ÜberConf

THE ULTIMATE CONFERENCE FOR SOFTWARE DEVELOPERS AND ARCHITECTS

July 14 - 17, 2026 - Denver, CO

Software Architecture · Generative AI · Modern Java · Leadership · Security · GitHub · DevOps · Microservices · Kubernetes · Developer Productivity Engineering · Machine Learning · Gradle

We are excited to come together again for in-person events! We will no longer offer on-line attendance. However, sessions will be recorded for replay by those who attend in-person.

Why Attend ÜberConf?

ÜberConf is our flagship educational event for software engineers and technical leaders. With 9 concurrent tracks, 100+ sessions, and 9 full-day workshops; ÜberConf is truly the ultimate destination for passionate technologists.

Learn from the Best Speakers

Our speakers are not vendor representatives. They are industry recognized experts. They are published authors, consultants, executives, and open source leaders.

In-Depth 90-Minute Sessions

Our longer session format, workshops, and multi-part sessions allow speakers to go in-depth and teach the detailed concepts you need to know.

Agile Practices

Speakers at ÜberConf emphasize and present on topics such as: Test Driven Development, Continuous Integration, Code Quality Measurements, Code Smells, Team Building, and Customer Collaboration.

Generative AI

Generative AI is revolutionizing how we create and interact with information, with the potential to dramatically increase productivity and innovation across various sectors.

Understand Web Security

The web is an increasingly hostile environment for web applications. ÜberConf will include security focused sessions and workshops so you will understand best security practices.

Cloud Architectures

ÜberConf will explore different cloud computing architectures and how you can take advantage of them.

https://uberconf.com

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Tuesday, Jul. 14

8:00 - 9:00 AM : UBERCONF 2026: EARLY REGISTRATION - OPTIONAL ONE DAY WORKSHOP

9:00 - 5:00 PM - Sessions

Session #1 : The Building Plan Thinking Architecturally by Raju Gandhi

You are ready to level up your skills. Or, you've already been playing _accidental_ architect, and need to have a structured plan to be designated as one. Well, your wait is over. From the author of O'Reilly's best-selling "Head First Software Architecture" comes a full-day workshop that covers all that you need to start thinking architecturally. Everything from the difference between design and architecture, and modern description of architecture, to the skills you'll need to develop to become a successful architect, this workshop will be your one stop shop.

Session #2 : System Design AI Mastery Architecting for Scale Speed Reliability Full Day by Rohit Bhardwaj

Modern system design has entered a new era. It’s no longer enough to optimize for uptime and latency — today’s systems must also be AI-ready, token-efficient, trustworthy, and resilient. Whether building global-scale apps, powering recommendation engines, or integrating GenAI agents, architects need new skills and playbooks to design for scale, speed, and reliability. This full-day workshop blends classic distributed systems knowledge with AI-native thinking. Through case studies, frameworks, and hands-on design sessions, you’ll learn to design systems that balance performance, cost, resilience, and truthfulness — and walk away with reusable templates you can apply to interviews and real-world architectures.

Session #3 : Thinking Functionally in Java and the JVM by Daniel Hinojosa

Java has quietly absorbed functional ideas over the last decade. Lambdas, streams, records, sealed types. It has been an amazing journey, but most teams still write code as if none of that really changed anything. This workshop asks a simple question: what if we actually took those features seriously? In Thinking Functionally in Java, we explore how far disciplined functional design can take us using plain Java with no rewrites, no new language mandates, and no academic detours. Along the way, we address reproducible development environments with Nix, replace exception-driven control flow with explicit error modeling, and uncover why concepts like flatMap, algebraic data types, and composability matter even if you never say the word “monad” out loud.

Session #4 : HandsOn AI Agent Engineering Habits for safe explainable domaingrounded AI by Russell Miles

AI agents are moving from novelty to necessity — but building them safely, predictably, and observably requires more than clever prompts. This workshop gives developers a practical introduction to AI agent engineering using Embabel, with an emphasis on the habits, patterns, and mental models needed to design trustworthy agents in real systems. Across two focused sessions, you’ll learn how to ground agents in strong domain models (DICE), design goal-driven behaviours (GOAP), enforce safety through invariants and preconditions, and make every action explainable through observability. You’ll run and inspect a fully working reference agent, extend its domain, add new actions, and validate behaviour through explainable planning logs. You'll explore how to select and deploy models tuned to provide the best and most cost-effective agent behaviour for your users.

Session #5 : Bootiful Spring Boot the DeepDive by Josh Long

Hi, Spring fans! Developers today are being asked to deliver more with less time and build ever more efficient services, and Spring is ready to help you meet the demands. In this workshop, we'll take a roving tour of all things Spring, looking at fundamentals of the Spring component model, look at Spring Boot, and then see how to apply Spring in the context of batch processing, security, data processing, modular architecture, miroservices, messaging, AI, and so much more.

Session #6 : AI Security for Developers and Practitioners full day by Brent Laster

This full-day, hands-on workshop equips developers, architects, and technical leaders with the knowledge and skills to secure AI systems end-to-end — from model interaction to production deployment. Participants learn how to recognize and mitigate AI-specific threats such as prompt injection, data leakage, model exfiltration, and unsafe tool execution. Through a series of focused labs, attendees build, test, and harden AI agents and Model Context Protocol (MCP) services using modern defensive strategies, including guardrails, policy enforcement, authentication, auditing, and adversarial testing. The training emphasizes real-world implementation over theory, using preconfigured environments in GitHub Codespaces for instant, reproducible results. By the end of the day, participants will have created a working secure AI pipeline that demonstrates best practices for trustworthy AI operations and resilient agent architectures.

Session #7 : Personal Knowledge Management PKM for the Modern Knowledge Worker by Michael Carducci

Join us for a transformative handson workshop on Personal Knowledge Management (PKM), designed specifically to empower developers, architects, and knowledge workers alike to master information in this information age. Based on Tiago Forte's Building a Second Brain methodology and implemented using the Logseq PKM application, this course aims to equip attendees with the strategies, tools, and insights to streamline their knowledge management, increase productivity, and stimulate creativity. Attendees will learn to construct a personal knowledge graph, effectively annotate and reference digital assets, manage tasks, journal for success, leverage templates, and much more. The ultimate goal is to create a personalized system that enables you to instantly find or recall everything you know and learn.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #8 : Staying Grounded in the Chaos The 4 Skills Every Tech Leader Needs by Hunter Milligan

Leadership isn't just about making the right calls — it's about staying steady while everything around you moves fast. From managing tension in meetings, juggling shifting priorities, and fielding last-minute requests, to navigating unclear direction or supporting a stressed-out team, software leaders are constantly pulled in multiple directions. That's where FLOW comes in: a practical, four-part skillset designed to help you handle pressure, stay grounded in uncertainty, and show up with clarity when it matters most.

5:00 - 6:30 PM : MAIN UBERCONF 2026 REGISTRATION - WESTMINSTER BALLROOM FOYER

6:30 - 7:20 PM : UBERCONF 2026: OPENING NIGHT DINNER - LEGACY BALLROOM

7:20 - 7:30 PM : UBERCONF 2026 WELCOME/ANNOUNCEMENT - LEGACY BALLROOM

7:30 - 8:30 PM : UBERCONF 2024: OPENING NIGHT KEYNOTE - LEGACY BALLROOM

8:30 - 10:00 PM : UBERCONF 2026: OPENING NIGHT OUTDOOR RECEPTION - SOUTH COURTYARD

Wednesday, Jul. 15

7:30 - 8:30 AM : UBERCONF 2026 WEDNESDAY: BREAKFAST (LEGACY BALLROOM) & LATE REGISTRATION (WESTMINSTER BALLROOM FOYER)

8:30 - 10:00 AM - Sessions

Session #9 : Architecture in the Age of AI by Michael Carducci

As code generation becomes increasingly automated, our role as developers and architects is evolving. The challenge ahead isn't how to get AI to write more code, it's how to guide it toward coherent, maintainable, and purposeful systems.

Session #10 : AI Inference at Scale Reliability Observability Cost Sustainability by Rohit Bhardwaj

AI inference is no longer a simple model call—it is a multi-hop DAG of planners, retrievers, vector searches, large models, tools, and agent loops. With this complexity comes new failure modes: tail-latency blowups, silent retry storms, vector store cold partitions, GPU queue saturation, exponential cost curves, and unmeasured carbon impact.

Session #11 : AI 3in1 Agents RAG and Local Models by Brent Laster

Learning and understanding AI concepts is satisfying and rewarding, but the fun part is learning how to work with AI yourself. In this 1/2 day workshop, author, trainer, and experienced technologist Brent Laster will help you do both! We'll explain why and how to run AI models locally, the basic ideas of agents and RAG, and show how to assemble a simple AI agent in Python that leverages RAG and uses a local model through Ollama. And you'll get to follow through with hands-on labs and produce your own instance running on your system in a GitHub Codespace

Session #12 : Whats New In Java HalfDay Part 1 by Daniel Hinojosa

Java has quietly grown into a more expressive, flexible, and modern language — but many developers haven't kept up with the latest features. This two-part workshop explores the most useful additions to Java from recent releases, with hands-on examples and real-world scenarios. Whether you're still catching up from Java 8 or already using Java 21+, this series will give you a practical edge in writing cleaner, more modern Java code.

Session #13 : The Role of Enterprise Architecture in an everchanging landscape by Raju Gandhi

Enterprise Architecture (EA) has long been misunderstood as a bottleneck to innovation, often labeled the "department of no." But in today's fast-paced world of Agile, DevOps, Cloud, and AI, does EA still have a role to play—or is it a relic of the past?

Session #14 : Bootiful Spring Boot A DOGumentary by Josh Long

Spring Boot 3.x and Java 21 have arrived, making it an exciting time to be a Java developer! Join me, Josh Long (@starbuxman), as we dive into the future of Spring Boot with Java 21. Discover how to scale your applications and codebases effortlessly. We'll explore the robust Spring Boot ecosystem, featuring AI, modularity, seamless data access, and cutting-edge production optimizations like Project Loom's virtual threads, GraalVM, AppCDS, and more.

Session #15 : Mastering Platform Engineering From Design to Value by Russell Miles

Internal developer platforms hold promise: faster delivery, better reliability, happier engineers. Yet they often stall—caught in organisational inertia, tool complexity, and unclear value. These two high-impact 90-minute sessions give you the core mental models and practical artefacts to shift from tool-chase to strategic platform value. You'll walk away with real maps, a clear focus, and next-step experiments—not just ideas. If you're looking to make your platform team a force for value, not just operations, this compact format delivers. By the end of the workshop you'll know: How to see your platform ecosystem clearly using tools like Wardley Mapping, User Needs Mapping, Value Stream Mapping and OODA loops. How to treat the platform as a product, shifting mindset from internal project to self-service, developer-centric product. How to apply the pattern language of platform design (Golden Path, Self-Service, Abstraction,

Composability, Guardrails, Observability, Extensibility, Incremental Roll-out). How to use DSRP + UNM + VSM to reveal where value stalls, flow breaks, and cognitive load spikes. How to design smallest viable changes, build an impact roadmap, and influence adoption through “Elephant & Rider” thinking (rational vs emotional mindsets). You'll also get a sneak peek into the future of platforms: AI/automation, evolving loops, and building resilience into your ecosystem. If your platform team is stuck in the grind—shipping tickets, fighting fires, juggling tools, and wondering why nothing ever seems to change—this workshop will give you the clarity and leverage you’ve been missing. You’ll learn to read your organisation like a map: where value flows, where it dies, where cognitive load spikes, and where small, strategic platform moves can unlock disproportionate impact. This isn’t another "tools tour". Whether you’re building an IDP from scratch or rescuing one that’s drifting, you’ll leave with a clear roadmap, a set of tested patterns, and the influence skills to actually make the work land. Platform engineering is no longer about managing complexity—it’s about creating the conditions where developers can thrive. Join us, map your ecosystem, design the future, and turn your platform team into the strategic engine your organisation needs.

Session #16 : Automating Security Fixes with OpenRewrite by Brian Sletten

Security problems empirically fall into two categories: bugs and flaws. Roughly half of the problems we encounter in the wild are bugs and about half are design flaws. A significant number of the bugs can be found through automated testing tools which frees you up to focus on the more pernicious design issues. In addition to detecting the presence of common bugs as we have done with static analysis for years, however, we can also imagine automating the application of corrective refactoring. In this talk, I will discuss using OpenRewrite and the Moderne cli to fix common security issues and keep them from coming back.

Session #17 : Core Leadership Principles by Robert Harris

In moments of uncertainty, teams don’t listen more closely to their leaders. *They watch them.* Across years of leading software organizations - and hundreds of documented leadership moments - one truth becomes unavoidable: engineers take their emotional cues from leadership behavior, especially under pressure. Stress, urgency, and fear propagate through teams not by announcement, but by example.

10:00 - 10:30 AM : UBERCONF 2026 - WEDNESDAY: AM BREAK

10:30 - 12:00 PM - Sessions

Session #18 : 3rd Generation Agentic AI by Michael Carducci

AI models are evolving fast, but the systems around them aren’t. Every backend change still breaks your carefully tuned AI client, while on the web, every change to a server doesn’t require you to download a new browser. What if AI worked the same way?

Session #19 : Dynamic Programming Demystified How AI Helps You See the Pattern by Rohit Bhardwaj

Dynamic Programming (DP) intimidates even seasoned engineers. With the right lens, it’s just optimal substructure + overlapping subproblems turned into code. In this talk, we start from a brute-force recursive baseline, surface the recurrence, convert it to memoization and tabulation, and connect it to real systems (resource allocation, routing, caching). Along the way you’ll see how to use AI tools (ChatGPT, Copilot) to propose recurrences, generate edge cases, and draft tests—while you retain ownership of correctness and complexity. Expect pragmatic patterns you can reuse in interviews and production.

Session #20 : AI 3in1 Agents RAG and Local Models (continued) by Brent Laster

Learning and understanding AI concepts is satisfying and rewarding, but the fun part is learning how to work with AI yourself. In this 1/2 day workshop, author, trainer, and experienced technologist Brent Laster will help you do both! We’ll explain why and how to run AI models locally, the basic ideas of agents and RAG, and show how to assemble a simple AI agent in Python that leverages RAG and uses a local model through Ollama. And you'll get to follow through with hands-on labs and produce your own instance running on your system in a GitHub Codespace

Session #21 : Whats New In Java HalfDay Part 1 (continued) by Daniel Hinojosa

Java has quietly grown into a more expressive, flexible, and modern language — but many developers haven’t kept up with the latest features. This two-part workshop explores the most useful additions to Java from recent releases, with hands-on examples and real-world scenarios. Whether you’re still catching up from Java 8 or already using Java 21+, this series will give you a practical edge in writing cleaner, more modern Java code.

Session #22 : Avoiding the Traps AntiPatterns in Modern Software Architecture by Raju Gandhi

Architecture is often defined as "hard to change". Within software architecture, an architecture pattern is a reusable solution to a commonly occurring problem in software architecture within a specific context. Architecture anti-patterns are their diabolical counterparts —wherein they sound good in theory, but in practice lead to negative consequences. And given that they affect both the architectural characteristics and the structural design of the system, are incredibly expensive and have far-reaching consequences.

Session #23 : Bootiful Spring AI by Josh Long

The age of artificial intelligence (because the search for regular intelligence hasn't gone well..) is nearly at hand, and it's everywhere! But is it in your application? It should be. AI is about integration, and here the Java and Spring communities come second to nobody.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #24 : Mastering Platform Engineering From Design to Value (continued) by Russell Miles

Internal developer platforms hold promise: faster delivery, better reliability, happier engineers. Yet they often stall—caught in organisational inertia, tool complexity, and unclear value. These two high-impact 90-minute sessions give you the core mental models and practical artefacts to shift from tool-chase to strategic platform value. You'll walk away with real maps, a clear focus, and next-step experiments—not just ideas. If you're looking to make your platform team a force for value, not just operations, this compact format delivers. By the end of the workshop you'll know: How to see your platform ecosystem clearly using tools like Wardley Mapping, User Needs Mapping, Value Stream Mapping and OODA loops. How to treat the platform as a product, shifting mindset from internal project to self-service, developer-centric product. How to apply the pattern language of platform design (Golden Path, Self-Service, Abstraction, Composability, Guardrails, Observability, Extensibility, Incremental Roll-out). How to use DSRP + UNM + VSM to reveal where value stalls, flow breaks, and cognitive load spikes. How to design smallest viable changes, build an impact roadmap, and influence adoption through "Elephant & Rider" thinking (rational vs emotional mindsets). You'll also get a sneak peek into the future of platforms: AI/ automation, evolving loops, and building resilience into your ecosystem. If your platform team is stuck in the grind—shipping tickets, fighting fires, juggling tools, and wondering why nothing ever seems to change—this workshop will give you the clarity and leverage you've been missing. You'll learn to read your organisation like a map: where value flows, where it dies, where cognitive load spikes, and where small, strategic platform moves can unlock disproportionate impact. This isn't another "tools tour". Whether you're building an IDP from scratch or rescuing one that's drifting, you'll leave with a clear roadmap, a set of tested patterns, and the influence skills to actually make the work land. Platform engineering is no longer about managing complexity—it's about creating the conditions where developers can thrive. Join us, map your ecosystem, design the future, and turn your platform team into the strategic engine your organisation needs.

Session #25 : Automating Security Fixes with OpenRewrite (continued) by Brian Sletten

Security problems empirically fall into two categories: bugs and flaws. Roughly half of the problems we encounter in the wild are bugs and about half are design flaws. A significant number of the bugs can be found through automated testing tools which frees you up to focus on the more pernicious design issues. In addition to detecting the presence of common bugs as we have done with static analysis for years, however, we can also imagine automating the application of corrective refactoring. In this talk, I will discuss using OpenRewrite and the Moderne cli to fix common security issues and keep them from coming back.

Session #26 : Delegation Without Guilt The Skills to Reclaim Focus by Hunter Milligan

If your to-do list keeps growing because "it's just faster to do it myself," this workshop is for you. Delegation isn't just about getting things off your plate—it's about building others up, focusing on the work only you can do, and staying sane in the process. You'll learn a practical 7 Steps of Delegation—a clear, practical framework for how to empower your team without micromanaging (that's time consuming, and no one likes it!). From clarifying the why behind a task to setting outcomes instead of instructions, you'll discover how to hand off work with trust, clarity, and accountability. You'll practice how to invite questions, listen with empathy, and create ownership while still keeping your team aligned on milestones and results.

12:00 - 1:00 PM : UBERCONF 2026 - WEDNESDAY: LUNCH - LEGACY BALLROOM & SOUTH COURTYARD

1:00 - 2:30 PM - Sessions

Session #27 : The Art of Being an Architect by Michael Carducci

The hardest part of software architecture isn't the technology, it's the people. Every architecture lives or dies by its ability to influence behavior, build consensus, and turn vision into change. In this session, Michael Carducci explores the real work of being an architect: communicating clearly, guiding decisions, and driving meaningful change in complex organizations. Drawing from decades of experience and the principles behind the Tailor-Made Architecture Model, Carducci shows how to identify where change is needed, package ideas for adoption, and lead with both clarity and empathy.

Session #28 : Enforcing Architecture Decisions Using Tests by Javiera Laso

In the fast-paced world of software development, maintaining architectural integrity is a continuous challenge. Over time, well-intended architectural decisions can erode, leading to unexpected drift and misalignment with original design principles. This hands-on workshop will equip participants with practical techniques to enforce architecture decisions using tests. By leveraging architecturally-relevant testing, attendees will learn how to proactively guard their system's design, ensuring consistency, scalability, and security as the codebase evolves. Through interactive exercises and real-world examples, we will explore how testing can serve as a powerful tool for preserving architectural integrity throughout a project's lifecycle. Key Takeaways Participants will learn to: Write architecture-driven tests that validate and enforce design constraints. Identify architectural drift early and prevent unintended changes. Maintain consistent, scalable, and secure architectures over time. Collaborate effectively within teams to sustain architectural excellence. Prerequisites Basic Understanding of Software Architecture: Familiarity with architectural patterns and principles Experience with Automated Testing: Understanding of unit, integration, or system testing concepts Collaboration and Communication Skills: Willingness to engage in discussions and teamwork Experience working with Java Optional Familiarity with Static Analysis and Code Quality Tools: Knowledge of tools like ArchUnit, SonarQube, or custom linters is beneficial but not required Experience with Large-Scale Systems: Prior work on complex systems can enhance the

Session #29 : AIPowered Coding with Claude Code by Brent Laster

This 1/2 day workshop introduces participants to Claude Code, Anthropic's AI-powered coding assistant. In three hours, attendees will learn how to integrate Claude Code into their development workflow, leverage its capabilities for productivity, and avoid common pitfalls. The workshop also introduces the concept of subagents (specialized roles like Planner, Tester, Coder, Refactorer, DocWriter) to show how structured interactions can improve accuracy and collaboration.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #30 : Whats New In Java HalfDay Part 2 by Daniel Hinojosa

Java has quietly grown into a more expressive, flexible, and modern language — but many developers haven't kept up with the latest features. This two-part workshop explores the most useful additions to Java from recent releases, with hands-on examples and real-world scenarios. Whether you're still catching up from Java 8 or already using Java 21+, this series will give you a practical edge in writing cleaner, more modern Java code.

Session #31 : Web Security for APIs by Brian Sletten

There's a clear need for security in the software systems that we build. The problem for most organizations is that they don't want to spend any money on it. Even if they did, they often have no idea how much to spend. No particular initiative is likely to imbue your system with "security", but a strong, deep defensive approach is likely to give you a fighting chance of getting it right. Web Security as applied to APIs in particular are an important part of the plan. In this workshop, we'll show you how approaches to defining "enough" as well as concrete techniques to employ incrementally in your designs. In this workshop, we will pick a hands on framework for implementation, but the ideas will generally be standards-based and transcend technology choice so you should have a strategy for mapping the ideas into your own systems.

Session #32 : HandsOn Workshop Building Agents with Spring AI MCP Java and Amazon Bedrock by James Ward and Josh Long

In this hands-on workshop you will learn how to build & deploy production-ready AI Agents. You will use Spring AI, MCP, Java, and Amazon Bedrock and learn how to deal with production concerns like observability and security. We will start with basic prompting then expand with chat memory, RAG, and integration through MCP. You will be provided a provisioned cloud environment and step-by-step instructions.

Session #33 : Building Scalable MultiAgentic AI Systems in Java with EventDriven Approach by Mary Grygleski

This talk will guide Java developers through the design and implementation of multi-agent generative AI systems using event-driven principles.

Session #34 : Your API Is Not Ready for AI Yet A Lifecycle Readiness Guide by Rohit Bhardwaj

PIs built for humans often fail when consumed by AI agents.#They rely on documentation instead of contracts, return unpredictable structures, and break silently when upgraded. Large Language Models (LLMs) and autonomous agents need something different: machine-discoverable, deterministic, idempotent, and lifecycle-managed APIs. This session introduces a five-phase API readiness framework—from discovery to deprecation—so you can systematically evolve your APIs for safe, predictable AI consumption.#You'll learn how to assess current APIs, prioritize the ones that matter, and apply modern readiness practices: function/tool calling, schema validation, idempotency, version sunset headers, and agent-aware monitoring.

Session #35 : Progress Over Perfection The Skills to Lead with Confidence by Hunter Milligan

For Tech Leaders, the fear of making mistakes can show up everywhere — from over-polishing a roadmap deck to rewriting your team's code "just to be sure." But leadership isn't about getting it perfect — it's about moving forward, learning fast, and helping others do the same. In this workshop, you'll learn how to adopt a growth mindset: a practical framework that reframes mistakes as part of progress, not failure. We'll explore how perfectionism sneaks into your leadership style — and what it's quietly costing you.

2:30 - 3:00 PM : UBERCONF 2026 - WEDNESDAY: PM BREAK

3:00 - 4:30 PM - Sessions

Session #36 : The Goal Flow Architecture by Michael Carducci

When Eliyahu Goldratt wrote The Goal, he showed how local optimizations (like adding robots to a factory line) can actually decrease overall performance. Today, AI threatens to repeat that mistake in software. We're accelerating coding without improving flow. In this talk, Michael Carducci explores what it means to architect for the goal: continuous delivery of value through systems designed for flow.

Session #37 : Enforcing Architecture Decisions Using Tests (continued) by Javiera Laso

In the fast-paced world of software development, maintaining architectural integrity is a continuous challenge. Over time, well-intended architectural decisions can erode, leading to unexpected drift and misalignment with original design principles. This hands-on workshop will equip participants with practical techniques to enforce architecture decisions using tests. By leveraging architecturally-relevant testing, attendees will learn how to proactively guard their system's design, ensuring consistency, scalability, and security as the codebase evolves. Through interactive exercises and real-world examples, we will explore how testing can serve as a powerful tool for preserving architectural integrity throughout a project's lifecycle. Key Takeaways Participants will learn to: Write architecture-driven tests that validate and enforce design constraints. Identify architectural drift early and prevent unintended changes. Maintain consistent, scalable, and secure architectures over time. Collaborate effectively within teams to sustain architectural excellence. Prerequisites Basic Understanding of Software Architecture: Familiarity with architectural patterns and principles Experience with Automated Testing: Understanding of unit, integration, or system testing concepts Collaboration and Communication Skills: Willingness to engage in discussions and teamwork Experience working with Java Optional Familiarity with Static Analysis and Code Quality Tools: Knowledge of tools like ArchUnit, SonarQube, or custom linters is beneficial but not required Experience with Large-Scale Systems: Prior work on complex systems can enhance the

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #38 : AIPowered Coding with Claude Code (continued) by Brent Laster

This 1/2 day workshop introduces participants to Claude Code, Anthropic's AI-powered coding assistant. In three hours, attendees will learn how to integrate Claude Code into their development workflow, leverage its capabilities for productivity, and avoid common pitfalls. The workshop also introduces the concept of subagents (specialized roles like Planner, Tester, Coder, Refactorer, DocWriter) to show how structured interactions can improve accuracy and collaboration.

Session #39 : Whats New In Java HalfDay Part 2 (continued) by Daniel Hinojosa

Java has quietly grown into a more expressive, flexible, and modern language — but many developers haven't kept up with the latest features. This two-part workshop explores the most useful additions to Java from recent releases, with hands-on examples and real-world scenarios. Whether you're still catching up from Java 8 or already using Java 21+, this series will give you a practical edge in writing cleaner, more modern Java code.

Session #40 : Web Security for APIs (continued) by Brian Sletten

There's a clear need for security in the software systems that we build. The problem for most organizations is that they don't want to spend any money on it. Even if they did, they often have no idea how much to spend. No particular initiative is likely to imbue your system with "security", but a strong, deep defensive approach is likely to give you a fighting chance of getting it right. Web Security as applied to APIs in particular are an important part of the plan. In this workshop, we'll show you how approaches to defining "enough" as well as concrete techniques to employ incrementally in your designs. In this workshop, we will pick a hands on framework for implementation, but the ideas will generally be standards-based and transcend technology choice so you should have a strategy for mapping the ideas into your own systems.

Session #41 : HandsOn Workshop Building Agents with Spring AI MCP Java and Amazon Bedrock (continued) by James Ward and Josh Long

In this hands-on workshop you will learn how to build & deploy production-ready AI Agents. You will use Spring AI, MCP, Java, and Amazon Bedrock and learn how to deal with production concerns like observability and security. We will start with basic prompting then expand with chat memory, RAG, and integration through MCP. You will be provided a provisioned cloud environment and step-by-step instructions.

Session #42 : Orchestrating Intelligence MultiAgentic Design Patterns for Production AI by Mary Grygleski

As generative AI systems evolve from single LLM calls to complex, goal#driven workflows, multi#agent architectures are becoming essential for robust, scalable, and explainable AI applications.

Session #43 : Scaling APIs for Millions of AIDriven Calls by Rohit Bhardwaj

AI agents don't behave like humans. A single prompt can trigger thousands of parallel API calls, retries, and tool chains—creating bursty load, cache-miss storms, and runaway costs. This talk unpacks how to design and operate APIs that stay fast, reliable, and affordable under AI workloads. We'll cover agent-aware rate limiting, backpressure & load shedding, deterministic-result caching, idempotency & deduplication, async/event-driven patterns, and autoscaling without bill shock. You'll learn how to tag and trace agent traffic, set SLOs that survive tail latency, and build graceful-degradation playbooks that keep experiences usable when the graph goes wild.

Session #44 : The Tech Leaders Guide to Difficult Talks That Actually Work by Hunter Milligan

Missed deadlines. Underperforming teammates. Stakeholders pushing too hard. If you're a technical leader, you're probably avoiding at least one conversation that matters. In this workshop, you'll learn a simple, proven 4-step framework (inspired by Nonviolent Communication) to handle tough conversations with clarity, confidence, and zero drama. Think of it like debugging communication: identify what's really going on, name it without blame, and make a request that moves things forward.

4:30 - 5:00 PM : UBERCONF 2026 - WEDNESDAY: PM BREAK

5:00 - 6:30 PM - Sessions

Session #45 : Architect as a Systems Thinker Turning Strategy into Action by Javiera Laso

In this hands-on session, participants will learn how to bridge the gap between technical strategy and execution using systems thinking principles. Through some exercises, software architects will practice mapping business goals, constraints, and feedback loops, then translate them into a clear and adaptable technical roadmap. This presentation focuses on helping architects/engineers to move from abstract vision to actionable outcomes, aligning architecture with value, sequencing initiatives, and communicating trade-offs effectively to stakeholders.

Session #46 : Exploring Architecture Patterns for Modern Systems by Raju Gandhi

An architecture pattern is a reusable solution to a commonly occurring problem in software architecture within a specific context. Architecture patterns affect the "-ilities" of a system, such as scalability, performance, maintainability, and security as well as impact the structural design of the system.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #47 : Incorporating AI into your SDLC by Brent Laster

Just as CI/CD and other revolutions in DevOps have changed the landscape of the software development lifecycle (SDLC), so Generative AI is now changing it again. Gen AI has the potential to simplify, clarify, and lessen the cycles required across multiple phases of the SDLC.

Session #48 : Javas Asynchronous Ecosystem by Daniel Hinojosa

Java has accumulated a diverse toolbox for concurrency and asynchrony over the decades, ranging from classic threads to parallel streams, from `Future` to `CompletableFuture`, and from reactive libraries to the latest innovations, including virtual threads, structured concurrency, and the Vector API. But with so many options, the question is: which ones should we use today, which still matter, and which belong in the history books?

Session #49 : ContextBased Software Engineering by Brian Sletten

There's an implied context to your software running in the world and processing data. The problem is that it is usually a reductive and insufficient context to capture the fluency of change that occurs at multiple layers. This need for shared context spreads to API usage which often necessitates fragile, custom development. In this talk we will address the importance of dynamic context in software systems and how to engender flexible, sufficiently rich context-based systems.

Session #50 : Deep dive into Model Context Protocol MCP by James Ward

The Model Context Protocol (MCP) standardizes how AI agents connect to external data and tools.

Session #51 : A Developers Guide to Building Your First MCP Server by Travis Gosselin

As AI agents become a first-class part of software systems, learning to build MCP servers is becoming as fundamental as learning to build RESTful APIs. Many engineers already interact with MCP through IDEs and agent tooling, and building these servers introduces a new set of design considerations beyond traditional request/response APIs. While MCP servers share familiar API concepts, the move to agent-driven, probabilistic clients changes how engineers think about contracts, tool design, output shaping, state management, error handling, and spec adoption. Building MCP servers is emerging as an important capability for all developers and teams.

Session #52 : Graph Thinking with AI Algorithms that Power Real Systems by Rohit Bhardwaj

Graphs aren’t just academic—they power the backbone of real systems: workflows (Airflow DAGs), build pipelines (Bazel), data processing (Spark DAGs), and microservice dependencies (Jaeger). This session demystifies classic graph algorithms—BFS, DFS, topological sort, shortest paths, and cycle detection—and shows how to connect them to real-world systems.#You’ll also see how AI tools like ChatGPT and graph libraries (Graphviz, NetworkX, D3) can accelerate your workflow: generating adjacency lists, visualizing dependencies, and producing test cases in seconds. You’ll leave with reusable patterns for interviews, architecture reviews, and production systems.

Session #53 : An Anthropologists Guide to Engineering Cultures by Robert Harris

What happens when a self-taught programmer with a background in anthropology finds himself leading engineering teams? In this candid, humorous, and emotionally resonant talk, Robert Harris shares his journey from BASIC on a Commodore 64 to building psychologically safe, high-performing cultures in modern software organizations. Blending fieldwork with frameworks, Robert explores the human side of engineering leadership—imposter syndrome, accidental management, and the painful lessons that shaped his philosophy. Drawing on his training in anthropology, he offers a practical guide to shaping team culture through shared language, rituals, experiences, and artifacts—from flaming pull request beacons to rubber duck onboarding kits. Attendees will leave with: •A fresh perspective on leadership rooted in emotional intelligence and cultural design •Actionable strategies for building trust, accountability, and psychological safety •A toolkit of metaphors, rituals, and artifacts to transform team dynamics

6:30 - 7:15 PM - Sessions

7:15 - 8:00 PM : KEYNOTE

8:00 - 9:30 PM - Sessions

Session #54 : Lost Developer Wisdom by Michael Carducci

In our rush toward the future, the software industry keeps forgetting its past—and with it, the hard-won lessons that could save us from repeating the same mistakes. In this live storytelling session, Michael Carducci revives the forgotten wisdom of the pioneers who shaped our craft.

Session #55 : Penning Powerful Prompts Crafting effective prompts to get the best from an LLM by Craig Walls

In this session, we'll cover several useful prompt engineering techniques as well as some emerging patterns that are categorized within the "Agentic AI" space and see how to go beyond simple Q&A to turn your LLM of choice into a powerful ally in achieving your goals.

Session #56 : Maximize Your Output The 8020 RuleToo Simple to Be True by Hunter Milligan

Ever feel like you're spinning your wheels on tasks that barely move the needle? It's time to flip the script with the 80/20 rule—where 20% of your efforts drive 80% of your results. This talk provides background on how the 80/20 rule transforms productivity at work and

home, helping you identify and focus on the tasks that truly matter while minimizing distractions, low-impact work, and yes—no more chasing squirrels! You'll learn practical strategies to prioritize with pinpoint clarity, eliminate time-wasters, and let go of the stuff that doesn't actually deserve your time or energy. Letting go is a skill—one you'll start applying here to reclaim focus and get results.

Thursday, Jul. 16

8:00 - 9:00 AM : UBERCONF 2026 THURSDAY: BREAKFAST - LEGACY BALLROOM

9:00 - 10:30 AM - Sessions

Session #57 : Six Secrets to Succeeding with Microservices by Michael Carducci

Microservices architecture has become a buzzword in the tech industry, promising unparalleled agility, scalability, and resilience. Yet, according to Gartner, more than 90% of organizations attempting to adopt microservices will fail. How can you ensure you're part of the successful 10%?

Session #58 : The Agentic Architect Session I by Dave Hendricksen

2025 shattered the old cadence of software architecture. AI agents now co#author code and refactors, compliance expectations tightened, and cost/latency signals moved inside everyday design loops. Static diagrams, quarterly review boards, and slide-driven governance can't keep up. This curated set of 3 sessions will help equip senior technologists to evolve from document stewardship to adaptive integrity management—blending human judgment, executable principles, and guided agent assistance.Architecture is shifting from static designs to adaptive, agent-driven execution. Come to the Agentic Architect session if you want to see: - how the role of architecture is evolving in the agentic era - practical tips and trick for how to embrace the new agentic toolset - how to lean into architecture as code - cut decision time from weeks and days to hours - stop redrawing diagrams forever

Session #59 : Architectural Principles Building the Foundations of Design Excellence by Ken Sipe

In the realm of architecture, principles form the bedrock upon which innovative and enduring designs are crafted. This presentation delves into the core architectural principles that guide the creation of structures both functional and aesthetic. Exploring concepts such as balance, proportion, harmony, and sustainability, attendees will gain profound insights into the art and science of architectural design. Through real-world examples and practical applications, this session illuminates the transformative power of adhering to these principles, shaping not only buildings but entire environments. Join us as we unravel the secrets behind architectural mastery and the principles that define architectural brilliance.

Session #60 : Java Testing Lab Test Driven Development with Ai by Daniel Hinojosa

In this half-day workshop, we'll practice Test-Driven Development (TDD) by solving a real problem step by step. You'll learn how to think in tests, write clean code through refactoring, and use your IDE and AI tools effectively. We'll also explore how modern Java features (like lambdas and streams) enhance testability, and discuss what's worth testing — and what's not.

Session #61 : Git features you arent using by Raju Gandhi

Git continues to see improvements daily. However, work (and life) can take over, and we often miss the latest changelog. This means we don't know what changed, and consequently fail to see how we can incorporate those in our usage of Git. In this session we'll take a tour of some features that you might or might not have heard of, but can significantly improve your workflow and day-to-day interaction with Git.

Session #62 : Demystifying Generative AI by Mary Grygleski

Everybody is talking about Generative AI and models that are better than anything else before. What are they really talking about? In this workshop with some hands-on exercise, we will discuss Generative AI in theory and will also try it in practice (with free access to an Oracle LiveLab cloud session to learn about Vector Search). You'll be able to understand what Generative AI is all about and how it can be used.

Session #63 : To Java 25 and Beyond by Billy Korando

Java 25 has been released, but the Java release train coninues chugging along with Java 26.

Session #64 : Building Intelligent Spring Applications with Spring AI by Craig Walls

By now, you've no doubt noticed that Generative AI is making waves across many industries. In between all of the hype and doubt, there are several use cases for Generative AI in many software projects. Whether it be as simple as building a live chat to help your users or using AI to analyze data and provide recommendations, Generative AI is becoming a key piece of software architecture. So how can you implement Generative AI in your projects? Let me introduce you to Spring AI. For over two decades, the Spring Framework and its immense portfolio of projects has been making complex problems easy for Java developers. And now with the new Spring AI project, adding Generative AI to your Spring Boot projects couldn't be easier! Spring AI brings an AI client and templated prompting that handles all of the ceremony necessary to communicate with common AI APIs (such as OpenAI and Azure OpenAI). And with Spring Boot autoconfiguration, you'll be able to get straight to the point of asking questions and getting answers your application needs.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #65 : BehaviorDriven REST An API Case Study by Brian Sletten

A client once asked me to take a team that was new to REST, Agile, etc. and put together a high profile, high value commerce-oriented API in the period of six months. In the process of training the team and designing this API, I hit upon the idea of providing rich testing coverage by mixing the Behavior-Driven Design testing approach with REST.

Session #66 : Master Your Focus and Stay on Track by Hunter Milligan

Ever find yourself slipping into endless distractions and losing hours on what started as a quick task? You're not alone, but it is possible to get much better at avoiding the cycle of going down the rabbit hole.

10:30 - 11:00 AM : UBERCONF 2026 - THURSDAY: AM BREAK

11:00 - 12:30 PM - Sessions

Session #67 : Architecture Blindspots Biases and Blunders Avoiding Common Pitfalls in System Design by Michael Carducci

Architectural decisions are often influenced by blindspots, biases, and unchecked assumptions, which can lead to significant long-term challenges in system design. In this session, we'll explore how these cognitive traps affect decision-making, leading to architectural blunders that could have been avoided with a more critical, holistic approach.

Session #68 : Agentic Architecture in Action Session II by Dave Hendricksen

2025 shattered the old cadence of software architecture. AI agents now co#author code and refactors, compliance expectations tightened, and cost/latency signals moved inside everyday design loops. Static diagrams, quarterly review boards, and slide-driven governance can't keep up. This live demo session takes the patterns from "The Agentic Architect" and runs them end-to-end starting with a blank slate.

Session #69 : Operationalizing AI From Prototype to Production by Ken Sipe

Building an AI model is the easy part—making it work reliably in production is where the real engineering begins. In this fast-paced, experience-driven session, Ken explores the architecture, patterns, and practices behind operationalizing AI at scale. Drawing from real-world lessons and enterprise implementations, Ken will demystify the complex intersection of machine learning, DevOps, and data engineering, showing how modern organizations bring AI from the lab into mission-critical systems.

Session #70 : Java Testing Lab Test Driven Development with Ai (continued) by Daniel Hinojosa

In this half-day workshop, we'll practice Test-Driven Development (TDD) by solving a real problem step by step. You'll learn how to think in tests, write clean code through refactoring, and use your IDE and AI tools effectively. We'll also explore how modern Java features (like lambdas and streams) enhance testability, and discuss what's worth testing — and what's not.

Session #71 : Git features you arent using (continued) by Raju Gandhi

Git continues to see improvements daily. However, work (and life) can take over, and we often miss the latest changelog. This means we don't know what changed, and consequently fail to see how we can incorporate those in our usage of Git. In this session we'll take a tour of some features that you might or might not have heard of, but can significantly improve your workflow and day-to-day interaction with Git.

Session #72 : Demystifying Generative AI (continued) by Mary Grygleski

Everybody is talking about Generative AI and models that are better than anything else before. What are they really talking about? In this workshop with some hands-on exercise, we will discuss Generative AI in theory and will also try it in practice (with free access to an Oracle LiveLab cloud session to learn about Vector Search). You'll be able to understand what Generative AI is all about and how it can be used.

Session #73 : Java is Data by Billy Korando

Data is at the center of any organization. So it stands to reason that data should be at the center of how we design and write our Java applications.

Session #74 : Building Intelligent Spring Applications with Spring AI (continued) by Craig Walls

By now, you've no doubt noticed that Generative AI is making waves across many industries. In between all of the hype and doubt, there are several use cases for Generative AI in many software projects. Whether it be as simple as building a live chat to help your users or using AI to analyze data and provide recommendations, Generative AI is becoming a key piece of software architecture. So how can you implement Generative AI in your projects? Let me introduce you to Spring AI. For over two decades, the Spring Framework and its immense portfolio of projects has been making complex problems easy for Java developers. And now with the new Spring AI project, adding Generative AI to your Spring Boot projects couldn't be easier! Spring AI brings an AI client and templated prompting that handles all of the ceremony necessary to communicate with common AI APIs (such as OpenAI and Azure OpenAI). And with Spring Boot autoconfiguration, you'll be able to get straight to the point of asking questions and getting answers your application needs.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #75 : Automating API Evolution with OpenRewrite by Brian Sletten

One of the nice operational features of the REST architectural style as an approach to API Design is that is allows for separate evolution of the client and server. Depending on the design choices a team makes, however, you may be putting a higher burden on your clients than you intend when you introduce breaking changes. By taking advantage of the capabilities of OpenRewrite, we can start to manage the process of independent evolution while minimizing the impact. Code migration and refactoring can be used to transition existing clients away from older or deprecated APIs and toward new versions with less effort than trying to do it by hand.

Session #76 : Thrive Without the Grind Creating a Frictionless Work Environment by Hunter Milligan

Picture this: another chaotic project push, overloaded schedules, endless revisions, and frayed tempers all in the name of meeting the impossible delivery deadline. The grind feels inevitable, but it doesn't have to be. Let us help you to transform high-pressure environments into spaces where you and your team can thrive. Learn to manage delivery stress, navigate challenging interactions, and lead with clarity.

12:30 - 1:30 PM : UBERCONF 2026 - THURSDAY: LUNCH - LEGACY BALLROOM & SOUTH COURTYARD

1:30 - 3:00 PM - Sessions

Session #77 : Data Architecture for AI by Michael Carducci

Everyone's talking about AI models, but almost no one is talking about the data architecture that makes them intelligent. Today's AI systems are brittle because they lack context, semantics, and shared understanding. In this session, Michael Carducci explores how linked data, RDF, ontologies, and knowledge graphs solve the very problems that leave the industry floundering: hallucination, inconsistency, and lack of interoperability.

Session #78 : Architectural Seismic Shifts Exploration Session III by Dave Hendricksen

2025 shattered the old cadence of software architecture. AI agents now co#author code and refactors, compliance expectations tightened, and cost/latency signals moved inside everyday design loops. Static diagrams, quarterly review boards, and slide-driven governance can't keep up. 2025 delivered unprecedented architectural disruption.

Session #79 : Secure by Design Building ZeroTrust API Architectures in the Cloud by Sumir Arora

API security goes beyond protecting endpoints—it requires defense across infrastructure, data, and business logic. In this talk, I'll present a structured approach to implementing Zero Trust security for APIs in a cloud-native architecture.

Session #80 : Monitoring Systems with Prometheus and Grafana by Daniel Hinojosa

Prometheus and Grafana form the backbone of modern metrics-based observability, yet many teams struggle to move from "we collect metrics" to "we understand our systems." This talk builds a clear mental model for Prometheus and Grafana: how metrics are exposed, scraped, stored, queried, and visualized — and how those metrics connect to real operational decisions. We'll explore Prometheus architecture, PromQL, Kubernetes integration via the Prometheus Operator, and how metrics power advanced workflows like canary deployments with Argo Rollouts and OpenTelemetry-based telemetry. Attendees will leave knowing what to measure, how to measure it, and where to start on Monday.

Session #81 : Building Agents with MCP and Data Streams by Tim Berglund

LLMs are incredibly powerful, but they have two problems: they only know what they read on the Internet, and they can't actually do anything—they can only chat with you. If you want to build agentic applications that have access to the immediate, non-public context of your business, and you want your agents to be able take actions in the world, you'll probably need some help from the Model Context Protocol, or MCP. And that "business context" increasingly exists in the form of real-time streaming data, often in Kafka topics. Once you're asking your microservices to interpret natural-language prompts, then deputizing them to take actions on your behalf—this is what an agent is!—you can't afford for them to be acting on out-of-data context. They need to remain deeply connected to the events that matter to your business.

Session #82 : Enterprise Architecture 40 The AIDriven Future Preview by Rohit Bhardwaj

AI, agentic workflows, digital twins, edge intelligence, spatial computing, and blockchain trust are converging to reshape how enterprises operate. This session introduces Enterprise Architecture 4.0—a practical, future-ready approach where architectures become intelligent, adaptive, and continuously learning.

Session #83 : Concurrency Concepts in Java by Douglas Hawkins

Unlike other languages, Java had a well-defined memory model from the very beginning, but over the years additional packages and low-level features have been added to make the most of today's hardware. In this talk, we'll discuss concurrency in detail starting at the hardware up to Java's latest synchronization mechanisms and finally onto high-level concurrent collections.

Session #84 : Agentic AI Workflows with Embabel by DaShaun Carter

We have all seen the "Hello, World" of Spring AI: sending a prompt and getting a response. But as we move toward production, the real challenge is not the LLM call; it is the workflow. How do you ensure an agent does not loop infinitely? How do you coordinate multiple tools without a mess of "if-else" blocks? And how do we keep our Java-centric domain models at the heart of the AI's reasoning? Enter

Embabel, a new JVM-based framework from Rod Johnson (creator of Spring) designed to bring discipline to agentic AI. Unlike Python-centric alternatives, Embabel is built on the philosophy of strong typing, OODA loops (Observe, Orient, Decide, Act), and Goal-Oriented Action Planning (GOAP).

Session #85 : Crafting Consistent APIs at Scale Embracing Linting Reusable Models by Travis Gosselin

In an era where digital transformation and AI adoption are accelerating across every industry, the need for consistent, scalable, and robust APIs has never been more critical. AI-powered tools—whether generating code, creating documentation, or integrating services—rely heavily on clean, well-structured API specifications to function effectively. As teams grow and the number of APIs multiplies, maintaining design consistency becomes a foundational requirement not just for human developers, but also for enabling reliable, intelligent automation. This session explores how linting and reusable models can help teams meet that challenge at scale.

Session #86 : Tech Trends for Tech Leaders by Brian Sletten

There are certain tech trends people at least know about such as Moore's Law even if they don't really understand them. But there are other forces at play in and around our industry that are unknown or ignored by the ever diminishing tech journalism profession. They help explain and predict the pressures and influences we are seeing now or soon will.

3:00 - 3:15 PM : UBERCONF 2026 - THURSDAY: PM BREAK

3:15 - 4:45 PM - Sessions

Session #87 : A Back to Front Approach to Design Patterns by Simon Roberts

For many beginning and intermediate software engineers, design is something of a secret anxiety. Often we know we can create something that works, and we can likely include a design pattern or two tif only to give our proposal some credibility. But sometimes, we're left with a nagging feeling that there might be a better design, or more appropriate pattern, and we might not be really confident that we can justify our choices.

Session #88 : AI Gateways Unifying Model Access Control by Marcelo Araujo

As AI model usage grows across enterprise systems, teams face new infrastructure challenges—fragmented integrations, inconsistent interfaces, and limited visibility into model performance. An AI Gateway bridges this gap by providing an abstraction layer for model routing, guardrails, and observability, standardizing how applications interact with AI models.

Session #89 : Architecting Scalable Clouds Integrating AI by Sumir Arora

As cloud architectures evolve, AI is quickly becoming a foundational component rather than an add-on.

Session #90 : Identity Tokens and Access Control with Keycloak by Daniel Hinojosa

Authentication and authorization are foundational concerns in modern systems, yet they’re often treated as afterthoughts or re-implemented inconsistently across services. In this talk, we’ll explore Keycloak, an open-source identity and access management system, and how it fits into modern application architectures. We’ll break down what Keycloak actually does (and what it doesn’t), explain the role of JWTs and OAuth2/OpenID Connect, and examine how identity, trust, and access control are handled across distributed systems. We’ll also compare Keycloak to secret management systems like Vault, clarify common misconceptions, and walk through integrations you will need with Spring, Quarkus, and other frameworks By the end, you’ll understand when Keycloak is the right tool, how to integrate it cleanly, and how to avoid the most common architectural mistakes.

Session #91 : Prompt Engineering Accelerator for AI by Brent Laster

In this intensive 3-hour hands-on workshop, you'll learn to master the art and science of prompt engineering. Learn systematic frameworks for constructing effective prompts, from foundational elements to cutting-edge techniques including multi-expert prompting, probability-based optimization, and incentive framing. Through five progressive labs using Ollama and llama3.2:3b in GitHub Codespaces, you'll build production-ready templates and see quality improvements in real-time. Leave with immediately applicable techniques, reusable prompt patterns, and a decision framework for selecting the right approach for any AI task.

Session #92 : Architecting Microservices for Agentic AI Integration by Rohit Bhardwaj

Autonomous LLM agents don’t just call APIs — they plan, retry, chain, and orchestrate across multiple services.#That fundamentally changes how we architect microservices, define boundaries, and operate distributed systems. This session delivers a practical architecture playbook for Agentic AI integration — showing how to evolve from simple request/response designs to resilient, event-driven systems.#You’ll learn how to handle retry storms, contain failures with circuit breakers and bulkheads, implement sagas and outbox patterns for correctness, and version APIs safely for long-lived agents. You’ll leave with reference patterns, guardrails, and operational KPIs to integrate agents confidently—without breaking production systems.

Session #93 : Understanding Java Garbage Collectors by Douglas Hawkins

Most of us don't want to go back to the days of malloc and free, but the magic of garbage collectors while convenient can be mysterious and hard to understand. In this talk, you'll learn about the many different garbage collectors available in JVMs. The strength and weaknesses of the different allocation and collection strategies used by each collector. And how garbage collectors keep evolving to support today's hardware and cloud environments.

ÜberConf

-Session Schedule-

(event schedule as of February 19, 2026)

Session #94 : Engineering Autonomous Agents with Embabel and Spring by DaShaun Carter

The "Hello, World" of Spring AI involves sending a prompt and receiving a text response. This is no longer enough for production. To build enterprise grade AI, we must move beyond simple request and response cycles toward autonomous agents capable of reasoning, planning, and executing complex workflows. The challenge is doing this without losing the type safety, observability, and domain driven design that makes the Java ecosystem the backbone of enterprise software.

Session #95 : Engineering the Context That Powers Developers and Agents by Travis Gosselin

Engineering teams have always struggled with fragmented documentation, tribal knowledge, and inconsistent standards. What we now call context engineering is closely related to long-standing knowledge management challenges, but the impact is amplified as AI agents enter everyday development workflows. Gaps, conflicts, and poorly organized context slow developers and lead automated systems to produce unreliable results. Without deliberate strategies, organizations risk scaling confusion instead of capability.

Session #96 : Empowering Generalists by Brian Sletten

Since the Scientific and Industrial Revolutions, there has been more to know every day. No individual can know it all and we have seen the entrenchment of the specialist for the past hundred or so years. When all of this tacit knowledge was locked in our heads, the specialist was rewarded for knowing details. In our industry we have seen professionals gravitate to specific languages, specific tiers in the architecture (e.g. front-end vs backend), and specific libraries or frameworks. Sometimes they will even go so far as to list specific versions of specific technologies on their resume. All of this specialization can be beneficial when you need resources that are deep within narrow confines. The ubiquitous glut of available information no longer requires us to know topics to this level of detail. Market realities are also such that nobody has the budget to employ only specialists any more. Developers have needed to learn to become designers, testers, data-experts, security-aware, AI-cognizant, and capable of communicating with various stakeholders. When your industry epitomizes unfettered change, you need to rely on generalists, not specialists; synthesizers, not knowledge keepers. How can you attract, hire, and benefit from technologists who identify as problem solving value adders rather than programmers of a specific language? How can you encourage their growth and measure success? Even more, how do you lead them yourself?

4:45 - 5:00 PM : UBERCONF 2026 - THURSDAY: PM BREAK

5:00 - 6:30 PM - Sessions

Session #97 : A Back to Front Approach to Design Patterns (continued) by Simon Roberts

For many beginning and intermediate software engineers, design is something of a secret anxiety. Often we know we can create something that works, and we can likely include a design pattern or two tif only to give our proposal some credibility. But sometimes, we're left with a nagging feeling that there might be a better design, or more appropriate pattern, and we might not be really confident that we can justify our choices.

Session #98 : Architecting a Secure Scalable and Cost Effective Health system by Javiera Laso

In this architectural kata, you will step into the shoes of a software architect tasked with designing a modern healthcare management system for a rapidly growing provider, MedBest.

Session #99 : 12 Essential Technical Skills for Software Architects by Dave Hendricksen

This session will focus on the essential technical skills that are needed by software architects on a daily basis from ideation to product delivery. For many architects, maintaining your technical skills can be a challenge.

Session #100 : DataMesh and Open Metadata by Daniel Hinojosa

Data Mesh rethinks data architecture in organizations by treating data as a product, owned and operated by bounded context teams rather than centralized platforms. This way, data owners can describe, enrich, and prove data sources to prevent any malicious poisoning.

Session #101 : Prompt Engineering Accelerator for AI (continued) by Brent Laster

In this intensive 3-hour hands-on workshop, you'll learn to master the art and science of prompt engineering. Learn systematic frameworks for constructing effective prompts, from foundational elements to cutting-edge techniques including multi-expert prompting, probability-based optimization, and incentive framing. Through five progressive labs using Ollama and llama3.2:3b in GitHub Codespaces, you'll build production-ready templates and see quality improvements in real-time. Leave with immediately applicable techniques, reusable prompt patterns, and a decision framework for selecting the right approach for any AI task.

Session #102 : Securing LLMs DevSecOps in the Age of AI by Rohit Bhardwaj

As enterprises rush to embed large language models (LLMs) into apps and platforms, a new AI-specific attack surface has emerged. Prompt injections, model hijacking, vector database poisoning, and jailbreak exploits aren’t covered by traditional DevSecOps playbooks. This full-day, hands-on workshop gives architects, platform engineers, and security leaders the blueprint to secure AI-powered applications end-to-end. You’ll master the OWASP LLM Top 10, integrate AI-specific controls into CI/CD pipelines, and run live red-team vs blue-team exercises to build real defensive muscle. Bottom line: if your job involves deploying, securing, or governing AI systems, this workshop shows you how to do it safely—before attackers do it for you.

Session #103 : JustinTime Compilation Isnt Magic by Douglas Hawkins

Fortunately for most Java developers the just-in-time compiler just works and appears to do so by magic. And yet sometimes, we find ourselves facing a performance problem, so what do we when the magic stops? In this talk, we'll learn a few key concepts behind the magic of modern optimizing compilers: intrinsics, basic blocks, static single assignment, and inlining. By learning these key concepts, you'll learn to save time not trying to optimize the things that the compiler can already do for you and to focus on the things that matter most.

Session #104 : Engineering Autonomous Agents with Embabel and Spring (continued) by DaShaun Carter

The "Hello, World" of Spring AI involves sending a prompt and receiving a text response. This is no longer enough for production. To build enterprise grade AI, we must move beyond simple request and response cycles toward autonomous agents capable of reasoning, planning, and executing complex workflows. The challenge is doing this without losing the type safety, observability, and domain driven design that makes the Java ecosystem the backbone of enterprise software.

Session #105 : AI Enablement in the SDLC A Strategy Sampler by Travis Gosselin

AI enablement isn't buying Copilot and calling it done; it's a system upgrade for the entire SDLC. Code completion helps, but the real bottlenecks live in reviews, testing, releases, documentation, governance, and knowledge flow. Achieving meaningful impact requires an operating model: guardrails, workflows, metrics, and change management; not a single tool.

Session #106 : Managing Tech Teams Rules Tools Insights by Michael Carducci

Software projects can be difficult to manage. Managing teams of developers can be even difficult. We've created countless processes, methodologies, and practices but the underlying problems remain the same.

6:30 - 7:30 PM : UBERCONF 2026 THURSDAY: DINNER - LEGACY BALLROOM

7:30 - 9:00 PM : UBERCONF 2026: THURSDAY - EXPERT PANEL DISCUSSION - LEGACY BALLROOM

Friday, Jul. 17

7:30 - 8:30 AM : UBERCONF 2026: BREAKFAST - LEGACY BALLROOM

8:30 - 10:00 AM - Sessions

Session #107 : Navigating the Cloud as a Cloud Architect by Ken Sipe

In the age of digital transformation, Cloud Architects emerge as architects of the virtual realm, bridging innovation with infrastructure. This presentation offers a comprehensive exploration of the Cloud Architect's pivotal role.

Session #108 : Improving API Design with the help of LLMs by Marcelo Araujo

Traditional API linting tools like Spectral, have helped teams identify issues in their OpenAPI specifications by surfacing violations of style guides and best practices. But the current paradigm stops at diagnosis—developers are still left with the manual burden of interpreting warnings, resolving inconsistencies, and applying often repetitive best practice fixes.

Session #109 : Agentic Assistive Predictive AI Design Patterns by Rohit Bhardwaj

Building AI isn't just about prompting or plugging into an API — it's about architecture. This workshop translates Salesforce's Enterprise Agentic Architecture blueprint into practical design patterns for real-world builders. You'll explore how Predictive, Assistive, and Agentic patterns map to Salesforce's Agentforce maturity model, combining orchestration, context, and trust into cohesive systems. Through hands-on modules, participants design a Smart Checkout Helper using Agentforce, Data Cloud, MCP, and RAG—complete with observability, governance, and ROI mapping.

Session #110 : Platform Engineering The What Why How by Raju Gandhi

Platform engineering is the latest buzzword, in a industry that already has it's fair share. But what is platform engineering? How does it fit in with DevOps and Developer Experience (DevEx)? And is this something your organization even needs?

Session #111 : Red Green Collaborate TestDriven Development with AI by Javiera Laso

In this immersive, hands-on workshop, participants will learn how to combine the discipline of Test-Driven Development with the creative support of AI-powered pair programming. Working in pairs, developers will build a Booking system from scratch using Java and VS Code, progressively applying the red-green-refactor cycle while integrating AI assistance for test authoring and design validation.

Session #112 : Inside MCP Live Protocol Messages RealTime Flows and Smarter Agents by David Parry

You've heard the buzz — now roll up your sleeves and build with it. In this hands-on workshop, you'll learn exactly how the Model Context Protocol (MCP) works — and you'll write your own MCP server tool from scratch, then author an Agent that uses it to deliver

real-time, context-aware help right inside your dev flow. We'll break down the raw MCP protocol step by step: • How it streams context between your IDE and Agents • How messages are structured and exchanged • How to wire up an MCP Client to talk to your new tool

Session #113 : Refactoring REST APIs for the Last Time Strategies for a FutureProof Design by Michael Carducci

REST APIs often fall into a cycle of constant refactoring and rewrites, leading to wasted time, technical debt, and endless rework. This is especially difficult when you don't control the API clients. But what if this could be your last major API refactor? In this session, we'll dive into strategies for designing and refactoring REST APIs with long-term sustainability in mind—ensuring that your next refactor sets you up for the future.

Session #114 : Rust Making Software Fast and Safe by Brian Sletten

New languages often carry an operational burden to deployment and involve tradeoffs of performance for safety. Rust has emerged as a powerful, popular, and increasingly widely-used language for all types of development. Come learn why Rust is entering the Linux kernel and Microsoft and Google are favoring it for new development over C++. This Introduction to Rust will introduce the students to the various merits (and complexities) of this safe, fast and popular new programming language that is taking the world by storm. This three day course will cover everything students from various backgrounds will need to get started as a successful Rust programmer.

Session #115 : Java Generics Clarified by Simon Roberts

Java's Generics syntax provides us with a means to increase the reusability of our code by allowing us to build software, particularly library software, that can work on many different types, even with limited knowledge about those types. The most familiar examples are the classes in Java's core collections API which can store and retrieve data of arbitrary types, without degenerating those types to java.lang.Object. However, while the generics mechanism is very simple to use in simple cases such as using the collections API, it's much more powerful than that. Frankly, it can also be a little puzzling.

Session #116 : Decision Making in Software Teams by Tim Berglund

Alistair Cockburn has described software development as a game in which we choose among three moves: invent, decide, and communicate. Most of our time at No Fluff is spent learning how to be better at inventing. Beyond that, we understand the importance of good communication, and take steps to improve in that capacity. Rarely, however, do we acknowledge the role of decision making in the life of software teams, what can cause it to go wrong, and how to improve it.

10:00 - 10:30 AM : UBERCONF 2026 - FRIDAY: BREAK

10:30 - 12:00 PM - Sessions

Session #117 : From Cloud Complexity to DevOps Simplicity by Sumir Arora

Here I'll break down how GitOps simplifies the operational challenges around cloud and Kubernetes environments. We'll look at how a Git-driven model brings consistency, automation, and better visibility across both infrastructure and application delivery.

Session #118 : Accelerating the API Design Lifecycle with Mocking by Marcelo Araujo

In today's fast-paced development environment, delivering robust and efficient APIs requires a streamlined design process that minimizes delays and maximizes collaboration. Mocking has emerged as a transformative tool in the API design lifecycle, enabling teams to prototype, test, and iterate at unprecedented speeds.

Session #119 : Agentic Assistive Predictive AI Design Patterns (continued) by Rohit Bhardwaj

Building AI isn't just about prompting or plugging into an API — it's about architecture. This workshop translates Salesforce's Enterprise Agentic Architecture blueprint into practical design patterns for real-world builders. You'll explore how Predictive, Assistive, and Agentic patterns map to Salesforce's Agentforce maturity model, combining orchestration, context, and trust into cohesive systems. Through hands-on modules, participants design a Smart Checkout Helper using Agentforce, Data Cloud, MCP, and RAG—complete with observability, governance, and ROI mapping.

Session #120 : Platform Engineering The What Why How (continued) by Raju Gandhi

Platform engineering is the latest buzzword, in a industry that already has it's fair share. But what is platform engineering? How does it fit in with DevOps and Developer Experience (DevEx)? And is this something your organization even needs?

Session #121 : Red Green Collaborate TestDriven Development with AI (continued) by Javiera Laso

In this immersive, hands-on workshop, participants will learn how to combine the discipline of Test-Driven Development with the creative support of AI-powered pair programming. Working in pairs, developers will build a Booking system from scratch using Java and VS Code, progressively applying the red-green-refactor cycle while integrating AI assistance for test authoring and design validation.

Session #122 : Inside MCP Live Protocol Messages RealTime Flows and Smarter Agents (continued) by David Parry

You've heard the buzz — now roll up your sleeves and build with it. In this hands-on workshop, you'll learn exactly how the Model Context Protocol (MCP) works — and you'll write your own MCP server tool from scratch, then author an Agent that uses it to deliver

real-time, context-aware help right inside your dev flow. We'll break down the raw MCP protocol step by step: • How it streams context between your IDE and Agents • How messages are structured and exchanged • How to wire up an MCP Client to talk to your new tool

Session #123 : Innovation Why the Majority Is Always Wrong by Michael Carducci

If everyone agrees with you, you're probably not innovating, you're just conforming faster. History's breakthroughs rarely came from consensus; they came from heretics, hackers, and the hopelessly curious. In this talk, Michael Carducci takes aim at the myth of collective wisdom and explores why the crowd is almost always optimized for the past. Through stories of misfits who changed the world—from computing pioneers to magicians who reinvented wonder; Carducci reveals the hidden patterns of real innovation: discomfort, doubt, and persistence in the face of polite disbelief.

Session #124 : Rust Making Software Fast and Safe (continued) by Brian Sletten

New languages often carry an operational burden to deployment and involve tradeoffs of performance for safety. Rust has emerged as a powerful, popular, and increasingly widely-used language for all types of development. Come learn why Rust is entering the Linux kernel and Microsoft and Google are favoring it for new development over C++. This Introduction to Rust will introduce the students to the various merits (and complexities) of this safe, fast and popular new programming language that is taking the world by storm. This three day course will cover everything students from various backgrounds will need to get started as a successful Rust programmer.

Session #125 : Java Generics Clarified (continued) by Simon Roberts

Java's Generics syntax provides us with a means to increase the reusability of our code by allowing us to build software, particularly library software, that can work on many different types, even with limited knowledge about those types. The most familiar examples are the classes in Java's core collections API which can store and retrieve data of arbitrary types, without degenerating those types to java.lang.Object. However, while the generics mechanism is very simple to use in simple cases such as using the collections API, it's much more powerful than that. Frankly, it can also be a little puzzling.

Session #126 : Mastering Tech Leadership in 60 Minutes by Tim Berglund

If only it were so easy! Leadership is a thing into which many find themselves thrown, and to which many others aspire—and it is a thing which every human system needs to thrive. Leading teams in technology organizations is not radically different from any other kind of organization, but does tend to present a common set of patterns and challenges. In this session, I'll examine them, and provide a template for your own growth as a leader.

12:00 - 12:30 PM : UBERCONF 2026: LUNCH - LEGACY BALLROOM

12:30 - 1:30 PM : CLOSING KEYNOTE