Openkoda

Insurance Core Platforms: Accelerating Custom Insurance Innovation

✓ No Vendor Lock-in   ✓ Unlimited Users   ✓ Unlimited Customizability

Contents

Authors

Arkadiusz Drysch
CEO at Openkoda

in Connect

Michał Głomba
Openkoda Founder

in Connect

Arkadiusz Krysik
Marketing Manager at Openkoda

The Need for Modern Insurance Core Platforms

Insurance companies today face intense pressure to innovate and launch new products faster than ever. Many still rely on aging legacy systems – in fact, over **70% of insurers worldwide use some legacy infrastructure.**

These outdated policies and claims systems are rigid, costly to change, and slow down product development. Modern insurance core platforms have emerged as the solution to this problem, acting as the digital backbone that streamlines day-to-day operations (policy admin, underwriting, billing, claims, etc.) into one integrated system. By unifying these mission-critical functions in one place, a core platform provides a single source of truth, automates manual work, and embeds compliance rules into workflows – all of which reduce operational risk and cost while enabling agility.

Over the past few years, these core platforms have risen in popularity due to industry-wide digital transformation. Most modern cores have shifted to cloud-native, API-first architectures. This gives insurers on-demand scalability, improved customer experiences (think instant digital quotes and self-service), continuous upgrades, and plug-and-play connectivity to emerging technologies. For example, a cloud-ready core can easily integrate new InsurTech tools – from generative AI that helps triage claims to real-time IoT-based pricing engines – much more easily than legacy systems.

Industry analysts like McKinsey note that such cloud-ready core systems have become the foundation for faster insurance product launches, real-time analytics, and stronger ecosystem integration, making core modernization one of the insurance industry's top strategic priorities.

$130 BILLION
per year spent on modernization of legacy insurance systems in the US alone in 2024.

Source: Intelias

74% of insurers still rely on legacy tech for core functions.

Source: Earnix

41%
higher IT cost per policy on legacy platforms compared to modern, scalable and customizable systems

Source: Sandis.io

Speed-to-Market as a Competitive Advantage in 2026

Customers now expect instant, digital experiences from insurers – whether it's getting a quote on a mobile phone in seconds or reporting a claim through an app.

To keep up, **insurers and MGAs (managing general agents) need the ability to introduce new products and services in weeks, not years**. This priority on speed-to-market has overtaken the old "build vs. buy" IT debate.

In the past, insurers chose between building custom software in-house (which was highly tailored but slow and expensive) or buying off-the-shelf suites (which were faster to deploy but rigid). Today, that binary choice is fading. The best approach is often **leveraging a modern insurance core platform that combines the flexibility of custom-builds with the speed of ready-made solutions**. Crucially, modern core platforms help insurers overcome the limitations of legacy cores. Legacy COBOL or AS/400 systems, for example, can take weeks or months to implement even small changes, making it nearly impossible to rapidly experiment with innovative insurance products or new distribution channels.

Contemporary platforms address this by offering configurable product definitions, automated workflows, and open APIs that dramatically shorten development cycles. Even traditional core suite vendors like Guidewire and Duck Creek have updated their offerings to be cloud-based and more flexible, yet they often still carry remnants of older architectures that make deep customization complex and upgrades vendor-controlled. This is why many insurers are looking at next-generation solutions that enable faster iteration. In short, the ability to launch insurance products quickly and adapt on the fly has become a decisive competitive advantage in today's market.

> Insurers no longer have to waste months and millions on rigid legacy systems. A modern core platform gives teams the freedom to launch new software and products rapidly, iterate continuously, and scale without friction. It's the most effective way to innovate while keeping budgets under control.
>
> **Michał Głomba**
> CEO Openkoda

How Core Platforms Speed Up Insurance Software Development

Modern insurance core platforms are expressly designed to accelerate the development of custom insurance applications and new products. They achieve this by providing pre-built components and frameworks that a development team would otherwise have to build from scratch. Key ways these platforms speed up the process include:

- **Unified Core Functions:** Instead of integrating dozens of siloed systems, a core platform offers all fundamental insurance functions (policy management, claims processing, billing, etc.) in one package
- **Pre-Built Insurance Logic & Templates:** Many platforms come with ready-made business logic and templates for common insurance processes. For example, a core platform might include a pre-built claims module with standard workflows for intake, adjudication, and settlement, or templates for policy administration with quote, bind, and renewal processes already modeled
- **Reusable Modules and APIs:** Modern core systems are often modular and API-driven. Insurers can plug in new modules (for instance, a new rating engine or an AI fraud detection tool) without starting over.
- **Low-Code/Config Tools:** Many platforms include configuration studios or low-code designers for defining products, business rules, forms, and workflows through visual interfaces.
- **Built-in Compliance and Data Models:** Insurance platforms typically embed regulatory compliance features (like ISO-standard forms/rates for P&C, or Solvency II reporting for EU insurers) and provide standard data models for policies, claims, etc.

All these factors combine to compress development timelines significantly. What might have been a 12-month software project can often be delivered in a few months with a modern core platform. In fact, **using a robust insurance platform as a foundation can cut development time by roughly 50–60%,** because teams start with a working system and only need to build the unique parts on top.

Openkoda Insurtech Platform: A Faster Path to Innovation

One standout example of this new breed of core platforms is Openkoda

<u>Openkoda</u> is an **open-source, core insurance application platform purpose-built to meet these MGA needs**. It enables insurance innovators – including MGAs and insurtech startups – to develop and deploy custom applications and new products far more rapidly than traditional methods.

Openkoda is essentially a modern insurance application development platform that comes with a fully customizable, enterprise-grade foundation out-of-the-box. It was created to let insurance organizations focus on innovating their products and processes, rather than building basic infrastructure or being constrained by vendor-provided software.

What is Openkoda?

 In summary, Openkoda provides a modular architecture, pre-built templates, and configurable components that significantly reduce the time, cost, and complexity of delivering new insurance software. It is open-source (MIT-licensed) and built on popular technology (Java, Spring Boot, PostgreSQL, and React/JavaScript), so users have full access to the source code and familiarity of a mainstream tech stack. The platform is tailored for a range of scenarios – from large insurers modernizing legacy systems, to agile MGAs launching a digital-native product line.

Out of the box, Openkoda ships with production-ready templates for core insurance applications, including claims management, policy administration, and embedded insurance offerings. Each template comes with built-in domain models (data structures for policies, claims, etc.), business logic workflows, and even UI components specific to that use case. In practical terms, this means when you start a project with Openkoda, you're not starting from zero – you have a working baseline system where, for example, a claims module already knows how to intake a claim, assign it a number, track status, and so on. You can then extend and customize it to fit your unique requirements.

Openkoda Details

Openkoda also includes a library of plug-and-play feature modules that address common needs across insurance applications. For instance, it provides a Reporting AI module for natural-language data queries and analytics, embeddable web forms for collecting customer inputs, an underwriter dashboard framework, automated document generation for policy/claims paperwork, role-based access control for permissions, and support for unlimited user management.

These features are all fully extendable via the open-source codebase – insurers can modify how they work or integrate new ones as needed. Moreover, Openkoda is API-first: every entity (policy, claim, etc.) automatically comes with REST and GraphQL APIs, plus event hooks, making integration with external systems or digital channels straightforward.

This is crucial for use cases like embedded insurance (integrating insurance offers into third-party websites/apps) or connecting with existing enterprise systems – the platform was built to connect easily with CRMs, payment gateways, underwriting tools, and more.

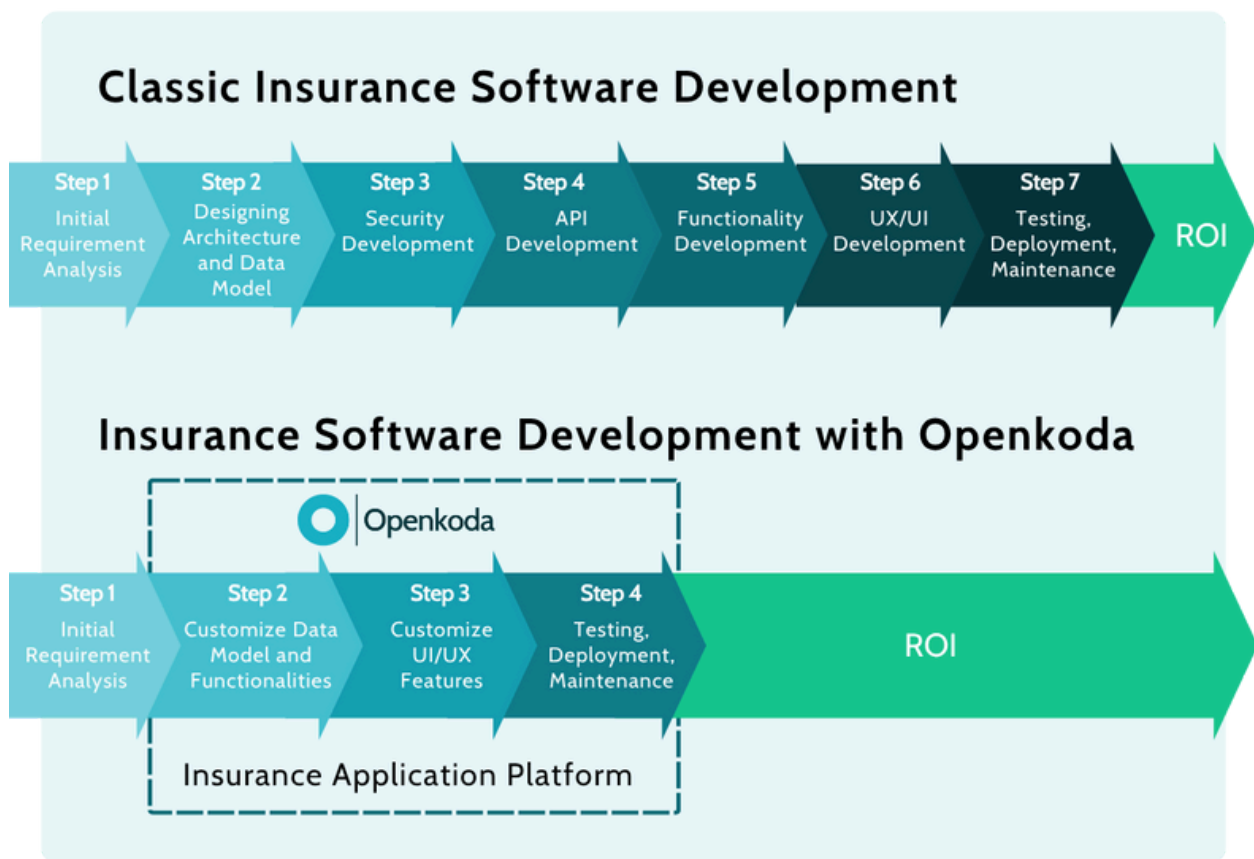## Insurance Software Development With Openkoda

From a development process standpoint, Openkoda acts more like a complete application foundation rather than just a toolkit. It provides core services like authentication, user/role management, audit logging, workflow engines, a visual data-model builder, and even a CMS for managing content and documents.

Developers using Openkoda can generate a new insurance application with a one-line Docker deployment, then immediately start adding custom logic or UI on top of the existing, working system. eams spend far less time on boilerplate coding and infrastructure setup. Openkoda's makers report that this approach lets insurers deliver new software modules up to 60% faster than traditional custom development would allow. Crucially, the insurer retains full control: you own the code and can decide how much to customize or extend it.

Adaptable Core Platform

Openkoda is deliberately designed to adapt to any insurance use-case. Whether it's building an AI-driven claims triage system, an automated underwriting solution, a usage-based insurance product with IoT data, or a niche insurance product for a specialized market, the platform provides the building blocks to get started quickly.

For example, a startup could use Openkoda to prototype a new parametric insurance offering (like flight delay insurance) in a sandbox environment, leveraging the built-in policy and claims templates, and later deploy it to production knowing the underlying architecture is already enterprise-grade. This ability to incubate innovation quickly while still maintaining a robust foundation is a major strategic benefit for insurance businesses using Openkoda.

Classic Insurance Software Development

Step 1 — Initial Requirement Analysis
Step 2 — Designing Architecture and Data Model
Step 3 — Security Development
Step 4 — API Development
Step 5 — Functionality Development
Step 6 — UX/UI Development
Step 7 — Testing, Deployment, Maintenance
ROI

Insurance Software Development with Openkoda

Openkoda

Step 1 — Initial Requirement Analysis
Step 2 — Customize Data Model and Functionalities
Step 3 — Customize UI/UX Features
Step 4 — Testing, Deployment, Maintenance
ROI

Insurance Application Platform

# Key Advantages of Openkoda Over Traditional Platforms

Openkoda's approach differs markedly from older insurance software suites (such as Guidewire or Duck Creek) and from generic platforms like Salesforce. Below are the strategic advantages that Openkoda offers, underscoring why it has gained attention as a next-generation core platform:

Complete Freedom to Customize

Openkoda gives insurers total flexibility to modify and extend the system in any way needed. Because the entire platform is provided as open source code, there are no hidden or locked modules – **every layer (data model, business logic, UI, integrations) is accessible and changeable.** his is in contrast to traditional core systems that, while configurable, often keep certain functions as "black boxes" or require using proprietary tools. For example, Guidewire allows custom extensions but only within its proprietary Gosu language and frameworks, imposing some limits on how you implement changes.

Full Code Ownership

With Openkoda, insurance organizations own the code and intellectual property of their solution. The platform's core is MIT-licensed, and clients are free to fork the code, inspect it, and maintain it independently if they choose. Any customizations your team builds remain yours. This is a stark departure from typical vendor relationships where the core software is licensed and remains the vendor's property. In legacy suites, you might heavily configure the system, but you don't own the base code – leaving you dependent on the vendor for upgrades or fixes. Openkoda eliminates that dependency. Carriers keep full ownership of the IP, avoiding long-term lock-in to a single vendor's roadmap.
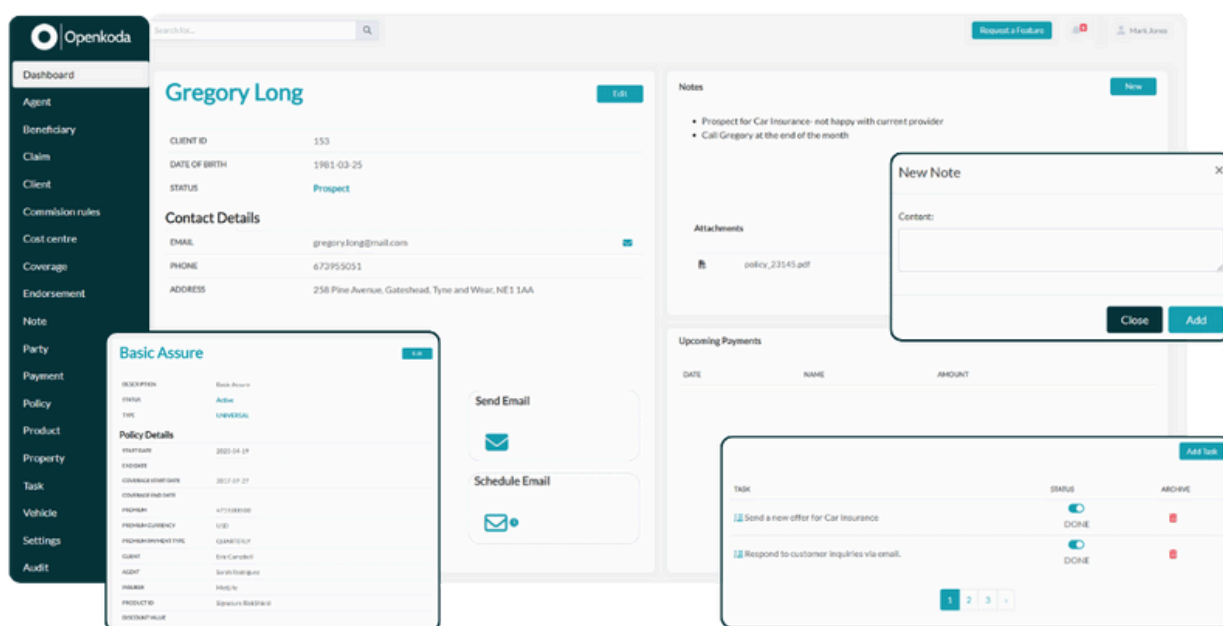
Flexible Deployment

Unlike many modern software platforms that are available only as vendor-hosted SaaS, **Openkoda can be deployed on your choice of infrastructure.** You can run it in your own private cloud or data center, on public cloud services like AWS/Azure/GCP, or even in hybrid modes. The platform is cloud-agnostic and even supports multi-tenant setups if needed, but importantly, you control where it runs. This flexibility is a huge advantage for insurers with strict data residency, security, or integration requirements that necessitate on-premise deployments. Even for those embracing cloud, having the ability to switch cloud providers or run in a private environment means you aren't tied to the vendor's hosting. By contrast, Salesforce runs only in its own cloud, and systems like Duck Creek or Guidewire's latest versions strongly encourage (or mandate) their proprietary cloud deployments for full functionality.

Openkoda's approach frees companies from such constraints – you get to leverage cloud benefits on your terms, or stay on-prem as needed. Moreover, you decide when to upgrade or apply updates; you're not forced into a vendor's update schedule or multi-tenant upgrade cycle

---

No Vendor Lock-In

Openkoda is designed to be vendor-agnostic in every sense. There is no long-term dependency that could "trap" a customer. Because the source code is open and standard tech, an insurer could continue using and evolving the platform even if they stopped taking services from the original vendor. Integrations are done via open APIs and standard protocols, not proprietary connectors that only the vendor can provide.

This stands in contrast to many traditional platforms where moving off the system is extraordinarily difficult (sometimes requiring a complete rewrite of business logic and data migration). For example, companies that built extensive workflows in Salesforce often find that leaving Salesforce means rebuilding those processes from scratch elsewhere.

Built for Speed and Innovation

Openkoda's core value proposition is helping insurance players get to market faster with new ideas. Its pre-built templates and modules jump-start development so that a minimally viable product can be stood up very quickly (often within a few months or even weeks, depending on complexity).

This agility is ideal for innovation teams, greenfield insurtech ventures, or MGAs testing novel insurance schemes. Because the platform handles generic requirements (user management, security, basic policy/claims logic, etc.).

In essence, it provides a sandbox that is already production-grade. Insurers can pilot new offerings with minimal development investment and iterate quickly. This speed-to-market, combined with the above advantages (freedom, ownership, no lock-in, etc.), means Openkoda can significantly reduce both the time and risk involved in insurance innovation. In fact, use cases have shown **development cycles being reduced by about 50–60% compared to a ground-up build.**

Example: Building an Innovative Insurance Product with Openkoda

To make the benefits more concrete, let's walk through a hypothetical scenario of creating a new insurance product using Openkoda. **Imagine an MGA has an idea for an innovative parametric insurance offering – say, a travel insurance that automatically pays out if a flight is delayed by more than 2 hours.** This product involves real-time data (flight status feeds), automated claims triggers, and a customer-facing portal for purchase and claims status. Here's how the MGA could rapidly build it with Openkoda.
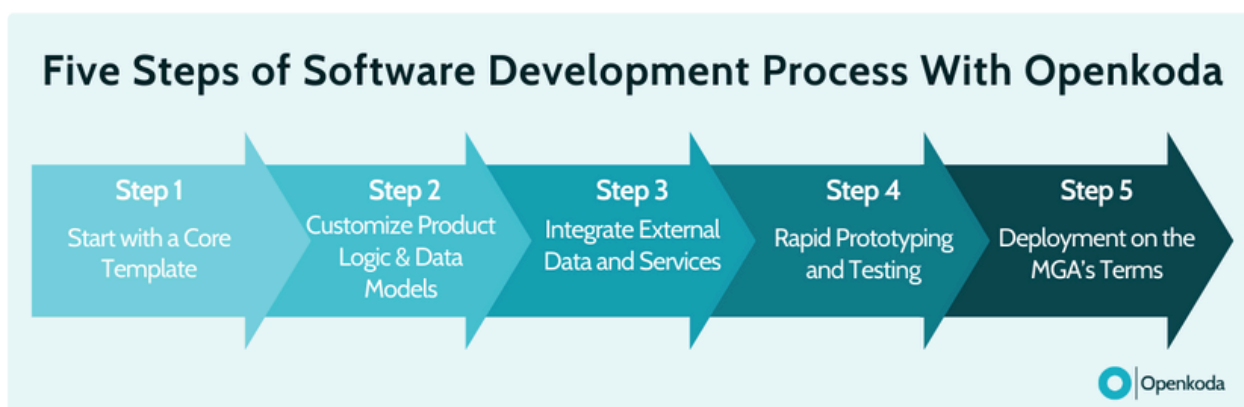
Step 1: Start with a Core Template

The MGA begins by spinning up Openkoda and selecting a relevant template as the foundation – for example, the **Policy Management Software Template** combined with elements of the **Claims Management Software Template** (since this product will need to issue policies and handle automatic claims). Openkoda's policy administration module already includes common fields (policyholder info, coverage details, etc.) and workflows for issuing and managing policies, while the claims module provides workflows for claim events. By leveraging these, the MGA doesn't need to code basic insurance processes from scratch. They get a working policy & claims system on day one, which they can then customize for the parametric features.

Step 2: Customize Product Logic & Data Models

Next, the MGA's team configures the unique aspects of the flight-delay insurance. Using Openkoda's data model builder and open codebase, they add custom fields like "Flight Number" and "Departure Time" to the policy schema. They also create a new claims rule: if a flight is reported delayed past the threshold, the system should automatically validate the policy and calculate a payout. Openkoda allows them to implement this logic either through its built-in workflow/rules engine or by writing a simple Java routine. This flexibility means even complex bespoke rules can be implemented without limitation. For instance, actuaries could define the payout formula and developers encode it directly, rather than being stuck with only what a vendor's rule engine supports.

Step 3: Integrate External Data and Services

Parametric insurance relies on external triggers, so the MGA integrates the platform with a flight data API. Thanks to Openkoda's API-first architecture, every part of the system is already exposed via REST/GraphQL endpoints. The team uses these APIs to create a seamless feed: a flight status service sends a webhook or periodic update to an Openkoda endpoint whenever a delay occurs, referencing the policy's flight number. Openkoda's integration capabilities (including support for webhooks and event handling) make this straightforward.

Step 4: Rapid Prototyping and Testing

Within a short few weeks, the MGA has a basic version of the product running. They use Openkoda's built-in client portal components to set up a simple web interface where customers can purchase the flight delay coverage and later check claim status. Because Openkoda already had a security and user management layer, setting up a portal for customers (and an internal dashboard for underwriters) is mostly configuration, not a heavy development task.

Step 5: Deployment on the MGA's Terms

Satisfied with the MVP, the MGA deploys the application to production. They choose to host it in their own cloud account for data control. With a one-line Docker deploy process, launching in the cloud is uncomplicated. Crucially, as usage grows, they can scale horizontally – Openkoda's microservices architecture and support for clustering means the application can handle increasing load by adding more server instances without rework. And because Openkoda does not charge per user or per transaction, whether they sign up 100 or 100,000 customers, the MGA's costs remain predictable.
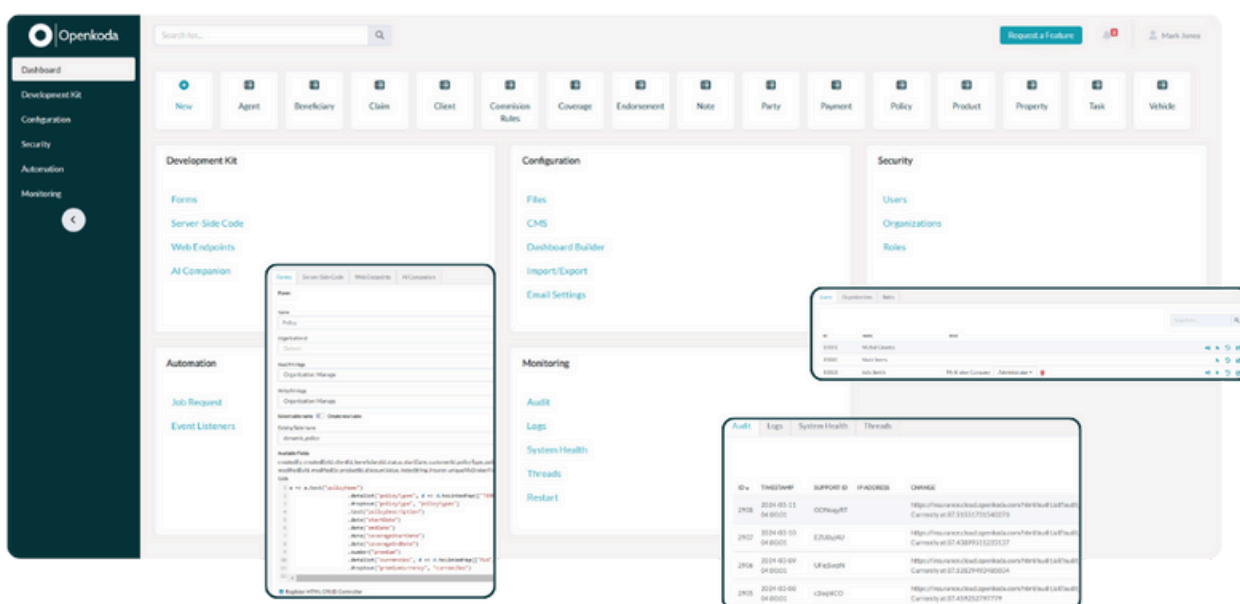
In this scenario, what might have been an extremely daunting project – connecting real-time external data, automating claims, building portals, ensuring scalability – becomes manageable and fast with the help of an insurtech platform.

Openkoda provided the <u>MGA</u> a running start, handling the heavy lifting of core insurance functionality and infrastructure, so the MGA's team focused on the innovative elements of their parametric product. The end result: in a matter of a couple of months, not only did they have a working product in market, but they also maintained full ownership of a cutting-edge system that they can build on in the future. This exemplifies how using a modern core platform like Openkoda can translate strategic ideas into operational reality with speed and flexibility that traditional approaches simply can't match.

Strategic Outlook: Embracing Platforms for Insurance Innovation

For insurance businesses – whether incumbent carriers modernizing their offerings, or nimble MGAs and startups creating the next big thing – core platforms like Openkoda represent a powerful strategic enabler. They combine the reliability and depth of insurance-specific functionality with the agility to adapt and innovate. By adopting such a platform, insurers can dramatically accelerate their product development lifecycles, launch new insurance products with far less risk, and avoid the pitfalls of vendor lock-in or runaway IT costs.

Openkoda in particular demonstrates that an open-source, customizable core platform can deliver on the key needs of the industry: fast time-to-market, complete control over the technology, and freedom from the constraints of traditional vendors. Companies that leverage Openkoda gain a competitive edge in being able to quickly roll out tailored solutions – whether it's entering a new niche, integrating insurance into new channels (embedded insurance), or automating and optimizing internal processes with AI and analytics.

In summary, insurance core platforms have risen to prominence because they allow insurers to do what once seemed impossible: develop custom insurance software and innovative products rapidly, at lower cost, and with the scalability and intelligence demanded by today's market. Openkoda exemplifies this new generation of platforms by marrying the strengths of open technology with insurance domain expertise.