



Justice/Law Enforcement Solution

Management Summary & Technical Overview

Version 1



The Stahura-Brenner Group, Inc.
5405 Alton Parkway, 5-A #359
Irvine, CA 92604
(949) 733-8526

Copyright

The programs and concepts mentioned herein are proprietary to The Stahura-Brenner Group, Inc., and are not to be reproduced, used or disclosed except in accordance with program license or upon written authorization by an officer of The Stahura-Brenner Group, Inc.

Copyright © 2014-2015 All Rights Reserved.

.NET®, Windows®, are registered trademarks of Microsoft Corporation.

Nlets® is a registered trademark of National Law Enforcement Telecommunication Service, Inc.

Management Summary

The main business requirement for this project was to interface a large State Agency's legacy application to several federal law enforcement systems as well as to the International Justice and Public Safety Network (Nlets). These federal systems included the National Crime Information Center (NCIC), the Interstate Identification Index (III), and the National Instant Criminal Background Check System (NICS). Modifications to the State Agency's existing application were to be kept to a minimum. As expected, security was of utmost importance.

Backstory

- Law Enforcement relies on the Nlets network for the communication of law enforcement-related messages between States.
- Law Enforcement relies on the NCIC network for access to federal law enforcement-related databases.
- Both communicate directly with State-level computing systems, which, in turn, communicate with local agencies.
- Some mainframe-based State-level computing systems operate on legacy mainframes that do not easily support the communications protocols required by Nlets and NCIC.

Business Objective

- Hide all technical interface complexities from existing back-end applications

Functional Requirements

- Support Hot File requests (stolen vehicles, wanted persons, missing persons) to the NCIC database
- Support III requests routed through NCIC
- Support III responses returned by Nlets
- Support National Instant Criminal Background Checks by firearms sellers routed to NCIC
- Support interstate queries and messaging such as:
 - Driver's license queries
 - Vehicle queries
 - LoJack® stolen vehicle recovery system notifications
 - AMBER alerts
 - Stolen vehicle feed from NCIC
- Recognize messages indicating when Hot Files, III, and NICS statuses change (online/offline)
 - Delay message submission when offline
 - Submit delayed messages when online
- Provide ultra-high availability—messages include information that impacts the safety of law enforcement officers in the field

Technical Requirements

- Provide interfaces to Nlets and NCIC using IBM WebSphere® MQ
- Support asynchronous and synchronous request/response handling
- Provide reliable delivery of asynchronous messages
- Audit all message transmissions as required by law

Outcome

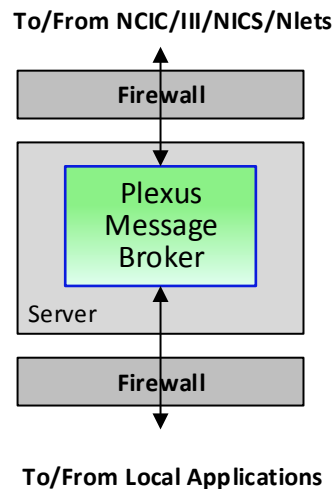
By the development of a number of Filters, the Plexus Message Broker was enhanced to provide the necessary interfaces to Nlets, NCIC, III, and NICS. Automatic starting/stopping of message routing to NCIC and NICS was automatically sensed and handled. Furthermore, the complexity of interfacing to these systems was kept hidden from the back-end application systems.

Technical Overview

This Plexus Message Broker solution features the following technology: WebSphere® MQ; synchronous and asynchronous messaging; flow control; and complex routing. This solution's site and Plexus Message Broker configurations touch on these features below.

Site Configuration

As illustrated below, this site has a single physical server that contains a single instance of the Plexus Message Broker. The site maintains a hot standby of the physical server.

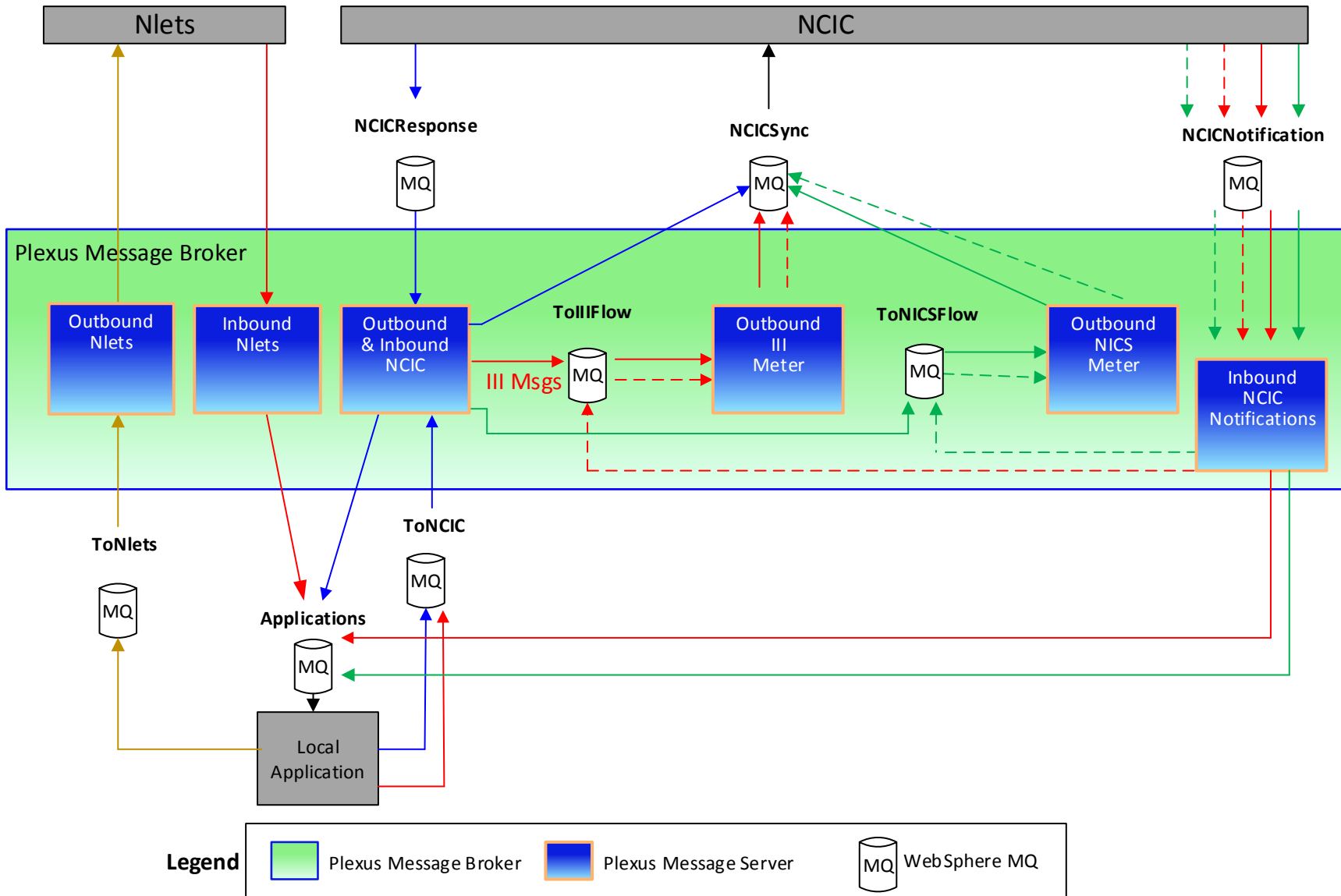


As with all Plexus Message Broker installations, adding physical servers, increasing Plexus Message Brokers instances per physical server, or transitioning to run the Plexus Message Broker in a virtual machine (VM) environment is only a matter of installation and configuration. No other Plexus Message Broker modifications are necessary. Resources are typically added for performance reasons, local redundancy reasons, or both.

At this particular site, the response time for request and responses is not hypercritical. Moreover, the IT procedure to hot swap the stand-by system is short enough to meet the business needs for response time. Hence, this very simple configuration is suitable.

Plexus Message Broker Configuration

As illustrated below, this Plexus Message Broker has sets of messages servers as well as individual messages servers to achieve the desired communications. There is an inbound message server and an outbound message server for Nlets; whereas for NCIC the same message server processes both inbound and outbound traffic. Then there are individual messages servers for NCIC unsolicited notification traffic, III metered traffic and NICS metered traffic. The latter two messages servers are the same type but with different configurations.

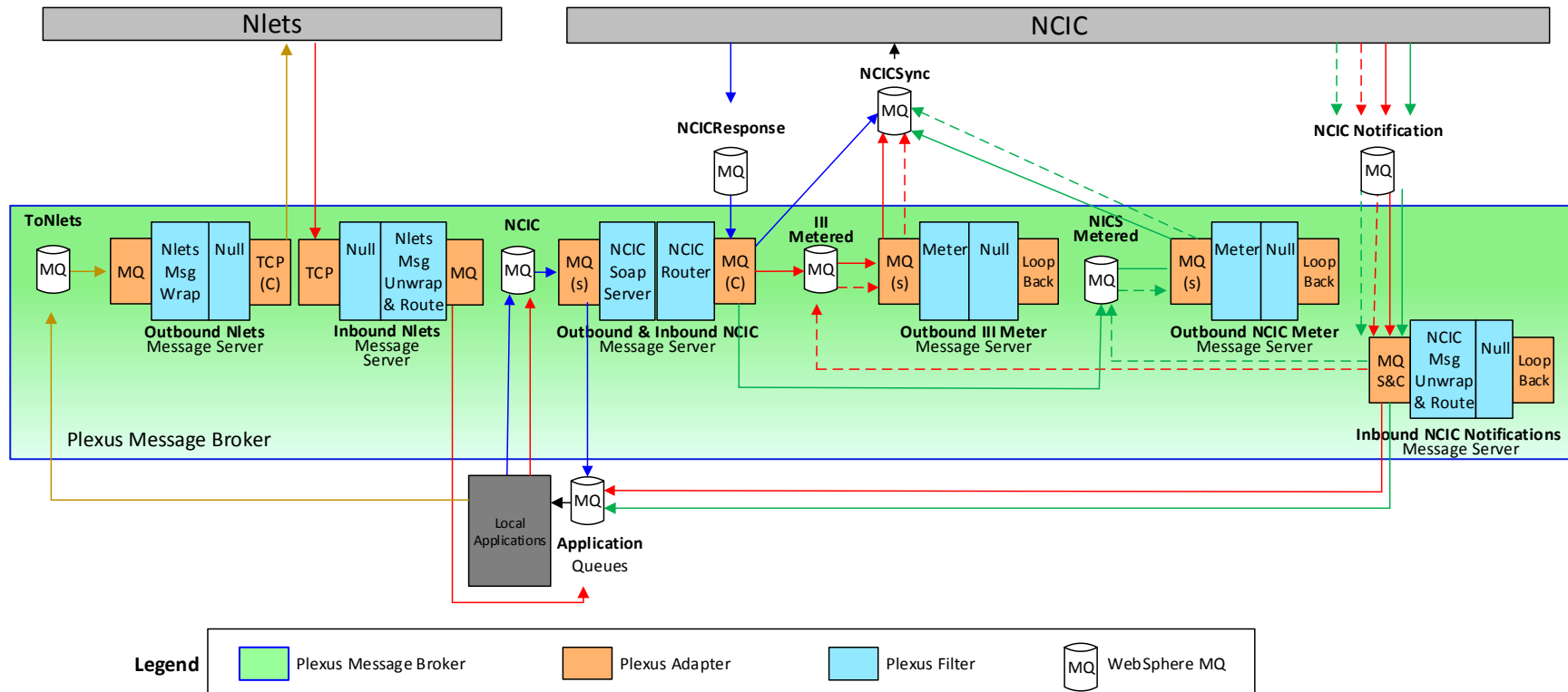


Plexus Message Server Functions

Message Server	Function
Outbound Nlets	Removes MQ message from queue; unwraps message, opens a TCP session with Nlets using a proprietary TCP protocol, and sends the message to Nlets. The Inbound Nlets Message Server handles the response message that is returned on a separate TCP session and sent to Nlets via a proprietary TCP protocol.
Inbound Nlets	Receives message from Nlets over a proprietary TCP protocol, determines which application should receive the message, wraps the message in an MQ message, and inserts the MQ message in the appropriate application queue
Outbound & Inbound NCIC	Receives messages from local applications in the ToNCIC queue and sends the messages to NCIC via the NCICSync queue at NCIC. Response messages are received in the NCICResponse queue and then are unwrapped and inserted into the appropriate application queue.
Outbound III Meter	Sends ping (test) message to III on a periodic basis to determine if the III service is active. If III is active, a III message is removed from the ToIIIFlow queue and sent to III via the NCICSync queue. The response to this message is received in the NCICNotifiation Queue which is managed by the InboundNCICNotification Message Server. If III is not active, messages are queued in the ToIIIFlow queue until the III service becomes active.
Outbound NICS Meter	Similar to Outbound III Meter Message Server except for the messages being routed to NICS instead
Inbound NCIC Notifications	Removes message from NCICNotification queue, determines which application should receive the response, and deposits the message in that applications queue

Plexus Message Server Details

The following figure drills down into the configuration of the Plexus Message Broker; in particular it provides insight into the Adapters and Filters used by each Plexus Message Server in this solution.



Adapters

Adapter	Description
MQ	Provides send/receive services for WebSphere® MQ
TCP(c)	Provides TCP connect services where the Adapter is acting as a TCP client
TCP(s)	Provides TCP connection services where the Adapter is acting as a TCP server

Filters

Filter	Description
Nlets Msg Wrap	Takes a message and wraps the message in an Nlets Web Services SOAP header
Null	An empty filter that is used as a placeholder when no specific filter is needed
Nlets Msg Unwrap & Route	Unwraps a message from an Nlets SOAP header, determines the message routing based on internal data, and routes the message to the particular application message queue
NCIC Soap Server	Wrap or unwraps a message in/from a NCIC SOAP header depending on the direction of the message
NCIC Router	Routes inbound messages to the particular application destination queue based on internal message data; routes the outbound messages to the NCICSync queue, the III Meter Queue, or the NICS queue depending on the message.
Meter	Ensures that metered messages do not flood the NCICSync queue. Also sends test messages to III and NICS to determine if these services are up and running. If not, these metered messages are queued until the particular service comes up.