**OSMS Design Principles and Preliminary Project Plan – Revised Jan. 4, 2018**

The Open Source Message Switch (OSMS) Project is funded by the Bureau of Justice Assistance Technology Innovation for Public Safety (TIPS) Program through a grant to the Puerto Rico Department of Justice. PRDOJ, in turn, subawarded funds to SEARCH, The National Consortium for Justice Information and Statistics, to lead this project and is joining the Open Justice Broker Consortium (OJBC). The goal of this project is to develop a viable law enforcement message switch computer system that uses open source technologies and components and that implements national justice information sharing standards.

The project team consists of representatives from the Puerto Rico CJIS agency (Sistema de Informacion de Justicia Criminal (SIJC)), the Montana Department of Justice – CJIS Division (MT DOJ), Nlets – the International Justice and Public Safety Network, and SEARCH.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Team** | **Role** | **Organization** | **Contact Information** |
| Lcda. Denisse Cintron | Project Sponsor/SME | PR SIJC |  |
| Miguel Soto Pastrana | PM/Lead Developer | PR SIJC |  |
|  |  |  |  |
| Jennifer Viets | SME | MT DOJ | [jviets@mt.gov](mailto:jviets@mt.gov) |
|  |  |  |  |
| Kate Silhol | SME | Nlets | [ksilhol@nlets.org](mailto:ksilhol@nlets.org) |
|  |  |  |  |
| Andrew Owen | Software Development Lead | SEARCH/OJBC | [Andrew@search.org](mailto:Andrew@search.org) |
| Mark Perbix | Project Administrator/SME | SEARCH/OJBC | [Mark@search.org](mailto:Mark@search.org) |
| Mike Jacobson | PM/Scrum Master | SEARCH | [mike@search.org](mailto:mike@search.org) |
| Christine Martin | Grant Administrator | PR DOJ |  |

A law enforcement message switch is a specialized computer system that links together multiple remote computer systems to provide law enforcement access to data resources and capabilities. The key business function of the message switch is to provide authorized users the capability to interact with multiple state and national law enforcement data systems to enter and retrieve law enforcement sensitive information. In providing this business capability, the message switch must provide the ability to monitor system performance and individual transactions that occur throughout the system. The message switch should provide the following capabilities:

* Allows users to access and manage information in national and state criminal justice information systems
* Connects disparate criminal justice information systems using asynchronous messaging in the format and protocols native to each information system
* Supports a wide variety of message formats and protocols that use configurable and flexible message processing and interface capabilities
* Provides guaranteed message delivery with high availability and high performance using message store and forward capabilities
* Acts as the control terminal message switch for systems such as NCIC and Nlets that require connection to a single system in a state
* Supports the latest information sharing technologies and standards
* Integrates with legacy technologies

Specific functional/business and non-functional requirements are described in the Requirements section below.

The purpose of this document is to identify the design principles to be followed in developing this software and provide a high-level description of the software requirements and development approach.

As a result of several planning meetings and discussions with the project team, the team identified the following guiding principles for this project.

Design Principles

1. Portable – the solution should be able to be readily deployed in different environments and by different users/customers.
2. Configurable – the solution should enable the customer to create and manage as many functions as possible using configuration capabilities rather than having to rely on a technology service provider to perform these functions. This will not remove the need for technological expertise, but this means if such expertise exists in-house, the customer could leverage that expertise and does not have to depend on a vendor to make such updates.
3. Standards-based – the solution will implement justice and industry standards. This will include the adoption and use of the Global Standards Package and National Information Exchange Model.
4. Non-proprietary – the solution will use open source components, products and technologies whenever possible.
5. Reusability – the solution will componentize key elements of the system in a more granular manner whenever possible to maximize flexibility and reuse. For instance, the message switch will be designed in a manner that decouples it from any single client application.
6. Message correlation – much of the functionality of the message switch relies on the ability to process transactions asynchronously with multiple disparate end points.
7. Security and Reliability – the solution must be able to meet CJIS security requirements at a minimum, provide high reliability, and guarantee message integrity and delivery.
8. Implement only the current XML standard messages and will not implement text-based socket protocols.

Business Requirements

The message switch must provide the capability to:

1. authenticate and authorize users;
2. validate messages;
3. route and manage messages;
4. provide connectivity to remote systems;
5. send, receive and correlate messages sent to multiple remote systems; and
6. log transactions

Non-functional Requirements

The message switch must provide tools to:

1. manage users and devices;
2. provide logging and auditing capabilities;
3. detect and resolve errors;
4. monitor system performance; and
5. monitor system functionality throughout the environment.

Project Scope Diagram

The above business and non-functional requirements are illustrated in the scope diagram *(enter document name. Can we set up a repository for everyone to access documentation?*

Capabilities Out of Scope

1. Client software. A client application is required for a user to interact with the message switch. Numerous client products exist that could be modified to use the OSMS. The scope of this project is to develop an open source interface/data exchange standard that enables other entities to develop client software to interact with the message switch.

The interface should receive incoming structured data from client so federated queries can be developed dynamically based upon configurations in switch (rather than clients sending in specific message key transactions).

Software Development Tools, Technologies and Methodology

The open source message switch will adhere to the OJBC software development architecture which is based on Java and other open source frameworks such as Spring, Maven, Apache Camel and Apache CXF. The OJBC methodology also puts a strong emphasis on unit testing as well as automated continuous integration using Jenkins. All source code will be checked into a GitHub repository. Initially this repository will be private, but it will ultimately be open this to the public.

Software development will follow an Agile methodology, according to the Scrum development process. Specifically, the process will rely on the following concepts established by Scrum:

* **Sprints** – these will be 2-3 week windows of development time where the development team focuses on a well-defined set of tasks
* **Back Log** – an ever-evolving list of prioritized development tasks, this is used to establish the scope of each sprint.
* **Estimation –** as an initial step to each Sprint, the Scrum Master will work with the development team to identify scope and cost of the upcoming sprint
* **Sprint Demo/Acceptance testing** – at the end of each Sprint, the development team will demonstrate functionality for acceptance and closure of the Sprint.
* **Scrum Master** – overall facilitator of the development team. This role ensures the team has clear requirements and is the primary person responsible for removing impediments and keeping each Sprint focused on scope. It is preferable that this person has received formal Scrum Master training and certification. Though there really isn’t the concept of a project manager (PM) in Scrum, this role is probably the closest thing to a PM.

Initial Approach

Initial back log tasks will be based on a business use case that exercises all of the core components and capabilities of the message switch. The business use case for initial development will be based on the **traffic stop** scenario. *Reference or insert the scope of this scenario from Jen’s spreadsheet.*

Each component or element will be defined as part of the agile development methodology following the core requirements defined previously.

Next Steps

1. Complete use case for vehicle stop (JV)
2. Design Service Specification for Client interface (MJ)
   1. Based on GRA and NIEM.
3. Acquire NCIC NIEM message specifications (KS)
4. Design configurable Authentication and Authorization mechanism (AO)
   1. Will be based on SAML.
   2. Consider use of JWT.
5. Complete hiring process for software developers and project manager (MP/AO)
6. Establish bi-weekly status meetings (MJ)

Task Plan

* *Client Interface – validation of requirements (1/4/18 –* 
  + *List of interfaces*
  + *Data items in each interface*
  + *SSP for each or all*
  + *Message back*
* *Prioritize interfaces*
* *Development (Agile) – user story every 2-weeks?*
* *Testing*

Interfaces

The following interactions and interfaces have been defined as the functions required to support the primary use case. These are listed in priority order and will addressed and included as total project resources allow.

1. Vehicle Registration (RQ)
2. Vehicle Query (QV)
3. Driver’s License (DQ)
4. Person Query (QW)
5. Gun Query (QG)
6. Boat Registration (BQ)
7. Snowmobile Registration (SQ)
8. Commercial Vehicle (AVQ)
9. Article (QA)
10. State Criminal History (IQ/FQ)
11. III Criminal History (QH/QR)
12. National Sex Offender (NSOR)
13. Protection Order
14. Wanted Persons
15. Order of Arrest
16. Missing Persons
17. Identity Theft
18. Supervised Release

Key Deliverables

1. OSMS (broker)
   1. Logging and audit reporting
   2. Message lookup
   3. Alerts
   4. Message key configuration
   5. User configuration
   6. Device configuration
   7. Help file configuration
2. Authentication and authorization
3. Client software (to be provided by resources external to the project)
4. Nlets interface
5. NCIC interface
6. State Hotfile interface
7. State Driver interface
8. State Vehicle interface

References

|  |  |
| --- | --- |
| CJIS Security Policy | <https://www.fbi.gov/file-repository/cjis-security-policy-v5_5_20160601-2-1.pdf> |
| Nlets File Reference Card | <https://www.nlets.org/nlets-resources/library?a=downloadRaw&documentid=98232c82-0451-11e3-bd4a-00155d003202> |
| Nlets Wiki | <http://wiki.nlets.org/index.php/Contents> |
| Overview of Criminal Justice Information Systems | <http://www.cfp2000.org/papers/dempsey.pdf> |
| NY State IEPDs | <https://troopers.ny.gov/IEPD/> - some IEPDs that we may find useful |
| XML Specification for Interstate Rap Sheets | <https://it.ojp.gov/NISS/iepd/402> |
| 2003 XML Rap Sheet spec | <https://it.ojp.gov/NISS/Downloads/IEPD/177> |
| Driver name search IEPD | <https://niem.gtri.gatech.edu/niemtools/iepdt/display/container.iepd?ref=g3oAo%2F%2BHUeI%3D> |
| Message switch diagram | <https://sgi1.sharepoint.com/SEARCHDocumentLibrary/OSMS%20High%20Level%20Diagram.vsdx?d=w87073ee4fd4846ea945813e948f1b839> |
| Federated queries spreadsheet | <https://sgi1.sharepoint.com/SEARCHDocumentLibrary/OSMSFederatedQueries.xlsx?d=wd31246330cee47d9a5516ef1582e6dba> |
| High-level BPDD | <https://sgi1.sharepoint.com/SEARCHDocumentLibrary/OSMS_BPDD_v_1.0.0.docx?d=w23735240cb2447728e1fc1073d637dcd> |