

# From Monolith to Microservices:

Modernizing Architecture for a  
High-Performance Online Trading Platform

# about the client

Our client, a prominent online investment platform in **South Asia**, faced limitations with their existing technology infrastructure.

This impacted their ability to:

- handle increasing user traffic,
- offer a seamless user experience,
- and adapt to the evolving financial landscape.



# existing challenges



## Performance Bottlenecks

Sluggish platform performance led to frustrated users and impacted transaction volumes.

## Scalability Limitations

The monolithic architecture couldn't handle anticipated user growth, restricting company's expansion plans.

## Lack Of Agility

Slow development cycles impeded client's ability to introduce new features and innovations quickly.

## Maintenance Complexities

Outdated technologies & complex architecture made maintenance cumbersome and resource-intensive.

## Security Vulnerabilities

Potential security gaps posed risks to user data and financial transactions. (data breach)

# technical upgrades



# ● Moving To **Microservices Architecture**

We decomposed the monolithic architecture into agile, scalable microservices unlocking:

## **Scalability**

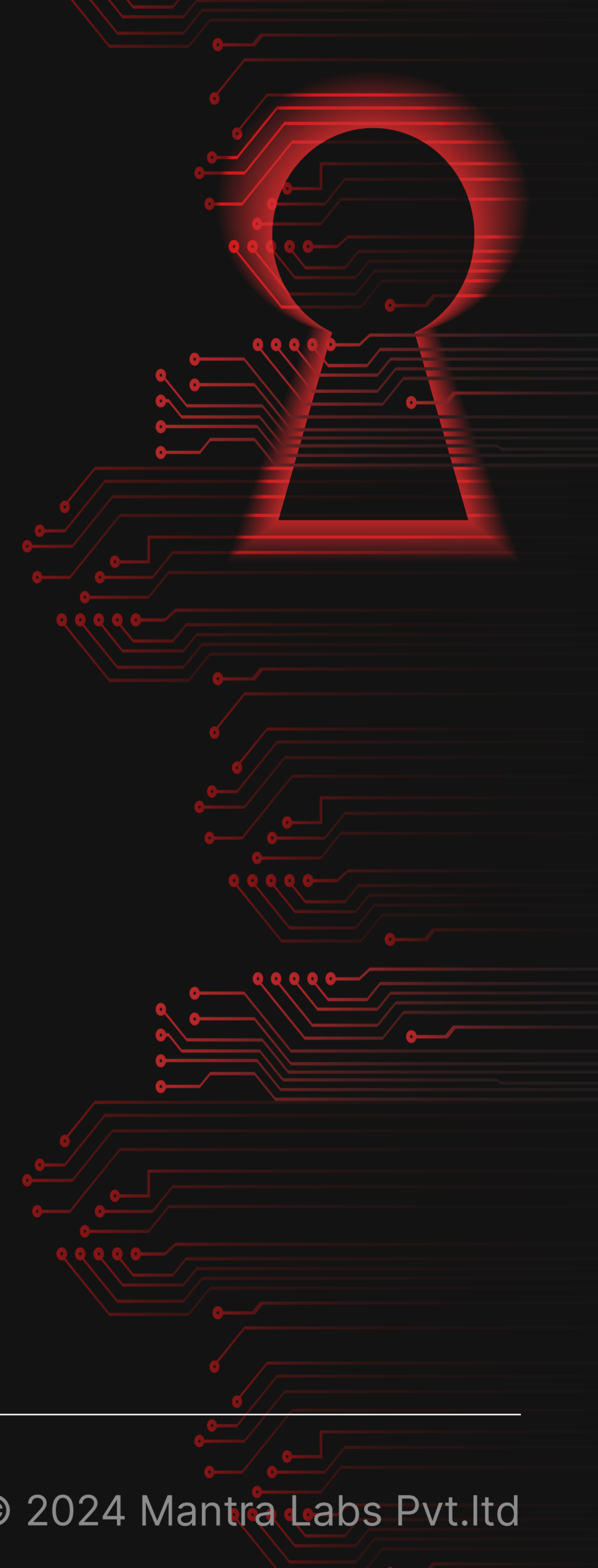
Individual microservices could be scaled independently based on real-time demand. This optimized resource allocation and reduced infrastructure costs.

## **Deployment Flexibility**

Enabled frictionless deployments of new features and updates with a continuous delivery pipeline for rapid iteration and market responsiveness.

## **Maintainability**

Slashed development overhead and empowered developers to make updates and fixes with lightning speed, boosting overall productivity.



# ● Containerization And Orchestration

**DOCKER**

Containerization Provided

## **Resource Utilization**

Optimized resource utilization by isolating applications and their dependencies.

## **Portability**

Unlocked seamless portability to deploy applications across different environments.

## **High Availability**

Automated scaling, and failover to ensure exceptional application uptime.

## **Efficiency**

Streamlined application lifecycle management for improving efficiency.

**KUBERNETES**

Management of containerized applications to ensure

- **Swagger Based Documentation**

Simplified API development, consumption, and maintenance.

- **Modernization From VB.Net To .Net Core**

This upgrade unleashed next-gen performance, modernized libraries and frameworks, and dramatically boosted maintainability for a future-proof application.

- **SP Optimization**

Stored Procedure Optimization and movement to middle tier wherever there are significant performance gains.

- **Solid Principles**

Adopting SOLID principles ensured a well-structured, modular, and reusable codebase, for scalability and future growth.

- **Cache Implementation**

To improve performance and reduce latency.

- **Setting Up Message Queuing**

To improve application scalability and decoupling.

- **CI/CD Pipeline Setup**

To deliver software faster with higher quality.

- **Code Quality Check Automation**

To streamline code quality checks for faster release cycles.

- **SonarQube Code Quality Setup**

For a more robust and maintainable codebase.

- **Moving To RHEL OS**

For stability and security.

- **Setting Up Unit Test Framework**

For early issue detection and code maintainability.

- **Moving To A New Web Server**

To unlock performance improvements and enhanced scalability.



# ● Cloud Migration To Azure

## Scalability On Demand

Cloud resources could be scaled up or down dynamically to meet fluctuating user needs.

## Cost Optimization

Pay-as-you-go model eliminated the need for upfront hardware investments.

## Advanced Security Features

Azure's built-in security service offered robust protection against cyber threats.

## Enhanced Disaster Recovery


Redundant data storage across geographically dispersed regions ensured business continuity.

# the outcome

## Performance Boosts

Significant improvement in page load times and transaction processing speed.

**18%**   
in User Abandonment

**21%**   
in Transaction Volume

## Maintenance Simplified

Reduced maintenance time and effort by **40%**, freeing up resources for innovation.

## Security

Security measures minimized vulnerabilities and ensured data protection.

## Improved Scalability

The upgraded platform is now able to handle a **50%** increase in user base and a **40%** growth in transactions without compromising performance.



# Building Intelligent Experiences That Matter



Key Domains

BFSI, Health Care, Consumer Internet

# our clients

STOCKX



abbvie

mLinkRx

Myntra

SBI general  
INSURANCE  
SURAKSHA AUR BHAROSA DONO

Hero  
INSURANCE BROKING



care  
HEALTH INSURANCE

sgtradex

ADITYA BIRLA  
CAPITAL

DIAGEO

ICICI Lombard  
Nibhaye Vaade

Shell

IIFL  
FINANCE

ALKEM

Canara Bank HSBC  
LIFE INSURANCE

edureka!

CHANGE  
HEALTHCARE

TATA AIA  
LIFE INSURANCE

ManipalHospitals  
LIFE'S ON

PATHOMIQ

ageasFEDERAL  
LIFE INSURANCE

OLA

thank **you**

☎ +1-872-362-0414  
✉ [parag@mantralabsglobal.com](mailto:parag@mantralabsglobal.com)  
📍 USA | North Carolina

☎ +91 987- 033- 3426  
✉ [hello@mantralabsglobal.com](mailto:hello@mantralabsglobal.com)  
📍 India | Bengaluru | Kolkata

---