LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

1.0 TOPIC TITLE:

Creating Niagara Files Processors for Transforming Legacy Data into XML

2.0 SUMMARY:

The effort seeks an Apache Niagara Files processor capable of parsing a legacy ASCII-based data format into a simple XML structure. The developed solution will rely on data definition files that identify the contents of each message and the output must be customizable via user-defined configuration files. After the parser and processor have been developed, performance metrics must be supplied.

3.0 BACKGROUND:

MIT Lincoln Laboratory's Intelligence and Decision Technologies Group (Group 104) is supporting the US Air Force to realize its vision of an Open Architecture information infrastructure that reaches across all levels of its data enterprise from sensors to platforms, command and control, and intelligence surveillance and reconnaissance (ISR). In this context, an open architecture has many benefits. It allows for rapid deployment of capabilities vs. large infrequent software upgrades; it reduces costs due to leveraging of open source software components, and it lowers barrier to entry, which empowers non-traditional DoD contractors to provide best-of-breed solutions. This is achieved in part by utilizing an event-driven service oriented architecture (SOA) based on a standard XML message format.

MIT Lincoln Laboratory (MIT LL) builds and deploys software systems, also known as software testbeds, to sites in support of program based events and demonstrations. These software systems regularly have to connect to and authenticate with a variety of data sources, subscribe to data being published, and route the data to specific endpoints for consumption. In some cases the data being subscribed to is of a format that the endpoint can’t consume and, as a result, the testbed must translate the data for the consumer.

Current testbeds deployed by MIT LL utilize an in-house Apache Tomcat web application called the Realtime Enhanced Situation Awareness (RESA) system to build event chains to accomplish the subscription, transformation, and routing tasks. Our goal is to leverage Niagara Files (NiFi) for these tasks by referencing our existing transformation code where possible, writing new parsers when necessary, and utilizing available open-source plugins where applicable (e.g. for ActiveMQ connections). The desired end result would utilize Niagara Files’ visual drag-and-drop web-based interface to create the data flows needed for our testbed deployments.

Additional concept ideas that support this challenge are also welcome.