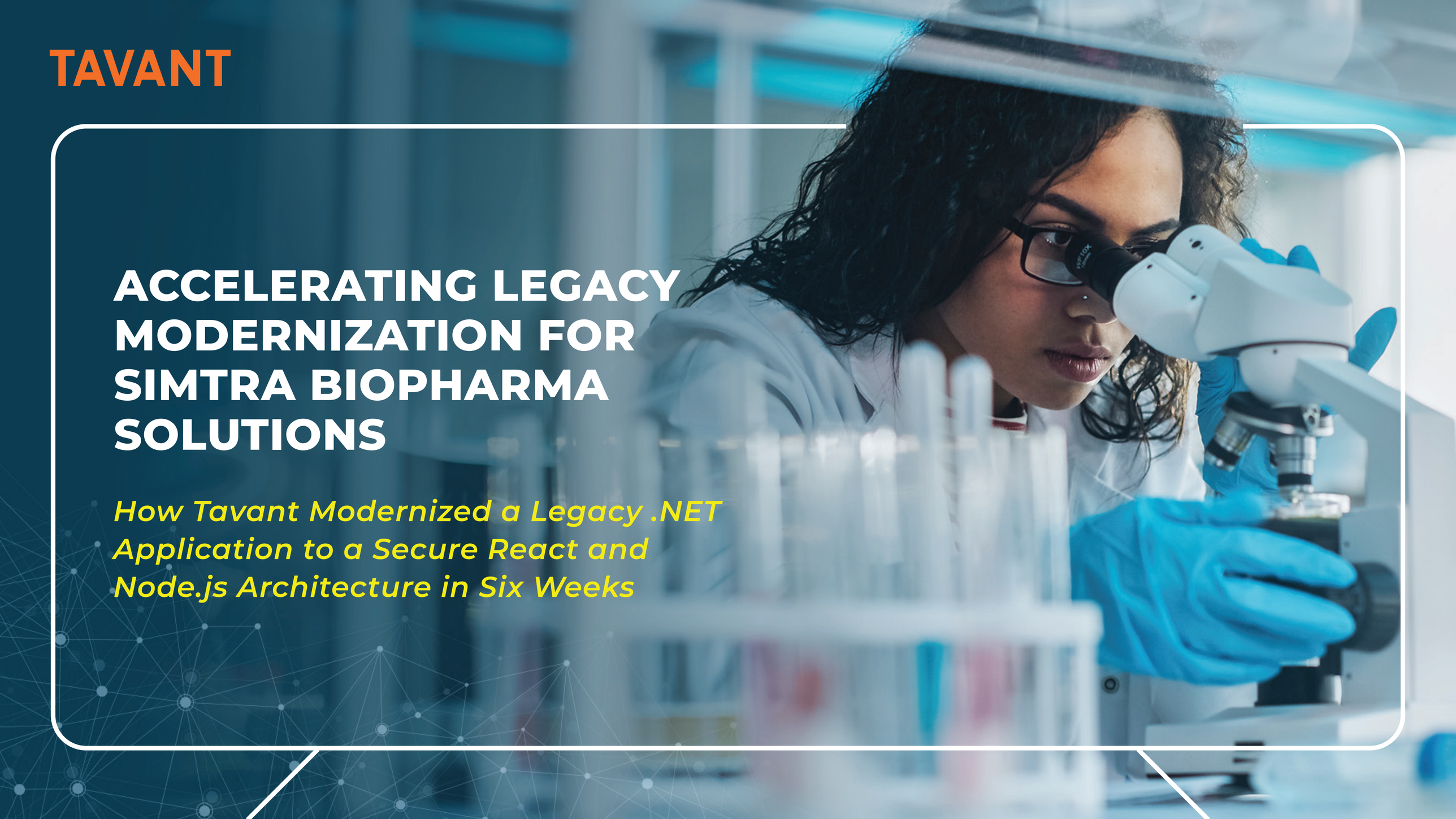
TAVANT

ACCELERATING LEGACY MODERNIZATION FOR SIMTRA BIOPHARMA SOLUTIONS

How Tavant Modernized a Legacy .NET Application to a Secure React and Node.js Architecture in Six Weeks



TAVANT

THE CLIENT AND THE CHALLENGE

Simtra Biopharma Solutions, a global CDMO specializing in sterile injectable medicines, relied on a legacy .NET Framework 2.0 web application that was past end-of-life, costly to maintain, and exposed to security risks including injection vulnerabilities, missing authorization controls, plaintext secrets, and weak sign-in. Simtra needed to modernize the application while preserving its existing behavior and output and continuing to support its on-premises SQL Server and IIS environment.

THE SOLUTION

Tavant used AIGNITE™ App Modernizer and Claude Code to modernize the application through a deterministic seven-stage methodology spanning Discovery, Intent Extraction, Capability Mapping, BDD Lock-in, Modernize Approach, CoCreate, and Parallel Run. A Strangler Fig approach allowed new and legacy functionality to coexist during the transition, while SME checkpoints, behavior-driven development, parallel-run validation, and 243 automated tests helped preserve application behavior and security as the application moved to React 18, Node.js 20, NestJS 10, and OIDC SSO.

THE IMPACT

The modernization replaced the end-of-life application foundation with a more secure and maintainable architecture while accelerating delivery and reducing engineering effort. Tavant addressed identified injection vulnerabilities, authorization gaps, and plaintext secrets, while behavior lock-in, automated testing, and parallel-run validation enabled Simtra to transition to the modernized application without disrupting users or changing application output.

TAVANT

RESULTS SNAPSHOT

45% Faster Delivery 6 weeks vs. an estimated 11-week traditional approach	**~61% Less Manual Engineering Effort** AI-assisted modernization reduced manual work across the migration lifecycle	**37% Lower Development Cost** Reduced effort and compressed delivery lowered development cost	**Zero Day-One Disruption** Parallel-run validation preserved application behavior through cutover

SERVICES SNAPSHOT

AGENT-LED LEGACY CODE MODERNIZATION

AIGNITE™ App Modernizer & Claude Code supported reverse engineering, intent extraction, capability mapping, code modernization, testing, and validation across the migration lifecycle.

DETERMINISTIC MODERNIZATION METHODOLOGY

A seven-stage modernization process provided a structured path from discovery through parallel-run validation, helping maintain control over application behavior throughout the transition.

CONTROLLED STRANGLER FIG MIGRATION

Priority components were modernized incrementally while new and legacy functionality operated together, reducing migration risk and maintaining business continuity.

BEHAVIOR LOCK-IN AND SECURITY HARDENING

SME checkpoints, BDD validation, parallel-run testing, and 243 automated tests preserved application behavior while addressing injection vulnerabilities, authorization gaps, plaintext secrets, and weak authentication.

MODERN STACK, WITHOUT BUSINESS DISRUPTION

The application moved from legacy .NET Framework 2.0 to React 18, Node.js 20, NestJS 10, and OIDC SSO, creating a more secure and maintainable foundation with zero disruption at cutover.