



State of Tennessee

Division of Health Care Finance and Administration

Tennessee Technical Advisory Services (TN TAS)

Deliverable Document

Extract: EA - BOM Management Plan

Version: Extract 1.0
March 31, 2016

CONTENTS

- 1. Executive Summary 5**
- 2. Introduction..... 6**
 - 2.1. Purpose 6
 - 2.2. Objective..... 6
 - 2.3. Scope..... 6
 - 2.4. Referenced Documents..... 9
 - 2.5. Assumptions, Risks & Constraints 10
- 3. Enterprise Architecture Vision..... 12**
 - 3.1. Enterprise Architecture 12
 - 3.2. Architecture Approvals 12
- 4. EA Development Methodology and Specifications 13**
 - 4.1. EA Development Methodology 13
 - 4.2. Specification for the Architecture Artifacts..... 13
- 5. Architecture Repository Tool & Traceability 14**
- 6. Architecture Roles & Responsibilities 15**
- 7. EA-BOM Governance 18**
 - 7.1. Architecture Policies and Standards 18
 - 7.2. External Mandatory – Legal, Regulatory and Legislative (Federal and State) Standards 18
 - 7.3. External Optional - Industry Standards 19
 - 7.4. Internal Standards 19
 - 7.5. Architecture Design Reviews 19
 - 7.6. Security Principles 19
 - 7.7. Architecture Principles..... 20
- Appendix A: Definitions, Acronyms and Abbreviations..... 21**
- Appendix B: SDLC RACI Chart Role Definition 23**
- Appendix C: EA-BOM Artifacts 25**
 - Appendix C.1: Business Process Workflow Model 26
 - Appendix C.2: Business Requirement 30
 - Appendix C.3: Business Rule..... 33
 - Appendix C.4: Conceptual Architecture Diagram..... 36
 - Appendix C.5: Conceptual Integration Architecture Model 39
 - Appendix C.6: Context Model..... 42
 - Appendix C.7: Control Artifact 45

Appendix C.8: Data Flow Diagram	51
Appendix C.9: Deployment Model.....	53
Appendix C.10: Functional Requirement	55
Appendix C.11: Logical Data Model & Dictionary	61
Appendix C.12: Non-Functional Requirement.....	66
Appendix C.13: Physical Data Model.....	73
Appendix C.14: Requirements Traceability Matrix	77
Appendix C.15: Software Services Model.....	79
Appendix C.16: System Landscape Model	82
Appendix C.17: System Process Workflow Model	85

TABLE OF TABLES

Table 1: Mapping of SI Contractor Deliverables to EA-BOM Artifacts 7

Table 2: Referenced Documents 9

Table 3: TEDS Solution Architecture Artifact RACI..... 15

Table 5: Acronyms Defined..... 21

Table 5: RACI Participants Definition 23

1. Executive Summary

This Management Plan is to be used as a guide for the development of architecture artifacts for the Eligibility Modernization Project (EMP) building the TEDS Tennessee Eligibility Determination System (TEDS) solution, specifically for the design, development and implementation (DDI) Phases with a Systems Integrator (SI) Contractor. This document is an extract of the overall Medicaid Modernization Program (MMP) Enterprise Architecture Business Operating Model (EA-BOM) Management Plan but specific to EMP.

The State has completed a series of current-state and target-state architecture models, which has guided the collection of solution requirements. The State refers to this set of architecture artifacts as an EA-BOM.

The State has issued a Request for Qualifications for Systems Integration (SI) Services (RFQ # 32101-15557) for the TEDS solution, which contains architecture tasks, processes and artifacts that the SI Contractor must complete. It is expected that the SI Contractor will maintain existing and create new architecture artifacts, managed in an architecture repository tool, and following the guidelines contained in this Management Plan.

The artifacts that the SI Contractor will build will document the solution architecture for the TEDS solution, and must be traced to the target-state architecture elements that were defined by the State. The solution architecture artifacts must also be traced to the TEDS solution requirements.

This document defines the development method, standards and templates to which architecture artifacts must adhere, the roles and responsibilities of stakeholders in the architecture development life cycle, and the design and approval process of architecture artifacts within the project System Development Life Cycle (SDLC). This document also references the governance process for the architecture function.

This Management Plan may be revised or updated to reflect an alignment between this Management Plan and the SI Contractor's methods, tools and approach to Enterprise Architecture (EA).

2. Introduction

2.1. Purpose

This document will guide the methodology and development of EA artifacts produced during the DDI Phase of the Eligibility Modernization Project (EMP) building the TEDS solution.

This document describes:

- the methodology that will be used to manage the development of EA artifacts
- the specifications and templates that will be used to define EA artifacts
- the roles and responsibilities of all stakeholders in the EA life cycle
- how architecture tools and the architecture repository will be used during the EMP
- how the enterprise architecture will be governed during the EMP
- how the EA artifacts will be approved through the SDLC design review process.

2.2. Objective

EA is the fundamental organization of an enterprise embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution. The benefit that the State will achieve by having an EA framework to guide the development of the MMP projects include:

- the ability to align solution design to business goals and objectives, and to other projects
- clarity and consensus among stakeholders with respect to current and future state business and Information Technology (IT) descriptions and design
- integration of different architecture perspectives into a coherent set of blueprints
- the ability to support effective change management and impact analysis with well-defined business and system descriptions
- the ability to capture the EA artifacts in an architecture repository, allowing for their reuse in other MMP projects, to support the development of a Health Care Finance and Administration (HCFA) enterprise architecture

2.3. Scope

The scope of this Management Plan covers those EA artifacts, tasks, and processes that must be completed by the SI Contractor during the DDI Phase of the TEDS solution.

The table below summarizes the EA artifacts that will be produced by the SI Contractor according to the Request for Qualifications for Systems Integration (SI) Services (RFQ # 32101-15557), Section A.26, Table 6, Deliverables. The table below (Table 1) contains the artifacts that will be produced at a minimum to document the solution architecture for the TEDS solution. The SI

Contractor may have other EA artifacts that will be included in their RFQ deliverables - an alignment exercise may be required to map the SI Contractor EA artifacts to these HCFA EA standards (in Column 3). The standards in Column 3 are attached to this document in Appendix C.

Table 1: Mapping of SI Contractor Deliverables to EA-BOM Artifacts

RFQ Deliverable #	RFQ Artifacts	EA-BOM Artifact Specification
19 - Detailed Requirements Traceability Matrix	Functional/Non-Functional Requirements	Requirements Traceability Matrix
		Functional Requirement
	Non-Functional Requirement	
	Business Rules	Business Rule
20 - Requirements Specification Document	Functional/Non-Functional Requirements	Functional Requirement
		Non-Functional Requirement
		Requirements Traceability Matrix
	Business Rules	Business Rule
	Business Process Flow Diagrams	Business Process Flow Diagrams
	System Process Flow Diagrams	
21 - System Architecture Design Document	Functional/Non-Functional Requirements	Functional Requirement
		Non-Functional Requirement
		Requirements Traceability Matrix
	High Level System Design	Conceptual Architecture Diagram
		Conceptual Integration Architecture Diagram

RFQ Deliverable #	RFQ Artifacts	EA-BOM Artifact Specification
		System Landscape Model
	Technical Architecture Diagrams	Context Model
		Deployment Model
23 - Database Design Document	Logical Data Model	Logical Data Model & Dictionary
	Physical Data Model	Physical Data Model
24 - Data Dictionary	Data Dictionary	Logical Data Model & Dictionary
25 – Service Oriented Architecture (SOA) Models	Technical Architecture Diagrams	System Landscape Model
		Functional Requirement
		Non-Functional Requirement
		Deployment Model
		Interface Control Document (please see Interface/Integration Management Plan)
		Software Services Model
27 - Technical Design Document	Technical Architecture Diagrams	System Landscape Model
		Deployment Model
		Conceptual Integration Architecture Model
		Control Specification
		Interface/Integration Control Document (please see Interface/Integration Management Plan)
	Business Rules	Business Rules

RFQ Deliverable #	RFQ Artifacts	EA-BOM Artifact Specification
	Entity Relationship Diagram	Logical Data Model & Dictionary
	Data Flow Diagrams	Data Flow Diagram

The tasks and processes that the SI Contractor must complete with respect to EA are contained within the Request for Qualifications for Systems Integration (SI) Services (RFQ # 32101-15557) Section A.7 (Enterprise and Solution Architecture Management) and A.10.9 (Requirements Management).

2.4. Referenced Documents

The Requirements Management Plan is interrelated with other deliverable as illustrated in the table below. These deliverables have been considered in the design of the Requirements Management Plan and will continue to be aligned to these and other relevant MMP management plans in future iterations.

Table 2: Referenced Documents

#	D	Content Overview
1	Program Governance Management Plan	Deliverable to define the process for establishing Requirements Management Governance and the Requirements Change Board.
2	System Development Life Cycle Methodology Plan	Deliverable to define the project's SDLC to which the EA artifact approval may synchronize.
3	Requirements Management Plan	Deliverable to define the standard for MMP projects the requirements management plan.

#	D	Content Overview
4	Request for Qualifications for Systems Integration (SI) Services (RFQ # 32101-15557)	RFQ defining the State’s requirements for an SI Contractor to develop, operate and maintain the Tennessee Eligibility Determination System (TEDS).
5	Centers for Medicare & Medicaid Services – Central Data Administration, Physical Data Design Standard, September 1, 2013.	CMS standard for physical data design standards.
6	DM OP-045 Operating Procedure for Constructing Physical Table and File Names or CMS’s guide DM OP-046 Operating Procedure for Constructing Physical Column or Element Names	CMS standards for physical database naming standard.
7	CMS Standard Terms and Abbreviation List and the CMS Data Management Operating Procedures and Guidelines.	CMS standard for physical database approved abbreviations and standard terms.
8	ISO 11179	Global standard guiding the standardizing and registering of physical data elements.

2.5. Assumptions, Risks & Constraints

Assumptions

- This document focuses on those architecture artifacts that the SI Contractor must produce for the TEDS solution. Architecture artifacts produced to define the target architecture has not been included in this set of specifications.
- The life cycle and standard presented in this document assumes that development is based on a waterfall or iterative software development methodology and would need to be updated if an Agile or other software development methodology is used.

Risks

- The SI Contractor may have a different EA methodology, tools, templates and approach. An alignment exercise may be required when these new stakeholders are engaged.

Constraints

- The EA artifact specifications defined within this document are aligned to The Open Group Architecture Framework (TOGAF), Object Management Group (OMG), and the Institute of Electrical and Electronics Engineers (IEEE) standards where possible.

3. Enterprise Architecture Vision

3.1. Enterprise Architecture

EA is defined as the use of structured methods and models, also known as a framework, which are used to plan and design complex business and IT transformations or changes. The primary goal of EA is to help reduce and manage the complexity inherent in designing and planning for change. EA is the fundamental organization of an enterprise embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.

HCFA will realize an enterprise architecture capability and an EA framework through the delivery of MMP projects. Therefore each MMP project is required to produce EA work products or artifacts that the State can leverage within future MMP projects. These products will be managed in an architecture repository. This will ensure that the target state as documented in the State's target architecture models will be achieved through the implementation of a solution.

3.2. Architecture Approvals

Reviews of the architecture artifacts will be held during the EMP SDLC Gate Reviews. The architecture artifacts are reviewed in conjunction with the project deliverables.

The review will assess the development of the architecture itself (compliance to the Management Plan) and the implementation of the architecture (compliance against design). Deliverables produced during the SDLC must meet architecture standards before being determined acceptable.

4. EA Development Methodology and Specifications

This chapter will describe the EA methodology and products, and how they will be managed within the EMP.

4.1. EA Development Methodology

The State and Technical Advisory Services (TAS) Contractor have used an EA development methodology that is aligned with TOGAF 9.1 and that has produced a series of conceptual and logical level current-state and target-state architecture artifacts. The methodology produced a set of design models that drove requirements gathering and analysis for the State. It is expected that the SI Contractor will continue to develop a set of solution architecture artifacts for the TEDS solution using the State's architecture artifacts as input into the solution design.

The architecture artifacts completed by the State are from the perspective of the business, information, application, and technology architecture domains. Security architecture requirements are represented as a perspective throughout the other architecture domains – usually as attributes within the architecture specifications themselves. For example, data privacy and protection requirements are represented within the attributes of the logical data model artifact.

It is expected that the SI Contractor will continue this top-down approach from the conceptual and logical level to a physical level of design across all of the architecture perspectives.

4.2. Specification for the Architecture Artifacts

Artifacts are defined as architecture diagrams, matrices or catalogs (lists). Artifacts contain elements or objects. Artifacts are packaged into deliverables. This specification defines the architecture artifact templates that must be followed by the SI Contractor within the scope of the TEDS solution.

The specifications to produce the artifacts within the EA-BOM are described in Appendix C. All of the artifacts in Appendix C specify models, with the exception of the Requirements Traceability Matrix (RTM), which specifies a document or report containing architecture elements mapped to one another, including the element relationships.

5. Architecture Repository Tool & Traceability

The State and TAS Contractor have developed a set of current-state and target-state architecture artifacts in an architecture repository for the EMP. These artifacts have driven the identification of business and solution requirements. As a solution architecture is completed and implemented, it is expected that the SI Contractor will build the EA artifacts as described in the RFQ for Systems Integrator Services (RFQ #32101-15557) Section A.27, Table 6, Deliverables and described in Section 2.3 of this Management Plan.

The SI Contractor will be required to import the State's target-state architecture artifacts into an architecture repository tool, and develop solution architecture artifacts aligned and traced to those target-state artifacts. The SI Contractor will also be required to continue the maintenance and development of the State's target architecture artifacts within their architecture repository as defined in the scope section of this Management Plan. These architecture artifacts must also be traced to the business and solution requirements in the SI Contractor's requirements management tool. The Requirements Management Plan gives guidance to the SI Contractor with respect to requirements traceability.

The State has traced EA artifacts to requirements by using an integrated architecture and requirements management tool, called Sparx Enterprise Architect. Each artifact or model contains elements or objects in a relational database. Relationships have been created in the repository tool's database between requirement elements and model elements using Unified Modeling Language (UML) associations (for example, between a functional requirement element and a business process element). This repository also contains relationships between Sparx Enterprise Architect artifact elements (for example, between an actor element and a business process element).

The SI Contractor is not required to use the State's tools or approach for traceability between EA artifact elements to each other or to requirements. The SI Contractor is required however to create a Requirements Management Plan that will describe how this traceability will be implemented.

The SI Contractor is also required to give the State visibility to the status of architecture artifacts as required during the project through such means as reports, web pages, a query tool, or other proposed means acceptable to the State.

6. Architecture Roles & Responsibilities

The following table summarizes the key roles and responsibilities for the State and MMP Contractors during the requirements life cycle. It is important to keep in mind that responsibilities identified within the signed contractual agreement will take precedence over the responsibilities identified within this section.

Refer to Appendix B for detailed role descriptions and RACI convention used.

Table 3: TEDS Solution Architecture Artifact RACI

EA-BOM ARTIFACT	STATE					CONTRACTOR			
	HCFA BUSINESS	HCFA IS	HCFA ENTERPRISE SECURITY	PROGRAM/ PROJECT MANAGER	STS (INFRASTRUCTURE)	SI	TAS	SPMO	IV&V
BUSINESS PROCESS WORKFLOW MODEL	A	C	C	C	C	R	SR	C	C
BUSINESS REQUIREMENT	A	C	C	C	C	R	SR	C	C
BUSINESS RULE	A	C	C	C	C	R	SR	C	C
CONCEPTUAL ARCHITECTURE DIAGRAM	C	A	C	C	C	R	SR	C	C
CONCEPTUAL INTEGRATION ARCHITECTURE MODEL	C	A	C	C	C	R	SR	C	C
CONTEXT MODEL	C	A	C	C	C	R	SR	C	C
DATA FLOW DIAGRAM	C	A	SR	C	C	R	SR	C	C
DEPLOYMENT MODEL	C	A	SR	C	C	R	SR	C	C
FUNCTIONAL REQUIREMENT	C	A	C	C	C	R	SR	C	C
LOGICAL DATA MODEL & DICTIONARY	C	A	C	C	C	R	SR	C	C

EA-BOM ARTIFACT	STATE					CONTRACTOR			
	HCFA BUSINESS	HCFA IS	HCFA ENTERPRISE SECURITY PROGRAM/ PROJECT MANAGER	STS (INFRASTRUCTURE)	SI	TAS	SPMO	IV&V	
NON-FUNCTIONAL REQUIREMENT	C	A	C	C	C	R	SR	C	I
PHYSICAL DATA MODEL	C	A	C	C	C	R	SR	C	I
REQUIREMENTS TRACEABILITY MATRIX	C	A	C	C	C	R	SR	C	I
SOFTWARE SERVICES MODEL	C	A	C	C	C	R	SR	C	I
SYSTEM LANDSCAPE MODEL	C	A	C	C	C	R	SR	C	I
SYSTEM PROCESS WORKFLOW MODEL	C	A	C	C	C	R	SR	C	I

HCFA IS will be responsible for filling an architecture role during the TEDS solution with the TAS Contractor’s support. Therefore, they are responsible for implementing the EA-BOM Management Plan, operating the Technical Architecture Review Board (TARB), and standing up the EA capability while managing and delivering the on-going operations of an EA function. This includes defining EA standards and principles and ensuring that the SI Contractor’s solution architecture meets the business and solution requirements and architecture standards.

HCFA Enterprise Security is responsible for ensuring that the Security Architecture standards are defined, and that the SI Contractor’s solution architecture meets the security requirements and standards.

HCFA Business is responsible for ensuring that the target state architecture accurately represents their business need and desired outcomes. **HCFA IS** may also play the role as a source of requirements, and are therefore responsible for ensuring that their business needs and outcomes are represented.

The **SPMO** is responsible for managing the architecture design review process and ensuring projects meet associated gate review requirements.

The **TAS Contractor** is responsible for supporting HCFA’s implementation of the EA-BOM Management Plan, and helping to fulfill the roles of domain architects on the State’s behalf.

STS is responsible for ensuring that the appropriate technology standards are represented within the architecture standards.

The **SI Contractor** is responsible for defining and designing the project level solution architecture in accordance with the HCFA's EA standards and principles (which include applicable State and Federal level legal, legislative and regulatory standards along with relevant industry standards).

The **IV&V Contractor** is responsible for the review of artifacts and deliverables to ensure they meet set standards. This contractor also assesses whether the contractual agreements between all parties are met.

7. EA-BOM Governance

The purpose of architecture governance is to ensure that architecture work adheres to standards and principles, supports business strategy and provides the functionality stated.

The following section describes the types of architecture reviews that will be conducted and the principles that apply to the TEDS solution. Described below are the standards to be enforced, the reviews to be conducted and the architecture principles to be enforced by the TARB. The Program Governance Management Plan (PGMP) describes the process for implementing architecture governance, the process for conducting a review, and the template for building architecture principles.

7.1. Architecture Policies and Standards

MMP is implementing and operating the TARB. The TARB is accountable for setting and enforcing the architecture standards that will be followed by MMP projects, and may adapt or grant exemptions to these specifications as needed.

It is noted that the TARB will appoint the Information Systems Steering Committee (ISSC) which will specifically be accountable for security architecture. The following describes (but is not limited to) the types of architecture policies and standards that will be enforced by the TARB:

7.2. External Mandatory – Legal, Regulatory and Legislative (Federal and State) Standards

These standards must be met by target and solution architectures without exception. The review board is accountable for program compliance to these standards. The MMP must incorporate and comply with all applicable federal and State laws, rules, regulations, sub-regulatory guidance, executive orders, CMS TennCare Waivers, and all current Court decrees, orders or judgments applicable to the TennCare and CHIP programs (collectively referred to herein as the Applicable State and Federal Requirements). This includes but is not limited to:

1. Patient Protection and Affordable Care Act ([PPACA](#))
2. [CMS Enhanced Funding Requirements: Seven Conditions and Standards](#)
3. [HIPAA](#) – specifically Title II, with the five rules regarding Administrative Simplification: the Privacy Rule, the Transactions and Code Sets Rule, the Security Rule, the Unique Identifiers Rule, and the Enforcement Rule
4. [Freedom of Information Act](#)
5. Medicaid Information Technology Architecture ([MITA](#))
6. Minimum Acceptable Risk Standard for Exchanges ([MARS-E](#))

7.3. External Optional - Industry Standards

These standards are set outside of the organization but may be adopted and enforced by the TARB. When external industry standards are adopted it is the responsibility of the TARB to monitor and enforce compliance or grant exceptions to the standard. These standards include:

1. [Business Process Modeling Notation \(BPMN 2.0.2\)](#) – business process model notation, from the Object Management Group (OMG)
2. [Unified Modeling Language \(UML 2.5\)](#) – for data, activity and use case model standards, from the OMG
3. [HIPAA EDI](#) – for interface transaction definitions, from CMS

7.4. Internal Standards

Organizational architecture policies and standards are set at the discretion of the TARB and are based on achieving business and architecture principles. Exemptions from these standards can be granted by the TARB. Organizational or program standards include:

1. EA-BOM Standards for architecture artifacts
2. State Technology Standards (e.g., application development, technology life cycle, infrastructure stack, security etc.)
3. Risk Management and Acceptance Standards

7.5. Architecture Design Reviews

The TARB will fulfill the role of approving architecture at the SDLC Gate Reviews. The TARB will review:

1. Solution architecture compliance to Legal and Regulatory standards
2. Solution architecture compliance to organizational technology standards (e.g. technology stack)
3. Solution architecture alignment with target architecture – ensuring that solutions under development, or that have been implemented, align with requirements
4. Business, information and application architecture compliance with industry and organizational methodology standards

7.6. Security Principles

Security principles include the MARS-E 2.0. Additional security and privacy standards can be set by the TARB at their discretion can include the following:

1. National Institutes of Standards and Technology (NIST)
2. Internal Revenue Service (IRS)

7.7. Architecture Principles

The TARB is the governing body of architecture principles and standards. Architecture standards and principles will evolve through the operation of the TARB during the DDI Phase of the EMP. The solution design will need to conform to applicable architecture principles and standards.

Appendix A: Definitions, Acronyms and Abbreviations

Table 4: Acronyms Defined

Acronym	Definition
BPMN	Business Process Modeling Notation
CHIP	Children’s Health Insurance Program
CMS	Centers for Medicare and Medicaid Services
DBMS	Database Management System
DED	Deliverable Expectations Document
DDL	Data Definition Language
EA	Enterprise Architecture
EA–BOM	Enterprise Architecture Business Operating Model
EMP	Eligibility Modernization Project
HCFA	Health Care Finance and Administration
HHS	Health and Human Services
HIPAA	Health Insurance Portability and Accountability Act
IE	Information Engineering
IRS	Internal Revenue Services
ISSC	Information Systems Steering Committee
IT	Information Technology
IV&V	Independent Verification & Validation
MARS-E	Minimal Acceptable Risk Standards for Exchanges
MITA	Medicaid Information Technology Architecture
MMP	Medicaid Modernization Program
NHSIA	National Human Services Interoperability Architecture

Acronym	Definition
NIST	National Institutes of Standards and Technology
OeHI	Office of eHealth Initiatives
OMG	Object Management Group
PGMP	Program Governance Management Plan
PMO	Project Management Office
PPACA	Patient Protection and Affordable Care Act
RDBMS	Relational Database Management System
RFQ	Request for Qualifications
RTM	Requirements Traceability Matrix
SDLC	System Development Life Cycle
SI	Systems Integrator
SPMO	Strategic Project Management Office
SQL	Structured Query Language
STS	Strategic Technology Solutions
TARB	Technical Architecture Review Board
TAS	Technical Advisory Services
TEDS	Tennessee Eligibility Determination System
TOGAF	The Open Group Architecture Framework
UML	Unified Modeling Language

Appendix B: SDLC RACI Chart Role Definition

This appendix defines roles and responsibilities that key stakeholders have in the SDLC Phase activities and deliverables.

The following defines what each letter in the RACI acronym means:

(R) Responsible: Those who are primary responsible for the work to complete the task or deliverable. Only one party shall be responsible for any activity, task, or deliverable.

(SR) Shared Responsibility: Those who are charged with completing some supporting work relative to the activity or task. There may be no, one, or multiple SR parties for any activity, task, or deliverable.

(A) Accountable: Those who are accountable for ensuring the correct and thorough completion of the task or deliverable. There should be only one Accountable party for any activity, task, or deliverable.

(C) Consulted: Those whose opinions and input are sought (two-way conversation).

(I) Informed: Those who are kept up to date on progress, often only on completion of the task or deliverable (one-way conversation).

Table 5: RACI Participants Definition

RACI Participants Definitions		
State	Program and Project Management	The management team that includes the Medicaid Modernization Program (MMP) Director and assigned Project Managers.
	HCFA Business	Organizational units that oversee the policies and operations of HCFA business functions, such as member services.
	HCFA IS	HCFA IS provides support for planning, design, implementation and operation of information technologies and methodologies.
	HCFA Enterprise Security	HCFA's enterprise security, includes HCFA & contractor resources responsible for reducing the risk of unauthorized access to systems and data.
	STS (Infrastructure)	Strategic Technology Solutions provides direction, planning, resources, execution, and coordination in managing the information systems needs of the State of Tennessee. STS is a division within the Department of Finance & Administration.

RACI Participants Definitions		
MMP Contractors	TAS	Technical Advisory Services supports and advises the State in completing the Medicaid Modernization Program (MMP) by offering Organizational Change Management and Training, Operations & Maintenance Planning, System Development Life Cycle Advisory Services, Quality Management, and Enterprise Architecture services.
	SPMO	The Strategic Program Management Office provides program and project management support to the State in completing the MMP
	IV&V	Independent Verification and Validation is an independent contractor responsible for verifying that any developed systems perform as designed and will continue to operate correctly in the future. IV&V provides objective evidence that all software requirements have been implemented correctly and completely. This includes evidence that the solution produces the intended results and that all functionality is traceable to solution requirements.
	SI	The System Integrator is responsible for the design, development, testing, implementation, and the operations and maintenance (O&M) of a new system to modernize and enhance eligibility determination, redetermination, and eligibility appeals for the State of Tennessee's Medicaid program (TennCare) and Children's Health Insurance Program (CHIP, known as CoverKids in Tennessee).

Appendix C: EA-BOM Artifacts

This appendix contains the following architecture artifacts:

EA Artifact
BUSINESS PROCESS WORKFLOW MODEL
BUSINESS REQUIREMENT
BUSINESS RULE
CONCEPTUAL ARCHITECTURE DIAGRAM
CONCEPTUAL INTEGRATION ARCHITECTURE MODEL
CONTEXT MODEL
DATA FLOW DIAGRAM
DEPLOYMENT MODEL
FUNCTIONAL REQUIREMENT
LOGICAL DATA MODEL & DICTIONARY
NON-FUNCTIONAL REQUIREMENT
PHYSICAL DATA MODEL
REQUIREMENTS TRACEABILITY MATRIX
SOFTWARE SERVICES MODEL
SYSTEM LANDSCAPE MODEL
SYSTEM PROCESS WORKFLOW MODEL

Appendix C.1: Business Process Workflow Model

Business Process Model	
Name	Business Process Model
Alias(es)	Process Flow, Swim Lane Diagram, Process Workflow
Objective	The business process model visually illustrates the sequence of activities required to achieve a business outcome within the current and/or future state design. These diagrams provide the capability of understanding the business processes and procedures by orchestrating activities within and across departments, identifying key stakeholders and accountabilities. This ideally allows for better collaboration and transactions between organizational departments.
Definition	<p>A business process model describes a sequence of discrete activities starting from an initial state of the process to some defined end state. The process model is a map of all of the possible paths including exceptions and alternate paths.</p> <p>Typically, a business process is informed by events identified within state transition diagrams and service activity diagrams and elaborate on the capabilities and requirements identified within the detailed business network model. The list of state transitions and service activity diagrams may be extended to address security concerns to represent failure states or other non-normal considerations as warranted by security risk assessment.</p> <p>The standard used to document business process models is the Business Process Modelling Notation (BPMN) from the Object Management Group (OMG). This standard is a non-proprietary, multi- contractor notation.</p> <p>Process models are composed of swimlanes or just lanes, which indicates the performer of each activity within the lane. Lanes are drawn as subdivisions of a rectangle containing the process, called the pool. Pools are sometimes labeled with the name of the organization but it is best practice for the pool to be labelled with the name of the process. Lanes are also commonly used to depict system services; either all in one lane or one lane per system – it is personal preference. These types of diagrams are called system process workflows within the EA-BOM standard. Activities are strung together in a sequence is called a sequence flow, which shows the progression of activities from the starting to ending states.</p>

Business Process Model

Processes can be triggered multiple times, and even occur concurrently – each time that the starting event(s) occur, the process is triggered and does not end until one of the end states is achieved.

Processes can be composed of collapsed processes or sub-processes that expand to show more detailed activities. There is no limit to the number of levels that can be collapsed into a hierarchy of process diagrams. ‘Level 0’ diagrams as a common term used to describe the highest level of the hierarchy, where each activity on the Level 0 diagram is a Level 1 process or activity. Sub-process activities should only contain those activities entirely within the activity box on the higher level diagram, nothing before or after the activity.

Customer activities are generally external to the process and are treated as a black box. Business processes are usually triggered by an external event, or when it is time to do something, but the process that has occurred with the Customer pool is not modelled explicitly because it is invisible to the process being modelled. This is called a black-box pool, and contains no flow elements.

Customers, like other external participants, interacts with the process by exchanging messages with the process pool. The term message means any communication between the process and an external participant. These connectors are called message flows (dashed line) and can only be used between pools on the model. A sequence flow (solid line) can only be used within a pool. A diagram should only contain one sequence flow within the process pool. However, sometimes it is necessary to model multiple pools in a collaboration diagram that shows multiple process pools on the same model, with parallel sequence flows. However these become complicated for the reader and should be avoided if possible.

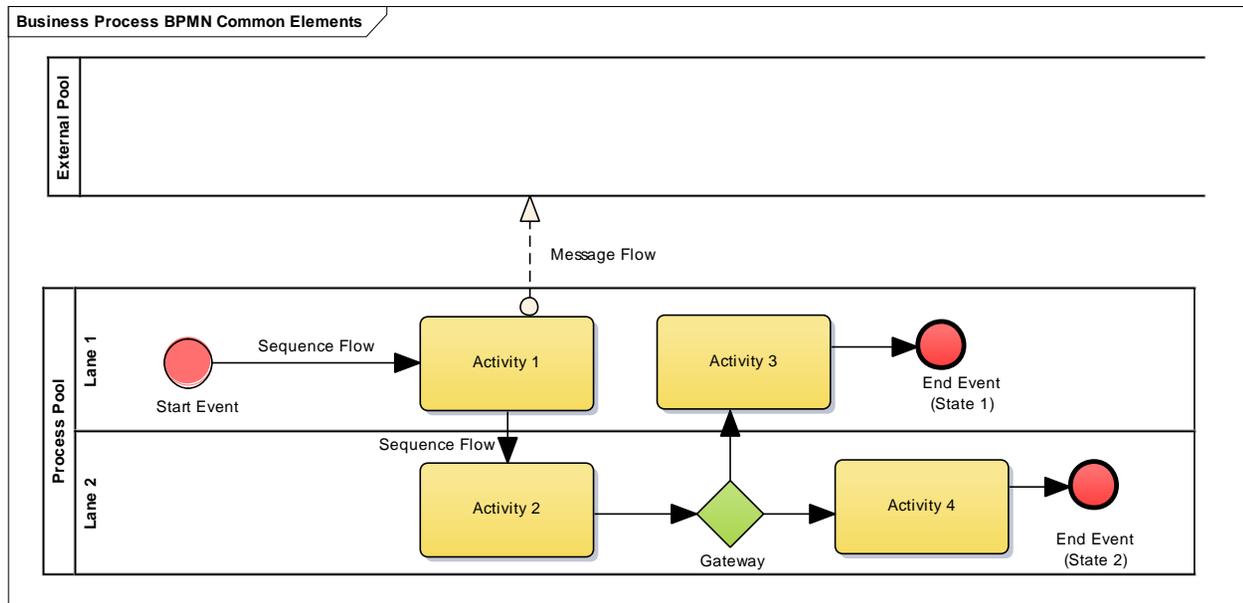
When designing a business process model, the following design standards should be kept in mind:

- Depict one process flow within a single process model
- Start with the “happy path” flow and then elaborate differences where required. If it is important to know the differences, it should be on the model – a good rule of thumb is that if it isn’t on the model, it’s not important.
- Use the sequence flow notation to illustrate the process moving from one activity to another only within the process pool
- Use the message flow notation flow to illustrate the process of moving messages across pools

Business Process Model

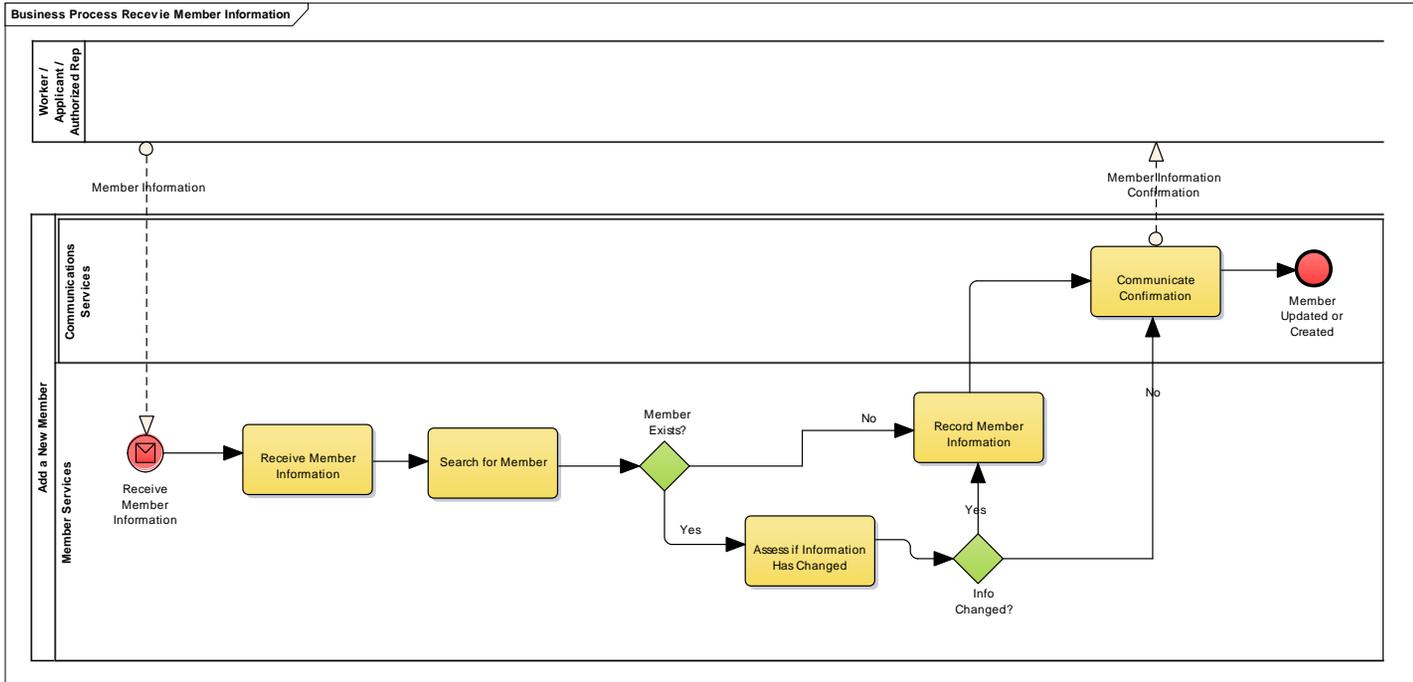
- Use the hierarchal approach, drilling down on specific and/or complex areas of interest
- Illustrate the business rules, validation rules, and calculation rules that are evaluated by the activities
- Identify all communications that are created, sent, or received during the process

The following elements shows the most commonly used notation called the Level 1 palette, or what BPMN 2.0 calls the *Descriptive Process Modelling Conformance Subclass* (event-triggered behavior requires elements outside of this palette)



Business Process Model

Sample



Appendix C.2: Business Requirement

Business Requirement	
Name	Business Requirement
Alias(es)	Requirement
Objective	The objective of a business requirement is to identify a condition or capability needed by a business to solve a problem or achieve an objective.
Definition	<p>Business requirements¹ are defined as higher-level statements of the goals, objectives, or needs of the business or enterprise. They describe the reasons why a project has been initiated, the objective that a project will achieve, and the metrics that will be used to measure its success. Business requirements describe needs of the organization as a whole, and not groups or stakeholders within it. They are developed and defined through enterprise analysis.</p> <p>Stakeholder requirements are the statement of the needs of a particular stakeholder or class of stakeholders. Stakeholder requirements serve as a bridge between business requirements and solution requirements. They are developed and defined through requirements analysis.</p> <p>Business requirements can describe past, present or future conditions or capabilities in an enterprise. Generally future state business requirements are structured to describe a capability that the business shall have.</p>

¹ International Institute of Business Analysis (IIBA®) Business Analysis Body of Knowledge (BABOK™) Version 2.0

Business Requirement	
	<p>Typically, each requirement will have the following attributes:</p> <ul style="list-style-type: none"> • Requirement ID: Unique reference that is not altered or reused if the requirement is changed • Name: Title • Type: Identifies what type of requirement (e.g. business, stakeholder, functional, non-functional, transition) • Description: Gives a detailed description of the requirement the ability that the business requires • Version: Requirement increment • Source: Identifies the origin of the requirement. The source is often consulted if the requirement changes or if more information regarding the requirement or the need that drove the requirement has to be obtained • Author: Provides the name of the person who needs to be consulted should the requirement later be found to be ambiguous, unclear, or in conflict • Owner: Indicates the individual or group that needs the requirement or the business owner after the solution is implemented • Traceability: Indicates any other requirement dependencies. Traceability can be between the same requirement types in a dependency chain, or can be a hierarchy of grouped requirements, or can be a decomposition of requirement types from business requirements -> stakeholder requirements -> solution requirements. • Status: Indicates the state of the requirement • Priority: Indicates relative importance. Priority can refer to the relative value of a requirement or to the sequence in which it will be implemented • Release: Indicates the release to which the requirement is allocated
Sample	<ul style="list-style-type: none"> • ID: BR-001 • Name: Eligibility Determination

Business Requirement

- Type: Business Requirement
- Description: The business shall have the ability to automatically determine the Medicaid eligibility of an applicant once an application is submitted to TennCare.
- Version: 1.0
- Source: HCFA Enterprise Architect
- Author: Jane Doe
- Owner: John Doe
- Traceability: Parent of solution requirement EM-001, EM-002
- Status: Verified
- Priority: High
- Release: Allocated to Release V1.1

Appendix C.3: Business Rule

Business Rule	
Name	Business Rule
Alias(es)	Policy, guideline, calculation
Objective	The objective of a business rule is to achieve consistent behavior or results.
Definition	<p>A business rule is defined as a directive intended to guide business behavior.² Rules exist independently of processes or workflow. Rules can act as constraints, calculations or facts, or as guidelines for influencing behavior.</p> <p>Requirements can reference business rules where they are enforced or invoked, and rules can be traced to the data elements that are used in calculations or when a rule is enforced. Business rules can be discovered during requirements analysis, when building fact or data models, and can be found in policy, legislation and regulation, and standards within an organization.</p> <p>Rule types include (but are not limited to):</p> <ul style="list-style-type: none"> • Constraints or restrictions - rules that must be enforced, using statements with the form 'must' or 'must not' • Guidelines – suggestions or guidelines that can be used to influence business behavior, but that may not necessarily restrict behavior or enforce a rule. These types of statements can be written with 'should'

² *Principles of the Business Rules Approach*, Ron Ross, 2003

Business Rule

statements instead of using ‘must’ or ‘must not’ statements.

- Computations - calculations, using the form ‘is calculated as’ or ‘is equal to’ or ‘is defined as’
- Inference – A condition that is met if a rule is true – for example, categorizations of customers
- Timing – Rules that must be enforced if a timing condition is met; for example, the automatic generation of calculation if a data condition is met
- Triggers – Events that are triggered if a condition is evaluated to be true, such as the automatic generation of an invoice or report.

Like requirement statements, business rules should be managed in a database and therefore have attributes that help management rules through their life cycle:

- Rule ID: Unique reference that is not altered or reused if the rule is changed
- Type: classification of the type of rule (e.g. guideline, computation, constraint)
- Statement: The rule statement
- Version: Rule increment
- Source: Identifies the origin of the rule. The source is often consulted if the rule changes or if more information regarding the rule has to be obtained
- Author: Provides the name of the person who needs to be consulted should the rule later be found to be ambiguous, unclear, or in conflict
- Owner: Indicates the individual or group that needs the rule or the business owner after the solution is implemented
- Traceability: Indicates any other rule dependencies, in a chain or relationship, or in a hierarchy or rule group, or requirements that reference the business rule.
- Status: Indicates the state of the rule (e.g. approved, cancelled, etc.).
- Style guides for writing effective business rules can be found in the Business Rules Group, but some

Business Rule	
Sample	<p>sample guidelines and rule statements are included below.</p> <ol style="list-style-type: none"> 1. A shipment must be held if any of the following are true: <ul style="list-style-type: none"> • Payment has not been received • The customer's security clearance has not been verified 2. The amount paid for an order must be calculated as the sum of all payment amounts applied to the order 3. A customer is considered high-risk if they have accounts receivable over \$1,000 4. An order for a high-risk customer cannot be approved if the order exceeds \$100

Appendix C.4: Conceptual Architecture Diagram

Conceptual Architecture Diagram	
Name	Conceptual Architecture Diagram
Alias(es)	Concept of Operations
Objective	<p>This diagram acts as a useful vehicle for effectively communicating architecture designs (e.g., “big picture” and essential design ideas) to non-technical audiences without having to explain detailed design elements. Essentially, this diagram serves as a high -level system and interaction map providing navigation around complex systems. This artifact is useful in supporting security assessments as it provides system context of systems, flows, interfaces and their dependencies.</p>
Definition	<p>A conceptual architecture diagram identifies the system components and interconnections between components and is used to describe the design ideas of a system at a conceptual level.</p> <p>This type of diagram consists of system components and interactions between these components. In-scope components can be extracted from the layers of Software Services Model (SSM). More specifically, the following layers of the SSM will contain components that should be included in a conceptual architecture diagram:</p> <ul style="list-style-type: none"> • Presentation layer – a user interface view (no storage of data) • Services layers – business, application and data • Infrastructure layers – high level components (e.g., platform, security, accessibility) <p>From the abovementioned layers of the SSM, a conceptual architecture diagram should contain components describing the major features of the system which could include:</p> <ol style="list-style-type: none"> 1) Points of Access/Communication Channels <ol style="list-style-type: none"> a) Portal b) Mobile Application

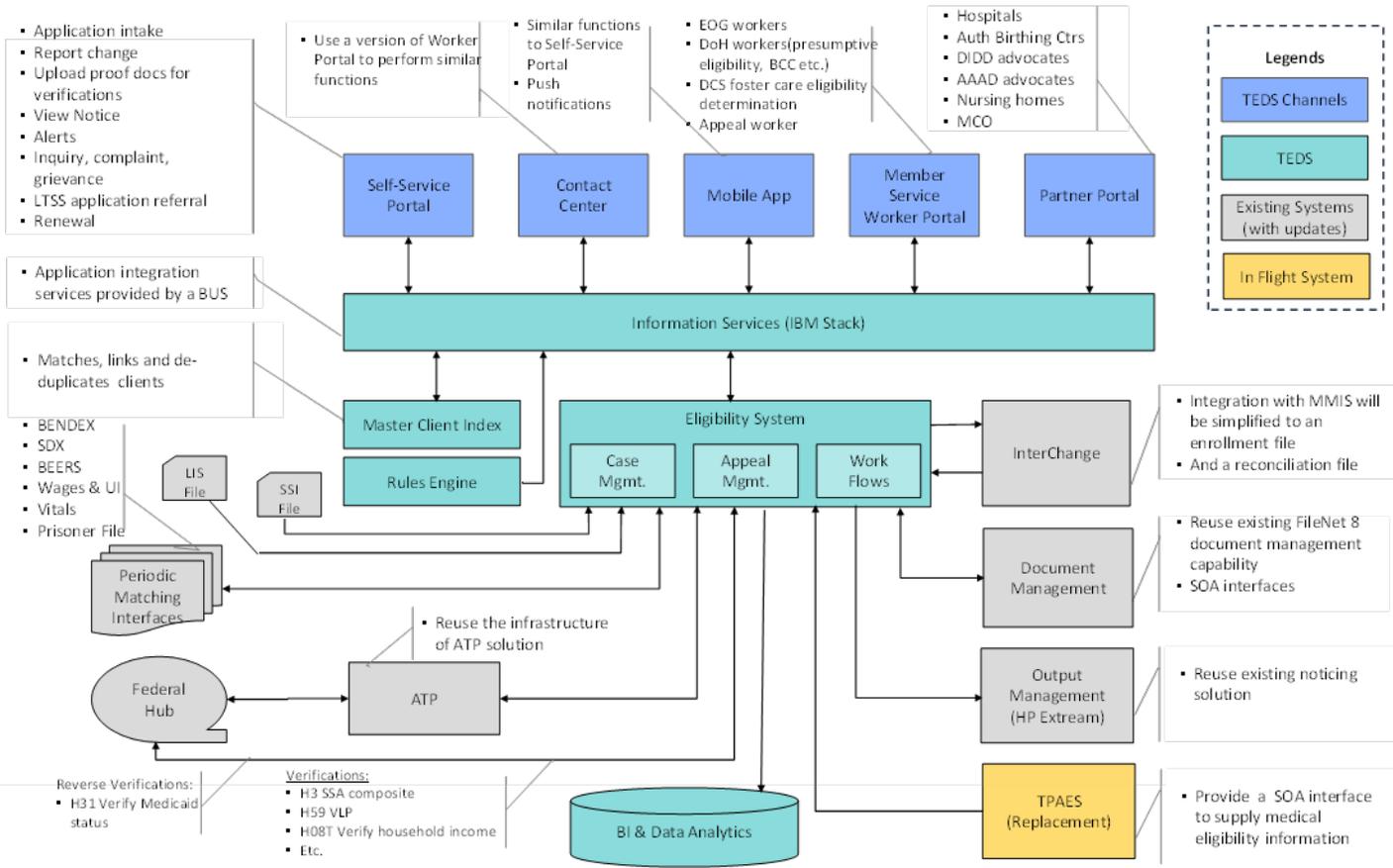
Conceptual Architecture Diagram

- c) Contact Center
- 2) Platform:
 - a) Security
 - b) Rules Engine
 - c) Information Services
 - d) Master Client Index
- 3) Major feature/capability groups of a system:
 - a) Case Management
 - b) Appeal Management
 - c) Workflow Management
- 4) Data Exchange with External Stakeholders:
 - a) Account Transfer Process
 - b) Information Verifications
- 5) Reporting and Analysis Tools
 - a) Business Intelligence and Analytics

In the diagram, system components are represented by “boxes” while interactions are represented by arrows connecting the components that interact. System components can be color-coded to show types of components and groupings.

Conceptual Architecture Diagram

Sample



Appendix C.5: Conceptual Integration Architecture Model

Conceptual Integration Architecture Diagram	
Name	Conceptual Integration Architecture Diagram
Alias(es)	Application Integration Architecture Diagram, System Integration Diagram, System Architecture Diagram
Objective	Similar to the Conceptual Architecture Diagram, the conceptual integration architecture diagram acts as vehicle for effectively communicating the high-level integration architecture without including detailed integration design specifications. This diagram provides a conceptual understanding of how system components send and receive data between internal and external components and entities. This artifact is useful in supporting security assessments as it provides system context, especially relevant to assess risks and security concerns related to sending and receiving data and the associated application and technology components.
Definition	<p>A conceptual integration architecture diagram identifies the type of integration and the associated high-level design along with key upstream and downstream internal and external systems and entities that are sending/receiving data and information.</p> <p>The conceptual integration architecture articulated in this diagram is designed to enable the high-level architecture as defined in the conceptual architecture diagram, and as such, system components from the conceptual architecture that require integration are included.</p> <p>There are various types of integration architectures defined depending on business and solution requirements, which are known to evolve as development approaches evolve (e.g., point to point integration, middleware-based integration, and SOA integration).</p> <ul style="list-style-type: none"> • Point to Point Integration – Direct integration from the sending application/system directly to the receiving application/system. A common method used when only a few applications/systems are involved • Middleware-based Integration – An integration approach that introduces an intermediate layer between the sending and receiving application/systems that provides generic interfaces through which the integrated systems are able to communicate.

Conceptual Integration Architecture Diagram

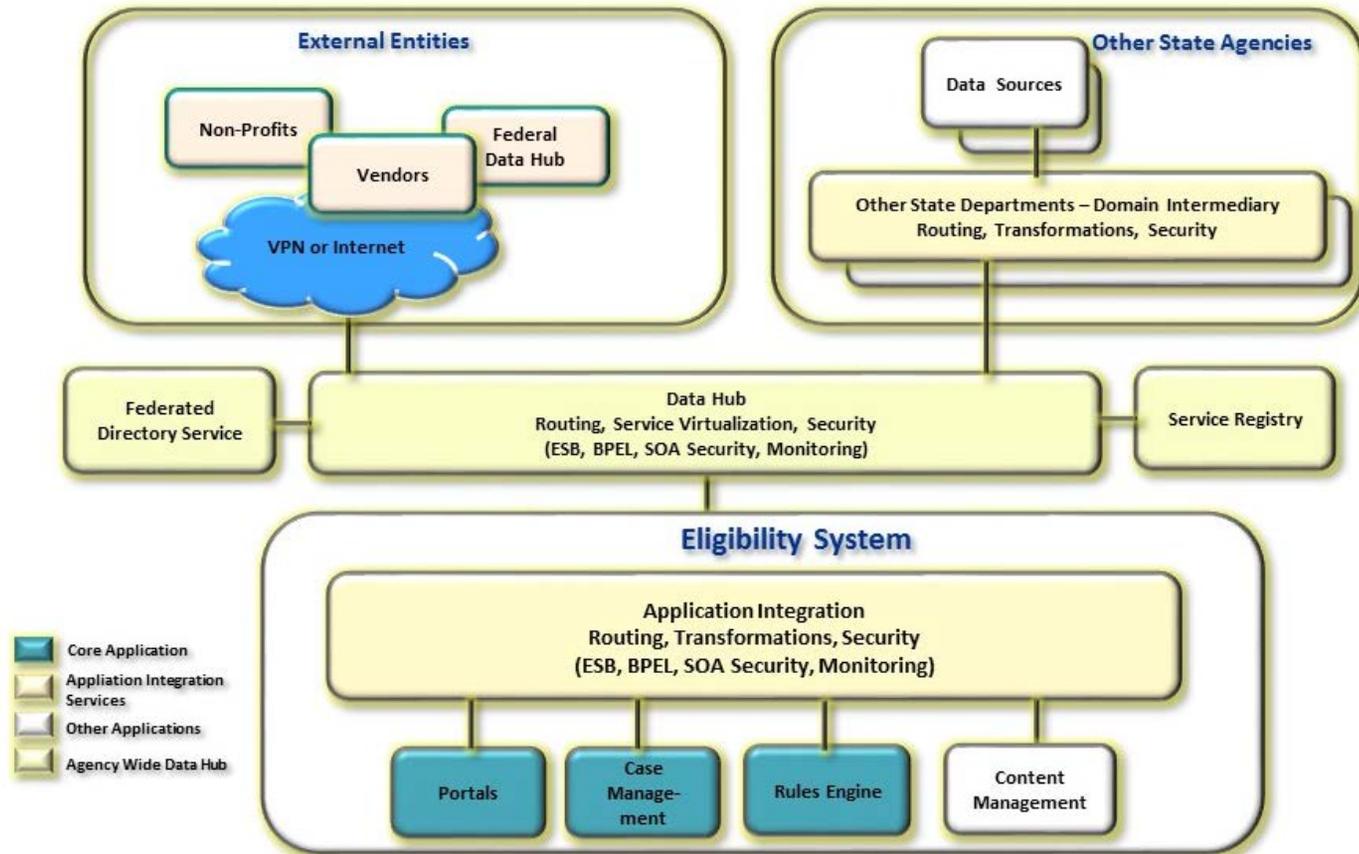
- Service Oriented Architecture (SOA) – The use of services to allow systems to communicate with one another. In this approach, enabling technology (e.g., web services and supporting infrastructure) is used to ensure seamless integration between systems, services and applications.

Depending on the requirements of the solution, the conceptual integrations architecture diagram should reflect the most appropriate approach and should align to integration architecture principles as set by the TARB.

This type of diagram consists of key system components, interactions between these components along with the integration design required to support such interactions. System components are represented by “boxes” while interactions are represented by arrows connecting the system components that interact. System components can be color-coded to show types of components and groupings.

Conceptual Integration Architecture Diagram

Sample



Appendix C.6: Context Model

Context model	
Name	Context Model
Alias(es)	Conceptual Blueprint, Context Diagram
Objective	The objective of a context model is to identify which capabilities are in scope and defines the interactions between the organization and key stakeholders. This artifact supports early identification of security or risk concerns related to external stakeholder interactions with the enterprise.
Definition	<p>A Context model depicts the organization as a single entity surrounded by key stakeholders. It establishes the focus and framing for design conversations and encourages discussion on the current state, identifying opportunities with respect to the scope and focus for the future state.</p> <p>Typically a context model includes a starter set of detailed interactions (message flows) paired as a request/response between the organization and external stakeholder groups at a high level. This implies that there is functionality within the organization to complete the interactions. Variations of the interactions are further detailed out within a Business Operating Model and will have corresponding business process flows.</p> <p>Stakeholder groups identified on the context model can be categorized as:</p> <ul style="list-style-type: none"> • Governors / Regulators: Defines and mandates the rules and regulations by which the organization is to operate within. Typically illustrated above the organization. • Clients / Customers: Individual and/or organizations who are currently using or will use the services provided by the organization. Typically illustrated to the right of the organization. • Business Partners: External partners who support the program and/or services delivered by the organization. Accountabilities and responsibilities of partners are defined within the partnership agreement. Typically illustrated at the bottom of the organization. • Suppliers / Contractors: Procured to provide goods and services enabling the organization to deliver a program and/or service. Procurement arrangements with suppliers are formalized by contracts which

Context model

define the accountabilities of each party and are up-held as part of the service level agreement. Typically illustrated to the left of the organization.

A stakeholder analysis is developed as part of the context model. The stakeholder analysis will further describe the interactions and the actors identified within their respective stakeholder groups. The context model will also aid in the quantification of:

- # of current state interactions and # currently automated
- Required # of target state interactions
- # of current and target state clients
- # of current and target state interactions per period, and
- # of current state interactions that will be eliminated (if any)

Appendix C.7: Control Artifact

Control	
Name	Control
Alias(es)	Security Control
Objective	The Control artifact is a subset of the System Security Plan Workbook. It identifies security and privacy controls as defined in MARS-E and other standards including IRS and NIST.
Definition	<p>Controls and safeguards are defined in sources such as IRS and MARS-E as protective measures taken to address vulnerabilities in the system that can be exploited by hackers. MARS-E Volume III: Catalog of Minimum Acceptable Risk Security and Privacy Control defines the different types of privacy and security controls with which MMP projects must comply. These controls are safeguards in response to security risk and privacy concerns standard designs to meet security requirements (functional and non-functional) in the solution requirements. The controls are also meant to look at safeguards.</p> <p>The MARS-E controls are just a minimum, and as such additional controls should be identified and applied through other standards and resources (e.g. IRS, NIST), as well as through any non-standards as the need arises.</p> <p>Security and privacy controls fall into one of three classes – management, operational, and technical. These classes are divided into families (e.g. access controls, awareness and training, security assessment and authorization, etc.) and are based on security functionality.</p> <p>The security control assessment determines are implemented correctly and operating as intended, and is meant to ensure and verify the controls are meeting stated security goals and objectives. An assessment can be carried out at various stages of the life cycle and is meant to increase confidence that the controls are effective in their application by continuous testing and improvement.</p> <p>The control specification will describe in detail:</p> <ul style="list-style-type: none"> • Control Family and Name: the primary family which determine the focus of the control, as well as the

Control				
	<p>control name. Located in header of specification</p> <ul style="list-style-type: none"> • Control Description: the description of the capability needed to protect the system • Control Object Type: the type of object that is the focus of the control (e.g. procedures, interfaces, application etc) • Implementation Standards: standards and guidance for the implementation of the control • Guidance: additional information on the intent of the control • Common, System-Specific or Hybrid Control: Whether it is a common control, system-specific control, or hybrid control • Control Owner: Where applicable, indicate the owner of the control • References: identifies source document (standards body, Authoritative Controls Document) • Related Controls: Set of additional controls that may cause conflict • Related Information Systems: IS system(s) which use this control • Assessment Objectives: Assessment requirements used to test and ensure that the control works • Potential System Impact: Any impact the control could have on the system • Compensating Controls: Indicate where control departs from standard and rationale (could be higher or lower than standard) • Risk: the risk(s) mitigated by the control 			
Sample	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="background-color: #003366; color: white; text-align: center; padding: 5px;">AC-2: Account Management</td> </tr> <tr> <td style="padding: 5px;">Control:</td> </tr> <tr> <td style="padding: 5px;">The organization manages information system accounts, including:</td> </tr> </table>	AC-2: Account Management	Control:	The organization manages information system accounts, including:
AC-2: Account Management				
Control:				
The organization manages information system accounts, including:				

Control					
	<p>a. Identifying account types (E.g., individual, group, system, application, guest/anonymous, and temporary);</p> <p>b. Establishing conditions for group membership;</p> <p>c. Identifying authorized users of the information system and specifying access privileges;</p> <p>d. Requiring appropriate approvals for requests to establish accounts;</p> <p>e. Establishing, activating, modifying, disabling, and removing accounts;</p> <p>f. Specifically authorizing and monitoring the use of guest/anonymous and temporary accounts;</p> <p>g. Notifying account managers when temporary accounts are no longer required and when information system users are terminated, transferred, or information system usage or need-to-know/need-to-share changes;</p> <p>h. Deactivating: (i) temporary accounts that are no longer required; and (ii) accounts of terminated or transferred users;</p> <p>i. Granting access to the system based on: (i) a valid access authorization; (ii) intended system usage; and (iii) other attributes as required by the organization or associated missions/business functions; and</p> <p>j. Reviewing accounts using the frequency specified in Implementation Standard 1.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Common, Hybrid, or System-Specific Control:</td> <td>Common Control</td> </tr> <tr> <td>Control Owner:</td> <td>N/A</td> </tr> </table> <p>Implementation Standards:</p>	Common, Hybrid, or System-Specific Control:	Common Control	Control Owner:	N/A
Common, Hybrid, or System-Specific Control:	Common Control				
Control Owner:	N/A				

Control				
	<ol style="list-style-type: none"> 1. Review information system accounts within every one-hundred-eighty (180) days and require annual certification. 2. Remove or disable default user accounts. Rename active default accounts. 3. Implement centralized control of user access administrator functions. 4. Regulate the access provided to contractors and define security requirements for contractors. 			
	Guidance:			
	<p>The identification of authorized users of the Exchange information system and the specification of access privileges is consistent with the requirements in other security controls in the Security Plan. Users requiring administrative privileges on information system accounts receive additional scrutiny by organizational officials responsible for approving such accounts and privileged access.</p>			
	Control Object Type:	Application	References:	HIPAA: 164.308(a)(3)(ii)(B), 164.308(a)(4)(i), 164.308(a)(4)(ii)(B); IRS-1075: 9.2
	Related Controls:	N/A	Related Information Systems:	HRM Finance Management System
Assessment Objectives:				

Control

Determine if:

the organization manages information system accounts, including;

- identifying account types (e.g., individual, group, system, application, guest/anonymous, and temporary);**
- establishing conditions for group membership;**
- identifying authorized users of the information system and specifying access privileges;**
- requiring appropriate approvals for requests to establish accounts;**
- establishing, activating, modifying, disabling, and removing accounts;**
- specifically authorizing and monitoring the use of guest/anonymous and temporary accounts;**
- notifying account managers when temporary accounts are no longer required and when information system users are terminated, transferred, or information system usage or need-to-know/need-to-share changes;**
- deactivating: (a) temporary accounts that are no longer required; and (b) accounts of terminated or transferred users;**
- granting access to the system based on: (a) a valid access authorization; (b) intended system usage; and (c) other attributes as required by the organization or associated missions/business functions;**

the organization reviews information system accounts in accordance with the frequency specified in Implementation Standard 1.

the organization meets all the requirements specified in the applicable implementation standard(s).

Control	
	Potential System Impact:
	Extra step needed to access system
	Any Deviations from Standard Controls: N/A
	Risk: Unauthorized Access

Appendix C.8: Data Flow Diagram

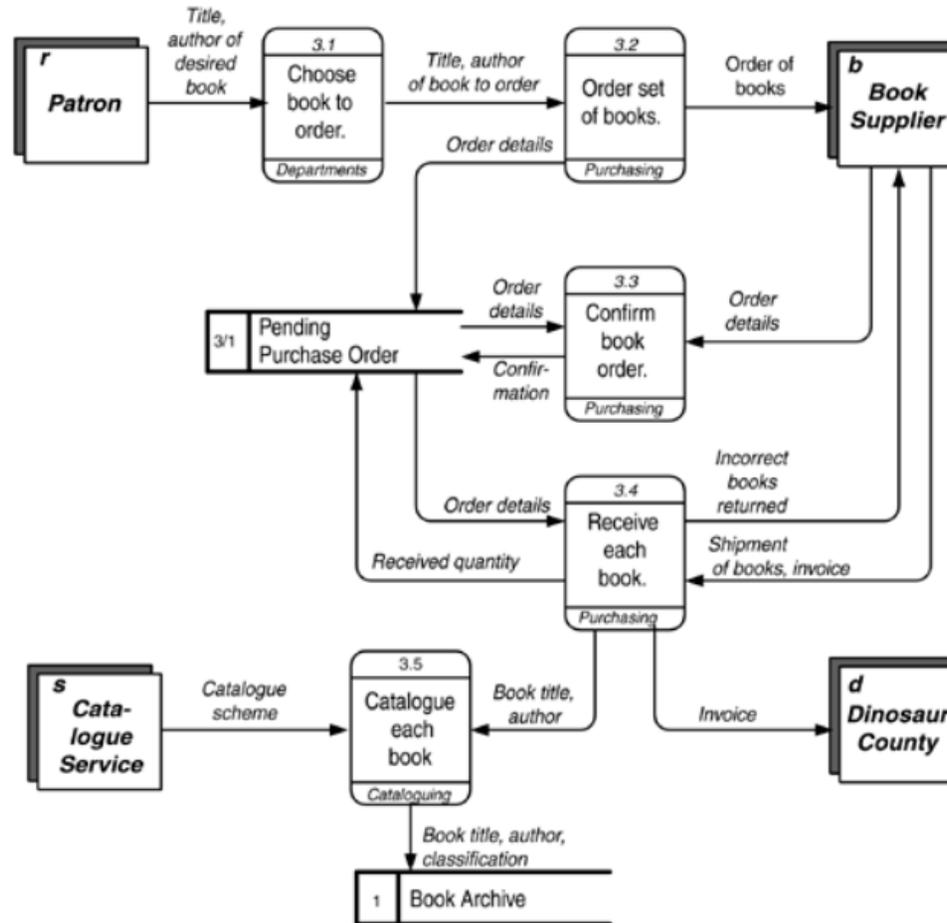
Data Flow Diagram	
Name	Data Flow Diagram
Alias(es)	DFD
Objective	The Data Flow Diagram (DFD) is used to model, trace, and analyze the flow of data through an information system.
Definition	<p>The Data Flow Diagram (DFD) shows, in varying levels of detail, what data is gathered from which processes, which processes data flows into, and what the process does or what process is carrying it out. It also shows where the data is stored. The DFD gives a functional view of a system, rather than how data is related - a Logical Data Model defines relationships between data entities.</p> <p>DFDs can be organized in a hierarchical manner, where models at a lower level of detail cascade from higher level models. The top level diagram - a context diagram – will show a high-level overview of the system, and below that the DFDs will focus on a smaller portion of the context model. It is at the discretion of the modeler as to whether DFDs should be organized in a hierarchical manner, or if these diagrams all exist as peers to each other.</p> <p>There are four symbols that are used in the DFD:</p> <ul style="list-style-type: none"> • Squares: represent the external entities – the group of stakeholders outside the control of the modelled system, and show where the data comes from and where it’s going. The DFD will not include any relationships between the external entities. If the relationship needs to be modelled, it will be added as a process to the DFD and not shown as an external entity. • Rounded rectangles: represent the processes within the system and shows how data is transformed (via data inputs and outputs to the data store). The process name is generally used to describe what the process does but on occasion can show the name of who or what is carrying out the process. • Arrows: represents the flow of data from one process to another. The direction of the arrow shows which

Data Flow Diagram

processes are the inputs and outputs.

- **Open ended rectangles:** represents the data stores.

Sample



Appendix C.9: Deployment Model

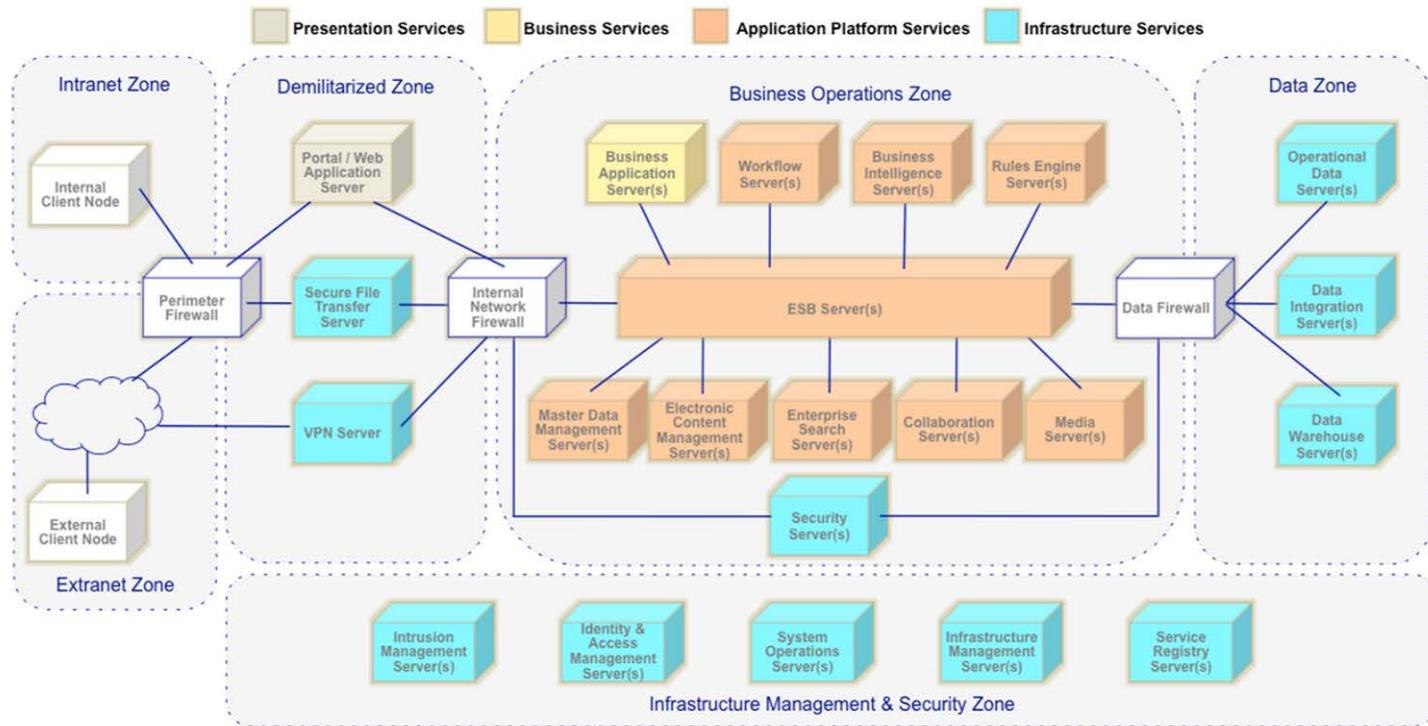
Deployment Model	
Name	Deployment Model
Alias(es)	Logical Technology Deployment Architecture
Objective	A deployment model is a technology/security model that describes a logical technology deployment architecture in the form of deployment nodes arranged into security zones. It illustrates how information systems will be implemented. This diagram will be used to define a System Security Boundary Model.
Definition	<p>A deployment model provides a blueprint for IT infrastructure planning and design, addressing relevant security concerns. It is used to describe the secure implementation of transformed systems and ensure compliance related to privacy, data protection and other concerns.</p> <p>A deployment model contains the following concepts:</p> <p>Nodes</p> <p>Nodes are the deployment targets that represent computational resources upon which artifacts (e.g. a software service) may be installed for execution. At the physical level, a device is a (hardware) node which represents a physical computational resource with processing capability upon which artifacts may be deployed for execution. An execution environment is a (software) node that offers a physical execution environment for deploying specific types of executable artifacts. Physical level nodes can be nested; for example, a device contains an operating system and an operating system may run an execution platform.</p> <p>Communication Paths</p> <p>Nodes are logically associated in a network through communication paths. Communication paths interconnect nodes for the purposes of exchanging signals and messages through network protocols (e.g. HTTP). A message is a structured set of data that follows a prescribed method of delivery, and the delivery method is called the message protocol (e.g. Web service standards). The message protocol results in a completely self-contained information exchange.</p>

Deployment Model

Security Zones

A security zone is a logical area within an interconnected network environment with a well-defined communication flow to other zones. Zones implement security controls to manage well defined communication flow between them. Security zones within a network are an effective strategy for reducing many types of risk. Blended networks without clear boundaries increase the need for a clear security strategy implemented through security zones that protect valuable data assets

Sample



Appendix C.10: Functional Requirement

Functional Requirement	
Name	Functional Requirement
Alias(es)	Solution Requirement
Objective	The objective of a functional requirement is to identify a condition or capability needed to solve a problem or achieve an objective.
Definition	<p>Requirements are defined by the IEEE³ as:</p> <ul style="list-style-type: none"> • A condition or capability needed by a user to solve a problem or achieve an objective. • A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. • A documented representation of a condition or quality as in 1 or 2. <p>A functional requirement describes the behavior that a solution must possess. Non-functional requirements describe constraints that a solution must operate within or quality attributes that a solution must possess.</p> <p>Functional requirements are typically written in the format of ‘the system shall’ or ‘the solution shall’ and are closely related to business rules, interface requirements, and data requirements – together with the non-functional requirements, these are the full description of what a solution must do.</p>

³ IEEE Standard Glossary of Software Engineering Terminology (1990)

Functional Requirement

Functional requirements do not contain any statements of design or implementation details, testing info or project planning information.

Good requirements are:

- **Complete** – Fully describes the functionality to be delivered.
- **Consistent** – Requirements should not conflict with other requirements of the same type or with higher-level business, system, or user requirements.
- **Correct** – Accurately describes the functionality to be built.
- **Feasible** – Can be implemented within the known capacities and limitations of the system and its operating environment.
- **Necessary** – Documents a capability customers need or one required to comply with an external system requirement or a standard.
- **Prioritized** – Is assigned an implementation priority (in the case of each functional requirement, feature, or use case) to indicate how essential it is.
- **Unambiguous** – Enables all readers of a requirement statement to arrive at a single, consistent interpretation of it.
- **Verifiable** – Indicates through tests or other verification approaches, such as inspection or demonstration, that it implements the required action properly.
- **Traceable** – Can be linked backward to its origin and forward to the design elements and source code that implement it as well as to the test cases that verify it was implemented correctly.
- **Testable** – the requirements must be able to be tested in the solution.

Typically, each requirement will have the following attributes:

- Requirement ID: Unique reference that is not altered or reused if the requirement is changed
- Name: Title

Functional Requirement

- Type: Identifies what type of requirement (e.g. business, stakeholder, functional, non-functional, transition)
- Description: Gives a detailed description of the requirement – the ‘shall’ statement
- Version: Requirement increment
- Source: Identifies the origin of the requirement. The source is often consulted if the requirement changes or if more information regarding the requirement or the need that drove the requirement has to be obtained
- Author: Provides the name of the person who needs to be consulted should the requirement later be found to be ambiguous, unclear, or in conflict
- Owner: Indicates the individual or group that needs the requirement or the business owner after the solution is implemented
- Traceability: Indicates any other requirement dependencies. Traceability can be between the same requirement types in a dependency chain, or can be a hierarchy of grouped requirements, or can be a decomposition of requirement types from business requirements -> stakeholder requirements -> solution requirements.
- Complexity: Indicates how difficult the requirement will be to implement. This will assist in estimating how long a development effort is required, or the release to which it will be allocated. Complexity is measured as:
 1. This requirement is not very complex.
 - a) The requirement does not require a large amount of effort to implement.
 - b) This requirement does not require input from a Subject Matter Expert to implement.
 - c) This requirement does not require integration with outside systems or interfaces.
 - d) This requirement does not require a large amount of change to the implementation or organization.
 - e) This requirement does not have a large security impact or risk profile.
 2. This requirement is somewhat complex.

Functional Requirement

- a) This requirement requires more than a small amount of effort to implement.
 - b) This requirement requires consultation with a Subject Matter Expert during implementation.
 - c) This requirement requires implementation with outside systems and services that are already in a steady operational state.
 - d) This requirement will require minor changes to the organization.
 - e) This requirement has some security impact and risk profile, but it is thoroughly managed.
3. This requirement is very complex.
- a) This requirement requires a large amount of effort to implement.
 - b) This requirement requires a Subject Matter Expert to implement.
 - c) This requirement has a major security impact or risk profile, and without proper management can cause major impact to the project or organization.
 - d) This requirement requires implementation with outside systems and services that are not yet in a steady operational state.
- Status: Indicates the state of the requirement
 - Priority: Indicates relative importance. Priority can refer to the relative value of a requirement or to the sequence in which it will be implemented.
1. High
- a) Implementing this requirement is imperative for the overall solution.
 - b) This requirement must be implemented along with the solution, and if it is not, it will be a major block to the solution.
 - c) This is a high risk requirement and should be addressed sooner rather than later.
 - d) This requirement's completion is a dependency on other high priority requirements.
 - e) This requirement has a set timeline to implement.

Functional Requirement	
	<p>f) This requirement requires additional staff focus to implement.</p> <p>2. Medium</p> <p>a) Implementing this requirement is needed for the overall solution.</p> <p>b) This requirement needs to be implemented along with the solution, but will not stop the overall solution if the requirement needs to be adjusted.</p> <p>c) This requirement has some risk, but the risk can be easily managed or mitigated in order to successfully implement the requirement.</p> <p>d) This requirement's implementation does not have dependencies on other high priority requirements. Any dependencies on medium requirements can be mitigated with workarounds if this requirement is not implemented.</p> <p>e) This requirement has a set timeline, but that timeline can be adjusted without introducing risk to the overall solution.</p> <p>3. Low</p> <p>a) Implementing this requirement is helpful in the overall solution, but is not pivotal.</p> <p>b) This requirement can be implemented at a later time without major impact to the overall solution.</p> <p>c) This requirement has little to no risk associated with its implementation.</p> <p>d) This requirement's implementation has no dependencies on other high or medium priority requirements.</p> <p>e) This requirement does not have a set timeline.</p> <ul style="list-style-type: none"> • Urgency: Indicates how soon the requirement is needed. It is usually only necessary to specify this separately from the priority when a deadline exists for implementation • Release: Indicates the release to which the requirement is allocated
Sample	<ul style="list-style-type: none"> • ID: FR-001

Functional Requirement

- Name: Automatic eligibility determination
- Type: Functional Requirement
- Description: The solution shall automatically validate if an eligibility application meets the eligibility criteria.
- Version: 1.0
- Source: HCFA Enterprise Architect
- Author: Jane Doe
- Owner: John Doe
- Traceability: Parent of requirements FR-010, FR-020, dependent on FR-101
- Complexity: Very Complex
- Status: Verified
- Priority: High
- Urgency: Nov 1, 2016
- Release: Allocated to Release V1.1

Appendix C.11: Logical Data Model & Dictionary

Logical Data Model and Dictionary	
Name	Logical Data Model and Dictionary
Alias	Entity Relationship Diagram (ERD)
Objective	<p>The objective of a logical data model is to 1) communicate data requirements between people of different levels of expertise 2) analyze the characteristics and data requirements for a subject or 3) help explain the data context and scope of a purchased application. The model and accompanying data dictionary describes the data that is required. Logical models are not necessarily used to prescribe physical database design; these types of models can be used for this purpose, but in the context of MMP, the logical data model will not be used to implement a DBMS, but will instead be used to perform gap analyses between logical business requirements and a solution.</p>
Definition	<p>For simplicity, this specification assumes that data requirements are going to be implemented into relational databases, as opposed to hierarchical, object oriented or associative databases, or flat files. This specification also assumes that the data model is being used to develop an operational application system optimized for transaction processing, as opposed to a data warehouse dimensional model, which requires a different approach to organizing data to optimize queries.</p> <p>A logical data model is a translation of a conceptual data model into structures that can be implemented using a database management system (DBMS) or met with a package solution. For a relational DBMS, this means that the logical model specifies entities and attributes. Logical data models are technology neutral and do not include database technology specific constraints, which are introduced with the physical data model.</p> <p>Common data model notations include UML Class diagrams or Entity Relationship (ER) diagrams with relationships drawn as 'crow's feet', more formally known as Information Engineering (IE) syntax by James Martin. The examples here are demonstrated using IE notation, but either modelling notation is acceptable.</p> <p>Best practice for logical data models includes but is not limited to:</p> <ol style="list-style-type: none"> 1. Singular entity names (nouns only)

Logical Data Model and Dictionary

2. Key and non-key attributes are included on the model
3. All relationships are named using verb phrases, with defined optionality and cardinality
4. All many-to-many relationships are resolved
5. Primary and foreign keys are identified
6. One fact per column or attribute
7. Attribute names include attribute types as the last part of the name (e.g. Appeal Unique Identifier, Appeal Type Code, Appeal Type Description, Appeal Submitted Date, Appeal Status Code, Appellant Name, Contact Number)
8. Generalized or abstract data models are preferred over specialized data models, as they will be more stable in the long term. However, the more generalized a data model is, the more difficult it may be for business users to understand the meaning. Finding a balance between the two is a soft skill that depends on the situation.
9. Developing the data model to at least a third-normal form. Normalization is the process for organizing data into tables in such a way that reduces redundancy and incompleteness. An entity is in third normal form if all of the non-key data depends on no other data element outside the key ('the key, the whole key and nothing by the key').
10. Reuse patterns where appropriate, reuse of industry standard data model patterns for a subject area are a highly effective method to jump start a modelling effort.
11. Use subtypes or super-types to generalize entities that share the same attributes; for example, a Party super-type contains the subtypes of Person and Organization.
12. Ensure that all referential integrity rules are modelled as relationships. For example, a Customer can exist without an Order, but an Order cannot exist without a Customer.
13. Define primary keys as surrogate keys – that is, use a meaningless generated unique identifier. Business keys are not stable over time and should not be used as primary keys.

Logical Data Model and Dictionary

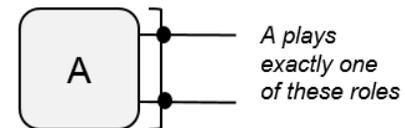
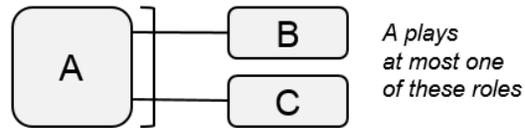
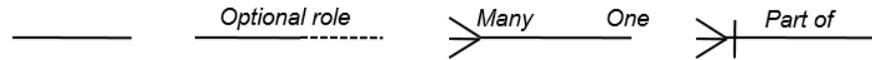
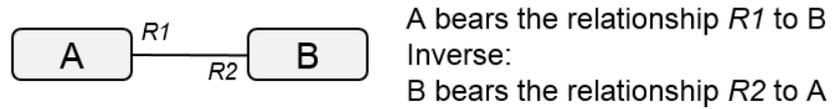
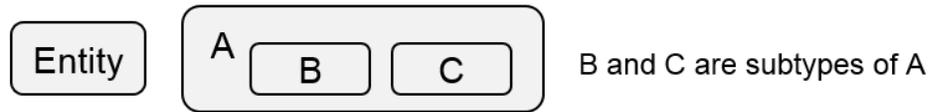
A **data dictionary** provides definitions for the entities and attributes shown on the data model. A data dictionary should have at a minimum the following metadata:

1. **Entity Name:** Unique noun for a business entity of interest to the organization, an object or an event which follows the rules of normalization and abstraction. Do not use acronyms or abbreviations in the entity or table name.
2. **Entity Definition:** Definitions should be a concise description of the entity that stands alone without context; that is, the entity definition is sufficient to understand the meaning of the entity without specialized knowledge of the business area.
3. **Entity Security Level.** Entities should be tagged with information on information sensitivity and classification as informed by an information security assessment.
4. **Attributes**
 - a) **Primary Key Indicator:** Flag indicating if the attribute is a primary key
 - b) **Business Key Indicator:** Flag indicating if the attribute is a part of the business key
 - c) **Attribute Name:** An attribute is a property of an entity whose values help identity or describe an entity instance. In a logical data model, attributes should be spelled out in full and are atomic; that is, they contain one and only one fact or piece of data that cannot be divided into smaller pieces. For example, the attribute phone number can be divided into components such as country code, area code, number and extension.
 - d) **Attribute Description:** Similar to entity description; concise and can stand alone without context.
 - e) **Attribute Privacy Determination:** Attribute that records whether associated data contains Private Information (PI) or Private Health Information (PHI).

The notation used in the sample below is Oracle/Barker notation:

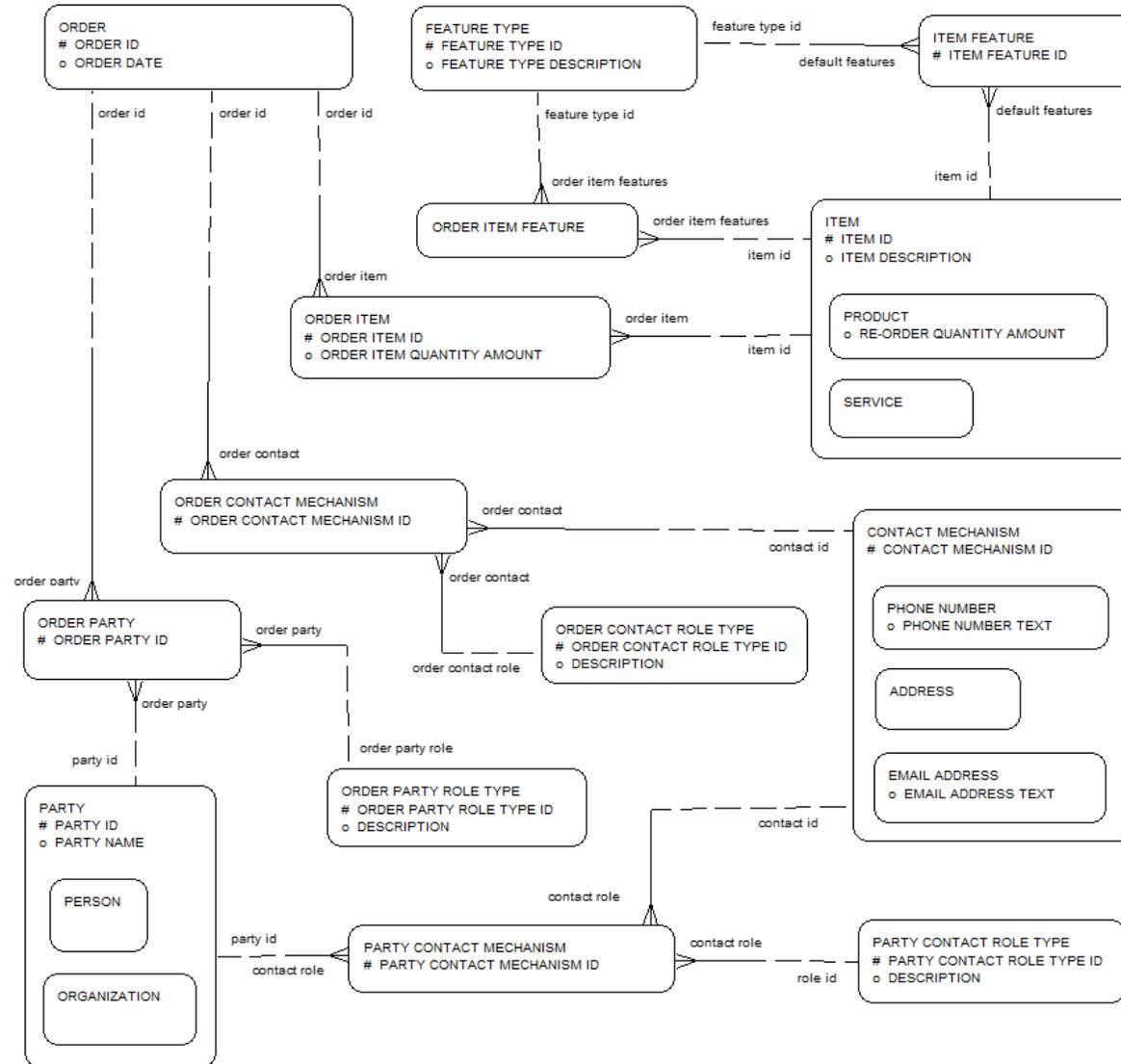
Logical Data Model and Dictionary

Components (Oracle / Barker notation):



Logical Data Model and Dictionary

Sample



Appendix C.12: Non-Functional Requirement

Non-Functional Requirement	
Name	Non-Functional Requirement
Alias(es)	NFR, Technical Requirement
Objective	The objective of a non-functional requirement is to identify a constraint within which a solution must operate, or a quality attribute that a solution must possess.
Definition	<p>Requirements are defined by the IEEE⁴ as:</p> <ol style="list-style-type: none"> 1. A condition or capability needed by a user to solve a problem or achieve an objective. 2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. 3. A documented representation of a condition or quality as in 1 or 2. <p>A functional requirement describes the behavior that a solution must possess. Non-functional requirements describe constraints that a solution must operate within, or quality attributes which the solution must possess.</p> <p>Non-functional requirements are typically written in the format of 'the system shall be' or 'the system shall' and together with the functional requirements, are the full description of what a solution must do.</p> <p>Similar to functional requirements, non-functional requirements do not contain any statements of design or</p>

⁴ IEEE Standard Glossary of Software Engineering Terminology (1990)

Non-Functional Requirement

implementation details, testing info or project planning information.

Typically, each requirement will have the following attributes:

- Requirement ID: Unique reference that is not altered or reused if the requirement is changed
- Name: Title
- Type: Identifies the requirement as a non-functional requirements, with the following allowed subtypes:
 - **Privacy** - These requirements identify the level of privacy required for the system and depend on the type of data being accessed, sent or received and who is accessing this information. The type of data is determined in the Logical Data Model. To determine the level of privacy required, refer to the Information Security standards.
 - **Security** - These requirements are informed by security risks, controls and safeguards.
 - **Usability** - These requirements specify how easy the system must be to use. To define these requirements, refer to functional requirements and define how that functionality should be perceived by the user.
 - **User Interface** - User interface requirements specify elements such as considerations for screen design, field placement, aesthetics, color choice, etc.
 - **System Interfaces** - These requirements define how the solution will communicate through protocols with other systems (internal or external). An example of an interface specification is located in the Interface Control Specification artifact, as interface requirements have more attributes that must be defined than the normal non-functional requirements.
 - **Accessibility** - These requirements identify where the users are accessing the system (e.g, VPN), through what channel (e.g., web portal, mobile device) and when they require access (e.g., 24/7).
 - **Business Environment** - These requirements identify what environment the system must be able to operate in or what other hardware or software components the system is compatible with (e.g., “Software contractors must provide an approved scanning device compatibility list.”).
 - **Information Security** - These requirements are determined by understanding the sources and target applications that are sending and receiving information, what information is being transmitted and

Non-Functional Requirement

where the systems are located. Information security requirements should enable the solution to adhere to internal and external information security standards.

- **User Access** - These requirements depend on which users and user groups are accessing the system. Refer to the User Group section of this document. Based on the defined user roles, the levels of accessibility required by the system can be identified and listed in this section.
- **Performance/Capacity** - These requirements set performance expectations of the solution and identify how much capacity the solution is expected to handle. Overall service level expectations should be defined in the Service Level Agreement.
- **Business Continuity/Recoverability** - This refers to requirements related to how fast the solution must resume functionality in the case of an emergency in which any or all aspects/functions of the system are down. This also refers to requirements related to recovering lost data.
- **Logging/Monitoring** - This refers to requirements related to the logging and monitoring of all system activities along with the time frame for detection of unauthorized activities and the associated response time.
- **Archive/Retention** - These requirements address how historic data is stored and how long it must be stored for. Refer to applicable internal and external data storage and protection standards. Refer to internal and external audit standards when defining these requirements.
- **Expected Life Span** - This requirement identifies how long the system must be fully functioning and sustainable.
- **Documentation/Training/Help** - This refers to requirements related to the contractor providing product documentation, training and ongoing help/support (e.g.. type of documentation and training, duration of support, response time etc.)
- **Communication** - This refers to operational and maintenance requirements related to how the contractor can be contacted, when they can be contacted, though which channels etc.
- **Legislative and Regulatory/Compliance** - These requirements are specific to addressing standards set by State and Federal Legislative and Regulatory Bodies (e.g., privacy standards, safety standards, information protection standards etc.). This section may duplicate business rule statements that cover

Non-Functional Requirement

legislative and policy requirements.

- **Integration** - These requirements identify the type of integration with which the solution must be compatible. Refer to the Conceptual Integration Architecture.
 - **Data Migration** - These requirements identify what data has to be migrated to the target. This is intended to identify the data that will require migration to the solution, but not the migration mapping. This is completed once the target data model is known.
 - **Deployment** - This refers to requirements related the infrastructure environment in which the system must be deployed (e.g., network requirements, hardware requirements, database requirements, licensing requirements etc.)
- Description: Gives a detailed description of the requirement – the ‘shall’ statement
 - Version: Requirement increment
 - Source: Identifies the origin of the requirement. The source is often consulted if the requirement changes or if more information regarding the requirement or the need that drove the requirement has to be obtained
 - Author: Provides the name of the person who needs to be consulted should the requirement later be found to be ambiguous, unclear, or in conflict
 - Owner: Indicates the individual or group that needs the requirement or the business owner after the solution is implemented
 - Traceability: Indicates any other requirement dependencies. Traceability can be between the same requirement types in a dependency chain, or can be a hierarchy of grouped requirements, or can be a decomposition of requirement types from business requirements -> stakeholder requirements -> solution requirements.
 - Complexity: Indicates how difficult the requirement will be to implement. This will assist in estimating how long a development effort is required, or the release to which it will be allocated. Complexity is measured as:
 1. This requirement is not very complex.

Non-Functional Requirement

- a) The requirement does not require a large amount of effort to implement.
 - b) This requirement does not require input from a Subject Matter Expert to implement.
 - c) This requirement does not require integration with outside systems or interfaces.
 - d) This requirement does not require a large amount of change to the implementation or organization.
 - e) This requirement does not have a large security impact or risk profile.
2. This requirement is somewhat complex.
 - a) This requirement requires more than a small amount of effort to implement.
 - b) This requirement requires consultation with a Subject Matter Expert during implementation.
 - c) This requirement requires implementation with outside systems and services that are already in a steady operational state.
 - d) This requirement will require minor changes to the organization.
 - e) This requirement has some security impact and risk profile, but it is thoroughly managed.
 3. This requirement is very complex.
 - a) This requirement requires a large amount of effort to implement.
 - b) This requirement requires a Subject Matter Expert to implement.
 - c) This requirement has a major security impact or risk profile, and without proper management can cause major impact to the project or organization.
 - d) This requirement requires implementation with outside systems and services that are not yet in a steady operational state.
- Status: Indicates the state of the requirement
 - Priority: Indicates relative importance. Priority can refer to the relative value of a requirement or to the sequence in which it will be implemented.

Non-Functional Requirement

1. High
 - a) Implementing this requirement is imperative for the overall solution.
 - b) This requirement must be implemented along with the solution, and if it is not, it will be a major block to the solution.
 - c) This is a high risk requirement and should be addressed sooner rather than later.
 - d) This requirement's completion is a dependency on other high priority requirements.
 - e) This requirement has a set timeline to implement.
 - f) This requirement requires additional staff focus to implement.
2. Medium
 - a) Implementing this requirement is needed for the overall solution.
 - b) This requirement needs to be implemented along with the solution, but will not stop the overall solution if the requirement needs to be adjusted.
 - c) This requirement has some risk, but the risk can be easily managed or mitigated in order to successfully implement the requirement.
 - d) This requirement's implementation does not have dependencies on other high priority requirements. Any dependencies on medium requirements can be mitigated with workarounds if this requirement is not implemented.
 - e) This requirement has a set timeline, but that timeline can be adjusted without introducing risk to the overall solution.
3. Low
 - a) Implementing this requirement is helpful in the overall solution, but is not pivotal.
 - b) This requirement can be implemented at a later time without major impact to the overall solution.
 - c) This requirement has little to no risk associated with its implementation.
 - d) This requirement's implementation has no dependencies on other high or medium priority requirements.

Non-Functional Requirement	
	<p>e) This requirement does not have a set timeline.</p> <ul style="list-style-type: none"> • Urgency: Indicates how soon the requirement is needed. It is usually only necessary to specify this separately from the priority when a deadline exists for implementation • Release: Indicates the release to which the requirement is allocated
Non-Functional Sample	<p>The following examples elaborate the description of a set of non-functional requirements:</p> <ol style="list-style-type: none"> 1 The system must process up to 1000 applications per minute 2 The system must have less than 1 hr of downtime per week 3 The system must be available during the hours of 5:00am and 12:00am local time 4 The system must be restored from failure within 6 hrs

Appendix C.13: Physical Data Model

Physical Data Model	
Name	Physical Data Model
Alias(es)	PDM
Objective	A physical data model defines how data will be physically designed and optimized for physical implementation. Physical data models also address issues such as audit and privacy requirements, data access and performance, and storage management.
Definition	<p>For simplicity, it is assumed that the specification will be used to model data that will be physicalized into a Relational Database Management System (RDBMS), and not an object-oriented, hierarchical or associative database. This specification also assumes that the physical data model is being used to develop an operational application system optimized for transactional processing, as opposed to a data warehouse dimensional model optimized for queries.</p> <p>A physical data model design shows the specific tables, columns and constraints involved in the physical implementation of data in into a database product. A physical data model is based upon a logical data model design. Organizations generally have physical database naming standards that must be followed when implementing a physical database. A physical data model is database technology specific, and is reflected in the model in such things as data types.</p> <p>Similar to logical data models, physical data models can be drawn in relational models with ‘crow’s feet’ or Information Engineering (IE) syntax by James Martin, or in UML Class Diagrams. For standards on drawing these types of models, see the Logical Data Model specification in this document.</p> <p>The physical data model design considers the following:</p> <ol style="list-style-type: none"> 1. The performance of record retrieval based on anticipated record volume and the caching, indexing, and clustering facilities available in the anticipated RDBMS. This may introduce objects which are required strictly for performance. 2. The mandated enterprise data security settings and any conflicts with the proposed business data

Physical Data Model

functional requirements. For example, whether data will be distributed to multiple platforms or outside of firewalls.

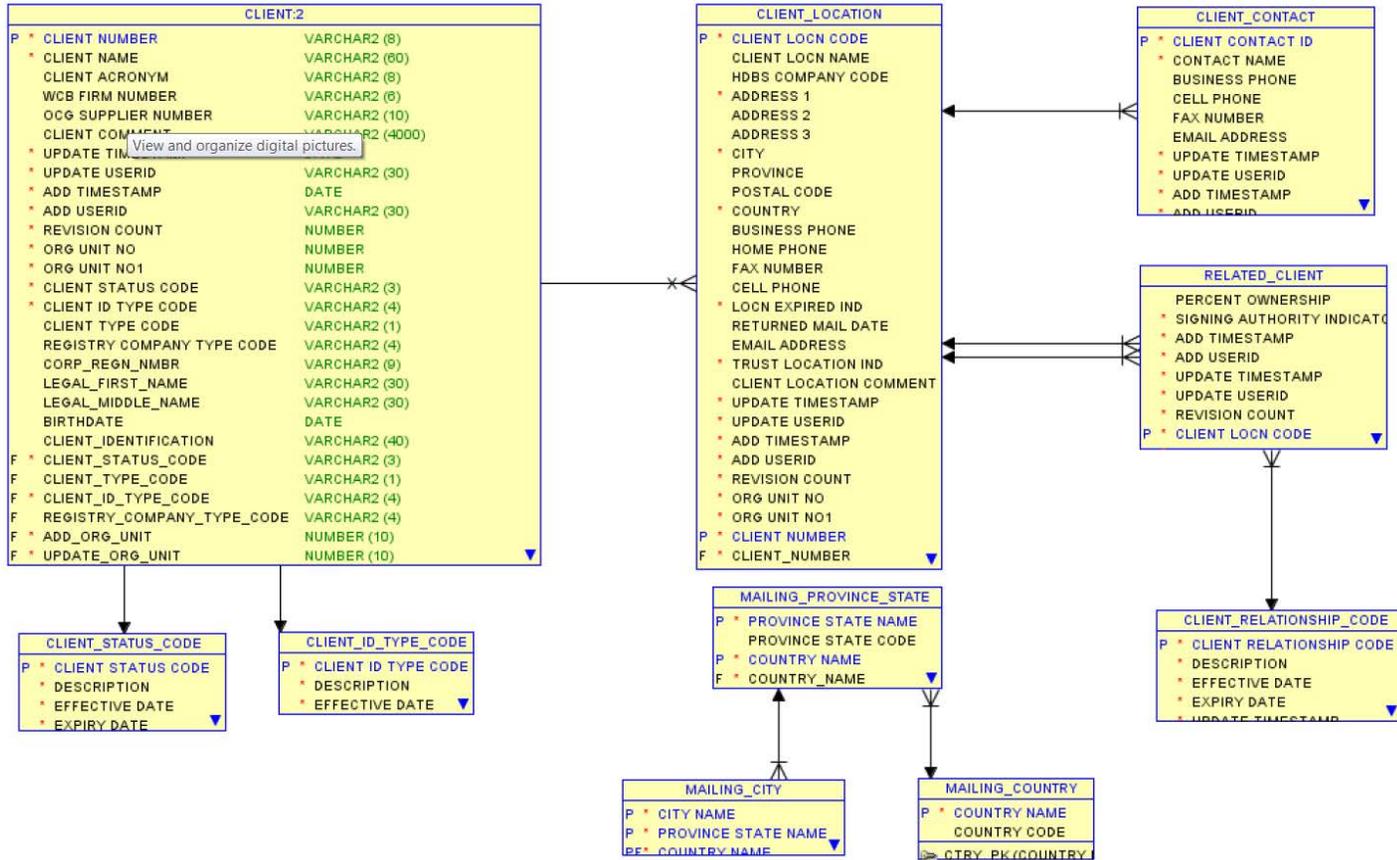
3. Physical design compromises such as de-normalization to roll down super-type attributes into subtype entities.
4. Surrogate or generated keys, business keys, primary and alternate indices, database triggers, stored procedures, default and allowed values, and constraints enforced within the RDBMS.
5. The Physical Data Model should be accompanied by a data dictionary and the DDL (data definition language) files. The data dictionary should include estimated physical file size, and annual table growth estimates.
6. Column order optimized for application development or according to organization standards.
7. Referential integrity constraints, (foreign key constraints), cascading deletes, and orphan dependencies.
8. Implementation of lookup or code tables, optimized for sharing and minimum redundancy.
9. Development of database views, to address performance or to meet data access and privacy standards (security).
10. Physical names should use the approved abbreviated terms and construction rules defined in CMS's guide DM OP-045 Operating Procedure for Constructing Physical Table and File Names or CMS's guide DM OP-046 Operating Procedure for Constructing Physical Column or Element Names.
11. Also refer to the CMS Standard Terms and Abbreviation List and the CMS Data Management Operating Procedures and Guidelines.
12. ISO 11179 is a global standard to provide guidelines for standardizing and registering data elements, and can be referenced when defining the elements that should be included in a physical data dictionary. The physical data dictionary should include, but is not limited to:
13. Table name

Physical Data Model

- i. Table Description
- ii. Unique Constraints
- iii. Foreign Keys
- iv. Indexes
- v. Table size and growth estimates
- vi. Notes on views
- vii. Security Classification
- viii. Column name
- ix. Sequence number
- x. Column Name
- xi. Null/not null rule
- xii. Data Type, Length & Precision
- xiii. Column Definition
- xiv. Domains
- xv. Allowed Values
- xvi. Default Values
- xvii. Edit Mask (e.g. date or area code)
- xviii. Security Classification or restriction on use

Physical Data Model

Sample



Legend: P = primary key, F = foreign key, *mandatory attribute

Appendix C.14: Requirements Traceability Matrix

Requirement Traceability Matrix	
Name	Requirement Traceability Matrix (RTM)
Alias(es)	RTM
Objective	The objective of a requirements traceability matrix (RTM) is to ensure completeness of solutions against business need. An RTM can be considered a report of requirements mapped against other design elements or other requirements.
Definition	<p>An RTM captures all business and technical requirements of a certain scope. It is used to manage individual requirements to design, solution or roadmap components to ensure that the blueprints are complete and reflect a legitimate design. Additionally, the RTM provides structure to track system development progress and quality.</p> <p>An RTM maps requirements for multiple purposes:</p> <ul style="list-style-type: none"> • To understand completeness against business rules, data elements, business processes, stakeholders, and architecture elements • Business & System Design • Fit/Gap analysis to support solution assessment • To set scope for procurement, contracting, testing, implementation releases • Review/Audit post implementation <p>Each row of an RTM is a view of a requirement. The actual requirements are often expressed as a statement that guide or constrain the business or system design.</p> <p>Depending on the objective of the RTM, different columns can be displayed to map to requirements. Likewise, the content of the matrix cell can vary depending on the relationship of the two elements to each other.</p>

Requirement Traceability Matrix											
Sample	Requirements Management Content			Business and System Component Traceability		Roadmap Traceability					
	Req. #	Requirement Description "The system shall..."	Status	Action Item	Assigned To	Business Component	Systems Component	In Scope for Release	In Scope for Release	In Scope for Release	
	1	Update Exchange Financial Management database with electronic Issuer payment data.	Identified	Change wording of requirement to further clarify	Jill	Fed APTCs and CSRs	TBD	TBD	TBD	TBD	TBD
	2	Provide capability to track and enforce premium payment timing guidelines and restrictions (SHOP)	Identified	Further specify to clarify harmonizing issues for employer grace periods for the Issuers	Mary	SHOP Premium Collection	Premium Processor	TBD	TBD	TBD	TBD
	3	Provide inquiry screens to identify the source of the discrepancy and make note of it electronically.	Identified	Determine where inquiry screens will reside	Bob	Employer Premium Discrepancy Resolution	Premium Processor	Y	TBD	TBD	TBD
4	Provide inquiry screens to identify the source of the discrepancy and make notes about what needs to be corrected.	Identified	Determine where inquiry screens will reside	Bob	Individual Premium Discrepancy Resolution	Premium Processor	Y	TBD	TBD	TBD	

Appendix C.15: Software Services Model

Software Services Model	
Name	Software Services Model
Alias(es)	Software Component Model, Application Component Model
Objective	The objective of a software services model is to provide a catalogue of software service (sometimes referred to as application components) in a layered or tiered view that a solution must contain to enable the business requirements and realize a client's future state vision.
Definition	<p>A software services models is structured in a tiered (or layered) manner and software services (or components) are grouped according to tiers (or layers).</p> <p>Each level within a layer in the model is a decomposition of a software service (or component) at a higher level (where relevant). Each layer of the model is enabled by the layer that follows described in the following definitions:</p> <p>Presentation Services Layer:</p> <ul style="list-style-type: none"> • This layer includes components for presenting information and services to people, IT services for people, and traditional (non-automated) interactions with people. • People may access human services information and services via electronic devices (computers, telephones, fax machines, or other). • People may also interact directly with other people to access information. <p>Business Services Layer:</p> <ul style="list-style-type: none"> • This layer includes software components/services that directly map to the business capabilities and reflect the software solutions to be provided to support business need. <p>Application Platform Services Layer:</p> <ul style="list-style-type: none"> • This layer has components that provide the actual technical functions that support various business needs

Software Services Model

addressed through the business services later.

- It provides components that provide reusable, and extendable common business support applications.

Infrastructure Services Layer:

- Components in the infrastructure layer offer various infrastructure services required to support the successful delivery of the technology solutions.
- Technology components in this layer offer broad technical services that are utilized by the application platform and business services layers to create a truly integrated, scalable and manageable architecture.
- This layer's components enable integration of the technical components in the enterprise architecture.

Sample



Appendix C.16: System Landscape Model

System Landscape Model	
Name	System Landscape Model
Alias(es)	System Model
Objective	The objective of the system landscape model is to provide a high-level representation of the system(s) required to enable business processes. It illustrates how the identified systems will interact to enable business operations.
Definition	<p>The system landscape model is a system blueprint identifying the key system messages and stakeholder systems in their current or future states. This model highlights changes that will be made to achieve the required system interoperability in the future state, and identify new or obsolete system messages, and required changes to stakeholder systems.</p> <p>The system landscape model may elaborate, measure, or further express functional, non-functional, data, policy, security and interface requirements.</p> <p>Due to its potential complexity, the system landscape model may be broken up into several views to support an overall scenario.</p> <p>The system landscape model assists with developing findings and recommendations for the future state. Some examples include:</p> <ul style="list-style-type: none"> • Quantification <ul style="list-style-type: none"> ○ # of systems involved in current and target state delivery ○ Volume of current and target state internal system interactions • Findings <ul style="list-style-type: none"> ○ # of systems involved in current and target state delivery ○ Volume of current and target state internal system interactions

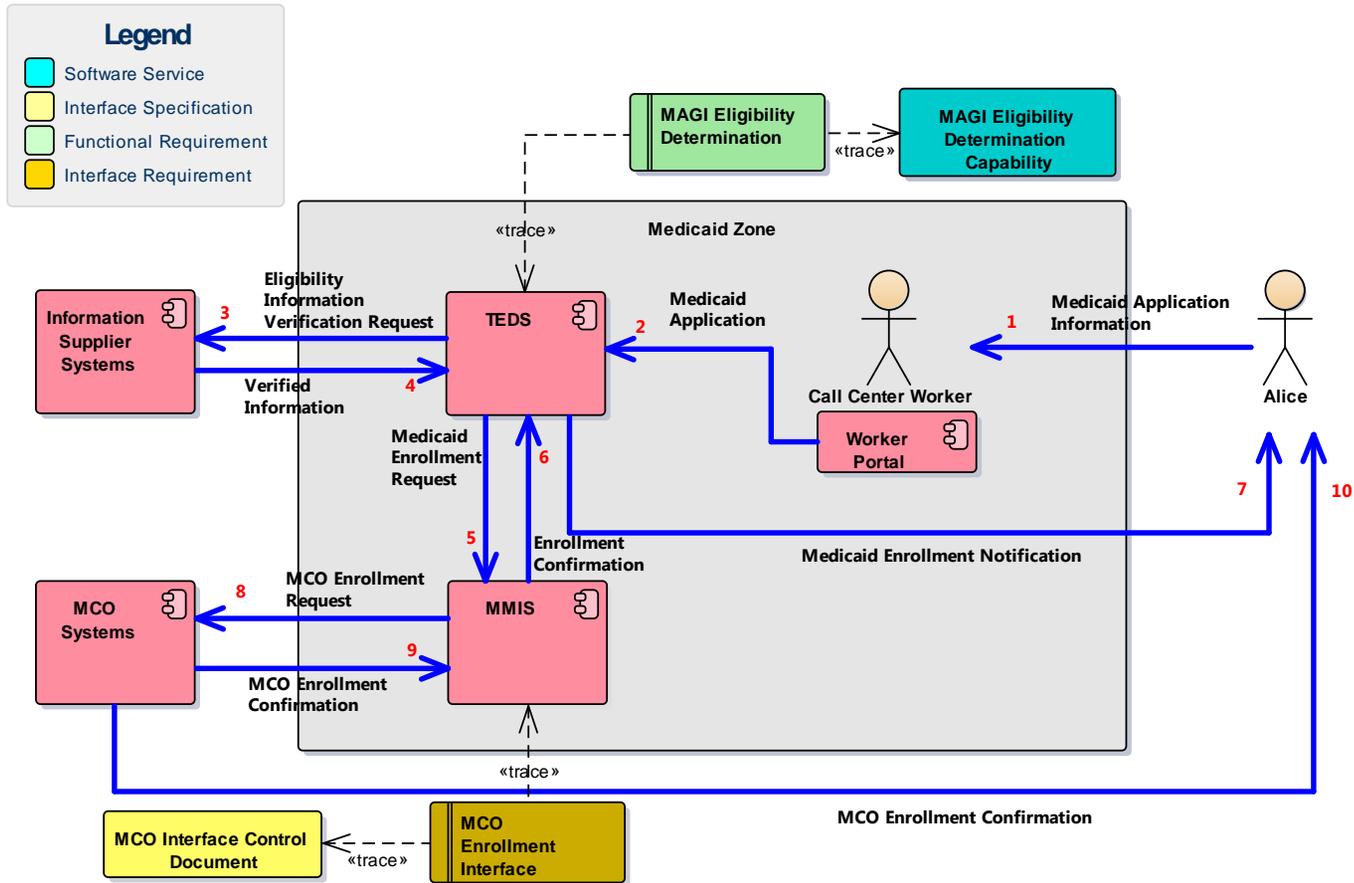
System Landscape Model

- Identification of flows that have privacy or security implications to inform risk assessment
- Identification of user interactions with functions that have security implications to inform risk assessment
- Identification of system interactions that have security implications to inform risk assessment
- Recommendations
 - # of systems involved in current and target state delivery
 - Volume of current and target state internal system interactions

The system landscape model may appear as either a working view or a detailed view. The working view is used for design and discovery workshops, while the detailed view records requirements pertinent to stakeholder interaction scenarios, and the capabilities/ICDs/etc. required to support the requirements. The detailed view may be reviewed in a workshop. The audience for the system landscape model varies, but may include executives, IT Teams, and Operational Teams.

System Landscape Model

Sample



Appendix C.17: System Process Workflow Model

System Process Model	
Name	System Process Model
Alias(es)	Process Flow, Swim Lane Diagram, Workflow
Objective	The system process model visually illustrates the sequence of activities required to achieve an outcome within the current and/or future state design, but illustrated at the level of application components or systems. These diagrams provide the capability of understanding a process by orchestrating activities within and across departments and systems.
Definition	A system level process model describes a sequence of discrete activities starting from an initial state of the process to some defined end state, but illustrates interactions and messages to and from application components or systems. See Business Process Workflow Specification for the detailed specification and definitions. Some modelers choose to show application components all within the same pool, and others differentiate the different application components within systems as their own lanes within pools. It is at the discretion of the modeler.

System Process Model

Sample

