Compile-Time Contracts & Fiber-Safe Data Pipelines: Scala’s Effect System in Action

---

Back to original talk
Edit

Submitted to

ScalaIO 2025

June 26, 2025 11:50 IST
submitted

Profile:
Vitthal Mirji

View Event

Talk Format

Talk (~30-45 minutes)

Audience Level

All

Tags

Scala, DataEngineering, DataPipelines, ZIO, CatsEffect, Refined, Akka, TypeSafety, Streaming, FunctionalProgramming, MetadataValidation, ContractDriven, BigData, Spark, Flink, Kafka, EffectSystems

Created on

June 26, 2025 11:50 IST

Updated on

June 26, 2025 11:51 IST

Elevator Pitch

Writing data pipelines? Say goodbye to config/metadata chaos! Let Scala's type & effect systems co-author them with you. We'll show how we built a contract-first, pluggable platform using Giter8, Refined, Cats Effect, ZIO, Kyo & Caprese - delivering compile-time guarantees and fiber-safe execution.

Description

What if your data platform stopped relying on configuration-driven / metadata-driven and postmortems - and instead enforced correctness, traceability, and effect boundaries at compile time?

In this talk, we'll walk through how we built a production-ready data pipeline archetype system in Scala. We used Giter8 templates to scaffold pipelines that are contract-driven, type-safe, effectful, and pluggable - all enforced through the Scala type system and effect libraries.

🔧 You'll see how: • 📦 Giter8 templates bootstrap consistent, compile-time safe projects • 🛡️ Refined types validate configuration before runtime • ✅ Cats ValidatedNel catches multi-rule violations in DQ checks • 🔌 Type classes enable pluggable validation and ingestion • ⚙️ ZIO Layers and Cats Effect offer fiber-safe orchestration • 🧠 Kyo and Caprese add effect composition and static guarantees • 🚀 Trait-based runners switch between Spark, Flink, and Kafka • 📊 Data Quality and custom rules enforce data contract quality at runtime

🧪 We'll demo a real pipeline: scaffolded from template, validated with contracts, and executed via a modular runtime - showing how functional design meets engineering scale.

Whether you're a platform engineer, data architect, or Scala practitioner, this talk showcases how to build systems that scale with correctness, traceability, and developer delight.

Notes

1. The speaker has over a 12 years of experience building distributed data platforms at scale.
2. This talk blends real production use-cases with advanced Scala features like type classes, refined types, ZIO layers, and effect capture checking.
3. Demos will include real code, showing Giter8 usage, validation DSLs, and runtime enforcement.
4. We do not require any special setup beyond internet and live code presentation support.
5. This topic appeals to both Scala infrastructure builders and applied functional programming communities.
6. The content also promotes Scala ecosystem tools (ZIO, Cats Effect, Caprese, etc.) with real-world impact.

Submitted Co-Presenters

Additional Information

Organizers Request

We select talks on a rolling basis (rather than doing the selection in bulk after closing the CFP), so increase your chances of getting selected by submitting early!

Your Response

Thank you for the rolling submission process — much appreciated! This talk is fully ready with a working demo, project scaffolding tools, and real-world examples built from experience leading data platform initiatives at scale. I'm happy to iterate with organizers if needed and excited to share this with the Scala community. Looking forward to hearing from you!

Feedback About Submission

Valentin Kasas on July 14, 2025 15:23 IST

Hi Vitthal,

Thank you a lot for your submission, we appreciate it a lot. In fact we are keen to accept it, but before doing so, we need to warn you that considering how tight our budget is this year, we are not sure at all we could cover your travel expenses.

If that isn't a problem for you please let me know and we'll officially accept your talk.

Cheers, Valentin

You on July 14, 2025 18:03 IST

Hi Valentin,

Thanks so much for the kind note - I'm genuinely excited about the opportunity to speak at ScalaIO and contribute to the community.

I had submitted the same talk to Scala Days 2025 Lausanne earlier this year, but just missed the deadline - so I'm especially glad it's being considered now in ScalaIO.