Tic Tac Toe Project Interface Specifications

SIO3A11 – SYSTEMS ENGINEERING AND DESIGN 3A

LA LUKUMBI (219046398), M SUKNANDAN (219096158), NJL MWANZA (200904709), T KOMANA (219021786), KB SEBILOANE (219003346), K KANHUKAMWE (219104385), S DUNGA (219002664), G VENGATSAMMY (201585904)

UNIVERSITY OF JOHANNESBURG

30 MARCH 2021

Table of Contents

1. INTRODUCTION

1.1. PURPOSE

The interface specification document has the aim to provide the necessary information on the different interactions that are to happen between all four sub-systems. It describes to some extent the type of information that should travel from one sub-system to the other as well as from one system to others. We specify the minimum interfacing requirements that are expected to be implemented to each individual subsystem first and then to the system as a whole to allow a smooth and coordinated input/output transmission.

1.2. SCOPE

The Interface Specifications Team shall send information between the different subsystems for the Tic Tac Toe system. The Interface team is responsible for implementing the subsystems on Arduino via all communication through the RS-232,5V TTL serial interface. The presence of an opponent in the system shall be detected by reading a discrete signal and all message structures and input/output buffers shall be specified for all interfaces between the subsystems.
The document shall cover the system perspective by explaining how the Interface Specifications team will interface with the subsystems and the requirements specifications (communications interface). The architectural diagram shall be included and elaborated on later in the document as well as the systems functional and non-functional requirements.
The Interface Specifications limitations, assumptions, hardware and software requirements shall be specified as well.

2. SYSTEM PERSPECTIVE

The various interfaces involved within this specification document occur between the learner sub-system, the mover sub-system, the translator sub-system, the player sub-system and the opponent systems.
The next four sections describe what interface interactions should be expected from each subsystem.

2.1. SYSTEM REQUIREMENT SPECIFICATIONS

2.1.1. Player Sub-system

Within its own system:
SRS1: The player subsystem shall be able to interface with the translator sub-system.
SRS2: The player subsystem shall be able to interface with the learner sub-system.
Outside its own system:
SRS3: The player subsystem shall be able to interface with the player sub-system of other systems.

2.1.2. Translator Sub-system

SRS4: The translator subsystem shall be able to interface with the mover sub-system.

SRS5: The translator subsystem shall be able to interface with the player sub-system.

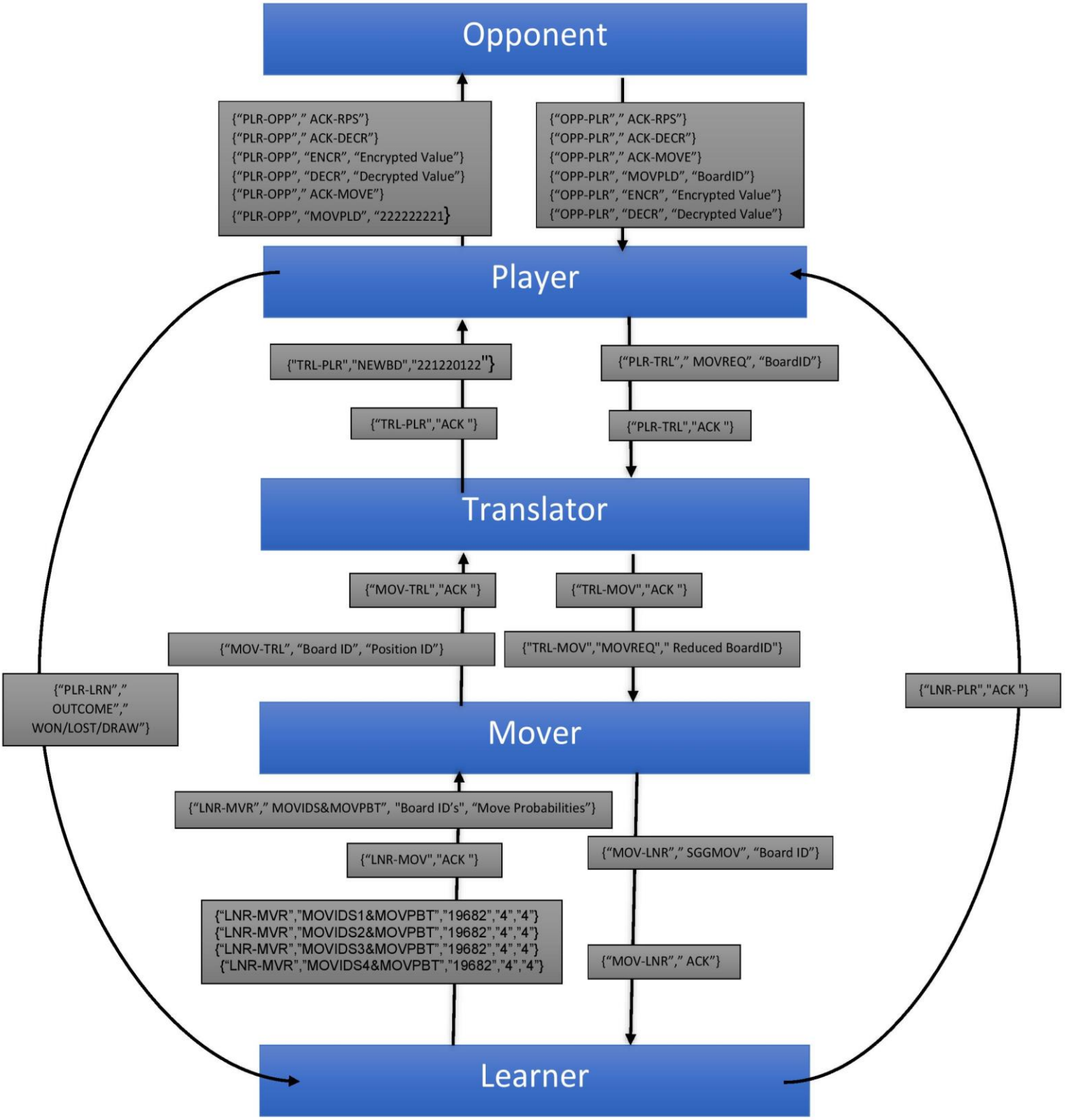
2.1.3. Mover Sub-system

SRS6: The mover subsystem shall be able to interface with the translator sub-system.
SRS7: The mover subsystem shall be able to interface with the learner sub-system.

2.1.4. Learner Sub-system

SRS8: The learner subsystem shall be able to interface with the mover sub-system.
SRS9: The learner subsystem shall be able to interface with the player sub-system.

3. ARCHITECTURE DIAGRAM



4. FUNCTIONAL REQUIREMENTS

FR1: The Player sub-system shall be able to transmit the outcome of the Tic Tac Toe game with the Learner Sub-system

FR2: The Player Sub-system shall be able to transmit a request for an updated board ID from the Translator Sub-system.

FR3: ThePlayer Sub-system shall be able to receive an updated board ID from the Translator Sub-system

FR4: The Player Sub-system shall be able to receive an encrypted Rock, Paper, Scissors value from the Opponent system

FR5: The Player Sub-system shall be able to receive a decryption value for the Rock, Paper, Scissors game from the Opponent System

FR6: The Player Sub-system shall be able to transmit an encrypted Rock, Paper, Scissors value to the Opponent System

FR7: The Player Sub-system shall be able to transmit a decryption value for the Rock, Paper, Scissors game to the Opponent System

FR8: The Player Sub-system shall be able to receive acknowledgment messages from the Learner and Translator Sub-systems and the Opponent System.

FR9: The Player Sub-system shall be able to send acknowledgement messages of receipt of information to the Learner and Translator Sub-systems and the Opponent System.

FR10: The Player Subsystem shall be able to receive the move played by the opponent from the Opponent System

FR11: The Player Subsystem shall be able to transmit the move it played to the Opponent system

FR12: The Translator Sub-system shall be able to receive a request for an updated board ID from the Player Sub-system

FR13: The Translator Sub-system shall be able to send a reduced board ID to the Mover Sub-system.

FR14: The Translator Sub-system shall be able to receive a reduced board ID and a positionID from the Mover Sub-system

FR15: The Translator Sub-system shall be able to send an updated board ID containing a new move to the Player Sub-system

FR16: The Translator Sub-system shall be able to send acknowledgement messages of receipt of information to the Player and Mover Sub-systems.

FR17: The Translator Sub-system shall be able to receive acknowledgment messages from the Player and Mover Sub-systems.

FR18: The Mover Sub-system shall be able to receive a request for an updated reduced board ID which contains a new move from the Translator Sub-system.

FR19: The Mover Sub-system shall be able to send a reduced board ID and a positionID to the Translator Sub-system.

FR20: The Mover Sub-system shall be able to receive the updated move probability from the Learner Sub-system.

FR21: The Mover Sub-system shall be able to send a request for the updated move probability to the Learner Sub-system

FR22: The Mover Sub-system shall be able to receive acknowledgement messages of receipt of information from the Learner and Translator Sub-systems.

FR23: The Mover Sub-system shall be able to send acknowledgment messages of receipt of information to the Learner and Translator Sub-systems

FR24: The Learner Sub-system shall be able to receive the outcome of the Tic Tac Toe game from the Player Sub-system

FR25: The Learner Sub-system shall be able to send the updated move probability to the Mover Sub-system.

FR26: The Learner Sub-system shall be able to receive a request for the updated move probability from the Mover Sub-system.

FR27: The Learner Sub-system shall be able to receive acknowledgement messages of receipt of information from the Player and Mover Sub-systems.

FR28: The Learner Sub-system shall be able to send acknowledgment messages of receipt of information to the Player and Mover Sub-systems.

5. INTERFACE REQUIREMENTS

5.1. PLAYER SUB-SYSTEM (PLR)

5.1.1. Inputs

[FR8]: MESSAGE FROM TRANSLATOR to PLAYER (Acknowledgement of receipt of request of new move):
{"Message header", " Message type"**}**
{"TRL-PLR","ACK "}

[FR3]: MESSAGE FROM TRANSLATOR to PLAYER (Sending updated board with new move):
{"Message header "," Message type", "Board ID"**}**
{"TRL-PLR","NEWBD","221220122"}

[FR8]: MESSAGE FROM LEARNER TO PLAYER (Acknowledgement of outcome):
{"Message header", " Message type "**}**
{"LNR-PLR", "ACK-OUTC"}

[FR8]: MESSAGE FROM OPPONENT TO PLAYER (Acknowledgment of receipt of encrypted Rock, Paper, Scissors value)
{“message header”, “message type”**}**
{“OPP-PLR”,” ACK-RPS”}

[FR8]: MESSAGE FROM OPPONENT TO PLAYER (Acknowledgment of receipt of decryption value)
{“message header”, “message type”**}**
{“OPP-PLR”,” ACK-DECR”}

[FR8]: MESSAGE FROM OPPONENT TO PLAYER (Acknowledgment of receipt of move played by the player)
{“message header”, “message type”**}**
{“OPP-PLR”,” ACK-MOVE”}

[FR10]: MESSAGE FROM OPPONENT TO PLAYER (Sending new move played)
{“Message header”, “message type”, “Board ID”**}**
{“OPP-PLR”, “MOVPLD”, “222222220}

[FR4]: MESSAGE FROM OPPONENT TO PLAYER (Sending encrypted Rock, Paper, Scissors value)
{“Message header”, “message type”, “Encrypted value”**}**
{“OPP-PLR”, “ENCR”, “15”}

[FR5]: MESSAGE FROM OPPONENT TO PLAYER (Sending decryption value for Rock, Paper, Scissors)

{“Message header”, “message type”, “Encrypted value”}
{“OPP-PLR”, “DECR”, “5”}

5.1.2. Outputs

[FR2]: MESSAGE FROM PLAYER to TRANSLATOR (Requesting move):
{"Message header "," Message type”, “Board ID"**}**
{“PLR-TRL”,” MOVREQ”, “222222222”}

[FR10]: MESSAGE FROM PLAYER to TRANSLATOR (Acknowledgement of receipt of new board ID with new move)
{“Message header”, "message type"**}**
{“PLR-TRL","ACK "}

[FR1]: MESSAGE FROM PLAYER to LEARNER (Sending the outcome of the game)
{“Message header”, "message type", “Outcome”**}**
{“PLR-LRN”,” OUTCOME”,” WON/LOST/DRAW”}

[FR9]: MESSAGE FROM PLAYER TO OPPONENT (Acknowledgment of receipt of encrypted Rock, Paper, Scissors value)
{“message header”, “message type”}
{“PLR-OPP”,” ACK-RPS”}

[FR9]: MESSAGE FROM PLAYER TO OPPONENT (Acknowledgment of receipt of decryption value)
{“message header”, “message type”}
{“PLR-OPP”,” ACK-DECR”}

[FR9]: MESSAGE FROM PLAYER TO OPPONENT (Acknowledgment of receipt of move played by the opponent)
{“message header”, “message type”}
{“PLR-OPP”,” ACK-MOVE”}

MESSAGE FROM PLAYER TO OPPONENT (Send new move played [FR11])
{“Message header”, “message type”, “Board ID”}
{“PLR-OPP”, “MOVPLD”, “222222221}

[FR6]: MESSAGE FROM PLAYER TO OPPONENT (Sending encrypted Rock, Paper, Scissors value)
{“Message header”, “message type”, “Encrypted value”}
{“PLR-OPP”, “ENCR”, “15”}

[FR7]: MESSAGE FROM PLAYER TO OPPONENT (Sending decryption value for Rock, Paper, Scissors)
{“Message header”, “message type”, “Encrypted value”}

{“PLR-OPP”, “DECR”, “5”}

5.2. TRANSLATOR SUB-SYSTEM (TRL)

5.2.1. Inputs

[FR12]: MESSAGE FROM PLAYER to TRANSLATOR (Requesting move):
{"Message header "," Message type”, “Board ID"**}**
{“PLR-TRL”,” MOVREQ”, “222222222”}

[FR17]: MESSAGE FROM MOVER to TRANSLATOR (Acknowledging receipt of request):
{"Message header", " Message type "**}**
{“MOV-TRL","ACK "}

[FR14]: MESSAGE FROM MOVER to TRANSLATOR (Reduced board ID and position ID):
{"Message header", "Message type”, “Reduced board ID”, “Position ID”}
{"MOV-TRL",” SUGMOV”,"19682",”9”}

[FR17]: MESSAGE FROM PLAYER to TRANSLATOR (Acknowledgement of receipt of new board ID with new move)
{"Message header", " Message type "**}**
{"PLR-TRL","ACK "}

5.2.2. Outputs

[FR16]: MESSAGE FROM TRANSLATOR to PLAYER (Acknowledgement of request to get a new move):
{"Message header”, " Message type"**}**
{"TRL-PLR","ACK "}

[FR13]: MESSAGE FROM TRANSLATOR to MOVER:(sending reduced board ID for a new move)
{"Message header", "Message type”, “Reduced board ID"}
{"TRL-MOV","MOVREQ","19683"}

[FR16]: MESSAGE FROM TRANSLATOR TO MOVER:(Acknowledgement of request to send new move to Player)
{“Message header”, " Message type "**}**
{“TRL-MOV","ACK "}

[FR15]: MESSAGE FROM TRANSLATOR to PLAYER (Sending updated board with new move):
{"Message header "," Message type”, “Board ID"**}**
{“TRL-PLR","NEWBD","221220122"}

5.3. MOVER SUB-SYSTEM (MOV)

5.3.1. Inputs

[FR18]: MESSAGE FROM TRANSLATOR to MOVER:(sending reduced board ID for a new move)
{"Message header", "Message type", "Reduced board ID"}
{"TRL-MOV","MOVREQ","19683",}

[FR22]: MESSAGE FROM TRANSLATOR TO MOVER:(Acknowledgement of request to send new move to Player)
{"Message header", " Message type "**}**
{"TRL-MOV","ACK "}

[FR20]: MESSAGE FROM LEARNER to MOVER (Updated Move Probabilities):
{"Message header "," Message type", "Board ID's", "Position ID", "Move Probabilities"**}**
{"LNR-MVR","MOVIDS1&MOVPBT","19682","4","4"}
{"LNR-MVR","MOVIDS2&MOVPBT","19682","4","4"}
{"LNR-MVR","MOVIDS3&MOVPBT","19682","4","4"}
{"LNR-MVR","MOVIDS4&MOVPBT","19682","4","4"}

[FR22]: MESSAGE FROM LEARNER to MOVER (Acknowledgement of reception of moves ID and move probability):
{"Message header", "Message type"**}**
{"LNR-MOV","ACK "}

[FR22]: MESSAGE FROM LEARNER to MOVER (Acknowledgement of reception of outcome of the game):
{"Message header", "Message type"**}**
{"LNR-PLR","ACK "}

5.3.2. Outputs

[FR19]: MESSAGE FROM MOVER TO TRANSLATOR (suggesting a move):
{"Message header", "Message type", "Reduced Board ID", "Position ID"}
{"MOV-TRL"," SUGMOV", "19682","9"}

[FR23]: MESSAGE FROM MOVER TO TRANSLATOR (acknowledging a request):
{"Message Header", "Message Type", "Content"}
{"MOV-TRL"," ACK", "ACKNOWLEDGE"}

[FR21]: MESSAGE FROM MOVER TO LEARNER (for suggested move):
{"Message header "," Message type", "Board ID's", "Position ID", "Updated Probabilities"**}**
{"LNR-MVR"," MOVIDS1&MOVPBT","19682","4","4"}

[FR23]: MESSAGE FROM MOVER TO LEARNER (acknowledging updated probability request):
{“Message Header”, “Message type”, “Displayed Content”}
{“MOV-LNR”,”ACK”, “Received update”}

5.4. LEARNER SUB-SYSTEM (LNR)

5.4.1. Inputs

[FR24]: MESSAGE FROM PLAYER to LEARNER (End game status):
{"Message header "," Message type”, “Board ID"**}**
{“PLR-LNR”,” OUTCOME”, “WON/LOST/DRAW”}

[FR26]: MESSAGE FROM MOVER to LEARNER (Moves made):
{"Message header "," Message type", "Board ID’s”, “Position ID”, “Updated Probabilities”**}**
{“LNR-MVR”,”MOVIDS1&MOVPBT”,”19682”,”4”,”4”}

[FR27]: MESSAGE FROM MOVER TO LEARNER (Acknowledgement of reception of updated move probabilities):
{ "Message header" , "Message type"**}**
{ "MOV-LNR","ACK "}

5.4.2. Outputs

[FR25]: MESSAGE FROM LEARNER to MOVER (Updated move probabilities):
{"Message header "," Message type", "Board ID’s”, “Position ID”, “Updated Probabilities”**}**
{“LNR-MVR”,”MOVIDS1&MOVPBT”,”19682”,”4”,”4”}
{“LNR-MVR”,”MOVIDS2&MOVPBT”,”19682”,”4”,”4”}
{“LNR-MVR”,”MOVIDS3&MOVPBT”,”19682”,”4”,”4”}
{“LNR-MVR”,”MOVIDS4&MOVPBT”,”19682”,”4”,”4”}

[FR28]: MESSAGE FROM LEARNER to MOVER (Acknowledgement of reception of moves ID and move probability):
{ "Message header" , "Message type"**}**
{ "LNR-MOV","ACK "}

[FR28]: MESSAGE FROM LEARNER to PLAYER (Acknowledgement of reception of outcome of the game):
{"Message header" , "Message type"**}**
{ "LNR-PLR","ACK "}

6. DESIGN AND CONSTRUCTION CONSTRAINTS REQUIREMENTS

DCR1: Each individual sub-system shall make use of an Arduino Uno for its construction

DCR2: The programming language used shall be limited to one that can be interpreted by the Arduino.

DCR3: Communication via the interfaces between sub-systems shall occur via the serial communication ports

DCR4: Serial communication is limited by the number of serial ports available: ATmega328 provides two digital I/O pins to receive and transmit serial data; the 12 other digital pins allow for serial communication using the SoftwareSerial library

DCR5: Sampling rate is limited by the Arduino Uno's 10 bit resolution and maximum internal clock-speed of 16MHz

DCR6: The construction of the circuit shall be limited to component availability

DCR7: Each sub-system shall be limited to being powered by a 5V source

7. CHARACTERISTIC REQUIREMENTS

NFR1: Reliability - The system shall be able to reliably be able to transmit messages from one subsystem to another.

NFR2: Correctness of Transmission - The system shall be able to send the correct message to the correct sub-system, every time.

NFR3: Correctness of Contents - The system shall be able to keep the integrity of the information it carries during transmission, no information may be lost or garbled during transmission.

NFR4: Error Detection - The system shall be able to detect when information has been transformed during transmission, and be able to relay this information to the relevant/receiving subsystem

NFR5: Speed - The system shall not take longer than 500ms to transmit any given message to any given subsystem.

NFR6: Lightweight - the protocol shall not be "heavy" for any particular subsystem to use (ie. should not do many CPU intensive calculations)

NFR7: Hardware Agnostic - The Protocol shall be hardware agnostic in the sense that it shall not depend on any Arduino specific functionality in order to function properly. In theory it shall be able to work even on top of other protocols like http/https.

NFR8: Debuggable - The Subsystem shall be able to report its status at any given point during the transmission process, to give meaningful indication about the status of said point in the transmission process, thus generating meaningful debug messages.

NFR9: Traceability - All messages coming in and out of the system shall be traceable to a source and a destination, allowing the system in theory to be able to keep a transaction ledger (assists in the debugging process)