copying snippets from search results. Several early sessions pushed the opposite habit: read the live system directly.

- **Reading Ruby (Nov 1, 2022):** The kickoff session skipped slides completely. We used `gem open` and `bundle open` to inspect library source code in a live environment instead of guessing how a dependency behaved.
- **Near, HAR, Wherever You Are (Mar 14, 2023):** "What happened in the browser?" often ends in a shrug. We walked through recording and exporting HTTP Archive (HAR) files from browser developer tools so frontend and backend engineers could inspect the exact network exchange.
- **A Bundle of Gems (Mar 21, 2023):** Every Rubyist runs `bundle install`, but few inspect how RubyGems and Bundler resolve dependencies. We built a minimal gem from scratch to demystify packaging and supply-chain risk.
- **Before, After, and Around with RSpec (Oct 3, 2023):** Large test suites hide surprising setup costs. We traced how RSpec executes test hooks so teams could write clearer, faster test setups.

Track 2: Design Fundamentals and Sense-Making

Frameworks change every few years. Design principles and problem-framing skills age much better.

- **Working Effectively with Legacy Code (Nov 15, 2022):** We watched and discussed Michael Feathers' core ideas on seams, characterization tests, and making safe changes in mature codebases.
- **The 5-Part SOLID Series (Dec 2022 to Jan 2023):** Rather than rushing through all five principles in one slide deck, we gave each letter its own 45-minute Tuesday: Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion. Spreading the arc across five weeks let engineers bring real refactoring questions back each Tuesday.

Curated Reference Talks from the SOLID Arc

To ground the Liskov Substitution (Jan 17, 2023) and Interface Segregation (Jan 24, 2023) sessions in primary sources and concrete low-level design examples, I maintained YouTube study playlists alongside the Tuesday discussions. Those public reference talks anchor the curriculum in time (metrics synced November 27, 2026):

- How Data Abstraction Changed Computing Forever (Barbara Liskov, TEDxMIT) (published Jul 2019, 21m 39s, 40.7k views)
- Liskov: The Liskov Substitution Principle (published Apr 2020, 4m 23s, 29.5k views)
- Low Level Design 106: Inheritance and Polymorphism (published Apr 2022, 8m 58s, 38.8k views)
- Low Level Design 107: Liskov Substitution Principle (published May 2022, 9m 05s, 88.4k views)
- Low Level Design 108: Interface Segregation Principle (published May 2022, 7m 50s, 49.1k views)

- **Cynefin, Kind of Fun (Aug 8, 2023):** Technical debates often stall because teams treat complex problems as simple checklists. We introduced David Snowden's Cynefin sense-making framework to give squads shared language for triaging work before committing to a solution.

Track 3: Databases and Polyglot Literacy

Enterprise platforms rarely run on one language. Engineers in one lane benefit when they understand the constraints of neighboring systems.

- **Databases and Practical PostgreSQL (Nov 22, 2022 & May 9, 2023):** We started with how relational databases store, index, and query data. Later, we ran a hands-on PostgreSQL session (`SELECT * FROM "SQL"`) covering foreign keys, triggers, audit tables, materialized views, join types, and real-time `LISTEN/NOTIFY` channels wired to a small Ruby `pg` listener.
- **Polyglot Exploration:** Sessions walked through building a small CLI in Go (*Do not pass Go(lang)*), demystifying Lisp syntax in Clojure (*Don't fear the `()`*), bridging Rails experience to Ionic and Node.js, and exploring TypeScript modules with peer guest speakers.
- **Cross-Boundary Infrastructure:** We covered Redis-backed feature flags with Rollout, server-side session state evaluation with DynamoDB, GraphQL load testing with Grafana k6, a dedicated Kafka fundamentals webinar, and a peer walkthrough of bridging modern APIs to z/OS mainframe backends.

Each of those sessions lowered the fear of unfamiliar boundaries. When an application engineer understands how a mainframe connector or a load test works, cross-team conversations get calmer and faster.

Track 4: Everyday Craft and Shared Hygiene

Some of the highest-energy Tuesdays had nothing to do with frameworks. They focused on the daily tools every engineer touches.

- **Spreadsheets. No, really. (Nov 8, 2022):** Before reaching for a custom microservice, ask whether a spreadsheet solves the problem faster.
- **Consoles, Terminals, and Shells, Oh My (ZSH)! (Apr 18, 2023):** Clarifying the difference between terminals, consoles, Bash, and Zsh, plus practical shell automation.
- **Editor Wars (May 23, 2023):** Five-to-fifteen-minute lightning demos where colleagues showed why they loved Vim, Emacs, VS Code, or RubyMine. Lighthearted, high-participation, and packed with workflow tips.
- **Regular Expressions (Jun 20, 2023):** A practical tour from `\A` to `\z` without mystique.
- **Spring Cleaning: Workstation Setup Docs (Feb 7, 2023):** One Tuesday, we skipped lectures altogether and spent 45 minutes updating stale developer onboarding documentation together.

Turning a brown-bag into live maintenance showed that peer communities can leave shared artifacts behind.

How the Curriculum Connected to Production Work

None of these Tuesdays were detached academic lectures. They tracked directly alongside the real engineering problems documented in my OneMain Financial position record and architecture case studies:

- **Debugging a distributed monolith:** Sessions on reading Ruby dependency source, exporting browser HAR traces, and Michael Feathers' legacy seams built shared diagnostic habits across teams maintaining decade-old acquisition workflows.
- **Session state resilience:** The May 2023 peer walkthrough evaluating DynamoDB for server-side session state intersected directly with diagnosing and eliminating a Rails session-cookie overflow defect affecting 4% of digital applicant traffic via a zero-downtime migration to AWS DynamoDB.
- **Enablement through shared hygiene:** Live workstation documentation maintenance and relational database internals reflected the Acquisition Lane enablement team's day-to-day focus on making legacy boundaries legible and safer for delivery squads to change.

What Made the Curriculum Work

Three patterns stood out across those 18 months:

1. **Playful titles lowered the stakes.** Calling a session *Near, HAR, Wherever You Are* or *Do not pass Go(lang)* signaled curiosity over corporate formality.
2. **Guest speakers multiplied reach.** Inviting peers from other teams to demo TypeScript, DynamoDB, GraphQL load testing, or mainframe API connectors broke single-speaker dependency.
3. **Hands-on beats abstract.** Live terminal sessions, scratch repositories, and shared problem solving produced far better discussion than polished slide decks.

In Part 3, I look at the two emerging topics that shifted Geekfest@OMF from general peer learning into an enterprise catalyst: privacy-conscious local AI and OpenTelemetry.

Geekfest@OMF, Part 3: Early Local-First AI and OpenTelemetry Demos

This is **Part 3** of a four-part series documenting **Geekfest@OMF**. Once the weekly Tuesday habit was established (Part 1 and Part 2), the room became something more valuable than a study group. It became a safe place to test unfamiliar technology before the organization had formal standards for it.

Between April 2023 and February 2024, two threads stood out: **privacy-conscious local AI** and **OpenTelemetry**.

Exploring Local-First AI Without Hype

In early 2023, Large Language Models moved from research papers into everyday developer conversations overnight. In a regulated financial enterprise, sending internal code or documents to third-party cloud APIs was a non-starter. Waiting passively for policies to settle meant engineers would fall behind on how the tools actually worked.

Geekfest@OMF took a pragmatic middle path: run open models locally and inspect them with engineering rigor.

- **Let's Look More at LLMs (Apr 4, 2023):** We stepped into unfamiliar territory openly. I demonstrated `llama.cpp` running local weights and OpenAI's `Whisper` speech-to-text tool (which I was using to transcribe archival user-group interviews). I explicitly invited ML specialists to join and correct our mental models. Starting with humility kept the room curious instead of defensive.
- **Local LLM with Ollama (Nov 28, 2023):** Seven months later, local tooling had matured dramatically. I walked through integrating **Ollama** into a Ruby on Rails prototype to analyze and score internal documentation while keeping every byte of data on local hardware. We debugged JSON parsing and streaming responses live on the call.
- **Spin up SQL with Vanna.ai (Jan 23, 2024):** We opened 2024 with a Python demo exploring natural-language-to-SQL generation, examining both where it helped and where schema ambiguity still required human judgment.

Those sessions planted an important norm: you can learn emerging AI patterns responsibly in a regulated environment by favoring local-first prototypes and honest boundary checks. That same hands-on thread carried directly into my own work outside and inside the enterprise: transcribing the UGtastic oral history archive locally with Whisper, and placing in two corporate hackathons (building a Rasa conversational agent prototype in 2022 and an OpenTelemetry-instrumented schema inference pipeline with local Ollama summarization in February 2024, recorded on my OneMain Financial position page).

The OpenTelemetry Spark

The second thread changed the trajectory of Geekfest@OMF completely.

In September 2023, distributed tracing was still a distant idea for most application squads. Rather than writing a 30-page architecture proposal, I brought a scratch demo to Tuesday lunch.

- **Looking at OpenTelemetry (Sep 12, 2023):** We generated a brand-new minimal Rails app (`rails new animal_canine_info --minimal`), wired up Ruby OpenTelemetry instrumentation, and exported live OTLP spans to a local collector on the standard OTLP/HTTP port (`4318`). Seeing a real trace waterfall appear in minutes made observability concrete.
- **OpenTelemetry the Demo (Sep 19, 2023):** The following week, we walked through the official multi-service OpenTelemetry Demo application. Engineers could see how traces crossed language and service boundaries in a distributed system.
- **KubeCon + CloudNativeCon Chicago Recap (Nov 21, 2023):** After attending KubeCon in Chicago, I brought field notes back to Geekfest@OMF with a focus on the OpenTelemetry stack and real-world adoption patterns.

KubeCon Chicago 2023 and OpenTelemetry Study References

Alongside the live demos, I tracked public CNCF and conference talks in study watchlists to keep our Collector, sampling, and logging discussions aligned with upstream standards (metadata synced November 27, 2026):

- Connected Observability Pipelines in the OpenTelemetry Collector (Dan Jaglowski) (published May 2023, 26m 52s, 3.4k views)
- Journey from Fluent Bit, Fluentd and Prometheus to OpenTelemetry Collector (Marcin Stozek) (published Nov 2023, 30m 03s, 2.4k views)
- Introducing Logs in OpenTelemetry: What It Unlocks and How to Use It (Morgan McLean & Dan Jaglowski) (published Nov 2023, 35m 56s, 6.2k views)
- OTTL Me Why: Transforming Telemetry in the OpenTelemetry Collector (Tyler Helmuth & Evan Bradley) (published Dec 2023, 28m 33s, 3.1k views)
- Dynamic Sampling in Practice (Kent Quirk, Honeycomb) (published Dec 2023, 30m 47s, 751 views)

Closing the Circle Back to Obtiva

One session in **February 2024** captured the full spirit of the original Obtiva Geekfest.

On **February 13, 2024**, I used Geekfest@OMF to rehearse an external user-group talk I was preparing for **SCMC** (Software Craftsmanship McHenry County), titled "*From Monitoring to Observability with OpenTelemetry.*" Colleagues reviewed the outline, challenged unclear transitions, and sharpened the story from traditional monitoring to context-rich observability.

That single session closed a 15-year loop. At Obtiva in 2009, we used Tuesday Geekfest to practice talks for Chicago user groups. In 2024, Geekfest@OMF served the exact same role, feeding lessons between the external software craftsmanship community and the internal engineering organization.

Why Small Demos Beat Big Mandates

Looking back at those nine months, a clear pattern emerges:

1. **Scratch apps remove politics.** Demonstrating OpenTelemetry on a throwaway Rails app meant nobody felt criticized about their production codebase.
2. **Local-first prototypes lower compliance anxiety.** Running Ollama or a local OTLP collector on a laptop let engineers test real mechanics safely.
3. **Curiosity builds coalition before budget exists.** By the time leadership was ready to sponsor formal observability work, engineers across multiple lanes already shared vocabulary and working mental models.

In Part 4, I trace what happened next: how those Tuesday OpenTelemetry demos grew into the company's first Community of Practice and how we handed it off to permanent full-time stewardship.

Geekfest@OMF, Part 4: From Brown-Bag to Community of Practice and Handoff

This is **Part 4** of a four-part series documenting **Geekfest@OMF**. In Part 3, I described how scratch OpenTelemetry demos sparked cross-team curiosity. Here I trace how that energy matured into an enterprise Community of Practice and ended with a deliberate stewardship handoff.

The Secret Value Starts Before the First Line of Code

Most organizations treat observability as a tooling purchase. Pick a vendor, install an agent, and wait for dashboards.

In a long-lived enterprise with heterogeneous stacks (Ruby on Rails, Node.js, Java, API gateways, Kafka, and mainframes), that approach stalls fast. The real value of **OpenTelemetry** appears before the first integration lands: **it forces siloed teams to talk to each other.**

You cannot trace a customer journey across six lanes without asking your neighbors how their service works, what headers survive their gateway, and what constraints they live under. Conway's Law reminds us that systems mirror the communication paths of the organization that built them. If you want a connected, observable distributed system, you have to build connected human communication paths first.

Scaling Into a Representative Forum

In **April 2024**, the observability thread outgrew a general Tuesday brown-bag and spun out as the **OpenTelemetry Working Group (OTel WG)**, establishing the first formal Community of Practice at OneMain Financial.

Over the next 17 months, the weekly Friday forum grew to **122 invited members**, with **40+ active engineers** participating across six product and platform lanes. That pattern soon inspired parallel Communities of Practice for event streaming, workflow orchestration, and domain API gateways.

The working group worked because it acted less like a top-down committee and more like a **representative forum**, closer to a United Nations of practitioners than an architecture board:

- **Two-way representation:** Engineers brought real integration blockers from their squads into the room, compared notes with adjacent teams, and carried shared patterns back home.
- **Respect for prior constraints:** Long-lived enterprises carry decisions made under past deadlines and technologies. Mocking legacy choices shuts down collaboration; acknowledging different realities opens it up.
- **Practical pairing over mandates:** Sessions focused on active work: propagating W3C Trace Context headers, correlating logs and spans across language boundaries, and verifying telemetry hygiene together.

Writing a Plain-Language Code of Conduct

As attendance grew past a few dozen engineers, implicit norms were no longer enough. In January 2025, I drafted a concise, one-page **Code of Conduct** for the working group.

I did not want a legalistic document. A first-time visitor needed to read one page and know how the room operated. Five principles anchored it:

1. **Assume good intent, even in stark disagreement.** Every squad operates under different constraints and risk tolerances.
2. **Leave title hierarchy at the door.** Good ideas cannot emerge in a room dominated by senior titles.
3. **Make corporate newcomers feel safe.** For many engineers hired into traditional enterprise roles, speaking up in an open cross-team forum is already a big step outside their comfort zone.
4. **Clarify what a working group is.** It is a collaborative connection point, not a dedicated delivery team that absorbs everyone else's backlog.
5. **Protect the practitioners doing the work.** Education and curiosity belong in the room, while active integration blockers get priority so real teams stay unblocked.

The September 2025 Stewardship Handoff

A grassroots community succeeds when it makes itself unnecessary as a volunteer side project.

By mid-2025, cross-lane adoption and working production traces had proved the strategic value of unified observability. Executive leadership funded **two dedicated full-time roles** (a Director and a Staff Engineer) and structured the effort into **three focused working groups** under permanent operational stewardship.

On **September 12, 2025**, I held my final OpenTelemetry Working Group session and formally handed off stewardship.

Stepping back from something you built over three years is never purely mechanical. It takes discipline to let a community evolve under new owners without clinging to control. Yet that handoff was the whole point from day one: turning a Tuesday lunch experiment from November 2022 into permanent institutional capability that no longer depends on one person's calendar.

What This Arc Means in Practice as a Staff Software Engineer

On my canonical resume, OneMain Financial role record, and Staff Platform & Enablement Lead profile, this three-year community arc is the human half of the same architecture work. Without direct reports or top-down authority, hands-on technical leadership still created measurable organizational leverage:

- **Scope:** Bridged six product and platform lanes spanning Ruby on Rails, Node.js, API gateways, Kafka, and mainframes.
- **Leverage:** Replaced isolated squad experiments with shared W3C Trace Context patterns, pairing loops, and a repeatable Community of Practice template.
- **Durability:** Ended not with a single champion burning out, but with two newly funded full-time roles and three permanent working groups carrying the work forward.

Four Lessons for Staff Engineers Building Internal Communities

If you are trying to spark a similar arc inside a large engineering organization, four lessons travel well:

1. **Start small and drop the ceremony.** A 45-minute weekly session with live demos beats a quarterly summit every time.
2. **Let curiosity lead, then specialize.** A broad peer forum (like Geekfest@OMF) builds trust first; focused working groups (like the OTel WG) turn that trust into shared architecture.
3. **Design for representation, not heroism.** Rotate voices, write down clear behavioral norms, and connect squads horizontally.

4. **Plan your exit early.** Measure success by whether the organization eventually funds and owns the capability without you.

External Talks That Informed This Arc

During the 2024 to 2025 working group phase, these public talks from my Conway's Law and OpenTelemetry study playlists shaped how I framed team topology and observability culture (metadata synced November 27, 2026):

- How I Finally Stopped Worrying & Learnt to Love Conway's Law (James Lewis, GOTO 2015) (published Jul 2015, 40m 53s, 19.6k views)
- Conway's Law: When Best Practices Are Not Enough (Julien Topcu) (published Jan 2024, 49m 04s, 758 views)
- Shift Into an Observability Mindset with OpenTelemetry (Daniel Gomez Blanco, Skyscanner) (published Mar 2024, 22m 06s, 1.1k views)
- Software Architectures, Team Topologies & Complexity Science (James Lewis, GOTO 2023) (published Jun 2024, 50m 26s, 30.6k views)