Mosaic

Call for Proposals

Abstract Submission Guide

9 May 2026 | Bangalore · Hybrid | ~750 Practitioners

Five ways to share your story with the community.

The Technology Tribe · thetechnologytribe.com

01

SPECIFICITY

Lightning Talks

15 minutes · A single problem, a working solution, and what really made it click

Guidance for speakers

• Choose one idea, not a landscape • Keep the abstract to one paragraph • Promise 1-2 learnings only •

Favor sharpness over breadth. Your talk should leave the audience with one impactful, usable idea

ABSTRACT TEMPLATE

(a) The hook line

Sample structure: Today, teams can [move faster / automate more / ship more], but [specific thing] still breaks when [real-world condition]. In 15 minutes, this talk shows [one sharp insight or practical fix]

(b) What problem are you solving?

Sample structure: In [specific system / team / context], [problem] becomes hard because [root cause / complexity]. Most teams try [approach 1] or [approach 2], but these often fail because [practical shortcoming]

(c) Core thesis / Talk Idea?

Sample structure: This talk makes one clear argument: [core idea / pattern / lesson]. Instead of [common but weaker approach], teams should focus on [context / better relevance / mindset]

(d) Why is this talk grounded?

The idea is drawn from [real system / migration / incident / production lesson / field experience], where [brief evidence of scale, complexity, or credibility]

(e) What attendees will learn / Talk takeaways?

Sample structure: By the end of this talk, attendees will have: one practical mental model, one pattern or tactic they can apply immediately, and one pitfall to avoid

WORKED EXAMPLE

Title:

Why retries make outages worse

Abstract:

Abstract: Modern distributed systems make retries feel like a safe default, but under real failure conditions they can amplify load and turn a local slowdown into a wider incident. This lightning talk shows why retries often fail when treated as a generic reliability mechanism, and why teams should instead think in terms of retry budgets, idempotency, backoff, and overload-aware design. Attendees will leave with one practical framework for deciding when retries help, when they harm, and one set of failure modes to watch for before shipping them into critical paths.

02

GENERALISATIONS

Deep Dives

30–45 minutes · Solving large or ambiguous problems — alternative designs, trade-offs, and learnings

ABSTRACT TEMPLATE

(a) The hook line

Sample structure: Today, teams can [do what faster/better than before], but [what still breaks/fails] when [real-world condition]. The result is [impact on systems / teams / customers / business]

(b) What problem are you solving?

Sample structure: In [type of systems / teams / environment], [specific problem] becomes difficult because [root cause / constraint / complexity]. Existing approaches such as [approach 1] or [approach 2] often fall short because [why they fail in practice]

(c) Core thesis / Talk Idea?

Sample structure: This talk presents a practical approach to [problem area] by focusing on [key idea / shift in thinking / method] instead of [common but weaker approach]

(d) Why is this talk grounded?

The ideas are drawn from [real system / migration / incidents / production environment / field experience], where [brief evidence of scale, complexity, or credibility]

(e) What attendees will learn / Talk takeaways?

Sample structure: In this talk, you'll learn: - how to [design / implement / scale / evaluate something] - how to avoid [common mistake / anti-pattern / failure mode] - how to measure or decide [success metric / trade-off / readiness / quality]

WORKED EXAMPLE

Title:

Building Observability for 99% Developers

Abstract:

Modern SaaS and API-driven development have made it dramatically easier for small teams to ship software quickly - but not to operate it with confidence. As systems become more distributed and releases more frequent, observability often remains too manual, too expert-dependent, and too costly for the majority of engineering teams that do not have dedicated platforms or DevOps specialists. In this talk, I'll explore what observability should look like for the "99% developer" team and how techniques drawn from compilers and program analysis can help build more automated, "one-click" visibility into application behavior. Attendees will learn how to think about observability as a developer experience problem, where automation can meaningfully reduce instrumentation and debugging effort, and what trade-offs matter when designing observability for lean teams. You'll leave with a practical mental model for right-sizing observability, a framework for reducing expert dependency, and a clearer view of where automated program analysis can change the game.

03

SHOWCASES

OSS Showcases & Demos

15–30 minutes · Open-source-ready solutions — inspire collaboration and contribution

In 15–20 minutes, the goal is to demonstrate one meaningful open-source project, the real technical problem it solves, and how the system works in practice.

The idea is to describe the wider applicability of the problem statement and the approach towards the solution. The audience, if they find it to be relevant, can explore the open source to potentially collaborate on the same and/or look to adopt the solution as an open source in the scope of their work.

ABSTRACT TEMPLATE

(a) The hook line

Sample structure: Today, teams building [type of systems / applications] often struggle with [a specific problem], even though [existing tools, frameworks, or approaches] exist. This open-source project explores a better way to handle this challenge.

(b) What problem does this open source solve?

Sample structure: We are building [project name], an open-source project designed to help [developers / platform teams / operators / researchers] solve [a specific technical problem]. Today, teams often rely on [workarounds / fragmented tools / manual effort / proprietary systems], which makes [desired capability] difficult or inefficient to achieve.

(c) What is the real problem and why is it hard?

Sample structure: This problem persists because [technical constraint / ecosystem gap / scaling challenge / developer workflow friction]. Common approaches such as [approach 1] or [approach 2] often fall short due to [limitation or trade-off].

(c) Why do you think this problem statement matters in a wider context?

Sample structure: The project is already showing signals such as: community adoption or contributors | production or pilot usage | performance or operational improvements. These suggest the project addresses a real need in the developer ecosystem.

(d) What will you showcase in the demo?

Sample structure: In this showcase, we will demonstrate [a concrete developer or system workflow], showing: [how the project is integrated or used | how it solves the target problem | what technical behavior or outcome becomes visible]. The demo will walk through [specific scenario or system path] so the audience can see the project operating in a realistic context.

(e) Why should this matter to a premium tech audience?

Sample structure: This project highlights a useful engineering pattern or technical approach relevant to areas such as [platform engineering / AI tooling / distributed systems / developer productivity / infrastructure]

WORKED EXAMPLE

Title:

Open-source agent testing harness for AI applications

Abstract:

Today, teams building AI agents can prototype quickly, but testing them reliably becomes difficult once workflows involve multiple tools, prompts, and probabilistic outputs. We are building an open-source testing harness that allows developers to define structured evaluation scenarios for agent workflows. Instead of relying on fragile scripts or manual checks, the framework provides reproducible test cases, trace capture, tool simulation, and scoring so teams can detect regressions and measure improvements. In this demo, we will show how a developer defines agent test scenarios, runs them against an agent pipeline, and interprets the resulting evaluation traces. Attendees will leave with a concrete understanding of agent testing challenges and a practical framework they can adopt or contribute to.

04

INSPIRE

Tech for Good Demos

15–30 minutes · Impact projects at ground level — inspire the community to contribute

A Tech for Good lightning showcase is not a generic product pitch, a broad innovation story, or a feel-good impact narrative. In 15-20 minutes, the goal is to show one meaningful tech-led project that serves a wider public need, the real problem it addresses, why the approach is materially enabled by technology, and what evidence the audience can see of practical impact, reach, or scalability. The strongest submissions make clear who benefits, what is hard in the real world, what is genuinely novel or thoughtfully engineered, and what the demo will prove within the time slot.

ABSTRACT TEMPLATE

(a) What common good problem are you addressing, and for whom?

Sample structure: We are building [project / platform / system / service] for [community / citizens / workers / patients / students / public systems / underserved groups], helping address [specific problem] in a way that is more accessible, effective, scalable, or inclusive than current alternatives

(b) Why does this problem remain hard in the real world?

Sample structure: In [sector / workflow / environment], this problem remains difficult because [constraint / affordability / trust gap / fragmented systems / data challenge / operational complexity / policy friction]. Existing approaches such as [approach 1], [approach 2], or manual workflows often fall short because [practical limitation]

(c) What is genuinely tech-led about the solution?

Sample structure: The core leverage comes from [AI / data systems / sensing / robotics / workflow orchestration / simulation / platform architecture / security / distributed systems / scientific method], which enables [capability] that would otherwise be hard to achieve through conventional delivery models alone

(d) What exactly will you show in the demo?

In this showcase, we will demonstrate [specific workflow / live product path / field scenario / user journey], so the audience can clearly see [what the system does], [what is technically differentiated], and [what proof of impact, usability, or scale is visible within the demo]

(e) What evidence suggests this can create real world impact?

Sample structure: The project is grounded in [pilot / deployment / field study / user testing / partner ecosystem / operational data / community usage], showing [signal of adoption / efficiency gain / access improvement / quality improvement / cost reduction / public-good relevance]

WORKED EXAMPLE

Title:

Creating early-risk maternal health outreach with multilingual AI triage

Abstract:

In many regions, maternal health risk is not only a clinical problem but an access and coordination problem: symptoms are reported late, frontline workers are overburdened, and many women still struggle to navigate care in their own language and context. We are building a tech-led outreach and triage system that helps community health workers and expectant mothers surface early warning signals through multilingual voice and chat interactions, structured risk workflows, and guided escalation paths. Rather than relying only on static forms or fragmented helplines, the system combines language AI, protocol-driven triage, and case-tracking infrastructure to make earlier intervention more feasible in low-resource settings. In this lightning showcase, we will demonstrate how a frontline worker or expectant mother can report symptoms, how the system classifies risk and recommends next actions, and how supervisors can track follow-up across cases. The audience will leave with a concrete view of the public-good problem, the technical leverage behind the solution, and the practical challenges of building responsible AI systems for real-world health access.

05

INSPIRE

Startup Tech Demos

15–30 minutes · Tech-led startups demonstrating products or ideas grounded in depth

A startup lightning showcase is not a general talk, a sales pitch, or a broad market overview. In 15-20 minutes, the goal is to show one meaningful product idea under development, the real problem it solves, and why the leverage comes from genuine technology depth rather than surface level solution. The strongest submissions make clear what is being built, what is technically differentiated, what the live demo will prove, and why the product matters now.

ABSTRACT TEMPLATE

(a) The hook line

Sample structure: We are building [product / platform / system] for [user / customer / industry], helping them solve [specific problem] in a way that is materially better than [current alternative]

(b) What is the real problem and why is it hard?

Sample structure: In [market / workflow / operating context], [problem] remains difficult because [constraint / complexity / industry friction]. Existing approaches such as [approach 1], [approach 2], or manual workarounds fall short because [practical limitation]

(c) What is genuinely tech-led or deep-tech about this product?

Sample structure: The core leverage comes from [model / algorithm / systems design / robotics / computer vision / simulation / data architecture / AI workflow], which enables [capability] that would otherwise be hard to achieve through conventional software alone

(d) What will you showcase in the demo?

In this showcase, we will demonstrate [specific workflow / live product path / technical capability], so the audience can clearly see [what the product does], [what is novel], and [what proof of value or technical depth is visible within the demo]

(e) Why should this matter to a premium tech audience?

Sample structure: This matters because it reveals [new capability / new market unlock / new product possibility / new engineering pattern] made possible by emerging technology, with implications for [industry / builders / future product design]

WORKED EXAMPLE

Title:

AutoSRE: An AI-Enabled Agentic Reliability Co-Pilot for Modern Engineering Teams

Abstract:

Modern teams can ship software faster than ever, but production reliability still depends heavily on experienced humans stitching together alerts, logs, dashboards, deploy context, and runbooks under pressure. As systems become more distributed, the real challenge is no longer collecting signals, but turning noisy telemetry into safe, actionable operational decisions. In this showcase, we will demonstrate AutoSRE, an AI-enabled agentic SRE platform designed to assist teams with incident triage, probable-cause analysis, blast-radius assessment, and guided response. By combining observability data, service topology, deploy history, historical incidents, and operational guardrails into a contextual reasoning layer, AutoSRE helps teams move from fragmented signals to faster, more confident action. The demo will show how the system handles a realistic production incident: correlating alerts, narrowing likely causes, surfacing relevant runbooks and past incidents, and recommending bounded next steps with human-in-the-loop safety. Attendees will get a practical view of how agentic AI can reduce operational toil, shorten triage time, and make modern reliability workflows more accessible beyond specialist SRE teams.