



Maximizing Longevity and Saving Costs Through Refactoring of SCE

////////

26-March-2026

Vaithi Bharath
&
Srinath Viswanathan





SCE Modernization Strategy

Refactoring a 20-Year Legacy Platform for the Future



CURRENT STATE
C & Java on Linux
20 years in production

KEY CHALLENGES

Outdated Technology

Retiring Engineers

Skills Shortage



REFACTORING APPROACH



Optimize Code

Streamline legacy
C & Java codebase



Automate

Introduce automation
across workflows



LLM Docs

AI-powered
documentation



Cloud Migration

Optimized file systems
for analytics



FOUNDATION FOR FUTURE INNOVATION

Posit RStudio — Open-Source Analytics

SAS Viya — Next-Gen Compute Engine

Preserving business functionality · Improving user experience · Enabling advanced analytics



Critical Challenges

Sustaining a 20-year legacy SCE built on C & Java

01



Outdated Technology Stack

- Legacy platforms with deprecated libraries
- Limited scalability and integration capabilities
- Difficult to adapt with modern data management capabilities

02



Retiring Engineers

- Institutional knowledge walking out the door
- Senior engineers with 20+ years of domain expertise in business as well as legacy technology stacks and proprietary systems approaching retirement

03



Talent Market Gap

- Severe shortage of professionals skilled in legacy technologies
- New graduates trained in modern languages (Python, R, cloud-native)
- Recruiting challenges for obsolete skillsets

Maintaining the status quo is not sustainable — modernization is imperative



Transformation Pathway

Preserving legacy value while modernizing for better management



Code Optimization

- Profile & optimize critical workflows
- Eliminate bottlenecks and redundancies while preserving validated business logic



Systematic Refactoring

- Migrate to modern stack (AWS AppStream, FSx for NetApp ONTAP, CloudWatch)
- Modularize code and adopt microservices



Intelligent Automation

- Automate tasks & data pipelines with modern orchestration
Implement CI/CD and Infrastructure-as-Code



Reverse Engineering & Docs

- Capture knowledge from source code systematically
- Build searchable, AI-assisted documentation



Success Strategy: Parallel modernization with gradual refactoring and comprehensive knowledge preservation — ensuring business continuity while building future-ready infrastructure.



Architecture of Legacy SCE

Citrix Infrastructure

- Multiple Citrix Servers for horizontal scaling
- Virtual Apps/Desktops for application delivery
- Session-based: each user gets a Citrix session

Application Deployment

- Macumba (SCE UI): pharma analytical interface on every server
- Uniform config: same application stack on all servers
- Resource pooling: users connect to any available server





Refactored SCE & Modernized Infrastructure

AWS AppStream 2.0

- Fleet: Auto-scaled streaming instances
- Stacks: Configuration templates for streaming sessions
- Images: Golden images with pre-configured apps

SCE Processing VPC

- Region: Frankfurt
- Multi-AZ: Dual zones for high availability

Compute Tier

- SAS 9.4 on FSx for NetApp ONTAP
- Samba file sharing (for Windows compatibility)
- Dedicated work storage volumes



Backups



Disaster Recovery



High Availability across AZ





Advantages of Modernized Infrastructure

Legacy (Citrix/On-Prem) vs Modern (AppStream/AWS)

Aspect	 Legacy (Citrix/On-Prem)	 Modern (AppStream/AWS)
 Scaling	Manual, fixed capacity	Auto-scaling, elastic
 Availability	Single region, limited HA	Multi-AZ, built-in redundancy
 User Experience	Citrix protocol, latency-sensitive	Streaming, adaptive to bandwidth
 Cost Model	Fixed costs (always-on VMs)	Flexible (scale up/down on-demand)
 Deployment	Manual server provisioning	AMI/Image-based, automated
 Geographic Reach	Centralized, potential latency	Regional deployment options
 Disaster Recovery	Not visible / limited	Multi-AZ with automated failover
 Maintenance	Updates across multiple servers	Golden image updates



Refactoring & Modernization Partner

Our partners in this journey

The Software Revolution Inc. (TSRI)



tsri.com

Expertise with legacy software modernization since 1995. Proven track record with programming language and industry-neutral automated transformation for government and commercial customers worldwide.



Established 1995



AI-based application transformation engine



Any programming language, size, or complexity can be handled



Successful transformations with government & commercial clients