



Solution Architecture PPM Version 2.0

<Project or Solution Name>

U.S. Department of Housing and Urban Development

<Month Year>



Document History

<Provide information on how the development and distribution of the Solution Architecture is controlled and tracked. Use the table below to provide the version number, date, author, and a brief description of the reason for creating the revised version.>

Version No.	Date	Author	Revision Description



Contents

1.	Solution Overview	1
1.1	Future Architecture	1
1.2	As-Is Architecture	2
1.3	To-Be Architecture	2
1.4	Transition Planning	2
2.	Architecture Goals and Constraints	3
2.1	HUD Enterprise Architecture	3
2.2	Architectural Assumptions and Decisions.....	3
2.3	Solution Architecture Attributes	3
2.3.1	Technology	4
2.3.2	Patterns	4
2.3.3	Common Services	4
2.3.4	Common Components	4
2.3.5	Portability	4
2.3.6	Capacity	5
2.3.7	Performance.....	5
2.3.8	Availability and Reliability	5
2.3.9	Scalability	5
2.3.10	System Management, Monitoring, and Administration	5
2.3.11	BC/DR and COOP	6
2.3.12	Other Solution Architecture Attributes	6
3.	Application Architecture	7
3.1	The System Architecture	7
3.2	Application Layer	7
3.3	Common Service Specifications	7
3.4	Component Models	7
3.4.1	Component Relationship Diagrams	7
3.4.2	Interaction Diagrams.....	8
3.5	Walk-Through Models.....	9
4.	Data Architecture	11
4.1	Data Flow and Context Diagrams.....	11



4.2	Conceptual/Logical Data Model.....	11
4.3	Authoritative Data Sources	11
4.4	Standard Data Elements	11
4.5	XML Resources Guidance.....	11
4.6	Data Migration Guidance	11
4.7	Data Dictionary (DD)	11
4.8	Electronic Records Management.....	13
5.	Security Architecture.....	14
5.1	Security Solution Overview	14
5.2	Security Architecture Goals and Constraints	14
6.	Infrastructure	15
6.1	Deployment Models.....	15
	References	18
	Appendix A - Key Terms	18

1. Solution Overview

<Provide one or more diagrams depicting an overview of the target solution architecture with supporting narrative text. Ensure that the diagram(s) depict the major components of the solution and the relationships between the components, input and output data flows, major processes, functions, and system tasks. Identify major HUD Commercial-Off-the-Shelf (COTS), infrastructure, and platform technology components.

Figure 1 below is a sample solution architecture overview diagram.>

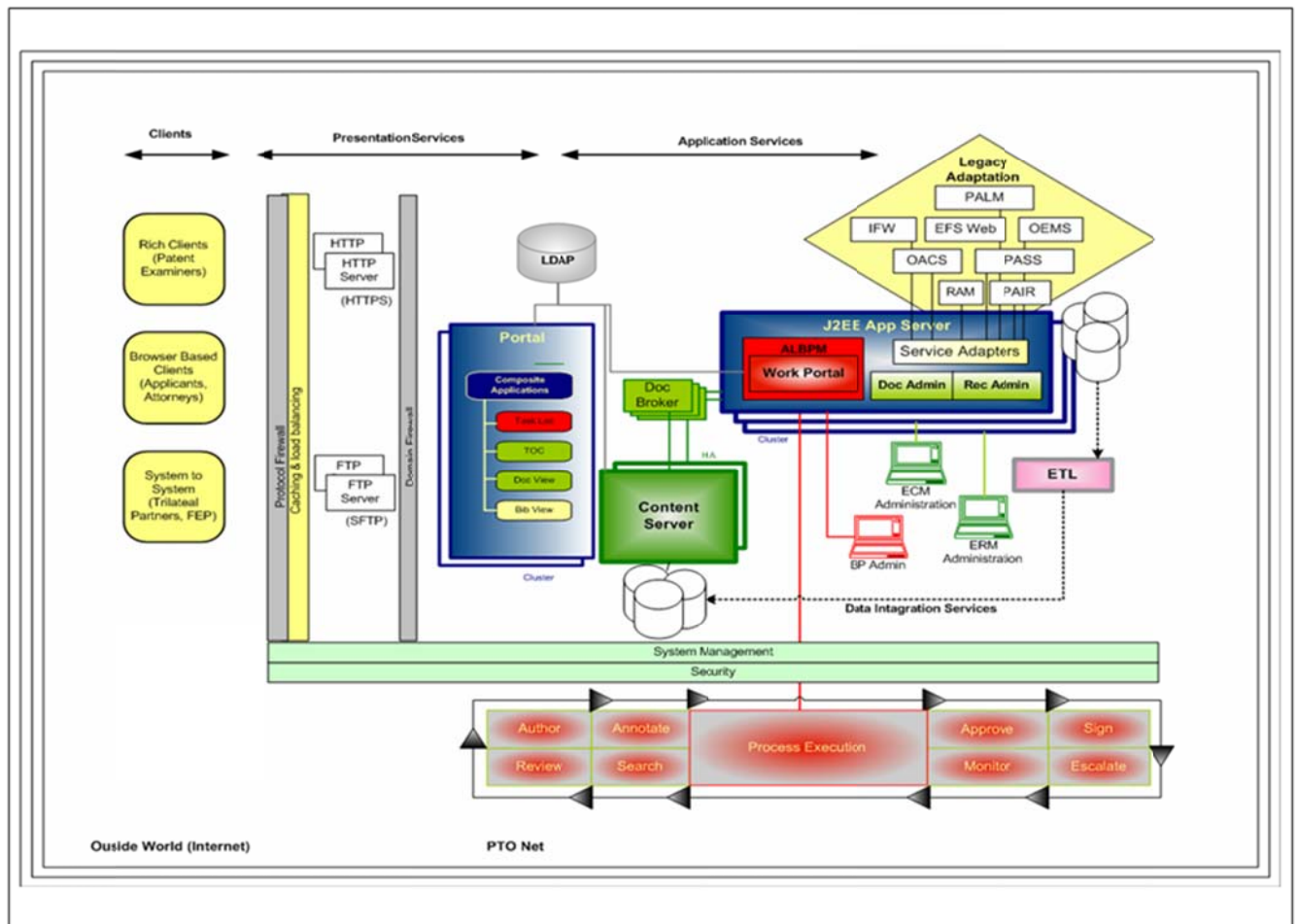


Figure 1 - Example Solution Architecture Overview Diagram

1.1 Future Architecture

<Provide future architectural solution guidance, if necessary. Identify and describe the application of potential new architecture components such as a rules engine, web service, or media server to address new solution architecture requirements.>



1.2 As-Is Architecture

<Describe the as-is architecture as it relates to the architecture component(s) that need to be transitioned.>

1.3 To-Be Architecture

< Describe the to-be architecture to which the component(s) that require change are to be transitioned.>

1.4 Transition Planning

<If the project involves migrating from current system architecture to the proposed/new architecture provide a description of the high-level transition plans.

Identify and describe the current business process to be migrated. Identify and describe the current system architecture.

Present a high-level plan to transition current business processes and architecture to the solution architecture. Describe and define the scope of major phases for a phased implementation and transition plan. Specify how current business processes will be migrated to the solution's business processes and architecture.

Focus on the current phase of the transition plan with detailed architectural guidance and descriptions of future phases at a level sufficient to enable consideration of future requirements during development of the current phase.>



2. Architecture Goals and Constraints

The Architecture goals and constraints section provides a description of goals and constraints of the *Solution Architecture*. The following subsections each provide specific architecture goals and constraints.

2.1 HUD Enterprise Architecture

<Provide the enterprise architecture context for the solution architecture. Describe, as applicable, the relationship of the solution architecture to HUD's enterprise architecture (EA) and how the solution architecture supports the goals and constraints of EA.

The HUD EA includes the HUD strategic goals and objectives and business, service, technology architectures, and the integration of those architectures. Each layer of the EA includes performance, data, and security/privacy dimensions. The EA includes segment architectures that provide the business area perspective of the enterprise architecture. The mapping of relevant enterprise and business area strategic goals and objectives to the business processes, services, technologies, data, security, and performance measures affected by the solution are represented in HUD EA model views and information.

Reference applicable HUD EA models as provided by the OCIO enterprise architecture division. The EA models include As-Is and To-Be architectures represented in system maps produced from the EA repository. Information will include relevant business processes, data exchange packages and interfaces to automated information systems, security attributes, supporting technology (hardware and software), and services.>

http://portal.hud.gov/hudportal/HUD?src=/program_offices/cio/ea/newea

2.2 Architectural Assumptions and Decisions

<List and describe the impact of any significant and central architectural assumptions and decisions. Where applicable, provide the history of decisions and identify how and when a decision was made in context of governance. For instance, describe any architectural constraint on solution design due to the capability of a required HUD software tool.>

2.3 Solution Architecture Attributes

*<Identify solution architecture attributes required to meet solution requirements. Do not duplicate statements of requirements already cited in the solution requirements. **If the specified attributes for the system have been fully spelled out in the non-functional requirements, then the following sections may be omitted.**>*



2.3.1 Technology

<Identify all software and hardware technologies that are to be used in the solution. Highlight any proposed technology choices that are not in compliance with the target HUD EA.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

Category	Vendor	Product	Version	Status

Table 1 - Hardware and Software Technologies

For easy reference of list of HUD's Product Standards Profile

<<http://www.hud.gov/offices/cio/sdm/devlife/def/trm/bussupp.cfm>>

2.3.2 Patterns

<Describe any architectural patterns that apply to the solution architecture (e.g. the Model-View-Controller pattern).

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.3 Common Services

<Specify any existing common services to be used by the solution architecture and any new common services that will be developed for the solution.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.4 Common Components

<Specify any reusable HUD common components to be reused as part of the solution.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.5 Portability

<Specify architectural attributes and guidance to support use of components that can be easily ported to other host hardware, operating systems, and software tools to avoid an architecture tied to specific hardware, operating systems, or tools.



Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.6 Capacity

<Provide architectural guidance and solution architecture attributes required to meet capacity requirements, including storage and network capacity. Describe solution architecture attributes to address database and data storage requirements such as specification for X GB of storage for X volume of specified records.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.7 Performance

<Describe architectural attributes necessary to meet solution performance requirements such as the expected responsiveness of critical system functions or timeframe benchmarks for system functions.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.8 Availability and Reliability

<Specify the architectural attributes necessary to meet solution requirements for system availability and reliability, such as specific business hours the system must be available to its users. Also use this section to provide architectural attributes necessary to meet system reliability requirements, such as specific system redundancy and recovery from failure timeframes.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.9 Scalability

<Describe the architectural attributes necessary to accommodate forecasted growth in terms of system function transactions and volume indicated by the solution requirements.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.10 System Management, Monitoring, and Administration

<Provide architectural attributes required to meet operational and administration requirements as indicated by the solution requirements, such as reporting and logging.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>



2.3.11 BC/DR and COOP

<Specify architectural attributes necessary to meet solution requirements for business continuity and disaster recovery (BC/DR) and continuity of operations (COOP) as identified by the business area in context of the business processes supported by the solution architecture.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>

2.3.12 Other Solution Architecture Attributes

<Specify other architectural attributes necessary to meet requirements for other miscellaneous solution requirements not captured by the other defined solution architecture attribute sections.

Do not duplicate statements of requirements already cited in the solution requirements. In particular, if the specified attributes for the system have been fully spelled out in the non-functional requirements, then this section may be omitted.>



3. Application Architecture

The application architecture section describes and defines the solution's application architecture, including identifying, describing, and defining major solution components and their relationships. Components defined and specified by the models included in the application architecture may include both custom and COTS components integrated into the solution architecture.

3.1 The System Architecture

The System Architecture is an enterprise architecture solution for visualizing, analyzing, and communicating enterprise architecture and business process analysis. This solution provides decision support, process optimization, and integration into solution delivery. The System Architecture addresses all aspects of your organization's enterprise architecture, including modeling, publishing, analysis, and execution.

3.2 Application Layer

<Organize system components by application layers.>

3.3 Common Service Specifications

<Provide service specifications to describe and define how all common services identified by the solution architecture common service attributes are integrated into the solution's application architecture.>

This section is optional and is only required for solutions integrating common services as identified by the solution architecture common service attributes.>

3.4 Component Models

3.4.1 Component Relationship Diagrams

<Provide component models illustrating static views of component relationships, such as Unified Modeling Language (UML) component diagrams, for the components identified by the application architecture. Include models that illustrate relationships of components across application layers supporting the solution's significant and central use case scenarios.>

Replace the example component relationship diagram below with one or more diagrams as required developing the level of detail required to provide unambiguous high-level architectural specifications and guidance to architects and designers developing the solution detailed design.

Annotate all diagrams with supporting narrative to define all objects and relationships depicted by the diagrams. These diagrams should easily cross reference with the components identified in the solution overview diagram(s) in section 1.>

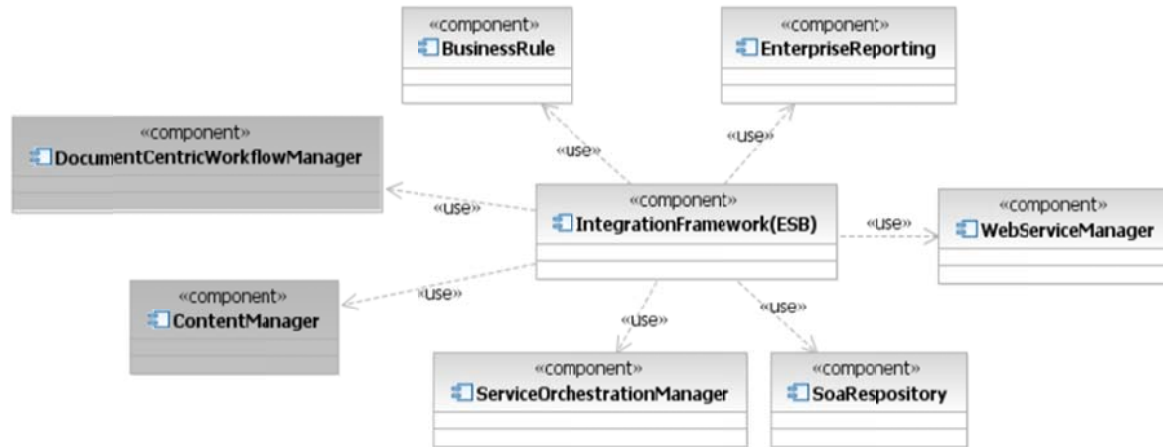


Figure 2 - Example Component Relationship Diagram

3.4.2 Interaction Diagrams

<Provide component models illustrating dynamic views of component interaction, such as UML sequence diagrams, for the components identified by the application architecture. Include models that illustrate interaction of components across application layers supporting the significant and central use case scenarios.

Replace the example component interaction diagram below with one or more diagrams as required in developing the level of detail required to provide unambiguous high-level architectural specifications and guidance to architects and designers developing the solution detailed design.

Annotate all diagrams with supporting narrative to define all objects and relationships depicted by the diagrams. These diagrams should easily cross reference with the components identified in the solution overview diagram(s) in section 1.>

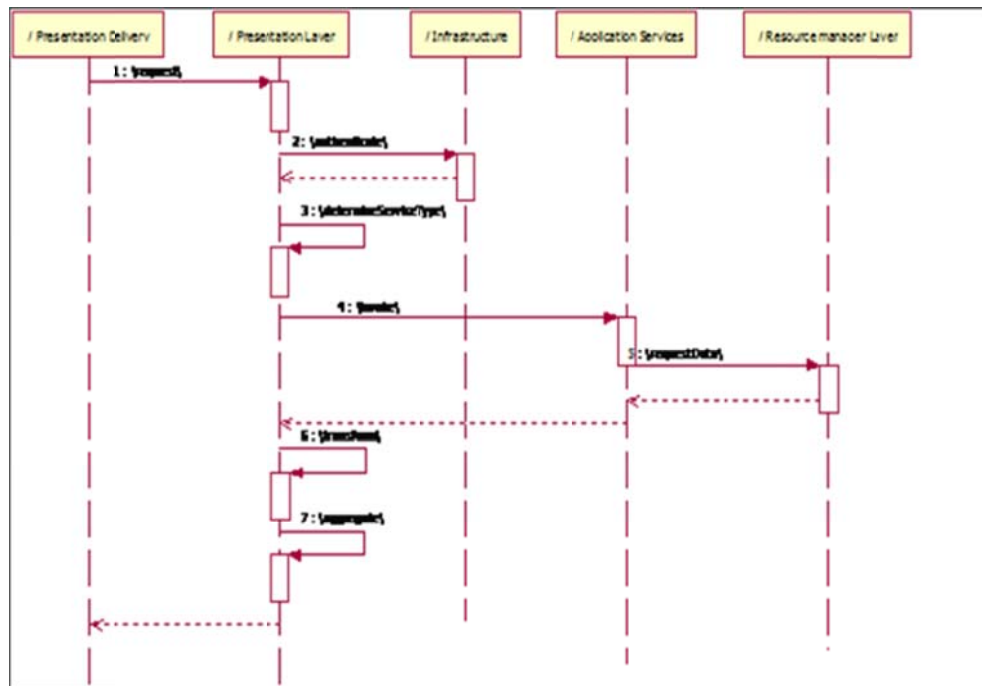


Figure 3 - Example Component Interaction Diagram

3.5 Walk-Through Models

<Provide walkthrough models to trace system execution and validate the solution's application architecture and component interaction required to implement significant and central use cases and business processes defined by the solution requirements.

Several models may be provided as required to enhance the description and validation of the solution architecture by tracing the execution of required external and internal system interfaces, data flows, and component interactions required for the system to react and process the events specified by the use cases.

Replace the example walkthrough model diagram below with one or more diagrams as required in developing the level of detail required to provide unambiguous high-level architectural specifications and guidance to architects and designers developing the solution detailed design.

For each walkthrough model describe a business scenario and process. Describe the process with a narrative as a numbered sequence of defined events, illustrated with a solution architecture diagram annotated to relate the pictured solution components to the narrative's numbered events.>

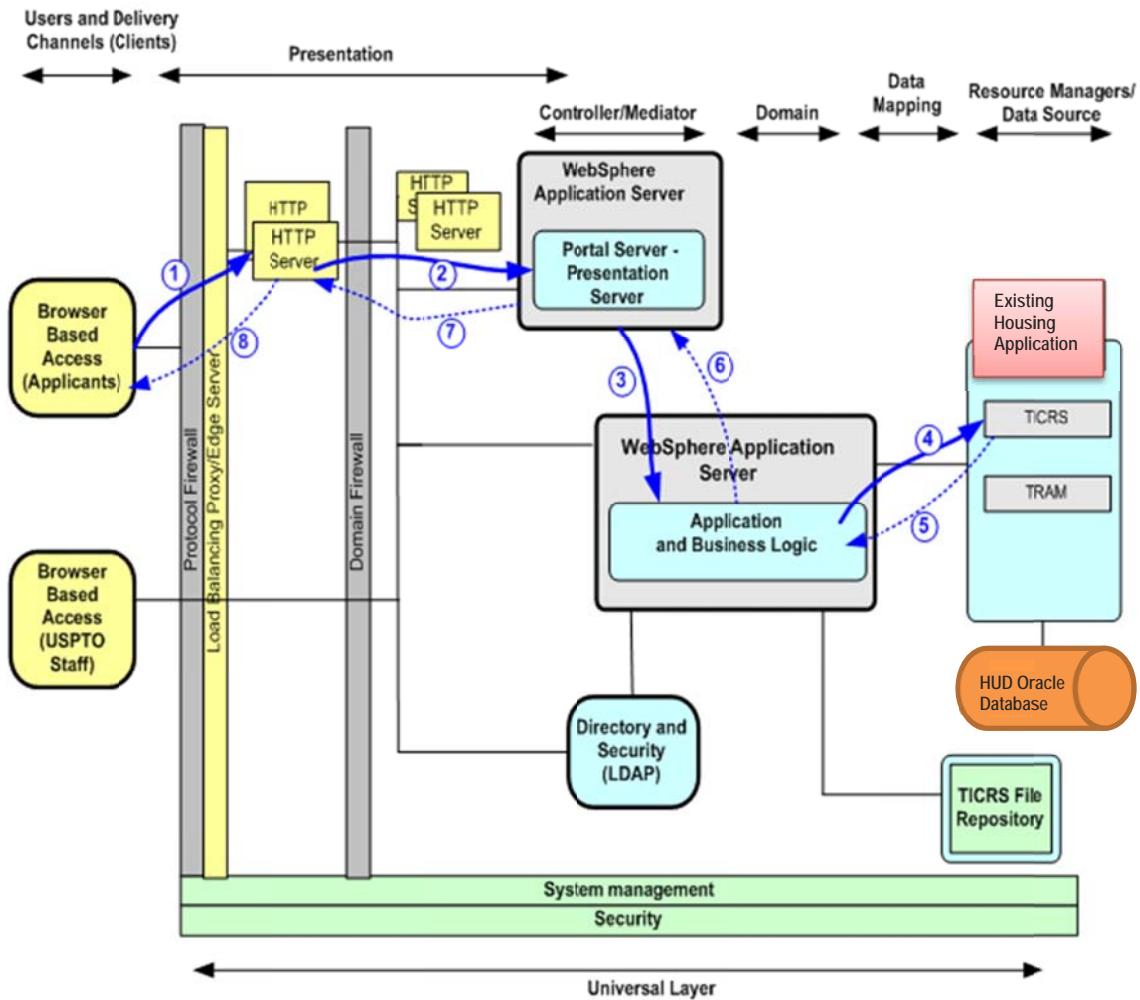


Figure 4 - Example Walkthrough Model Diagram

4. Data Architecture

All projects that are updating or designing a new data system must follow all Federal Government and HUD data requirements and standards. The Project Lead or Lead Data Architect for the project should contact the Data Stewards Advisory Group (DSAG) to review their initial data requirements and high level data architecture design to determine what Government data requirements are applicable to the project.

4.1 Data Flow and Context Diagrams

<Provide context and data flow diagrams showing data flows between a generalized application within the domain and the other entities and abstractions with which it communicates. A data flow diagram is a graphical representation of the "flow" of data both internal to the system and with external systems.

This is an optional section of the data architecture section. >

4.2 Conceptual/Logical Data Model

<Provide conceptual and logical data models. In the conceptual model, describe the logical grouping of the basic data building blocks of the solution. In the logical model describe the major processes and data requirements of the business. Note: the technical design document will extend the logical model with a physical data model to define the physical representation of the current data.>

4.3 Authoritative Data Sources

<Identify the authoritative data sources required for access during this project. This enables rapid, trusted transactions in HUD core business functions.>

4.4 Standard Data Elements

<Identify standard data element (SDE) usage and/or propose new SDEs. This promotes a consistent data standard that makes information understandable and reusable enterprise wide.>

4.5 XML Resources Guidance

<If your project is using XML, please check with the DSAG for guidance on best practices in using XML.>

4.6 Data Migration Guidance

<Provide a reference to the appropriate solution architecture transition plan to indicate data migration sequencing requirements in relationship to the solution architecture's transition from current baseline architecture to the solution's target architecture.

This section provides architectural guidance for any significant data migration and conversion effort required by the solution.>

4.7 Data Dictionary (DD)

A Data Dictionary (DD) Template provides HUD a standard DD template for use in new database systems development. It provides a standard format for loading data element information into a future



Metadata Repository (MDR). A DD template is required by the HUD System Development Methodology (SDM) as specified in the SDM Database Specifications Checklist document.

This template defines the required format and contents for that DD requirement. It will be filled out in two phases, initially after the SDM Define System phase is completed and then it is completed after the SDM Build System phase. The “When required” column identifies which DD template column should be completed during which phase.

Template Column	Description	When Required (Defined or Build)
<i>Primary Key? Y/N</i>	<i>Answers the question: Is this data element the primary or part of the primary key for the table that the data element is a part of?</i>	<i>Define</i>
<i>Foreign Key? Y/N</i>	<i>Answers the questions: Is this data element foreign or part of a foreign key for the table that the data element is a part of?</i>	<i>Define</i>
<i>Listed or Enumerated Values</i>	<i>The valid values for this data element if the data element value is constrained.</i>	<i>Define</i>
<i>Default Value</i>	<i>The default value for this data element if no value is entered at the time the record is created.</i>	<i>Define</i>
<i>Required Y/N</i>	<i>Answers the questions: Is the data element required to contain data when the record is created?</i>	<i>Define</i>
<i>Read-only Y/N</i>	<i>Answers the questions: Does this data element contain read only or static data?</i>	<i>Define</i>
<i>Integrity Requirements</i>	<i>The dependence of the data element on the existence of another data element and, if so, what the requirements of the dependency are.</i>	<i>Define</i>
<i>Format Requirements</i>	<i>Data format requirements (i.e., Social Security number must include dashes).</i>	<i>Define</i>
<i>Security Classification</i>	<i>The security classification of the data element.</i>	<i>Define</i>
<i>Business Rules</i>	<i>The constraints on the data element’s value based on a business requirement (e.g., a purchase order number may not be created if the customer’s credit rating is not adequate).</i>	<i>Define/Build</i>
<i>Data Source System</i>	<i>If this data element originates from another system, enter that system name, table name, and data element name in this field.</i>	<i>Define</i>
<i>Data Source Requirement Reference</i>	<i>The form, law, act, or other HUD requirement that this data element meets or comes from, if applicable. Some possible examples are; Sponsor Address Street Name from HUD-40112 (form), or</i>	<i>Define</i>



	<i>Section of Act 504 (2) (a), Education Institution Name (Congressional act), or Type of Assistance Transaction from the Federal Funding Accountability and Transparency Act</i>	
<i>XML Tag</i>	<i>The XML tag for this data element if the data element is part of an XML document</i>	<i>Build</i>
<i>Logical Business English Name</i>	<i>The full business mean used at HUD to identify this element and which corresponds to the data dictionary element.</i>	<i>Define</i>
<i>Data Steward Name</i>	<i>Name and contact information for the data element owning HUD Data Steward.</i>	<i>Build</i>

4.8 Electronic Records Management

<Provide data architecture attributes necessary to meet solution requirements for electronic records management and related requirements specified by the E-Government Act of 2002 § 207.>



5. Security Architecture

5.1 Security Solution Overview

<Provide a high level solution overview and description of the security architecture. Identify and describe how the security architecture meets the solution's security requirements.

The Security Architecture section is described in a security architecture model. Security architecture is based on the "Least Privilege" principle. A least privilege enterprise model designed for architectural assurance is implemented in a comprehensive access control model. As a result, logical access controls are based on the principle of role based access control (RBAC). User access is granted to the minimum resources required to perform their official job functions.

RBAC is the security requirement model to be implemented at all levels of the enterprise from network access control at the device level, database access control at the data level, application level access control, as well as user access. The infrastructure components for this model at HUD include active directory.>

5.2 Security Architecture Goals and Constraints

<Identify and describe the significant and central security goals and constraints of the solution's security architecture.

Goals:

Security Architecture goals are to provide integrity, confidentiality, availability, non-repudiation, and privacy, transparently to the user, and seamlessly to all business conducted via the HUD network.

Constraints:

Security architecture constraints include all Federal mandated security requirements from the Office of Management and Budget (OMB) and the National Institute of Standards and Technology (NIST) necessary for Federal agencies to comply with Title III of the E-Government Act of 2002 (Public Law 107-347 December 2002) entitled the Federal Information Security Management Act (FISMA). FISMA requires each federal agency to develop, document, and implement an agency-wide information security program.

For additional information please refer to the "Comprehensive Access Control Model".

Additional security documents >

6. Infrastructure

<Map application architecture deployment models to hardware and software infrastructure specifications including memory and central processing unit specifications required to meet volume and performance requirements. Include in this section infrastructure architecture guidance and specifications for:

1. **Software** < The programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation.>
2. **Hardware** < A computer and the associated physical equipment directly involved in the performance of data-processing or communications functions. Machines and other physical equipment directly involved in performing an industrial, technological, or military function>.
3. **Network** < network is a group of two or more computer systems linked together.
4. **Middleware** < Software that serves as an intermediary between systems software and an application.

Provide infrastructure architecture guidance and specifications for all environments required for developing, testing, deploying, and operating the solution.>

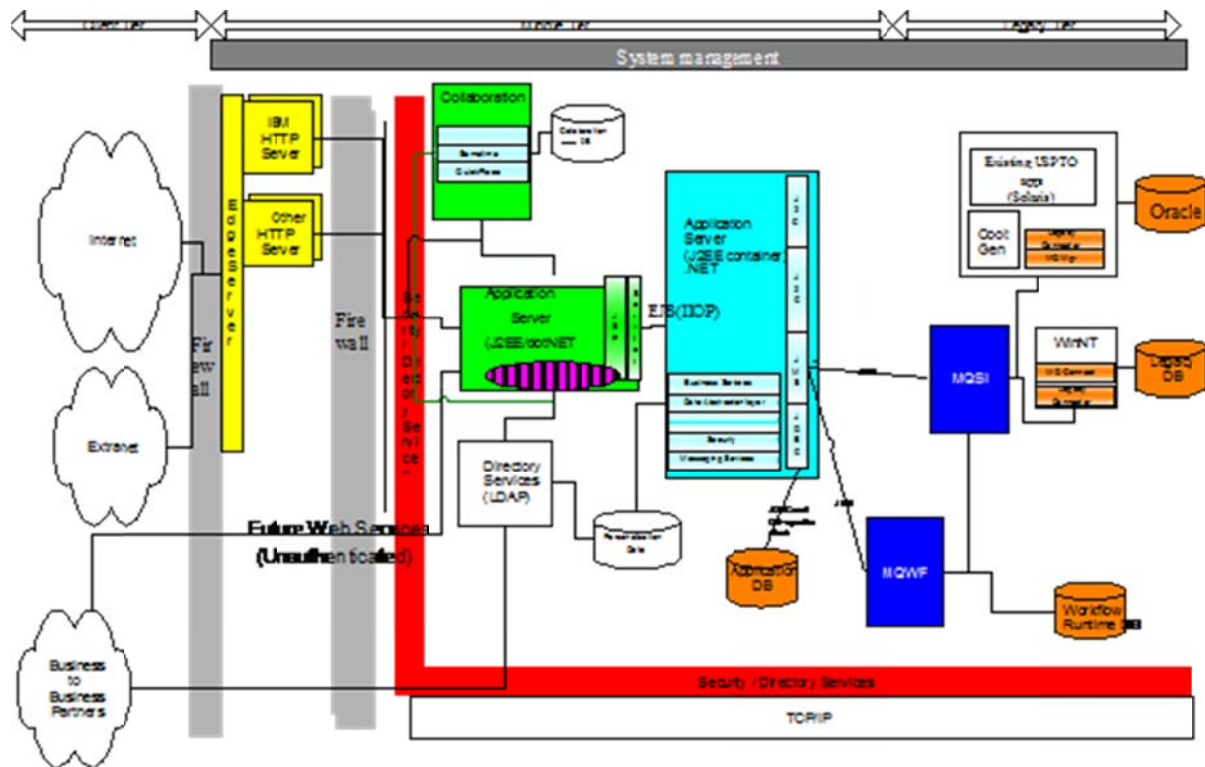


Figure 5 - Example Infrastructure Architecture Overview Diagram

6.1 Deployment Models

<Describe how the application architecture is deployed into one or more physical network (hardware) configurations.

Replace the example web application middleware deployment and Oracle RAC deployment diagrams below with one or more diagrams to illustrate significant and central components of the infrastructure architecture, these diagrams should be easily cross-referenced with the infrastructure architecture overview diagram.

Provide diagrams as required to develop the level of detail required to provide unambiguous high-level architectural specifications and guidance to architects and designers developing the solution detailed design.

Annotate all diagrams with supporting narrative to define all objects and relationships depicted by the diagrams.>

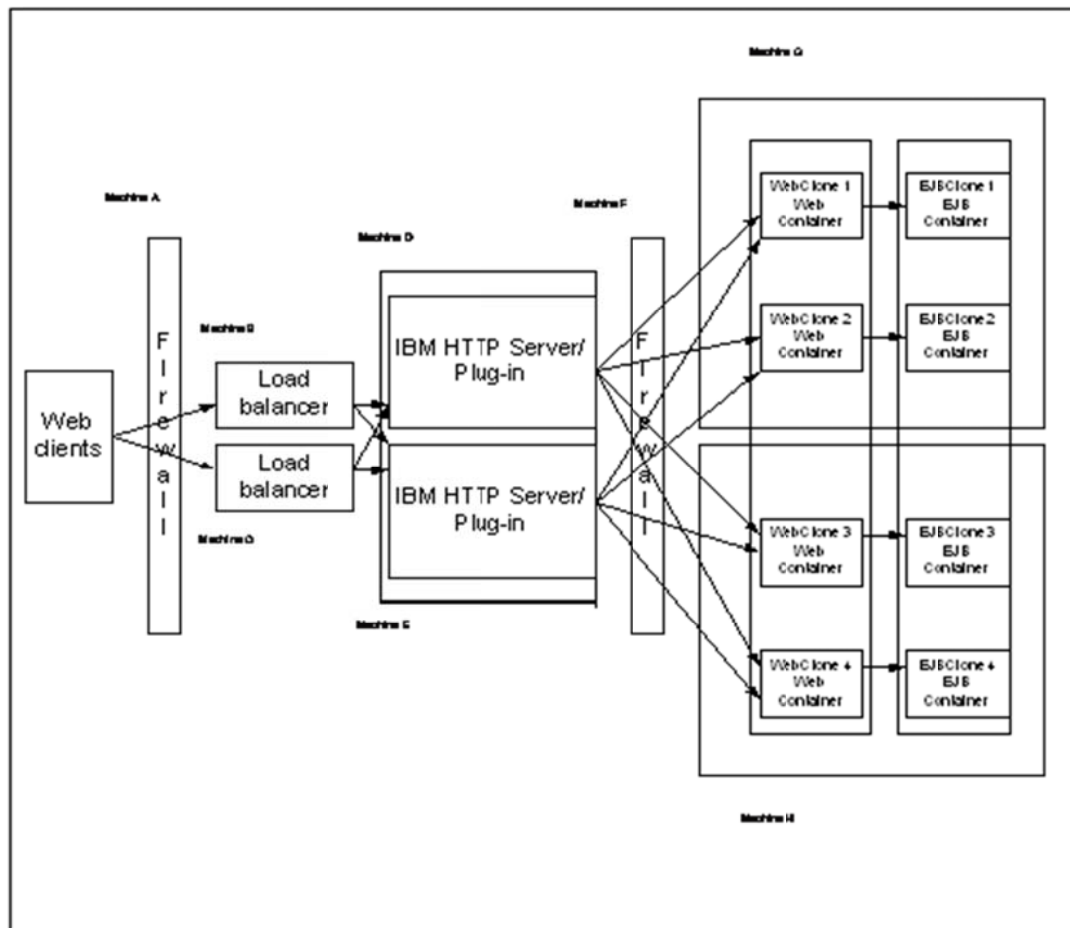


Figure 6 - Example Web Application Middleware Deployment Diagram

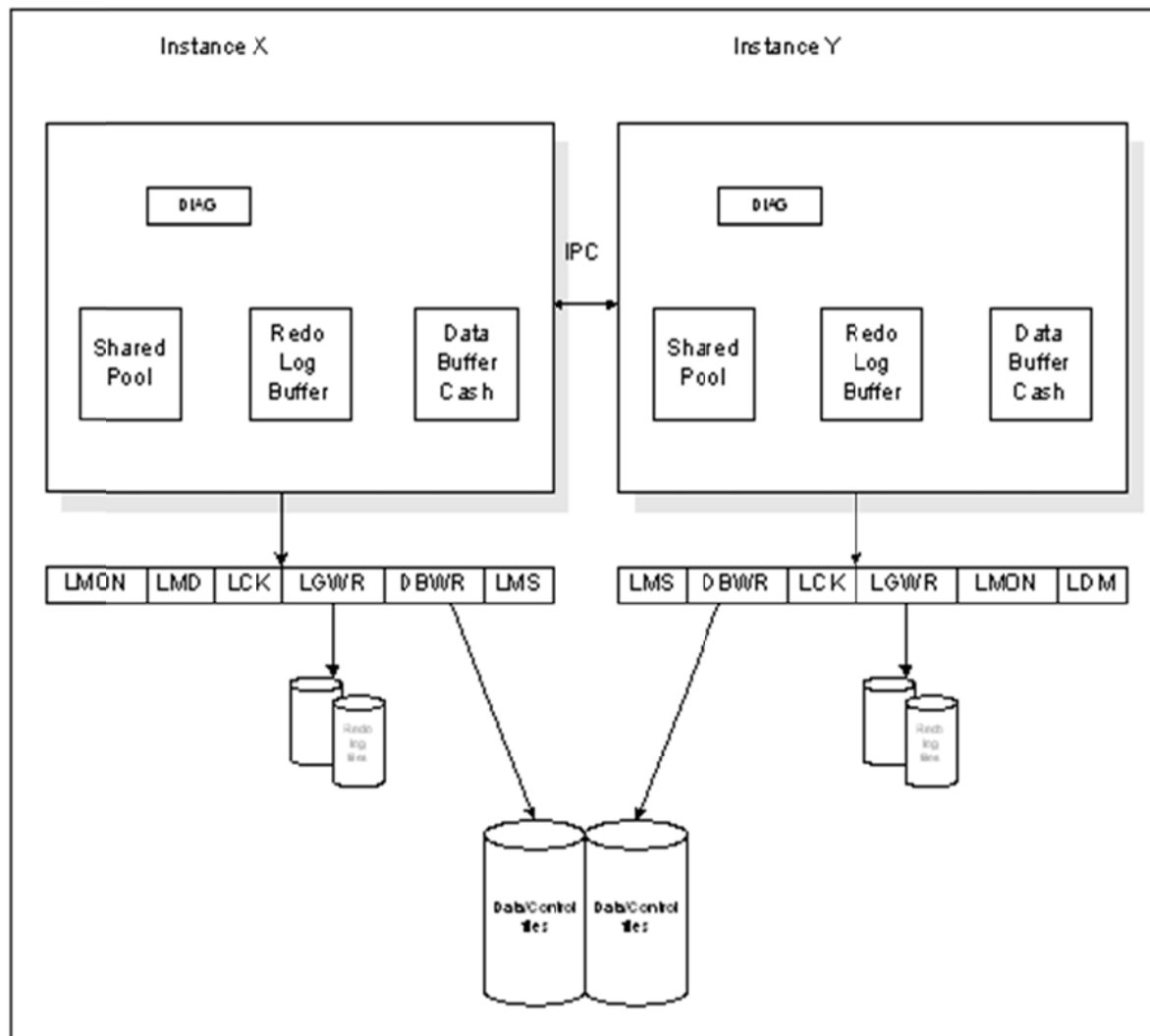


Figure 7 - Example Oracle RAC Deployment Diagram

References

<Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.>

Table 2 below summarizes the documents referenced in this document.

Document Name	Description	Location
<i><Document Name and Version Number></i>	<i><Document description></i>	<i><URL to where document is located></i>

Table 2: References

Appendix A - Key Terms

Table 3 below provides definitions and explanations for terms and acronyms relevant to the content presented within this document.

Term	Definition
<i>[Insert Term]</i>	<i><Provide definition of term and acronyms used in this document></i>
<i>Architectural attributes</i>	<p><i>Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements". Informally these are sometimes called the "ilities", from attributes like stability and portability. Qualities, that is non-functional requirements, can be divided into two main categories:</i></p> <p><i>Execution qualities, such as security and usability, which are observable at run time.</i></p> <p><i>Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.</i></p>
<i>Common service attributes</i>	<p><i>A service is defined by a set of attributes. Some attributes are common to all service instances created from one service definition, and are therefore set during service definition time. Other attributes are specific to a service instance and must be set in that instance. Some attributes can be set either in the service definition or in the instance; in such cases it is up to the service designer to determine when the attribute will be set.</i></p>

Table 3 - Appendix C: Key Terms