System Architecture Document - Disposition Reporting Management

Prepared for the

Arizona Criminal Justice Commission

ARIZONA CRIMINAL JUSTICE COMMISSION
1982

November 2004

spherion SM

NORTHROP GRUMMAN
Information Technology

eCorridor

TABLE OF CONTENTS

1 INTRODUCTION

This System Architecture Document is the second of two major deliverables prepared for the Conceptual Design Phase of the Disposition Reporting Management (DRM) project sponsored by the Arizona Criminal Justice Commission (ACJC). This document compliments the Conceptual Design Document by recommending a system architecture that both supports the business processes identified in the conceptual design, and also positions ACJC for future integration efforts. Spherion prepared this document for the ACJC with subcontractors Northrop Grumman Information Technology and eCorridor.

1.1 PURPOSE

The purpose of this document is to provide a comprehensive architectural overview of the DRM system and its supporting infrastructure. As indicated in the *Arizona ICJIS Strategic Plan* (2002) preceding this work, it is the intention that this platform also serves as an environment to host future statewide integration efforts. The approach utilizes different architectural views to depict various aspects of the system, from broad system components to the physical database design. It is intended to provide ACJC with an understanding of the environment needed to support the DRM, while also demonstrating how this solution takes into consideration national standards and emerging technology directions with respect to systems integration.

1.2 SCOPE

The System Architecture Document describes the framework required to support the vision presented in the Arizona ICJIS Strategic Plan for criminal charge disposition reporting, and specifically to support the requirements identified in the DRM Conceptual Design. This document also makes recommendations for the technology platforms necessary to implement the DRM system and to establish an environment conducive to wider adoption in statewide integration projects.

1.3 OVERVIEW

This document recommends a system architecture utilizing a Service Oriented Architecture (SOA) approach. We believe an SOA is the best fit of available technology for the current and future business needs of the AZ ICJIS system. An SOA system architecture supports the business processes identified in the DRM Conceptual Design Document as well as being flexible and scalable enough to meet the future criminal justice integration needs to be identified in Arizona. It is an open system architecture, that is consistent with current best practices, fits within the State's and local agencies computing environments, and adheres to accepted standards in the field of criminal justice information systems integration. The recommended architecture will readily support the current expected usage patterns and allow for easy expansion of processing power, system capabilities and external interfaces.

The System Architecture Document contains the following sections:

- *Section 1 Introduction:* This section provides an overview of the purpose and scope of the document.

- *Section 2 Architecture Requirements:* This section contains a discussion of the goals of the DRM, the expected size and scalability needs of the system, and issues related to reliability, availability and portability. This section also discusses the cost benefits of the recommended architecture implemented to take advantage of a clustered computing environment. In such an environment, the physical hardware components may be expanded over time. As it is intended that the DRM be implemented as a "day forward" system with the processing demands, memory usage and data storage requirements growing over time, the recommended architecture can be easily expanded to grow in capacity as the number of managed cases and participating agencies increases.

- *Section 3 Recommended System Architecture and System Components:* This section contains a discussion of the integration framework, system architecture components and interface strategy in the implementation of the DRM. The integration framework will be based on Service Oriented Architecture and supporting technologies. This Section contains a diagram of the recommended system architecture that we believe will best serve the cost, implementation timeframe and functionality needs of the State of Arizona and participating agencies. This section also gives an overview of system components, how they interact within the DRM and the strategy for data exchange and interfacing with external systems.

- Section 4 Database Model: This section discusses the identification of the structures that will persist the data and support the business functionality requirements. The proposed system component for persistent data storage for the DRM is a relational database. The data model suggested for the system includes not only a representation of the required entities but also defines the relationships that these entities must have with one another to support the functionality of the system. The data entity model shows the entities that are required for storing disposition-reporting data as well as attributes that will be required for internal DRM processing and for linking DRM transactions to various external systems.

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

The table below identifies technical terms and acronyms used within this document, providing a definition of those terms.

Term/Acronym	Definition
ACCH	Arizona Computerized Criminal History system.
ACJC	Arizona Criminal Justice Commission
ADS	Active Directory Service – a Microsoft product that can be used for access control to networked resources
API	Application Programming Interface
Attribute	In simple terms, an attribute can be defined as an inherent characteristic of something, or as an object or thing that is closely related to or belonging to something else. In this writing, attributes belong to entities, and can be thought of as characteristics of an entity. For example the characteristics of an agency could include the agency name, and the ORI. When implemented in a physical database, attributes become the columns of a table.
BPEL	Business Process Execution Language - a vendor-neutral mechanism for describing the behavior of business processes
BPM	Business Process Management
Criminal Cycle Identifier (CCID)	The criminal cycle identifier (CCID) is proposed as a replacement for the AFIS-generated PCN when used in integrated justice data exchanges. The CCID would serve as the unique identifier for all cycles regardless of how the charges are initiated.
COTS	Commercial Off-the-Shelf Software
DML	Data Manipulation Language – SQL that manipulates rather than defines data in an existing data structure
DRM	The Disposition Reporting Management System
DMZ	Demilitarized Zone – an isolated area of a computer network where publicly accessible servers are typically placed to enhance system security

Term/Acronym	Definition
EFTS	Electronic Fingerprint Transmission Standard
Entity	In a logical data model, an entity refers to a thing of significance. When building a physical database, each entity translates to a table that is used to hold similar data.
HTTP	Hypertext Transfer Protocol – the communication protocol used by web servers and web clients on a TCP/IP network.
HTTPS	Hypertext Transfer Protocol with SSL – encrypted HTTP communications using the SSL standard.
JAD	Joint Application Design
JDBC	Java Database Connectivity
ICJIS	Integrated Criminal Justice Information System
Metadata	A description or definition of data
RACF	Resource Access Control Facility - an IBM product which provides access control
RDBMS	Relational Database Management System
SOA	Service Oriented Architecture – a software architecture style that relies on loose coupling of software agents.
SOAP	Simple Object Access Protocol - a lightweight XML-based messaging protocol
SQL	Structured Query Language – a programming language used to interact with a database management system
SSL	Secured Sockets Layer – A standard defined encryption protocol for use in TCP/IP networks.
TCP/IP	Transmission Control Protocol/ Internet Protocol is the basic networking protocol on the Internet.
UDDI	Universal Description, Discovery and Integration - a web-based distributed directory of available web services
WSDL	Web Services Description Language - an XML-formatted language used to describe a Web service's capabilities
XML	EXtensible Markup Language - A flexible way to create standard information formats
XSL	EXtensible Stylesheet Language - A language created for describing stylesheets for XML documents

1.5 REFERENCES

The system architecture recommended in this document is derived from references that include background reports published by ACJC, industry standards for integration of criminal justice systems published by several ICJIS organizations, and system manuals and interface definitions for existing systems currently in use within the Arizona criminal justice community. Also, several Joint Application Design (JAD) participants provided documents that describe their particular systems and business processes in use within specific jurisdictions.

Document Title	Publisher	Publish Date	Location Reference
Arizona ICJIS Strategic Plan	ACJC	March 21, 2001	http://www.acjc.state.az.us/pubs/032801_ICJIS_FINAL.pdf
Criminal Justice Information Portal Budget and Planning Estimate	ACJC	November 2002	
Global Justice XML Data Model	USDOJ and Global Justice Information Sharing Initiative	Version 3	http://it.ojp.gov/topic.jsp?topic_id=43
NIST Special Publication 500-245 - American National Standard for Information Systems—Data Format for the Interchange of Fingerprint, Facial, & Scar Mark & Tattoo (SMT) Information	National Institute of Standards	September 2000	http://it.ojp.gov/servlet/ShowDocument?attachment_id=106
AZAFIS Data Dictionary	Arizona Department of Public Safety	November 25, 2003	
GSP III – Arizona Fingerprint Data Router Interface Control Document (ICD)	Arizona Department of Public Safety	February 25, 2004	
Local Agency – Arizona Computerized Criminal History System Interface ACCH Disposition Transaction User Specifications	Arizona Department of Public Safety	May 2004	
JIEM Reference Model 1.0.1	SEARCH	May 2004	http://www.search.org/integration/JRM1.0.1.pdf
Arizona Computerized Criminal History User Manual	Arizona Department of Public Safety	June 2004	
Maricopa County ICJIS Strategic Plan	Maricopa County ICJIS	June 2003	
Maricopa County Disposition Report Map	Maricopa County ICJIS	June 11, 2004	
Pinal County Data Process Flow	Pinal County Justice Integration Project Office	December 29, 2003	

2 ARCHITECTURE REQUIREMENTS

2.1 OVERVIEW

In planning for any large scale information technology project, it is vitally important to define as early as possible the basic system requirements and performance goals of the planned system. This type of capacity planning and documentation of system requirements and project goals facilitates communication among the various stakeholders and allows for a common understanding of the background, expectations and success criteria for implementing the planned system. Furthermore, in order to develop even rough estimates of project cost, implementation effort and timeframe, system requirements such as numbers of users, types of system functions, complexity and number of reports, transaction throughput, data storage requirements and desired performance statistics must all be known. To that end, this section describes requirements for usage, scalability, reliability, and portability, based on extrapolations of known statistics from production systems currently processing disposition data in Arizona.

2.2 DRM USAGE EXPECTATIONS

The DRM will be used in two different modes:

1. As an on-line transaction processing system for the recording and maintenance of disposition reports and related data; and;
2. As a data warehouse for statistical reporting purposes.

In order to ensure adequate processing time and speed for the on-line requirements of the system, complete cycles will be moved into the data warehouse area of the system where the database schema will be optimized for query performance rather than throughput. The on-line transaction processing area of the system will be indexed and optimized for easy retrieval of active disposition reports, and will support the necessary Data Manipulation Language (DML) for processing and responding to incoming transactions.

As with the development of any new system, one of the most critical parts of the design is ensuring that it meets the expected sizing and performance requirements. Therefore, an essential part of the system design is the quantification of these expectations and performance goals. The types of metrics that should be identified and quantified include:

- Expected user load and connection means
- Number of transactions to be processed
- Acceptable response times
- Required system availability
- Acceptable batch processing time
- Expected record counts and growth rate
- Impact of auditing on storage requirements

2.2.1 Expected User Load and Connection Means

The Arizona criminal justice community is made up of 480 criminal justice agencies serving fifteen counties. The agencies that will interact with the DRM include law enforcement, courts, prosecutors, jails, and corrections, as well as State central repository staff. Communication with the DRM will be for on-line reporting, batch reporting, and inquiry purposes.

The usage estimates in this document are based on historical volume data from the AZAFIS and ACCH systems combined with incarceration and release statistics from the Department of Corrections. As there is no historical data available for some DRM functions that are currently not available in any existing system, assumptions have been made based on feedback gained during JAD sessions regarding the level of expected end-user activity in the new DRM system and extrapolated to give some indication of expected usage patterns. Statistics and assumptions used for the following usage estimates include:

- 443,568 AZAFIS transactions in 2003
- 209,612 arrests recorded in ACCH in 2003
- 462,341 charge counts recorded in ACCH in 2003
- 825 ORIs listed in AZAFIS
- 217,392 IQ and FQ transactions processed in 2003
- 2987 AZ CJIS devices
- 8505 registered users of AZ CJIS
- Assume eight disposition updates per charge (average two per agency)

For purposes of interaction with the DRM, at least 5,000 total registered users (that is, users with passwords and authority to use the system) are expected to access the system, with several-hundred concurrent users at any one time. In order to support the expected volume of users, the DRM must have the ability to accept both dedicated and shared server connections. By enabling shared server connections, a system can support a greater number of individual users, as groups of users can share a single connection to the server, utilizing that connection only when their individual session is active.

2.2.2 NUMBER OF EXPECTED TRANSACTIONS

The DRM is anticipated to manage around 500,000 criminal cycles per year. Each cycle may receive updates from various agencies, as they add their segments to the cycle. At an average of 8 updates per cycle, this equates to 4 million updates (or incoming transactions) to the DRM per year. Based on current AFIS Type 01 transactions and charges initiated in ACCH, it is estimated that 50% or more of the transactions supported by the DRM will be through interfaces with other criminal justice agency systems; all transactions (either web-initiated or batch interface) will follow the same path within the system for validation and processing. Besides processing of transactions for updates, it is estimated that the DRM will handle at least two-thirds as many query transactions. Having an enterprise system capable of supporting both the on-line transaction processing needed by the DRM as well as the projected data warehousing reporting requirements will be essential to the success of the system. In addition, a Relational Database Management System (RDBMS) that handles concurrency of transactions efficiently will also become paramount to the DRM's overall success.

2.2.3 ACCEPTABLE RESPONSE TIMES

One of the largest concerns by the criminal justice community about dispositions is the lack of ability to determine the status of a disposition. Part of this is due to the manual, paper-centric means in which disposition states are communicated between agencies, making access to the information at any point in the process extremely difficult. Another problem is the inability to immediately correct any errors with a reported disposition and to resubmit that disposition to the ACCH. While the DRM will need to support batch interface communication of disposition information, it is anticipated that most of the transactions will be submitted and processed in real-time, with an expectation of real-time response. The DRM must be able to support sub-minute turnaround of transactions in order to provide the timely feedback necessary to attain high accuracy levels, and availability of "up-to-the-minute" status on cycles and charges.

2.2.4 Required System Availability

Due to the nature of the data that will be processed in the DRM and the client base that it will serve, the DRM must have 24x7 availability. It will be a policy decision within the DPS data center that balances the end user need for maintaining true 24x7 availability against the cost of implementing a fault tolerant, highly available system. The technology exists today to maintain redundancy and fail-over capability for each component in the DRM, however, it will require careful analysis of available hardware and software platforms during the detailed design phase to determine the most appropriate balance between project budget and guaranteed system availability

2.2.5 Acceptable Batch Processing Time

While much of the interaction with the DRM is anticipated to occur in real-time, updates of disposition data will continue to occur in batch mode. The batch processing of disposition data must not interfere with on-line processing needs and requests for information. Therefore, a batch-processing window must be scheduled for a time other than peak usage times of the system. Typical enterprise-level batch processing windows are two-to-four hours per day.

2.2.6 Expected Record Counts and Growth Rate

The DRM system is expected to handle at least 500,000 criminal history cycles per year. Based on the assumption that it will take an average of three years for a charge cycle to go from initiation to the processing of final disposition, this gives an estimate of 1,500,000 records after 3 years in the active database, and a growth rate of 500,000 records per year in an archive data warehouse. As some charges initiated in the DRM may wait longer than three years before association of fingerprints, we have assumed a growth rate of 5% of initiated charges per year after the first three years of operation for the active charge cycles database. This estimation would mean that the DRM would have 1.8 million active charge cycle records in its eighth year of operation and would have statistical data on another 2.6 million charge cycle records that will have been passed to ACCH.

One means to handle anticipated record growth is through the archival or partitioning of complete cycle data into a data warehouse-type environment for statistical reporting purposes. In a data warehouse, data tables are usually structured to optimize access and analysis, as opposed to transactional throughput. In the case of the DRM, charge cycle records that have been completed and reported to the ACCH will be moved from the active records database to ACCH; retention time within the DRM has yet to be determined, but the system must be capable of changing to meet retention policies over time. In addition, requiring that the hardware and software solutions for the DRM support hardware clustering will guarantee that the system can adequately accommodate future growth.

Due to the audit requirement of transaction logging and notification processing, the record volumes in the notification area of the database will grow at a substantially greater rate. This rate could potentially reach around ten or more times the rate of the number of cycles within the DRM, depending upon required notifications and the extent to which agencies use the subscription features of the system.

The DRM system, as recommended, is intended to be a day forward implementation. In a "day forward" system, the transition from an existing system (paper yellow sheet) to a new system (DRM) is based on the initiation date of a record relative to a chosen "go live" date for the jurisdiction. ***Every*** disposition record initiated in a particular jurisdiction after its chosen "go live" date will be processed in the new (DRM) system, while records initiated prior to the "go live" date continue to be processed in the existing (paper yellow sheet) system. In a "day forward" system, no existing records from the existing system are loaded into the new system.

2.2.7 IMPACT OF AUDITING ON STORAGE REQUIREMENTS

The order of events that occur within the DRM will be important information for the tracking of progress against a cycle. In addition, all notifications that the system sends will also need to be preserved as an audit trail. Therefore, the DRM will need to include adequate storage capability to handle not only the current state of dispositions, but also an audit trail of how the dispositions have changed, by whom, and when. The hardware must support storage, both on-line and in warehouse mode, of the notifications and data necessary to support at minimum 500,000 criminal events per year.

2.3 SCALABILITY

General system architecture decisions must be based on not only the current but also the future needs of the DRM, with special consideration to the placement of this solution in the overall integration of criminal justice data in the State of Arizona. Scalability is the ability of a system to grow as the data storage and performance needs grow in a graduated means. In order to ensure that the DRM solution is scalable, one of the best options is the implementation of hardware platform clustering.

Clustering is the implementation of a hardware configuration that allows multiple computers to be connected in a network and share the workload, including both the data storage and the processing needs of the system. Where a single server eventually loses its ability to scale vertically (through the addition of CPU's or RAM), a clustering solution allows the enterprise to scale horizontally. In other words, two computers can be used to split the necessary CPU, memory, and data storage workload and become more powerful than a single server. Designing the DRM in a modular fashion ensures system scalability because parts of the application (modules) can be offloaded to additional servers. By simply purchasing servers that have clustering capabilities, the processing power of the DRM can be expanded, and the DRM investment is protected from size obsolescence, without having to purchase today the processing power or storage that may not be necessary until sometime in the future.

2.4 RELIABILITY

Criminal justice information is a set of data that has one of the most stringent requirements for reliability. Therefore, the DRM must be supported by a system architecture that provides zero-tolerance for loss of data. One means to provide such reliability is through the implementation of clustering. By definition, clusters are highly available platforms. This is because as long as one of the servers in the cluster is still performing, the application remains available.

Other means for ensuring the reliability and availability of the data include implementation of a guaranteed messaging solution that will track and log the outcome of each transaction that is sent to or from the DRM to ensure that none are lost and that each is delivered one time and only one time.

2.5 PORTABILITY

The DRM will be created in a Service Oriented Architecture (SOA) to provide for portability, flexibility and extensibility. The goal of SOA is to define software components in terms of their inputs and outputs rather than the internal data processing. In such an environment, individual software components, or modules, may be implemented in any manner that provides the required services to calling software modules. SOA allows for implementation of several software modules on a single hardware server when processing demands are low, such as at the beginning of project implantation, and the migration of modules to additional servers, new platforms, or even new programming environments when the system needs to be expanded either due to increased workload, or the need for additional functionality.

3 RECOMMENDED SYSTEM ARCHITECTURE AND SYSTEM COMPONENTS

An expandable and extensible solution, based on emerging trends related to data exchange, information sharing and integration in the criminal justice community is recommended for implementation of the DRM. These emerging trends include the implementation of standardized XML schemas in a Service Oriented Architecture (SOA), using a Business Process Management (BPM) approach to system design, as well as utilization of the XML-based Business Process Execution Language (BPEL), to create web services supporting a complete business process application. By incorporating these technologies in the recommended design, the DRM solution offers the greatest flexibility for the future growth and management of the system. This approach to Information Technology projects is driven from the business perspective of the enterprise, so that the business processes and information flows defined by the organization determine how and when data is processed within the system.

In order to meet the DRM design requirements identified in the Conceptual Design Document for the system, the DRM architecture will need to support certain architectural concepts as well as adhere to certain architectural constraints based on policies and technology. These concepts and constraints are discussed in the sections that follow.

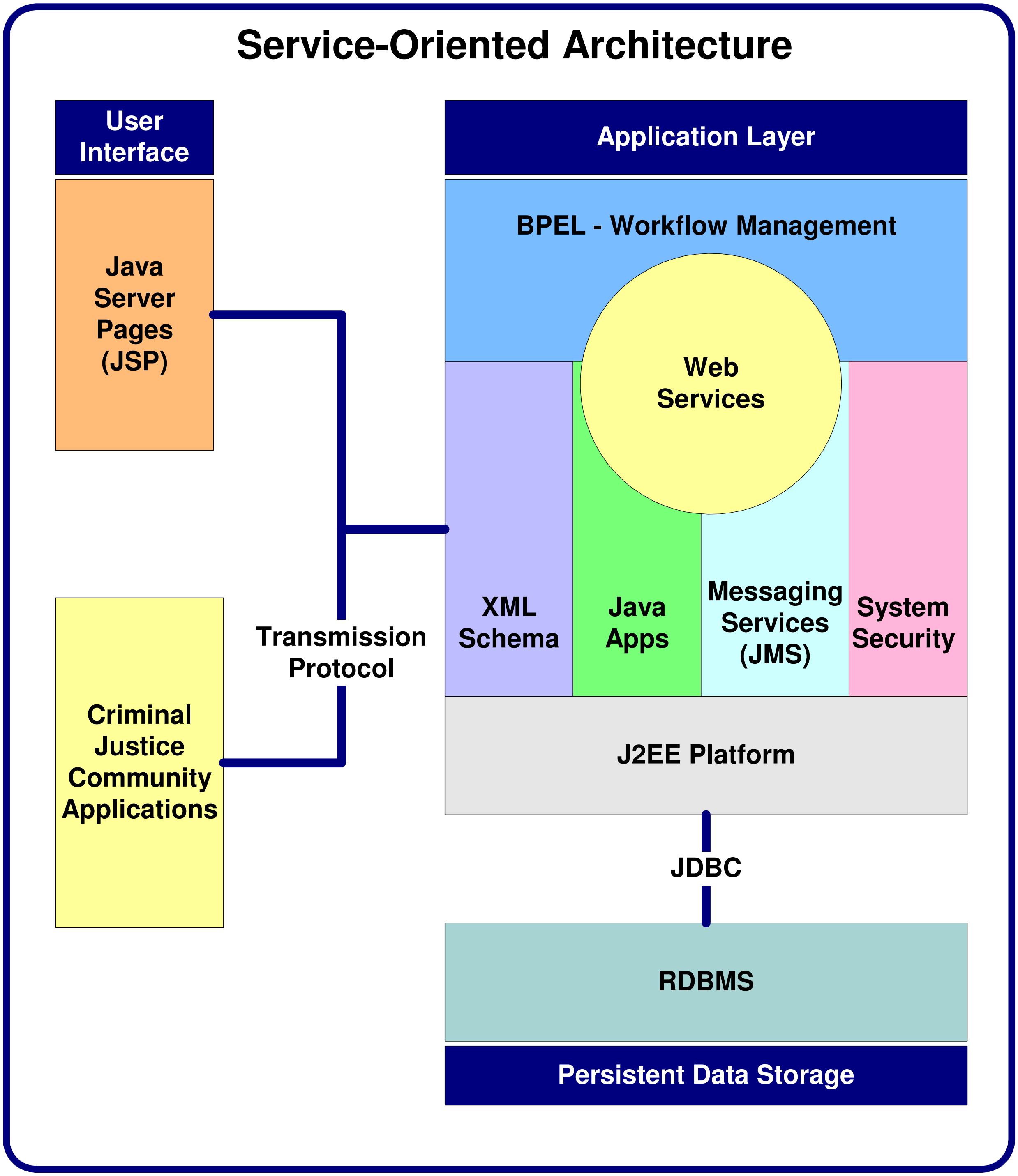
3.1 INTEGRATION FRAMEWORK

The System Architecture diagram represented in Figure 3.1 is a graphical depiction of the combination of technologies that are inherent in the DRM solution. These technologies are the best the industry has to offer and provide a solid base for the future integration efforts of the State of Arizona Criminal Justice Community. The major categories of technology that will be used to implement the DRM Service Oriented Architecture include:

- Service Oriented Architecture (SOA)
- Interfaces Layer
  - Java Server Pages (JSP)
- Applications Layer
  - Java 2 Platform, Enterprise Edition (J2EE)
  - XML Schema Definition (XSD)
  - Java Applications
  - Java Messaging Service (JMS)
  - Java Database Connectivity Connector (JDBC)

  - System Security
  - Web Services
  - Business Process Execution Language (BPEL)
- Data Persistence Layer
  - Relational Database Management System (RDBMS)

The following sections outline the role these technologies play in the implementation of the DRM.


Service-Oriented Architecture
User Interface
Java Server Pages (JSP)
Criminal Justice Community Applications
Transmission Protocol
Application Layer
BPEL - Workflow Management
Web Services
XML Schema
Java Apps
Messaging Services (JMS)
System Security
J2EE Platform
JDBC
RDBMS
Persistent Data Storage

3.1.1 Service Oriented Architecture (SOA)

3.1.1.1 Introduction to SOA

To achieve the system integration desired, the solution must be configured to share information in a real time environment among dissimilar systems. Over time, there have been varied and evolving approaches taken to address information sharing between disparate information systems. Early approaches implemented point-to-point interfaces, where each system would define and maintain a unique, custom interface between every other system with which it interacted. As the number of interfaces increased this approach grew very cumbersome and expensive to maintain. More recent approaches to integration have attempted to reduce the cost and complexity experienced with point-to-point interfaces. These attempts have led to approaches such as Enterprise Application Integration (EAI), business-to-business (B2B) and Service-Oriented Architecture (SOA).

While EAI and B2B are viable integration frameworks, we believe that the SOA framework offers the most flexibility and overall integration benefits. Therefore, the recommended DRM solution is constructed using the Service-Oriented Architecture (SOA) framework. SOA creates an environment where one computing entity performs a discrete task on behalf of another computing entity. The task, or unit of work, is called a ***service***.

The protocol for requesting a service is defined through the use of a description language. Each interaction is self-contained. The interactions are also loosely coupled, so that each interaction is independent of any other interaction. While these interactions often occur using XML-based messaging formats such as SOAP and communicate through use of Web Services, SOA architecture can be achieved without these constructs. For example, a legacy application may not have the ability to support a Web Services deployment, but it may support a JDBC connector and SQL. In this instance, a wrapper service for retrieving data could be constructed. This wrapper service would perform the data conversion into XML, and deliver the new document to the DRM. Response message would also go through a wrapper service to convert them back into the legacy application's native format. It is this independent management layer between the computing entities that provides the flexibility, reusability and cost savings provided by the SOA environment.

3.1.1.2 Why a Service-Oriented Architecture?

With the emergence of second-generation web services, and with the impact of XML standards on the way data is shared, technology has served to pave the way for true enterprise integration. Service-Oriented Architecture (SOA) has become the best-suited framework for building an architecture that is flexible, agile, and sophisticated enough to take advantage of the new technologies while also leveraging legacy environments. This architecture builds on the Business Process Management (BPM) approach to system design. The BPM methodology recognizes the importance of having the business processes drive both the means in which the system operates and the critical data exchange points. This approach, when applied to new system development advocates defining the business processes as the first step in the design process. Once the business processes are defined, they are then modeled in a BPM tool. It is only after the modeling of the processes that these models are passed to the development team to implement the processes as distinct system functions or programs.

By starting at the business process level, individual module development now takes a global approach, whereby generic functions are written that can apply to several different business processes. Each distinct business function becomes a stand-alone component of the system. This builds into the system the flexibility to change business models or change the workflows, by putting the processes together in alternate configurations without having to change the underlying program source code.

The BPM design concept falls into the realm of Service-Oriented Architecture when each of the distinct functions of the system is exposed as a web service. Through utilization of the XML-based Business Process Execution Language (BPEL), the web services are connected to create the complete business process, otherwise known as "orchestrating" the web services.

The result is a design that offers the greatest flexibility for the future. It forces IT projects to be driven from the business perspective, and in essence allows the business processes and information flow defined by the organization to drive how and when data is processed within the system. It also allows the state to make changes to systems as requirements change over time.

This type of architecture lends itself well to system integration efforts, because of the flexibility it offers. It provides a means to separate the interfaces themselves from the implementation of those interfaces, defining information exchanges utilizing XML standards, such that systems are not dependent upon proprietary code in order to be able to exchange information. With XML, the metadata is available with the data; thus, specifics on the underlying systems that utilize that information become less and less important.

With the future integration goals of the ACJC in mind, it is critical that the initial investment in technologies and system architecture for the DRM reflect not only the needs of the DRM but also the foundation required for future integration efforts. The recommended architecture will help ACJC to meet both current and future integration needs, while providing the flexibility to utilize existing applications as well as those to be developed in the future.

3.1.2 SOA Components

The individual components that comprise a SOA are introduced in the sections that follow.

3.1.2.1 Java 2 Platform, Enterprise Edition (J2EE)

A critical component in the proposed DRM architecture is utilization of the J2EE (Java 2 Platform, Enterprise Edition) standard. The use of J2EE in the DRM is a recommendation of the ACJC Strategic Plan because of its portability to various hardware platforms and its growing popularity as a platform for enterprise applications. J2EE is a Java platform designed for enterprise scaled systems. It was designed by Sun Microsystems to simplify application development in a thin client, tiered environment. J2EE simplifies application development and decreases the need for programming and programmer training by creating standardized, reusable modular components and by enabling the tier to handle many aspects of programming automatically.

J2EE offers the following components and functionality:

- The Java Development Kit (JDK) is included as the core language package.
- Write Once Run Anywhere technology is included to ensure portability.
- Java Database Connectivity 2.0 (JDBC), the Java equivalent to Open Database Connectivity (ODBC), is included as the standard interface for Java databases.
- A security model is included to protect data both locally and in Web-based applications.
- Full support is included for Enterprise JavaBeans (EJB). EJB is

a server-based technology for the delivery of program components in an enterprise environment. It supports XML and has enhanced deployment and security features.

- The Java servlet API (application programming interface) enhances consistency for developers without requiring a graphical user interface (GUI).
- Java Server Pages (JSP) is the Java equivalent to Microsoft's Active Server Pages (ASP) and is used for dynamic Web-enabled data access and manipulation.

Because SOA is built upon loosely coupled and interoperable machine-to-machine interactions, J2EE provides to the solution a much-needed mature network-based platform.

3.1.2.2 XML Schema Definition (XSD)

The DRM architecture must make extensive use of XML Schema Definition (XSD) technology to ensure transactions conform to data elements contained in the Arizona Justice Data Dictionary, compiled by the XDD Subcommittee. XSD is a W3C recommended way to describe and validate data in an XML environment. Because XSD is also written in XML, XSD has great advantages over other XML schema languages because it eliminates the need for a parser. XSD created for the DRM will be based upon the GJXDM elements adopted by the XDD subcommittee and set forth in the latest release of the Arizona Justice Data Dictionary. The XSD will be used to verify that each item of content in a transaction adheres to the description of the element in which the content is to be placed.

3.1.2.3 Java Applications

The custom programming necessary to achieve the functionality required by the DRM will be written in the Java programming language. Java is an object-oriented language that runs in the J2EE provided Java Virtual Machine (JVM). Because Java is not operating system dependent, it offers a great deal of flexibility and portability when choosing both hardware and operating systems. JVMs exist for most operating systems, including UNIX and Windows. In addition bytecode (which is the compiled java source code files) can be converted into machine language instructions.

3.1.2.4 Java Messaging Service (JMS)

Reliable message delivery is very important for the success of Web services. It provides the structure for use of Web services over unreliable networks and the Internet, as well as the guaranteed messaging necessary for mission-critical operations such as real-time enterprise integrations. This type of reliability is the result of message receivers responding to messages with acknowledgements that let the service know the message was received. Lack of receipt of such an acknowledgement triggers the service to continue to try and deliver the message until either the acknowledgement is received or a predetermined number of tries have elapsed.

The goal of the JMS is to ensure delivery of a message "once, and only once" to the intended receiver. Key requirements of reliable messaging that are important to the successful implementation of the notification processes inherent in the DRM include:

- Delivery of messages reliably in support of business processes whose lifetimes commonly exceed the up times of the components on which these processes are realized
- Guaranteed quality-of-service including:
  - Ensuring that each message sent can be received exactly once (once and only once)
  - Guaranteeing that messages are received in the same order in which they were sent
  - Providing notification of failure to deliver a message to both the sender and receiver
- Flexibility to accommodate the nature of web services, such as mobility of a business process to different channels or physical machines
- Ability to support message transfer via intermediaries
- Leverage of the standard SOAP extensibility mechanism to achieve reliable messaging
- Enabling reliable messaging bindings to a variety of underlying reliable and unreliable transport protocols
- Support of security and other message delivery services

3.1.2.5 Java Database Connectivity Connector

The communications between the J2EE platform and the relational database will be via the standard Java database connector, JDBC. Use of the JDBC open standard API allows a loose coupling between execution of programming logic and access to stored data. The JDBC API provides for the passing of SQL statements to the database application for execution and return of requested data or result codes. This abstraction layer allows for easy access to multiple databases, possibly running on distinct hosts, or even different RDBMS applications without changes to the code in the J2EE platform. In the proposed DRM system architecture, this is particularly important since the archival data will be separate from the active records database. Use of JDBC in system implementation will allow for upgrades to the archival data with minimal changes to the configuration of the JDBC Connector and none to the executing program code.

3.1.2.6 System Security

Security for the DRM is focused on protecting the confidentiality, integrity and availability of the DRM and attached systems. The security infrastructure must address risks posed by malicious attacks, inappropriate or inadvertent use by registered users, and the protection of sensitive data while in transit and while in persistent storage. At a minimum, the DRM System Security addresses the following points:

- User identification and authentication
- Authorization policies and procedures
- Acceptable use policies
- Data classification and access control
- Protection of data communications and system interfaces
- Ability to audit all transactions
- Virus protection
- Detection of intruders
- Screen-saver passwords and automatic timeout for logins

3.1.2.7 Web Services

A Web service, as defined by the W3C (World Wide Web Consortium), is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet Protocols. Web services provide a standardized way of integrating web-based applications. Utilizing open standards such as XML for tagging data, SOAP for transporting the data over HTTPS, WSDL for providing description of the data and UDDI for listing what services are available, the Web service allows for the communication between disparate systems, without having to first acquire intimate knowledge about these systems. Web services should be used in the DRM solution to expose certain services to interface partners.

3.1.2.8 Business Process Execution Language (BPEL)

The DRM will need to support workflow requirements. A common means for this is the utilization of Business Process Execution Language (BPEL). BPEL for Web Services is an XML-based language designed to enable task sharing for distributed computing. This is accomplished by utilizing a combination of Web Services and other common connectors (JDBC, ODBC, etc). Using BPEL, a programmer describes a business process that will take place across the enterprise environment in such a way that any cooperating entity can perform one or more steps in the process in the same way. The BPEL will be used to orchestrate the performance of these critical tasks in order to ensure the necessary data exchange occurs:

- Management of business rules for processing in the application
- Orchestrating the transaction flow between the various systems
- Maintaining transaction state across multiple applications and user interfaces for the period of time necessary to complete a process instance (the period of time might last from seconds to months depending upon the business requirements)
- Transformation of the transaction structure to the appropriate format (XML, flat file etc.) for the application targeted to received it
- Transport of the transaction to the target applications using a Web Service, JDBC, ODBC, or a custom built API as required by the participating systems
- Utilization of a Java Messaging Service (JMS) to ensure guaranteed message delivery between applications

3.1.2.9 Java Server Pages (JSP)

The user interface component of the DRM will be constructed as JSP web pages. JSP is a technology for controlling the content or appearance of Web pages through the use of servlets. Servlets are small java programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it.

3.1.2.10 Relational Database Management System (RDBMS)

The DRM must utilize a robust RDBMS to perform data storage and management tasks. Because communication between the application layer and the database layer will follow the JDBC standard, this RDBMS must support standard JDBC connections. The physical database(s) may reside on one computer or on multiple computers, providing a high level of availability and failover. The RDBMS is an integral part of the DRM solution and must provide the following features:

- Referential Integrity
- Transaction Control and multi-version read consistency
- Fail over Support
- Load Balancing/Clustering
- High-volume on-line transaction processing support
- Query-intensive data warehousing support
- Security assurance
- High availability

3.2 DRM SYSTEM ARCHITECTURE COMPONENTS

The system architecture recommended for the DRM complies with Service-Oriented Architecture design principles recognized by the IT community as building blocks for system integration. This design will enable the Arizona criminal justice community to take advantage of new technologies for the sharing of data, while also leveraging existing legacy components that are crucial sources of information. The DRM architecture will exist within the DPS data center infrastructure and must adhere to the standards and policies imposed by DPS including: server hardware, operating systems, supported software, application review, network security, performance, safety, and module reuse.

The recommended architecture and its components are executed in a mid-range server platform with support for web-based and asynchronous messaging (batch) interfaces. The DRM application links several existing criminal justice systems that currently play a role in the process of disposition reporting. The DRM also prepares for future integration initiatives by establishing a transport methodology and standardized schema for data exchange. The recommended architecture consists of four layers each containing related functionality. The diagram below shows these layers as a Partner Systems layer, an Interfaces Layer, a Business Services Layer and a Database Layer. .

To provide the greatest flexibility, some DRM services may be executed on separate hosts. For example, the database management system and the reporting service may reside on separate mid-range server platforms so as to minimize the effects of the report-processing load on the system response time for online and real-time application use. As the use of the DRM system grows and the demand for computing resources grows, the system may be easily reconfigured to add computing resources by migrating one or more of the business services to the additional computing platform.

Implementation of the DRM will allow criminal justice agencies to bridge the gap between islands of information and unify fragmented business processes to create an environment where multi-agency, multi-system integration can exist. Examples of this functionality include:

- The ability to maintain transaction states over the length of the disposition reporting process from the charge-initiating event to final disposition reporting.
- The re-use of existing functionality in multiple enterprise business processes by using Web Services to expose this functionality as a discrete service and use Business Processing Execution Language (BPEL) to orchestrate the transaction flow between these services
- The ability to capture complex business rules as BPEL and thus leverage these rules to determine proper transaction handling
- Transformation of the transaction structure; as a transaction flows through the DRM, it may be transformed several times between file structure standards (e.g. flat file, fixed width, XML, NIST EFTS) to accommodate the various source and target applications for generating and processing the transactions when the applications are unable to accommodate the standard, enterprise transaction structure.
- Translation of the data element values; as a transaction flows through the enterprise, the data values contained within the transaction may be translated in order for the application receiving the transaction to process the data when the application does not utilize the standard, enterprise data dictionary.
- Transport of the information between the applications in an open standard format (e.g. SOAP, FTP, HTTP, etc).

We recommend that national standards related to criminal justice data exchange and information sharing be incorporated into the integration services of the DRM. These include:

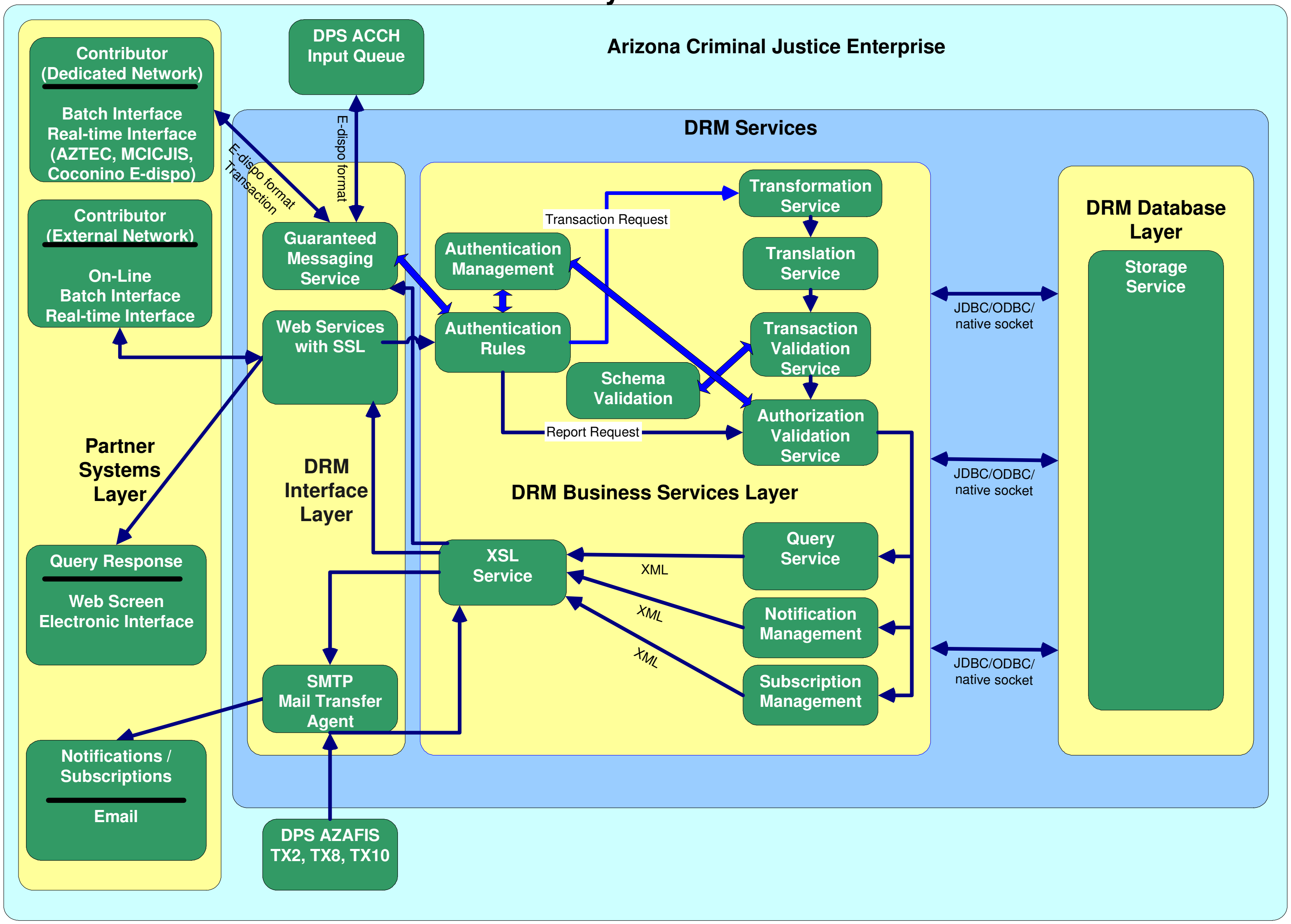
- The business rule-based communication as promoted by SEARCH and the nationally recognized Justice Information Exchange Model (JIEM).
- Utilization of the Global Justice XML Data Model (GJXDM) Version 3.0 (or subsequent revisions) as the basis for the enterprise schema used in XML transactions.
- Conformance to the Regional Alliances for Network

Infrastructure and Security (RAINS), Open Specification for Sensitive Information Sharing.

Future interfaces to the DRM will take advantage of the SOA and utilize either of the available web service input methods (HTTPS or Guaranteed Messaging) and the published XML format for transaction data (an extended version of GJXDM). In this manner, future interfaces will reuse the existing web services of the DRM. Enabling the DRM to communicate with a new SOA compliant interface will require a minimal amount of configuration changes to code and control tables in the DRM.

The diagram that follows is representative of the system components necessary to support the recommended architecture. Each of the components of this model is described and its function explained in the sections that follow.

DRM – System Architecture



3.2.1 PARTNER SYSTEMS LAYER

The Partner Systems Layer consists of all initial and future systems that will interface with the DRM. At initial implementation the partner systems will be the current automated systems performing electronic disposition reporting to ACCH as well as a web client interface for direct end-user processing of charge cycle events. Also shown in the Partner Systems Layer is the receipt of output data from the DRM whether that output is a requested query, a notification event, a scheduled scorecard report or other electronic data processed through the DRM.

3.2.1.1 Contributor (Dedicated Network)

The Contributor (Dedicated Network) function is a transaction-based asynchronous interface, allowing an agency to communicate to and from the DRM system. This function is specific to systems that have dedicated network connections to the internal DPS network, and therefore do not need to have VPN, SSL or other network security measures added to protect sensitive data while in transit on the network. This includes the interfaces to the current electronic disposition reporting systems: the Administrative Office of the Courts' AZTEC, Coconino ICJIS and Maricopa ICJIS. This function communicates with a Guaranteed Messaging Service within the DRM Application Server to deliver disposition data and receive return messages using the DPS-defined format for electronic disposition reporting. The DRM will verify and process the transaction before forwarding an appropriate transaction to the ACCH input queue.

3.2.1.2 Contributor (External Network)

The Contributor (External Network) function is either a web-based interface or a transaction-based interface, either real-time or asynchronous, allowing an agency to communicate to and from the DRM system. For the web-based interface, the browser will communicate via a HTTP(S) request to the DRM web server for transaction requests. The web server will take in the request and deliver the web page based on the request. For message-based interfaces, the agency system will communicate to the DRM message server via a defined XML message used to request/receive a transaction. This interface will be implemented in the SOA as a Web Service, so that a requesting agency may implement its own functions to interact with the DRM.

3.2.1.3 Web Browser Client

The user interface to the DRM for online real-time transactions will be a standard web browser client. Since the DRM contains sensitive data and will be available through the public Internet, the web browser client should be a "domestic version" web browser client capable of using 128-bit SSL connections. Any hardware device capable of making a network connection to the Internet and running a domestic version browser may be used as a client to the DRM.

3.2.1.4 Query Response

The Query Response function is a web-based or transaction-based interface that controls the communication to the agency when a notification is needed. The agency will be able to subscribe to a particular notification and choose to be notified via a web-based interface or via a messaging interface. If subscribing occurs via a messaging interface, the DRM messaging server will be used to provide the notification message.

The Query Response function is the recipient of report requests and transaction notifications that may contain sensitive criminal justice information. As the system architecture diagram shows in Figure 3.2, DRM data can be delivered to the Receiver function through an SSL-enable web server connection (HTTPS) or a VPN connection to the recipient's service client.

3.2.1.5 Notifications / Subscriptions

The Notifications / Subscriptions function provides the network facility for delivery of DRM output through e-mail to the recipient's configured e-mail account. While HTTPS or VPN in the Query Response function will protect sensitive data while on the public Internet, standard SMTP e-mail may not. It will be a matter of policy in the administration of the DRM whether to allow sensitive data (reports and messages) to be transported by e-mail at all, or possibly, only notifications that contain no sensitive data will be allowed. It is recommended that DPS implement a policy (and associated DRM business rule) that will only allow sensitive data to be sent to e-mail addresses that can be reached either completely through the internal DPS network, dedicated network link, encrypted network link (such as with IPSec), or protected with a public key encryption system for e-mail (such as PGP or S/MIME). It is also recommended that DPS initiate attributes in its security management service (LDAP) to support public key cryptography exchanges with each registered user of DRM.

3.2.2 INTERFACES LAYER

The Interfaces Layer consists of the services that provide network transport endpoints to clients in the Partner Systems Layer and the Business Services Layer. The Interfaces Layer provides a guaranteed message delivery queue to communicate with existing electronic disposition reporting systems, an SSL-enabled web server to provide real-time transaction processing to web browser clients and a batch transaction queue for future integration initiatives. The Interfaces Layer also provides the SMTP mail transfer agent for processing the input queue from AZ AFIS and for processing the e-mail gateway for notifications from DRM to end users.

3.2.2.1 Guaranteed Messaging Service

The guaranteed messaging service in the DRM will bridge between the electronic disposition reporting systems (AZTEC, Maricopa ICJIS, Coconino County ICJIS) and the existing ACCH input queue implemented using IBM MQ Series. The DRM guaranteed messaging service will copy and replicate all input data on the current ACCH input queue to the ACCH while it determines the applicability of input data to cycles being tracked in DRM. The guaranteed messaging queue will provide an alternative network transport methodology for future partner systems that are connected directly to the secured intranet of DPS. Future partner systems that are not directly connected to the DPS intranet will need to use the Web Server with SSL transport in order to provide data security appropriate for sensitive data on a public network.

3.2.2.2 Web Server with SSL

The web server host is the DRM application layer available on the publicly accessible Internet. This host is responsible for accepting web page requests, passing transaction requests to the Application Server and delivering the DRM web pages and return transaction web screen to the client web browser. The Web Server will have an IP address in the DPS public network demilitarized zone (DMZ) and a domain name used by the browser to access the Web Server. Connections made through the Internet will be encrypted and secured using Secured Sockets Layer (SSL) encryption module.

The mid-range server platform hosting the web server may also be enabled with VPN software to allow asynchronous or batch submission of transactions through an encrypted “tunnel” on the public Internet. The receipt of transactions will be implemented in an SOA architecture as a Web Service so that agencies may interface their own legacy information systems to the DRM through this transaction gateway. An agency's legacy information system may be interfaced in a real-time manner to the DRM by emulating a web browser client and submitting transactions through the web server in the same manner and format that a web browser would. It will also be possible to interface a legacy information system by submitting batch transactions in a specified XML format. In the latter case, the web server platform provides an encrypted network transmission for the data file between the agency system and the DRM. Batch transactions submitted in XML format will be verified using a Schema Validation function.

3.2.2.3 SMTP Mail Transfer Agent

The SMTP mail transfer agent provides the network communication endpoint necessary for receiving TX messages from the AZAFIS system. This same SMTP mail transfer agent will provide the gateway for processing DRM notifications messages and report outputs that have been configured to be delivered by e-mail to and end user. The DRM will be configured to not allow delivery by e-mail of confidential data except to addresses marked as security-enabled, such as an internal network endpoint, or an address set-up to communicate with DRM using a public key encryption system. It is recommended that the Authentication Management Service implementation providing end-user credentials also include data attributes required for a public key infrastructure and processing of secure e-mail on a public network.

3.2.3 BUSINESS SERVICES LAYER

The Business Services Layer provides the business logic components used to execute the required business logic, such as input data translation, user authentication, workflow management and message routing. The Business Services Layer accepts transaction and reporting requests received through the Interfaces Layer verifies the end-user authentication, the authorization for access to the requested data view or manipulation, provides for access to the Database Layer and queues output messages and data for delivery through the Interfaces Layer. It should be noted that these business services are loosely coupled so that they can be reused for other enterprise efforts.

3.2.3.1 Authentication Rules

The Authentication Rules function interacts with the DPS Security Management Service (an LDAP server) to verify authentication credentials and to deliver the rules necessary to authorize the execution of transactions and provide user roles for verified access to the information contained in the DRM data store. To avoid duplication of effort within DPS and to maintain consistency of access levels across DPS applications, it is recommended that the DRM utilize the common security management service in use within DPS. As DPS is currently in the process of migrating this access control function from IBM RACF to a standard LDAP web service, the recommended architecture shows the DRM utilizing this shared LDAP.

In the System Architecture diagram, the DRM utilizes a security architecture that implements user-based access to specific DRM functions, transactions, reports, database entities and notifications. User credentials, access control matrices and role definitions will be verified through standard queries to the LDAP service. This security management function may be implemented inside of the DRM Services group as an independent user access control module; however the preferred configuration utilizes a common security management service within the DPS data center.

3.2.3.2 Schema Validation

The Schema Validation function is used to store standardized XML schemas, including the ACJC-adopted XML schema based on the national GJXDM standard. This function will also store an XML schema created to structure requests for data, DRM administrative functions, and user authentication.

3.2.3.3 Transformation Service

The Transformation Service function reads input to the DRM in known formats and uses XSL to transform the input data into the standard XML transaction formats used in the following DRM processing functions. Because of the DRM's SOA architecture, a change in input format, or expanding the DRM to read a new input format would require changes to the XSL routines in this function only.

3.2.3.4 Translation Service

The Translation Service function handle the business and process logic necessary to perform the requested transaction. This layer also provides for data abstraction of incoming transactions. It is in this layer that transactions may have data codes translated through look-up tables, may have the data format transformed, or have other business rules applied in order to have properly formatted transactions submitted to the DRM internal transaction queue. This layer will process both web-based requests as well as message-based requests to ensure that the same business logic is applied to both types of requests. Code look-up tables will be used to translate between data codes of legacy systems and data codes in use for DRM and output to other external systems. This includes access to the DRM interagency index for linking data between disparate legacy systems and for processing charge cycle disposition queries based on the case or person identifiers in use at the requesting agency. For example, the Translation Service allows a law enforcement agency to search for a person based on a booking number, while a prosecutor's office might search based on a case number, and a court may search based on a name or known alias.

3.2.3.5 Transaction Validation

The Transaction Validation function is the layer in which the XML-formatted transaction is validated for input and output to or from the DRM. The validation of disposition reporting data transactions will occur based on the DRM implementation of the GJXDM standard. Messages sent to the DRM that are rejected will have an error condition message sent back to the calling agency.

As shown in the System Architecture diagram in section 3.2, XML transactions are validated against a schema validating web service in the DRM Services infrastructure. This design feature promotes consistency of data elements and structures among production criminal justice information systems within the DPS data center as a whole. If a common schema validating web service is not in production within DPS at the time of detailed design and implementation, this web service should be implemented within the DRM Services infrastructure so that queries to this service may be redirected to a DPS common service when it becomes available.

3.2.3.6 Authorization Validation

The Authorization Validation function provides a security check to ensure that a submitting user is authorized to perform the requested action or to access the requested data in order to process the properly formatted and validated input transaction. User validation will be performed through standard requests to the security management service in the DPS infrastructure. These verification requests are made to a common LDAP server in the DPS data center infrastructure.

3.2.3.7 Query Service

The Query Service function processes query and report requests that are submitted to the DRM. This report function could be provided by a third party package such as Crystal Reports or a custom report module developed for the DRM. This function must handle ad-hoc report requests for data retained in the DRM Data Store. In the diagram above, query and report requests are processed through the Authorization Validation service to verify that the submitting user is assigned an appropriate role and has been authorized to access the requested data.

For web-based report requests, the user may request that the report be processed and returned to the web browser client. Alternatively, the user may submit the report to be run as a background process and returned to the user's configured e-mail notification address, network printer or other network device capable of receiving the SOA standard messages and notifications of the DRM.

The Query Service will communicate to the data persistence layer (Storage Service) via the database native or JDBC/ODBC driver to process SQL queries and generate reports. The Query Service will forward all completed reports to the XSL Service for proper formatting and delivery to the appropriate network location.

3.2.3.8 Notification Management

The Notification Management function controls the notification events that will occur within the DRM system. The notification management system will interact with the DRM data store to retrieve notification preferences and message formatting. This function will maintain tracking and logging mechanisms for statistical reporting of message traffic and to find the delivery status of messages sent. The Notification Management function processes pre-defined reports, such as statistical reports (including the Agency Scorecard), when requested or on a defined scheduled and passes the output to the XSL Service for proper formatting and delivery to the appropriate network location. The Notification Management function will support both point-to-point messaging used for mandatory notifications and alerts between agencies as well as publish and subscribe messaging used for subscription-based notifications. For example, some law enforcement agencies may subscribe to a notification when the prosecuting attorney files charges in a particular court. Some events might cause escalation of notifications to other users, or cancellation of a notification suppression setting, such as when charge disposition data does not change within a certain timeframe. After this time period, these time-based events will be triggered and notifications sent. For example, a notification may be sent to a law enforcement supervisor if an arrest was made but no charge decision has been input to DRM by either law enforcement or any other agency.

3.2.3.9 Subscription Management

The Subscription Management function allows agencies to subscribe to the various event notifications and pre-defined reports published by the DRM. Each agency will have the ability to subscribe to certain notifications and reports within the DRM as they deem fits their internal business processes. Also, it is assumed that certain event notifications will be mandatory within the DRM according to business processing rules and will not be capable of being modified by the agency receiving the notification. For example, time-sensitive notifications may be subscription based for a period of time, but become mandatory within a certain time proximity to a known process deadline. Some pre-defined statistical reports, such as Month-end charge submission counts, will be available on a subscription basis as well as being available by request through the DRM online interface.

3.2.3.10 XSL Service

The XSL Service function accepts messages, notifications and reports from other DRM service modules and applies business logic (XSL) to apply proper formatting, message encapsulation, and network delivery according to the requestor's configured preferences. This function delivers online transactions to the web server module for delivery as a web page, sends notifications to the user's configured network location, and sends AZAFIS TX messages to the internal transaction queue.

Users will be allowed to configure a preferred network location to receive notifications and messages. This network location may be any connected device or service capable of processing the SOA message type. Some possible target locations include e-mail addresses, network printers, splash screens of the DRM web application, or asynchronous message queues for a DRM interfaced system located in the user's jurisdiction.

3.2.4 DATABASE LAYER

The Database Layer provides persistent storage for criminal cycle data as it is being collected before reporting to the ACCH. Additional data stored in the Database Layer will include indexing information that relates individual criminal cycles to local system identifiers, data modification audit trails, canned reports, and system management information such as user preferences and subscriptions and notifications configurations. Communication with the Database Layer may be established through an abstraction layer such as JDBC or

ODBC, or through the implemented database native communication socket.

3.2.4.1 DRM Storage Service

The DRM Storage Service is the layer in which the DRM data will be persisted (stored). This layer will consist of a relational database capable of transaction management through native read-consistency concurrency control. It should support both dedicated and shared server connections, hardware clustering, and should be capable of supporting both high-volume on-line transactions as well as query-intensive data warehousing application use. It will also contain the necessary native or JDBC/ODBC data access layer allowing the application server to communicate necessary data updates and data requests.

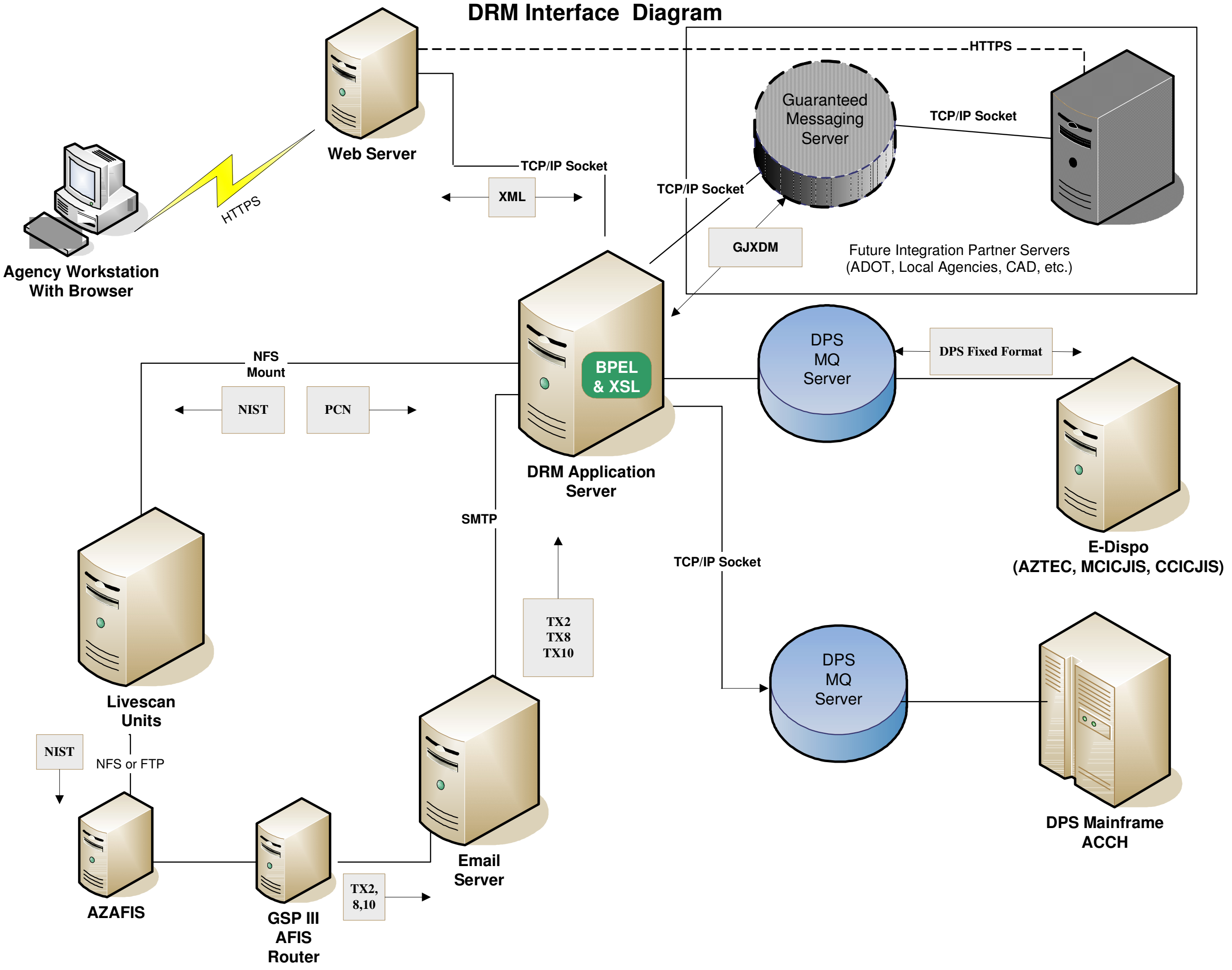
The DRM data store will also need to support data level constraints controlled by the security management function and will need to adhere to the J2EE security framework imposed by the application server. The data level constraints consist of the ability to make certain information sensitive as well as control the access of information based on certain settings, such as disposition status and archive status. The data access layer will reside on a different mid range server platform from the application server and will allow for application-native TCP/IP communication and JDBC/ODBC access from the services executing on the application server.

3.3 INTERFACE STRATEGY

The interface strategy for the DRM takes advantage of the system's recommended Service Oriented Architecture. At initial implementation, a browser based end user client will be created, as well as interfaces for legacy systems as described below. These interfaces will define data exchanges and transport methods to provide a framework of application interfaces that can be used to integrate future partner systems with the DRM. Any authorized system that is able to create and receive messages within this defined enterprise standard specification may be interfaced with the DRM.

Some types of systems that may be interfaced with the DRM in the future might include, systems at AZ DOT Division of Motor Vehicles, Human Services systems, transportation systems and police dispatch systems. Although the preference would be that these systems utilize the defined enterprise standard specification interfaces, it is also understood that some systems may need to have translations or transformations built into the DRM. The SOA design of DRM allows for this type of system extension with minimal effort.

The diagram that follows is representative of the system components necessary to support the DRM system interface requirements. Each of the interface components of this model is described and its function explained in the sections that follow.



DRM Interface Diagram

3.3.1 DRM TO LIVE SCAN UNITS

To minimize data entry duplication, an interface between the DRM and Live Scan Units will allow non-fingerprint supported charge data maintained in DRM to be populated into the data screens of a Live Scan. This interface will be designed to utilize the data import capabilities of the existing Live Scan units at the time of detailed design and implementation. Currently, Live Scan units support the import of NIST standard formatted records from an NFS-mounted disk drive. Newer Live Scan units also support access to records on an FTP host. To maintain compatibility with older Live Scan units, the current architecture design shows this communication via an NFS mounted drive.

During the booking process, the Live Scan operator will query the DRM system from a web client to locate possible charges existing in the DRM related to the person being fingerprinted. The Live Scan operator will enter the CCID into the Live Scan for the charges existing in DRM so that these charges can be associated with the fingerprints. The Live Scan unit will place a record query request on an NFS-mounted disk drive where it will be picked up by a daemon process running in the DRM and processed to return a NIST formatted transaction to the NFS mounted disk drive containing the existing DRM charge data. This NIST formatted transaction will be picked up by the Live Scan unit and will populate the arrest data fields in the Live Scan.

In cases where no charges exist in DRM at the time of fingerprinting, the assigned CCID will be reported from the DRM to the Live Scan unit through the AFIS interface.

3.3.2 AFIS TO DRM

The interface between DRM and AZAFIS is an extension of the existing Live Scan to AZAFIS interface where status messages are sent to an agency in response to submitted Live Scan transactions. For this interface, the DRM will be "copied" on appropriate messages sent from the AZAFIS to participating agencies. The anticipated messages that the DRM would receive are the TX2, TX8, and TX10 messages defined in the referenced document ***GSP III – Arizona Fingerprint Data Router Interface Control Document (ICD).*** The TX messages sent from AZAFIS are in MIME-encoded NIST format and transported via SMTP. A Mail Transfer Agent (MTA) will be implemented in the DRM system for receiving the messages from AZAFIS.

3.3.3 DRM TO ACCH

The interface between the DRM and ACCH will support two types of transactions in a two-way communication. Transactions that originate from the DRM will include disposition reporting and record modification requests. Transactions originating from the ACCH will include response transactions from disposition data submissions and record modification requests.

As charge cycles are completed in the DRM system, the DRM will submit the disposition data to the ACCH system using a fixed length record format defined in DPS document, ***Local Agency – ACCH Disposition Transaction User Specifications***. The DRM will use a Java Messaging Service to provide the transport of transaction message and control files to the existing ACCH queue currently used by DPS for electronic disposition reporting. . The DRM will submit all final disposition transactions to this existing queue. The DRM will also retrieve from the existing response queue the response transactions generated by ACCH for each submitted transaction and data queries.

3.3.4 DRM TO AGENCY SERVER

The DRM system will have the capability to interact with agency servers via XML transactions following the GJXDM standard schema and the augmented version to be adopted and published by the ACJC XML Data Dictionary Subcommittee. For agencies connected to DPS through the public Internet, these transactions will be accepted through an SSL tunnel to the application web server. For agencies connected directly to the DPS internal network, these transactions may be submitted to a queue on the existing DPS guaranteed messaging server. A service will exist within the DRM to read the GJXDM formatted message from the guaranteed messaging server via a TCP/IP socket connection. Each agency will be responsible for building the mechanism to interact with the SSL web service or with the guaranteed messaging Server.

The initial agency messaging server interface will include interfaces to the three current e-disposition projects: AZTEC, Coconino and Maricopa ICJIS. All three of these e-disposition projects currently submit transactions to the DPS guaranteed messaging service using the IBM MQSeries™ product. Upon implementation of the DRM, the ACCH will be configured to read from a new queue on the Guaranteed Messaging Server. The DRM will pick up messages from the existing queue, perform validation and update processing for each transaction and then submit the message to the new ACCH queue. Likewise, return messages from ACCH will be sent to a new queue where DRM will pick up the messages, perform its own updates and then transfer the return messages to the existing return message queue.