# Specification Document: AI-Powered Trading System for MetaTrader 4 (MT4) & Java AI Server

## 1. Overview

This document defines the technical specifications for the AI-powered trading system, which integrates MetaTrader 4 (MT4) with a Java-based AI trading server via ZeroMQ. The system enables automated trading based on AI-generated signals.

### 1.1 Objective

• Automate trading using AI-based market predictions.  
• Provide seamless communication between MT4 and the AI trading server.  
• Improve trade execution efficiency using ZeroMQ.

### 1.2 Scope

• Real-time market data exchange between MT4 and the AI server.  
• AI-based prediction models using LSTM and reinforcement learning.  
• Secure and optimized trade execution in MT4.

## 2. System Components

### 2.1 MetaTrader 4 (MT4) Expert Advisor (EA)

• Language: MQL4  
• Functionality:  
 - Collects market data (OHLC, Volume, Spread, etc.).  
 - Sends data to the AI Trading Server via ZeroMQ.  
 - Receives AI-generated trade signals.  
 - Executes trades automatically.

### 2.2 Java AI Trading Server

• Language: Java  
• Dependencies:  
 - ZeroMQ (JeroMQ) for communication.  
 - Deep Java Library (DJL) for deep learning models.  
 - RL4J (DeepLearning4J) for reinforcement learning.  
• Functionality:  
 - Listens for market data from MT4.  
 - Uses an LSTM model to predict price trends.  
 - Uses reinforcement learning to optimize trading strategies.  
 - Sends BUY/SELL signals back to MT4.

## 3. Functional Requirements

3.1 Market Data Processing  
• The MT4 EA must send the following market data to the AI server:  
 - Symbol  
 - Open Price  
 - High Price  
 - Low Price  
 - Close Price  
 - Volume

3.2 AI Signal Generation  
• The AI server must:  
 - Analyze the received market data.  
 - Predict BUY or SELL trade signals using LSTM.  
 - Optimize trade decisions using reinforcement learning.  
 - Respond with a JSON message containing the trading signal.

3.3 Trade Execution  
• The MT4 EA must:  
 - Receive AI-generated trade signals.  
 - Execute BUY or SELL orders based on received signals.  
 - Implement Stop-Loss and Take-Profit mechanisms.

## 4. Non-Functional Requirements

|  |  |
| --- | --- |
| Requirement | Description |
| Performance | System must process and execute trades within 1 second. |
| Scalability | Must support multiple currency pairs and multiple trading accounts. |
| Security | Ensure encrypted communication between MT4 and AI Server. |
| Fault Tolerance | System should recover gracefully from network failures. |
| Logging & Monitoring | Store logs of all trades, signals, and errors for debugging. |

## 5. Data Exchange Format

Market Data Sent from MT4 to AI Server:  
{  
 "Symbol": "EURUSD",  
 "Open": 1.1050,  
 "High": 1.1070,  
 "Low": 1.1030,  
 "Close": 1.1060,  
 "Volume": 100000  
}

AI Server Response (Trade Signal):  
{  
 "signal": "BUY"  
}

## 6. System Constraints

• MT4 must be running continuously for real-time trading.  
• AI Server must be online to process market data and return signals.  
• ZeroMQ connection must be stable to avoid trade execution delays.

## 7. Error Handling & Debugging

|  |  |
| --- | --- |
| Error | Solution |
| libzmq.dll not found | Ensure libzmq.dll is in MQL4/Libraries/, then restart MT4. |
| No response from AI Server | Verify Java server is running with mvn exec:java. |
| Invalid JSON Format | Check Java server logs to ensure correct response format. |

## 8. Future Enhancements

✅ Add AI-based Stop-Loss & Take-Profit optimization  
✅ Enable multi-symbol trading  
✅ Deploy AI models on cloud servers for live trading