



# Customer Case Study

*A financial services solution provider in anti-money laundering space had its product deployed in banks and other financial service institutions. The application, hitherto used in other segments, was actively selling to the banking industry. This needed the application to be highly available and scalable; for this, it had to be clustered and optimized for performance. MiddlewareWorks was engaged to review the application for clusterability and performance.*

## Key Findings

- Engineers from MiddlewareWorks Technology group reviewed design and code from a clusterability point of view - both high-availability and failover - including architecture, detailed design, code walkthrough, and configuration.
- The application used extensive information in Web sessions, which had assumptions that prevented effective failover of sessions in a cluster. The application was studied for non-serializable objects in sessions and for typical size of session objects. Code changes were identified to enable session failover.
- Clusterability had problems in the nature of data cached in Web sessions. Analyzing product architecture showed cache and data structures that were not cluster safe - caches also assumed single global instance. Fundamental design changes were needed to undo this assumption and make a cluster safe coherent cache.
- Clusterability was also challenged by a wide use of JMS that had to be clusterable. Complex global initialization operations were involved, some to be initialized in each cluster node, and some only once for all nodes.
- A review of software configuration management showed fundamental problems in the build process, modularity in application code, packaging, and build automation. The modules were not optimally organized - a single EJB per module in most cases.
- A review of low-level design of modules and implementation revealed inadequacies in coding standards, code readability, and use of utilities. Full functionality of frameworks like Struts was underutilized, and exception handling and propagation of exception messages was minimally used.

## Clusterability

# Review app for Clusterability & Performance

### Recommendations:

- A multi-tier cluster with at least two nodes of J2EE server, one node JMS Server, and a load balancer configured with session stickiness is required along with a solution using subtle messaging semantics. This includes re-engineering a major module to be moved to the server from a fat-client.
- Reorganize software configuration management - based on a functional organization that groups the UI and business modules of the application by application functional module.
- A build process with exception handling, logging, and tracing capabilities. Implement Util package creation along with proper code documentation and automate sanity testing (for verifying regression). Restructure and modify application for source modules, modularity, and enabling Clustering failover.
- Identify the test cases and generate tests. Modify the logging framework to handle different logging levels. The Struts framework should also be used. The Security implementation should be AppServer independent.

MiddlewareWorks is a business unit of Pramati Technologies, the leading applications server and tools company that has been building products since 1998. Pramati is well known for its Java™ application server (J2EE™ platform) and several integrated tools products. MiddlewareWorks provides high quality and practical technical consulting and services to corporate clients that are using middleware. You get the technical edge that is needed to successfully deploy applications for business benefits without risks, productivity-loss or high spending. MiddlewareWorks does everything required to make middleware work for you. Customers includes startups, venture funded companies, multi-billion dollar giants and some of the best names in the technology world. Visit <http://www.middlewareworks.com>.