scalac.io
best scala hAkkers!

indeni

Indeni Case Study

- **Category**: Back-end solution
- **Industry**: Automation platform for downtime sensitive industries including Financial Services and Healthcare
- **Client website**: indeni.com

Challenge

With a significant number of devices under constant analysis to intelligently find out if something is wrong or needs handling, **Indeni needed a way to ingest a high rate of metrics into its system for storage and display**.

That's why the company decided to **design a new, custom storage system**, which is tailored to assure high availability, responsiveness, and throughput. **A real Reactive System**. It was meant to **add scale, recovery and data persistency**.

Indeni wanted to utilise the potential of the Scala platform and tooling to create such a system, and they requested Scalac's expertise to help them achieve this goal.

Solution

Scalac supported Indeni by:

- designing and development;
- evaluating strategies and technical solutions to **assure** the system fulfils the required expectations;

- creating a “brainstorming group” as a support that advises on the system’s architecture and highlights best-practices and pitfalls;

The system is built upon the following main technologies, all integrated into a cohesive solution

- akka
- akka-streams
- akka-http
- apache cassandra
- apache kafka
- elasticsearch

This stack choice provides a fault-tolerant and distributed solution with best-of-breed components.

Results

We worked daily and remotely with the client in a team of 2 (**a senior developer from Scalac** and Indeni’s tech lead) and coordinated using a lightweight approach of daily updates and constant interaction through a chat application and the occasional call and screen-sharing.

To achieve the goals **we set in the project were included** :

- Scalac’s Happiness Maker who could smooth out all aspects of the collaboration
- Time-limited development support from another senior scala expert
- One-shot in-depth brainstorming and design review from an expert pool made of 3 Scalac seniors (**the project developer was included**)

The milestones reached in the project:

- **Converting** the initial solution from akka actors to a fully streaming data system based on akka-streams.
- **Reaching** the status of feature-completeness regarding the specification.
- **Increasing** as much as possible the test coverage of the whole codebase.
- **Simplifying** the system to be modular and remove bottlenecks for an increase in performance.
- **Applying** well-known patterns and custom designs to make the system resilient to failure from ground-up.
- Fine-tune the configuration of all the moving pieces to reach the optimal operational level.