CJIS Broker
Overview

Version 2.0 – 07/20/2015

THE COMMONWEALTH OF MASSACHUSETTS
EXECUTIVE OFFICE OF PUBLIC SAFETY AND SECURITY
Department of Criminal Justice Information Services
200 Arlington Street, Suite 2200, Chelsea, MA 02150
MASS.GOV/CJIS
TEL: 617-660-4600 | TTY: 617-660-4606 | FAX: 617-660-4613

Table of Contents

CJIS Broker Overview

About This Document

The purpose of this document is to describe the technical concepts and processes of the XML transaction exchange framework implemented by the Department of Criminal Justice Information Services (DCJIS) for exchanging criminal justice information via XML and web services. The document and its appendices cover the processing of a CJIS Broker transaction from a high-level overview down to a hands-on initiation of an actual transaction. It also explains the relevance of the supporting materials available to assist during the development processes.

This document covers the following topics:

- **Overview** of the DCJIS service-oriented architecture and CJIS Broker transaction types, including Request, Response, and Event Transactions
- A detailed look at the complete CJIS Broker **transaction process for Request, Response, and Event Transactions**
- Information about the basic **XML schema** structure utilized for processing CJIS Broker transactions
- An overview of **transaction handling** for CJIS Broker transactions, including technical errors and business validations

Supporting appendices to this document also include:

- A **list of the CJIS transactions** supported, including CJIS queries, data exchanges, and notifications
- Technical information on **connecting and communicating with the CJIS Broker**, including a check list of items to follow to establish connectivity
- A list and complete description of each **header element** and its structure for Request, Response, and Event Transactions
- More detailed information about specific **transaction status codes and descriptions** utilized
- **Frequently asked questions** when interacting with the CJIS Broker
- An overview of each of the **CJIS queries** available, along with technical documentation on the elements in each query transaction
- An overview of the **CJIS data exchanges** available, along with technical documentation for each data exchange

CJIS Broker Architecture

The mission of the Department of Criminal Justice Information Services (DCJIS) is to "enhance public safety through information exchange by providing timely and accurate criminal justice information and services to authorized law enforcement and non-criminal justice agencies and individuals in support of promoting the public safety and security of the Commonwealth of Massachusetts." To assist in this mission, the DCJIS operates the Criminal Justice Information System (CJIS) to electronically exchange information between local and state agencies and to serve as a gateway to the National Law Enforcement Telecommunications System (NLETS) and the National Crime Information Center (NCIC).

To facilitate the sharing of criminal justice information in a device and presentation independent manner, the DCJIS has developed the CJIS Broker. The CJIS Broker is part of a larger initiative at the DCJIS to move towards a service oriented architecture (SOA). The key objectives of the CJIS Broker are to:

- Provide for speedy messaging handling
- Be capable of handling a large volume of messages, up to 10 million messages per day
- Provide for guaranteed delivery of secure messages
- Provide for fault tolerance of services
- Work as a Receiver of and Responder to messages
- Work as a Store and Forward station
- Provide for continuity of other CJIS services when certain subsystems are impaired
- Completely decouple the Technology Service (the CJIS Broker) from the Business Service (message content)

The CJIS Broker is a Message Oriented Middleware (MOM) which serves as an Enterprise Service Bus (ESB). Enterprise Service Buses provide a pluggable architecture with suitable adapters intended to be 1) platform neutral and 2) agnostic to which programming language is used. Below is a high-level illustration of the different components of the CJIS Broker. Additional information about each of the main components immediately follows. A salient feature of the model is that as new CJIS transactions are developed, they can readily be integrated within the CJIS Broker's Service Oriented Architecture and be made available via the same underlying SOAP/XML architecture and transaction exchange process.

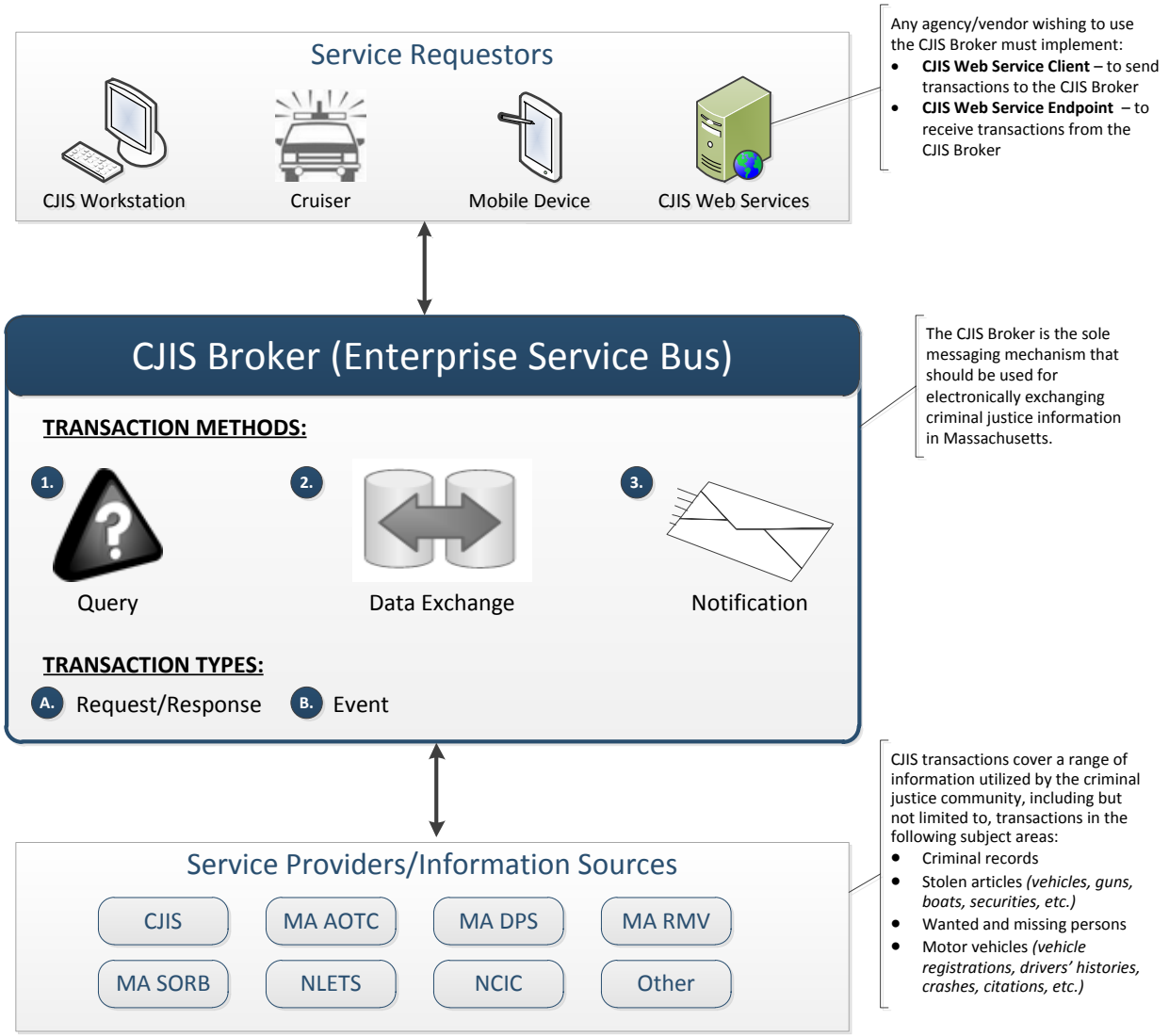

Figure 1 – Conceptual Overview of the CJIS Broker

Service Requestors and Service Providers

One of the most important concepts to understand when interacting with the CJIS Broker is the role being played by an agency/vendor. The role an agency/vendor plays can vary based on the transaction and may include:

- **Service Requestor:** A service requestor asks something of the CJIS Broker. This includes agencies/vendors that are using the CJIS Broker to query for criminal justice information, such a person's driver's history. In some situations, it also includes agencies/vendors that are using the CJIS Broker to send data via a data exchange to another agency/vendor. In certain contexts, such as a data exchange, the Service Requestor may also be referred to as the "client", or "sender".

- **Service Provider:** A service provider provides content and responds to a Request Transaction from the CJIS Broker. In the case of queries, a service provider is the source of information, such as the Registry of Motor Vehicles for Massachusetts driver's records. In some situations, it also includes agencies/vendors that are receiving data as part of a data exchange. In certain contexts, such as a data exchange, the Service Provider may also be referred to as the "responder", or "receiver".

☝ The CJIS Broker interacts with a variety of different service providers and it is possible that at any given time a specific service provider may be "down". For example, the Registry of Motor Vehicles system may be unavailable during a maintenance window. This does not mean that the CJIS Broker is down. See the transaction handling sections in this document for additional information.

The processing methods from the CJIS Web Service Definition that an agency/vendor will need to implement for their CJIS Web Service Endpoint depend on whether the agency/vendor is acting as a Service Requestor and/or Service Provider for that transaction.

Note – Any agency/vendor who interacts with the CJIS Broker in any capacity (as either a Service Requestor and/or as a Service Provider) must implement a CJIS Web Service Endpoint. This will allow the CJIS Broker to communicate with that agency/vendor.

Transaction Methods

Different kinds of transactions are available via the CJIS Broker including:

1. **Queries** – Used to inquire about a particular aspect of information, typically criminal justice related. For example, a police department may submit a query to learn more about a person's driver history. In queries, the service requestor provides input parameters and expects to receive a result set that matches the parameters specified.

Presentation of the data and application-specific formatting is entirely at the discretion of the agency/vendor entity based on its own needs and resources.

2. **Data Exchanges** – Used to pass data records between agencies/vendors for further processing. For example, a police department may wish to send motor vehicle crash reports electronically to the Registry of Motor Vehicles. In data exchanges, the sender provides a set of data records to the receiving agencies/vendors who may then store or further process the data provided.

3. **Notifications** – Used to send a communication for information purposes. For example, another state may wish to send an administrative broadcast to a police department. In notifications, the sender sends a one-way communication to the recipient. The recipient is not expected to respond electronically via the CJIS Broker to the notification.

See the *Available CJIS Transactions* appendix for a complete list of CJIS transactions currently available as part of the CJIS Broker, including queries, data exchanges, and notifications.

Transaction Types

To manage the exchange of CJIS data within the SOA, the CJIS Broker sends and receives transactions using structured XML documents. CJIS Broker transactions are raw, device-independent, XML documents. The CJIS Broker accepts well-formed XML documents and returns well-formed XML documents.

CJIS Broker transactions come in two types:

A. **Request/Response** – A Request/Response is a two way transaction sent through the CJIS Broker and includes:

- Request – The first part of a two-way transaction, a Request Transaction is sent through the CJIS Broker for which a Response Transaction is expected.
- Response – The second part of a two-way transaction, a Response Transaction is sent through the CJIS Broker as the result of a Request Transaction.

B. **Event** – An Event Transaction is a one-way transaction sent to or from the CJIS Broker for which no Response Transaction is expected.

Basic XML Document Structure for CJIS Broker Transactions

The XML documents sent in a CJIS Broker transaction must contain certain elements and be properly structured and well-formed. The structures of the XML transaction documents are dictated by a series of **XML schemas** which describe the structure of the XML document. The schemas allow all parties involved in the exchange to understand the specifications for CJIS Broker XML document structures. All CJIS Broker transactions fall into one of three types, a Request Transaction, a Response Transaction, or an Event Transaction. Accordingly, the child of the root element of every transaction is RequestTransaction, ResponseTransaction or EventTransaction, which have **a header and a body**.

Because the Request Transaction document conforms to an expected structure established by the schema, the CJIS Broker can properly interpret the type of Request Transaction and the data within the Request Transaction. The same is true for Response Transactions and Event Transactions where the receiving CJIS Web Service Endpoint knows how to properly interpret the type of Response or Event Transaction and the data within.

Schemas

The CJIS Broker schemas make heavy use of data components from the Global Justice XML Data Model (GJXDM) and the National Information Exchange Model (NIEM).

The GJXDM was developed by the Office of Justice Programs, together with the Global Justice Information Sharing Initiative. The schema has been published for use by the criminal justice community and is a broad, all-encompassing data dictionary. The purpose of the GJXDM is to provide a common vocabulary and structure to be used by various criminal justice agencies nationwide, in order to facilitate XML data exchanges. For more information and documentation about the GJXDM, please visit https://it.ojp.gov/initiatives/gjxdm.

The NIEM was an effort started by U.S. Department of Homeland Security when the homeland security community began working towards standardization. The collaborative effort to produce common and well defined data elements for data exchange by the justice and homeland security communities led to the beginnings of NIEM. The NIEM was built upon the successes and lessons learned from the GJXDM and progressed to the uniting of federal, state, local, and tribal governments to develop and deploy a national model for information sharing. The NIEM is now the broader successor to the justice specific GJXDM. For more information and documentation about the NIEM, please visit https://www.niem.gov.

Where local definitions are needed that are not provided for in the GJXDM or NIEM, the GJXDM or NIEM schema may be extended so that required elements are added within a local namespace.

For each CJIS Broker transaction, sample XML instance documents are available for additional guidance and definition for document structure.

Request and Response Transaction Handling

The complete Request/Response Transaction type between the client and the CJIS Broker is **asynchronous** and is comprised of multiple steps and individual transactions. It is initiated with a Request Transaction and ends with one or more Response Transactions. In between the first and last steps are several intermediary processing steps, including multiple Acknowledgments and processing of the Response(s) sent to CJIS Web Service Endpoint.

To maintain the guaranteed delivery obligation of the CJIS Broker, Acknowledgments are used to confirm the receipt of each transaction. The **Acknowledgment** is a synchronous confirmation that the transaction was received. A separate asynchronous Acknowledgement Transaction is NOT sent to the requestor and the Acknowledgement is NOT the Response Transaction. The Acknowledgement is provided in the return value of the web service method. The Acknowledgment consists of the elements: AcknowledgmentHeader and AcknowledgmentBody. AcknowledgmentHeader will contain the values sent in the header of the transaction being acknowledged. AcknowledgmentBody will always be empty.

The responsibility of generating an Acknowledgement depends upon the role which the CJIS Broker and agency/vendor play in the transaction. Typically for Request/Response Transaction handling, the Acknowledgement for a Request Transaction will be generated by the CJIS Broker and the Acknowledgment for the Response Transaction will be generated by the Requestor's CJIS Web Service Endpoint. The Acknowledgment will include the original header content of the specific transaction with the appropriate Transaction Status elements added. See the *Transaction Header* appendix and the *Transaction Status* appendix for more information.

The basic process for a standard Request/Response Transaction is as follows. As noted above, the entity that serves as the requestor or responder varies based on the specific transaction.

1. A Requestor builds the XML for the Request Transaction, including request header and body data, using the schemas specified.

   ☝ *The header of the Request Transaction includes information about where the Response Transaction will be sent (CJIS Web Service Endpoint). See the Transaction Header appendix for additional information.*

   The Responder's **processRequest web service method** will
   - ✓ Accept an XML document containing the Request Transaction
   - ✓ Return an XML Acknowledgment confirming receipt of the Request Transaction

2. A Requestor (Client) sends an initial **Request Transaction** through the CJIS Broker leveraging the Responder's *processRequest* web service method.

3. The Responder accepts and confirms the Request Transaction and returns a **synchronous Acknowledgment.**

4. The Responder processes the Request Transaction and generates a **Response Transaction** based on the transaction requested, and sends the Response Transaction to the CJIS Web Service Endpoint leveraging the *processResponse* web service method.

The CJIS Broker sends the Response Transaction to the CJIS Web Service Endpoint that was provided in the header of the Request Transaction.

5. Upon receiving the Response Transaction, the Requestor (CJIS Web Service Endpoint) confirms the successful receipt of the transaction by sending back a **synchronous Acknowledgment**.

The Requestor's **processResponse** web service method will

- ✓ Accept an XML document containing the Response Transaction
- ✓ Return an XML Acknowledgment confirming receipt of the Response Transaction

The diagram below depicts the multiple exchanges in a Request/Response Transaction. In this example, the Request Transaction is for a CJIS query being initiated by an agency/vendor and the CJIS Broker is acting as the responder. In other examples, such as a data exchange, CJIS Broker may be the requestor and another agency/vendor system may be the responder.

Additional details about the steps involved in these transactions immediately follows.

Figure 2 – **Request/Response Transaction** (Query to CJIS Broker Example)

Request/Response Transaction Message Calls

The following segments discuss each step in a Request Transaction and corresponding Response Transaction in greater detail, including information needed in each step.

1. Build the Request Transaction

The first step in a CJIS Broker transaction is building the Request Transaction XML document. These documents naturally vary from transaction to transaction, but each transaction will always have a **header** and a **body**. The header contains all of the metadata about the transaction and is vital for a successful transaction. The body contains the transaction payload data.

Request Header and Body

The RequestHeader element in a Request Transaction document contains metadata about the transaction, like the transaction source, a transaction ID, and other values used in tracking and processing the request. For example, the RequestHeader includes a RespondTo element which specifies the URL for the CJIS Web Service Endpoint to where the CJIS Broker will send responses. For queries, the RequestBody element contains the query input values like the query subject's name and date of birth. For data exchanges, the RequestBody element contains the payload like a crash report.

A conceptual diagram of the RequestTransaction is pictured below.

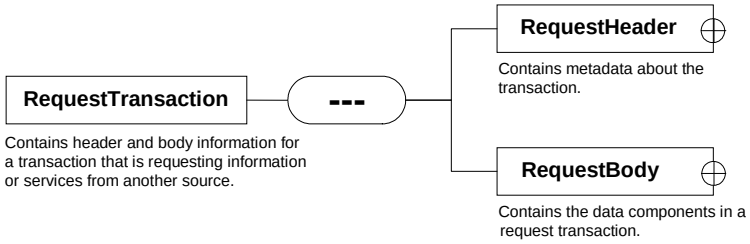

Figure 3 – **Request Transaction Conceptual Document Structure**

See the *Transaction Header* appendix for a complete description of each header element and its structure. Please also see the package document for additional information, including the schemas and sample XML instance documents, for the specific transaction being implemented.

2. Connect and Send the Request Transaction to the CJIS Broker

To send a Request Transaction, the agency/vendor must create a CJIS Web Service Client based upon the CJIS Web Service Definition (CJISService.wsdl) provided. The client must then establish a connection with the CJIS Broker.

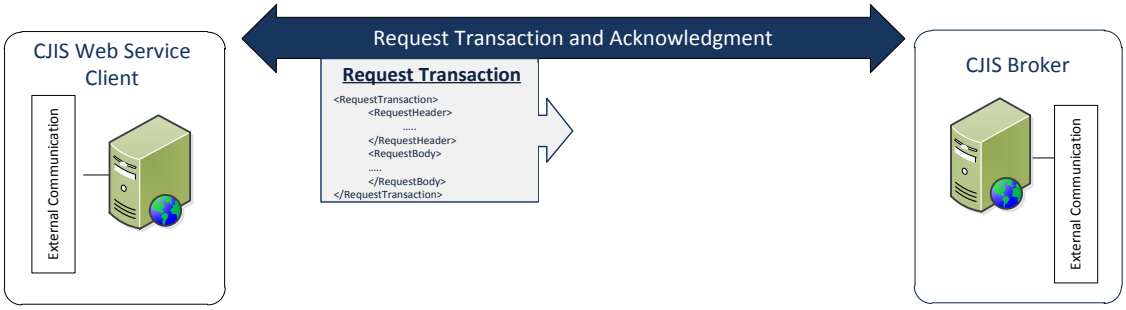

Figure 4 – **CJIS Web Service Client and Request Transaction** (Query to CJIS Broker Example)

An agency/vendor needs to receive **authorization from the DCJIS** to access the CJIS Broker via the CJIS network. As part of the authorization process, the DCJIS will configure its firewall to permit access and provide specific connection parameters. See the *Connecting and Communicating* appendix, for additional information.

3. Receive Acknowledgment of Request Transaction (Synchronous)

The Responder (in this case the CJIS Broker) will confirm receipt of the Request Transaction in the form of an Acknowledgment.

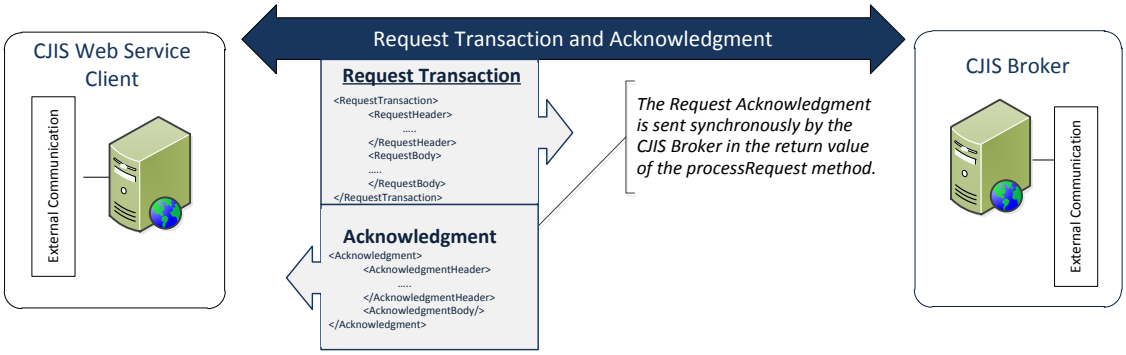

Figure 5 – **Acknowledgment of Request Transaction** (Query to CJIS Broker Example)

This Acknowledgement is not the Response Transaction; it is simply a confirmation that the Request Transaction has been received.

There is not a stand-alone Request Acknowledgment transaction. The Acknowledgment is provided in the return value of the processRequest web service method.

The requestor must check the return value information provided to determine if the Request Transaction was accepted and is being processed.

- ✓ **Success**: If the Request Transaction was accepted and is being processed, a non-negative value (zero or a positive number) will be returned in the TransactionStatus element and additional information will be returned in the TransactionDescription element.
- × **Failure**: If the Request Transaction failed due to invalid XML or system errors, a negative value will be returned in the TransactionStatus element. A description of the error will be in the TransactionDescription element.

Please refer to the "Error Handling" section of this document as well as the *Transaction Status* appendix for additional information.

4. Receive Response Transaction(s) (Asynchronous)

Since the CJIS Broker Request/Response Transactions are **asynchronous**, to receive a Response Transaction, the agency/vendor must create a CJIS Web Service End Point based upon the CJIS Web Service Definition (CJISService.wsdl) provided. The CJIS Broker will send Response Transactions to the CJIS Web Service Endpoint URL found in the RespondTo element in the header of the Request Transaction.

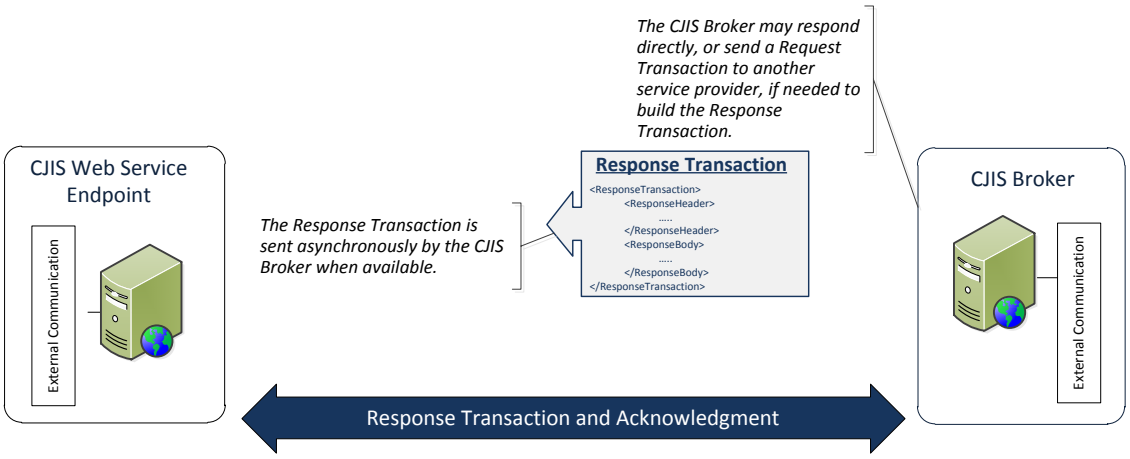

Figure 6 – **CJIS Broker and Response Transaction** (Query to CJIS Broker Example)

Most transactions will generate one and only one response. Responses to some queries, including various NLETS queries, are sometimes broken into multiple response segments by the Service Provider before they are returned to the CJIS Broker. Together, these segments comprise the logical query response. When this occurs, the CJIS Broker will send each response segment individually as a separate Response Transaction in order to complete the record. There is no way to know in advance how many query segments will be returned for any given NLETS query. See the *CJIS Query* appendix for additional information.

☝ See the "Error Handling" section of this document for additional information about what happens if the CJIS Broker is unable to send a Response Transaction to a CJIS Web Service Endpoint, for example if there is a technical connectivity issue with the CJIS Web Service Endpoint.

Response Header and Body

With the exception of a few key elements such as MessageIdentifier, which will contain a unique value for each message in a transaction, the Response Header will always return most of **the original values from the Request Header.** This is so that they are available to the client should it require any of those values for identification or processing. In addition to those elements, the Response Header will contain new values pertaining to the status of the transaction. Please see the *Transaction Header* appendix for further information.

For queries, the ResponseBody element may contain the original request input values plus the actual response, a driver's license record for example. A conceptual diagram of the ResponseTransaction is pictured below.

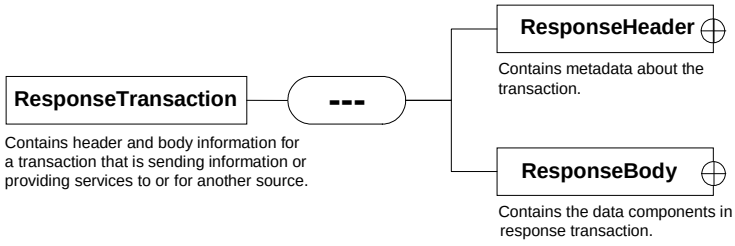

Figure 7 – Response Transaction Conceptual Document Structure

Please also see the package document for additional information, including the schemas and sample XML instance documents, for the specific transaction being implemented.

5. Send Acknowledgment of Response Transaction (Synchronous)

The CJIS Web Service Endpoint that accepts the Response Transaction from the CJIS Broker will need to confirm the receipt of the Response Transaction in the form of an Acknowledgment. This Acknowledgement is used as a means of confirming to the CJIS Broker that the CJIS Web Service Endpoint has successfully received the Response Transaction, thereby relieving the CJIS Broker of its guaranteed delivery obligation.

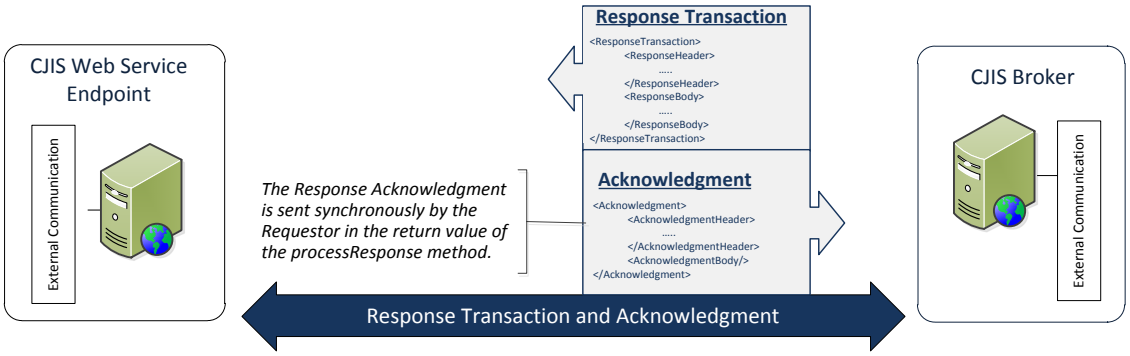

Figure 8 – Acknowledgment of Response Transaction (Query to CJIS Broker Example)

There is not a stand-alone Response Acknowledgment transaction. The Acknowledgment is provided in the return value of the processResponse web service method.

If an Acknowledgment indicating a successful receipt of the Response Transaction is not generated by the CJIS Web Service Endpoint, the **CJIS Broker will continue attempting to send the Response Transaction until it receives a successful Acknowledgment**. The CJIS Web Service Endpoint must aim to acknowledge the receipt of the Response Transaction as soon as it accepts the Response Transaction from the CJIS Broker.

☝ Any time-consuming processing of the Response Transaction must be handled after the Acknowledgment has been sent to the CJIS Broker.

The CJIS Broker will wait for 5 seconds before it considers the delivery of the Response Transaction to have failed and attempt re-sending the Response Transaction.

- ✓ **Success**: If the CJIS Web Service Endpoint is able to accept the Response Transaction it must return a non-negative value (zero or a positive number) in the TransactionStatus element and provide additional information in the TransactionDescription element.
- × **Failure**: If accepting of the Response Transaction failed due to invalid XML or system errors, a negative value must be returned in the TransactionStatus element. A description of the error must be contained in the TransactionDescription element.

Please refer to the "Error Handling" section of this document as well as the *Transaction Status* appendix for additional information.

Event Transaction Handling

A CJIS Broker Event Transaction is a **one-way transaction** sent to or from the CJIS Broker for which **no response is expected**. A transaction is initiated with an Event Transaction and ends with the Acknowledgment. Unlike Request Transactions, a separate asynchronous Response Transaction will not be delivered after the transaction has been successfully acknowledged.

Send an Event Transaction

The diagram below depicts the exchanges in an Event Transaction. In this example, an agency/vendor is sending an event to the CJIS Broker.

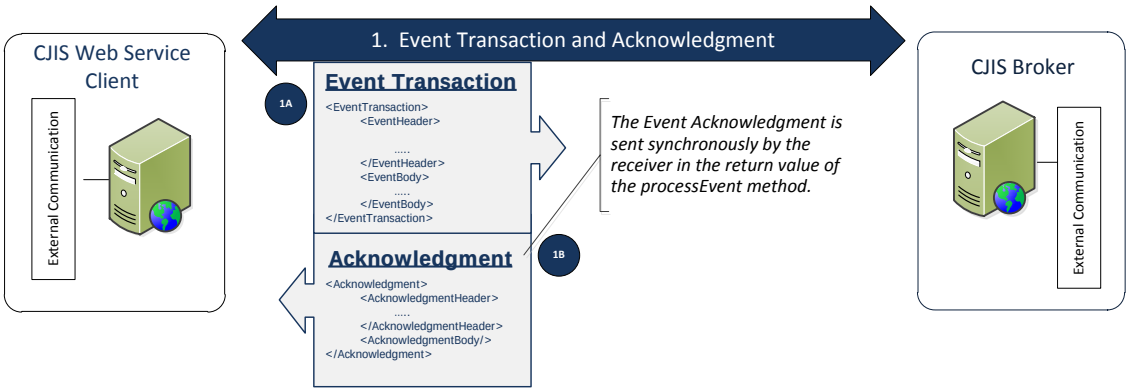

Figure 9 – Send Event Transaction Conceptual Overview

Send Event Transaction Message Calls

The following segments discuss each message call in a send Event Transaction in greater detail including information needed in each step.

1. Build the Event Transaction

The first step in a CJIS Broker Event Transaction is building the Event Transaction XML document. These documents naturally vary, but each transaction will always have a header and a body.

Event Header and Body

Similar to the header in a Request Transaction, the EventHeader element in an Event Transaction document contains metadata about the transaction, like the transaction source, a transaction ID and other values used in tracking and processing the event. The EventBody element contains the data values like a text based notification message. A conceptual diagram of the EventTransaction is pictured below.

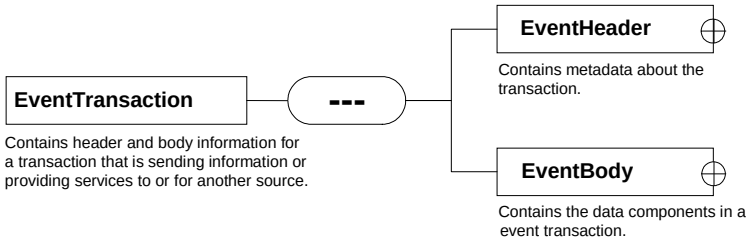

Figure 10 – **Event Transaction Conceptual Document Structure**

An Event Transaction header is populated in a similar manner as a Request Transaction with a few key exceptions as follows:

- ResponseMechanism element in the header must be set to "NONE"
- RespondTo element in the header must be empty

See the *Transaction Header* appendix for a complete description of each header element and its structure. Please also see the package document for additional information, including the schemas and sample XML instance documents, for the specific transaction being implemented.

2. Connect and Send Event Transaction to the CJIS Broker

To send an Event Transaction, the agency/vendor must create a CJIS Web Service Client based upon the CJIS Web Service Definition (CJISService.wsdl) provided. The client must then establish a connection with the CJIS Broker.

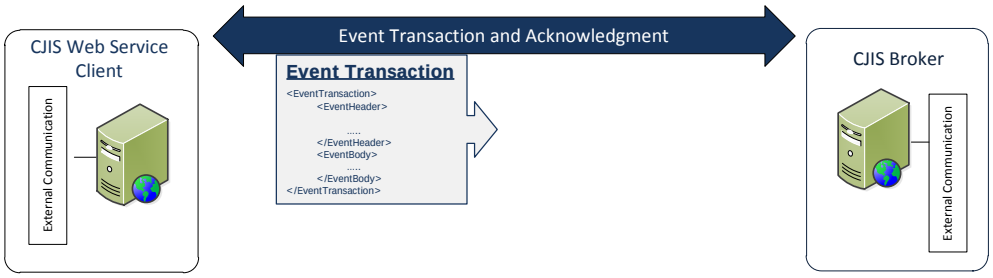

Figure 10 – **CJIS Web Service Client and Event Transaction** (Event to CJIS Broker Example)

An agency/vendor needs to receive **authorization from the DCJIS** to access the CJIS Broker via the CJIS network. As part of the authorization process, the DCJIS will configure its firewall to permit access and provide specific connection parameters. See the *Connecting and Communicating* appendix, for additional information.

3. Receive Acknowledgment of Event Transaction (Synchronous)

When it receives an Event Transaction, the CJIS Broker will confirm receipt of the Event Transaction in the form of an Acknowledgment.

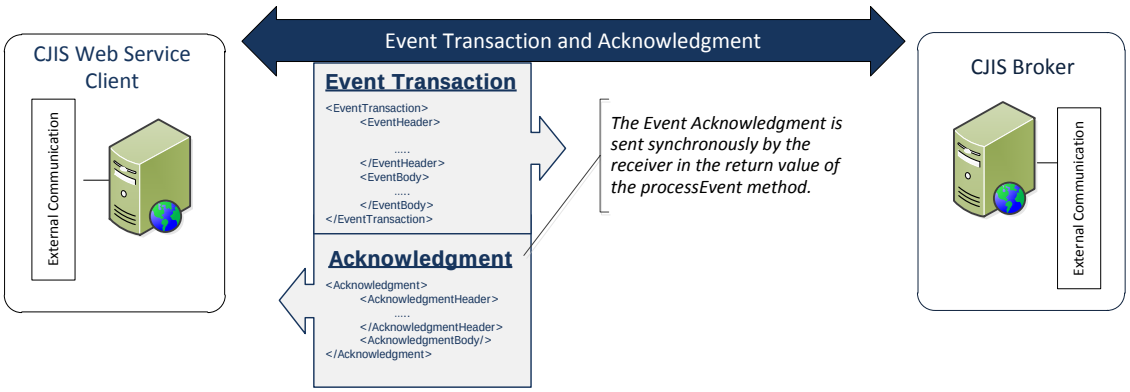

Figure 11 – **CJIS Broker Acknowledgment of Event Transaction** (Event to CJIS Broker Example)

There is not a stand-alone event acknowledgment transaction. The Acknowledgment is provided in the return value of the processEvent web service method.

The requestor must check the return value information provided to determine if the Event Transaction was accepted and is being processed.

- ✓ **Success:** If the Event Transaction was accepted and is being processed a non-negative value (zero or greater) will be returned in the TransactionStatus element and additional information will be returned in the TransactionDescription element.
- × **Failure**: If the Event Transaction failed due to invalid XML or system errors a negative will be returned in the TransactionStatus element. A description of the error will be contained in the TransactionDescription element.

Please refer to the "Error Handling" section of this document as well as the *Transaction Status* appendix for more information.

Receive an Event Transaction

The example above demonstrates how to send an Event Transaction to the CJIS Broker. There are also instances where an agency/vendor needs to receive an Event Transaction from the CJIS Broker, including but not limited to planned event notifications.

Receive Event Transaction Message Calls

The following segments discuss each step in receiving an Event Transaction from the CJIS Broker in greater detail, including information needed in each step.

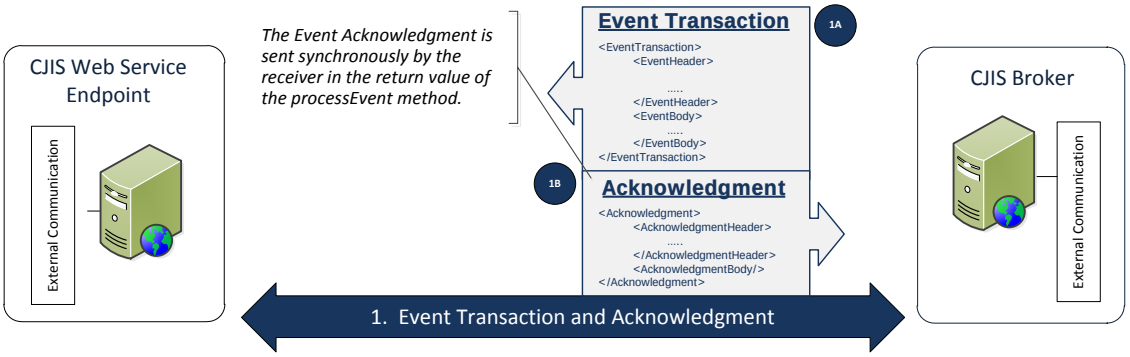

Figure 12 – Receive Event Transaction Conceptual Overview

1. Receive Event Transaction

Similar to receiving a Response Transaction, to receive an Event Transaction from the CJIS Broker, the agency/vendor must create a CJIS Web Service Endpoint based on the CJIS Web Service Definition (CJISService.wsdl) provided. The web service will include the processEvent web service method which allows the recipient to receive an Event Transaction from the CJIS Broker and provide a synchronous Event Acknowledgment back to the CJIS Broker.

The receiver of the event's **processEvent** web service method will:

- ✓ Accept an XML document containing the Event Transaction sent
- ✓ Return an XML Acknowledgment confirming receipt of the Event Transaction

2. Send Acknowledgment of Event Transaction

The CJIS Web Service Endpoint that accepts the Event Transaction from the CJIS Broker will need to confirm receipt of the Event Transaction in the form of an Acknowledgment. This Acknowledgment is used as a means of confirming to the CJIS Broker that the CJIS Web Service Endpoint has successfully received the Event Transaction, thereby relieving the CJIS Broker of its guaranteed delivery obligation.

There is not a stand-alone event acknowledgment transaction. The Acknowledgment is provided in the return value of the processEvent web service method.

If an Acknowledgment indicating a successful receipt of the Event Transaction is not generated by the CJIS Web Service Endpoint, the **CJIS Broker will continue attempting to send the Event Transaction until it receives a successful Acknowledgment**. The CJIS Web Service Endpoint must aim to acknowledge a receipt of the Event Transaction as soon as it accepts the Event Transaction from the CJIS Broker.

☝ Any time-consuming processing of the Event Transaction must be handled after the Acknowledgment has been sent to the CJIS Broker.

The CJIS Broker will wait for 5 seconds before it considers the delivery of the Event Transaction to have failed and attempt re-sending the Event Transaction.

- ✓ **Success**: If the CJIS Web Service Endpoint is able to accept the Event Transaction it must return a non-negative value (zero or a positive number) in the TransactionStatus element and provide additional information in the TransactionDescription element.
- × **Failure**: If accepting of the Event Transaction failed due to invalid XML or system errors, a negative value must be returned in the TransactionStatus element. A description of the error must be contained in the TransactionDescription element.

Please refer to the "Error Handling" section of this document as well as the *Transaction Code* appendix for more information.

Planned Event Notification Handling

There are instances when the DCJIS wants to communicate with CJIS Broker agencies/vendors about planned events, such as planned system outages and maintenance cycles, new CJIS Broker transactions, and more. When this occurs, an unsolicited message will be sent from the CJIS Broker with the notification information. Like other Event Transactions, these notifications will include an XML header and body and users will need to synchronously acknowledge that the Event Transaction was received.

Error Handling

During any portion of a CJIS Broker transaction, technical errors can occur on either the client or server side. If the error is initiated on the client side (outside of the DCJIS' processing), due to network failure, connectivity loss, server failure, server unavailability, or any other reason, the client should generate an error or exception which the application can handle.

If the error occurs during the processing of a transaction within the DCJIS' environment, the Acknowledgment or Response Transaction will be returned to the client with details about the error in the response or event header. Below is an example of header Transaction Status data from a failed transaction:

Each agency/vendor using the CJIS Broker should determine how best to communicate notifications, including CJIS planned event communications and other unsolicited messages, with their user community.

```
...

  <dcjis:Transaction>

    <dcjis:TransactionStatus>-
999</dcjis:TransactionStatus>

    <dcjis:TransactionDescription>

      An error has occurred while processing this transaction

    </dcjis:TransactionDescription>

  </dcjis:Transaction>

...
```

Transaction Status and Description

In a successful transaction, the Transaction Status code will always be a non-negative value (zero or greater). An unsuccessful transaction will always contain a negative value. Transaction Descriptions are for informational purposes only and may change over time. Agencies/vendors should have a mechanism in their system to handle errors. Agencies/vendors should NOT program any automated error handling functionality based on Transaction Description values.

☝ *Any automated error handling logic should be based exclusively on Transaction Status codes and not on description information.*

See the *Transaction Status* appendix for additional information, including specific Transaction Status code and Transaction Description values.

Disabled CJIS Web Service Endpoint

If the CJIS Broker is unable to send a Response Transaction to a CJIS Web Service Endpoint, the status of that CJIS Web Service Endpoint will be managed as per the following specifications:

1. At the first failure, the CJIS Web Service End Point will be marked as **potentially unavailable**. All new Request Transactions received by the CJIS broker pertaining to such a CJIS Web Service Endpoint will continue to be accepted and processed by CJIS Broker.
2. If the CJIS Web Service Endpoint continues to have issues, it will be marked **disabled** based on a DCJIS specified criteria. Such criteria includes disabling the CJIS Web Service Endpoint if it has been potentially unavailable for a specific amount of time and/or if a specific number of continuous unsuccessful connection attempts have been made to the CJIS Web Service Endpoint.
    a. The CJIS Broker will not accept any incoming Request Transactions for a disabled CJIS Web Service Endpoint. Under some conditions the CJIS Broker will also not accept any incoming Event Transactions.
    b. The Acknowledgment sent by the CJIS Broker for any new Request Transactions will have a specific Transaction Status code and description containing further information about the CJIS Web Service Endpoint being disabled. See the *Transaction Status* appendix for additional information.

Re-enabled CJIS Web Service Endpoint

The CJIS Broker will periodically check to determine whether the CJIS Web Service Endpoint is available.

1. Once the CJIS Web Service Endpoint is available, the CJIS Broker will 1) begin to accept new Request Transactions and 2) will send the Response Transactions that were unable to be delivered when the CJIS Web Service Endpoint was marked as potentially unavailable.
2. The Requestor will be responsible for tracking and resubmitting transactions for which they received an Acknowledgment indicating that the transaction was not successfully processed (e.g., negative Transaction Status code).

Business Rules

In addition to technical errors (e.g., network failure, connectivity loss, etc.) additional business logic may be implemented specific to each transaction. For example, the Registry of Motor Vehicles may choose not to accept new crash records that do not meet their minimum data quality standards. Depending on the transaction, two different types of business validations may be available including:

- **Business Errors (Reject)** – Indicates that the recipient of the data exchange is unable/unwilling to process the information until a corrective action is taken.
- **Business Warnings (Accept with Warning)** – Indicates that the recipient of the data is willing and able to process the information, however, there is information they want the sender to be aware of regarding the exchanged information.

Information about both business errors and warnings are included as part of the body of certain data exchange transactions. For additional information related to business errors and warnings, please see the package document for additional information, including the schemas and sample XML instance documents, for the specific transaction being implemented.

Business validations are not the same as the Transaction Status codes which are used in the transaction header to indicate that the transaction was successfully received, processed, and no technical issues or errors occurred that prevented the transaction from being processed. Business validations are usually specific to each transaction or data exchange, if they are supported.

Additional CJIS Broker Documents

For each transaction to be implemented, the DCJIS also provides an additional package that contains a number of documents and tools to assist in the development of technologies to interact with the CJIS Broker environment. The content of the package depends on the transaction and may include:

- **CJIS Web Service Definition** (CJISService.wsdl) – including information about the CJISService web service, including the processRequest, processResponse, and processEvent web service methods.
- **XML Schemas** – Including document, constraint, subset, and extension schemas as appropriate.
- **Sample XML Documents** – A sample XML instance document for each transaction that conforms to the schema definitions and contains the various elements (or tags) available within the schema. These documents are samples of what valid transaction documents look like. There are multiple document structures that will be valid (e.g. query requests with multiple and mutually exclusive input values), but these samples provide a foundation of what the documents will look like. These examples are available to demonstrate tag order and document structure. The sample XML documents are located in the **sample_xml** directory.
- **Component Mapping Sheet** – A tool to help navigate from system database structures to formatted XML schemas for data exchanges. Includes the proposed elements to be exchanged and other information such as data type, length, cardinality, and reference code

information. It also includes information about business validations (rejects and accepts with warning) associated with individual data elements or across data elements.

- **Valid Code List Worksheet** – For data exchanges, identifies the major categories of reference code information to be used, including the proposed source of each code list as well as the specific values (codes and descriptions).

Appendices

The following information is available as appendices to this document for reference when interacting with the CJIS Broker.

- Available CJIS Transactions
- Connecting and Communicating
- Transaction Header
- Transaction Status
- Frequently Asked Questions
- Query Technical Details
  - CJIS Queries
  - NLETS Queries
  - NCIC Queries
  - RMV Queries
- Data Exchange Technical Details
  - Motor Vehicle Citation and Crash System Exchanges
  - Electronic Application for Criminal Complaint

Revision History

The CJIS Broker Documentation is a living document that is updated as required over time as additional transactions and functions become available. Following is a summary of key changes made, including the date and nature of changes.

Version	Date	Nature of Changes
1.0	06/27/2012	Initial documentation.
1.1	07/26/2013	Minor edits and formatting adjustments.
1.2	04/24/2014	Updated to include additional information related to event transactions.
2.0	07/20/2015	Major updates to reflect additional queries and data exchanges. Included technical documentation about CJIS queries, data exchanges, connectivity, FAQs, and more as appendices to this document.