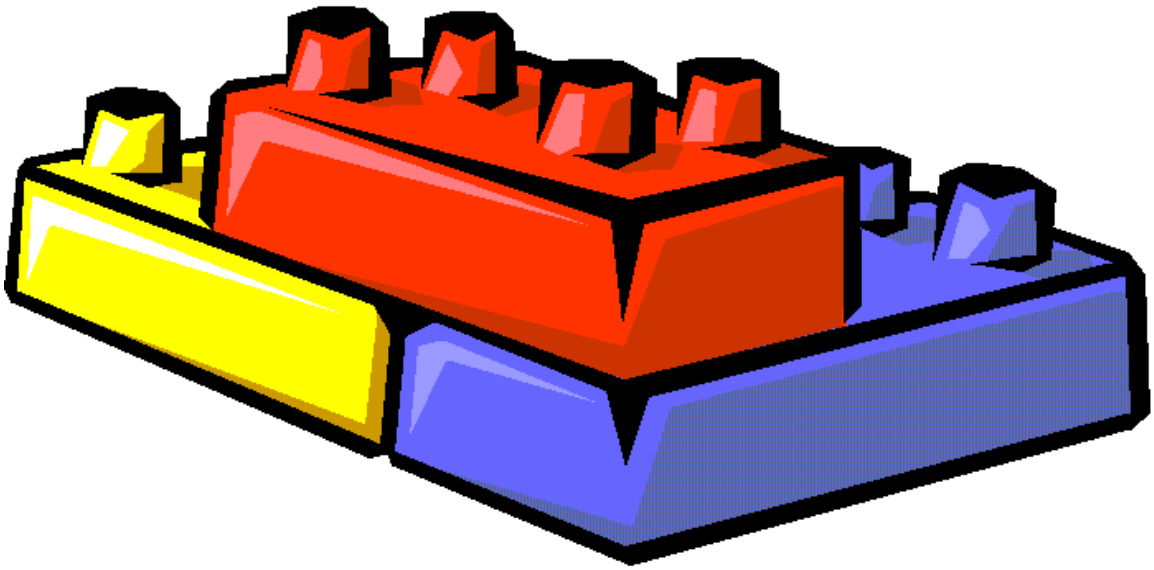
Integrated Justice Information System Architecture
Building Blocks Construction

May 2001



Prepared by Jim Threatte

Copyright © 2001 - Jim Threatte

CONTENTS

1 Executive Summary 1

2 Background and Introduction 2

3 Business Case for Integration 4

4 Alternative Integration Architectures 5

4.1 Alternative System Architectures 5
4.1.1 Consolidated System – Centralized 6
4.1.2 Coordinated System – Point-to-Point 6
4.1.3 Hybrid System – Hub-and-Spoke 8

4.2 Alternative Data Architectures 12
4.2.1 Data Warehouse Architecture 12
4.2.1.1 Data Warehouse Architecture Overview 12
4.2.1.2 Data Warehouse Architectures 13
4.2.1.2.1 Direct Query 13
4.2.1.2.2 Virtual Query 14
4.2.1.2.3 Single-Subject Warehouse 15
4.2.1.2.4 Mixed Workloads (Database Propagation) 15
4.2.1.2.5 Virtual Global Warehouse 16
4.2.1.2.6 Event-Driven Warehouse 17
4.2.1.3 Data Warehouse Architecture Evaluation 18
4.2.1.3.1 Client Software Alternatives 18
4.2.1.3.2 Alternative Approaches for Transferring Data 19
4.2.1.3.3 Advantages and Disadvantages 21
4.2.2 Data Integration Hub Architecture 22
4.2.2.1 Data Integration Hub Overview 22
4.2.2.2 Data Integration Hub Architecture 23
4.2.2.2.1 1. Physical Data Storage 25
4.2.2.2.2 2. Data-Access Servers 25
4.2.2.2.3 3. Gateway 25
4.2.2.2.4 4. Native Data Layer 26
4.2.2.2.5 5. Metadata Catalog 26
4.2.2.2.6 6. APIs 26
4.2.2.2.7 7. Users, Portals, and Other Systems and Applications 26
4.2.2.3 Data Integration Hub Architecture Evaluation 27
4.2.2.3.1 Client Software Alternatives 27
4.2.2.3.2 Alternative Approaches for Accessing Data 28
4.2.2.3.3 Advantages and Disadvantages 29
4.2.3 NASIRE Document Exchange Architecture 30
4.2.3.1 NASIRE Document Exchange Architecture Overview 30
4.2.3.2 Document Exchange Architecture 31
4.2.3.2.1 Push Document Modality 33
4.2.3.2.2 Pull Document Modality 34
4.2.3.2.3 Publish Document Modality 34
4.2.3.3 An Evaluation of the Document Exchange Architecture 35

5 Sample Integrated Justice Information System Architecture 38

5.1 Overview 38

Copyright © 2001 – Jim Threatte

5.2 Critical Component 1 – Document Exchange 40
5.2.1 Enabling Data Extraction and Preparing the Document To Be Exchanged 41
5.2.2 Transporting the Document 42
5.3 Critical Component 2 – Data Warehouse 43
5.4 Public Access 44
5.5 Security Considerations 45
5.6 Conclusion 46

FIGURES

Figure 1 - Sample Consolidated System Architecture 6
Figure 2 - Sample Coordinated System Architecture 7
Figure 3 - Sample Hybrid System Architecture – Index Only 9
Figure 4 - Sample Hybrid System Architecture – Data Warehouse Only 10
Figure 5 - Sample Hybrid System Architecture – Index and Data Warehouse 11
Figure 6 - Direct Query Data Warehouse 13
Figure 7 - Virtual Query Data Warehouse 14
Figure 8 - Single-Subject Data Warehouse 15
Figure 9 - Mixed Workloads Data Warehouse 16
Figure 10 - Virtual Global Data Warehouse 17
Figure 11 - Selected Data Warehouse Model 18
Figure 12 - Data Integration Hub Architecture 24
Figure 13 - Sample Data Integration Hub Model 27
Figure 14 - Document Exchange Architecture – Push Mode 34
Figure 15 - Document Exchange Architecture – Pull Mode 34
Figure 16 - Document Exchange Architecture – Publish Mode 35
Figure 17 - Sample Document Exchange Architecture 36
Figure 18 - Sample Architecture 39
Figure 19 - Public Access 45

The author of this report welcomes comments, questions, corrections, or any other feedback.

Jim Threatte
Principal Consultant
IBM Business Consulting Services
12902 Federal Systems Park Drive
Fairfax, VA 22033-4412

Telephone: (703) 322-5135
Fax: (703) 322-2569
E-mail: James.L.Threatte@us.ibm.com

Copyright © 2001 – Jim Threatte

THIS PAGE BLANK

Copyright © 2001 – Jim Threatte

1 Executive Summary

Throughout the nation, there is much enthusiasm regarding the general objectives for the integration of justice systems. Some of the general objectives include smooth and effective information sharing and use, increased public safety, enhanced justice in society, and more efficient government operations. The problems come when actual integration initiatives generate conflict over specific impacts on budgets, organizational relationships, and established procedures. These are not problems of technology or of conflicting visions. These are problems of organizational and political interests and relationships.

A recent study[1] was conducted by the Center for Technology in Government under a grant from the Department of Justice, Office of Justice Programs (OJP). This grant was provided as part of OJP's Integrated Justice Information Systems initiative. The study showed that achieving a high level of integration is feasible and has been achieved in some states and localities. The report states, "These examples of success can serve as lighthouses for integration efforts elsewhere by illustrating problems to be solved, successful strategies, and benefits to be obtained."

This report, *Integrated Justice Information System Architecture Building Blocks Construction*, provides an evaluation of optional approaches and a sample of an Integrated Justice Information System (IJIS) architecture. Because the term "architecture" has a variety of meanings, this report uses the term architecture to refer to "the definition of an information system using models."

The sample IJIS architecture established by this report is constructed using a building block approach. Each of the building block options is presented, and the selection of specific building blocks is established by providing a description of the advantages and disadvantages of the options. Because advantages and disadvantages will differ from jurisdiction to jurisdiction, the sample architecture that is constructed is only an example that was established using the decision criteria envisioned by this report. However, based on the research conducted leading up to this report, many of the decision criteria may be familiar and an appropriate selection. Therefore, the resulting sample IJIS architecture may be a close fit to one's individual needs.

The sample architecture can serve as a roadmap for the construction of an IJIS. One of the strengths of this architecture is that a "big bang" approach is not required in order to arrive at the final destination. In fact, the sample architecture provides for a variety of technology approaches to be used to allow justice entities to join the integrated system by employing small, incremental technology solutions.

---

[1] The Center for Technology in Government, University at Albany/SUNY published its Reconnaissance Study report at www.ctg.albany.edu/projects/doj/dojmn.html.

Copyright © 2001 – Jim Threatte

The term "entity" is used throughout this report as a generic term to identify the various agencies, departments, courts, and stakeholder groups that will participate in the integrated justice information system. This includes, but is not limited to, all aspects of adult and juvenile law enforcement, prosecution, adjudication, incarceration, and supervision activities such as probation and parole.

This report first presents an overview of alternative system architectures that could be used to achieve system integration. The consolidated system approach is presented and described as a centralized concept developed to meet all functional requirements of all participating entities. This approach frequently involves a single database or central computer system. Next, the coordinated system is presented and described as one that relies on the cooperation of multiple, independent systems participating in the integration effort for their mutual good and for the direct benefit that each system participant will receive.

Finally, a hybrid system architecture is introduced and defined by this report as the most feasible approach that could be implemented by most organizations pursuing an IJIS project. Therefore, a hybrid system architecture is selected as the first building block that forms the foundation for the sample IJIS architecture.

The report then presents an overview of alternative data architectures and defines the data warehouse, data integration hub, and document exchange architectures. Each of these data architectures is reviewed, and the advantages and disadvantages of each are described. A hybrid data architecture is introduced and defined as the most feasible approach that could be implemented by most organizations pursuing an IJIS project. Therefore, this hybrid data architecture is selected as the next building block for the sample IJIS architecture.

The hybrid system and data architectures are combined to define a sample IJIS architecture that can provide the roadmap for achieving an IJIS for many jurisdictions. The specific structure for this sample IJIS is defined and named the Integrated JUstice System Architecture™ (IJUSA). The critical components of the IJUSA are then described in order to further the understanding of the advantages of this approach.

2 Background and Introduction

With more than 22 years of information technology experience working primarily in the law enforcement and criminal justice domains, the author of this report has become familiar with a wide variety of law enforcement and justice systems and their implementations. Recently, he had the privilege of leading the redesign of the state criminal history repository systems for Washington and Oregon. While working on these projects, he was also provided the opportunity for close exposure to the design, development, and deployment of the national integrated justice system. The Federal Bureau of Investigation (FBI) Interstate Identification Index (III) is the national model for

Copyright © 2001 – Jim Threatte

the integration of arrest, court conviction, and correction/custody status data. The III model is adopted by state repositories such as Oregon and Washington and used as the foundation for integrating their state-level justice data.

While working on these projects, the author also had the opportunity to be involved with, and exposed to, the design, development, and deployment of the FBI National Instant Check System (NICS). Also known as the Brady Bill system, NICS integrates data from a variety of databases in order to provide a complete picture of a subject's involvement in the justice system that might preclude him or her from purchasing a firearm.

The author attended the 1999 BJA/SEARCH Symposium on Integrated Justice Information Systems[2] and regularly attends SEARCH[3] membership meetings. As a founding member of the Integrated Justice Information Systems – Industry Working Group[4] (IWG), he has been participating in its IJIS initiatives at the national level for several years. As the subcommittee chairman for the IWG standards subcommittee, he keeps in constant touch with technology and standards initiatives in the IJIS area.

This report contains the results of research and analysis over the past few years. It contains comparative information on alternative system and data architectures. In many ways, this report acts as a primer on the topics of system and data architectures. It assembles technology approaches, using a building block approach that arrives at a unique sample architecture. However, it is also possible to assemble a perfectly valid IJIS architecture using different building blocks that are each selected using different decision criteria.

The primary purpose of this report is to add substance to, and facilitate, the national dialogue regarding the technical approaches that will result in the successful integration of justice systems into a fully integrated justice solution.

This report is recommended reading for those who (1) wish to be more informed regarding justice system integration initiatives, (2) desire a different perspective regarding a technical approach for establishing an Integrated Justice Information System architecture, and (3) are convinced that there is a business case for the integration of the justice systems and would like a roadmap to begin their project.

---

2 The Bureau of Justice Assistance (BJA), U.S. Department of Justice and SEARCH, the national consortium for justice information and statistics, sponsored this symposium to provide practical resources to state and local justice agency representatives considering, or in the midst of implementing, integrated justice information systems.

3 SEARCH, the National Consortium for Justice Information and Statistics, is a nonprofit membership organization created by and for the states that is dedicated to improving the criminal justice system through the effective application of information and identification technology. Additional information regarding the SEARCH integration program can be found at www.search.org/integration/default.asp.

4 More information on the IJIS IWG is available via the Internet at www.ijis.org.

Copyright © 2001 – Jim Threatte

3 Business Case for Integration

There is growing attention to, and increased funding available for, justice integration initiatives at the local, state, and federal levels. As attention and resources have increased, the cost of information technology has continued to decline, bringing higher capability within the budgets for new initiatives.

A recent study, *Reconnaissance Study Developing a Business Case for the Integration of Criminal Justice Information*, was conducted by the Center for Technology in Government under a grant from the Department of Justice, Office of Justice Programs (OJP).[5] The grant was provided as part of OJP's Integrated Justice Information Systems initiative.

The results of this study are based primarily on 26 interviews conducted with persons who were knowledgeable about specific state and local initiatives. The study also draws on published materials about the integration initiatives obtained either directly from the participants or by searching print and electronic sources. The study showed that achieving a high level of integration is feasible and has been achieved in some states and localities. As the report states, "These examples of success can serve as lighthouses for integration efforts elsewhere by illustrating problems to be solved, successful strategies, and benefits to be obtained."

Successful achievement of integration objectives was demonstrated to be clearly possible, using a number of different paths. The approaches range from a "full frontal assault" on comprehensive integration objectives to limited strategic objectives as part of a larger strategic plan. In either case, several factors played a role in success. One of the three primary factors indicated was "a building block implementation philosophy that allows systems to be constructed from a number of interrelated 'blocks' or components, guided by an overall vision or strategic plan."[6]

The technical development strategy selected by the state of Colorado was based on analysis of business requirements and an extensive business process mapping exercise of each component of the criminal justice enterprise. This resulted in a strategy of integrating existing legacy systems while maintaining their own independence as much as possible. Information is passed from one agency to another with as little disruption as possible to the environment of each agency. This approach did not assume that future reengineering or migration toward more common systems would not be necessary; instead, it placed priority on achieving a

---

[5] The Center for Technology in Government, University at Albany/SUNY published its *Reconnaissance Study* at www.ctg.albany.edu/projects/doj/dojmn.html. The executive briefing for the full report is located at www.ctg.albany.edu/resources/pdfrpwp/doj_exec_briefing.pdf. The full report can be located at www.ctg.albany.edu/resources/pdfrpwp/doj_guide.pdf.

[6] Anthony M. Cresswell and David Connelly, *Reconnaissance Study, Developing a Business Case for the Integration of Criminal Justice Information*, Center for Technology in Government, University at Albany, SUNY, September 1999.

Copyright © 2001 – Jim Threatte

quick success. Colorado's approach has provided the ability to extract and join data across the entire system for the purposes of decision support at all levels. This has allowed for a complete criminal history that has up to this point been impossible to achieve.[7]

The technical development strategy selected for Harris County, Texas, was based on several elements. The first element was that design and development decisions were based on a comprehensive and well-grounded understanding of information flow, business rules, and user needs. Another element was to incorporate an appropriate mix of centralized standards and controls with decentralized or distributed repositories and systems. The result was a mix of centralized and controlled components with flexible and adapted components for the collaborating agencies. The third element was taking an incremental (building block) approach to system development within a longer-range planning framework to develop components in smaller, more manageable steps as part of a long-range strategy.[8]

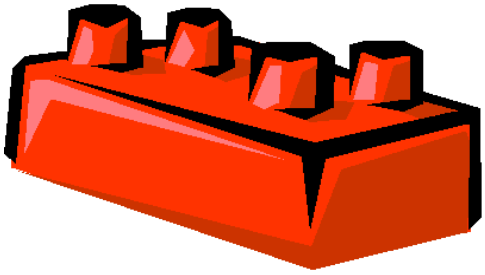
These are only two examples of projects that can provide insight and guidance toward the execution of a successful integrated justice project.

4 Alternative Integration Architectures

This section presents an overview of the alternative integration architectures that were considered when forming the sample architecture.

4.1 Alternative System Architectures

Integration architectures fit into one of three major categories. One category is the consolidated system. Another category is the coordinated system. A third option would be a hybrid version, combining both the consolidated and the coordinated systems. The following sections of this report describe the basic differences between these system architectures and identify the architecture that was selected as a sample for this report.



[7] Ibid.

[8] Ibid.

Copyright © 2001 – Jim Threatte

4.1.1 Consolidated System – Centralized

In a consolidated system, a centralized platform concept is developed in order to meet all functional requirements of all participating entities. This approach frequently involves a single database or central computer system. Alternatively, the software components and data may be distributed across different sites and entities. However, if the software and data components are tightly coupled, the resulting system is a consolidated system. The following figure depicts a sample of a consolidated system:

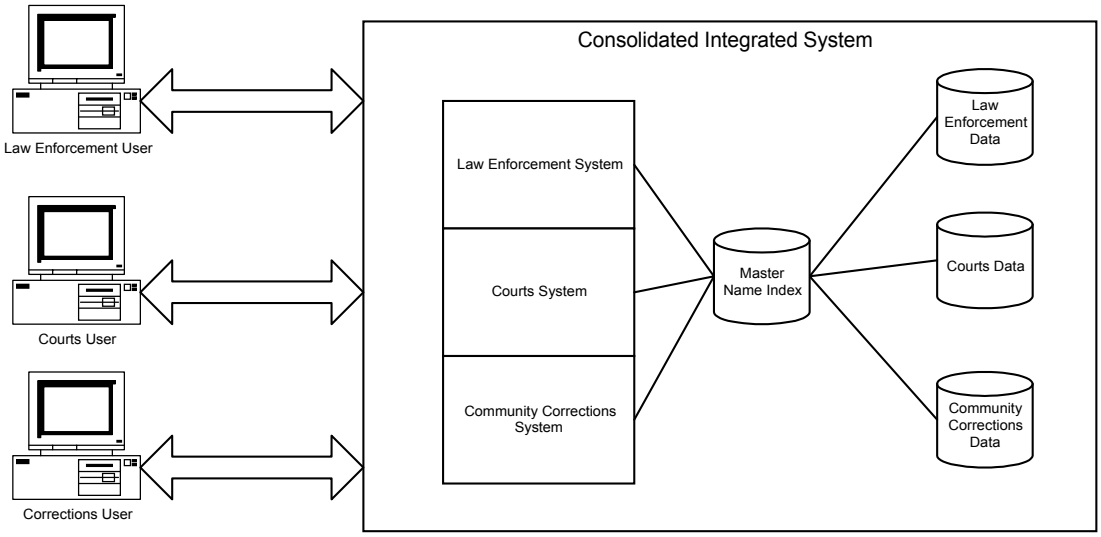


Figure 1 - Sample Consolidated System Architecture

Although a consolidated system might be the most cost-effective solution with the lowest technical risk and lowest cost of support, this solution is not usually a feasible alternative because of political, policy, and environmental issues. The existing justice information enterprise comprises separate branches of government as well as entities that have a necessarily adversarial relationship. In addition, the political and policy infrastructures that are in place make the establishment of consolidated system architecture extremely difficult, if not impossible, to achieve for most jurisdictions.

Some systems may be consolidated within their own functional area. For example, the Police Computer-Assisted Dispatch software may be consolidated with the Police Records Management and Jail Management software. However, these functionally consolidated systems are not considered consolidated from the entire IJIS enterprise perspective.

4.1.2 Coordinated System – Point-to-Point

In a coordinated system, the design and architecture follow agency lines and use different data-processing platforms, applications, and operating systems. Working in a loosely coupled manner, the entities agree on the basic data structures and business rules necessary to

Copyright © 2001 – Jim Threatte

exchange the information to achieve the goals of system integration. The following figure depicts a sample of a coordinated system:

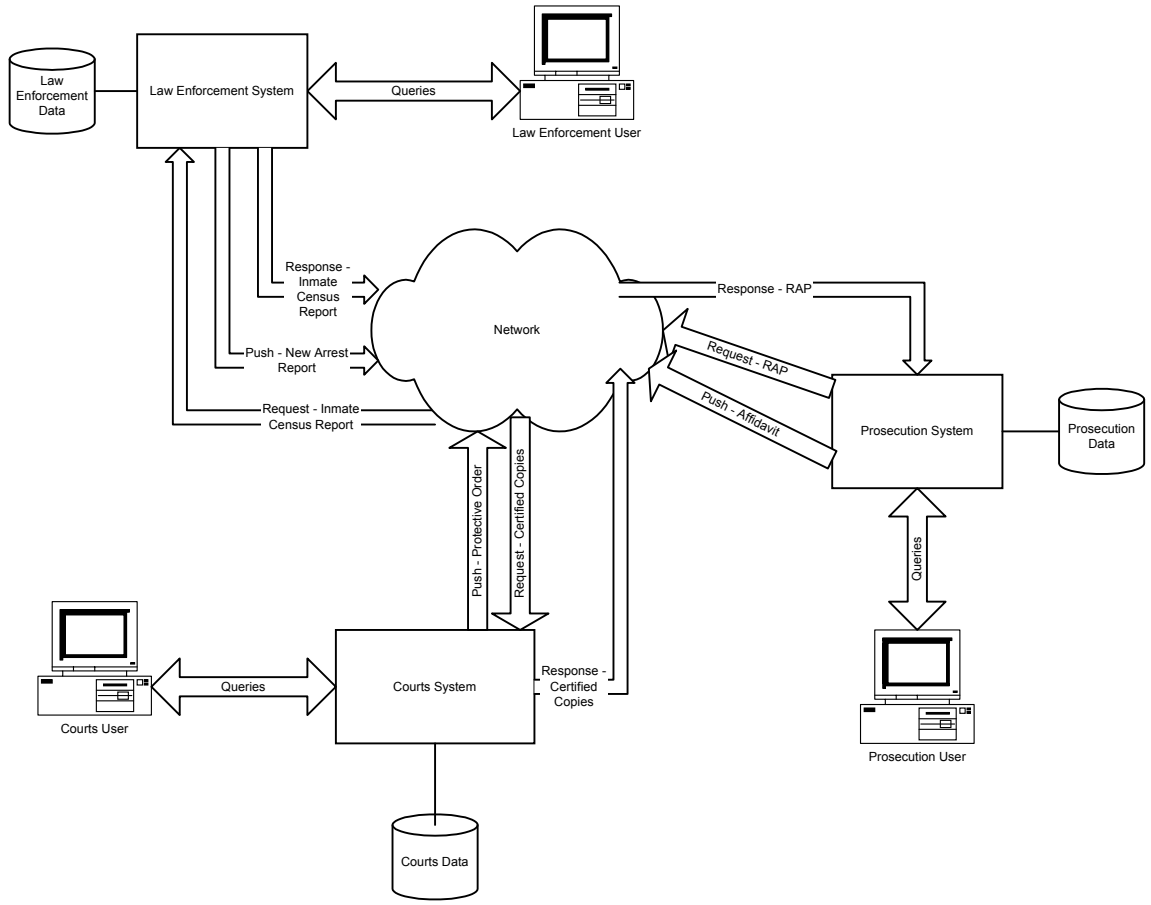


Figure 2 - Sample Coordinated System Architecture

The coordinated system relies on the cooperation of parties to participate in the integration effort for the mutual good of all and for the direct benefit that each system participant will receive. This will typically require that memorandums of understanding or similar kinds of agreements be established among all participating entities in order to facilitate the free exchange of agreed-upon information. The coordinated integrated system also allows for a diverse set of software business applications that meet the needs of each entity or agency.

Coordinated solutions have grown over time as information technology organizations have attempted to integrate information from different units of the same organization, using the technology that was available at that time. This approach suffers from a number of problems that multiply geometrically when inserted into a heterogeneous hardware and software environment. While it is initially appealing, the main problems (scalability and maintainability) end up being very costly, even for small implementations.

Copyright © 2001 – Jim Threatte

This approach requires that each application must write interfaces to every other application with which it needs to share data. This results in intricate, expensive, and redundant interfaces. The combination of overly complex interface schemes and redundant data entry results in errors and inefficiency in the process. In addition, substantial effort is expended in minimizing duplicate data entry, checking multiple systems to obtain information, and resolving inconsistencies in information between systems. The consequences of decisions made with incomplete, inaccurate, or untimely information are of great significance.

4.1.3 Hybrid System – Hub-and-Spoke

A hybrid architecture for integration can also be defined as one that allows the individual entities to maintain their unique data, with central master indexes and common data supported by an agency responsible for the mission of integration. In a hybrid solution, there are two principal components, and at least one of the two components is required. However, in many cases, both of the following components might be advisable:

1. A central system index that controls the flow of information from one system to another
2. A central repository of information from all entities

With a central system index, a central computer is responsible for maintaining a common reference to data contained on other systems. For example, a master name index for all individuals associated with the justice system would be an essential component of the central index. The complete data associated with an individual's involvement with each entity would be maintained on the entity's computer. For example, arrest data would be on the law enforcement computer, court convictions on the court case management system, and incarceration status on the corrections system. All systems would need to adhere to the established rules for supporting the centralized indexes (such as the master name index). They would also need to participate in a mutually agreed-upon approach allowing all authorized entities to retrieve the source data indicated by the master index.

This type of central index system allows each entity to maintain its own operational system(s) and receive critical information from others. [The FBI's current integrated national criminal history repository system, the Interstate Identification Index (III), is an example of this architecture.]

The following figure depicts a sample hybrid system configuration with a central index approach:

Copyright © 2001 – Jim Threatte

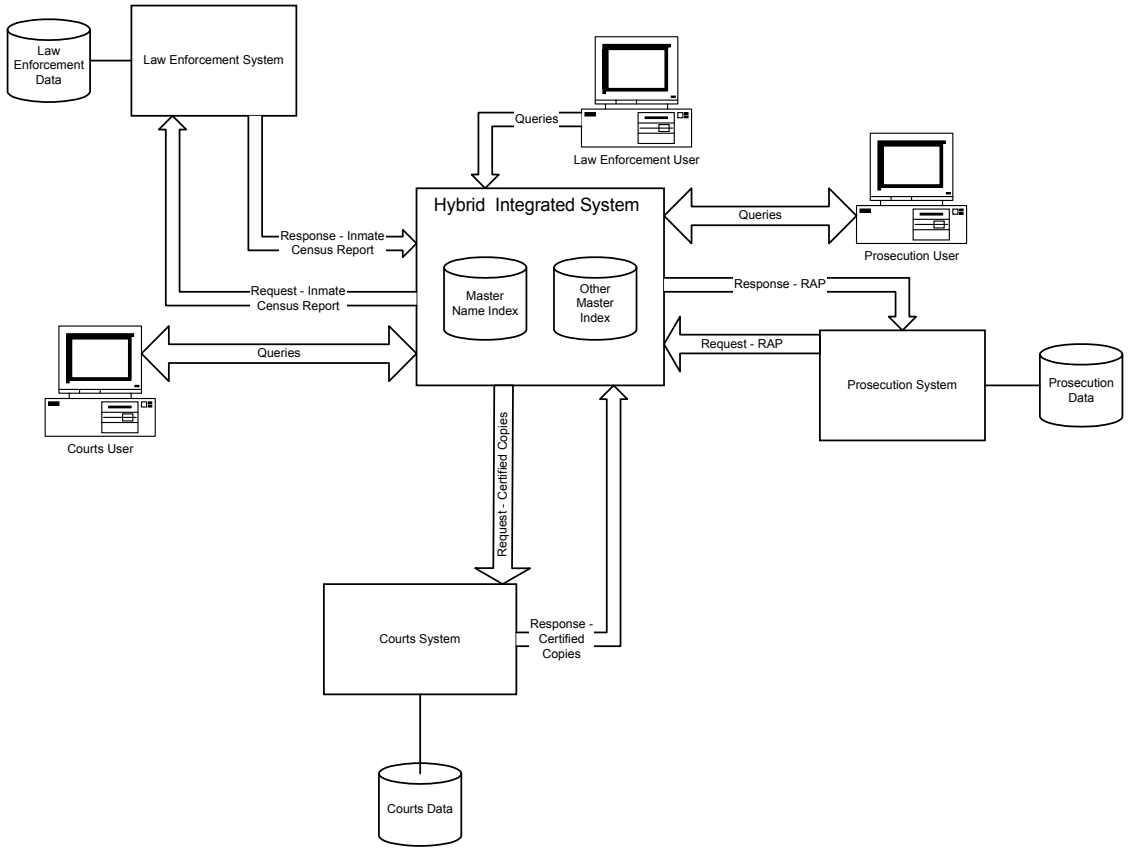


Figure 3 - Sample Hybrid System Architecture – Index Only

An alternative approach for constructing a hybrid system architecture is to use a data warehouse instead of a central system index. With a data warehouse, the complete data associated with an individual's involvement with each entity is be maintained on a centralized computer. Data warehouse systems require that data must first be extracted from each of the participating operational systems. This extracted data must then be evaluated and modified in order to meet a common data model and data element definition required by the data warehouse. This process transforms the data from its original format to the common format required by the centralized data storage mechanism that is typically a relational database system. (Additional information regarding the definition of data warehouses and their options is presented later in this report.)

All participating systems would need to adhere to the established rules for supporting the extraction of all required data to the centralized data warehouse and would need to participate in the approach and rules necessary to support a centralized data warehouse.

With a data warehouse system, each entity maintains its own operational system and participates in the integration of data for query purposes by providing information to a central repository. This hybrid method of integration can allow the individual entities to maintain their data while at the same time sharing information with one another, using

Copyright © 2001 – Jim Threatte

the data warehouse on a central computer. [The FBI's original integrated national criminal history repository system, the Identification Division Automated System (IDAS), is an example of this architecture.]

The following figure depicts a sample hybrid system configuration using a data warehouse approach:

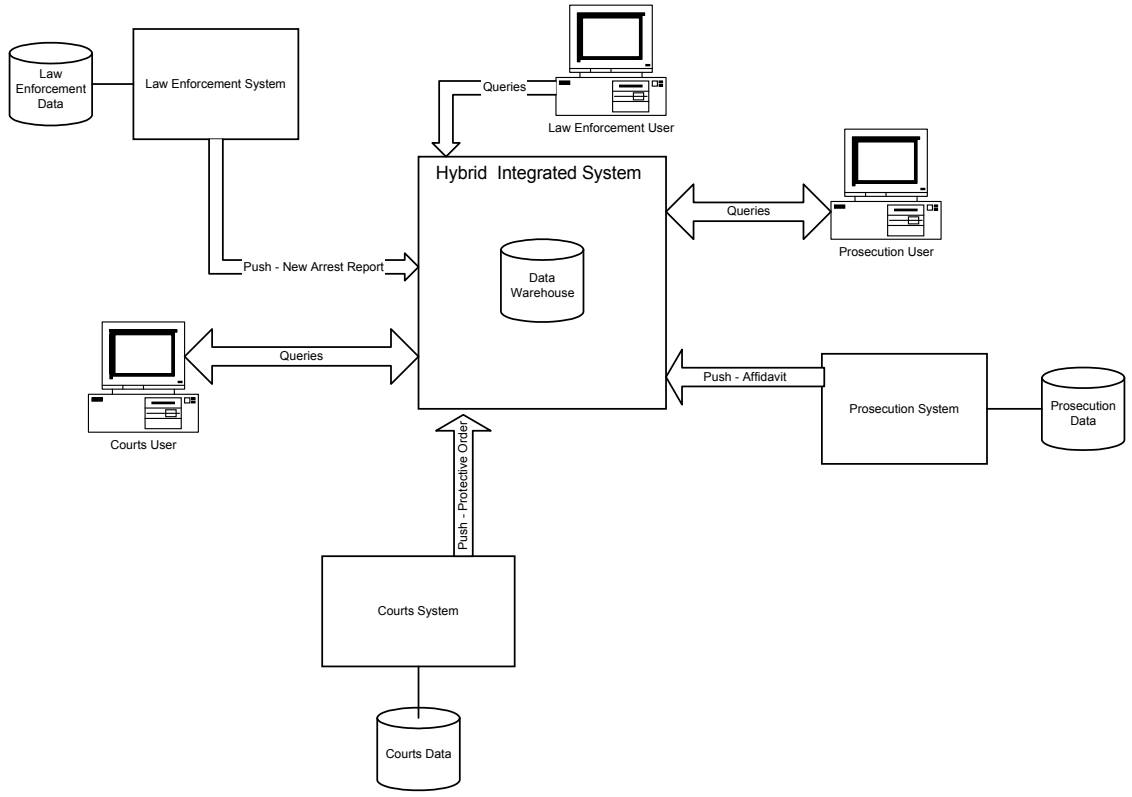


Figure 4 - Sample Hybrid System Architecture – Data Warehouse Only

Finally, the hybrid system architecture provides for a combination of both a central index and a data warehouse on a centralized computer system. This approach allows each entity to maintain its own operational system; however, each entity can participate in the integration of data, using a variety of modalities. This hybrid method of integration can allow the individual entities to maintain their data while at the same time sharing information with one another, as well as with a central computer.

The advantage of this hybrid approach is that it allows a majority of the general inquiry transactions to be serviced by the data warehouse on the central server, which reduces the impact of queries on the operational systems. For data that is difficult to place into a data warehouse or where data security is a concern, the central index approach is available to provide access to data directly from the operational systems.

Copyright © 2001 – Jim Threatte

The FBI's current integrated national criminal history repository system currently operates by providing both the central index architecture of the III and the centralized data warehouse approach of its predecessor, the IDAS. Both logical architectural components are combined and are generally referred to simply as the III.

The following figure depicts a sample hybrid system configuration using both a central index and a data warehouse approach:

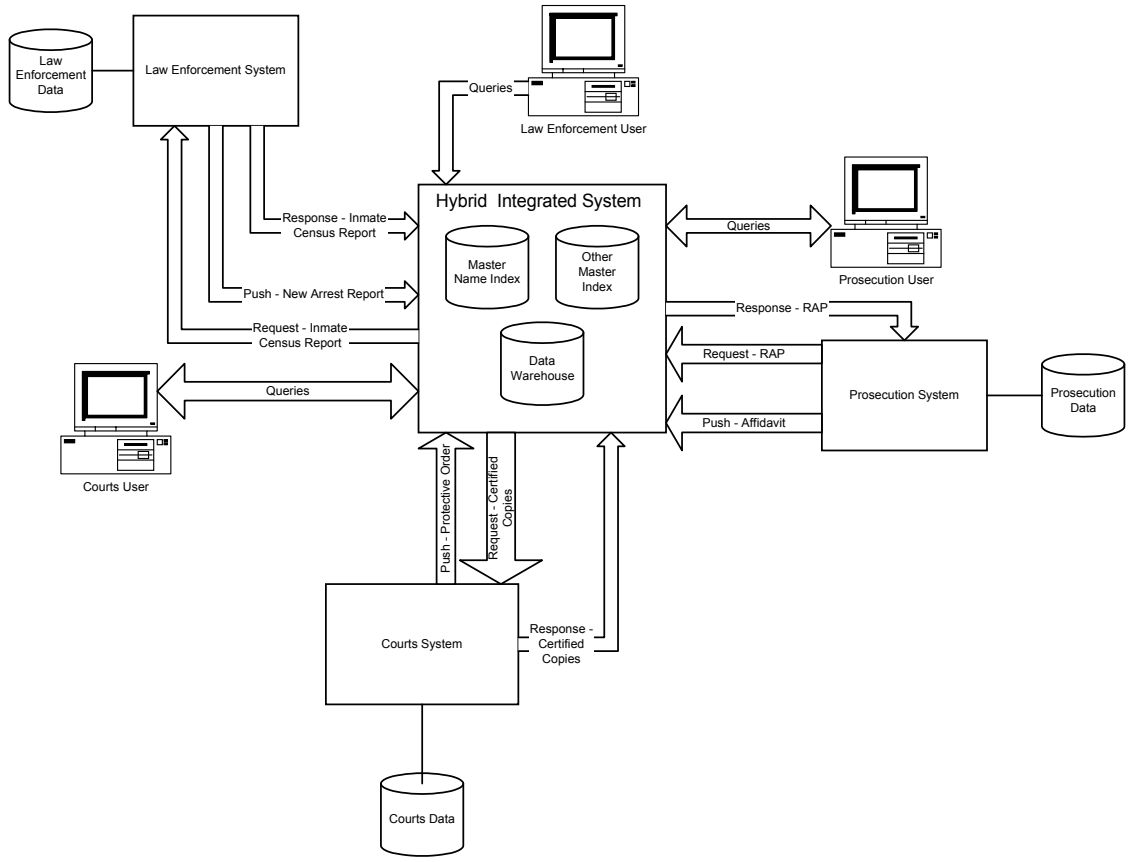


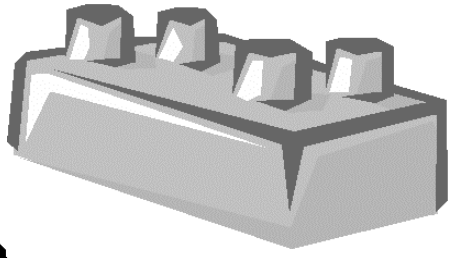
Figure 5 - Sample Hybrid System Architecture – Index and Data Warehouse

The principal advantage of the hybrid system architecture over the coordinated system architecture is that the hybrid architecture integrates disparate systems in a way that reduces the need for unique interface programs to be imbedded into each application. Each application that needs to share data does not need to write a separate interface to every other application. Instead, each application uses the services of a common, shared data exchange facility. In this model, each application "publishes" its shared data when relevant events occur in its system. Other applications can then "subscribe" to the event's data.

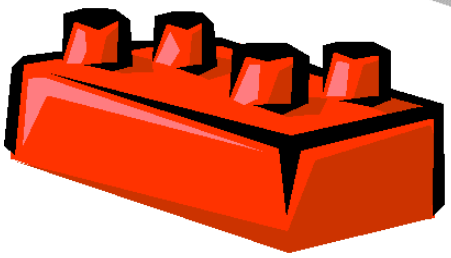
Copyright © 2001 – Jim Threatte

Because the hybrid architecture addresses shortcomings of both the consolidated and the coordinated systems, this approach was selected as the first building block of the sample IJIS architecture.

4.2 Alternative Data Architectures



In addition to considering different system architectures, this report reviews three different data integration architectures. The data integration architectures fit into one of three major categories: data warehouse, data integration hub, and document exchange. The following section of this report describes the basic differences between these data integration architectures.



4.2.1 Data Warehouse Architecture

The term "data warehouse" is intuitively understood to mean a central place (warehouse) that stores data. A more formal definition is that a data warehouse is a large electronic storage area that is structured so that the data stored in different formats is converted to a consistent format for use by a variety of users.

4.2.1.1 Data Warehouse Architecture Overview

Data warehousing provides many benefits to the process of disseminating business intelligence into the enterprise. First, by acting as a single repository for information from many applications, a data warehouse removes the burden of disparate data access from the user application. Data is presented to the user applications in an easy-to-access, typically relational database format. Second, the act of populating the data warehouse provides the opportunity to cleanse the data. In other words, before the data is put into the warehouse, it can be checked and altered to achieve the intended use. This data-cleansing function is not an inherent feature of a data warehouse, but a facility that is provided by tools and procedures used to populate data warehouses.

Because data residing with multiple sources must often be combined for effective decision making, many organizations have implemented data warehouses. This creates essentially a hub-and-spoke model, whereby each enterprise application sends information to the warehouse where it can then be accessed throughout the enterprise.

The term "data warehouse" can mean different things to different people. Six basic models can be used, and are referred to, as data warehouses. Each of the data warehouse architectures requires complex data-modeling and system integration projects to be deployed. They are neither easy nor inexpensive to build. They require system engineers

Copyright © 2001 – Jim Threatte

to work with business users to define current and future information requirements to support decision making. They also require system engineers to piece together tools and products from multiple vendors and establish procedures and operations to manage the flow and transformation of data from legacy systems to warehouses.

4.2.1.2 Data Warehouse Architectures

This report analyzed the six basic data warehouse models. Each of these six models is defined, and the status and disposition of each, with regard to the selected sample architecture, are provided. Finally, a data warehouse architecture is identified for incorporation in the sample IJIS architecture.

4.2.1.2.1 Direct Query

The direct query model is a tools-driven architecture, rather than a data warehouse architecture; however, the end result is the same: end users gain direct access to enterprise data. The problem here is that users must know Structured Query Language[9] (SQL) and arcane database structures, and their queries can impact the performance of operational systems. The direct query model is not considered a viable or reasonable approach for an IJIS data architecture because (1) it requires the user to know how to use the SQL language, (2) it does not provide an integrated response back to the user, and (3) the databases in many justice organizations do not support SQL access. The following figure is a diagram of a direct query data warehouse:

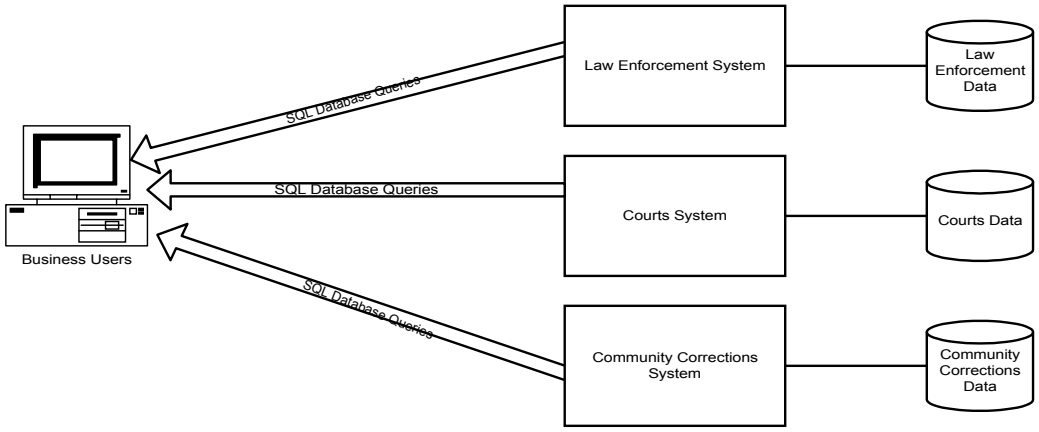


Figure 6 - Direct Query Data Warehouse

[9] Structured Query Language (SQL) is a procedural software language used by database software to perform business logic actions.

Copyright © 2001 – Jim Threatte

4.2.1.2.2 Virtual Query

A virtual query model is supported by database gateways[10] and second-generation query tools. These tools hide SQL and database structures from end users behind a semantic mapping layer that maps generic requests for business objects into proper SQL that is transparent to users. Thus users don't have to know SQL or arcane table names.

Typically, requests are directed to one data source at a time, and the results are presented separately. Most tools of this type can't combine information that is stored on different computer systems in different data formats. A significant characteristic of virtual query environments is that the back-end data source is an online system that supports the business's core operations. This causes contention between query and update activity, which may erode performance of critical online systems.

The virtual query model is not considered to be a viable or reasonable approach for an IJIS because (1) it does not provide an integrated response to the user, (2) concerns exist regarding the potential performance impact on operational systems, and (3) some of the databases in use by typical justice agencies do not support a direct SQL access. The following figure is a diagram of a virtual query data warehouse:

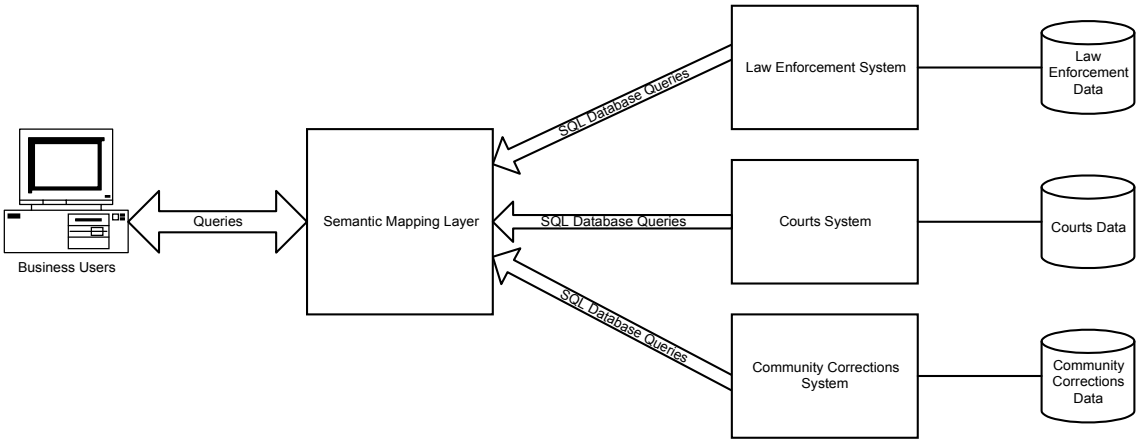


Figure 7 - Virtual Query Data Warehouse

[10] A "gateway" is a generic term used to describe a system that employs business logic, procedural code, and other more complex technology to transform and translate between two or more computer systems.

Copyright © 2001 – Jim Threatte

4.2.1.2.3 Single-Subject Warehouse

The single-subject model is considered the classical data warehouse architecture. Using this approach, users directly access subject-oriented data that has been extracted and transformed from one or more operational systems and loaded into a separate computer that contains a data repository or data warehouse.

Many organizations start with a single-subject warehouse and then either expand the number of subjects in the warehouse or create new single-subject warehouses. The single-subject warehouses are often called "data marts," depending on the volume of data and number of users supported. A number of data warehouse vendors adhere to this model, especially the major relational database vendors such as Oracle, IBM, Microsoft, Sybase, and Informix.

The single-subject warehouse has been selected as the data warehouse option for the IJIS architecture sample of this report. The following figure is a diagram of a single-subject warehouse:

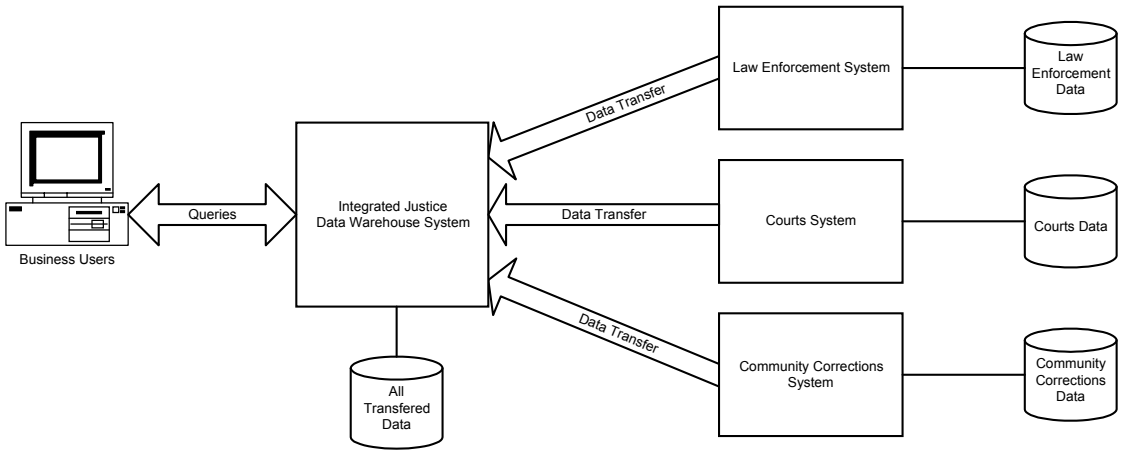


Figure 8 - Single-Subject Data Warehouse

4.2.1.2.4 Mixed Workloads (Database Propagation)

The mixed workload model provides support for both query and update capabilities. The architectural diagram depicts how mixed workload data warehouses evolve by spawning new operational systems, as well as smaller regional or personal data warehouses. This approach allows a warehouse to be reloaded, thus causing a change in data values that, in turn, trigger an update in a remote operational system. An example could be an inventory warehouse triggering a new, just-in-time purchasing system.

Copyright © 2001 – Jim Threatte

The mixed workload model is not a viable or reasonable approach because it is very complex and expensive to procure and maintain. In addition, the data warehouse features offered by this complex architecture are not required to address typical functional requirements of the integrated justice community. The following figure is a diagram of a mixed workloads data warehouse:

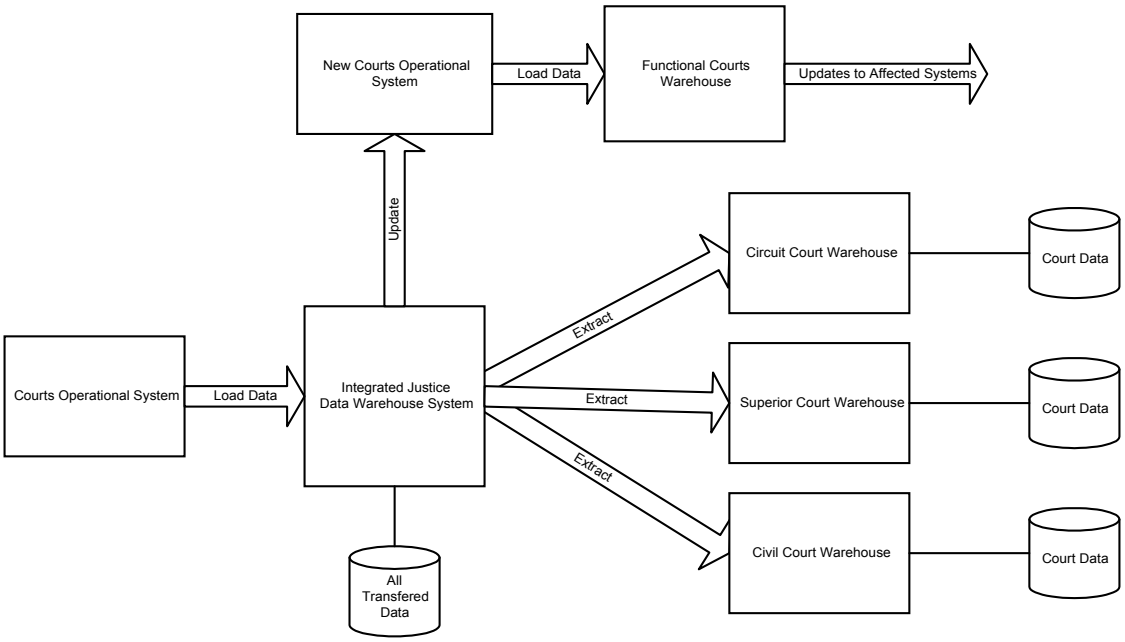


Figure 9 - Mixed Workloads Data Warehouse

4.2.1.2.5 Virtual Global Warehouse

The virtual global architecture integrates all subject-oriented warehouses that have cropped up willy-nilly throughout an enterprise. The virtual global warehouse enables organizations to create a global view of all data. The integration mechanism is an SQL parsing-and-mapping engine.

This is not too different from the virtual query architecture, except that the back-end data sources are all data warehouses, not operational systems. The one major requirement for virtual global warehouses is that all warehouses share identical data definitions. This is a difficult feat, involving strong doses of cross-organizational cooperation. Virtual global warehouses can also be built from the ground up, knitting together disparate groups that realize that they can each benefit more by sharing definitions and data than by trying to reinvent a global warehouse on their own.

The virtual global model is not considered an appropriate architecture for most integrated justice initiatives at this juncture. Research indicates that not many data warehouses exist

Copyright © 2001 – Jim Threatte

in the justice environment at this time. This architecture is appropriate for integrating multiple data warehouses. For many, an IJIS project will, for the first time, create a data warehouse that may, in the future, become one of the data warehouses integrated by a virtual global model. It may be advisable to establish a vision that recognizes the ability to create a virtual global warehouse as a goal as other data warehouse projects are deployed or improved. The following figure is a diagram of a virtual global data warehouse:

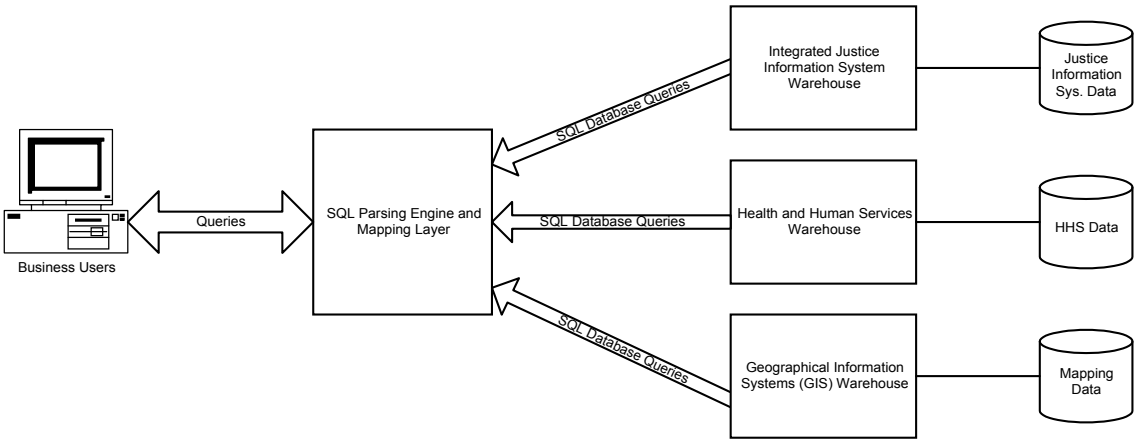


Figure 10 - Virtual Global Data Warehouse

4.2.1.2.6 Event-Driven Warehouse

Organizations deploy event-driven architectures in order to integrate disparate applications into an integrated system that can mimic and adapt to the business. In event-driven systems, applications typically "publish" data to an "information bus" and then "subscribe" to the data they need. Event-driven systems typically use a universal application interface implemented through some form of "middleware"[11] in order to transform all shared data into a common format with common terms and definitions.

This architecture has become very specialized, and a variety of software companies have developed a variety of middleware products designed specifically to accomplish the goals of the event-driven warehouse. In fact, this report has classified this model in a category by itself called the "data integration hub."

This report has evaluated the data integration hub architecture as a unique and independent data architecture.

[11] "Middleware" is a generic term used to describe a variety of technological approaches for connecting systems for the purposes of data transfer and a variety of other system interoperability requirements.

Copyright © 2001 – Jim Threatte

4.2.1.3 Data Warehouse Architecture Evaluation

The single-subject warehouse is the classical data warehouse approach that was selected as the building block for this report. Research indicates that this data warehouse option is the most appropriate and viable data warehouse architecture for use in an IJIS solution.

The following figure depicts several critical features of the data warehouse architecture. The first critical feature is that end users are retrieving all integrated criminal justice information system data, using software designed specifically to extract and report data stored on the data warehouse. This aspect reveals some significant architectural considerations regarding client access:

- New application software must be acquired and customized to meet the user's needs to access data on the data warehouse. This includes the screens for establishing ad hoc queries, as well as the screens necessary to run predefined reports.
- New application software must be developed to execute the ad hoc queries and predefined reports.

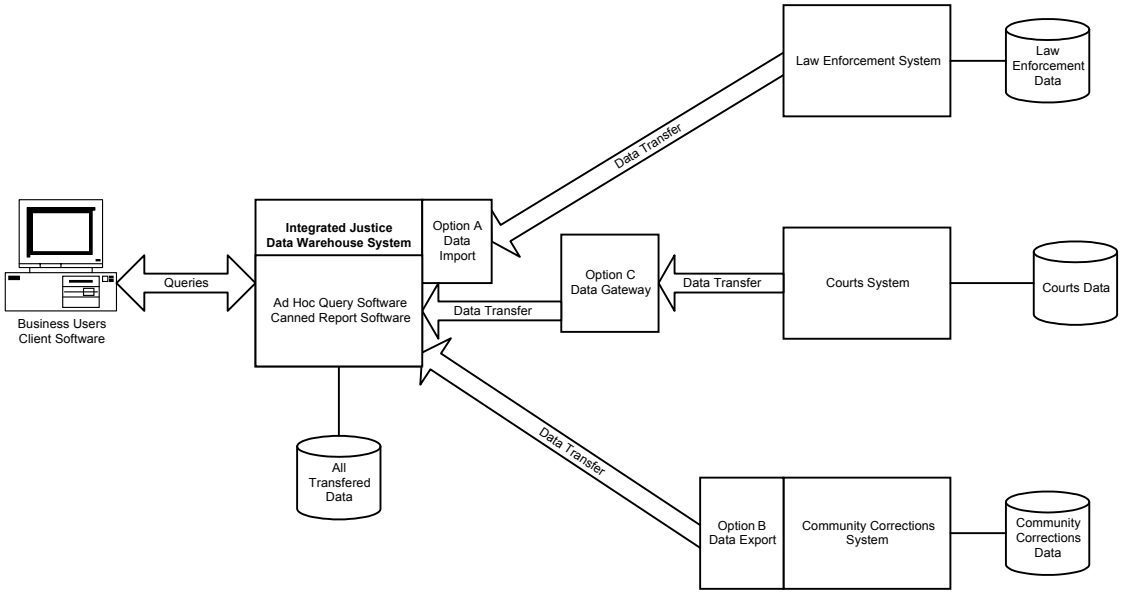


Figure 11 - Selected Data Warehouse Model

4.2.1.3.1 Client Software Alternatives

The new client software that allows users to access the data could utilize Web browser and database search engine technology that can provide an easier-to-use, lower-cost client approach. However, this "thin client" approach sometimes sacrifices the flexibility to define queries and reports. Thin client software typically uses a Web browser for the client workstation interface where the business logic code and procedures are executed on

Copyright © 2001 – Jim Threatte

the server. This allows for centralized maintenance of business logic code that can be frequently subject to change.

Alternatively, the client software could be a commercial off-the-shelf (COTS) query-and-report generation package. This "thick client" approach would provide the power and flexibility to define a variety of reports and queries. Thick client software incorporates business logic software operating on the client's workstation computer. This architecture requires any software changes to be propagated to each individual user workstation located throughout the enterprise. The justice entity responsible for the IJIS would need support resource(s) with expertise in data processing to provide for the more powerful, flexible thick client package. This approach is also usually more expensive because client licenses would be required for all users.

The second salient feature of the database warehouse architecture is that the operational systems' "live" databases are not accessed. Copies of operational systems' data are maintained on the separate data warehouse computer. This aspect of the architecture reveals some significant considerations regarding data access and transfer:

- Data stored in the data warehouse is not current. The data will only be as current as the last time that it was updated from each operational system.
- Data stored in the data warehouse may not be complete. Not all data from the operational systems can be adequately mapped to a common database model. This results in some data elements being orphaned by the data warehouse or stored in such an obscure way that the relevance of the data is not readily apparent.

4.2.1.3.2 Alternative Approaches for Transferring Data

As Figure 11 illustrates, there are three approaches (or options) for transferring the data from the operational systems to the data warehouse:

A. Data import
B. Data export
C. Data gateway

For any of the three approaches for transferring data, remember that the data transfer is defined by two factors. The first factor is the data transfer interval, which is the measure of time that has elapsed since the data was last provided to the data warehouse. The second factor is the volume of the extracted data. One example of extracted data volume would be a full database snapshot. A full database snapshot will transfer all data from the operational system to the data warehouse. An alternative to a full database snapshot is to transfer a subset of the data. The subset can be a "logical subset," which provides a predefined extraction of selected data elements that are made available to the data warehouse for its use. Another data subset is a "transaction subset," which would contain the records that have been changed since the last full data transfer.

Copyright © 2001 – Jim Threatte

A full database snapshot takes a longer time to perform than a subset data transfer. Therefore, the relationship between volume and interval will be important factors when establishing the procedures for transferring data to the data warehouse.

Data Import

Option A depicted in Figure 11 represents the data import approach. With this approach, the data warehouse contains software that retrieves the data from the operational system and stores it in the data warehouse. This data transfer can take place periodically (e.g., weekly, daily, or hourly), based on the rules established. Performance impact to the operational system will occur when the data is transferred. In addition, this data transfer can be either an entire database import or an import of only the data that has changed since the last data import. Importing the entire database will significantly impact the operational system. Importing only the changed data will have less of an impact on the operational system. The options available for the data import depend on the type of database that resides on the operational system, as well as the likely impact of database import activity.

The strength of this approach is that many current-generation tools come with—or work with—a variety of data warehouse software. These tools are capable of attaching to a variety of legacy databases or data storage approaches. They are also capable of extracting the data and transforming it into the data warehouse database model.

Data Export

Option B depicted in Figure 11 represents the data export approach. With this approach, the operational system contains software that extracts the data from the operational system and stores it on the data warehouse. Like the data import approach, this data transfer can take place periodically (e.g., weekly, daily, or hourly), based on the rules established. In addition, this transfer can take place immediately whenever data is modified. This is especially true when the operational system uses a relational database such as Microsoft SQL Server. With a relational database, a “trigger” can be established so that whenever a single data value is modified, a “trigger program” is executed that can transfer the required data to the data warehouse.

Performance impact to the operational system will occur when the data is transferred. Exporting the entire database will significantly impact the operational system. Exporting only the changed data will have less of an impact, and responding to triggers will have the least amount of impact on the operational system. The options available for the data export depend on the type of database that resides on the operational system, as well as the likely impact of database export activity.

The strength of this approach is that sometimes it is more cost-effective to utilize current support staff to develop the software that extracts the data required by the data warehouse. This can be required when proprietary file storage approaches have been used that do not lend themselves to a data import solution. Also, this approach is effective

Copyright © 2001 – Jim Threatte

when the data files do not record the "changed" data and the impact of an entire database transfer to the data warehouse is not desired. Therefore, custom software designed to determine the changed data must be developed in order to provide the data warehouse with the needed data.

Data Gateway

Option C depicted in Figure 11 represents the data gateway approach. With this approach, a new computer is established that contains software that extracts the data from the operational system and then transforms and stores that data on the data warehouse. As with the other options, this data transfer can take place periodically (e.g., weekly, daily, or hourly), based on the rules established. In addition, this transfer could take place immediately whenever data is modified if the operational system contains a trigger (as described above).

Performance impact to the operational system will occur when the data is transferred. Exporting the entire database will significantly impact the operational system. Exporting only the changed data will have less of an impact, and responding to triggers will have the least amount of impact on the operational system.

The strength of this approach is that a separate computer is used to extract data and transform it into the data warehouse data model. This can reduce the time necessary to update data on the data warehouse, as well as leave the full computing power of the data warehouse always available for user inquiry needs.

Sometimes it is simply more cost-effective to utilize third-party database gateway software running on a different hardware platform to accomplish data extraction and cleansing in a cost-effective manner. For instance, some very inexpensive data gateway programs run on a personal computer. If the data warehouse or operational system is operating on a high-availability UNIX computer that only offers expensive software alternatives for database import, export, and cleansing, a PC-based gateway solution can often offer a very attractive alternative solution.

4.2.1.3.3 Advantages and Disadvantages

These are the major advantages of the data warehouse architecture:

- A well-defined, integrated data model with a full copy of the cleansed data is available to all users for query-and-reporting purposes.
- Data access response time and system availability is predictable and provides a high level of service to the users.
- A variety of COTS query-and-reporting tools are available for use by clients to create powerful queries and reports.

Copyright © 2001 – Jim Threatte

These are the major disadvantages of the data warehouse architecture:

- The data represents a snapshot in time and is only as current as the last update by the last system.
- The architecture provides no mechanism for data to be transferred from one system to another in order to reduce or eliminate duplicate data entry.
- This architecture can lead to thick client query-and-reporting applications that can place a significant burden on the data network and the data warehouse server.

4.2.2 Data Integration Hub Architecture

Organizations deploy event-driven architectures in order to integrate disparate applications into an integrated system that can mimic and adapt to the business. The data integration hub architecture is a unique and specialized architecture designed to address these requirements.

4.2.2.1 Data Integration Hub Overview

In event-driven systems, applications typically "publish" data to an "information bus." They also simultaneously "subscribe" to the data they need. Event-driven systems typically use a universal application interface implemented through some form of middleware in order to transform all shared data into a common format with common terms and definitions.

This architecture has become very specialized, and a variety of software companies have developed a variety of middleware products designed specifically to accomplish the goals of the event-driven warehouse. This model exists in a classification by itself called the "data integration hub."

This section evaluates the data integration hub architecture as a unique and independent architecture. Another name commonly given to this architecture is "virtual data warehouse." (These terms are used interchangeably throughout this report.)

A traditional data warehouse provides a central repository for information, requires the development of metadata[12] definitions, and provides data-cleansing facilities. The data integration hub uses middleware in order to provide access to the corporate data stored in heterogeneous data sources. This middleware is used to build direct connections among disparate applications. As with the data warehouse, it relies on the creation of an independent metadata definition of the entire justice system data. It then maps the justice system data schema to each of the operational systems' data. The essential differentiation between the data warehouse and the data integration hub is that the data integration hub accesses raw data from the operational system as it is needed. Because the raw data is accessed directly from the operational system, the data-cleansing activity that is

[12] "Metadata" is data about data or a description of the data that is being referenced.

Copyright © 2001 – Jim Threatte

performed in the data warehouse architecture is not performed in the data integration hub architecture.

Recently, there has been significant movement toward data integration hubs, which has implications for both information dissemination and improved decision making. Data integration hubs allow users to distill the most important pieces of data from disparate legacy applications without the time, expense, and risk to data required by traditional data warehousing. In addition, as more companies are using Web architectures as the backbone for their enterprise networks, they are moving back toward developing their own information-presentation applications. Many organizations are building, or at least considering, enterprise portals. An enterprise portal is a single information gateway, typically browser-based, that can be used to navigate and examine both internal and external data through Web technology.

Most of the major data integration hub vendors have enabled their products to be deployed and used over a Web-based architecture. Indeed, they provide a quick solution for deploying Web-based applications that are easy to implement. The Web-based development environments and growing strength of eXtensible Markup Language[13] (XML) have contributed to their ease of implementation. XML is the standard Web-based, data-manipulation language that can make the development and deployment of custom-built applications feasible for many organizations.

The great strides made in the area of enterprise middleware now provide an interesting alternative to traditional data warehousing. Middleware can act as data hubs, allowing access to the corporate data stored in heterogeneous data sources. Whereas a traditional data warehouse provides a central repository for information, a virtual data warehouse uses middleware to build direct connections among disparate applications. This virtual approach can require less time and expense to develop and entails less risk of data being lost.

4.2.2.2 Data Integration Hub Architecture

Like data warehousing, this middleware approach to direct data access relies on the creation of an independent metadata definition of the enterprise data and therefore provides the same ease-of-use advantages. Layering data-access middleware over the enterprise data allows a user to create a virtual data warehouse, providing access to information without the complexity of building a traditional data warehouse system.

However, data access middleware generally will give a user access to data only in its raw form. It doesn't provide the data cleansing that is a major benefit of building an actual data warehouse. This can be a significant disadvantage to the virtual data warehouse, but the importance of it depends greatly on the requirements of the system.

---

[13] "eXtensible Markup Language (XML)" is a national standard data-tagging language that allows the data to be both identified and contained within the same document.

Copyright © 2001 – Jim Threatte

The following figure depicts the data integration hub architecture. This information was developed based largely on a review of the Enterworks Content Integrator™ product; however, based on a review of similar data integration hub products, the author believes that the basic architecture exists for all products with these capabilities. (An explanation of the major components follows the diagram.)

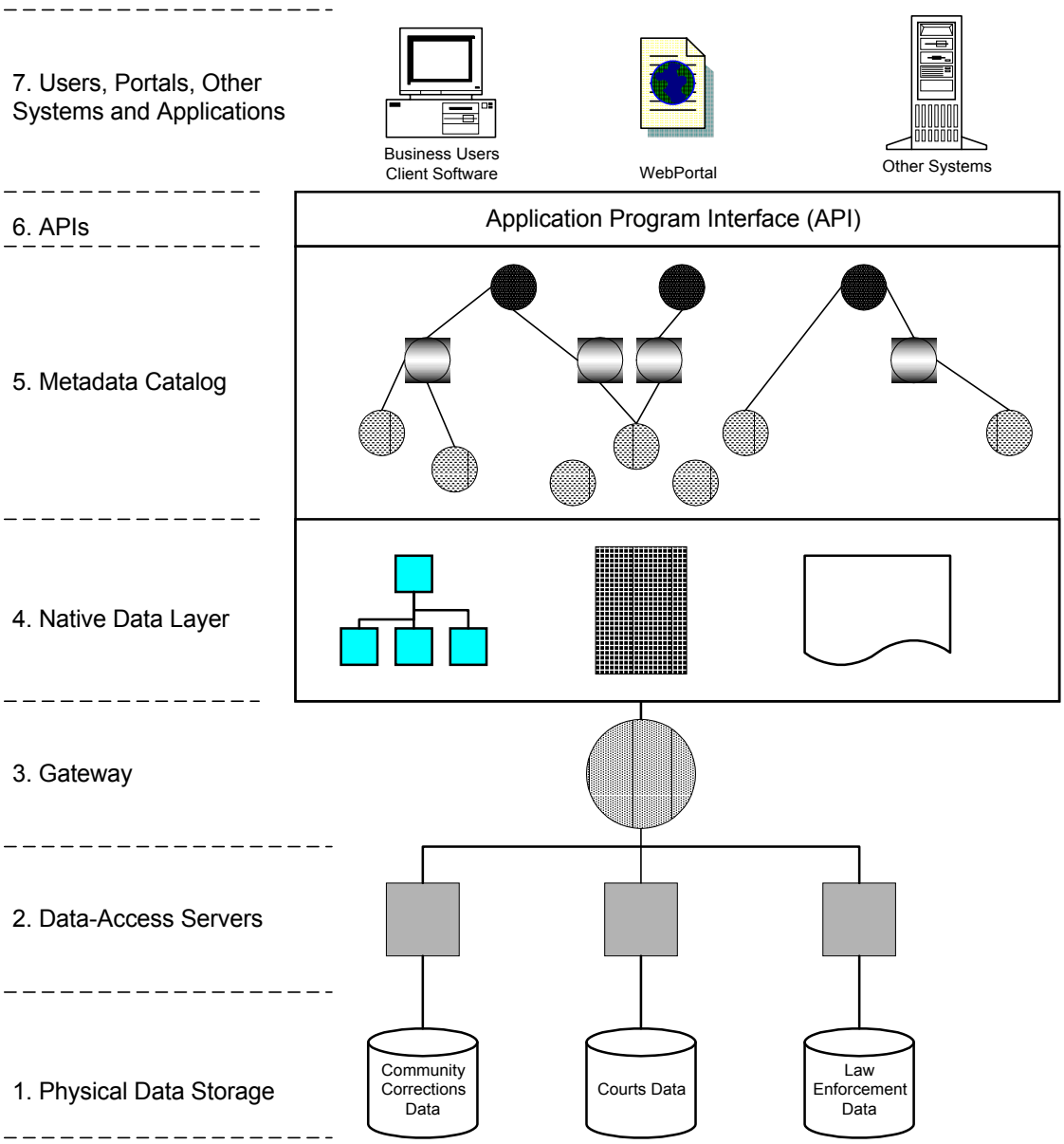


Figure 12 - Data Integration Hub Architecture

Copyright © 2001 – Jim Threatte

4.2.2.2.1 1. Physical Data Storage

The Physical Data Storage layer represents existing databases from which one wishes to obtain information. These databases are updated and queried by their own applications. They may be hosted on any number of platforms (e.g., mainframes, UNIX servers, and Windows NT servers). Most modern databases are controlled by DBMS (DataBase Management System) software such as the ones offered by Oracle, IBM, Microsoft, Sybase, and Informix; however, some are merely structured text files into which applications place data for future retrieval. Some DBMSs are considered proprietary because they were created long ago and are no longer supported (with bug fixes or upgrades) or because they were created to support such narrow requirements that they do not communicate through a standard language or interface.

4.2.2.2.2 2. Data-Access Servers

Each Physical Data Storage speaks its own unique language; however, the Gateway (see item 3 below) speaks only a single common language [e.g., SQL (Structured Query Language)—the standard language for relational DBMSs]. The Data-Access Servers convert the common language (SQL) into the equivalent language of the underlying Physical Data Storage. Data-Access Servers convert commands made in one language into the equivalent commands in the language that the Physical Data Storage understands. They also work in the reverse: once the data is retrieved from the Physical Data Storage, its format must be converted back into the language that the Gateway understands.

4.2.2.2.3 3. Gateway

The Gateway is the plumbing system that connects the Physical Data Storage to the Metadata Catalog (see item 5 below). Gateways provide access to heterogeneous Physical Data Storage devices. The Gateway will accept a single common language—typically SQL commands generated by any database tool. It will then

- Determine the optimal plan of execution,
- Generate the required number of lower-level, database-accessing statements (one for each participating Physical Data Storage),
- Establish a physical connection to each underlying Physical Data Storage device (through the Data-Access Servers),
- Pass any required user ID and password to each system, and
- Pass the appropriate data-access statements to each participating Physical Data Storage for execution.

When the results are returned, the Gateway passes the results back for subsequent transformation. Gateways are also capable of performing heterogeneous database joins if the Physical Data Storage devices are relational databases. Heterogeneous database joins permit data from disparate databases to be logically merged when all of the databases contain a field in common.

Copyright © 2001 – Jim Threatte

4.2.2.2.4 4. Native Data Layer

The Native Data Layer is actually part of the Metadata Catalog (see item 5). The data in the underlying Physical Data Storage is mapped into the Metadata Catalog at the Native Data Layer. For each data element that the Metadata Catalog deems important, the data integration hub software creates a virtual object that points directly to the Physical Data Storage through the Gateway.

Access to any of the content stored within any of the Physical Data Storage can be gained in a standard and consistent manner through the Metadata Catalog. The Native Data Layer maps the logical database viewed by the user and user applications to the individual physical data elements that actually exist in each of the target systems.

4.2.2.2.5 5. Metadata Catalog

"Metadata" is data about data. The Metadata Catalog allows data modelers to create the logical data models they want by associating the Native Data Layer with the logical data model. The result of the Metadata Catalog is a tailored view of an organization's data without regard to the actual physical storage of the data. The actual physical data is not stored in the Metadata Catalog—it is stored only once within the Physical Data Storage. The data integration hub uses the metadata contained within the Metadata Catalog to generate the queries necessary to populate the logical views from the actual Physical Data Storage.

4.2.2.2.6 6. APIs

The APIs (Application Program Interfaces) allow software application developers to talk to the Metadata Catalog as though it were a single data source. Different tools speak different languages. Many application development tools speak the Open DataBase Connectivity (ODBC) standard. This interface permits client software applications to access all of an organization's data sources through a single database connection.

4.2.2.2.7 7. Users, Portals, and Other Systems and Applications

One of the strongest features of a high quality data integration hub will be its openness. Application developers can use any ODBC-compliant tool with the data integration hub in order to create custom software applications, a generalized Web search-and-reporting portal, or a direct connection to other systems and applications.

Copyright © 2001 – Jim Threatte

4.2.2.3 Data Integration Hub Architecture Evaluation

The following figure depicts several critical features of the data integration hub architecture. The most critical feature that differentiates this architecture from the data warehouse architecture is that end users are retrieving all integrated criminal justice information system data from the live systems as the data is requested. This fact reveals some significant architectural considerations regarding the client software:

- New application software must be acquired and customized to meet the user needs to access data from the data integration hub. This includes the screens for establishing ad hoc queries, as well as the screens necessary to run predefined reports.
- The metadata dictionary must define the relationship between the data that a user requires and the specific place where that data can be accessed on the target systems.

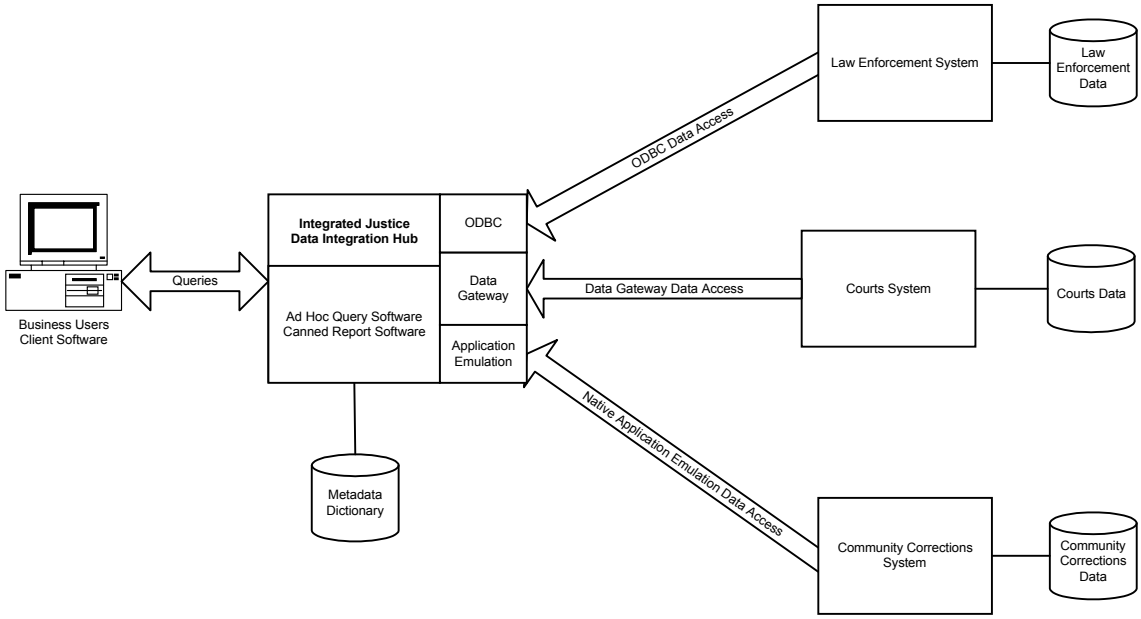


Figure 13 - Sample Data Integration Hub Model

4.2.2.3.1 Client Software Alternatives

The client software that is used by most data integration hub solution providers employs Web browser technology to access the database search-engine technology that resides on the data integration hub server computer. This client software approach can provide an easier-to-use, lower-cost client approach; however, this “thin client” approach sometimes sacrifices the flexibility to define queries and reports.

In addition to the Web-browser-searching software, purpose-built software must be developed for any complex queries or reports. This software can operate on the client

Copyright © 2001 – Jim Threatte

computer or on the data integration hub computer. A "thick client" or "server-based" approach would provide the power and flexibility to define a variety of reports and queries; however, the entity responsible for the IJIS system would need support resource(s) with expertise in data processing to provide for the more powerful, flexible thick client approach.

The second salient feature of this architecture is that the operational systems' "live" data is accessed. This fact reveals some significant architectural considerations regarding data access:

- If a target operational system is down or responding very slowly, the response can be delayed or may be incomplete. In other words, this architecture is only as strong as its weakest link in terms of system response time and availability.
- It may not be possible to map all data elements required by the data integration hub into the data elements of a target system. (For instance, if the name of an individual is stored in a free-form text field on one system, there may be no practical way to extract that data, determine whether that name is the subject of the inquiry, and return the related record.)
- Data defined by the data integration hub may not be complete. Not all data from the operational systems can be adequately mapped to a common database model. This results in some data elements being orphaned or referenced in such an obscure way that the relevance of the data is not readily apparent.

4.2.2.3.2 Alternative Approaches for Accessing Data

As indicated above, the data integration hub accesses live data from the operational systems. Copies of the operational system data are not maintained on the data integration hub server; instead, the data integration hub server can use a variety of middleware technology approaches to access data from the live, operational systems. Middleware technologies that can be employed to access data include the following:

- Open DataBase Connectivity (ODBC) database access
- Data gateway access
- Native application emulation data access

<u>Open DataBase Connectivity (ODBC)</u>

Open DataBase Connectivity (ODBC) database access is an industry-accepted standard that allows a variety of relational databases to be accessed. All of the major relational database vendors (including Oracle, IBM, Microsoft, Sybase, and Informix) support the ODBC standard for their relational database products. ODBC drivers located on the data integration hub server translate requests for data into the native SQL syntax that is supported by the target database. ODBC database access methods are very well established and consist of well-proven products from top-tier vendors. These products are

Copyright © 2001 – Jim Threatte

inexpensive and easy to administer. If changes are made to the target systems' database environment, ODBC tools are usually easy to maintain.

Data Gateway Access

Data gateway access uses a variety of middleware software to access data that is stored in structured data files on a variety of systems. Typically, this data is not stored in a relational database, but in a variety of older industry-standard data storage mechanisms such as IBM VSAM, IDMS, IMS, COBOL-ISAM files, Adabas, and a assortment of other legacy system data storage mechanisms. A variety of middleware products are available that can access data in an assortment of legacy formats. Data-access middleware tends to be well-proven products operating in a variety of environments. However, these products can be expensive to acquire. In addition, if changes are made to the target system environment, maintenance is usually required on the middleware product.

Native Application Emulation Data Access

To acquire data from the operational system, data integration hub solution vendors use native application emulation when ODBC and data gateway access middleware solutions are not available. The native application emulation middleware is designed to access the operational system by emulating a live user. The software constructs system logon, commands, and transaction or screen parameters in order to establish the necessary paths to access the data that resides on the target system. Typically, this approach involves automating terminal emulation modes and preparing the terminal data stream recognized by the target system. This includes approaches such as VT-100 terminal emulation and 3270 terminal emulation. Because of the labor-intensive nature of this approach, this method tends to require the most resources to define and implement the target system interface. This approach is also more volatile and subject to maintenance if changes are made to the host system.

4.2.2.3.3 Advantages and Disadvantages

These are the major advantages of the data integration hub architecture:

- Data access is provided to the live, operational systems; therefore, the data returned is the most current information available.
- There is a well-defined integrated data model.
- A variety of COTS query-and-reporting tools is available for use by clients to create powerful queries and reports.

Copyright © 2001 – Jim Threatte

These are the major disadvantages of the data integration hub architecture:

- Live data queries to target operational systems that are unavailable or under severe loads may cause query response times to be very long, be aborted, or return incomplete results.
- The architecture provides no mechanism for data to be transferred from one system to another in order to reduce or eliminate duplicate data entry.
- Raw and uncleansed data that is not related to a common data model is provided.
- Responses to the user from multiple systems are not integrated.
- This architecture is complex to implement and requires a higher degree of maintenance than the more simple data warehouse architecture.
- This architecture can lead to thick client query-and-reporting applications that can place a significant burden on the data network and the data warehouse server.

4.2.3 NASIRE Document Exchange Architecture

In 1998, the National Association of State Information Resource Executives[14] (NASIRE), which represents the chief information officers of the states, and the Office of Justice Programs (OJP) of the U.S. Department of Justice (DOJ) began a cooperative, grant-funded effort to facilitate governmental information sharing through a national information architecture. This project is part of OJP's larger, ongoing Justice System Integration Initiative. This effort was supported by a one-year grant from the Bureau of Justice Assistance (BJA) to NASIRE.

4.2.3.1 NASIRE Document Exchange Architecture Overview

The concept of the national information architecture (NIA) is based on the fact that each instance of governmental information sharing involves two sharing entities. The holder entity is always a governmental unit; the receiver entity may be a governmental entity or a private entity (e.g., private attorney, daycare center, or private citizen). The holder must be willing, able, and entitled to provide the information. The receiver must be able to demonstrate a need for the information, be able to receive it, and agree to abide by usage rules set by the holder. No holder of information is forced to share.

The focus of the NIA is on sharing dynamic, structured information. Dynamic information is information that is subject to change under the normal business processes. It includes arrest reports, wanted person reports, and parole status reports. It does not

[14] The National Association of State Information Resource Executives (NASIRE) represents state chief information officers and information resource executives and managers from the 50 states, 6 U. S. territories, and the District of Columbia. State members are senior officials from any of the three branches of state government who have executive-level and statewide responsibility for information resource management. Additional information regarding NASIRE can be obtained on the Internet at www.nasire.org.

Copyright © 2001 – Jim Threatte

include data such as mission statements and organization charts. Structured information means that the information is rigorously defined (e.g., date of birth in a standardized format such as CCYYMMDD), rather than free text, at the point of exchange.

The NIA focus is on information used in transaction reporting (a police department reporting an arrest to a prosecutor) and data collections (monthly summary reports) and intentionally excludes batch reporting (deferred transmissions of transactions). The shared information may be character-based, photographs or graphics, fingerprints, page images or facsimiles, or any information that can be reduced to a form that can be sent between computers through a telecommunications network.

Information sharing requires several different modalities. Some information systems need the ability to push preagreed-upon information to another governmental branch or agency (prosecutor system files charges to a court system). Other information systems need the ability to pull preagreed-upon information (arrest booking system automatically retrieves subject's criminal history record). Some information system users need the ability to query systems, and these systems need the ability to respond to queries (e.g., check for outstanding warrants). Some information systems need the ability to publish notifications that their database content has changed in a way meaningful to a particular user, and the user needs a way to subscribe to such a notification service (e.g., parole officer to be notified if her client has been arrested or was the subject of a traffic stop). The NIA must be rich enough to support all these modalities.

In order for an NIA to make sense and be defensible, it must be based on open standards and it must be attuned to the technological direction of information technology—especially the Internet and Internet-like solutions.

The NIA requires a sharing vocabulary, which is a definition of terms that can be defined and understood by all entities involved in the system. This does not require a common vocabulary for internal use in any governmental branch or agency. For example, the terms "incident report," "event report," and "crime report" may—or may not—mean the same thing to all involved parties; therefore, a common understanding of these terms needs to be established. The sharing vocabulary does require that a common definition for the data elements must be established. These data elements make up the documents that will be exchanged.

The NIA requires the adoption of a set of common sharing documents. Document examples might include an agreed-upon inquiry format for a subject and the resulting response format for a criminal history record.

4.2.3.2 Document Exchange Architecture

The document exchange nature of the NIA is both appealing and easy to comprehend; therefore, this architecture is referred to from this point forward as a "Document eXchange Architecture (DXA)."

Copyright © 2001 – Jim Threatte

The simplest and most direct form of the DXA has these characteristics:

- Selecting from a library of document descriptions, each entity wishing to be a holder identifies those documents that it is willing and able to deliver.
- For each such document description, it prepares a set of transformations that map storage element formats from its internal database to the format specified for the sharing document and develops the ability to assemble the elements into the specified document format and to transmit that document.
- Selecting from the same library of document descriptions, each entity wishing to be a receiver identifies those documents that it desires to be able to receive.
- For each such document description, it prepares a set of transformations that map storage element formats to its internal database from the format specified for the sharing document and develops the ability to receive that document.
- The holder entity keeps the directory updated concerning which document descriptions it supports.

The DXA system becomes operational as soon as there are any two entities that register in the directory for the same sharing documents. Thereafter, the rollout of information sharing is completely within the control of the willing participants.

NASIRE has recommended a list of minimum open Internet standards that are appropriate for the tasks of national information sharing:

- TCP and IP telecommunications standards
- HTTP transmission protocol
- XML tagged-field data structure metalanguage
- XSL and CSS style sheets for information transformations and formatting
- LDAP to access directories for passwords and permissions
- ISO 8859-1 for character sets (a–z, 0–9, etc.)
- MIME for e-mail attachment wrapping
- IPsec for transmission security

XML is a subset of the Standard Generalized Markup Language (SGML) defined in ISO standard 8879:1986, which is designed to make it easy to interchange structured documents over the Internet. XML files always clearly mark where the start and end of each of the logical parts (called "elements") of an interchanged document occur.[15]

XSL and CSS are style sheets that offer precise control over the presentation of Web pages. Not only can Web designers specify the visual effects they want, but also style sheets give control over voice, pitch, and other aspects of how the text will sound when rendered into speech. The eXtensible Stylesheet Language (XSL) has document

[15] Martin Bryan, *An Introduction to the Extensible Markup Language (XML)*, The SGML Centre, 1997.

Copyright © 2001 – Jim Threatte

manipulation capabilities beyond styling. The Cascading Style Sheets (CSS) language is evolving to provide even richer stylistic control and to ensure consistency of implementations.

The specification of XML as the data structure metalanguage is an extremely important recommendation from NASIRE. Survey results show that the states understand the value of standardized documents as the unit of information exchange. In fact, most local-to-state and local-/state-to-federal information sharing is already done in this way. The persistent problem is the proliferation of nonstandard approaches being used. The result is that most information systems must support several different information transfer data dictionaries and several information transfer document encapsulation methods.

XML provides a robust extensible encapsulation method superior to any now in use. In a related report, the organization "LegalXML"[16] has published a standard for electronic court filing. The Joint Technology Committee of the Conference of State Court Administrators (COSA) and the National Association of Court Managers (NACM) both intend to use XML as the basis for a national standard that will allow court filing through the Internet.

In addition, the FBI Criminal Justice Information Services Advisory Policy Board (CJIS/APB) formed a Joint Task Force (JTF) on RAP[17] Sheet Standardization. The JTF has updated its proposed electronic RAP sheet specification to use XML.

The vendor community has embraced and is expanding support for XML in Web product offerings, as well as in relational database product support.

The DXA has three document exchange modalities. The following subsections provide a brief overview of each modality.

4.2.3.2.1 Push Document Modality

The following functional diagram depicts an example of the DXA using a push modality. In these examples, the systems have agreed on which documents are to be provided without requesting them and what the content of those documents is. In this example, the law enforcement system automatically sends an arrest report through electronic means to the prosecution system whenever a new arrest report is created. The prosecution system has already established the desire and ability to receive arrest reports from the law enforcement system. The specific data elements within the arrest report have been defined and are transmitted, using the accepted standard.

---

16 Additional information regarding LegalXML can be found on the Internet at www.legalxml.org.

17 Report of Arrests and Prosecutions (RAP)

Copyright © 2001 – Jim Threatte

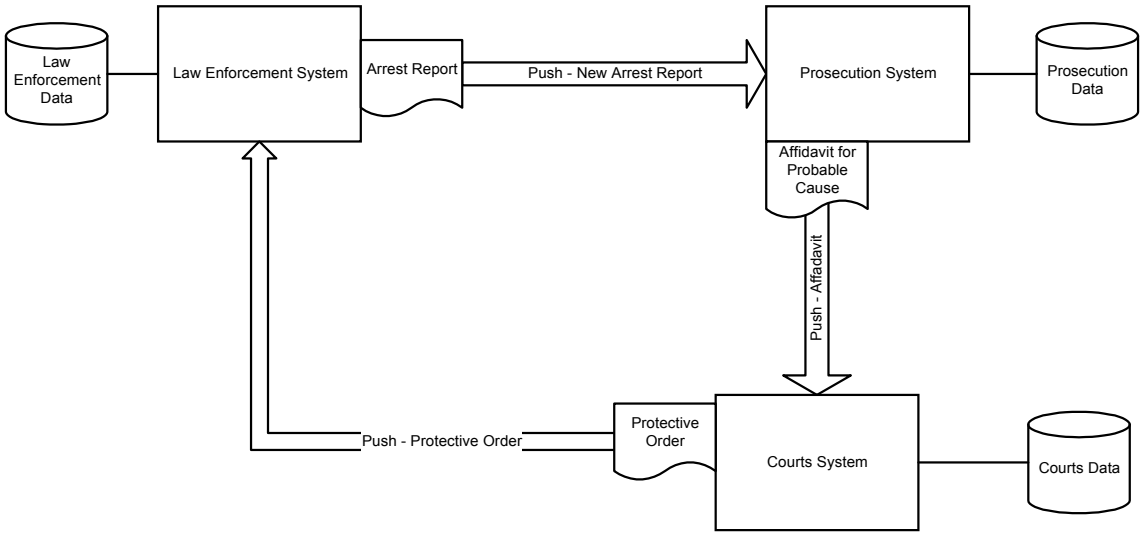


Figure 14 - Document Exchange Architecture – Push Mode

4.2.3.2.2 Pull Document Modality

The following functional diagram depicts an example of the DXA using a pull (query/response) modality. In this example, the systems have agreed on what kind of documents can be requested and which documents are to be provided in response to that request.

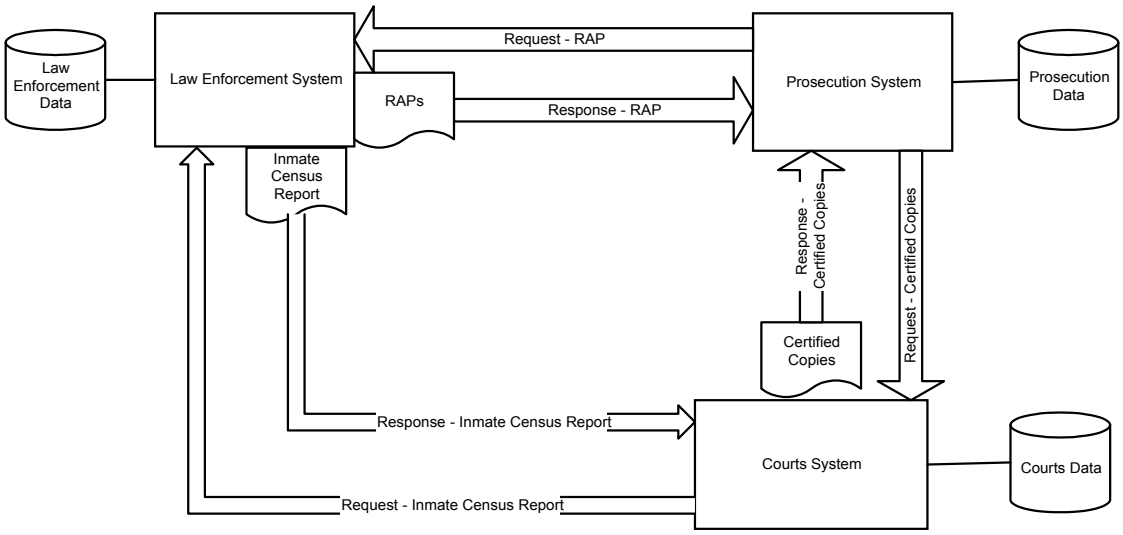


Figure 15 - Document Exchange Architecture – Pull Mode

4.2.3.2.3 Publish Document Modality

Copyright © 2001 – Jim Threatte

The following functional diagram depicts an example of the DXA using a publish modality. In this example, the systems have agreed on what kind of documents will be published and which systems desire to receive the published documents.

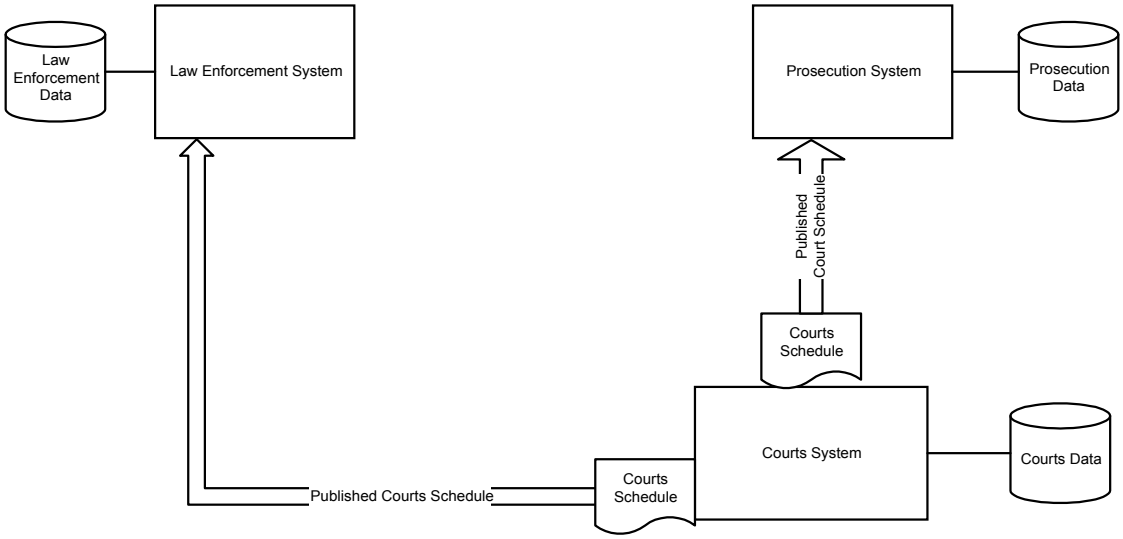


Figure 16 - Document Exchange Architecture – Publish Mode

4.2.3.3 An Evaluation of the Document Exchange Architecture

The figure below depicts several critical features of the DXA. The most critical feature that differentiates this architecture from the data warehouse and data integration hub architectures is that data is automatically provided from system to system in order to reduce or eliminate redundant data entry. In addition, this architecture provides for a variety of standard document-related queries to be automated by the application software user interface that is operating on each system; therefore, for most typical transactions, a new query-and-reporting capability is not needed. Users of legacy systems will be capable of retrieving information from other systems through their current user interface.

Copyright © 2001 – Jim Threatte

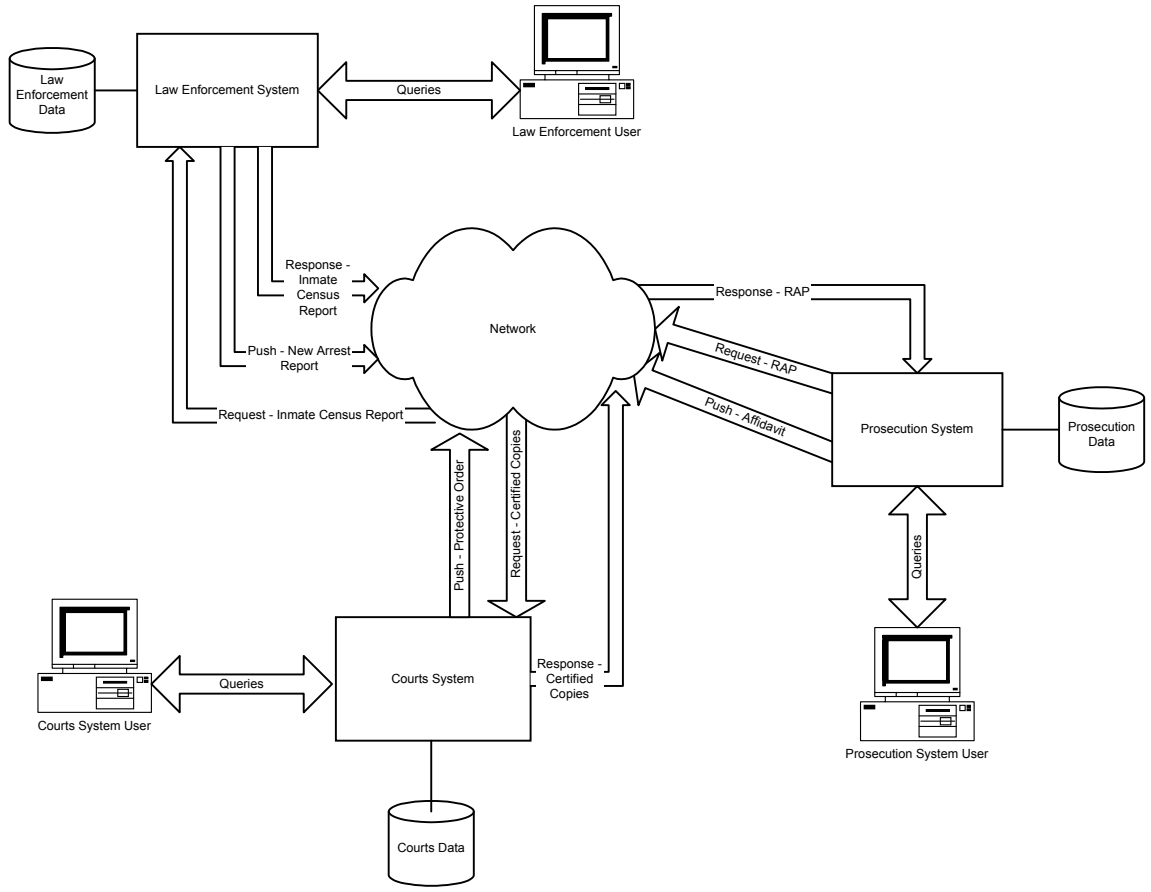


Figure 17 - Sample Document Exchange Architecture

Client Software Approach

The client software that is used by the DXA uses the existing legacy system client interface. This architecture relies on the fact that the existing software in place will be capable of retrieving the data and displaying it to the users. The data provided by other systems is processed in one of two ways:

- For data that is pushed or provided through subscription, the legacy system must store the data in its native storage mechanism. It must then provide the software necessary to retrieve this data when needed by its users and display it for their review.
- For data that is pulled, the legacy system must retrieve the response document and be able to translate it from the standard communication format to a format that is acceptable to the users on their client workstations.

The second salient feature of this architecture is that the operational systems' "live" data is provided. This aspect of the architecture reveals some significant considerations regarding the exchange of data:

Copyright © 2001 – Jim Threatte

- The document exchange method provides a one-to-one exchange of data between entities, based on their predefined definition of the document's data contents and the security aspects of exchanging the data. For example, if one version of a police report was provided to the prosecutor's office and another version was provided to the public defender's office, two separate document definitions would be established.
- Because the document exchanges are on a predefined system-to-system basis, if any individual operational system is down or responding very slowly, its response will not impact the response of other systems.

Alternative Approaches for Exchanging Data

As described above, data from the operational systems is provided, based on document exchange rules. This data is pushed, pulled, or published. The NASIRE report recommended that an XML approach be used for the definition of documents and the data within the documents. Web technology and current generation database technology products have recognized the importance of XML and have adopted, or are adopting, XML support in their products. The transmission of data elements is easily established and identified. For example, the data element "date of arrest" for June 8, 2000, might look like this:

```
<ArrestDate>20000608</ArrestDate>
```

The XML syntax carries both the information content (e.g., 20000608) and the semantic content.

Once a logical document is assembled by the legacy system, its transmission mechanism is less important. Again, industry standards and the NASIRE recommendations would indicate that TCP/IP data communication protocols are used. However, the document could be transmitted by any possible means available to the legacy system. This might even include attaching the document to an e-mail that is automatically processed by the receiving system.

Advantages and Disadvantages

These are the major advantages of DXA:

- Individual entities have complete freedom of their own system design, limited only by whatever state and local laws and policies may apply.
- Incremental agency participation is easy to support. A big bang approach is not required for an integrated system. System integration begins once two entities define and start to exchange data.
- A variety of strategies exists that will enable individual entities to participate in the integrated system.

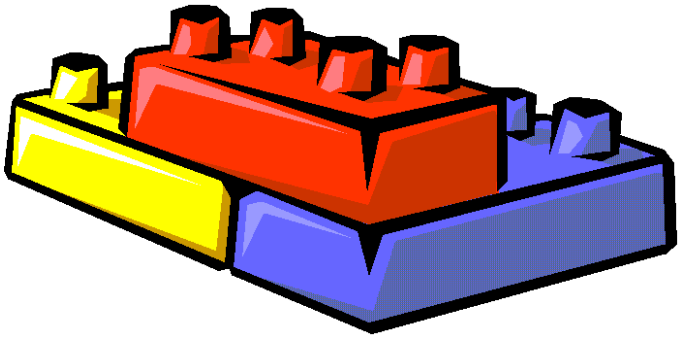
Copyright © 2001 – Jim Threatte

- A well-defined integrated data model is not required for the entire enterprise, only between the parties that agree to define and exchange a document.
- Data access is provided to the live, operational systems; therefore, the data returned is the most current information available.

These are the major disadvantages of DXA:

- Ad hoc queries and reports are not supported. Only the predefined documents with their predefined data elements can be obtained.
- Data pull requests from operational systems that are unavailable or under severe loads may cause response times to be very long or may be aborted.
- Data published when a system is down may result in that published data being unavailable at the target system.
- This architecture only works when the application software has been modified or created to support the data exchange and the reporting of the new data.
- A national standard for document exchange that defines the documents and their data elements has not been established.

5 Sample Integrated Justice Information System Architecture



This section describes the sample integrated justice system architecture that is recommended for serious consideration by any initiative or project dedicated to the integration of justice data. This architecture is a roadmap for a final system definition destination. One of the strengths of the architecture is that a big bang approach is not required in order to arrive at the final destination. In fact, the sample architecture provides for a variety of technology approaches to be used that will allow justice entities to join the integrated system by employing small, incremental technology solutions.

5.1 Overview

The architecture example provided below represents a unique synthesis of a variety of technology options and architectural alternatives. In order to identify this architecture in a manner that will distinguish itself and to recognize the intellectual capital invested in the research and definition of this approach, the author will refer to this architecture as the Integrated JUstice System Architecture™ (IJUSA).

Copyright © 2001 – Jim Threatte

The following figure depicts a sample IJUSA configuration. The remainder of this section describes the components of this sample architecture. (Please note that all figures contained in this document include only a few of the anticipated justice entities. These samples were provided as a subset of the actual participating entities in order to simplify the diagrams.)

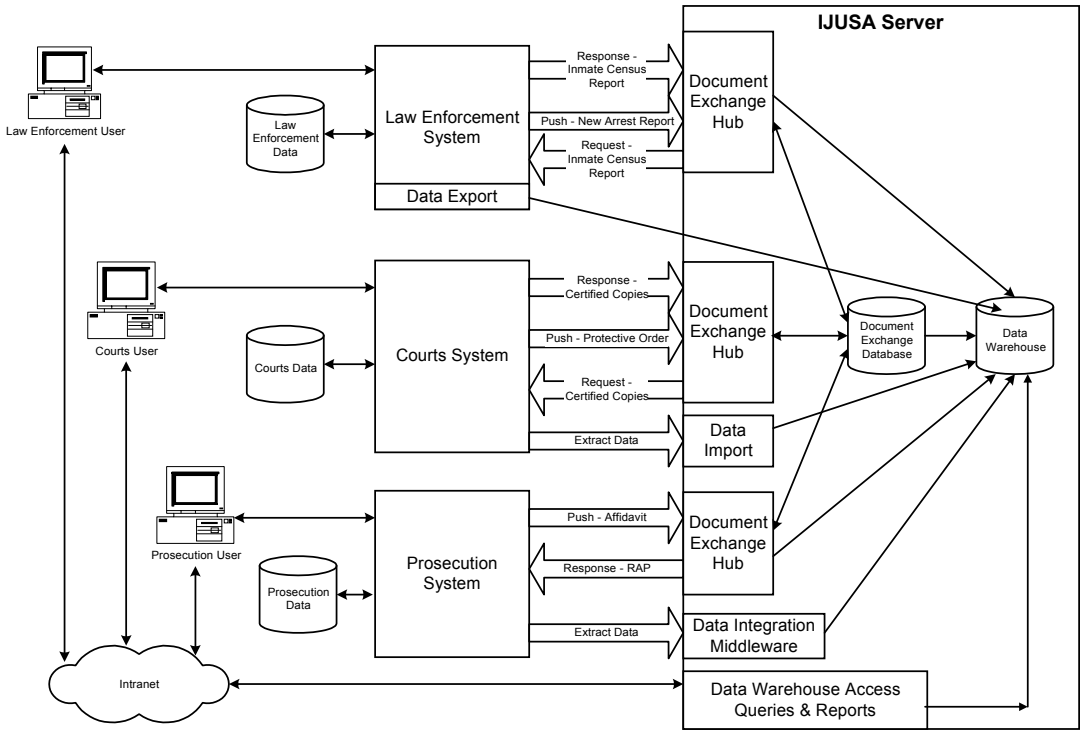


Figure 18 - Sample Architecture

One important advantage to this approach is that it reduces redundant data entry. Data that is common to many systems is entered once in the originating system. Other systems that use that data can subscribe to the appropriate document and apply the data into their systems. This process reduces the effort required to reenter the data and minimizes the errors introduced in the data-entry process.

The second component of the IJUSA is its central repository. Central repositories play two important roles in integrating justice systems: one role is to provide a resource for inquiry-and-reporting purposes, and the other role is to provide an alternative source of operational data from which the disparate justice entity applications can draw shared data. In each of these roles, the key to the value of the repository is in the consolidation of shared justice data.

As events occur in the originating justice systems, these systems publish their shared data. The data exchange facility controls the distribution and transformation of the data based upon centrally managed business rules. Each of the subscribing applications has access to this data from a single source. One of these subscribers is the consolidated

Copyright © 2001 – Jim Threatte

repository or data warehouse itself. An interface layer in the repository is responsible for consolidating data from many different sources into its common database.

The IJUSA can vastly simplify the integration process. By providing a single, consolidated data resource, many systems and their applications will subscribe to this single source of data, rather than the event data from the individual system that is the originating source of the data.

The IJUSA server also plays an important role as a source of integrated data for a variety of decision support, inquiry, and analysis functions. Its data warehouse provides a single resource for justice information that originated in many separate and independent systems. The IJUSA consolidated repository is the best way to provide a single, composite view of this data. The data warehouse provides a single point of reference for operational inquiries. This alleviates the inquirer from the burden of searching through multiple systems to find needed information. More important, by drawing from a single source of data, the information produced will be consistent across applications.

5.2 Critical Component 1 – Document Exchange

The first critical component of IJUSA is to fully adopt the document exchange architecture as the primary method for the exchange of justice information between entities and stakeholders. Adopting this approach provides several advantages:

- Flexibility – Individual justice entities can utilize software best suited for their needs. They can simply add a data extraction and import layer to the existing software in order to support the document exchange architecture.
- Immediate, lowest-cost participation – Individual justice entities do not have to acquire new systems to participate. They can employ one of a variety of strategies described in this report to support the document exchange architecture.
- Maximum immediate benefit – The document exchange methodology immediately addresses the most severe problems associated with system integration: duplicate data entry, reduced data quality and reliability, and access to critical data already available on other systems.
- Roadmap for the future – The document exchange approach will allow integrated justice projects to specify the exact requirements necessary for all future system procurements and enhancements. A requirement for any future request for proposals (RFP) can simply specify the types of documents, the data elements within the documents, and the data exchange standard as part of the procurement requirements.

Document exchange requires two major process steps: the first step is enabling the data extraction and preparing the document to be exchanged; the second step is transporting the document. The following two sections review these two process steps.

Copyright © 2001 – Jim Threatte

5.2.1 Enabling Data Extraction and Preparing the Document To Be Exchanged

These are the three principal technology approaches for enabling the document exchange and preparing the document to be exchanged within the IJUSA architecture for any given justice entity:

- Modify existing software
- Request that the software provider modify the existing software
- Develop a new gateway application

Each approach is described below:

Modify Existing Software

Some justice entities have access to, and rights to modify, the source code for the application software that they use. In addition, these entities have the software development resources necessary to modify the existing system. To accomplish data extraction and document preparation, new software routines are added to create a document exchange record at a variety of appropriate points within the existing business logic. Also, software would then need to be created that would transmit the document exchange record to the IJUSA server.

Request That the Software Provider Modify the Existing Software

If the justice entity does not have access to, and rights to modify, the source code for the application software, they can contact the product provider and supply them with a specification of the document extraction requirements. They would submit a request that the software provider satisfy these new requirements as an immediate product modification or as a general system enhancement included in a new release. With the national movement toward a document exchange standard, providers committed to the justice industry should be willing to make these product enhancements.

Develop a New Gateway Application

For those cases where the above two options are not feasible, a new gateway application can be developed, using the data integration hub and middleware technology necessary to extract the data from the target system. This gateway application would act as an agent of the operational system providing data to and from the IJUSA.

The IJUSA architecture allows each participating entity to select the most appropriate method for preparing the document to be exchanged. The entire IJUSA system will likely comprise a combination of the above methods.

Copyright © 2001 – Jim Threatte

5.2.2 Transporting the Document

Transportation of the document between systems can be accomplished in a variety of ways. The method selected will depend on the capabilities of the systems and the resources available to enable the transfer of information. These are the four principal methods for the transportation of documents between systems:

- Peer-to-peer communication
- Shared folder storage and polling
- E-mail encapsulation
- Store-and-forward server/agent

Peer-to-Peer Communication

Peer-to-peer communication is a software solution that allows computers to communicate, using custom software that connects systems. The software establishes a data communication connection between systems and ensures the reliable delivery of a document from one system to another. This approach includes SNA LU6.2,[18] TCP/IP Sockets,[19] and Windows Sockets[20] programming approaches. It requires a software development effort and is one of the most reliable, efficient, and timely methods for data communication.

Shared Folder Storage and Polling

Shared folder storage and polling is a method in which systems establish a shared "folder" on a file server where they agree that they will exchange data. A document is added to the shared folder by the sending system. Receiving systems periodically look into the shared folder (called "polling") to see whether a new document has arrived for their use. From this shared folder, they retrieve and delete documents. This approach is very easy to implement, using standard file server technology. The receipt of documents is usually not immediate.

E-Mail Encapsulation

E-mail encapsulation is an approach whereby a document to be exchanged is provided as an attachment to an electronic mail message. However, unlike most e-mail messages, this message does not originate from a human and does not necessarily go to a human. Instead, the e-mail message is automatically processed by software on the receiving

---

[18] IBM's System Network Architecture (SNA), Logical Unit (LU) 6.2, defines a peer-to-peer data communication standard.

[19] Transmission Control Protocol (TCP)/Internet Protocol (IP) is a data communication standard that includes a peer-to-peer interface specification named Sockets.

[20] Windows Sockets is the Microsoft version of the TCP/IP Sockets standard.

Copyright © 2001 – Jim Threatte

system. This method provides the reliable delivery of e-mail without the expense of a peer-to-peer software development effort. Custom software is required to process the e-mail and forward the document for extraction into the receiving system. Alternatively (although not ideal), a human can open the e-mail and process the attachment manually.

Store-and-Forward Server/Agent

A store-and-forward server/agent provides a central system repository that is responsible for receiving documents from systems and forwarding them to the appropriate receiving systems. The store-and-forward server/agent could employ a variety of software similar to the data integration hub and use middleware to effect communication among all systems.

The IJUSA architecture allows each participating entity to select the most appropriate method for transporting the document to be exchanged. The entire IJUSA system will likely comprise a combination of the above methods.

5.3 Critical Component 2 – Data Warehouse

The second critical aspect of the IJUSA is the incorporation of a data warehouse. The document exchange solution can provide great immediate value for both data entry and standard data-viewing needs. However, it does not provide access to the variety of data needed to answer questions that are more complex and support the requirements in the area of ad hoc queries. To address these requirements, a separate data warehouse needs to be provided.

The significant question to be addressed when deploying a data warehouse is how data will be received from the operational systems for storage in the data warehouse. The IJUSA addresses this, using one or more of these approaches:

- Document exchange (store-and-forward) information hub
- Data integration middleware
- Import/export processing

Document Exchange (Store-and-Forward) Information Hub

The first and principal method that will be used for the receipt and storage of data on the data warehouse is provided by an approach in which the data warehouse server computer also acts as a store-and-forward server/agent for the document exchange component of the architecture. In addition to this server providing data warehouse services, it will also be the repository for the definition of all documents that can be exchanged under the DXA. The server will manage the receipt and distribution of documents from the providers of information to all entities allowed to receive the document.

Copyright © 2001 – Jim Threatte

In addition to providing the store-and-forward services for all documents, the document exchange information hub will, as provided for by security and business logic, open the documents and add the data contained in the documents to the data warehouse database.

With this approach, the IJUSA server accomplishes two results for the cost of one transaction that provides both the IJUSA data warehouse and authorized receivers with the data contained in a document.

Import/Export Processing

Import/export processing would be provided as optional components that would extract data from operational systems on a timely basis and store that data on the data warehouse.

Data Integration Middleware

Another optional component of the IJUSA is the incorporation of data integration middleware. This component would be incorporated if the data from operational systems could not be acquired through any other practical method necessary to populate the data warehouse. The data integration middleware would not typically be used to capture live data when requested, but would be used to extract data from operational systems on a periodic basis and store that data on the warehouse. Real-time access to critical data on operational systems is provided by the document exchange (query/response) mechanism, not by the data integration middleware.

5.4 Public Access

The IJUSA provides the foundation for the entity responsible for integrated justice information to provide secure, controlled access to justice data to anyone on the Internet. Secure, controlled access can be provided to the data warehouse through a Web server computer that will act as an intermediary that establishes what information the public may request. The Web server also provides a secure path to the data that has been authorized to be released to the public.

There is a national trend toward using the Internet and Web technology to provide information for a variety of citizen, corporate, and nonprofit uses. Providing access to justice data can reap enormous rewards for the citizens as they find that they can "self-service" many requests on a responsive Web site that provides outstanding customer service to the citizens. Many public access requirements and needs are being met in various parts of the nation today. Additional requirements and needs have been voiced throughout the nation and will increase as federal legislation is passed. Some of these requirements involve victim notification, sex offender registration, Megan's law[21] registration, and background checks for a variety of purposes (including handgun

[21] The Sex Offender Registration Act of 1996, more commonly known as "Megan's law."

Copyright © 2001 – Jim Threatte

purchases, childcare workers, elder care workers, health care workers, and volunteers who work with children).

In some cases, jurisdictions throughout the nation are providing information on a fee basis to organizations such as insurance companies, lawyers, for-profit agencies, and the general public.

The following figure depicts how the IJUSA supports public access through the Internet:

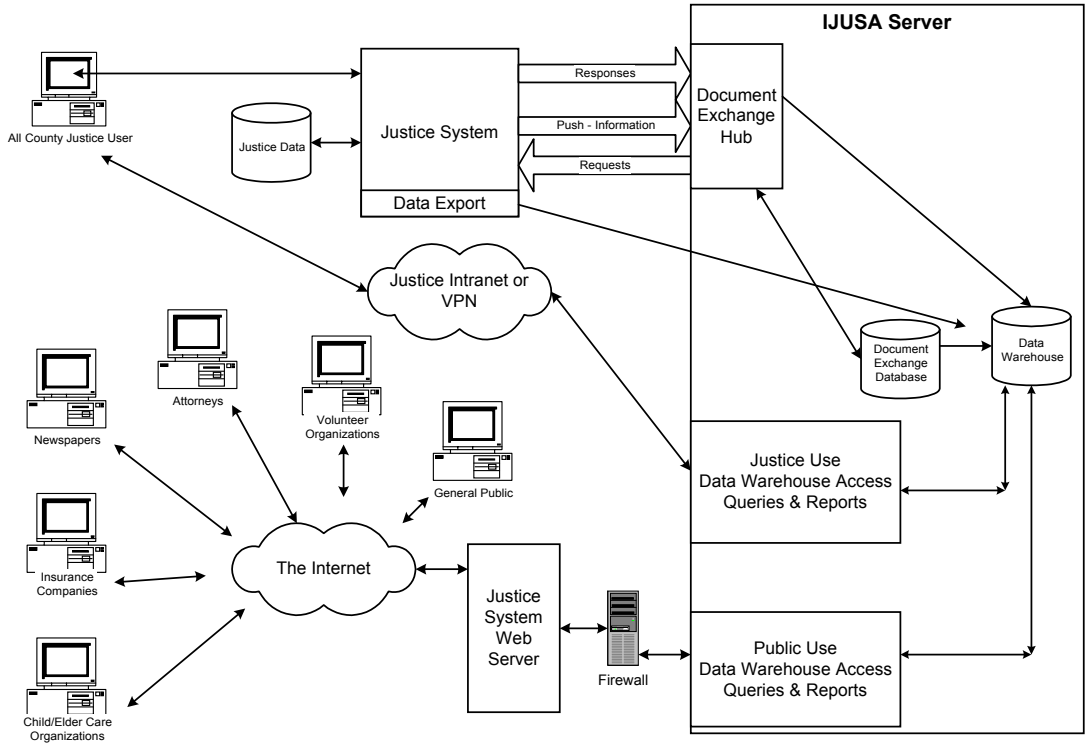


Figure 19 - Public Access

5.5 Security Considerations

The IJUSA addresses security concerns using four principal methods. The first method is that the document exchange approach is, by its very nature, secure. Only those DXA documents that have been defined for sharing with the public are made available by the IJUSA server to the public Web server. Each document and its contents are fully defined, and the provider and receiver of each document are preestablished. Data security concerns are addressed by the definition of these documents, and the authority required for the receiver to view the data is provided in the document.

The document exchange information hub will not provide documents to unauthorized systems. For example, law enforcement users might have access to a criminal history document that has been defined to include all arrests, regardless of conviction. The public

Copyright © 2001 – Jim Threatte

Web server might have access only to a criminal history document that has been defined to include only certain classifications of arrests for which a court conviction record must be present.

The second security method is that data from operational systems is added to the data warehouse, based only on the import or export software that is developed. The specification for this software will require the review and approval of the involved entities to ensure that only data that is allowed to be placed on the data warehouse will be stored on the data warehouse.

The third security method is that a comprehensive Relational DataBase Management System[22] (RDBMS) will be used on the data warehouse server. All modern RDBMS software provides for comprehensive security of data down to the element level. System logon security is used to establish each participant (or groups of participants) in the data warehouse. For example, security can be established for the "police" group or the "public defender" group. Access to data is then defined, based on the group or individual logon.

The fourth security method employs networking hardware and software solutions such as virtual private network (VPN) technology,[23] firewalls,[24] and other similar technology. These methods allow for authorized, registered users to access the IJUSA server and create a strong barrier between the Internet and the IJUSA server.

5.6 Conclusion

Technology alone will not, in and of itself, result in an integrated justice information system. Other critical components will include an executive steering committee or similar body that will represent each of the entities involved in the IJIS initiative. Governance structures and organizational strategies will need to be reviewed and established. In fact, the greatest challenges to implementing an IJIS will not be problems of technology; they will arise out of conflict over the specific impacts on budgets, organizational relationships, and established procedures. These are problems of organizational and political interests and relationships that are well beyond the scope of this paper to address.

---

[22] An RDBMS is a commercial off-the-shelf software package that is used for the storage and retrieval of data. Examples include Oracle, IBM DB/2, Microsoft SQL Server, Sybase, and Informix.

[23] A VPN is one or more wide area network (WAN) links over a shared public network—typically over the Internet or an IP backbone from a network service provider (NSP)—that simulates the behavior of a secured WAN network established using dedicated leased lines.

[24] A firewall is a system (or group of systems) that enforces an access control policy between two networks. The actual means by which this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one that exists to block traffic and one that exists to permit traffic.

Copyright © 2001 – Jim Threatte

A variety of technology building blocks are available for constructing an integrated justice information system (IJIS). These building blocks are the components of the model that will represent the IJIS architecture. A variety of unique decision criteria will determine which optional building blocks are selected in order to construct the IJIS required by any given jurisdiction. Most building block combinations are valid and will represent the requirements of the entity establishing the decision criteria that selected the building block components.

The roadmap established by the IJUSA architecture presented as a sample in this report provides the flexibility necessary for individual justice entities to plan and implement their migration to participate in an integrated system. The sample architecture allows for a variety of technical approaches that will enable each individual justice entity to participate in the integrated system.

Whichever IJIS architecture a user constructs with these building blocks, by creating a model, the user will create the roadmap that will be used to guide his or her IJIS program to its final destination.

Copyright © 2001 – Jim Threatte