Resolute

case study –

Zero-latency, real-time processing of vast data volumes in .NET

Resolute enables a cutting-edge geospatial mapping solution to dynamically visualize and track over 1,000 events simultaneously every second.

Blazor .NET CesiumJS GIS DBs SignalR

Telerik UI for Blazor SQL Server

Company

A global aviation technology provider

Customer since

2021

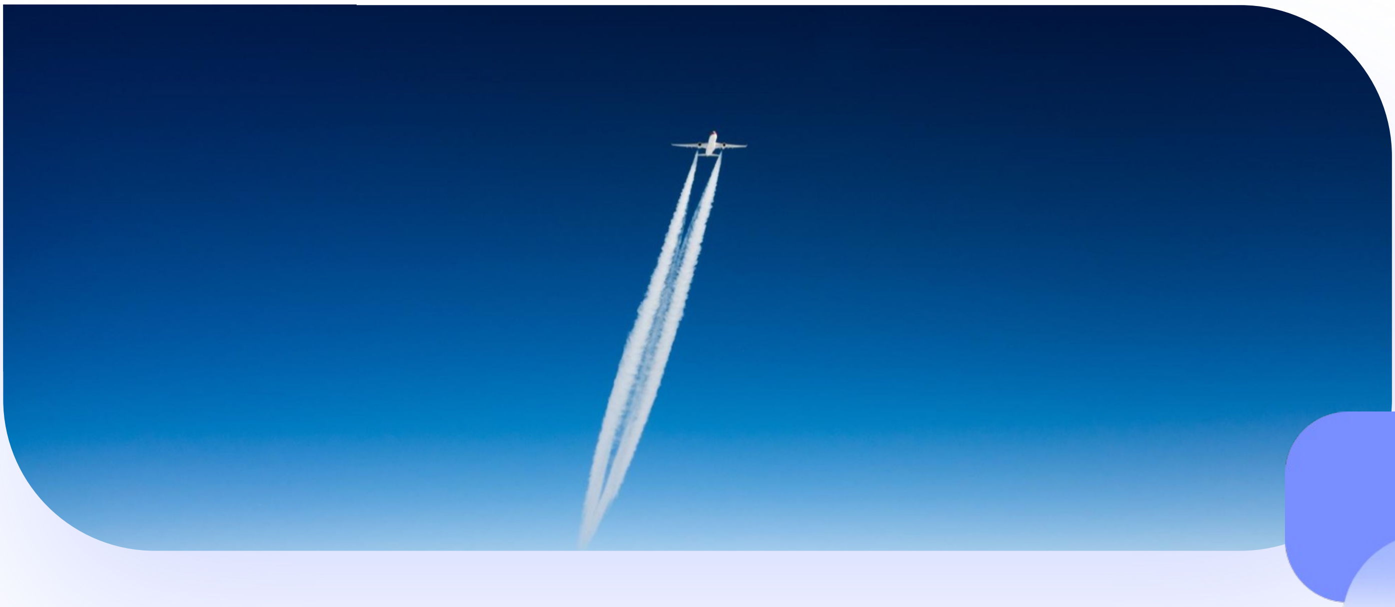
Goals

- ✓ The platform must simultaneously process thousands of events, like geographic coordinate changes. The platform works with live data, updating continuously to reflect real-time locations with every change in coordinates and object type.

- ✓ A pluggable, framework-agnostic solution architecture, enabling the integration of new frameworks at any time without affecting the core functionality of tracking and displaying objects in real time.

- ✓ The platform must accommodate both live and history data modes.

- ✓ The platform must enable real-time audio and video streaming capabilities.



background –

A global aviation technology manufacturer

The platform leverages Blazor for its frontend and .NET for its backend architecture. It incorporates ArcGIS to dynamically visualize a multitude of objects, enabling the display of geospatial data in real time.

Instead of using preset data, the platform is built on real-time data, accounting for coordinate changes of various aircraft object types to accurately determine each object's live location. The system is designed to monitor the positions and comprehensive properties of radio-emitting objects, processing well over 1,000 events and their changes in geographic coordinates concurrently.

the challenge –

Overcoming real-time data and integration challenges within an event-driven platform architecture

One notable engineering challenge the Resolute team had to tackle was processing many events in real time. This issue was addressed by adopting Kafka, an open-source distributed event streaming platform. An additional complication arose from the system's frontend being developed in Blazor, while the frameworks intended for use were JavaScript-based. A critical performance challenge that arose was managing the flow of real-time events between Blazor and JavaScript efficiently through Blazor's JavaScript Interop capabilities. To overcome this, Resolute engineered two separate SignalR hubs – one for the Blazor UI and another for the JavaScript-based UI. This approach effectively reduced the burden on the JS Interop, ensuring the system's performance was maintained at an optimal level despite the high volume of concurrent events.

Another challenge the Resolute team addressed was enabling real-time updates for all users on the platform, operating in an event-based system that conforms to REST-ful client-server communication principles. A key feature required by the client was to allow multiple users to concurrently edit the same object attributes, while observing real-time updates on its status. This functionality was developed by integrating Kafka with a thin layer developed by Resolute. The team implemented a standard observer pattern, consistently applied across both the server and the client sides, powered by Kafka's streaming capabilities.

At the client end, this thin layer captures real-time events, forwarding them to the server through a SignalR connection. Updates are subsequently managed on the server using the same observer pattern. Leveraging Blazor has enabled significant code reuse between the server and the client, boosting efficiency and ensuring system uniformity.

Finally, Resolute addressed the challenge of enabling users to independently develop and locally deploy 3rd party plugins, without modifying the system's core, by creating a specialized internal plugin SDK. This SDK leverages Kafka's event-based architecture and SignalR infrastructure, allowing for the seamless integration of 3rd-party plugins. This approach ensures users can easily extend the system's functionality with their own plugins, maintaining the integrity and efficiency of the application.

"In the face of real-time event processing hurdles, our team harnessed Kafka and SignalR's power, bridging the gap between Blazor and JavaScript. This synergy, coupled with a dedicated SDK for plugin development, not only streamlined event processing but also unlocked new dimensions of flexibility and efficiency for our customer." explained Veli Pehlivanov, Co-founder and CTO, Resolute Software. "It's a testament to our commitment to innovation and our relentless pursuit of excellence."

technologies we used –

- → .NET
- → Blazor
- → SQL Sever
- → Kafka
- → SignalR
- → 3rd party JavaScript cards
- → GIS DBs (Postgres with GIS extensions)

A robust platform equipped to process 1,000s of concurrent events in real time

The Resolute team not only overcame the complexities of processing a massive volume of concurrent, real-time events, but they also introduced long-awaited functionalities to the platform, elevating its value and versatility:

Dual data modes

support for both live and historical data modes

Screensharing

with real-time video and audio to enhance collaboration within teams

Robust system health checks & auto-recovery

health monitoring and automatic UI recovery based on microservice performance, ensuring system resilience

Resolute

sales@resolutesoftware.com

Let's talk about your technology requirements.

Get in touch

USA

MA 01701, Framingham,
945 Concord St,

+1-617 386-9697

Deloitte. Company to watch | Tech Fast 50

Business Professional Services

Clutch ★★★★★

People First Company Award
2020 | 2021 | 2022