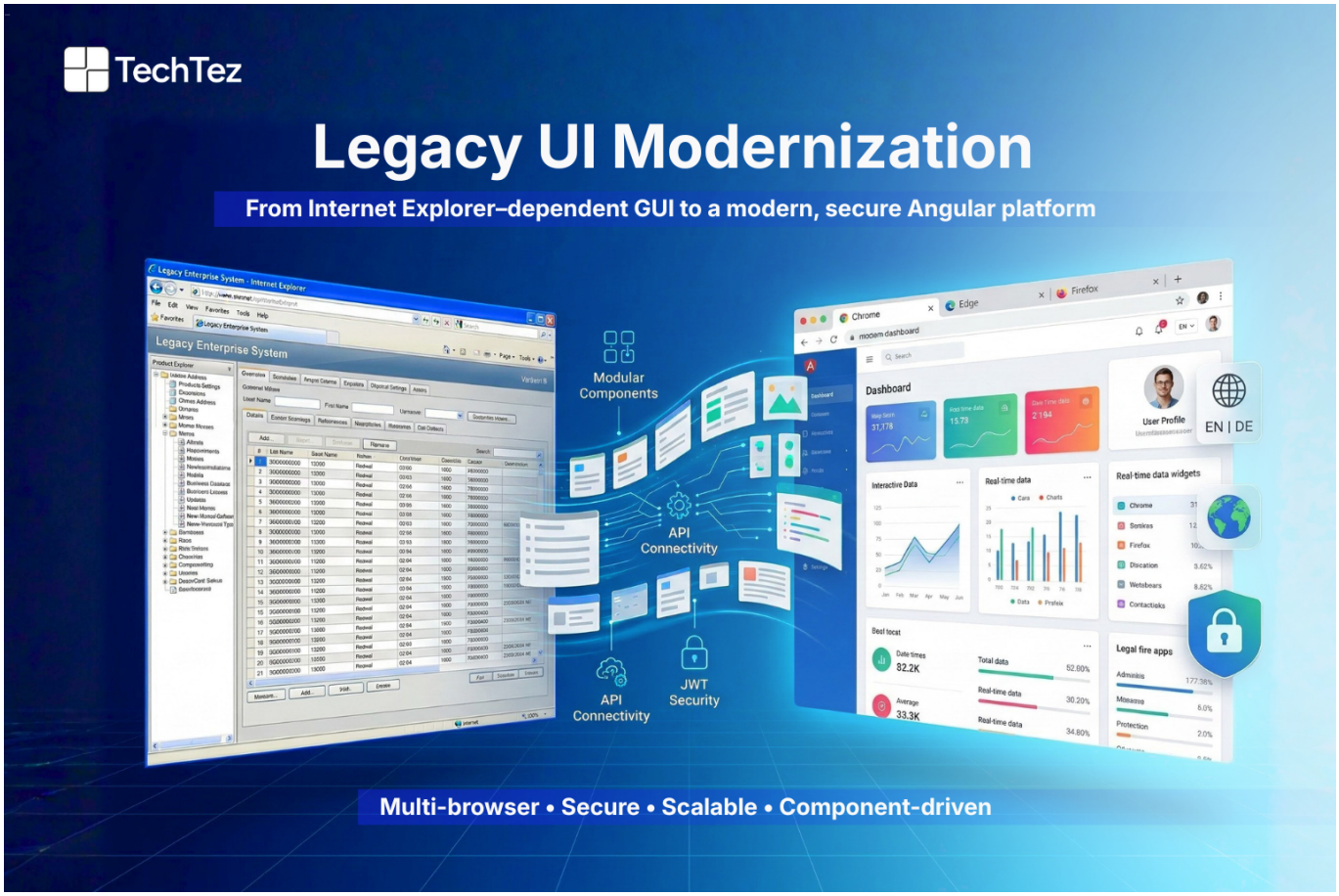
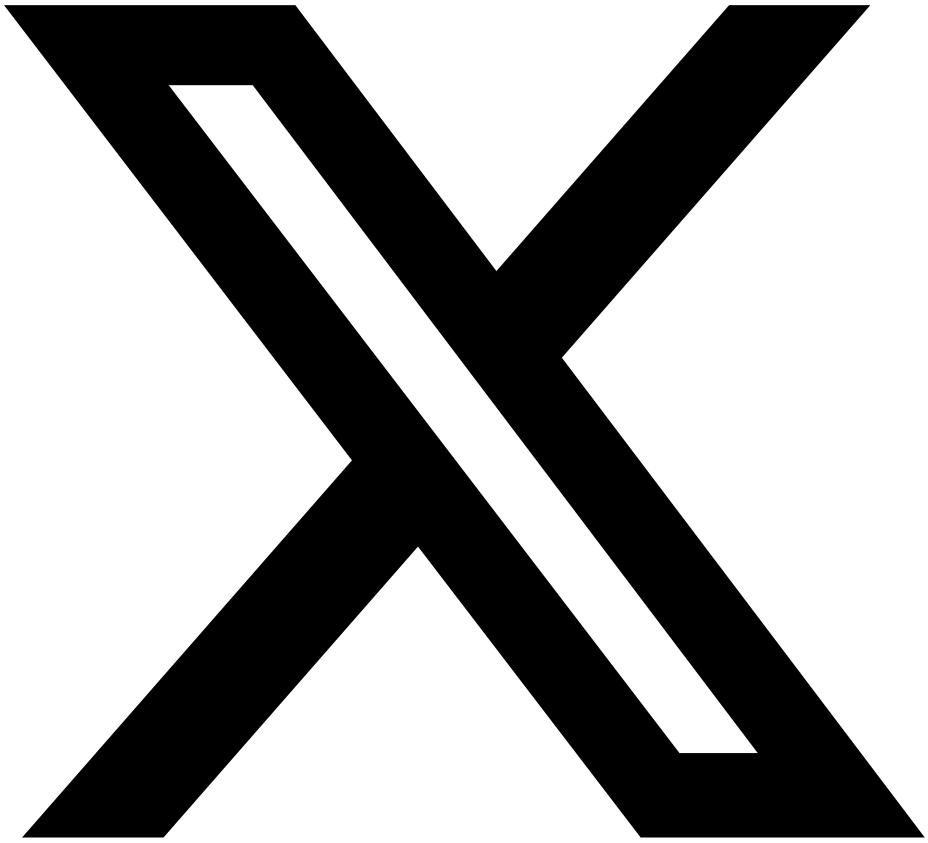
Legacy UI Modernization – Digital Transformation

Posted on April 17, 2026 by Sony Battina


TechTez
Legacy UI Modernization
From Internet Explorer–dependent GUI to a modern, secure Angular platform
Legacy Enterprise System
Modular Components
API Connectivity
API Connectivity
JWT Security
Chrome
Edge
Firefox
Dashboard
User Profile
EN | DE
Interactive Data
Real-time data
Real-time data widgets
Legal fire apps
Multi-browser • Secure • Scalable • Component-driven

Case Study

Linkedin X-twitter



Facebook

Modernizing Legacy Systems for a Leading Telecom Company

Transforming a legacy GUI into a modern Angular platform by enabling secure, scalable, multi-browser access

Client

An enterprise telecom software provider delivering customer-facing applications to multiple service providers across regions. The application played a critical role in day-to-day operations and customer interactions, making stability, security, and usability essential.

The client’s vision was to modernize an aging GUI framework that had become increasingly risky and costly to maintain, while preserving existing backend systems and business workflows.

Challenges

During analysis, we found that most booking and assistance platforms operate in silos, requiring users to manually connect multiple tools to complete a single task.

Key challenges included:

- The application supported only Internet Explorer 11 and earlier versions, making it incompatible with modern browsers

- 
With Internet Explorer reaching end-of-life, continued usage posed operational and compliance risks

- The UI was built on a home-grown proprietary framework that was no longer supported by internal development teams

- Increasing security vulnerabilities due to outdated UI technologies and lack of modern authentication mechanisms

- UI changes and customer-specific customizations required significant development effort, slowing down delivery cycles

-  Limited flexibility to support localization, theming, or future feature expansion

The client needed a future-proof UI architecture that could modernize the user experience, enhance security, and reduce long-term maintenance costs without disrupting backend systems or ongoing customer operations.

Our Strategy:

TechTez adopted a structured, risk-aware UI transformation strategy designed to modernize the application while ensuring business continuity and long-term sustainability.

The strategy focused on decoupling, modularity, and security, enabling the UI layer to evolve independently without impacting backend systems or operational workflows.

- **UI Re-Architecture Using Angular** : The legacy GUI was fully re-architected using Angular’s component-based framework, enabling clear separation of concerns and improved maintainability. This allowed individual UI components to be developed, tested, and enhanced independently.

- **Component-Driven Templating for Reusability** : A templating approach was implemented where reusable UI components could be easily configured and customized for different customers. This significantly reduced duplication and simplified customer-specific adaptations.

- **Frontend–Backend Decoupling via REST APIs** : New RESTful APIs were developed to enable clean communication between the frontend and backend systems. This decoupling ensured that UI changes could be rolled out without requiring backend modifications.

- **Enterprise-Grade Security Implementation** : JWT-based authentication and authorization were introduced to secure UI access and API communication. This enhanced the overall security posture and aligned the application with modern enterprise security standards.

- **Localization and Global Readiness** : Internationalization (i18n) support was built into the UI, enabling easy localization and future global deployments without structural changes.

- **Accelerated Development Using Open-Source UI Components** : PrimeNG open-source widgets were leveraged to standardize UI elements, improve consistency, and accelerate delivery while avoiding proprietary licensing dependencies.

Tech Stack:

- Angular Framework

- PrimeNG Open-Source UI Components

-  RESTful APIs

- JWT-based Authentication and Authorization

- Component-based UI architecture

- Modular and reusable design patterns

- Designed for scalability, security, and long-term maintainability

- Internationalization (i18n) Library

Platform Highlights

- **Component-Based UI Architecture:** Modular Angular components enabled high reusability, easier testing, and faster enhancement cycles across customer deployments.

- **Reduced Customization Effort:** Reusable templates and configurable components reduced successive customer development effort by up to 50%. The assistant operates as a toolbox of functions and services, dynamically assembling tool-augmented workflows.

- **Modern Multi-Browser Support :** Full compatibility with modern browsers eliminated dependency on Internet Explorer and reduced operational risk.

- **Secure API-Driven Communication:** REST-based integration with JWT security ensured safe and controlled data exchange between UI and backend services.

- **Improved Security Posture:** Modern authentication mechanisms significantly reduced vulnerabilities associated with the legacy framework.

- **Globalization-Ready UI:** Built-in i18n support enabled seamless localization for multi-region deployments.

- **Future-Proof Architecture:** The decoupled, modular design allows new features, UI enhancements, and integrations to be introduced with minimal disruption.

Results & Impact


TechTez
Measurable Outcomes of UI Modernization
↓50%
Component-driven Angular UI
+ reusable templates
↓100%
Modern multi-browser
Angular web platform
↑40%
Frontend–backend decoupling
via REST APIs
↓60%
JWT authentication
+ modern access control
↑30%
Reusable PrimeNG components
+ modular design
Global-Ready UI
Built-in i18n and theming support

- 50% reduction in successive customer development effort due to reusable UI components

-  Eliminated dependency on unsupported proprietary frameworks

-  Enabled seamless multi-browser support, improving accessibility and usability

- Significantly enhanced application GUI security through modern authentication mechanisms

- Faster delivery of UI enhancements and customer-specific features

-  Established a scalable UI foundation for future digital transformation initiatives

Why It Matters

Many enterprise platforms continue to rely on outdated UI technologies that expose organizations to security vulnerabilities, browser compatibility issues, and high maintenance overhead.

This case study demonstrates TechTez’s ability to:

-  Modernize mission-critical enterprise applications without disrupting operations

- Design secure, scalable, and maintainable UI architectures aligned with modern web standards

-  Enable faster innovation through component-based and API-driven design

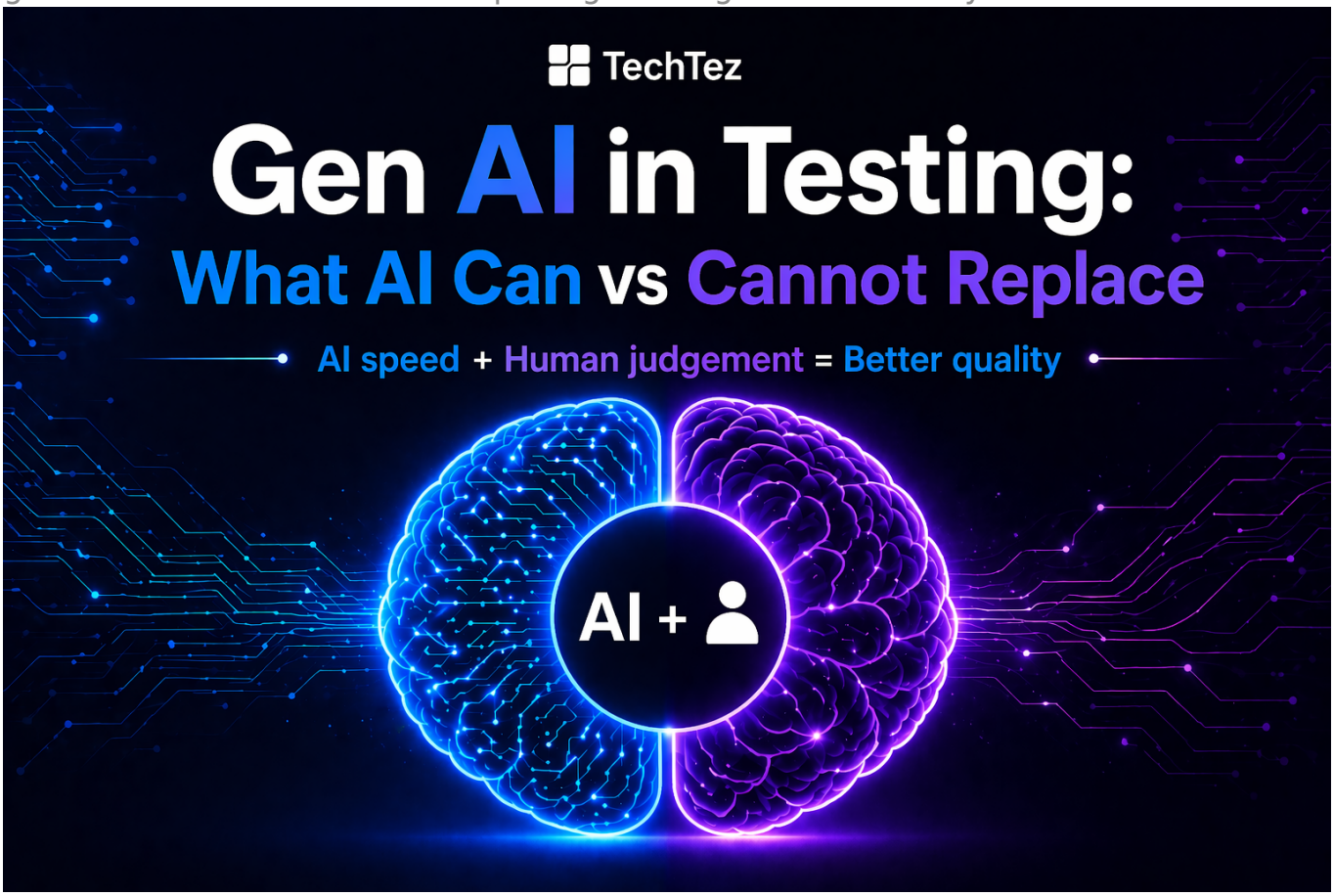
By transforming a legacy, IE-dependent GUI into a modern Angular-based platform, TechTez helped the client future-proof their application ecosystem delivering improved security, faster customization, and a superior user experience.

Our Thought Leadership Guides

- Case Study

GenAI for Test Case Generation: Hype vs Reality

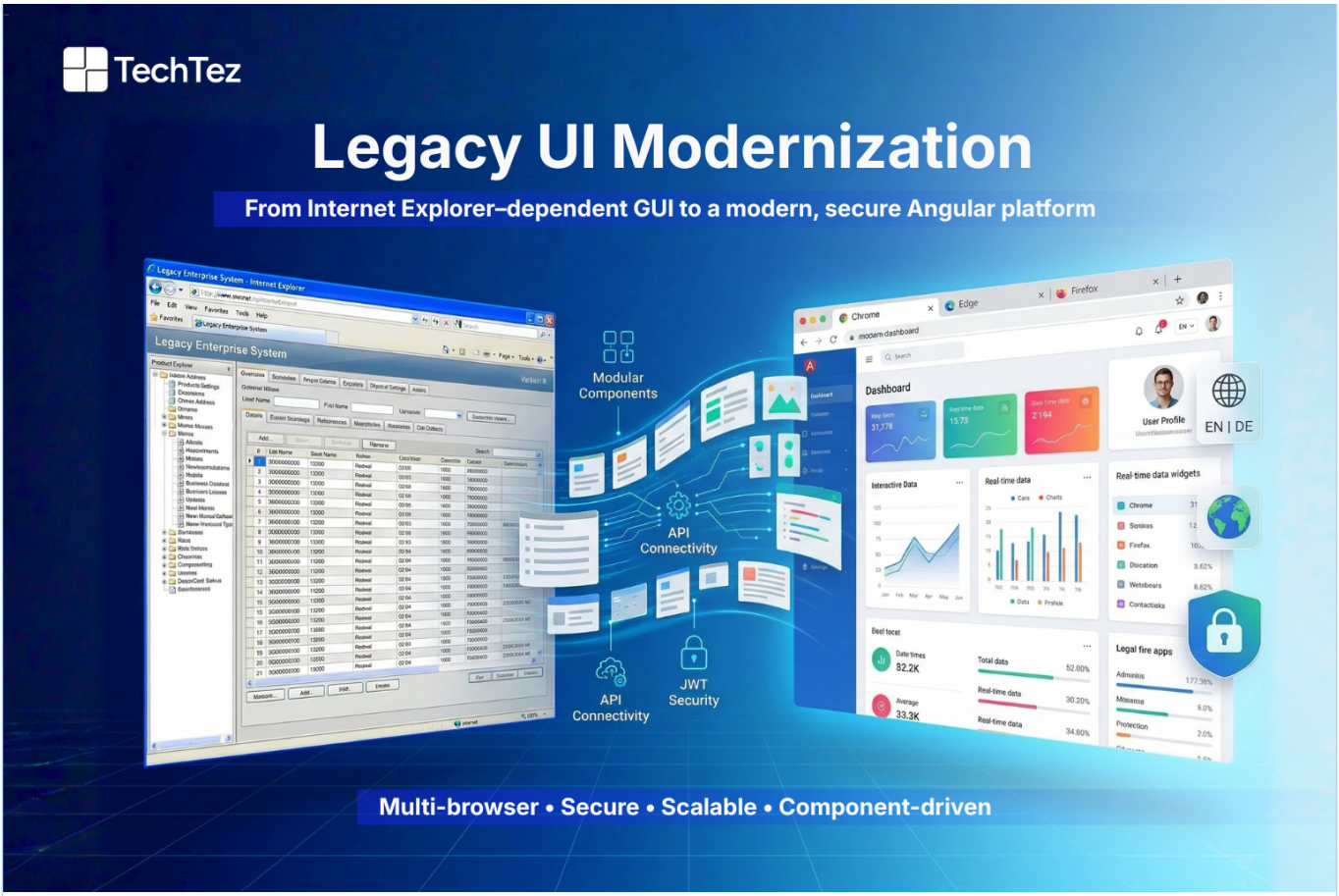
Compare AWS and Azure cloud costs and learn why architecture, automation, and governance matter more than pricing for long-term efficiency.



- Case Study

Legacy UI Modernization – Digital Transformation

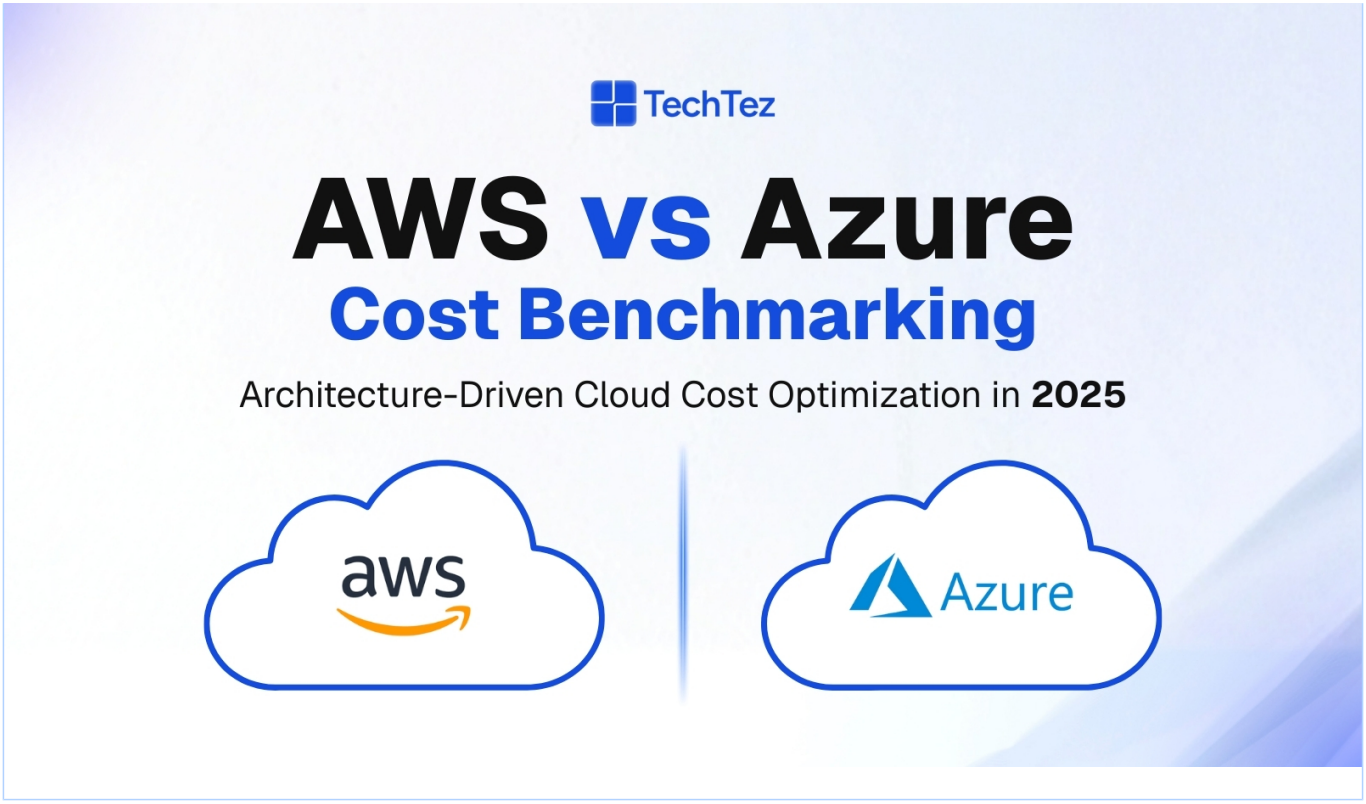
A legacy UI modernization initiative that transformed an IE-dependent, proprietary GUI into a secure, scalable, multi-browser Angular application using reusable components, REST APIs, and modern authentication enabling faster customization, lower maintenance effort, and future-ready digital operations.



- Case Study

AWS vs Azure Cost Benchmarking: Architecture-Driven Cloud Cost Optimization in 2026

Compare AWS and Azure cloud costs and learn why architecture, automation, and governance matter more than pricing for long-term efficiency.


TechTez
AWS vs Azure
Cost Benchmarking
Architecture-Driven Cloud Cost Optimization in 2025
aws
Azure