State of Tennessee

Division of TennCare

TennCare Enterprise Architecture Modeling Standard

Version: 1.0

November 2020

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

Table 1: Enterprise Architecture Modeling Standard Version History

Revision	Description of Change	Author	Date
1.0	Approved by TARB.	KPMG	November 13, 2020

1. Introduction

1.1. Purpose of Standard

This standard defines the artifacts (types of architectural models) to be submitted to TennCare throughout the duration of an implementation project. It provides modeling guidance to TennCare and vendor partners who are developing and reviewing TennCare's enterprise architecture and solution architecture for TennCare Information Systems (IS) projects.

Following the guidance in this standard will lead to more complete, consistent, and understandable architecture and design. See the TennCare Enterprise Architecture Framework Standard (EA Framework) for the benefits of enterprise and solution architecture, within each project and for the long-term management of TennCare IS.

Additionally, this standard informs vendors of TennCare's expectations for high-quality architectural modeling, consistent with industry standards and best practices. It encourages a pragmatic approach to developing enough models to yield the benefits of coherent and aligned architecture.

1.2. Audience of Standard

This standard is primarily written for the enterprise architects and solution architects of TennCare and its vendors. (See the TennCare EA Framework for definition of the architecture roles.)

Solution vendors will normally develop **Solution Design** artifacts only (see section 1.3). They may use the definitions of **Enterprise Architecture** and solution **Conceptual Design** artifacts as a reference.

This standard also serves as a reference to others who are reviewing enterprise and solution architectures for TennCare.

It is assumed that the reader is familiar with architecture and modeling of information systems, and with the business and technology they are modeling.

1.3. Architecture Scopes

Table 2: Architecture Scopes outlines three scopes that are fully defined in the TennCare EA Framework Standard. It shows the colour scheme used throughout this document: blue for Enterprise Architecture, red for Conceptual Design, and orange for Solution Design.

Table 2: Architecture Scopes

Architecture Scope	Description	Created By
Enterprise Architecture	Blueprints describing the current state of TennCare's business, information systems, infrastructure, and the vision for their future. Usually created by harvesting solution architectures.	TennCare and TAS enterprise architects
Conceptual Design	Describes the target state of a new or modified solution. Created during the Concept phase of the TennCare Solution Implementation Lifecycle. Solution vendors may also provide conceptual models of their solution, for comparison with TennCare's conceptual design.	TennCare,TAS, and Solution Vendor
Solution Design	Describes the target state of a new or modified solution at the logical and physical level. Usually created during the Design phase of the Solution Implementation Lifecycle.	Solution Vendor

Artifacts defined in this standard may be adapted for a different scope. For example, a Conceptual Design artifact may be created for the whole enterprise.

1.4. Architecture Domains

This standard defines model artifacts from all domains of architecture, including:

- Business Architecture
- Data Architecture (also known as Information Architecture)
- Application Architecture
- Technology Architecture
- Security architecture

In this standard, each artifact is aligned with one primary domain. This should not limit their use, as an architecture should be integrated across domains. In particular, see section 8.1 about the inclusion of security architecture in artifacts of other domains.

1.5. Models and Deliverables

The terms “artifact”, “model”, and “model artifact” all refer herein to architectural diagrams, charts, and their supplementary documentation.

The term “deliverable” refers to a document that must be delivered by a vendor under contract to TennCare. A deliverable may include enterprise architecture or solution architecture. One deliverable may include zero, one, or multiple models, along with sections that are not models.

Deliverables required for solution projects are listed in the TennCare Solution Implementation Lifecycle Standard (SILC) and its RACI and Deliverables chart. That chart describes the deliverables, including the models that they might contain.

The choice of models to include in deliverables depends upon the nature and size of the solution project and architectural judgement. Thus, the models and other sections to be included in a deliverable may be specified by any of the documents in *Table 3: Documents that may require models*, which all take precedence over this standard:

Table 3: Documents that may require models

Documents that may require models
• A template or standard from CMS • Documents from the TennCare Solution Implementation Lifecycle Standard (SILC) RACI and Deliverables chart • The RFP for the solution, or an attachment to it • The vendor’s proposal, and subsequent agreement with TennCare • The vendor’s contract • A Deliverable Expectations Document

1.6. Required, Recommended, and Suggested Artifacts

This standard specifies which artifacts are needed, within a given scope (i.e. Enterprise Architecture, Conceptual Design, or Solution Design):

- A **Required** artifact is mandatory
- A **Recommended** artifact is an expected best practice but may not be needed under some circumstances
- A **Suggested** artifact may be developed where warranted, per discussion with TennCare. Some artifacts are suggested for use during the consultation and ideation stages of architecture and design work but are not necessary for the final design documentation

The TennCare EA Framework Standard outlines the deliverables for Enterprise Architecture, Conceptual Design, and Solution Design activities. Architectural judgement is needed to select

from the recommended and suggested artifacts and to choose the level of detail in models. Architects are also encouraged to create artifacts that are not defined in this standard, where useful to further describe the enterprise or a solution Solution vendors will normally develop **Solution Design** artifacts only.

1.7. Scope and Applicability

1.7.1. In Scope

This standard:

- Provides definitions of “model artifacts” as defined in the TennCare EA Framework Standard.
- Defines model artifacts that are suitable for TennCare’s enterprise and solution architectures, including the Enterprise Architecture, Conceptual Design and Solution Design scopes defined in *Table 2: Architecture Scopes.*
- Defines model artifacts that are required, recommended and suggested for fulfilling the requirements of CMS, other regulatory bodies, and TennCare’s Solution Implementation Lifecycle (SILC) Deliverables.

1.7.2. Out of Scope

Solution Implementation Lifecycle (SILC) deliverables and phases are not in scope as they are external to the EA modeling standard. Refer to the TennCare Solution Implementation Lifecycle Standard (SILC) and its RACI and Deliverables chart for:

- A description of the lifecycle phases of an Information Systems solution projects
- The RACI chart, specifying who is responsible for deliverables and activities in each phase
- The chart of deliverables, specifying what documents must be delivered in which phase, and describing each one (see section 1.5).

Deliverables that do not contain model diagrams are not defined in the TennCare EA Modeling Standard.

Also not in scope of the EA modeling standard are non-modeling architecture practices and requirements management. Refer to the TennCare EA Framework Standard for:

- Descriptions of enterprise architecture and solution architecture
- Description of the Conceptual Design and Solution Design phases of solution architecture, and how they align with the SILC
- The process of architecture governance, which reviews the models

Refer to the TennCare Requirements Management Standard for the definition of functional and non-functional requirements, and of the Requirements Traceability Matrix.

This standard does not specify the order in which artifacts are created. Additionally, this standard does not define all the possible artifacts that might be useful. TennCare and its vendors should use their architectural judgement to choose additional artifacts that effectively portray the enterprise and solution architectures.

1.8. Structure of the Artifact Definitions

For each artifact listed and described in sections 3 to 8, this standard provides a table containing:

- The artifact **name**
- The architecture **scope** where the artifact is primarily used, from *Table 2: Architecture Scopes*
- The primary architecture **domain**
- **Optionality:** Whether the artifact is required, recommended, or suggested (see section 1.6)
- **Summary** of what the artifact expresses
- **Aliases:** other names for the artifact
- **Format:** the type of diagram or chart
- Mentions industry standard notations where applicable.
- Format is further elucidated in the definition and sample.
- **Purpose:** the artifact's contribution to the design and documentation effort
- **Related:** alignment with other artifacts

Below each table, this standard provides:

- **Definition:** What information goes in the artifact, and other instructions
- **Extensions:** Additional charts or diagrams that use the artifact's information
- **Sample:** An example diagram or chart, usually from a TennCare project's architecture.

The samples provided are illustrative and should not be considered as authorative representations of TennCare architecture components. Some samples have been redacted or fictionalized. This standard makes note of differences between the sample and the preferred artifact definition.

1.9. Support of TennCare's IS Guiding Principles

Creating architectural models according to this standard supports the TennCare IS Guiding Principles outlined within *Table 4: Support of TennCare IS Guiding Principles.*

Table 4: Support of TennCare IS Guiding Principles

TennCare IS Guiding Principle	EA Modeling Impact
Business-IT Alignment	TennCare's business is described by Enterprise and Conceptual Design artifacts in the Business Architecture domain. The Solution Design artifacts should be traceable to those Business Architecture artifacts, to ensure that the IS solution will support business operations.

TennCare IS Guiding Principle	EA Modeling Impact
Common Enterprise Processes	Models from solution architectures can be harvested to form TennCare's enterprise architecture. If the Solution Design models consistently follow this standard, it will be easier to find patterns that can be generalized into common processes. Harmonizing common processes across TennCare can result in cost savings and a more regular "customer experience" for all Stakeholders involved.
Comprehensive Integration	This standard defines the artifacts to be used in designing the integration of TennCare's information systems, such as the Solution Architecture Model. If the Solution Design models consistently follow this standard, it will be easier to interpret them when designing integrations between systems designed by different architects.
Technology Risk	Certain Solution Design artifacts can be used to manage technology risk, such as by checking for compliance with the TennCare Technology Standard.
Quality of Solutions and Services	If Conceptual and Solution Design models follow this standard, they will be more effective for making and checking design decisions, which can contribute to a higher quality of IS solutions and services.
Buy over Build	The modeling principles (section 2) recommend less-detailed modeling requirements when buying a solution than when custom-building a solution. This standard defines artifacts such as the Solution Architecture Model that map business concepts to the vendor's product names.

TennCare IS Guiding Principle	EA Modeling Impact
Managed Business Services over Software Applications	The Business Architecture artifacts should be used to describe the relationship of TennCare to the third-parties that manage business services on its behalf. The modeling principles (section 2) recommend less-detailed modeling when buying a managed business service than when custom-building a solution.
Cloud Deployment over On-Premises Deployment	Technology and security artifacts have variations for cloud and on-premises deployments.
SaaS over Source Code Transfer	The modeling principles (section 2) recommend less-detailed modeling when purchasing SaaS than when custom-developing source code.

1.10. Referenced Documents

TennCare Standards

TennCare Solution Implementation Lifecycle Standard
TennCare Enterprise Architecture Framework
TennCare IS Guiding Principles
TennCare Requirements Management Standard
TennCare Technology Standard
TennCare Data Policies and Standards

External Standards

1. Business Process Modeling Notation (BPMN 2.0.2), Object Management Group (OMG)

2. Unified Modeling Language (UML 2.5), Object Management Group (OMG)
3. MITA Toolkit: Medicaid Enterprise Certification Toolkit Version 2.3
4. CMS Data Management Operating Procedures and Guidelines
5. A Guide to the Business Architecture Body of Knowledge™ (BIZBOK™ Guide), Business Architecture Guild (2013)
6. The TOGAF® Standard, Version 9.2, The Open Group (April 2018).

2. Modeling Principles

The following section provides principles applicable to all architecture models, including specific standards and best practices, and making models consistent, traceable, and readable.

2.1. Assessment

Architectural models shall be developed so that TennCare can assess whether a solution will:

- Meet TennCare's business requirements
- Comply with CMS requirements
- Comply with TennCare standards (including the TennCare Technology Standard)
- Align with TennCare's IS Guiding Principles and enterprise architecture vision

When choosing what to show in models, architects should emphasize any risky, new, uncertain or expensive aspects of a solution, to aid in assessment.

2.2. Choice of artifacts

For TennCare projects supported by Federal Financial Participation (FFP), compliance with CMS toolkits and templates is required. The SILC Deliverables are based upon those requirements.

Within those requirements, this standard recommends a pragmatic and flexible approach when choosing model artifacts according to a project's needs. TennCare and its vendor partners should use their architectural judgement to:

- Choose artifacts defined in this standard
- Use artifacts not defined in this standard, per approval from TennCare
- Create models at varying levels of detail
- Adapt artifact attributes to show relevant information, such as by adding a property to elements, or color-coding a diagram
- Adapt an Enterprise artifact for use in a Conceptual Design, or vice versa
- Adapt a Conceptual Design artifact for use in a Solution Design, or vice versa

2.3. Modeling for Custom Development and Purchased Technology

As an extension of the principle above, *Figure 1: Glass Box and Gray Box* contrasts two sets of circumstances that require different levels of detail in the solution architecture.

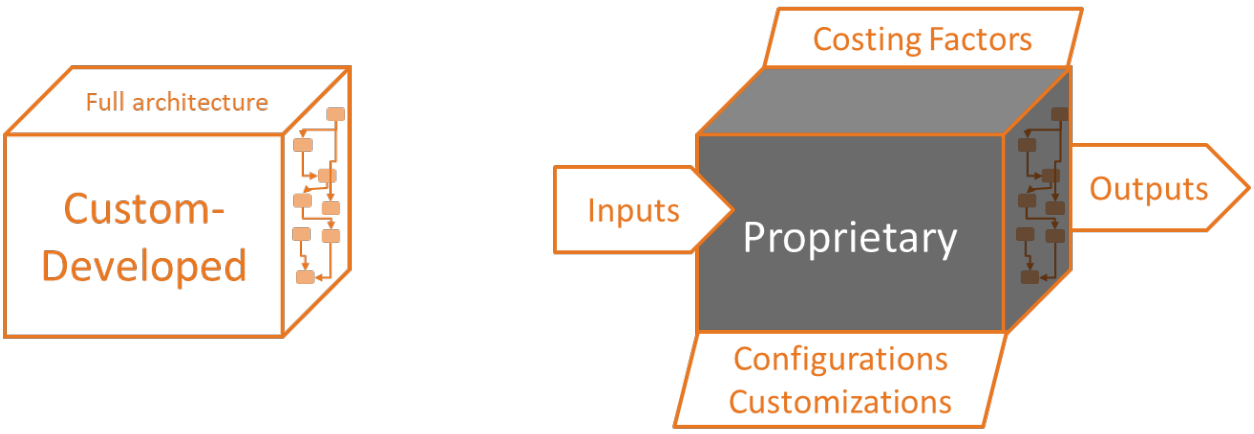

Figure 1: Glass Box and Gray Box

2.3.1. Glass Box for Custom Development

The Solution Design artifacts should be a “glass box”, providing **more design details** to make the full architecture visible, in any of the following circumstances:

- Development of a custom solution
- Extensive customization of a commercial off-the-shelf (COTS) solution
- On-premises deployment of technology
- Cloud-based deployments where more architecture insight is required such as IaaS and PaaS solutions

In these circumstances, TennCare IS staff will control or manage the solution directly, so they need full documentation for maintaining it. If TennCare is investing in custom-built software, the full Solution Design needs to be assessed to reduce risk. TennCare IS Guiding Principles discourage custom developed solutions, so glass box modeling will be less common in the future but will still be required for new releases of custom developed legacy solutions.

2.3.2. Gray Box for Purchased Technology

The Solution Design may be a “gray box”, providing **limited design details**, in any of the following circumstances:

- Purchase of COTS solution with minimal customization
- Subscription to a self-contained, managed service (e.g. SaaS or BaaS on the Cloud)
- Deployment on a cloud hosting service. For cloud deployments where TennCare has less management responsibility, less information will be needed in the technology architecture.

In these circumstances, TennCare is purchasing information technology that it does not own but has a license to use. This has different operations and maintenance than custom-developed solutions.

The architecture within any of these services or solutions may be proprietary to the vendor, meaning that TennCare will not have visibility into the “gray box”. TennCare IS will still need to know enough about the internal architecture of the systems to support the application and potentially integrate it with the application ecosystem. At a minimum, when there are known and potential architectural and oversight impacts (e.g. interfaces, data flows, audit requirements, volumetrics, and procurement/configuration insight), TennCare needs to know:

- The inputs and outputs of the service, such as data layouts and commands or procedures
- The parameters, configurations, and customizations of the service
- Factors that affect costing, such as data volumes or the number of users
- Defintions and/or specifications for application program interfaces (APIs)
- Whether the service meets TennCare’s IS Design Principles, Conceptual Design, Requirements, and quality expectations

The above can be expressed using architectural models and documentation. With sufficient models, the service will be a “gray box” that TennCare can assess and interact with.

In these circumstances, the Conceptual Design becomes a very important tool for communicating what TennCare needs. There is less opportunity to change the business requirements during Solution Design and development, without significant costs or delays.

2.4. Consistency and Traceability

The following practices should be followed while developing architecture model artifacts:

Consistency:

- Use the same name, properties and relationships for any element that appears in multiple artifacts.
- Use a modeling tool as discussed in section 2.7 to maintain consistency

Traceability:

- If a Conceptual Design artifact is derived from an Enterprise artifact, use the same element names, or provide a mapping chart or diagram
- If a Solution Design artifact is derived from a Conceptual Design artifact, use the same element names, or provide a mapping chart or diagram

2.5. Readability of Summary Views

Summary views of models are appropriate for executives and non-technical audiences. The reader will expect to grasp the content quickly, during a meeting. Guidelines for constructing summary model views include:

- Use of five to nine[1] elements per page which are easier to remember
- Focusing the reader's eye on the most important information by showing it in large symbols or fonts, in the upper middle of the diagram, and in bright colors like red.
- Ensuring, if the diagram shows a flow, go from left to right, and/or from top to bottom
- Use of fewer simple symbols that do not require a legend to understand
- Use of multiple colors or symbols only if the distinctions are important to the audience
- Simplification of Industry-standard notations or removal where they are not useful.
- Providing examples, icons or illustrations for any abstractions
- Optimization for readability with graphic design in Visio or PowerPoint
- Use of readable fonts with a minimum 14 point size on PowerPoint
- Sizing diagrams to fit on a PowerPoint slide (landscape orientation). If the diagram has large elements, this diagram will remain readable when copied into a Word document
- Reserving of details for other diagrams, charts and documents

2.6. Readability of Detailed Views

Detailed views are appropriate for more technical audiences where the reader can be expected to take some time to read and interpret the diagram or chart.

Guidelines for detailed diagrams:

- Provide information on one page, to the limits of readability
- Use readable fonts with a minimum 8 point size
- Use visually distinct symbols, line styles and colors to denote the properties of elements
- Provide a legend
- Make a visual hierarchy of information:
  - Put the base layer of important information in large symbols, large fonts, and black or saturated colors
  - To highlight a few elements, use bright colors such as red, and thicker lines
  - Moderately important details may be shown in smaller fonts and symbols, thinner lines, gray, and pastel colors
  - Less-important details should go on supplementary diagrams, charts and documents
- Design the diagram size and shape for how it will be presented:
  - PowerPoint slide (landscape orientation)
  - Word document page (portrait orientation)

---

[1] "Miller's Law" is that the number of objects an average human can hold in short-term memory is 7 ± 2. This is based on "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", by George A. Miller, Psychological Review, 1956.

- “Placemat” diagram (11”x17” landscape orientation, viewable on a large monitor or printout)
- Large poster (36” wide, variable length) (not suitable for remote work)

Organize detailed information systematically in charts/tables. Guidelines for detailed charts or documentation:

- Use readable fonts with a minimum 10 point size
- Ensure the details clearly relate to the diagram. Use numbered circles on the diagram to refer to sections of detailed documentation.
- Avoid tables with long text in many narrow columns, wrapping from page to page. Consider instead a structured document with a repeating pattern of headings and text stretched across the full page width

2.7. Format and Tools

All model artifacts must be available to TennCare in editable files, in a commonly-readable format. The following tools are acceptable:

- Sparx Enterprise Architect (preferred)
- An architecture modeling tool that is interoperable with Sparx Enterprise Architect (e.g. Erwin)
- Microsoft Visio
- Microsoft PowerPoint
- Microsoft Excel
- Microsoft Word

Models may also be presented in Adobe PDF format, in addition to the editable files.

Where appropriate for a particular architecture model, use of industry-standard modeling notations such as UML or BPMN is preferred.

Detailed diagrams should be developed or constructed in an architecture modeling tool, to ensure consistency across many diagrams. Refer to the EA Framework about Sparx software and interoperability.

Summary diagrams may be in Visio or PowerPoint to optimize graphic design, so long as they are consistent with the detailed information in the modeling tool.

Charts may be in various tools, with the following order of preference:

1. Output from a modeling tool (element properties, or a matrix of relationships between elements)
2. A spreadsheet
3. A table in a Word document

Detailed text documentation should be in a modeling tool, stored as the properties of the elements described. The text should be output from the modeling tool for display in a Word document, PDF or spreadsheet.

If the detailed text supplements a diagram, the diagram elements should be clearly linked to the detailed text, with matching names or circled numbers.

2.8. Naming Vendors and Products

As architectural models are often provided as part of a procurement and selection process, it is important to be careful in labeling model elements. When developing model artifacts:

Do use names of vendors and brands of IT products and services when:

- The product has already been procured (e.g. current state)
- Connection to the product is required by a Conceptual Design
- The product is proposed for use in a Solution Design

Do not use names of vendors or brands of IT products and services when it might compromise the fairness of procurement. Use vendor-independent names for:

- Target state of Enterprise Architecture and Conceptual Design
- Business Architecture, both current and target state

3. Summary of Artifacts

This standard defines the artifacts (types of architectural models) suitable for TennCare.

Table 5: Summary of Artifacts. For **Solution Designs** see sections 1.5 and 1.6 about artifact requirements for each solution project.

Table 5: Summary of Artifacts

Domain	Type	Enterprise Architecture	Solution Architecture	
			Conceptual Design	Solution Design
Business	Required	Capability Model Context Model	Business Process Model Business Scenarios	-
	Recommended	Business Goals Value Logic Model Maturity Assessment Business Service Model	Stakeholder Model Business Function Model	Business Rules
	Suggested	-	Business Operating Model Semantic Model	State Transition Diagram
Data	Required	Data Landscape	Conceptual Data Model	
	Recommended			Logical Data Model Physical Data Model Data Transformation Mapping
	Suggested			Report Specification

Domain	Type	Enterprise Architecture	Solution Architecture	
			Conceptual Design	Solution Design
Application	Required	Application Landscape	Application Component Model	System Process Model Interface Control Document
	Recommended	Application Roadmap Value to Quality Matrix	Alternatives Analysis	User Roles Solution Architecture Model
	Suggested	Heat Map		
Technology	Required	Interface Landscape	Conceptual Integration Architecture	
	Recommended		Volume & Performance Expectations	Deployment Model Network Topology
	Suggested			
Security	Required	Security elements are generally included in the architectural artifacts of Data, Technology, and Application domains. See Section XX – Security.	(Security elements are generally included in the architectural artifacts of Data, Technology, and Application domains. See Section XX – Security.	
	Recommended			System Boundary Diagram
	Suggested			

4. Business Architecture Artifacts

4.1. Core Artifacts and Relationships: Business

Figure 2: Core Artifacts and Relationships: Business shows the main relationships between the recommended Business Architecture artifacts. Artifacts from other domains are shown if they provide input to the Business Architecture or need alignment with it.


Business Architecture
Enterprise Artifacts
Conceptual Design
Solution Design
Business Goals
DERIVE
Value Logic Model
ELEMENT OF
DERIVE
Capability Model
ELEMENT OF
Maturity Assessment
Context Model
Business Service Model
AGGREGATION
ELEMENT OF
ELEMENT OF
AGGREGATION
Stakeholder Model
DERIVE
Application Component Model
FUNCTION AUTOMATION MAPPING
Business Function Model
ELEMENT OF
ELEMENT OF
AGGREGATION
Conceptual Data Model
ALIGN
Business Process Model
ALIGN
Business Scenarios
ELEMENT OF
Business Rules

Figure 2: Core Artifacts and Relationships: Business

4.2. Business Goals

Table 6: Business Goals

Business Goals		
Enterprise	Business Architecture	Recommended
Summary	The enterprise's long-term mission and goals.	
Aliases	Mandate, Mission, Objectives, Outcomes TOGAF artifact: Business Principles, Goals, Drivers, Objective Catalog	
Format	Text, or a graphic design containing text	
Purpose	Ensure that all target-state architecture is aimed at meeting the enterprise's goals.	
Related	Goals appear in the Value Logic Model.	

4.2.1. Definition

This artifact states what the enterprise (TennCare) is aiming to achieve in its long-term strategy. It includes any statements of purpose or intent, including the mandate, vision statement, mission statement, purpose, objectives, outcomes, benefits, priorities, etc.

Business Goals should be set and approved by business Stakeholders, not by architecture governance. The goals should be obtained from a strategy document, such as a business plan or enabling legislation. The enterprise's vision, values, principles, and/or drivers may be provided alongside the mission statement and goals.

Enterprise KPIs should measure achievement of the goals.

4.2.2. TennCare Business Goals

TennCare expresses its Business Goals as a mission statement and its agency priorities on their website.

The TennCare official site provides a current Vision and Mission of the organization.

4.3. Value Logic Model

Table 7: Value Logic Model

Value Logic Model		
Enterprise	Business Architecture	Recommended
Summary	A cause-and-effect diagram showing how programs and initiatives lead to outcomes that achieve the Business Goals.	
Aliases	Value Map, Value Model, Results Chain, Strategy Map, Program Logic Model Similar to: Logic Model, Theory of Change	
Format	Diagram, plus chart of performance metrics	
Purpose	Ensures initiatives and activities are aimed at meeting the enterprise's goals. See Usage, below.	
Related	Inputs: Business Goals, Capability Model	

4.3.1. Definition

The Value Logical Model (VLM) visually depicts the path of how program actions (including projects and other change initiatives) drive outcomes that ultimately lead to the achievement of benefits and longer-term business goals.

A VLM is structured as a chain of causes and effects, as depicted in *Figure 3: Form of the Value Logic Model*. The first step in the chain is a change initiative that should improve an organizational capability, leading to a transformational change to the enterprise. Those changes cause a series of outcomes that are intended to ultimately lead to the enterprise's strategic goals. The likelihood of each outcome can be considered during the strategy development. The actual progress towards the outcomes can be measured.

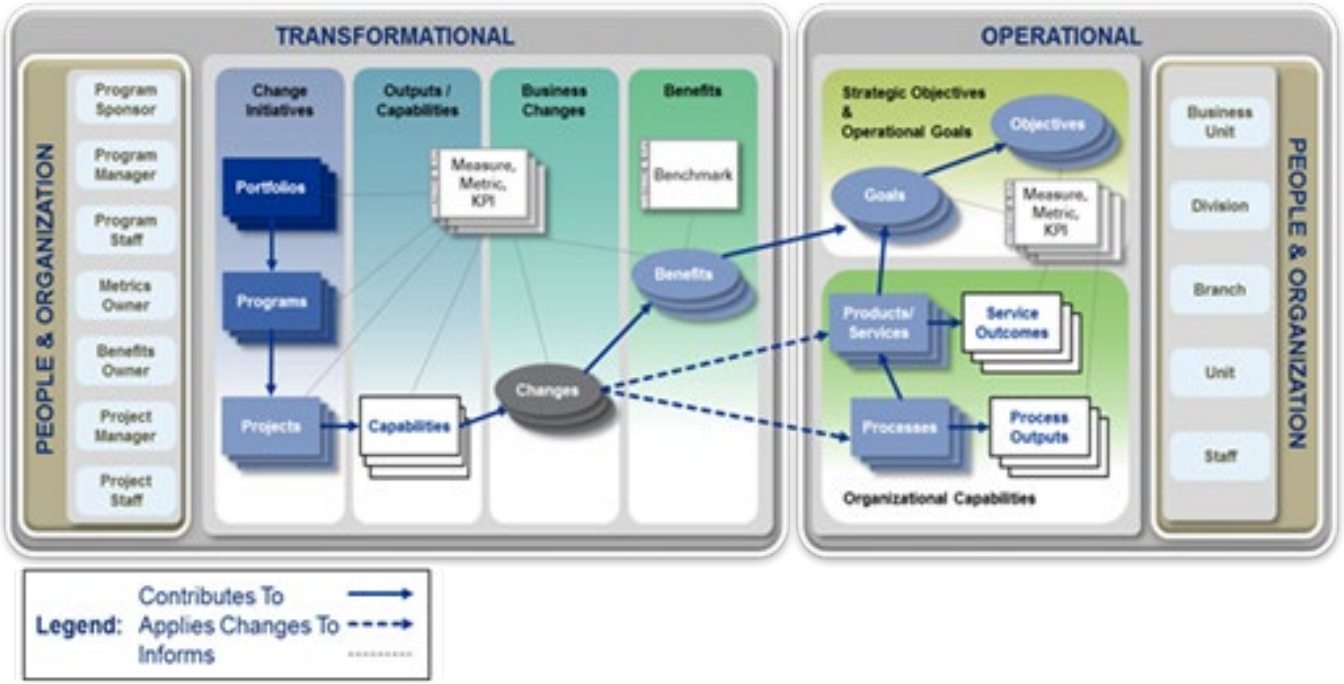

Figure 3: Form of the Value Logic Model

This pattern is depicted in *Figure 3: Form of the Value Logic Model* above, where outcomes are depicted as ovals, and four different levels of outcomes are portrayed. At the highest level are objectives (goals), which are the outcomes of a strategic vision. At the lowest level are business changes, which are direct outcomes from instituting new capabilities through actions taken by projects. The

levels inbetween are intermediate outcomes that need to be achieved to attain the objectives.

The diagram also illustrates that metrics are defined for changes and outcomes, to set performance targets and measure achievement of each level of outcome. Developing the VLM enables the definition of performance measures, since it makes explicit what the results of the actions are, and how they contribute to the achievement of the outcomes. The metrics for earlier stages in the chain of outcomes will be contributions to the Key Performance Indicators (KPIs) measuring the enterprise goals.

A Value Logic Model can be created for an entire enterprise or any program or part thereof, to articulate the strategic benefits of an operational business program or a transformation program, and how program activities contribute to the enterprise goals.

The VLM should be accompanied by a chart of performance metrics defined by each level of outcome in the Value Logic Model. Follow best practices for performance measurement to define the calculation of each metric, and set target levels.

4.3.2. Usage

The Value Logic Model ensures that program activities are aimed at meeting the enterprise's goals. Once completed, this artifact can be used to:

- Formulate a business case to justify an initiative
- Enable operational measurement and management to assure that program activities actually realize the expected performance and benefits
- Demonstrate business-IT alignment (one of TennCare's IS Guiding Principles) by showing how IS solution projects will achieve business goals

The development of a Value Logic Model should:

- Provide analytic clarity during strategy and planning
- Engage Stakeholders through the consultations that develop the model
- Compare options for program activities, based on the differences in outcomes and benefits that accrue from each option
- Define performance metrics that measure relevant outcomes

4.3.3. Sample

Figure 4: Sample of the Value Logic Model illustrates a Value Logic Model for an information management program, which may not represent current TennCare architecture.

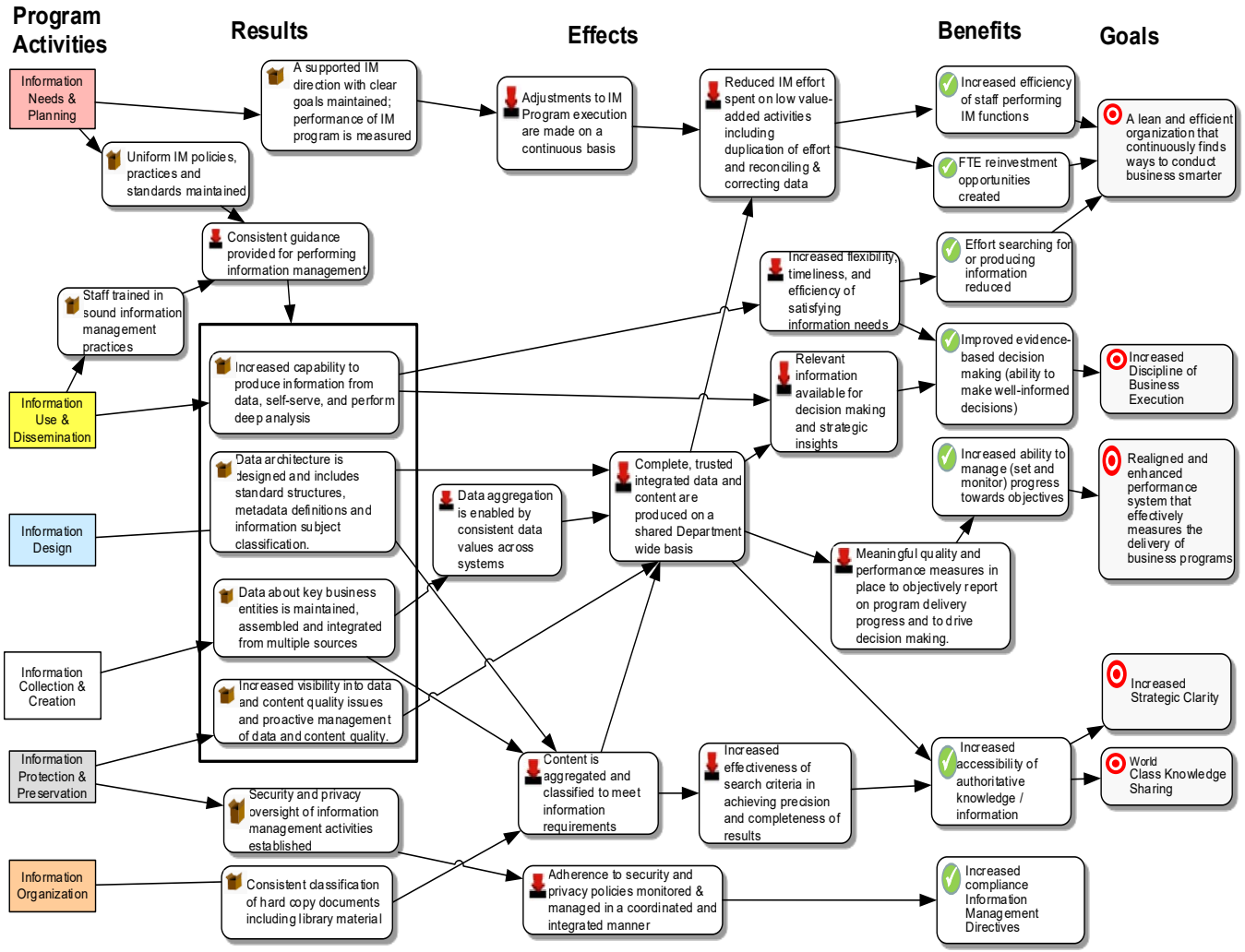

Figure 4: Sample of the Value Logic Model

4.4. Maturity Assessment

Maturity Assessment		
Enterprise	Business Architecture	Recommended
Summary	An assessment that scores an enterprise's operational capabilities on a pre-defined scale of maturity levels. Sets targets for maturity levels and recommends initiatives to reach the target maturity.	
Aliases	State Self-Assessment (SS-A), Capability Maturity Model (CMM)	
Format	Document	
Purpose	Evaluates the enterprise's capabilities relative to an industry norm. Provides clear and achievable targets for maturing the capabilities.	
Related	Capabilities assessed should be aligned with the Capability Model.	

4.4.1. Definition

Maturity assessments evaluate an enterprise's current capabilities, and state the enterprise's intention to improve each capability to a certain maturity level (the target state).

The capabilities and the maturity levels are defined in a capability-maturity model common to the industry and domain to be assessed. Typically, there are five (5) maturity levels defined for each capability.

Following the maturity assessment, initiatives such as IS solution projects, may be proposed to improve the capability maturity levels. The recommendations may state each initiative's anticipated impact on the maturity levels.

For State Medicaid Agencies, CMS defines the MITA State Self-Assessment (SS-A), which includes capabilities, maturity levels, and an assessment method.

4.4.2. Sample

An example from the MITA SS-A is provided in *Figure 5: Sample of the Maturity Assessment*, which may not represent current TennCare architecture.

Contractor Management Business Processes and MITA Maturity Levels

Business Process Name	MITA Maturity Level	
	As-Is	To-Be Goal
CO01 – Manage Contractor Information	1	2
CO02 – Manage Contractor Communication	1	2
CO04 – Inquire Contractor Information	1	2
CO05 – Produce Solicitation	1	2
CO06 – Award Contract	1	1
CO07 – Manage Contract	1	1
CO08 – Close Out Contract	1	1
CO09 – Manage Contractor Grievance and Appeal	1	2

Figure 5: Sample of the Maturity Assessment

4.5. Business Service Model

Table 8: Business Service Model

Business Service Model		
Enterprise	Business Architecture	Recommended
Summary	Illustrates the value chain of business service outputs delivered to customers/clients, other external Stakeholders, and from one part of the enterprise to another.	
Aliases	Value Chain Model Service Integration and Accountability Model (SIAM) Service Integration Model Value Mapping (similar)	
Format	Diagram. Ovals indicate services, arrows indicate outputs, person figures indicate clients/customers, and boxes indicate service provider organizations.	
Purpose	Ensures that the enterprise's mandate will be fulfilled and Stakeholder needs will be met, with no gaps or overlaps in the value chain. Used to determine and show the horizontal accountabilities (see below) of suppliers, partners, and organization units within the enterprise. Makes outsourcing arrangements clear. Identifies accountabilities that need to be formalized by legislation, policy, contract or agreement.	

Related	Use Stakeholder names and needs from the Stakeholder Model. Services are mapped to the Business Function Model to indicate the functions and processes that operate each service.

4.5.1. Definition

The Business Service Model illustrates the services that the enterprise delivers to customers/clients and other external Stakeholders. It may also show internal services delivered from one part of the enterprise to another.

A Business Service Model shows who delivers what output to whom. It only shows “final valued outputs”, meaning results that the recipient needs, such as eligibility for a TennCare program. It does not show every single output a service produces, but the most important one that the service exists to deliver.

The delivery of services makes up a “value chain”, in which the enterprise receives valuable outputs from suppliers, adds value through its own operations, and delivers valuable outputs to customers/clients. The value chain may include contracting or partnering with other Stakeholders who receive and/or deliver value.

The delivery of a service output is a “horizontal accountability” to a client or customer such as a Member. This is distinct from vertical accountability to a governing Stakeholder (which is shown in the Context Model). The horizontal accountability to deliver a type, quality and/or quantity of output may be specified in legislation, policy, a contract, or a Service Level Agreement.

Before a final valued output can be delivered, there are many pieces of information (such as application forms and validation data) that flow between the service provider, recipient, and other parties. Those intermediate information flows are not shown in the Service Model but they do appear in the Context Model and/or the Business Scenarios.

After a service output is delivered, it may be modified or withdrawn by the same service. This is implied, not shown on the model. For example, the Member Eligibility Service grants, updates, and removes eligibility for TennCare programs.

Descriptive information about each service is shown in a Service Profile chart (section 4.5.3). Each service is delivered by executing one or more business processes. A Process may be “reused” to execute multiple Services. Include a mapping from the Services to the Business Function Model, which groups processes into a hierarchy of functions. See example in section 4.5.4.

Note that the “business services” shown in *Figure 6: Sample of the Business Service Model* may not be the same as the automated application “services” in a service-oriented architecture (SOA).

The method of Value Mapping defined in BIZBOK[2] diagrams the transfer of value to stakeholders but does not identify services.

4.5.2. Sample

Figure 6: Sample of the Business Service Model may not represent current TennCare architecture.

[2] A Guide to the Business Architecture Body of Knowledge™ (BIZBOK™ Guide), Business Architecture Guild (2013).

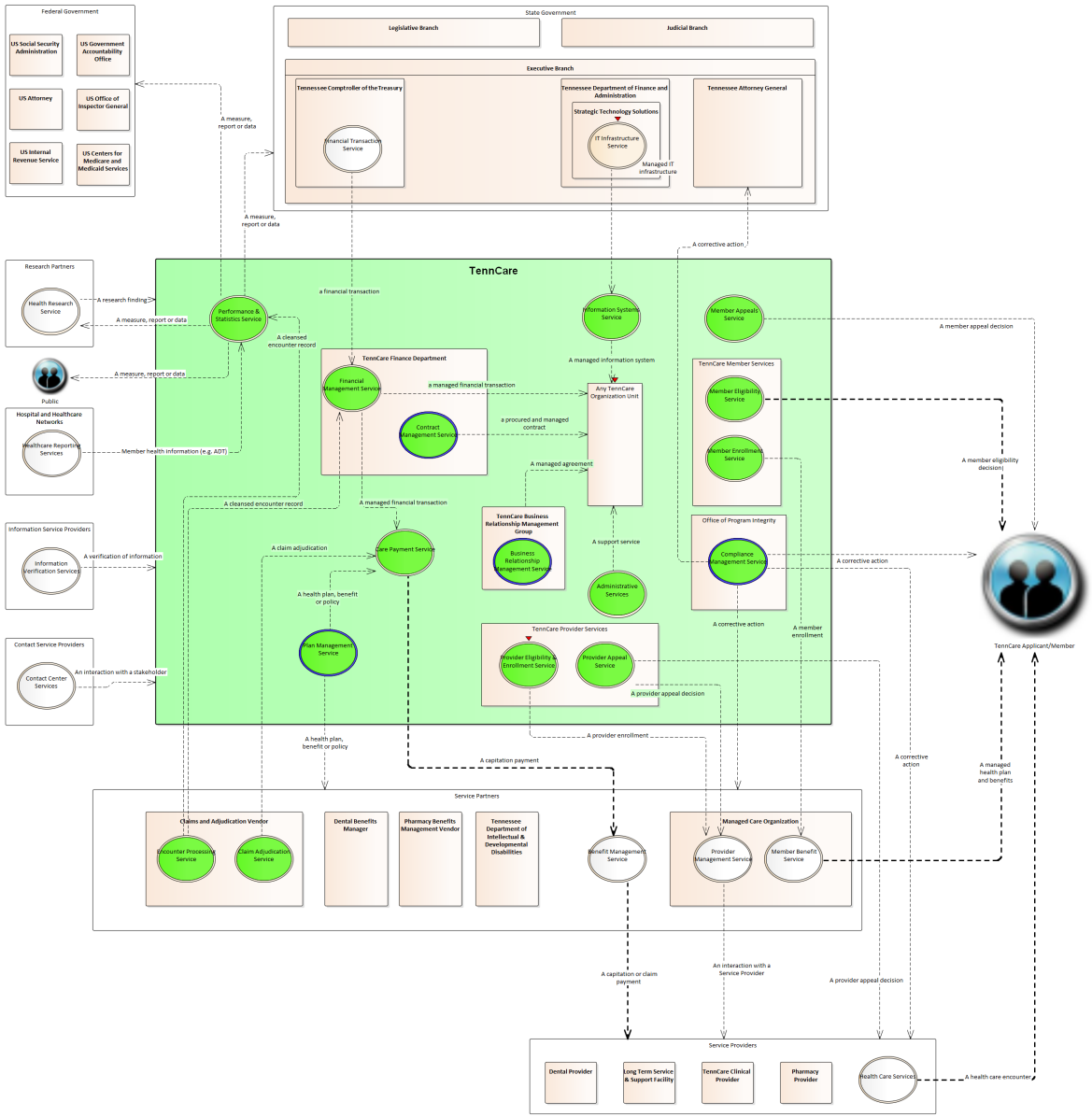
Federal Government
US Social Security Administration
US Government Accountability Office
US Attorney
US Office of Inspector General
US Internal Revenue Service
US Centers for Medicare and Medicaid Services
State Government
Legislative Branch
Judicial Branch
Executive Branch
Tennessee Comptroller of the Treasury
Financial Transaction Service
Tennessee Department of Finance and Administration
Strategic Technology Solutions
IT Infrastructure Service
Managed IT infrastructure
Tennessee Attorney General
A measure, report or data
A measure, report or data
A corrective action
Research Partners
Health Research Service
A research finding
A measure, report or data
TennCare
Performance & Statistics Service
A cleansed encounter record
a financial transaction
Information Systems Service
A managed information system
Member Appeals Service
A member appeal decision
Public
A measure, report or data
TennCare Finance Department
Financial Management Service
a managed financial transaction
Any TennCare Organization Unit
TennCare Member Services
Member Eligibility Service
Member Enrollment Service
Hospital and Healthcare Networks
Healthcare Reporting Services
Member health information (e.g. ADT)
Contract Management Service
a procured and managed contract
A managed agreement
A member eligibility decision
A cleansed encounter record
A managed financial transaction
TennCare Business Relationship Management Group
Business Relationship Management Service
Office of Program Integrity
Compliance Management Service
Information Service Providers
Information Verification Services
A verification of information
A claim adjudication
Care Payment Service
A support service
Administrative Services
A corrective action
A corrective action
TennCare Applicant/Member
A health plan, benefit or policy
Plan Management Service
TennCare Provider Services
Provider Eligibility & Enrollment Service
Provider Appeal Service
A member enrollment
Contact Service Providers
Contact Center Services
An interaction with a stakeholder
A provider appeal decision
A health plan, benefit or policy
A provider enrollment
A capitation payment
A corrective action
A managed health plan and benefits
Service Partners
Claims and Adjudication Vendor
Encounter Processing Service
Claim Adjudication Service
Dental Benefits Manager
Pharmacy Benefits Management Vendor
Tennessee Department of Intellectual & Developmental Disabilities
Benefit Management Service
Managed Care Organization
Provider Management Service
Member Benefit Service
A capitation or claim payment
An interaction with a Service Provider
A provider appeal decision
Service Providers
Dental Provider
Long Term Service & Support Facility
TennCare Clinical Provider
Pharmacy Provider
Health Care Services
A health care encounter

Figure 6: Sample of the Business Service Model

4.5.3. Service Profile

The Business Service Model should be accompanied by a chart of information describing each service. *Table 9: Service Profile* specifies the information elements to include in each service profile. Information shown on the Service Model diagram may be omitted from the chart.

Table 9: Service Profile

Profile Element	Profile Element Content	Optionality
Service Name	Name of the service.	Recommended; shown on diagram
Description	Describes what the service is and what its objectives are. Notes what is in and out of scope.	Recommended
Output	A measurable outcome of value from the clients perspective.	Recommended; shown as an arrow label on the diagram
Contribution to Business Goal	Describes how the overall program needs are being met.	Suggested
Owner	The organization accountable for the overall success of the service. (Might not be the provider if the service is outsourced.)	Recommended

Profile Element	Profile Element Content	Optionality
Provider	The organization responsible for providing the service	Recommended; shown as a box on the diagram
Delivery Model	Indicates if the service is directly delivered or outsourced (managed service).	Suggested
Client Group	Identifies who the service is being offered to. A Stakeholder name.	Recommended; shown as a symbol on the diagram
Need(s) Addressed	Identifies the benefits and/or client needs which will be met through the delivery of the service. From the Stakeholder Needs, see section 4.11.3.	Suggested
Performance Measures	Metrics for efficiency, quality and effectiveness.	Suggested
Risks	Identify key risks associated with the service	Suggested
Key Service Processes	Name the business functions or processes that are key to operating the service. May be shown as a list, mapping chart or mapping diagram. See example below in section 4.5.4.	Recommended

4.5.4. Service to Function Mapping

Figure 7: Sample excerpt of Service to Function mapping is an example of a mapping from Services to Functions. Each Function represents a group of processes as described in Section 4.7. This sample may not represent current TennCare architecture.

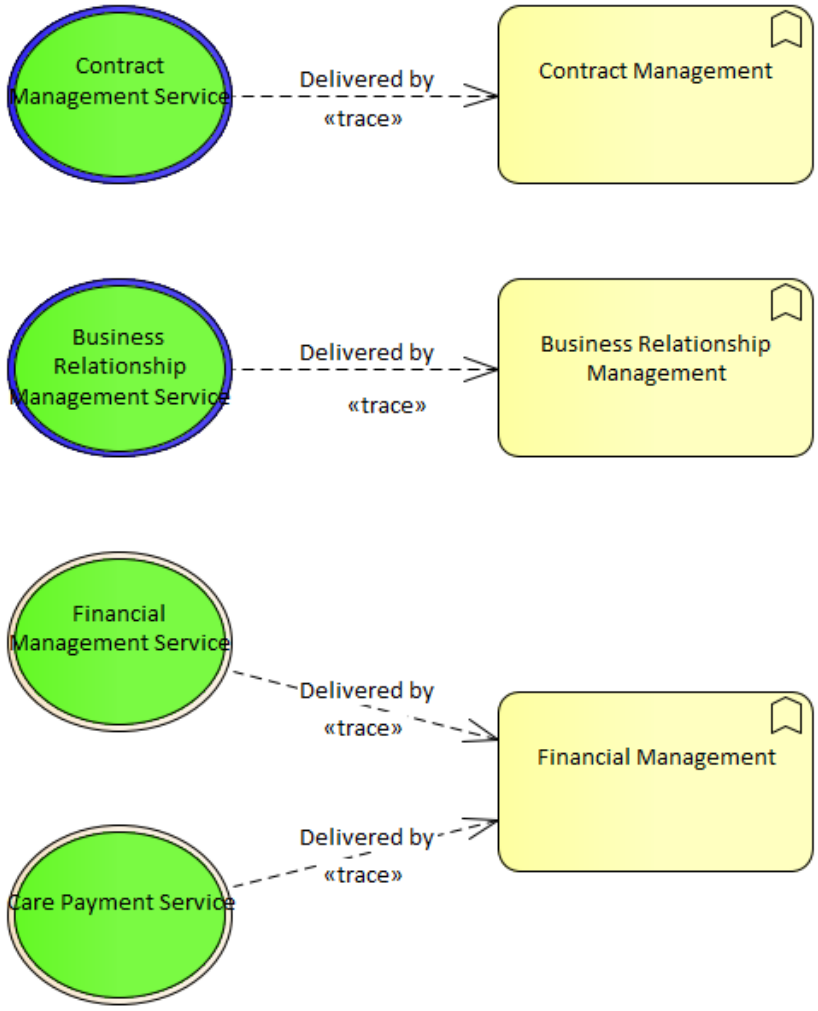

Figure 7: Sample excerpt of Service to Function mapping

4.6. Capability Model

Capability Model		
Enterprise	Business Architecture	Required
Summary	A hierarchical decomposition of the enterprise into its capabilities.	
Aliases	TOGAF artifact: Capabilities	
Format	Diagram with nested boxes, or tree diagram, plus a chart with definitions	
Purpose	Basis for assessing the maturity of capabilities, to pinpoint where automation or staff development may be required. Basis for designing the organization structure to better deliver the target state vision.	
Related	The Application Landscape maps applications to the capabilities they deliver. The capabilities in-scope for a solution are decomposed by the Business Function Model. The Maturity Assessment is applied to the capabilities.	

4.6.1. Definition

The Capability Model is a hierarchical decomposition of all the capabilities of the enterprise, in its current or target state.

According to CMS, a capability is the competence of an individual, organization, or system to perform a function or process[3]. According to TOGAF[4], a capability is "An ability that an organization, person or system possesses." According to BIZBOK[5], a capability "defines what a business does".

A capability results from combining resources, including funding, skills, expertise, information systems, and physical infrastructure. Capabilities reflect the enterprise's mandate to serve clients. Most Capabilities name business outcomes, while some may be internal supports. An example capability is "Provider Management".

Capabilities are the building blocks needed to facilitate performance improvement and transformation analysis and design. Automation design is based on capabilities that are available as categories of software. The Capability Model may show which staff are doing similar work and thus could be organized to work together.

A Maturity Assessment scores each of the capabilities. The Capability Model may indicate the maturity level of each capability that the organization needs. The target-state Capability Model may include lower-maturity capabilities that the organization needs to develop.

The capability model provides a fairly stable view of the organization because 'what' the organization does seldom changes. There is more change in 'how' it is done through operational procedures and information systems, as well as 'who' does it.

If an organization has delegated a responsibility to a third party, the Capability Model should show the relevant capabilities of the third party within a separate boundary.

---

[3] MITA *State Self-Assessment (SS-A) Companion Guide*, Centers for Medicare & Medicaid Services, v.3.0 (February 2012).
[4] THE TOGAF® STANDARD, VERSION 9.2, The Open Group (April 2018).
[5] A Guide to the Business Architecture Body of Knowledge™ (BIZBOK™ Guide), Business Architecture Guild (2013).

4.6.2. Capabilities and Functions

Capabilities are defined at the Enterprise level in the Capability Model. For the Conceptual Design of each solution, the Business Function Model defines the functions in scope, and decomposes them into processes.

To differentiate between Capabilities and Functions:

Capabilities are defined to reflect the enterprise's mandate to serve clients. Most Capabilities name business outcomes, while some may be internal supports.

A Function is a grouping of operational activities within an enterprise. The Business Function Model defines internal activities without duplication. See section 4.8.

Therefore, one Capability may be delivered by multiple Functions. One Function may support delivery of multiple Capabilities. The business architecture should keep this many-to-many relationship as simple as possible. In some cases, the Function and the Capability are the same, and should use the same name.

Any capability matching a function in the MITA Toolkit must be named as defined by CMS.

4.6.3. Sample

Figure 8: Sample of the Capability Model . It may not represent current TennCare architecture.

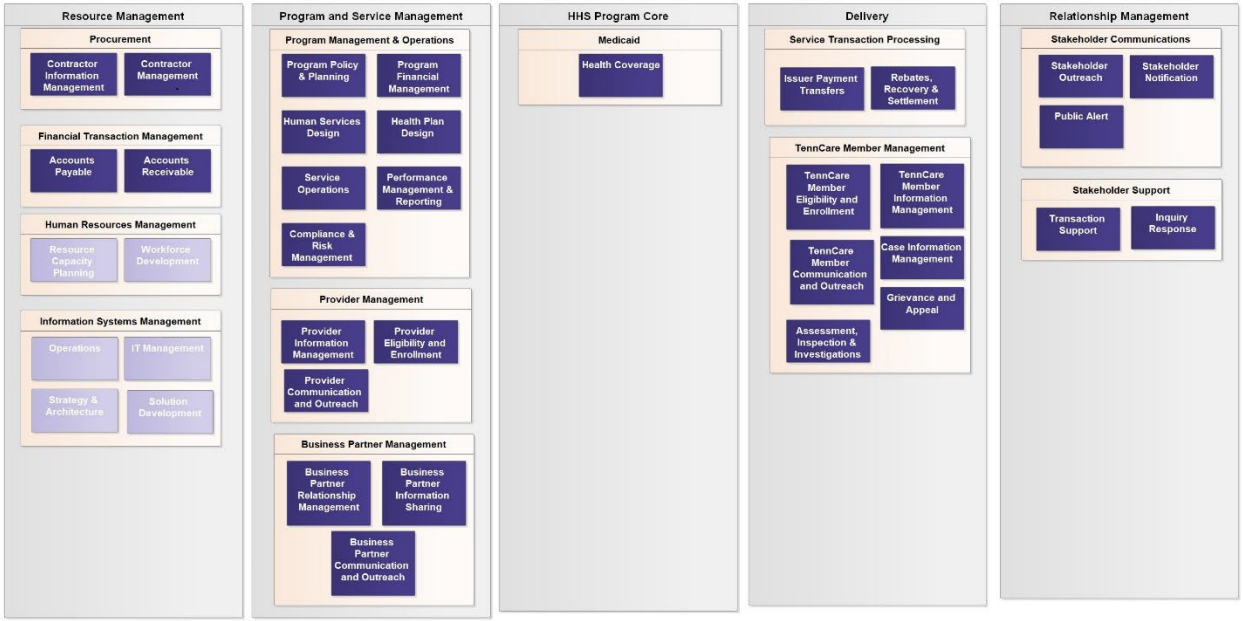

Figure 8: Sample of the Capability Model

4.7. Context Model

Table 10: Context Model

Context Model		
Enterprise	Business Architecture	Required
Summary	The Context Model shows the interactions of the enterprise with its Stakeholders.	
Aliases	Conceptual Blueprint, Context Diagram, Business Context Model	
Format	Diagram	
Purpose	Used to establish the scope of the enterprise in focus. Identifies Stakeholder relationships that need to be supported by the business. Identifies information flows that may need to be automated. Supports early identification of security or risk concerns related to external interactions.	
Related	All Stakeholders should match the Stakeholder Model.	

4.7.1. Definition

The Context Model shows the interactions of the enterprise with its Stakeholders.

The enterprise is depicted as a box with no details inside it, as depicted in *Figure 9: Sample Context Model*. Four types of Stakeholders are shown surrounding the enterprise:

- Above: Governors (government and other organizations that regulate, control, or monitor the enterprise)
- At right: Clients and customers (groups of individuals and organizations that benefit from the services of the enterprise)
- At left: Suppliers (organizations that provide goods and services, which the enterprise procures)
- Below: Partners (organizations that agree to collaborate with the enterprise)

Arrows indicate the information and resource flows in and out of the enterprise. Example: “Fee payment”. This model shows high-level (generalized) flows that may be detailed in the Business Scenarios.

The Context Model does not depict the functions within the enterprise that complete the interactions, nor does it decompose the enterprise into organization units. (See the Business Operating Model for this model decomposition.)

4.7.2. Sample

Figure 9: Sample Context Model illustrates TennCare’s relationships with stakeholders. It may not represent current TennCare architecture.

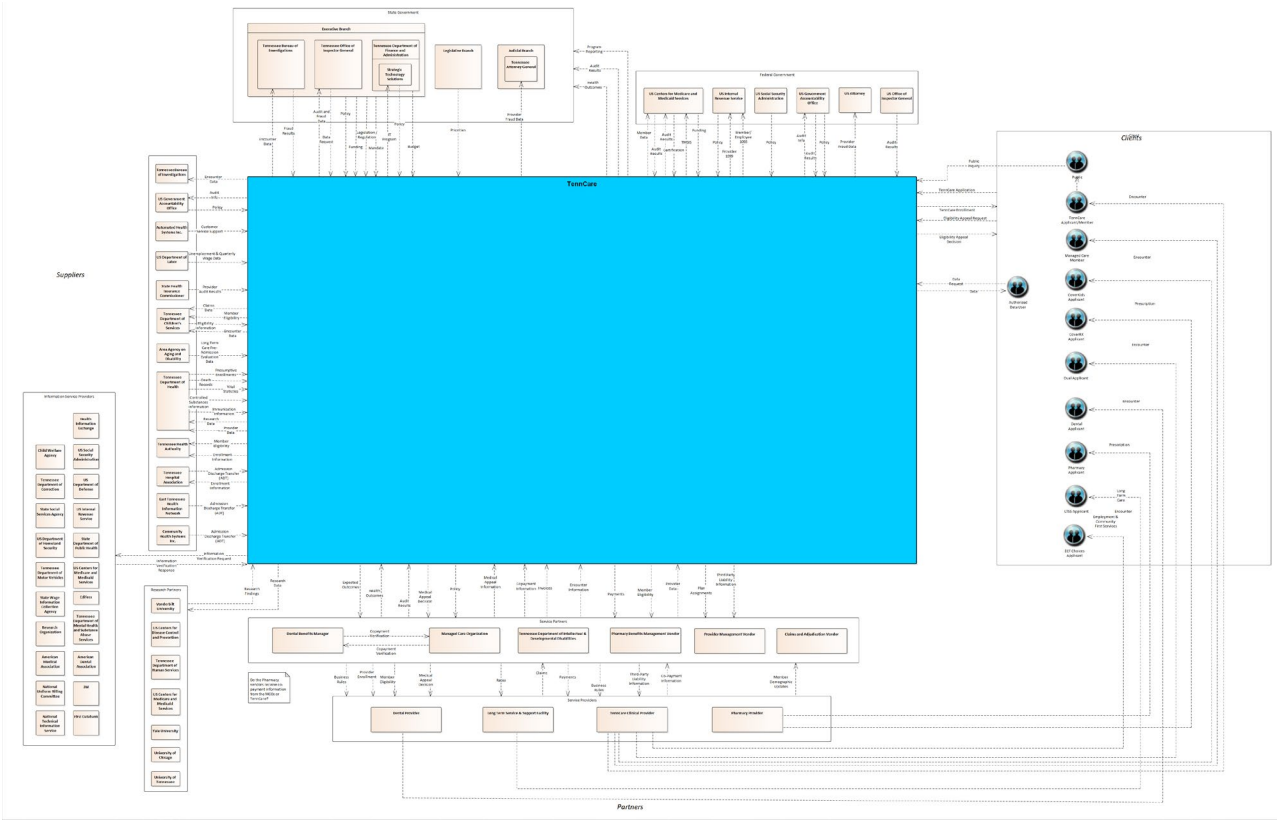

Figure 9: Sample Context Model

4.7.3. Extension: Stakeholder Interactions

A chart with columns such as the following can help to plan the Application Roadmap, to design solutions, and to estimate the effort required for automation:

- Stakeholder name
- Information flow (short description)
- Currently automated?

- Automated in target state?
- Volume of interactions per period

Some of these columns are shown in this example, *Figure 10: Sample of Stakeholder Interactions*, which may not represent current TennCare architecture

Stakeholder	Data Exchange	Exchange Type
US Attorney	Provider Fraud Data	Manual
Office of Inspector General	Audit Results	Automated
Tennessee Bureau of Investigations	Encounter Data	Automated
GAO	Audit Info	Automated
	Policy	Manual
Automated Health Systems	Customer Service Support	Automated
Department of Labor	Unemployment and Quarterly Wage Data	Automated
State Health Insurance Commissioner	Provider Audit Results	Automated

Figure 10: Sample of Stakeholder Interactions

4.8. Business Function Model

Table 11: Business Function Model

Business Function Model		
Conceptual Design	Business Architecture	Recommended
Summary	Outlines the business functions in scope for a solution, and decomposes them into a hierarchy of processes.	
Aliases	Process Hierarchy, Function-Process Decomposition	
Format	Diagram with nested boxes, or tree diagram, plus definitions	
Purpose	Shows what business activities are in scope for a solution. Provides categories to organize processes, data and other details.	
Related	The Capability Model outlines all functions of the enterprise. The capabilities decompose into functions. The functions decompose into the processes, which are detailed in the Business Process Model. Provide a Function Automation Mapping of applications (from the Application Component Model) to the functions they automate.	

4.8.1. Definition

The Business Function Model defines the functions in scope for a solution, and decomposes them into a hierarchy of processes.

A function is an ongoing activity done by the organization. The Business Function Model defines internal, operational activities without duplication. See section 4.6.2 for the difference between Functions and Capabilities, and how those models should be aligned.

The function model provides a fairly stable view of the organization because ‘what’ the organization does seldom changes. There is more change in ‘how’ it is done through operational procedures and information systems, as well as ‘who’ does it.

The functions and processes in the MITA Toolkit must be used as defined by CMS. Other functions are defined by TennCare.

If the same function or process is in scope for multiple solutions, it should be defined the same way. (The Conceptual Designs should be harvested into the Enterprise Architecture to enable aligning the functions and processes.)

4.8.2. Sample

Figure 11: Sample excerpt of the Business Function Model shows the decomposition of one MITA function into processes. This may not represent current TennCare architecture.

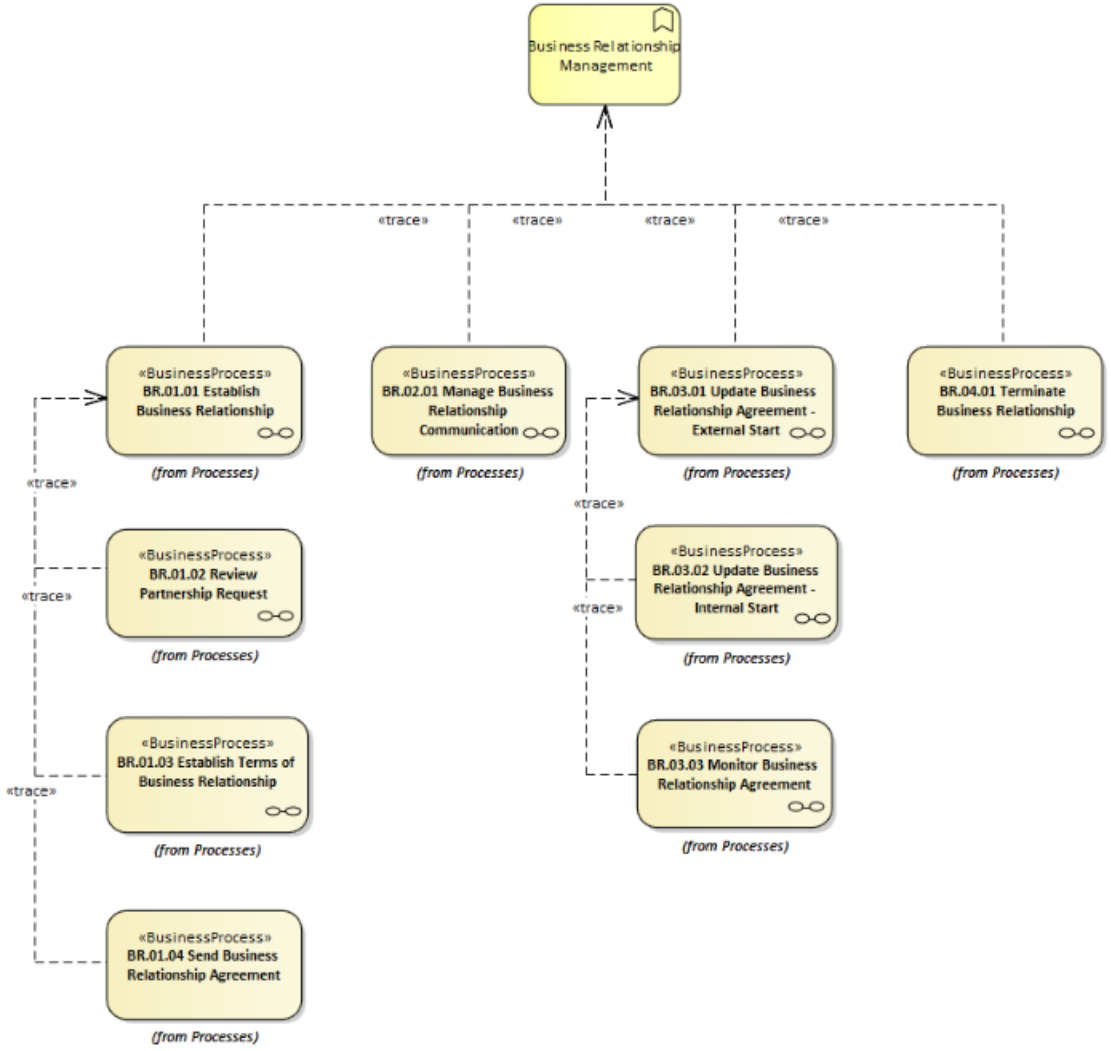

Figure 11: Sample excerpt of the Business Function Model

4.8.3. Extension: Function Automation Mapping

The Business Function Model should be extended by this recommended supplementary mapping:

Provide a chart or diagram that maps the Business Functions to the applications that automate them. Automation phases may be indicated with colors or separate columns. The mapping may show current applications with product names, and/or future application modules with business names from the Application Component Model, as seen in *Figure 12: Sample excerpt of Function Automation Mapping*. This may not represent current TennCare architecture.

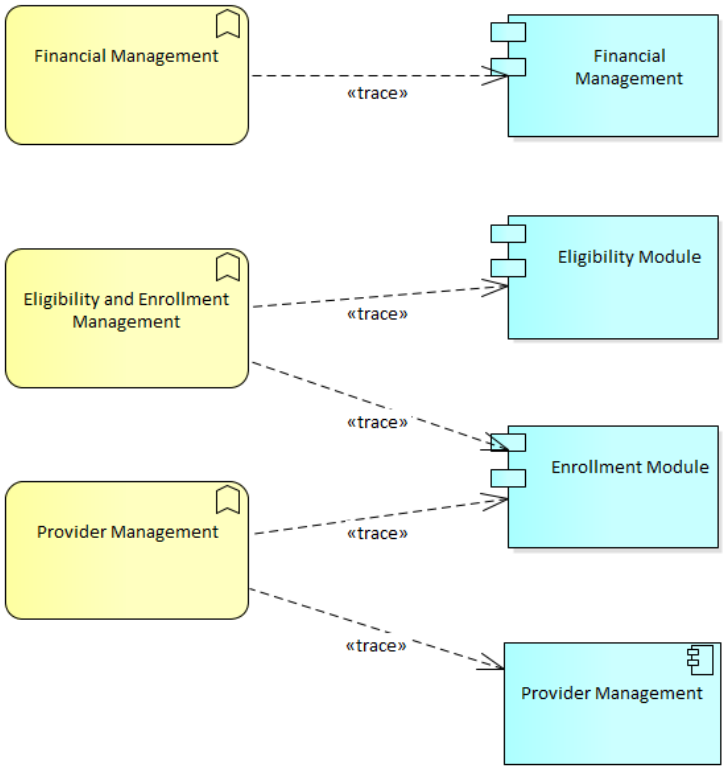

Figure 12: Sample excerpt of Function Automation Mapping

If more detail is required, a mapping from business processes (instead of functions) to applications should be completed, as seen in *Figure 13: Sample of mapping processes to applications*. This fictionalized example does not represent current TennCare architecture.

APPENDIX E – MITA PROCESS APPLICATION MAPPING LIST

MITA Process	Application Name
Apply Mass Adjustment	System ABC
Authorize Referral	BCD Tool
Award Contract	CDE Financial System
Award Contract	DEF Document Manager
Calculate Spend-Down Amount	System EFG
Determine Adverse Action Incident	System ABC
Determine Member Eligibility	System ABC
Determine Member Eligibility	System EFG
Determine Provider Eligibility	System ABC
Determine Provider Eligibility	FGH System
Disenroll Member	System EFG

Figure 13: Sample of mapping processes to applications

4.9. Business Process Model

Table 12: Business Process Model

Business Process Model		
Conceptual Design	Business Architecture	Required
Summary	Workflow diagrams showing the business processes in scope for a solution. Shows who does which step, in what sequence.	
Aliases	Swimlane Diagram, Process Workflows TOGAF Artifact: Process Flow Diagram	
Format	Workflow swimlane diagrams in simplified BPMN format	
Purpose	Provides enough detail about business activities to design or select a solution that can automate the required processes. Modeling the current state of a process may show opportunities for improving it. Modeling the target state of a process should determine how the process will be re-engineered for efficiency, effectiveness and alignment across the enterprise.	
Related	The Business Function Model organizes the processes into a hierarchy. Swimlanes should match the Stakeholder Model. Requirements are specified for many steps of the processes (see the TennCare Requirements Management Standard). Information required within processes should be compiled in the Logical Data Model.	

4.9.1. Definition

The Business Process Model is a set of workflow diagrams describing the business processes that are in scope for a solution as seen in *Figure 14: Sample of the Business Process Model*. The workflow diagram shows who does which step of the process, in what sequence.

A business process is a series of activities that, once completed, will accomplish an organizational objective. Business processes are defined by describing the tasks and which organization is responsible for doing them, without describing the technology they might use to automate the tasks.

In simplified BPMN notation (section 4.9.4), each activity (task) is shown within a rectangular “swimlane” representing the organization that does the activity. The arrows show the flow or sequence of events, activities and messages. The workflow may follow different paths depending on branching logic at gateways (decision points).

Requirements, business rules or information resources may be linked to the workflow step where they apply.

4.9.2. Functions and Processes

For the Conceptual Design of a solution, the Business Function Model defines the functions in scope, and decomposes them into named processes. The Business Process Model shows how each process happens.

Any process matching a process in the MITA Toolkit must be named as defined by CMS.

If the same process is in scope for multiple solutions, it should be defined the same way. (The Conceptual Designs should be harvested into the Enterprise Architecture, to enable aligning functions and processes.)

4.9.3. Sample

Figure 14: Sample of the Business Process Model shows the set of workflow diagrams describing the business processes that are in scope for a solution, which may not represent current TennCare architecture.

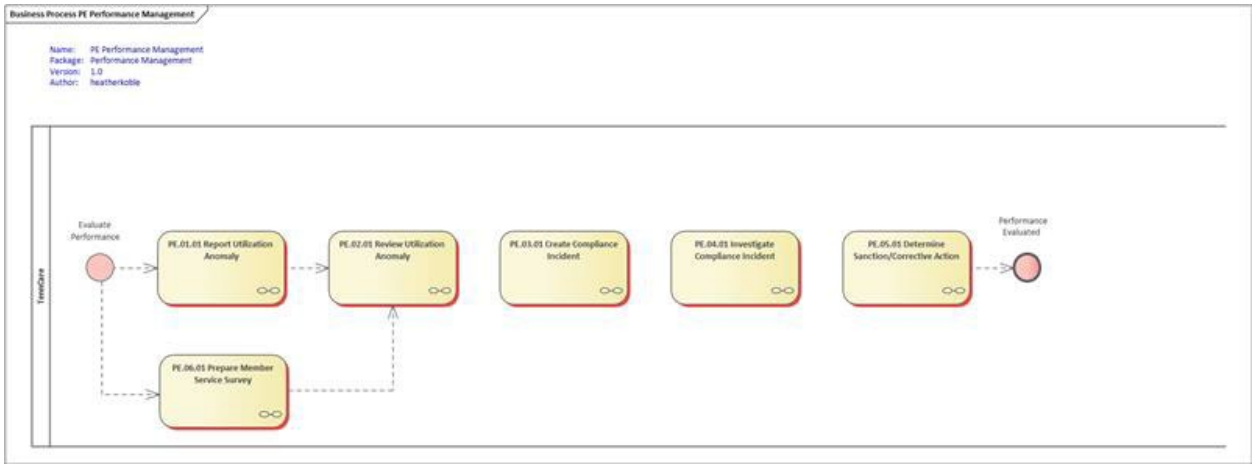

Figure 14: Sample of the Business Process Model

4.9.4. Simplified Business Process Modeling Notation

Figure 15: Simplified Business Process Modeling Notation (BPMN) and *Figure 16: Simplified BPMN Notation Set* display simplified elements of the business process modeling notation standards that should be followed when producing process models.

Business Process Modeling

Business Process Modeling Notation (BPMN) is a flow chart method that models the steps of a planned business process from end to end. A key to Business Process Management, it visually depicts a detailed sequence of business activities and information flows needed to complete a process.

BPMN depicts these four element types for business process diagrams:

- Flow objects: events, activities, gateways
- Connecting objects: sequence flow, message flow, association
- Swimlanes: pool or lane
- Artifacts: data object, group, annotation

There are many symbols in BPMN but for a business modelling context we traditionally only use a couple of these symbols:

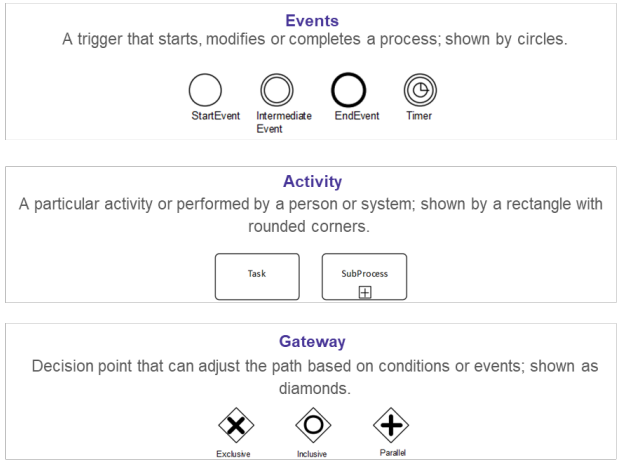

Figure 15: Simplified Business Process Modeling Notation (BPMN)

Simplified BPMN Notation Set

Name	Symbols	Description
Pools & Lanes		Lanes organize process flow elements and are grouped into pools. A pool can only contain one process.
Events (Start & End)		How a process starts and when it is complete are indicated by Start and End events.
Gateways (Exclusive & Parallel)		Tests a data condition to determine the appropriate exclusive or concurrent sequence flow(s).
Activities (Sub-process or Task)		Work (functionality) that is performed and can be atomic (task) or compound (sub-process). Always triggered by the arrival of the incoming sequence flow.
Flows (Sequence & Message)		Sequence flows indicate the flow of process logic from the start event to the end event. Message flows are optional.

Figure 16: Simplified BPMN Notation Set

4.10. Business Scenarios

Table 13: Business Scenarios

Business Scenarios		
Conceptual Design	Business Architecture	Required
Summary	The story of a business transaction or interaction scenario, from end to end. Shows the process and information flows between Stakeholders and their Business Functions or Applications.	
Aliases	N/A	
Format	Diagram with text narrative	
Purpose	Developing scenarios elicits business requirements and the application design. Documented scenarios ensure solution vendors understand and implement each step in the scenario, so that the integration of applications may be tested.	
Related	A scenario includes organizations from the Stakeholder Model. A scenario details interactions from the Context Model and/or Business Operating Model. A scenario summarizes many workflows from the Business Process Model. The information flows inform the Conceptual Data Model. Scenarios may include applications from the Application Component Model.	

4.10.1. Definition

Business scenarios are narratives that can be used to illustrate, in plain business language, current state or target state business stories with clear outcomes.

The scenario diagram includes:

- Boundary boxes representing Stakeholder organizations or individuals who interact
- Nodes within the Stakeholder boundaries, representing Capabilities, Business Functions, or conceptual Application Components
- Numbered arrows from one node to another, indicating the sequence of steps
- Labels on the arrows, indicating what information flows between the nodes. (These information requirements should also be reflected in the Conceptual Data Model.)

The scenario's narrative text includes:

- Precondition(s) and starting points
- Numbered explanation of what happens at each step in the sequence of interactions
- An outcome (result at the end)

This artifact becomes a conceptual "data flow diagram" if it shows information flowing between Application Components.

Figure 17: Sample Business Scenario provides an example business scenario, which may not represent current TennCare architecture. The Business Scenarios may update and extend the Stakeholder Interactions chart (section 4.7.3) that accompanies the Context Model.

4.10.2. Sample

The following sample may not represent current TennCare architecture.

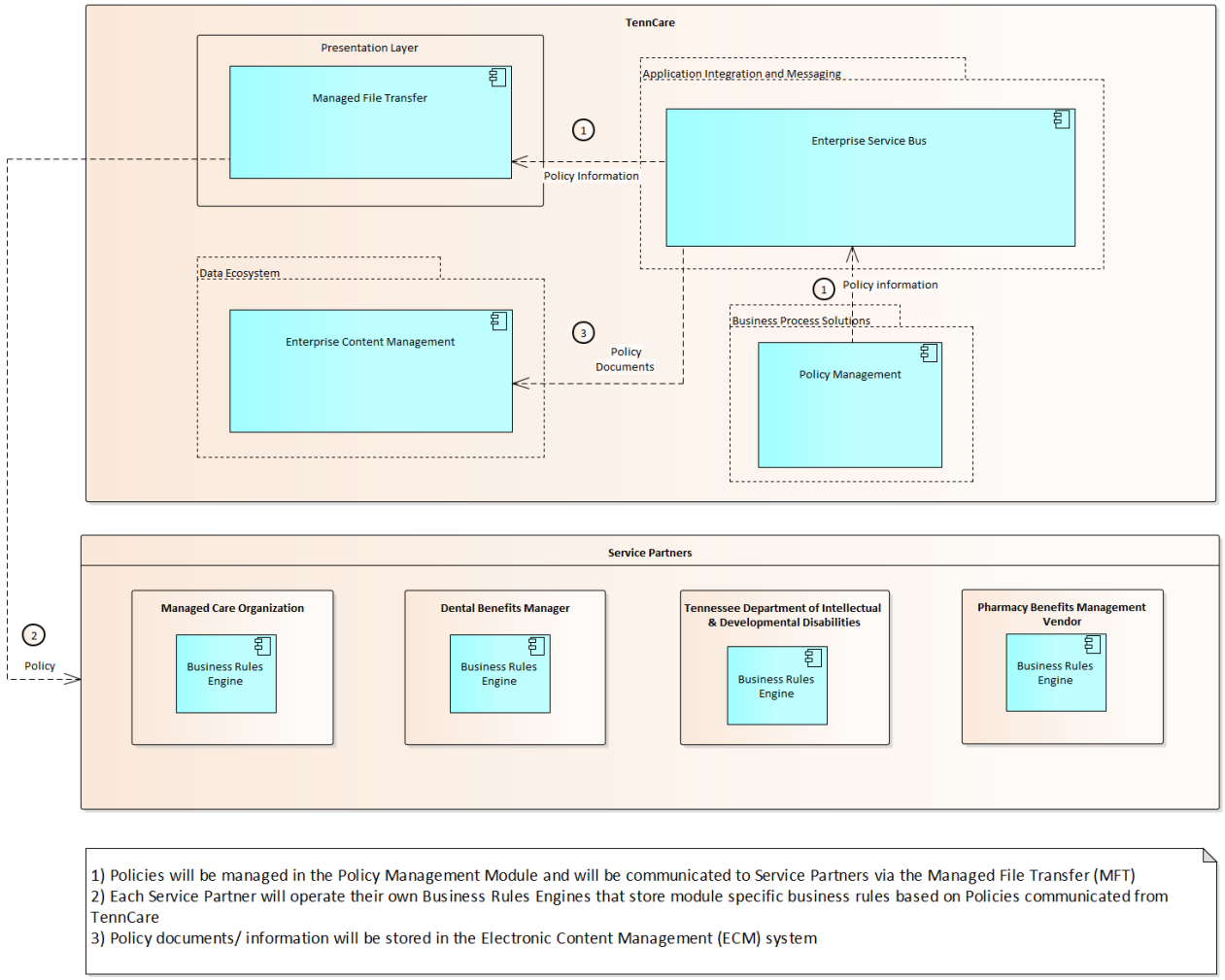

Figure 17: Sample Business Scenario

4.11. Stakeholder Model

Table 14: Stakeholder Model

Stakeholder Model		
Conceptual Design	Business Architecture	Recommended
Summary	Lists, categorizes and describes the types of individuals and organizations that will be affected by the business functions of a solution.	
Aliases	Target Groups, Clients, Client Groups, Organizations, Parties, Roles	
Format	Chart with names and descriptions. Hierarchy (tree) diagram if needed to classify types of Stakeholders.	
Purpose	Ensures awareness of all individuals and organizations that will interact with the solution or be affected by its business functions. Provides a framework to identify the processes and information flows involving each Stakeholder, to ensure the solution is complete.	
Related	Stakeholder interactions are shown on the Context Model. User Roles will include Stakeholders interacting with the solution. If data is needed about a Stakeholder, include it in the Data Models.	

4.11.1. Definition

Stakeholders are defined as anyone that has an interest in business activity being described. This can include customers, clients, suppliers, partners, governing and regulatory bodies, support organizations, and individuals within the organizations in the scope of the effort.

Stakeholders may or may not interact with an automated solution, as users or data-exchange partners.

Stakeholders may be individuals or organizations.

The model should include types of Stakeholders, such as “Program Applicants” or “Hospitals”, to represent many Stakeholders that play a similar role in the architecture.

The Stakeholder Model may name single organizations, such as “Vanderbilt University”, if their role is unique. The model should not be a long list of all organizations involved in the business. Use full names of organizations, not acronyms.

A Stakeholder Model may be displayed as a list, or a cluster of symbols. The Stakeholders may be organized into a hierarchy and displayed with a tree diagram or indented list as seen in *Figure 18: Sample excerpt of the Stakeholder Model.* Hierarchy is particularly useful to show the structure of government organizations.

For each Stakeholder, provide a description specifying who is included in the type, or stating the role of a single organization (as client, supplier, governor, partner, etc.)

4.11.2. Extension: Personas

The Stakeholder Model may be extended by this suggested supplementary model:

- Provide fictional examples of client/customer Stakeholders, to help understand and empathize with their needs and experiences
- Follow experience-design best practices

A Persona is usually presented on one slide, and includes a name, portrait (stock photo), demographic, and psychographic characteristics. Personas also describe life situations that affect the Stakeholder’s interaction with the business services, especially situations that will challenge the solution’s ability to serve all Stakeholders.

4.11.3. Extension: Stakeholder Needs

The Stakeholder Model may be extended by this suggested supplementary model:

- For each type of Stakeholder, list their Needs that are relevant to the Business Goals or intended outcomes of the project
  Examples of Needs: health care, safety, housing, nutrition, etc.
- This may be limited to client and customer Stakeholders
- If Needs are shared by multiple Stakeholders, display as a matrix

The Stakeholder Needs may be used in early consultations to identify the Business Services, or some Requirements.

4.11.4. Sample

Figure 18: Sample excerpt of the Stakeholder Model displays a hierarchy of stakeholders with a tree diagram. This may not represent current TennCare architecture.

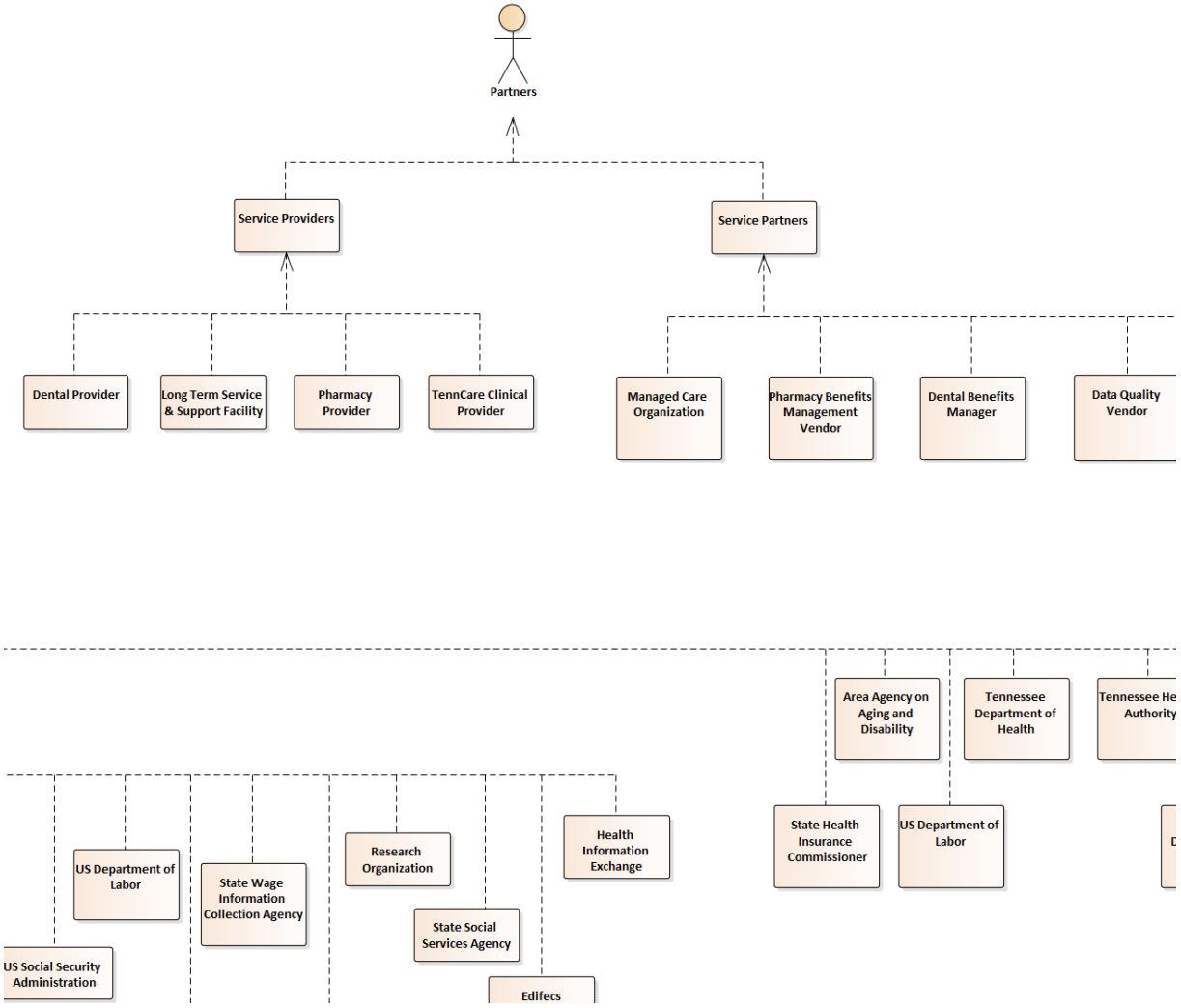

Figure 18: Sample excerpt of the Stakeholder Model

4.12. Business Operating Model

Table 15: Business Operating Model

Business Operating Model		
Conceptual Design	Business Architecture	Suggested
Summary	Shows how the internal divisions of the enterprise are integrated to operate the business, by interacting with each other and external Stakeholders.	
Aliases	Operating View, Business Network Model, Operating Model Backplane	
Format	Diagram	
Purpose	Enables discovery of how the business operates to conduct interactions with Stakeholders and how the operations should change in the target state.	
Related	All Stakeholders should match the Stakeholder Model. External interactions should match the Context Model. Capabilities should match the Capability Model.	

4.12.1. Definition

The Business Operating Model is an integrated business blueprint identifying the internal divisions of the organization and how they interact with external Stakeholders as seen in section 4.7.3.

The Business Operating Model is formed around boxes representing the TennCare enterprise and its major organization units (for the current state) or its capabilities (for the target state). The external Stakeholders are arranged around the enterprise, in the same positions as in the Context Model (see section 4.7).

Arrows indicate the information and resource flows between organizations. Example: “Fee payment”. This model shows high-level (generalized) flows that may be detailed in the Business Scenarios. These information requirements should also be reflected in the Conceptual Data Model.

The Business Operating Model may update and extend the Stakeholder Interactions chart (see section 4.7.3) that accompanies the Context Model.

As an alternative to this artifact, use the Context Model for external flows, the Capability Model for internal structure, and a series of Business Scenarios combining both.

4.12.2. Sample

Figure 19: Sample of the Business Operating Model, which may not represent current TennCare architecture, shows how internal divisions of TennCare interact with external stakeholders.

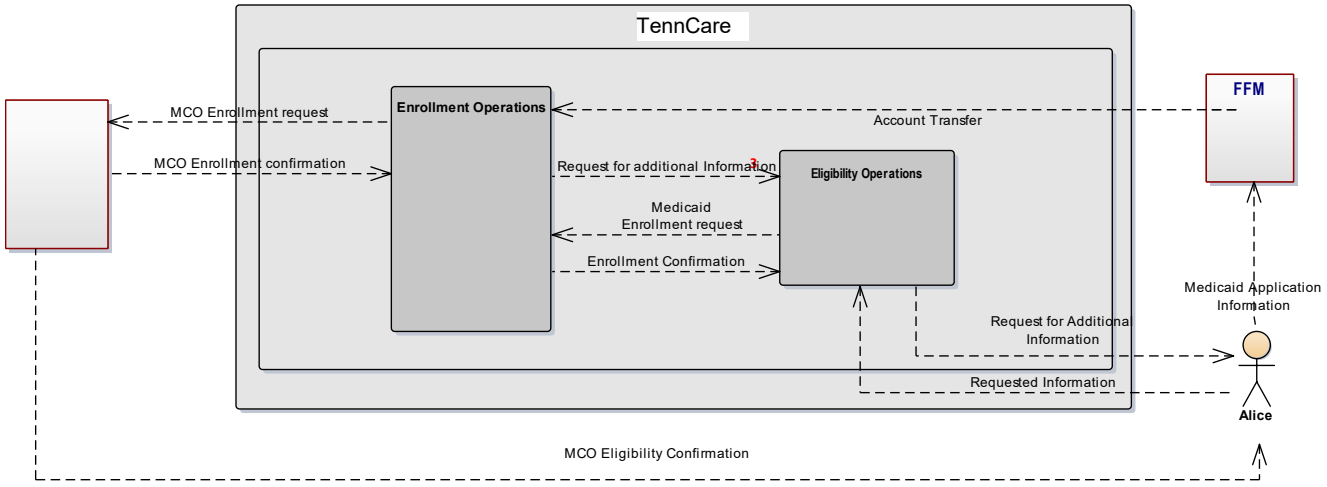

Figure 19: Sample of the Business Operating Model

4.13. Semantic Model

Table 16: Semantic Model

Semantic Model		
Conceptual Design	Business Architecture	Suggested
Summary	A structural diagram used to clarify major business concepts, terms, roles, policies and information requirements.	
Aliases	Semantic Diagram	
Format	UML Class Diagram, simplified	
Purpose	Used during consultations to explore key business concepts and how they relate to each other. May clarify scope, terminology, service outputs, processes, or other business architecture.	
Related	Evolves into the Conceptual Data Model and the Business Operating Model.	

4.13.1. Definition

Semantic models lay out and structure basic business concepts and knowledge, in clear and consistent terms as seen in *Figure 20: Sample of the Semantic Model.* This consultation tool can elicit and clarify Stakeholder roles, information requirements and business

policies.

The Semantic Model may be developed as multiple diagrams, one for each Capability, Function, Service, or other topic to be explored.

Boxes on the diagram (UML Classes) represent any elements of importance to the business, including Stakeholder roles, Service outputs (section 4.5), and information resources.

Use UML class diagram symbols to describe relationships between the elements:

- Association: A direction of flow, indicated by an arrow with a filled arrowhead and an action label.
- Generalization: Classifies elements into a hierarchy of types. Use a line with a hollow arrowhead pointing to the parent (super-type).
- Aggregation: Decomposes an element into its parts. Use a line with a transparent diamond arrowhead pointing to the aggregate element.

4.13.2. Sample

Figure 20: Sample of the Semantic Model, which may not represent current TennCare architecture, shows the stakeholder roles and business policies of a particular service.

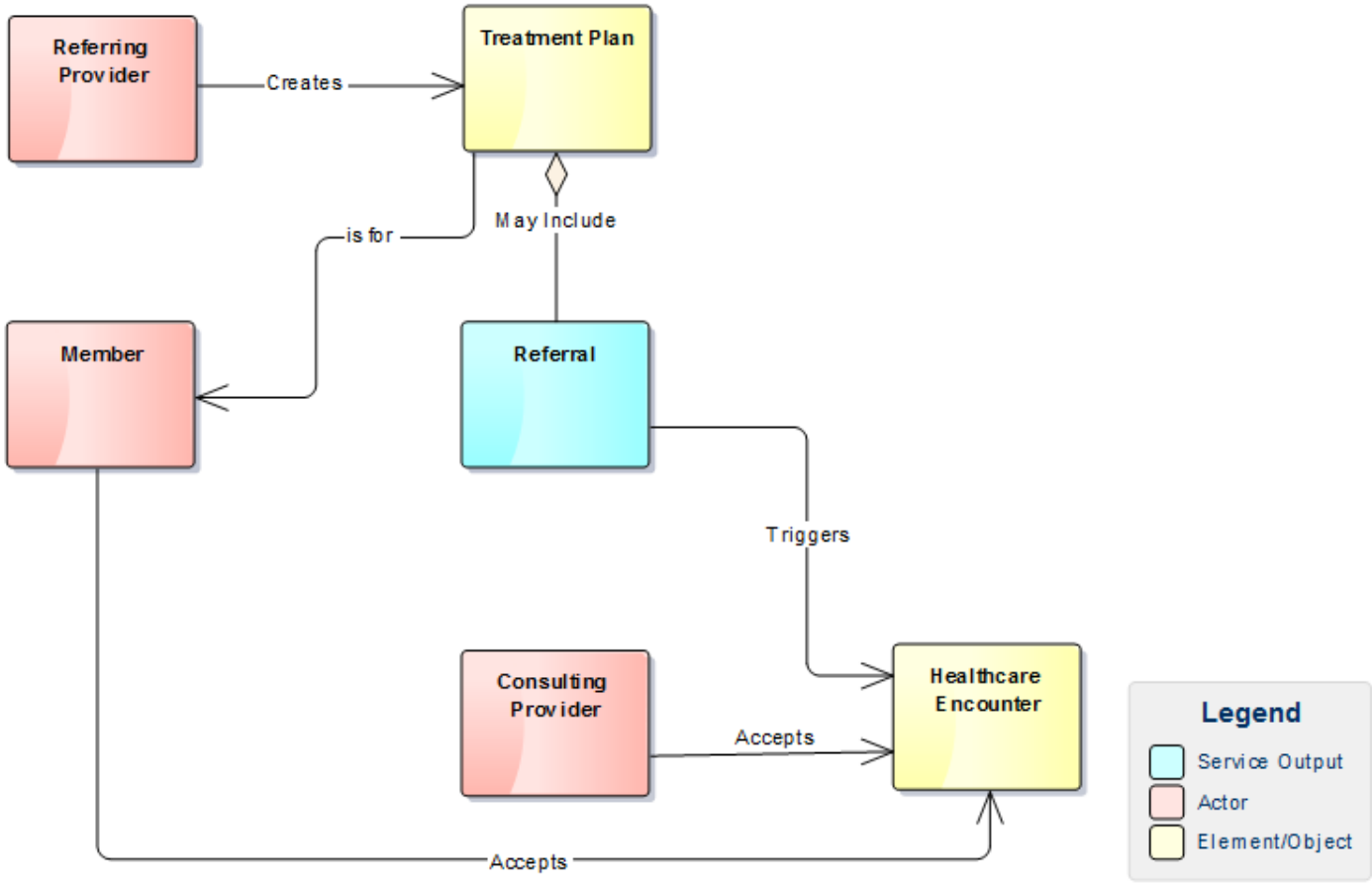

Figure 20: Sample of the Semantic Model

4.14. Business Rules

Business Rules		
Solution Design	Business Architecture	Recommended See section 1.5
Summary	Formal specifications of business policies to be implemented by an automated solution.	
Aliases	N/A	
Format	Flexible, see below	
Purpose	Expresses the enterprise's policies so that they can be implemented by an automated solution, and tested. Ensures that solutions are customized for any rule variations between TennCare and other organizations.	
Related	Rules apply to workflows in the Business Process Model or the System Process Model. Rules may calculate or be calculated by data elements in the most detailed available Data Model.	

4.14.1. Definition

If a solution will implement legislation, regulations, or other business policies, those policies need to be expressed formally as Business Rules as seen in *Figure 21: Sample of a Business Rule*.

A Business Rule is defined as a directive intended to guide business behavior, including constraints (musts) and guidelines. Business rules include computations, logical inferences, timing rules and conditional triggers.

Refer to industry guidance such as the Business Rules Group or *Principles of the Business Rules Approach*, Ron Ross, 2003.

4.14.2. Formats

Specify the rules in a business-readable format appropriate to the solution, such as:

- A business rules engine, containing a database of rules
- Structured charts (see the sample below)
- A spreadsheet of rules
- Chart of parameter values to be configured in COTS or SaaS software (e.g. “maximum days”, “refill limits”)
- Included in other models, such as the System Process Model or Logical Data Model
- Included as Project Requirements (see the TennCare Requirements Management Standard)

4.14.3. Sample

Figure 21: Sample of a Business Rule, which may not represent current TennCare regulations, displays a business rule for determining if an individual meets disability requirements.

Business Rule Name:	Non-Financial, Disability		
Purpose:	Determine if the individual meets the disability requirements		
Policy Section:	Non-Financial Eligibility		
Policy Chapter:	Disability		
Program(s) & Policy #:	ABD	Family Medicaid	CoverKids
	115.015	N/A	N/A
Applies at: IN – Intake CG - Change RN - Renewal	N/A	N/A	N/A

Business Rule(s):

The following rules are applicable to each individual requesting assistance coverage ("Applying for Coverage?" is "Yes" on the Program Request – Individual screen) when being evaluated for ABD categories.

An individual is considered disabled IF:

- The individual is receiving one of the following unearned income types ("Unearned Income Type" on Unearned Income Details screen) as on the authorization date:
  - SSI OR
  - Social Security Disability Benefit (SSDI)

  OR
- The individual is meeting Length of Stay (30 day continuous confinement) non-financial requirement based on *EDBC_NonFinancialEligibility_LengthofStay_ALL*
  OR
- The individual has a PAE – if the "Level of Care" is not "Unknown" and "Verification" is "PAE" AND PAE Status is either "Approve" or "Denied (at risk)" on Nursing Facility/ICF Details screen
  OR
- The individual has an approved PAE – if a valid value is selected for "Level of Care" (other than Unknown) on HCBS Details screen with an ongoing PAE ("PAE Effective Date" is LESS than or EQUAL to authorization date and "PAE End Date" is GREATER than authorization Date) AND PAE Status is either "Approve" or "Denied (at risk)" on HCBS Details screen
- Note: If Level of Care is Unknown and living arrangement is Nursing Facility, HCBS, or Institutionalized Hospice, an indvidual will meet the disability criteria in order to group for Instutional Medicaid Disabled

Figure 21: Sample of a Business Rule

4.15. State Transition Diagram

Table 17: State Transition Plan

State Transition Diagram		
Solution Design	Business Architecture	Suggested See section 1.5
Summary	Shows the states that an object goes through during its lifecycle.	
Aliases	Lifecycle Diagram, Component Lifecycle Diagram, State Transition Model	
Format	UML State diagram	
Purpose	Enables the discovery, analysis and specifications of processes, business rules, and data requirements, especially status codes.	
Related	Each diagram is about one entity in the Conceptual or Logical Data Model. Each state in the diagram could become a status code for the object, shown as an allowed value in the most detailed available Data Model. Each transition in the diagram could be an activity or event in a process of the Business Process Model or System Process Model.	

4.15.1. Definition

A state transition diagram illustrates the business behavior of an object throughout its lifecycle.

This artifact is used in the discovery and analysis phase of designing a custom solution. Make state-transition diagrams about a few objects whose lifecycle or status is important in the solution. These may be Business Service Model outputs and/or Conceptual Data Model entities.

Each node represents a state, or status, that the object may go through during its life. An object has exactly one state at any given time. The diagram has an initial state (solid circle), intermediate states (rounded rectangles), and a final state (solid circle with a ring around it).

The arrows represent the possible transitions between states, which occur during a business or system process. Analyze both the expected and alternate paths to discover success and failure states.

The diagram may include guard(s) indicating exceptions, or conditions which must be met before a state change can occur. *Figure 22: Sample of the State Transition Diagram* illustrates an example, which may not represent current TennCare architecture.

4.15.2. Sample

The following sample may not represent current TennCare architecture.

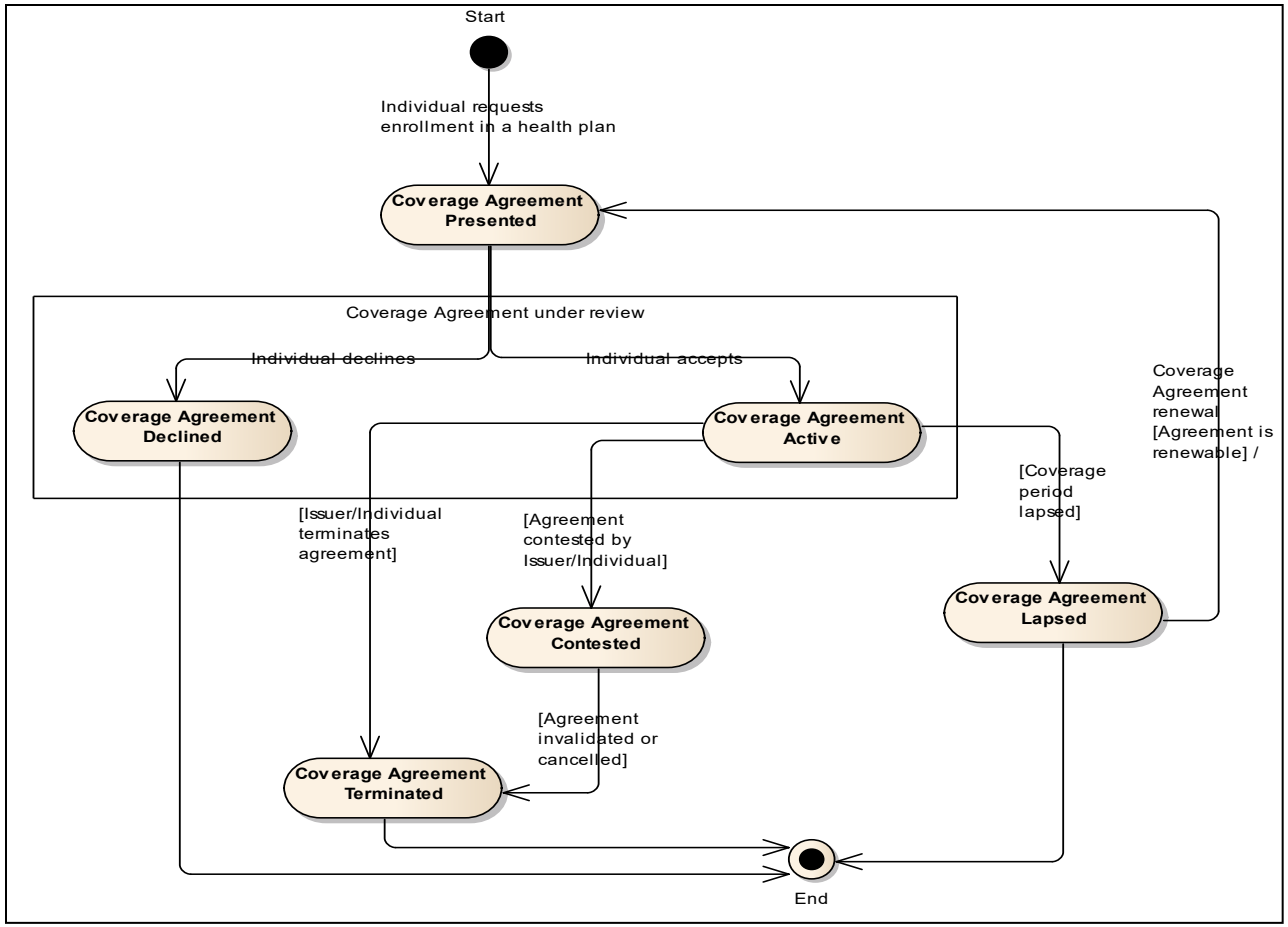

Figure 22: Sample of the State Transition Diagram

5. Data Architecture Artifacts

5.1. Core Artifacts and Relationships: Data

Figure 23: Core Artifacts and Relationships: Data shows the main relationships between the required and recommended Data Architecture artifacts. Artifacts from other domains are shown if they provide input to Data Architecture or need alignment with it.

Blue boxes are Enterprise artifacts, red boxes are Conceptual Design, and orange boxes are Solution Design.

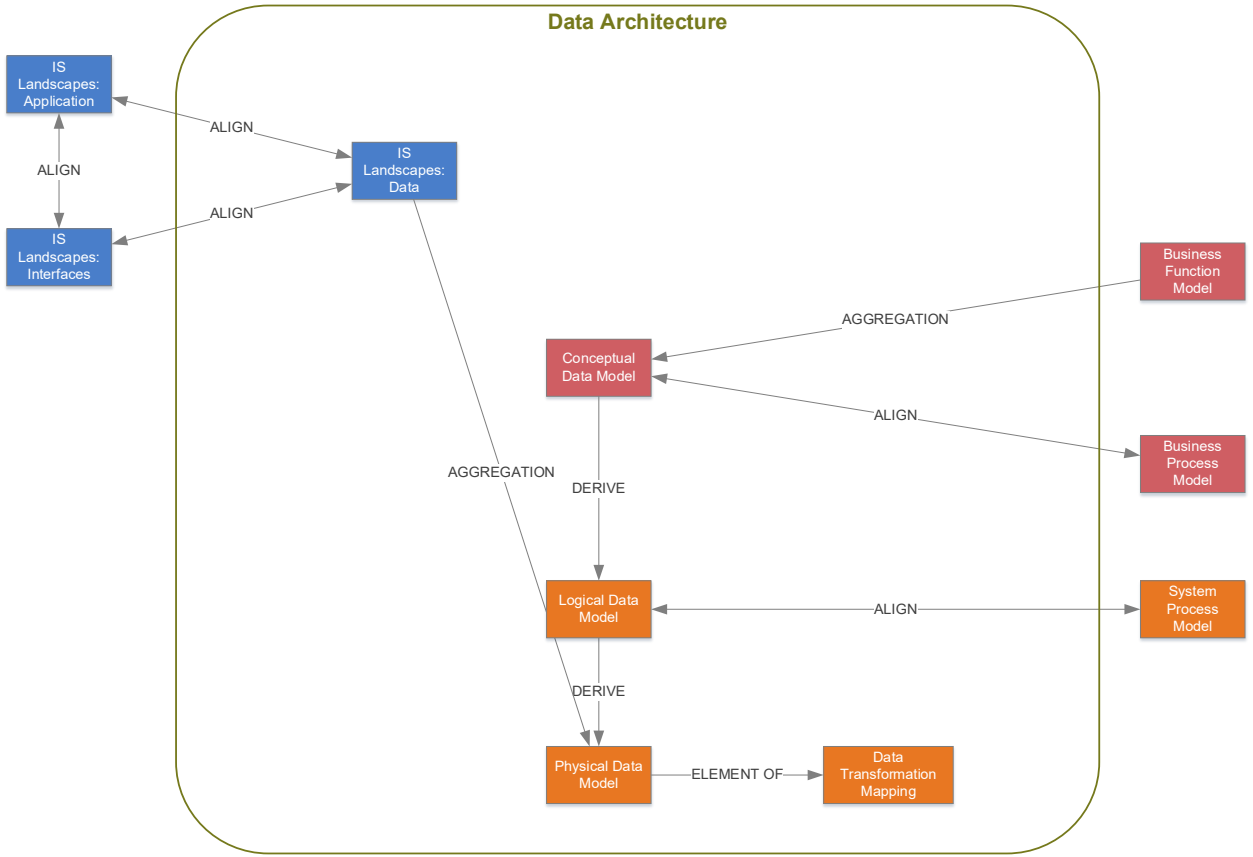

Figure 23: Core Artifacts and Relationships: Data

5.2. Data Landscape

Table 18: Data Landscape

Data Landscape		
Enterprise	Data Architecture	Required
Summary	A visual summary of the databases and information assets that TennCare owns or has access to.	
Aliases	Asset Inventory, Data Catalog	
Format	Poster-sized diagram, showing data assets connected to applications	
Purpose	The Application, Data and Interface Landscapes are a comprehensive reference tool for TennCare IS to: • Set enterprise-wide IS strategy • Identify cost savings opportunities by leveraging existing investments • Manage performance, security, business continuity and disaster recovery	
Related	IT assets are connected and shown on the three aligned Application, Data and Interface Landscapes. The data asset inventory provides details about each database or repository on the landscape. Detailed by the Physical Data Models. Provides a reference list of data assets to be used in many artifacts.	

5.2.1. Description

The Data Landscape is part of the TennCare IS Application, Data and Interface Landscapes. The IS Landscapes visually represent all of the information systems assets of TennCare, and how they are connected. The IS Landscapes are the enterprise architecture blueprint for both the current state and target state (future).

The Data Landscape displays limited attributes, based on a more detailed inventory of TennCare's databases and other information assets. The Data Landscape, including the status of each data asset, shall be updated to reflect changes made at each Architecture Governance Review.

Table 19: Data Landscape Attributes are the minimum attributes required for the Data Landscape. *Figure 24: Sample excerpt from the TennCare Data Landscape* provides a visual depiction of the Data Landscape. While the detailed attributes are not shown on the visual, these attributes should be available on further interrogation of the model.

Table 19: Data Landscape Attributes

Attribute	Description
ID	Unique Identifier for the asset
Name	Name or Acronym for the asset
Description	Description of what the asset is and its purpose

Status	Indicates whether the asset is: Planned In development In production To retire Retired
Capability	Groups each asset as part of a business capability, which may match a MITA function. See section 4.6.
Business Owner	Indicates which organization owns this asset. If TennCare, which organization unit is the business owner.
Type	Transaction Database, Analytical Database, Unstructured, Paper, etc.
Sensitivity	Indicates the types of sensitive data that this asset manages, including Personally Identifiable Information (PII), Personal Health Information (PHI), Federal Tax Information (FTI), proprietary data, etc.

5.2.2. Sample

Figure 24: Sample excerpt from the TennCare Data Landscape provides a visual depiction of the Data Landscape. It is fictionalized and may not represent current TennCare architecture.

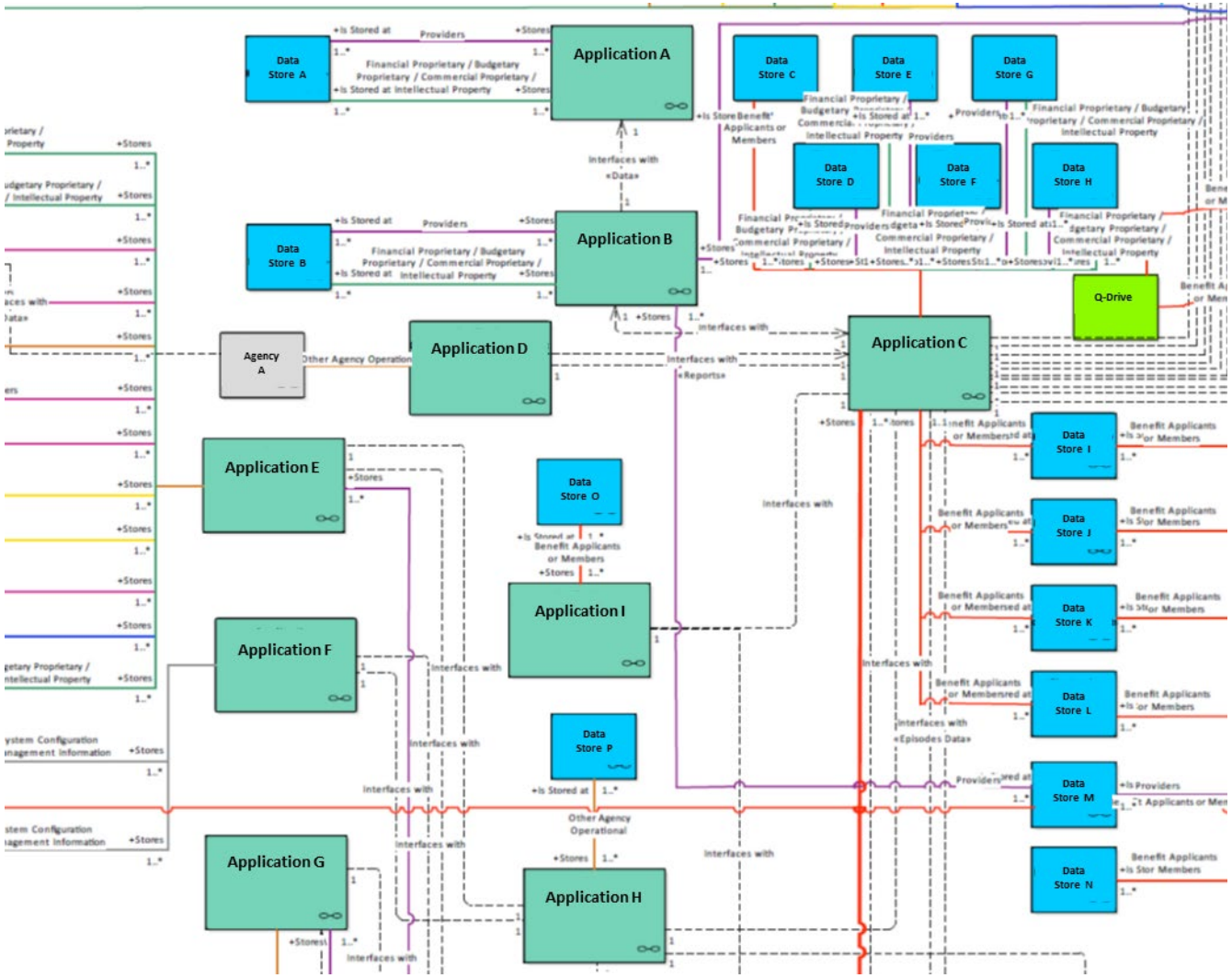

Figure 24: Sample excerpt from the TennCare Data Landscape

5.3. Data Model Artifact Comparison

Table 20: Comparison of Conceptual, Logical, and Physical Data Models summarizes the differences between the Data Model artifacts in the following sections.

Table 20: Comparison of Conceptual, Logical, and Physical Data Models

Artifact Name	Conceptual Data Model	Logical Data Model	Physical Data Model
Arch. Scope	Conceptual Design	Solution Design	Solution Design
Model Scope	Entire solution	One database, or a closely integrated set of databases	One database
Recommended for	Any solution	If TennCare will require light-gray or glass-box insight to the data design as defined in Section 2.3.	Any data that TennCare will have direct (like via a reporting tool or SQL - not via an application interface) access to
Audience	Business and Technical staff who validate the information requirements Architects and analysts who design the solution	Business and Technical staff who validate the information requirements Architects and analysts who design the solution	Technical staff who validate the information requirements Technical personnel who develop and maintain the database, and develop or configure applications to use the data

Artifact Name	Conceptual Data Model	Logical Data Model	Physical Data Model
Notation	Entity-relationship or UML class diagram	Entity-relationship or UML class diagram	Entity-relationship diagram or UML class diagram
Attribution	Representative attributes (optional)	Keys All business attributes	Keys All business and administrative fields
Relationships	Verb phrase Many-to-many OK	Verb phrase(s) Many-to-many resolved Foreign keys	Foreign keys
Generalization	Subtypes generalized	Subtypes generalized	Subtypes resolved
Normalization	Not strictly normalized	Third normal form	Denormalized as appropriate to the implementation
Variations		Logical Dimensional Model (star schema)	Physical Dimensional Model (star schema)
Data dictionary	Entity definitions Data sensitivity	Entity and attribute metadata as listed in section 5.5.2	Table and field metadata as listed in section 1.12

Artifact Name	Conceptual Data Model	Logical Data Model	Physical Data Model
If TennCare approves database design			Specifications listed in section 1.12

5.4. Conceptual Data Model

Table 21: Conceptual Data Model

Conceptual Data Model		
Conceptual Design	Data Architecture	Required
Summary	A summary of information required in a solution, with concepts and relationships defined in a formal structure.	
Aliases	CDM, Subject Area Model, Domain Model TOGAF artifact: Conceptual Data Diagram	
Format	Diagram (entity-relationship notation or UML class diagram), plus data dictionary.	
Purpose	Shows the scope of information required in a solution and how it will be organized and related. Clarifies business terms and concepts. A basis for designing databases and for specifying which data is used in business processes and application components.	
Related	Transformed into the Logical Data Model and Physical Data Model. The data entities should be grouped into subject areas based on the high-level functions in the Business Function Model. See also the TennCare Data Policies and Standards.	

5.4.1. Definition

A Conceptual Data Model organizes information requirements into a structure of entities and relationships as seen in *Figure 25: Sample of Conceptual Data Model in UML notation*. The CDM defines key concepts in terms familiar to business personnel.

When constructing a CDM:

The CDM should cover all information in scope for the solution. This may be implemented in one or multiple databases. (A Conceptual Data Model may also be created for the entire enterprise.)

Provide a one-page summary diagram. Large CDMs may also need to be displayed with multiple detail diagrams. Prioritize readability.

Entities should be grouped into subject areas based on the high-level functions in the Business Function Model.

Subject Area names should use a plural form and Entity names should use a singular form. Normally, the model should include either Subject Areas or Entities but not both. A CDM is best presented as a Subject Area model. When Entities are depicted, the model should include all respective Entities with the exception that many to many relationships may be used that will result in additional entities in the Logical Model.

The CDM does not need to be strictly normalized. Use normalization principles as a guide to group information into meaningful entities.

Generalization is encouraged for clarity, but subtypes do not need to be exhaustive. Avoid generalization that is too abstract for business understanding.

The CDM may optionally include attributes for various reasons:

- To make an entity's meaning clear, model a few representative attributes
- To analyze whether an acquired solution will meet business requirements, model the business attributes needed
- To show integration across databases, model the important unique ID attributes

Relationship names (verb phrases) are recommended. Relationship cardinality should be shown: one-to-one, one-to-many, or many-to-many relationships are all acceptable. Relationship optionality need not be shown; that detail may make the notation less business-friendly.

The diagram should be in a well-known Entity Relationship notation, such as Information Engineering or Oracle/Barker, which use the familiar “crow’s feet” relationship line ends. A UML Class diagram is also acceptable.

A data dictionary must be provided, containing:

- Subject Area/Entity name
- Subject Area/Entity definition
- Data sensitivity (e.g. the entity contains Personal Health Information)
- Any representative attributes that have been named

5.4.2. Sample

Figure 25: Sample of Conceptual Data Model in UML notation organizes information requirements into a structure of entities and relationships. It may not represent current TennCare architecture.

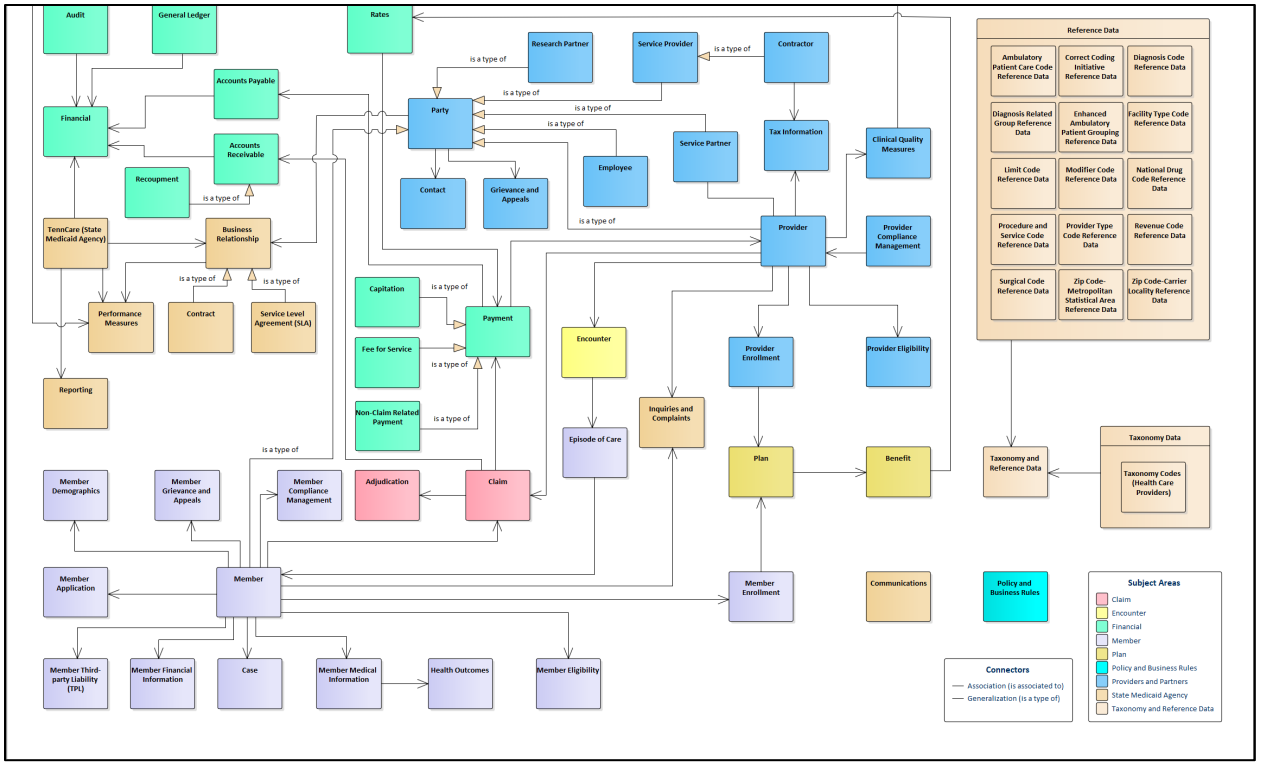

Figure 25: Sample of Conceptual Data Model in UML notation

5.5. Logical Data Model

Table 22: Logical Data Model

Logical Data Model		
Solution Design	Data Architecture	Recommended See section 1.5
Summary	A comprehensive and formal specification of the data required in a database, as normalized entities, attributes and relationships.	
Aliases	TOGAF artifact: Logical Data Model and Dictionary	
Format	Diagrams (entity-relationship notation or UML class diagram), plus data dictionary.	
Purpose	Shows the data elements required or provided in a solution and how they will be organized and related. A basis for designing databases and for specifying which data is used in business processes and application components.	
Related	Derived from the Conceptual Data Model. Transformed into the Physical Data Model. See also the TennCare Data Policies and Standards.	

5.5.1. Definition

A Logical Data Model (LDM) organizes the information required in a database into a structure of entities, attributes and relationships. The LDM translates business terms and concepts from the Conceptual Data Model into a comprehensive data structure that can be implemented using a custom-developed database or a packaged solution.

The scope of an LDM is usually one database. A closely integrated set of databases may be covered by one LDM.

Developing the LDM is an important stage in designing a new database, as it is more complete and rigorously structured than the Conceptual Data Model. If TennCare is obtaining access to an existing database, the Physical Data Model is more relevant than the LDM. The business meaning of the data fields must be available from a logical or physical data dictionary.

The LDM is technology-neutral, meaning it does not reflect constraints that are specific to a data structure or database engine, which are introduced in the Physical Data Model.

5.5.1.1. Traceability

LDM entities should be grouped into the same subject areas as in the Conceptual Data Model, based on the high-level functions in the Business Function Model.

One CDM entity will often be transformed into multiple LDM entities when the attributes are detailed and normalized.

5.5.1.2. Attributes

The LDM must be fully attributed, meaning:

- The model includes all attributes required by the business, including data to be collected, acquired, consumed, produced, or published by this database
- Administrative attributes, such as “last update date”, may be included
- Each LDM entity must have a unique identifier (a primary key)
- Foreign key attributes reflect the relationships

5.5.1.3. Naming

Entities and attributes should be named with singular nouns.

Longer names are acceptable for clarity. In the LDM, use only acronyms recognized by the business.

Begin all attribute names with the entity name; this helps to distinguish similarly-named attributes, such as Request Date and Decision Date.

End attribute names with the attribute type, such as ID, Name, Date, Code, Number, or Description.

5.5.1.4. Relationships and Normalization

The following practices ensure that the data requirements are well-understood, regardless of what technology and structure will be used to manage the data.

Relationships must be diagrammed. Relationships must be named by verb phrases, preferably in both directions. This ensures that the business meaning of the relationship is clear.

Relationship cardinality and optionality must be shown.

Many-to-many relationships must be resolved.

Generalization is encouraged, to the level of abstraction that is reasonable to implement in the database. All subtypes expected in business operations should be modelled.

In particular, generalize all individuals and organizations into a Party Role structure, to ensure consistent storage of names and addresses.

The LDM should be normalized, to organize the data without redundancy. Use third normal form, or a method appropriate to the intended data structure.

LDM attributes should be atomic, containing only one fact or piece of data that cannot be divided into smaller pieces.

5.5.1.5. Diagrams and Notation

Display the LDM on multiple diagrams as required, showing all entities, attributes and relationships.

Provide a one-page summary, which may show only core entities and relationships, without attributes or details.

The LDM diagram should be in a well-known Entity Relationship notation, such as Information Engineering or Oracle/Barker, which use the familiar “crow’s feet” relationship line ends. A UML Class diagram is also acceptable.

5.5.2. Logical Data Dictionary

With the LDM, a data dictionary must be provided, at a minimum containing:

- Entity name
- Entity definition
- Logical Entity mapping to Conceptual Data Model entity (traceability)
- Attribute name
- Attribute definition
- Attribute datatype and length
- Attribute allowable values (e.g. codes and their meanings)
- Attribute data sensitivity (e.g. contains Personal Health Information)
- Attribute Primary Key, Foreign Key, Business Key indicators
- Attribute optionality

The above details may or may not be displayed on the LDM diagram.

All of these properties of the data entities and attributes should be stored in a modeling tool (see section 2.7), so that a data dictionary can be automatically extracted.

Follow industry norms for writing informative definitions. The attribute definition should include any derivations or calculations. Also indicate any attributes that are unique identifiers in other databases, to ensure alignment for integration.

Refer to the TennCare Data Policies and Standards, Metadata Management chapter, for additional elements to specify in a metadata repository.

5.5.3. Variation: Logical Dimensional Model

If modeling an analytical database (data warehouse, etc.), it may be appropriate to use a “star schema” of fact and dimension entities, instead of a normalized relational model. Follow the above guidance for naming, notation and data dictionary.

5.5.4. Sample

Figure 26: Sample of the Logical Data Model shows a suitable entity-relationship diagram format. This small, fictional example does not represent current TennCare architecture. Within each entity (yellow box), the attributes are listed with their data type and length,

and key indicators (PK = primary key, FK = foreign key). The relationships are in a crow's foot notation and have verb phrases in both directions.

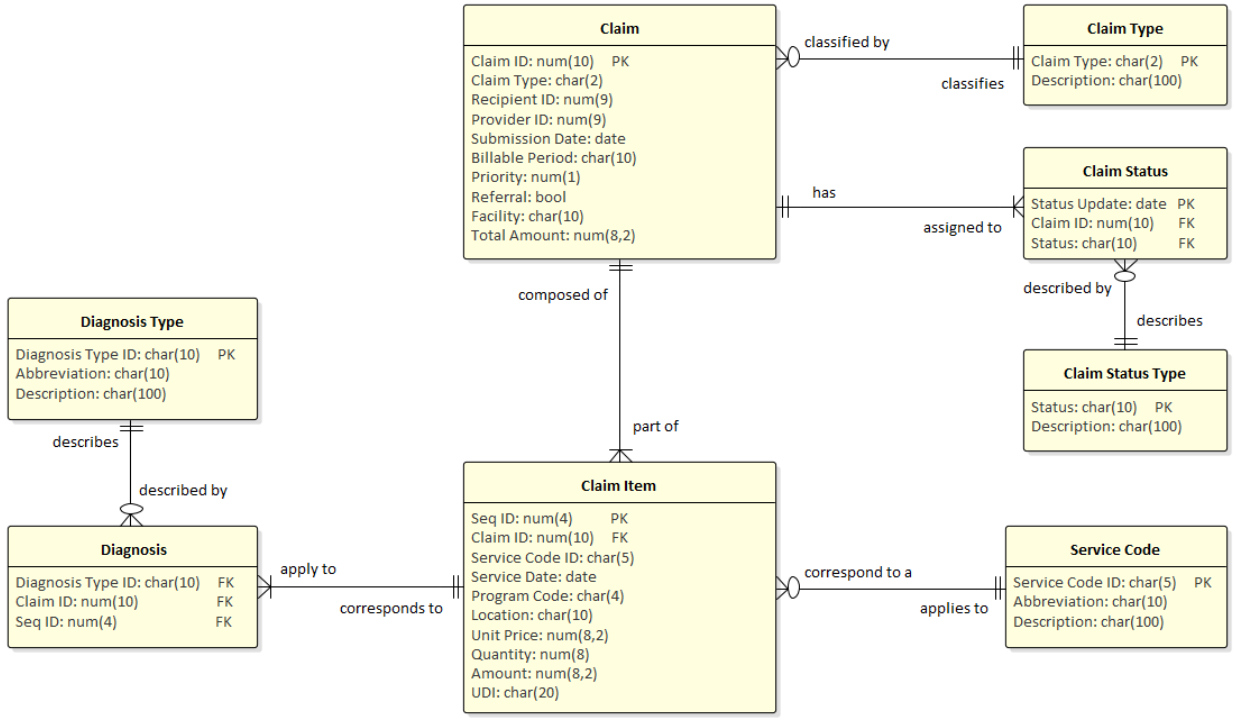

Figure 26: Sample of the Logical Data Model

5.6. Physical Data Model

Table 23: Physical Data Model

Physical Data Model		
Solution Design	Data Architecture	Recommended See section 1.5
Summary	The physical layout of data in a database, as tables, fields (columns) and relationships (foreign keys). Accompanied by definitions and specifications for the database, tables, fields, domains and indexes.	
Aliases	Data Layout TOGAF artifact: Physical Data Model and Dictionary	
Format	Diagrams (entity-relationship notation) plus Data Dictionary charts plus Specification charts	
Purpose	Documents the design of a database. The diagrams, data dictionary, and specifications are used to develop (create) and manage the database. The diagrams and data dictionary are needed when extracting, converting, or analyzing the data, to access and interpret the data. Describes the databases named in any application, technology, or security architecture artifacts.	

Related	Derived from the Logical Data Model. Volumetrics are derived from the Volume and Performance Expectations. Used to update the Data Landscape and the data inventory. See also the TennCare Data Policies and Standards.

5.6.1. Definition

A Physical Data Model (PDM) documents the information required in a relational database, as a structure of tables, fields (columns) and relationships. At the time of publication (2020), physical data models are not typically used to document or design non-relational databases and so therefore this definition only applies to relational databases.

The PDM is technology-specific, meaning it is structured for performance and other practical considerations on a given database engine.

The PDM artifact has four parts:

- Database Descriptors
- Diagrams
- Data Dictionary
- Specifications, such as indexes and partitioning. These are only required for custom-developed databases, not for externally-managed databases.

5.6.1.1. Usage for Custom-Developed Databases

When custom-designing a database, the PDM translates business terms and concepts from the Logical Data Model into a database structure with the technical details. The PDM should be similar to the LDM but adapted for the selected database engine or other technological constraints.

The full PDM, including diagrams, data dictionary, database descriptors and specifications, is needed for the Solution Design Review and for TennCare to manage the implemented database.

5.6.1.2. Usage for Pre-Packaged Databases

With managed services and packaged solutions, TennCare may receive or have access to a database designed by a vendor for use in multiple organizations.

In these situations, the Solution Design Review needs the PDM diagrams and data dictionary to review and approve the solution architecture.

After implementation, TennCare needs the PDM diagrams and data dictionary to read and interpret this data. It is important for this PDM to clearly show the business meaning and sensitivity of the data because this PDM may not closely resemble the Conceptual Data Model and there may not be a Logical Data Model.

The level of insight required into a COTS, SaaS, or vendor-managed solution is dependent on the contractual arrangement. If a database is managed by a vendor or third party, TennCare may not need specifications to manage performance. This would require some discussion between the vendor and TennCare IT to determine if the vendor solution is a ‘gray/black box’ for which TennCare may not need any deeper level of management knowledge and/or control. A common related example is software solutions with ‘embedded’ databases which are managed as part of the software application and do not typically have separate database management activities.

For purposes of TennCare estimating the costs of database management, volumetrics are included in the Database Descriptors section of 5.6.2.

5.6.1.3. Data Elements

The PDM includes tables and fields containing all data elements needed for business transactions and/or analysis, plus the administrative data elements.

Table 24: Pre-Packaged Databases Data Elements

Section of PDM	Required for	Object	Property of Object
Data elements	All databases	Tables and fields	• Tracking history, such as update dates • Looking up codes and allowed values • Security access control • Audit • Privacy • Storage management • Archiving (records management)

5.6.1.4. Identifiers

Each PDM table must have a unique identifier (a primary key). Follow the industry best practices applicable to the technology, such as using a surrogate key (a meaningless generated number).

Table 25: Pre-Packaged Databases Identifiers

Section of PDM	Required for	Object	Property of Object
Identifiers	All databases	Table	Unique identifier (primary key)

5.6.1.5. Naming

The table, field and other names in the PDM must exactly match the names in the database it describes, and should reflect the meaning of the element.

If the database is custom-designed for TennCare, physical names should use the approved abbreviated terms and construction rules defined in CMS's guides:

- DM OP-045 Operating Procedure for Constructing Physical Table and File Names
- DM OP-046 Operating Procedure for Constructing Physical Column or Element Names
- CMS Standard Terms List and Abbreviation List
- CMS Data Management Operating Procedures and Guidelines.
- Denormalization and Redundancy

5.6.1.6. Denormalization and Redundancy

The PDM must be structured so it is possible to implement it in the chosen technology. For relational databases, this means resolving subtypes and denormalizing the Logical Data Model as required for performance. Note and manage any redundancy that is introduced by denormalization.

5.6.2. Sections and Necessary Properties

Table 26: Sections and Necessary Properties of the Physical Data Model

Section of PDM	Usage	Required for	Object	Property of Object
Database Descriptors	The following information about the whole database relates it to the technology architecture, and is used to update the Data Landscape. Specify the descriptors in the "Property of Object" column for each database in the solution, whether managed by TennCare, a vendor or a third party. Additional details on data ownership roles, owners, and accountability can be found in the TennCare Data Governance Organization Charter.	All databases	Database	• Name • Description • Data owner (organization name) • Database manager (organization name) • Types of sensitive data (PHI, PII, FTI, etc., or "none") • Schema name in each environment • Volumetrics required for costing (e.g. initial total data size, annual growth rate)
		Glass box only (as described in section 2.3.1)	Database	• Hosting provider • Location: IP address (Internet Protocol), URL (/Uniform Resource Locator), (IP/URL) or File name and path • Data structure (relational, XML, columnar, etc.) • Database engine (e.g. Oracle)
Diagrams	The PDM may be displayed on one page or on multiple diagrams, according to its complexity. The PDM diagram should be in a Entity Relationship notation, such as Information Engineering or Oracle/Barker, which use the familiar "crow's feet" relationship line ends. A	All databases	Diagram	• Table names • Field (column) names • Indication of mandatory and optional fields • Indication of primary key fields • Relationship lines between tables, with cardinality and optionality

	UML Class diagram is also an acceptable notation. The diagram may include other metadata listed below, repeating properties from the Data Dictionary or the Specifications. For example, datatype and length may be shown on the diagram. If the PDM is generated by reverse engineering from the database, ensure that it is human-readable. Relationship lines may need to be added to the diagram if the database did not have referential integrity.			• Indication of foreign key relationships (shown as migrated keys in the field lists, or on the relationship line, or both) • Indication of tables that are inactive or not used for TennCare • A notation legend, or reference to a standard notation • Other metadata (repeated from the Data Dictionary or Specifications)
Physical Data Dictionary	Refer to the TennCare Data Policies and Standards, Metadata Management chapter, for additional properties to specify in a metadata repository. All of these properties should be stored in a modeling tool (see section 2.7) with the capability of automatically extracting a data dictionary. If there is a Logical Data Dictionary that provides the business meaning of attributes, and the physical fields are traceable to the logical attributes, then the physical data dictionary does not need to include field definitions. The other properties listed above are still required to access and manage the	All databases	Table	• Name • Definition • Security Classification • Archiving / purge settings • Volumetrics (initial or current size, and annual growth rate) • Authoritative source table(s) or file(s) • Name of the entity in the Logical Data Model or Conceptual Data Model that is implemented by this table (traceability)
		All databases	Field	• Name • Definition, including: ○ Business meaning ○ Derivations and/or

	data. If a database includes tables that are inactive or not used for TennCare, do not detail them in the Data Dictionary. To prevent confusion, provide a list of the inactive table names that will be visible to TennCare. If the PDM is reverse-engineered from the database, ensure that the data dictionary includes all the metadata listed above. For example, the definition of fields may need to be added so the data can be interpreted by business users.			calculations ○ Code value meanings ○ Indication if this field is a key in another database, requiring integration • Primary Key flag • Foreign Key flag • Authoritative source field(s) or file(s) • Optionality (Null/not null rule) • Data Type (using the datatypes provided by the database engine) • Length (where applicable) • Precision (where applicable) • Data sensitivity (e.g. contains Personal Health Information) • Name of the attribute in the Logical Data Model that is implemented by this field (traceability), if available
Database Specifications	The following physical data specifications are needed if TennCare will be developing or managing the database; see section 1.1. These specifications for relational databases may be adapted for other technologies.	Glass box only	Database	• Database hosting location • Database partitions
		Glass box only	Varies	• Access Rights • At the database, table, field or row level, as required • Other security requirements • Volumetrics • Current size & growth rate for: Database, Tables, Rows, Indexes

		Glass box only	Table	• Table alias • Table partitions • Managed redundancy • Referential integrity constraints • History of changes to table or fields
		Glass box only	Views	• Source table(s) and fields • Selection criteria • Purpose (access security, performance, etc.)
		Glass box only	Stored Procedures	• Name • Description
		Glass box only	Domains	• Data Type, Length and Precision • Allowable Values • Default Values • Edit Mask (e.g. date or area code)
		Glass box only	Indexes	• Index identification • Type of Key identification • Field Level Constraints

5.6.3. Sample

Figure 27: Sample of the Physical Data Model provides an example of a PDM diagram, which does not represent current TennCare architecture. Within each table (yellow box), the fields are listed with their data type and length in green. P indicates a primary key and F indicates a foreign key. The diagram is in one of many acceptable variations on entity-relationship notation.

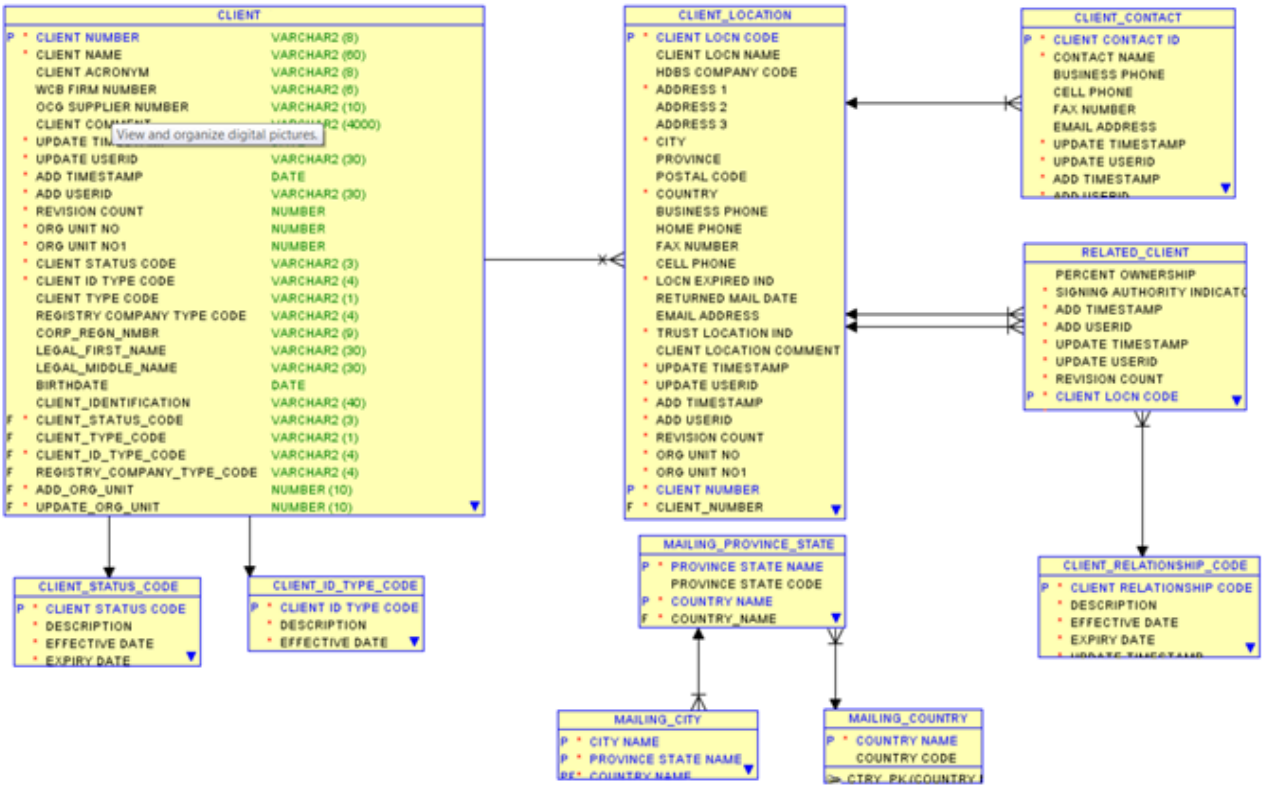

Figure 27: Sample of the Physical Data Model

5.7. Data Transformation Mapping

Table 27: Data Transformation Mapping

Data Transformation Mapping		
Solution Design	Data Architecture	Recommended See section 1.5
Summary	A mapping of data elements from a source database to a destination database, either for a one-time conversion or recurring transformation.	
Aliases	Crosswalk, ETL Mapping	
Format	Chart, in spreadsheet or Extract-Transform-Load tool	
Purpose	Ensures that data quality is preserved or improved through migrating data from a legacy system to a new solution, or converting from a transactional database to an analytical system (data warehouse, etc.)	
Related	Use the data elements (table names, field names, and codes) from the Physical Data Models of the source and destination databases. Instructions for data quality checks and cleansing should be aligned with this mapping.	

5.7.1. Definition

This artifact is needed for any situation where data will be converted from a source database to a destination database. This includes replacing legacy systems and developing analytical systems (data warehouses).

This mapping may provide instructions for a one-time migration and/or an ongoing conversion (which may happen as transactions happen in real-time, or in regularly scheduled batches).

The Data Transformation Mapping specifies:

- Scope of source data to select
  - Source database name
  - Table names

    - Field (column) names
    - Row selection criteria
    - Sensitivity of data elements, to be protected during transformation
- Destination database name
- Mapping for each field to be converted as seen in *Figure 28: Sample of the Data Transformation Mapping*:
    - Source table name
    - Source field name
    - Primary keys that must match, if migrated data will be added to existing records
    - Destination table name
    - Destination field name
    - Transformation rule, if the data will be changed
- Mapping of old codes to new codes, for each field where data values will be changed as seen in Figure 29: Sample of the Data Transformation Mapping Code Values:
    - Context:
        - Source table name
        - Source field name
        - Destination table name
        - Destination field name
    - Source code value
    - Source code meaning
    - Destination code value
    - Destination code meaning
    - Transformation rule, if needed
- Types of Data Conversion as seen in Figure 30: Sample of the Data Conversion Types:
    - Conversion Type
    - Description

5.7.2. Sample

The following samples may not represent current TennCare architecture.

Mapping Fields:

Target					Source				Design		
Target Table Name	Field Name	Field Data Type	Field Size	Nullable (Y/N)	File/Table Name	Field Name	Field Format	Field Size	Convert Type	Transformation/Validation Rule(s)	Business Logic
CASE_CROSSREF	HH_NUM	VARCHAR2	10	N	PERSON_HOUSE	HOUSE_NUM	NUMBER	10	As-Is	Transform the household number from NUMBER to VARCHAR2 format.	The household number exists on the H_PERSON table for each person in the same household.
CASE_CROSSREF	CONVERT_DATE	DATE	7	Y					Default	Default to conversion date	
CASE_CROSSREF	CONVERT_TYPE	VARCHAR2	20	Y					Not Converting	Set value to null	
CASE_CROSSREF	ALLO_NUM	VARCHAR2	10	N	NUM_ALLOCATION	ALLOC_NUM	NUMBER	8	Select Criteria	This value will be retrieved from NUM_ALLOCATION.ALLOC_NUM and transform it from number format to varchar format.	
CASE_CROSSREF	XYZ_CASEID	NUMBER	38	N				9	Generated	Unique number generated from sequence in Application XYZ.	
CASE_CROSSREF	PROGRAM_NAME	VARCHAR2	20	N					Default	Set value to 'HealthProgress'	
AUTH_REPN	ISPARENT_SW	CHAR	1	Y					Default	Set value 'Y'	Column added as per Project XYZ-8565.
AUTH_REPN	EXPIRE_DATE	DATE	7	N					Default	Set to high date 12/31/2999	If we have multiple authorized representatives on a case, we have to create only one authorized representative.

Figure 28: Sample of the Data Transformation Mapping

Mapping Code Values:

File/Table Name	Field Name	Code Value	Code Description	Table Name	Code Table Value	Code Table Description	Default Value	Transformation/Validation Rule(s)
Table_A	PRIMARYID	3	English	FIRSTLANGUAGE	5	English		
Table_A	PRIMARYID	6	Albanian	FIRSTLANGUAGE	19	Other		
Table_A	PRIMARYID	7	Amharic	FIRSTLANGUAGE	6	Amharic		
Table_A	PRIMARYID	5	Arabic	FIRSTLANGUAGE	2	Arabic		
Table_A	PRIMARYID	1	Armenian	FIRSTLANGUAGE	19	Other		

Figure 29: Sample of the Data Transformation Mapping Code Values

Conversion Types:

Convert Type	Field Nullable?	Description
As-Is	Y	As-Is is used to identify source data that is converted exactly as it exists in the source system, without transformation. A nullable indicator identifies whether the database enforces a column to have a value or if it may be left blank. The N/A value is used to indicate fields that are not being converted and therefore are not applicable.
Convert Picklist	N	Convert Picklist is used to indicate how source code values are mapped to code values. The description of the field in the Mapping tab identifies the name of the reference table and mapping code value pairs listed on the Mapping Code Values tab.
Default	N/A	Default is used to indicate a specific value that is set for all converted transactions of that record type. No transformation logic is performed.
Not Converting	TBD	Not Converting is used to identify fields in the data model that are not populated via the data conversion processes.
Generated		Generated is used to identify fields that utilize unique identifiers generated from database sequences. These database sequences are necessary to maintain database integrity and relationships between database objects.
Select Criteria		Select Criteria is used to describe logic to look up a value from another table. This is used when one field should contain the same value or transformation result as another field in the data model and it is better to look up the result than to reapply the data transformation logic.
Transformation		Transformation is used to describe the business logic to convert a value from the source into a value in the target field. Transformation is used for changes in field data types and specialty logic to derive the appropriate value for the field.
N/A		Stands for Not Applicable. The N/A value is used to indicate fields that are not being converted and therefore are not applicable.
TBD		Stands for To Be Determined. This field is used during the design process to identify fields that require further discussion or clarification.

Figure 30: Sample of the Data Conversion Types

5.8. Report Specification

Table 28: Report Specification

Report Specification		
Solution Design	Data Architecture	Suggested See section 1.5
Summary	Lists the reports to be developed. For each report, shows what data it will include, and how it will be displayed with totals, calculations, graphs and layout.	
Aliases	Report inventory, Report design, Report mock-ups	
Format	Chart of reports (spreadsheet) plus Layouts (mock-up and specifications) for each report	
Purpose	Ensures that the solution can provide the reports needed by business users.	
Related	Specify the source data to be reported, using the Data Landscape and the most detailed available Data Models.	

5.8.1. Usage Situations

Report Specifications should be created in any of the following circumstances:

- Reporting is a core function of the solution
- Mocking up a few reports will elicit user feedback to refine the design
- Users require analyst assistance to develop complex reports
- Software has an inflexible or limited reporting component, which needs to be evaluated to determine if it will meet user requirements

Some solutions include a flexible reporting component that is easily customized to output any data that is likely to be needed. Instead of making Report Specifications, it may be feasible to iteratively develop prototype reports using the solution component.

Report Specifications may be also used during Conceptual Design to determine data requirements. Gather samples of existing reports needed, or make mockups of needed reports.

Analyze the data on these reports to develop the Conceptual Data Model. For this purpose, the calculations, graphs and layouts of the reports are not important.

5.8.2. Definition

This artifact should specify:

- Report name
- Report purpose, requestor or audience
- Report frequency, if not on-demand
- Source database(s)
- Source tables and fields
- Source data selection criteria
- Report layout (which data to put in rows, columns, groupings)
- Aggregation (such as sum total row or group-by fields)
- Graph type and layout specifications
- Parameters the user can specify
- Sensitivity of data to be protected

These specifications may be shown by any combination of:

- Spreadsheet listing of all reports
- Formally structured specifications for each report
- Mock-up of each report, with requirements indicated
- Sample of existing report, marked up with required changes

6. Application Architecture Artifacts

6.1. Core Artifacts and Relationships: Application

Figure 31: Core Artifacts and Relationships: Application shows the main relationships between the required and recommended Application Architecture artifacts. Artifacts from other domains are shown if they provide input to Application Architecture or need alignment with it.

The Conceptual Integration Architecture diagram (section 7.4.2) includes both application and technology architecture.

Blue boxes are Enterprise artifacts, red boxes are Conceptual Design, and orange boxes are Solution Design.

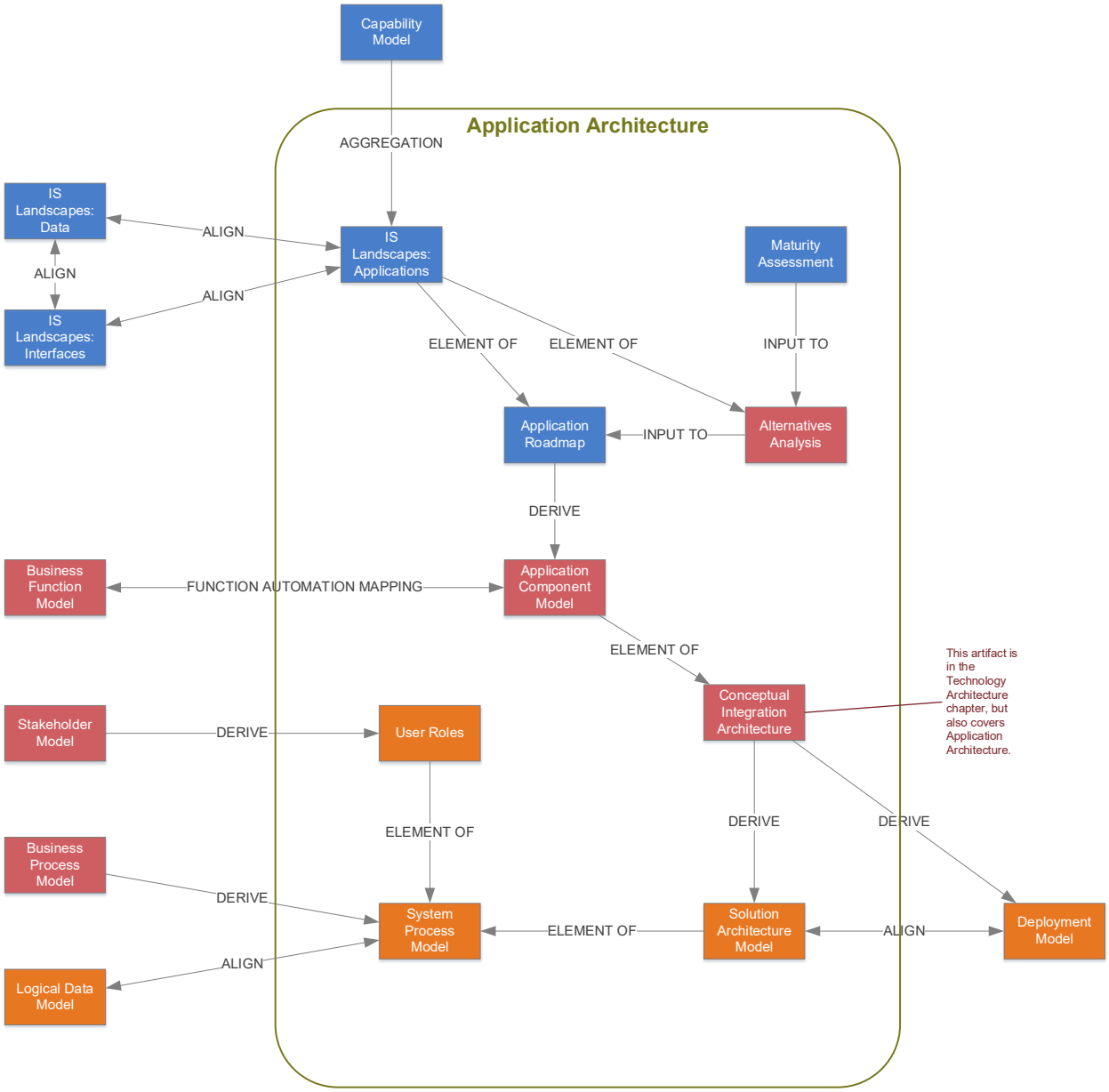

Figure 31: Core Artifacts and Relationships: Application

6.2. Application Landscape

Table 29: Application Landscape

Application Landscape		
Enterprise	Application Architecture	Required
Summary	A visual summary of the applications that TennCare owns or has access to.	
Aliases	Application Portfolio, Application Blueprint, Asset Inventory	
Format	Poster-sized diagram, showing applications, grouped and linked.	
Purpose	The Application, Data and Interface Landscapes are a comprehensive reference tool for TennCare IS to: • Set enterprise-wide IS strategy • Identify cost savings opportunities by leveraging existing investments • Manage performance, security, business continuity and disaster recovery	
Related	IT assets are connected and shown on the three aligned Application, Data, and Interface Landscapes. The IT asset inventory provides details about each application on the landscape. Provides a reference list of application assets to be used in many artifacts. The Capability Model is used to group applications on the Landscape. The Application Roadmap includes applications planned, in development, and to be retired.	

6.2.1. Description

The Application Landscape is part of the TennCare IS Application, Data and Interface Landscapes. The IS Landscapes visually represent all of the information systems assets of TennCare, and how they are connected. The IS Landscapes are the enterprise architecture blueprint for both the current state and target state (future).

The Application Landscape displays limited attributes, based on a more detailed inventory of TennCare's databases and other information assets. The Application Landscape, including the status of each application asset, shall be updated at each Architecture Governance Review, as described in the TennCare Enterprise Architecture Framework Standard.

Table 30: Application Landscape Attributes provides the minimum attributes required for the Application Landscape.

Table 30: Application Landscape Attributes

Attribute	Description
ID	Unique Identifier for the application
Name	Name or Acronym for the application
Description	Description of what the application is and its purpose
Status	Indicates whether the application is: Planned In development In production To retire Retired
Capability	Groups each asset as part of a business capability. See section 4.6.
Business Owner	Indicates which organization owns this application. If TennCare, indicates which organization unit is the business owner.
Type	SaaS, COTS, Custom Built, etc.

Attribute	Description
Sensitivity	Indicates whether this application manages sensitive data, including PII, PHI, etc.

6.2.2. Sample

Figure 32: Sample excerpt of the Application Landscape illustrates with fictionalized application names that do not represent current TennCare architecture.

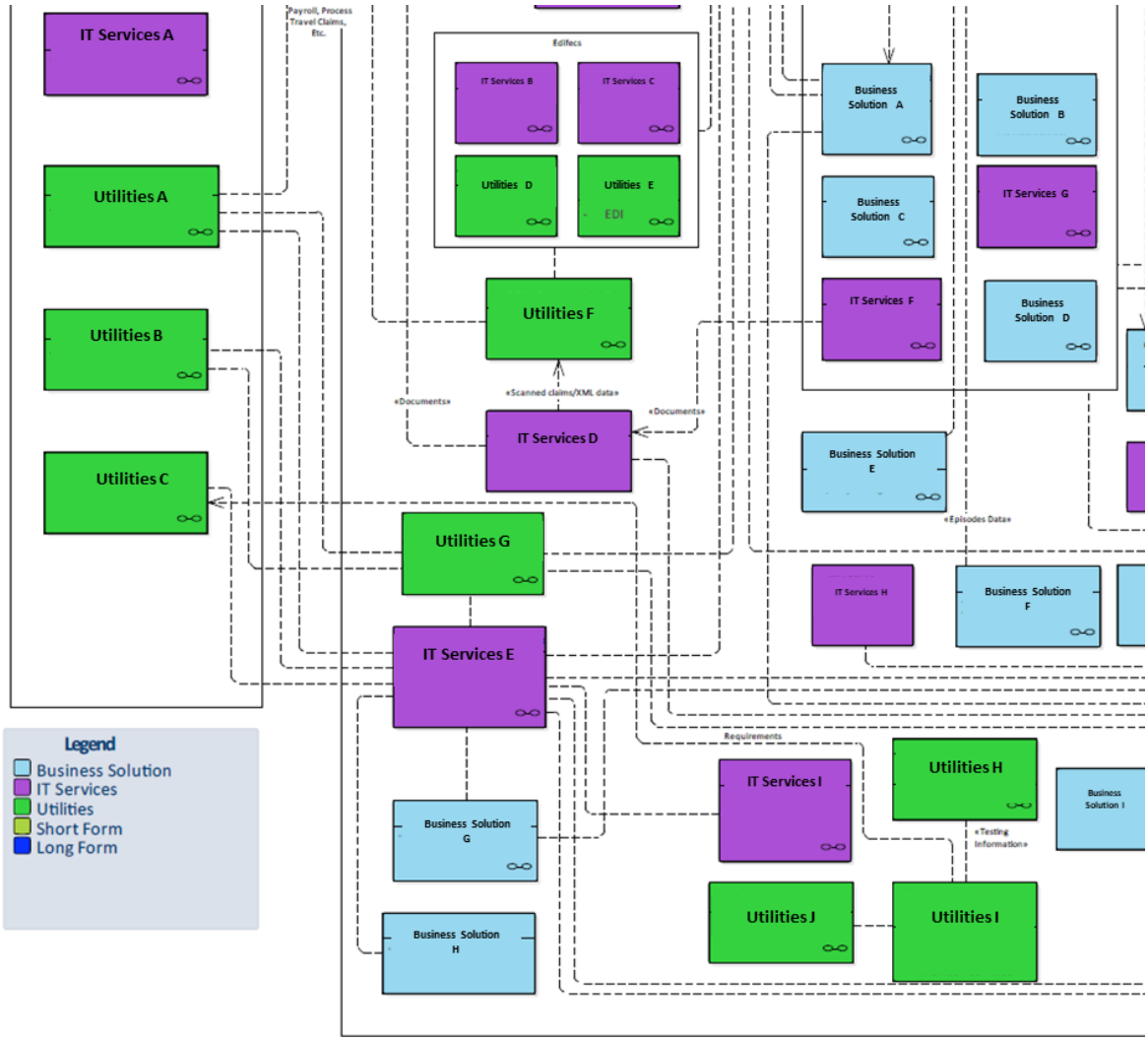

Figure 32: Sample excerpt of the Application Landscape

6.3. Application Roadmap

Application Roadmap		
Enterprise	Application Architecture	Recommended
Summary	A diagram showing the existing and future applications, organized by phases.	
Aliases	Target Architecture Vision	
Format	Summary diagram	
Purpose	Developing this artifact helps to plan a portfolio of solution projects that develop applications: • Ensuring the planned applications cover the required scope • Choosing which projects should be done, when • Ensuring resources are available to be allocated across all the projects The completed artifact provides context to project participants, to see how their work fits with related initiatives. Stakeholders will also use this artifact to know what application changes are planned, when.	
Related	Aligned with a portfolio projects and its timeline of phases. Application Component Model will detail the applications. Aligned with the Capability Model.	

6.3.1. Definition

The Application Roadmap summarizes the current and future applications of the enterprise. Applications are grouped according to phases and/or capabilities.

Application modules are shown with business names. Functions of the applications may be summarized. Major integrations between the applications may be shown as lines or arrows.

6.3.2. Sample

Figure 33: Sample of the Application Roadmap shows a structure for this artifact. Details are obscured, and may not represent TennCare's current plans.

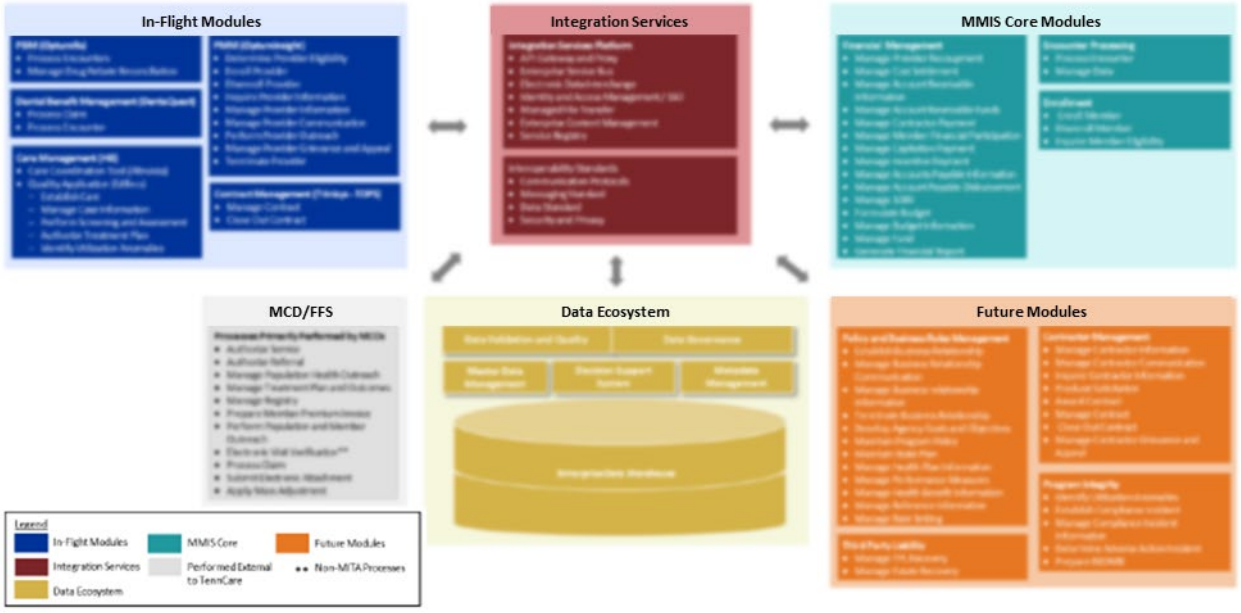

Figure 33: Sample of the Application Roadmap

6.4. Value to Quality Matrix

Table 31: Value to Quality Matrix

Value to Quality Matrix		
Enterprise	Application Architecture	Suggested
Summary	Places applications or other technology assets in a 2x2 matrix, according to their business value and their technical quality.	
Aliases	Business Value to Technical Quality Matrix Business Value to Technical Condition Matrix BV/TQ Matrix Business Value Assessment Technique	
Format	Scatterplot, placing each asset on two axes: Business Value and Technical Quality.	
Purpose	Assesses a portfolio of technology assets such as applications. Shows where investments need to be made (e.g., upgrade the technical quality of a valuable application). Identifies cost savings opportunities (e.g., decommission an application that is no longer required).	
Related	The Application Landscape provides the list of assets to assess.	

6.4.1. Definition

In the Value to Quality Matrix, the horizontal axis represents business value while the vertical axis represents technical quality. Each asset, from a portfolio of technology assets such as applications, is represented by a dot or symbol anywhere on the 2x2 grid formed by these axes as seen in *Figure 34: Sample of the Value to Quality Matrix*.

The Business Value axis represents how important the asset is to the organization, based on criteria such as:

- Does the application provides functionality to support business needs?

- Does the application perform necessary steps automatically without manual calculations or overrides?
- Is the application easy to use?

Technical Quality of the asset is based on criteria such as whether an application is:

- Aligned with the target Enterprise Architecture vision
- Aligned with TennCare's IS Guiding Principles
- Listed in the TennCare Technology Standard Relatively low-cost to operate and maintain

The Business Value should be a numerical score derived from surveying or consulting business users and Stakeholders. The Technical Quality should be a numerical score derived from surveying or consulting technical Stakeholders.

Based on these scores, assets will fall in one of four quadrants on the matrix as illustrated in *Table 32: Value to Quality Matrix Quadrants.*

Table 32: Value to Quality Matrix Quadrants

Technical Quality		
	Enhanced Business Value • Low business value • High technical quality	**Maintain and Consolidate** • High business value • High technical quality
	Retire or Replace • Low business value • Low technical quality	**Enhanced Technical Quality** • High business value • Low technical quality
	Business Value	

The matrix will inform the decision of which assets need to be enhanced, retired, replaced or maintained.

There may be a chart showing numerical scores of each asset for both Business Value and Technical Quality. There may be a written justification or other notes about each asset.

The scores may be based on a written justification or a survey of Stakeholders.

6.4.2. Sample

Figure 34: Sample of the Value to Quality Matrix has fictionalized software names and may not represent TennCare's current architecture.

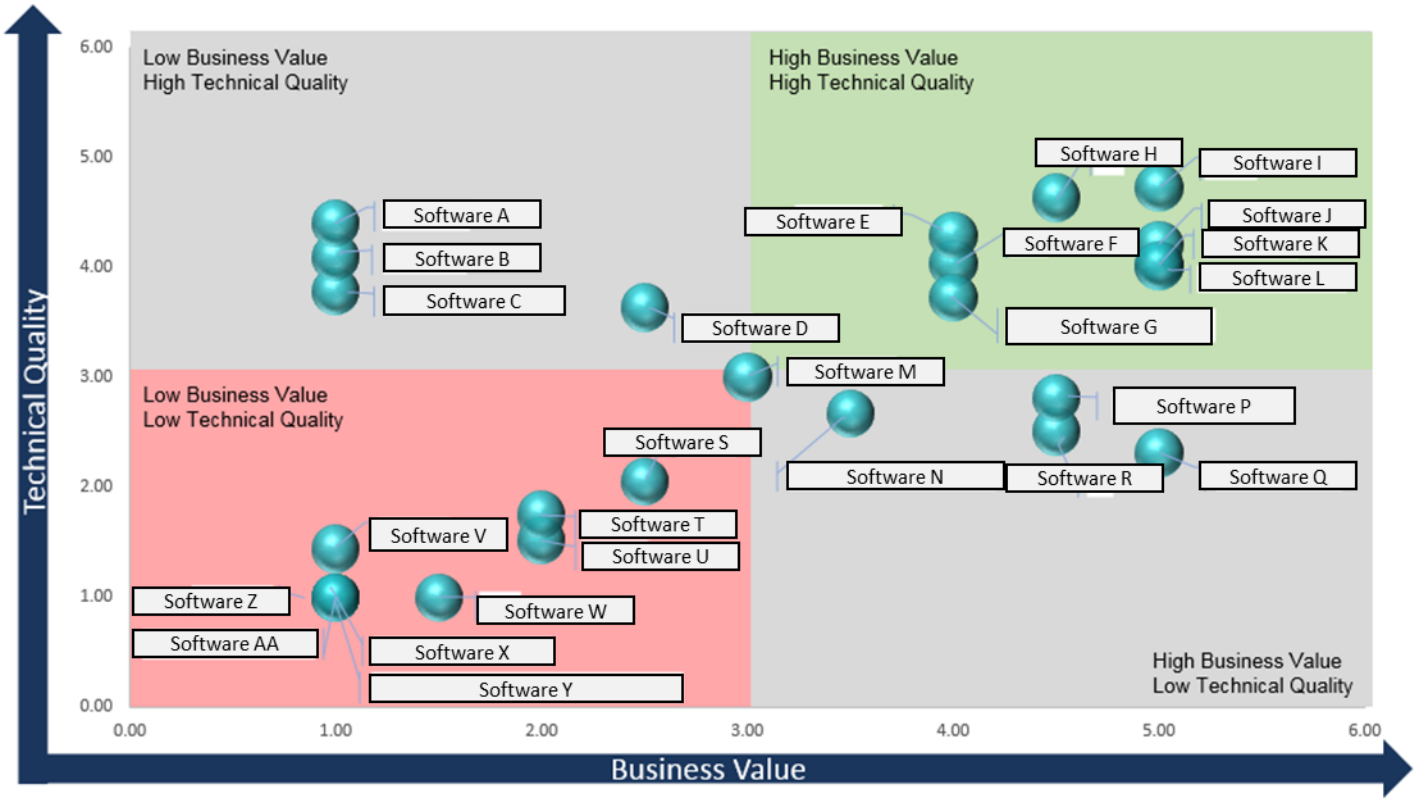

Figure 34: Sample of the Value to Quality Matrix

6.5. Heat Map

Table 33: Heat Map

Heat Map		
Enterprise	Application Architecture	Suggested
Summary	Counts and shows the number of applications used for each function of the business	
Aliases	System Heat Map by MITA Area	
Format	Chart	
Purpose	Identifies opportunities for application rationalization	
Related	Count the applications listed in the TennCare Application Landscape. Organize applications using the Capability Model or Business Function Model or an organization chart.	

6.5.1. Definition

The Heat Map visualizes the number of applications used for each business function, capability, or organizational unit. Business areas with a high number of applications are primary candidates to be examined for consolidation, to reduce the number of applications needing maintenance.

6.5.2. Sample

Figure 35: Sample of the Heat Map is an illustration, which may not represent TennCare's current architecture.

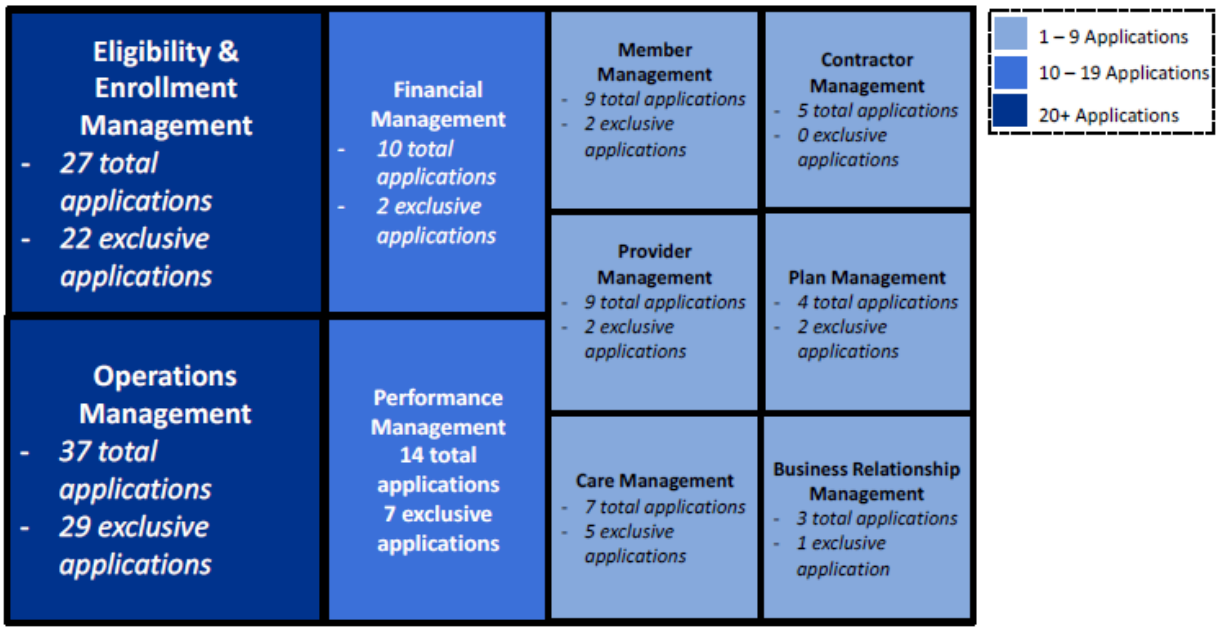

Figure 35: Sample of the Heat Map

6.6. Alternatives Analysis

Table 34: Alternatives Analysis

Alternatives Analysis		
Conceptual Design	Application Architecture	Recommended
Summary	Identifies a preferred solution approach by analyzing options based on their costs, benefits, feasibility, and alignment with business requirements.	
Aliases	Options Analysis, Cost Benefit Analysis, Recommendation TOGAF artifact: Potential Solutions	
Format	Chart	
Purpose	Analyzing the benefits and costs of alternatives aids in choosing a solution approach.	
Related	The Maturity Assessment may include analysis of the existing system. Influences decisions shown in the Application Roadmap.	

6.6.1. Definition

Alternative solution options are generally identified at the onset of a project once the vision (preliminary target architecture) has been determined.

Alternatives to compare may include:

- No change: Use the existing system
- Reuse: Build upon the existing system
- Build: Develop a new custom solution
- Build: Rewrite the existing solution with modern tools
- Buy: Acquire COTS software
- Buy: Subscribe to a managed service (Software, Platform or Infrastructure as a Service)

The alternatives may be compared using factors such as:

- Feasibility of meeting business requirements
- Alignment with target technology architecture
- Cost estimate (relative to other alternatives)
- Operational benefits
- Longevity of solution
- Risks
- Known issues

The analysis concludes by recommending a preferred alternative and giving a rationale.

6.6.2. Sample

Figure 36: Sample of the Alternatives Analysis below is an extract of this artifact, which may not represent current TennCare architecture:

Alternative	Description	Pros	Cons	Preferred Alternative?	Rationale
Maintain and enhance current interChange	TennCare maintains the current interChange system and gradually upgrades it through system change requests. Also, TennCare enhances and replaces peripheral modules, such as PBM and Provider.	This approach minimizes operational and organizational impact. Familiarity with the current vendor, which has been providing services for more than 15 years. Fewer procurements.	TennCare is changing its operating model in response to changes in regulatory environment and healthcare marketplace. The legacy system is not agile enough to support these changes.	No.	interChange is more than 15 years old. It was originally based on an FFS model and TennCare has transitioned to a Managed Care model. The current patchwork of legacy systems includes a fragmented data warehouse, a disjointed medical appeals system, and a limited capability to engage members and providers. Unable to cost-effectively support current and future TennCare operational needs.

Figure 36: Sample of the Alternatives Analysis

6.7. Application Component Model

Table 35: Application Component Model

Application Component Model		
Conceptual Design	Application Architecture	Required
Summary	Names, describes and organizes the application components that are in-scope for a solution.	
Aliases	Software Services Model, Software Component Model TOGAF artifact: Application Components	
Format	Diagram with UML Component symbols plus Chart of descriptions	
Purpose	Sets the scope of a solution and its phases if applicable. Names and describes the application components, for reference throughout the Conceptual Design and Solution Design.	
Related	Describes one or more applications on the Application Roadmap. Provide a mapping between the applications and the functions they automate (from the Business Function Model). The integration of applications is shown in the Conceptual Integration Architecture (vendor-neutral) and Solution Architecture Model (vendor's specification of products).	

6.7.1. Definition

This artifact catalogs the applications to be included in a solution.

The diagram shows a hierarchy, decomposing the solution into layers (or tiers), modules and perhaps smaller components of application software.

The diagram and/or accompanying chart should include:

- Name of layer, module or component (use vendor-neutral business names that summarize the functionality)
- Functions to be performed by the application (from the Business Function Model)
- Status of the application (in production, in procurement, planned, etc.)
- Phase when application will be implemented
- Other descriptive or strategic information as needed

This artifact does not show the connections between applications.

Figure 37: Sample of the Application Component Model and *Table 36: Sample of the Application Component Model details* provide an example of the required elements for the Application Component Model. These may not represent TennCare's current architecture.

6.7.2. Sample diagram

The following sample may not represent current TennCare architecture.

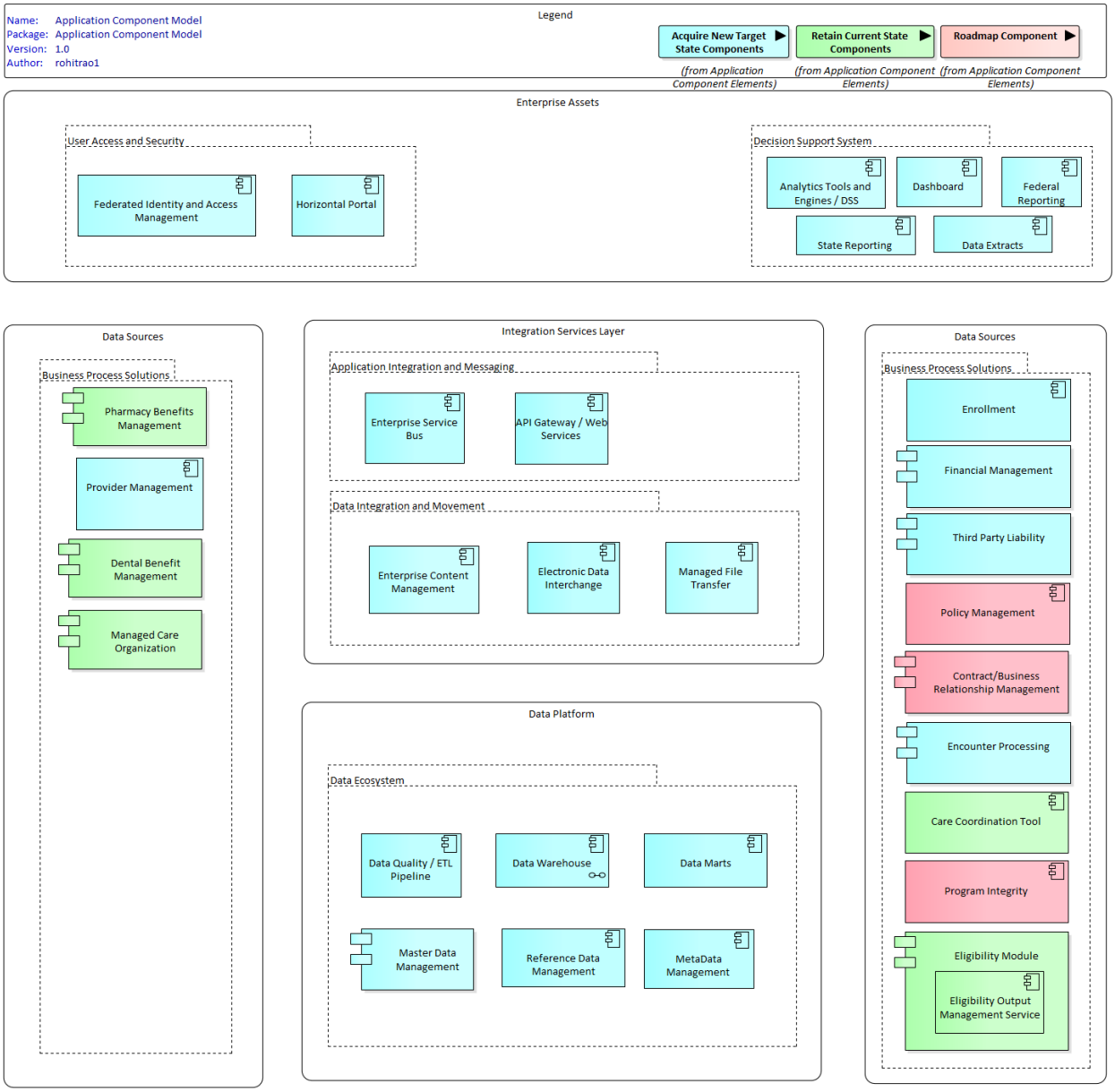

Figure 37: Sample of the Application Component Model

6.7.3. Sample chart

The following sample may not represent current TennCare architecture.

Table 36: Sample of the Application Component Model details

Application Component	Description (Business Process Supported, Purpose of Component)	Type (Identify both – (1) Operational or Analytical; (2) Batch or Online?)	Strategy (Build, Buy, Reuse, Rewrite)	Alternatives	Preferred Alternative?
Pharmacy Benefit Management (PBM) Module	The PBM Module provides support for prescription drug coverage for eligible individuals. TennCare has already procured a vendor (OptumRx) through competitive bidding.	Operational Online	Buy	None	N/A
Provider Management Module (PMM)	The PMM will be the system of record for managing all providers who participate in TennCare programs. TennCare is in the process of negotiation with a potential vendor identified through the NASPO process.	Operational Online	Buy	None	N/A
Contract Management Module	The Contract Management Module will consolidate existing platforms (TennCare Oversight Processing System – TOPS, and Serena TeamTrack) into a comprehensive	Operational Online	Buy	The TOPS project is currently in-flight. Neither TOPS nor	No

6.8. User Roles

Table 37: User Roles

User Roles		
Solution Design	Application Architecture	Recommended See section 1.5
Summary	Describes all the roles that users can play in a solution. Shows the hierarchy of user roles, if any.	
Aliases	Stakeholder Catalog, Actor Hierarchy TOGAF Artifacts: Organization/Actor Catalog, Role Catalog, Actor/Role Matrix, Stakeholder Map Matrix	
Format	Chart and Tree Diagram if there is hierarchy	

User Roles	
Purpose	Shows that all Stakeholders who need to use the automated solution will have a User Role. Distinguishes the access rights of different User Roles.
Related	Derived from the Stakeholder Model. Include all User Roles in the Logical and Physical Data Models.

6.8.1. Definition

The User Roles artifact is a chart that lists all the user roles in a solution. This artifact can also contain a hierarchy diagram showing the relationship of one user to another user.

Any Stakeholder that will use the automated solution should have a User Role. Each User Role will have access to different application components and data, at specified security levels. Roles should not be confused with job descriptions or titles, although those can overlap.

The chart should specify:

- User Role Name
- User Role Description
- User Role Parent (if there is a hierarchy of User Roles)
- Access Rights (may be denoted by security level, class of sensitive data, etc.)
- Stakeholder(s) with this User Role (traceability mapping)

6.8.2. Sample diagram

Figure 38: Sample of User Role hierarchy diagram illustrates this artifact. This sample may not represent current TennCare architecture.

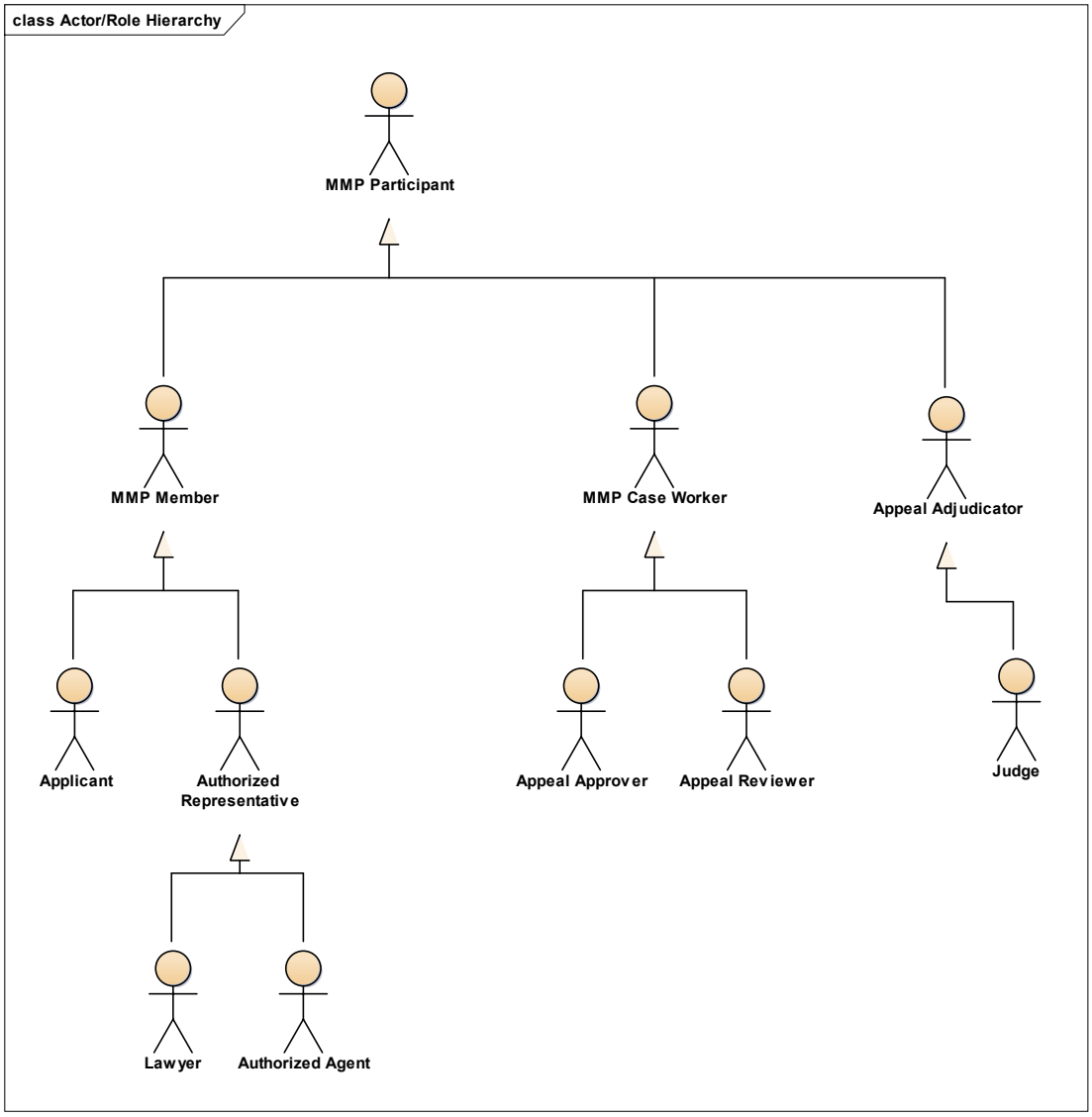

Figure 38: Sample of User Role hierarchy diagram

6.8.3. Sample chart

Table 38: Sample of the User Role system mapping shows a small extract of the mapping of application components to user organization units, showing who will use what. (This sample may not represent current TennCare architecture.) Similar charts are suggested for solutions with many components and many groups of users.

Table 38: Sample of the User Role system mapping

MMIS User Communities		
System	***Organizational Unit***	***Users***
Data Warehouse	TennCare Policy Group	• Program Management
	Managed Care Ops	• Program Integrity • Behavioral Health Operations
	Finance	• Accounts
interChange	Member Services	• Member Services • Business Operations
	Managed Care Ops	• OCM • MCC Compliance
	TennCare Information Systems	• Chief Information Officer • Systems Operations • Encounters and Claims • Department of eHealth • MMIS Eligibility, Enrollment, and Financial Staff

6.9. Solution Architecture Model

Table 39: Solution Architecture Model

Solution Architecture Model		
Solution Design	Application Architecture	Recommended See section 1.5
Summary	A logical overview of all application components in the vendor's solution. Arrows summarize the message flows between the components of the solution and with other systems.	
Aliases	Solution Architecture Diagram	
Format	A diagram of nodes and links within boundaries.	
Purpose	Confirms that the solution will include the required components and integrations, using products on the TennCare Technology Standard. Outlines the servers and interfaces needed in the technology architecture.	
Related	Includes all applications from the Application Component Model. Includes the major User Roles. Based on the Conceptual Integration Architecture. Basis for the Deployment Model.	

6.9.1. Definition

This artifact shows how a Solution Design will implement the Solution Design of applications and integrations.

The nodes on the diagram are application components, showing both the business name (from the Application Component Model) and the product name proposed by the solution vendor. User Roles or groups thereof should also be shown as nodes (person symbols).

The links on the diagram are arrows showing the message flows between the components of the solution and with other systems. The arrow labels are short summaries of the data that is passed between the components, and/or the process that is executed by sending messages between the components. Example: “Fee payment”. Ensure the message flows are clearly traceable to the System Process Model and the Logical or Physical Data Model.

The nodes are shown within boundary boxes. The outer boundaries represent TennCare and other organizations. The inner boundaries represent groups of applications (systems, solutions, tiers, or layers).

This diagram of information flowing between applications also serves as a high-level “data flow diagram”.

If one diagram is insufficient space, provide an overview diagram showing the most important applications and integrations. Provide additional diagrams or charts with further details.

Following the “gray box” principle (section 2.3.2), this diagram will have little information about the application component structure within a purchased product. It is important to show the inputs and outputs of that product as integration arrows.

The Application Component Model specified the functionality of each application component, in the Solution Design. If this differs in the Solution Design, provide an updated chart with:

- Application component business name
- Application component product name
- Function, or high-level system processes, executed by the component
- Other descriptive information as needed

6.9.2. Sample

As seen in *Figure 39: Sample of the Solution Architecture Model*, this simplified fictional sample shows suitable boundaries and applications with both business and product names, which may not represent current TennCare architecture.

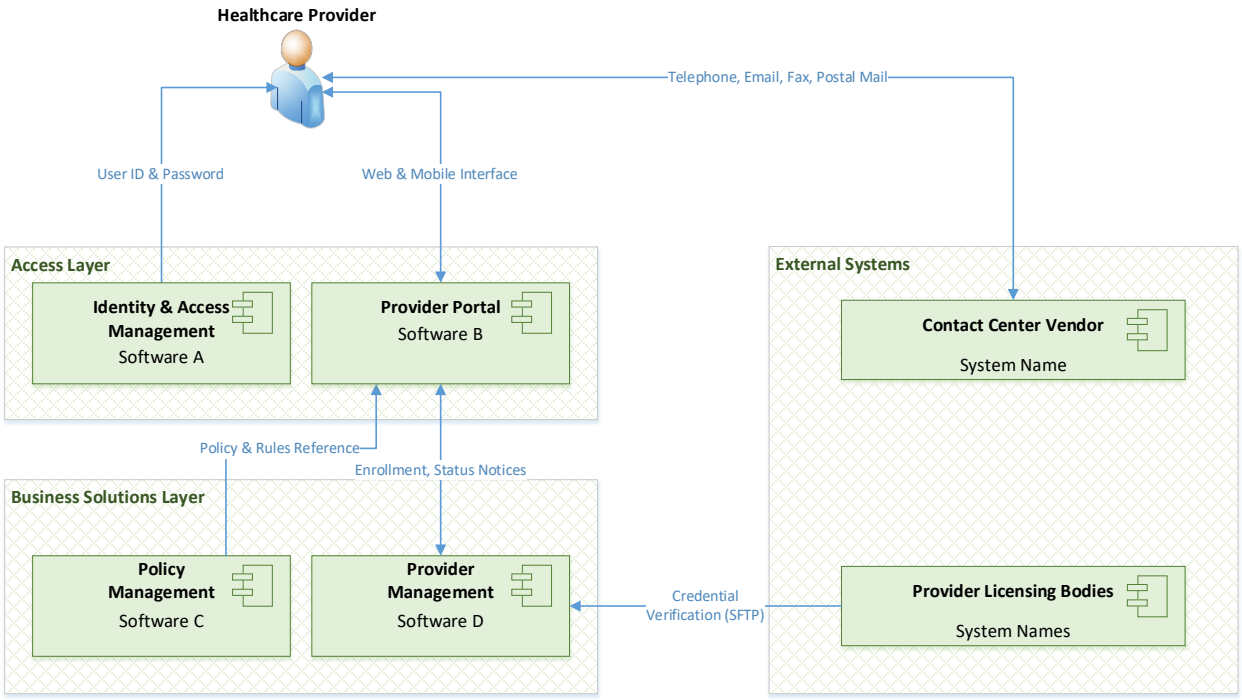

Figure 39: Sample of the Solution Architecture Model

6.10. System Process Model

Table 40: System Process Model

System Process Model		
Solution Design	Application Architecture	Recommended See section 1.5
Summary	Workflow diagrams showing the system processes that implement the business processes.	
Aliases	Workflow Diagrams, Swimlanes Similar to System Use Case Model and Specification	
Format	Preferred: Workflow swimlane diagrams in BPMN format Acceptable: Use case models and specifications	
Purpose	Shows how a solution will satisfy business requirements. Shows who will do what, and how their activities will be automated and orchestrated. Enables discussion of how an application will be designed, customized or configured.	
Related	Based on the Business Process Model and the Business Scenarios. Hierarchy of system processes is aligned with the Business Function Model decomposition of processes. Swimlanes include the User Roles and application components from the Solution Architecture Model. Process steps may be linked to Requirements as defined in the TennCare Requirements Management Standard. Process steps may be governed by Business Rules. Process steps may use or modify data elements specified in the Logical or Physical Data Model.	

6.10.1. Definition

The System Process Model is a set of workflow diagrams that show how the solution executes its functions as seen in *Figure 40: Sample of the System Process Model.* The functions are decomposed into a hierarchy of system processes, which should be aligned with the Business Function Model and Business Process Model.

Each system process is modelled on a BPMN workflow diagram, which specifies:

- A sequence of steps (activities and events)
- Swimlanes, showing which User Role or Application Component does each step
- Arrows showing the sequence of steps
- Decision logic
- Requirements that are satisfied by the process steps
- Business Rules governing the process steps, where applicable
- Data elements that a process creates, modifies, shares or uses

This artifact shows how the Business Processes are implemented in the system. Therefore, use the solution vendor's names for processes and software products.

6.10.2. Sample

The following sample may not represent current TennCare architecture.

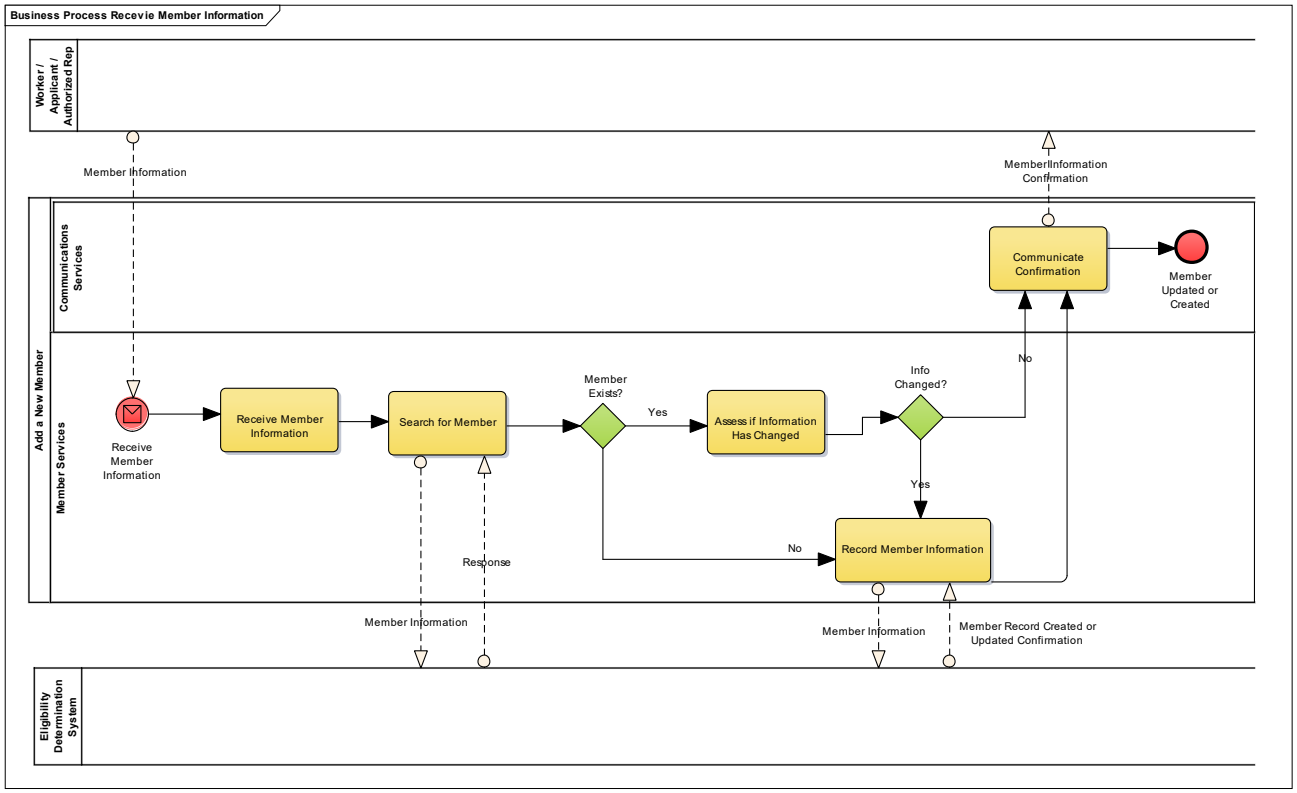

Figure 40: Sample of the System Process Model

7. Technology Architecture Artifacts

Technology architecture also includes document artifacts that are not defined as models in this standard, such as:

- Technology architecture descriptions in the Concept of Operations
- Interface Control Design Document
- System Architecture Design Document
- System Design Package
- Cloud Design Package

See section 1.7.2 about templates for these documents.

7.1. Core Artifacts and Relationships: Technology

Figure 41: Core Artifacts and Relationships: Technology illustrates the main relationships between the required and recommended Technology Architecture artifacts. Artifacts from other domains are shown if they provide input to Technology Architecture or need alignment with it.

Many of the Technology Architecture artifacts are compared in section 7.3, *Table 42: Comparison of Integration Overviews*.

Blue boxes are Enterprise artifacts, red boxes are Conceptual Design, and orange boxes are Solution Design.

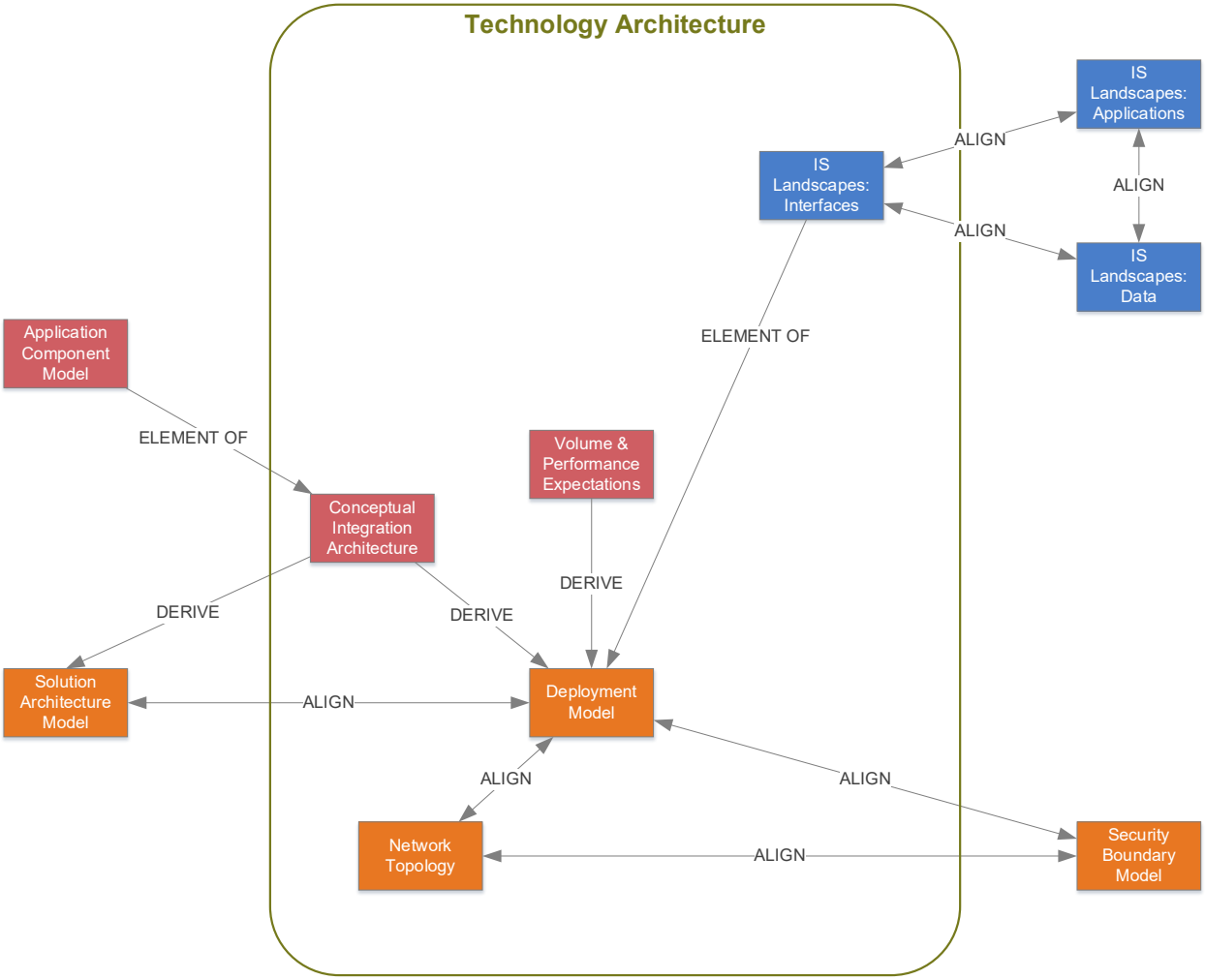

Figure 41: Core Artifacts and Relationships: Technology

7.2. Interface Landscape

Interface Landscape		
Enterprise	Technology Architecture	Required
Summary	A visual summary of the interfaces that TennCare uses to integrate its applications and databases.	
Aliases	Interface Catalog, Asset Inventory	
Format	Poster-sized diagram, showing interfaces between applications and data assets	
Purpose	The Application, Data, and Interface Landscapes are a comprehensive reference tool for TennCare IS to: • Set enterprise-wide IS strategy • Identify cost savings opportunities by leveraging existing investments • Manage performance, security, business continuity and disaster recovery	
Related	IT assets are connected and shown on the three aligned Application, Data and Interface Landscapes. The Capability Model is used to group assets on the Landscapes. The IT asset inventory provides details about each interface on the landscape. Detailed by the Interface Control Design Documents (a Solution Design deliverable). Provides a reference list of interfaces to be used in many artifacts.	

7.2.1. Definition

The Interface Landscape is part of the TennCare IS Application, Data and Interface Landscapes. The IS Landscapes visually represent all of the information systems assets of

TennCare, and how they are connected. The IS Landscapes are the enterprise architecture blueprint for both the current state and target state (future).

The Interface Landscape displays limited attributes, based on a more detailed inventory of TennCare's databases and other information assets. The Interface Landscape, including the status of each data asset, shall be updated at each Architecture Governance Review.

The following are the minimum attributes required for the Interface Landscape:

Table 41: Interface Landscape Attributes

Attribute	Description
ID	Unique Identifier for the interface
Name	Name or Acronym for the interface
Description	Description of the interface purpose
Trading Partner	The external organization that this interface interacts with, if any
Providing System	The application system that sends data over this interface
Consuming System	The application system that receives data over this interface
Type	The interface type, e.g. Real-Time, Batch, EDI/X12
Protocol	The interface technology, e.g. SFTP, SOAP, EDI, HTTPS
Status	Indicates whether the interface is: Planned In development In production To retire Retired

Attribute	Description
Capability	Groups each interface as part of a business capability, which may match a MITA function. See section 4.6.
Owner	Indicates which organization, or unit of TennCare, owns this interface
Sensitivity	The types of sensitive data handled by this interface, including PII, PHI, proprietary, etc.

7.2.2. Sample

Interfaces between data and applications are shown as connection lines on Figure 24 (section 5.2.2), colored by sensitivity.

Figure 42: Sample Interface Landscape Extract is an excerpt of relevant fields from a fictionalized interface inventory as would be used to diagram the Interface Landscape:

Interface Name	Purpose	Functional Area	Inbound/ Outboun	Trading Partner	Providing System	Consuming System	Type of Interface
Address Validation	Address verification enables Worker, Member and Partner Portals to perform effective outreach, have proper mailing locations, and provide correct residence information for MCO assignments.	E&E	N/A	TennCare Internal	XYZ	TEDS	Real-time
Beneficiary Exchange (BENDEX)	Retrieve and update Medicare information daily for effective crossover claim processing, (e.g. access SSA common working file) for access to the data.	Eligibility	Inbound	SSA	N/A	ABCDE	Batch
Carrier Master Extract	MCC Carrier File	TPL	Outbound	MCC's	ABCDE	N/A	Batch
Defense Enrollment Eligibility Reporting System (DEERS) Inbound File	The data required to generate 270 files to be sent to DEERS will be coming from TPL. The 271 file coming from DEERS will be sent to TPL. 270 Outbound and 271 Inbound will be created through the EDI translator.	TPL	Inbound	DOD	Defense Enrollment Eligibility Reporting System	ABCDE	EDI/X12
Encounter NCPDP	Drug Encounters are received from the PBM and once processed are loaded to claims history. Drug Rebate determines on a quarterly basis which claims are appropriate, and retrieves them directly from claims history for the purpose of invoicing manufacturers.	Claims	Inbound	PBM	N/A	ABCDE	EDI/X12
Vital Statistics Date of Death	Vital Statistics Date of Death	SUR	Inbound	Dept. of Vital Statistics	N/A	EDW	Batch

Figure 42: Sample Interface Landscape Extract

7.3. Integration Overviews

Table 42: Comparison of Integration Overviews compares the artifacts that provide overviews of how applications and technology will be integrated. These artifacts show distinct information but should be aligned with each other. The artifact definitions explain the cells of this table.

Table 42: Comparison of Integration Overviews

Artifact Name	Conceptual Integration Architecture	Solution Architecture Model	Deployment Model	Network Topology	System Boundary Diagram
Section	Section 7.4	Section 6.9	Section 7.6	Section 7.7	Section 8.2
Scope	Conceptual Design	Solution Design	Solution Design	Solution Design	Solution Design
Domain	Technology (and Application)	Application	Technology	Technology	Security
Bounds	Organizations, app layers, clouds, platforms	Organizations, app layers	Security zones, clouds, hosting locations	Security zones, clouds, hosting locations	Security zones
Nodes	Application & technology components	Application components, user roles	Servers	Network HW & SW, locations, servers, devices	Network HW & SW, locations, servers, devices
Links	Message flows and integration technology	Message flows	Message protocols	Network connections	Network connections, message protocols, data sensitivity

7.4. Conceptual Integration Architecture

Table 43: Conceptual Integration Architecture

Conceptual Integration Architecture		
Conceptual Design	Technology Architecture	Required
Summary	An overview of how the applications and technology will be integrated to form a solution. Shows conceptual application and technology components within boundaries of organizations, solutions, layers, clouds, and/or security zones. Arrows summarize the major message flows between the components of the solution and with other systems, and indicate the integration technology.	
Aliases	Conceptual Architecture Diagram, Software Services Model, Application Integration Architecture Diagram, System Integration Diagram, System Architecture Diagram TOGAF artifact: Application Communication Diagram	
Format	Diagram with boundaries, nodes, and links	
Purpose	Informs the solution vendor which applications need to be connected to form an integrated solution. The basis for designing technology to provide interfaces and the underlying network. Outlines where security is needed to protect data in transit.	
Related	Includes all applications from the Application Component Model. Basis for the Solution Architecture Model and the Deployment Model.	

7.4.1. Definition

This artifact is a vendor-neutral overview of how the application components and technology will be integrated to form a solution as seen in *Figure 43: Sample Conceptual Integration Architecture*. It is the high-level design of how internal and external systems will send and receive data, to carry out business functions.

The nodes on the diagram are aligned with business names of applications from the Application Component Model. Diagram nodes may also indicate middleware, network equipment, or other integration technology.

The links on the diagram are arrows showing the messages, integrations and interfaces between the components of the solution and with other systems. These links may be labeled with one or both of:

- Short summaries of the data that is passed between the components, and/or the process that is executed by sending messages between the components. Example: “Fee payment”. Ensure the message flows are clearly traceable to the Context Model, Business Operating Model, or Business Scenarios.
- The integration technology that will send the messages, including point-to-point, middleware-based and Services Oriented Architecture (SOA) technologies.

The nodes are shown within nested boundary boxes, which may represent:

- TennCare and other organizations
- Application layers for presentation, business applications, databases, and infrastructure
- Clouds, platforms, or other technology groupings
- Security zones, aligned with the above

7.4.2. Sample

Figure 43: Sample Conceptual Integration Architecture illustrates this artifact for a fictional solution that would add the yellow servers to both cloud and on-premises hosting. This small example may not represent current TennCare architecture, which has more nodes, connections, and layers of boundaries.

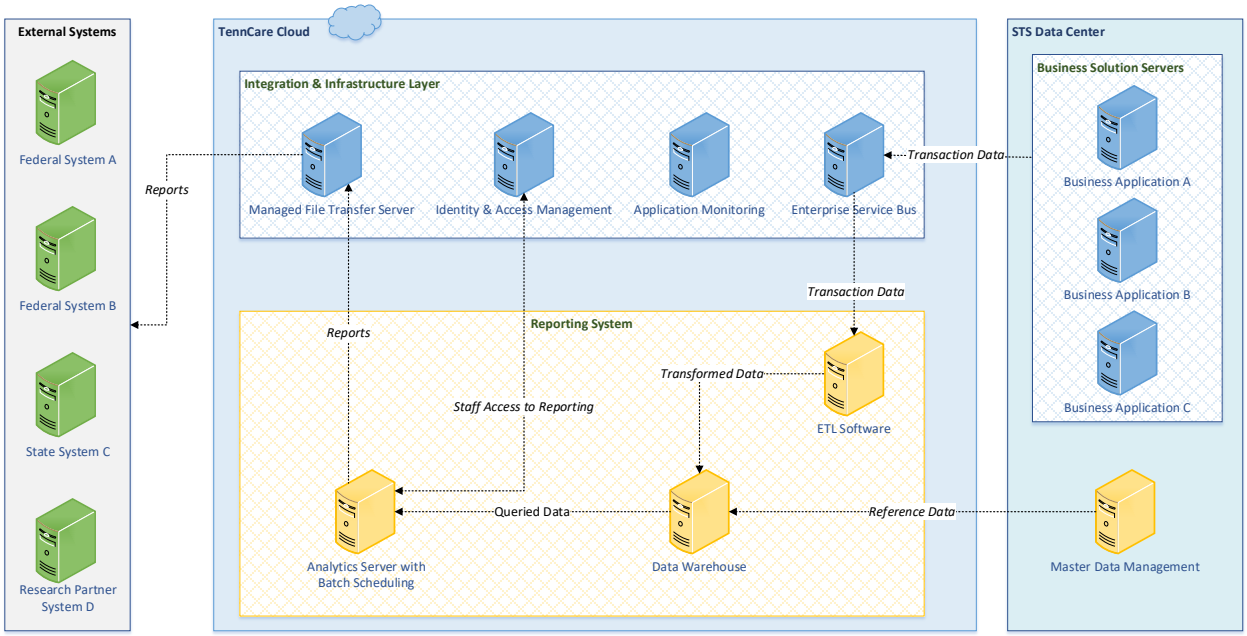

Figure 43: Sample Conceptual Integration Architecture

7.5. Volume and Performance Expectations

Table 44: Volume and Performance Expectations

Volume and Performance Expectations		
Conceptual Design	Technology Architecture	Recommended
Summary	Describes the expected volume of users, messages and transactions in a solution, and the expected performance.	
Aliases	User counts; Transaction counts; Volumetrics; Volume estimates	
Format	Charts	
Purpose	Requires a solution vendor to support the quantity and speed of users, messages and transactions needed. Basis for technology and network design.	
Related	May be part of the Non-Functional Requirements. Network user counts align with the Stakeholder Model. May be based on Stakeholder Interactions extension of the Context Model.	

7.5.1. Definition

For a solution, this artifact specifies the expected volume (number and size) of:

- Network Users, total and concurrent, internal and external as seen in Figure 44: Sample Volume and Performance Expectations
- Inbound and outbound messages to be transmitted
- Transactions to be processed as seen in Table 45: Sample of Volume and Performance Expectations

Combined with the required performance (speed) for transmittal or processing one transaction, a solution vendor can calculate the computing and network capacity needed to achieve the performance expectations.

7.5.2. Sample: Network user estimation chart

The following sample may not represent current TennCare architecture.

User Group	Description / Expected Use of System	Type (Federal Employee, Contractor)	Geographic Location	Network Profile (LAN, WAN, External)	Total Users (Approx.)	Concurrent Users (Approx.)
Authorized Data Users	Users that have been granted authorized access to the TennCare data environment—data visualization tools, data warehouse, dashboards, etc.	All – external stakeholders	Various	External	250	25
Service Partners	Partners that will send or receive TennCare data through EDI transactions	MCOs CMS PBM DBM DIDD VANS – Value-Added Networks	Various	External	20	20

Figure 44: Sample Volume and Performance Expectations

7.5.3. Sample: Transaction volumes

The following sample may not represent current TennCare architecture.

Table 45: Sample of Volume and Performance Expectations

What is the anticipated volume of records per day/month/year ?	Currently, the Medicaid Enterprise system maintains a member enrollment volume of approx. 1.4 million citizens and receives data from different trading partners and MCOs.
	Major types of transactions the system processes include:
	• Claims
	• Encounters
	• Capitation payments
	
	The below volumes are approx. based on a five-day business week
	Outbound transactions:
	• 834 transactions: 75,000 to 150,000 transactions per week approx. and will be sent to each of the entities: AAA, BBB, CCC, DDD, and EEE systems/providers
	• 835 claims financial transactions are approx. 130,000 to 300,000 per week
	
	Inbound Transactions:
	• Enrollment and Eligibility: 175,000 per week
	• 834 transactions: 20,000 to 50,000 per week
	• DCS, ACS (US post office) transactions are a few hundred per week
	• DCS & In Custody: 8,000 to 10,000 – Everyday processing – replacement file sent everyday
	• Provider daily file transactions:
	  ○ DOH: 400,000 per day
	  ○ Periodic updates from MCO: 100,000–150,000 per MCO per day
	  ○ PBM: 5,000 per day
	
	Capitation payments:
	• Capitation volume is approximately 135% of the currently enrolled volume per month

Will there be peak processing periods throughout the day/week/month/year?	TennCare accepts applications anytime of the year and redetermination is based on enrollment date. However, it does seem to have a peak with applications around January–February for referred applications from FFM. The peak is believed to be introduced by the open enrollment period of FFM, not by TennCare members themselves.

What is the anticipated nature of transactions in the system? Will transactions be evenly distributed or clustered around specific times?	The transactions into the system will vary. For near real-time needs, open socket web services with near real-time could be used. For more traditional needs, daily or weekly batch aggregation will be incorporated. The timing of the batch ingestion will be such as to cause the least impact on other processes.
Will the average transaction be small (e.g., data entry of single records) or large data transmissions (e.g., retrieval of large data sets for reports)?	The very nature of a Medicaid enterprise system leverages X12 standards for EDI transactions. By determining the business need and required response times choosing between web services and batch aggregation, speed and performance issues can be mitigated.

7.6. Deployment Model

Table 46: Deployment Model

Deployment Model		
Solution Design	Technology Architecture	Recommended See section 1.5
Summary	An overview of the technology of a solution, showing hardware and software nodes (such as servers) within security zones, and communication paths between the nodes.	
Aliases	Logical Technology Deployment Architecture	
Format	Overview diagram	
Purpose	Provides a blueprint for technology infrastructure planning and design. Shows how applications will be integrated, securely, to implement an IS solution that meets the Non-Functional Requirements.	
Related	A translation of the Conceptual Integration Architecture to physical technology architecture. A translation of the Solution Architecture Model from applications to technology architecture. The Deployment Model, Network Topology, and System Boundary Diagram are three aligned views on the same technology. Communication paths should be inventoried in the Interface Landscape.	

7.6.1. Definition

A deployment model describes a logical technology architecture in the form of hardware and software **nodes** (such as servers) arranged into **security zones** (such as clouds, platforms, or tiers) and linked by **communication paths** (interface technology).

Nodes

Nodes are the deployment targets that represent computational resources upon which software services 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.

The Deployment Model may show nodes as boxes, server symbols, etc., labeled with their functionality.

Communication Paths

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.

The Deployment Model shows communication paths as lines or arrows between nodes, labelled with the technology to be used for messaging, interface and integration.

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

The Deployment Model shows security zones as boundary boxes, labelled as clouds, platforms, tiers, etc. Organization names may be part of the zone name. The zones should be aligned with the boundaries in the Conceptual Integration Architecture.

7.6.2. Sample

In *Figure 45: Sample Deployment Model*, server nodes and zone boundaries are shown. The communication path technology is not indicated on the links, as this is an early (conceptual) iteration of the model. Color is used to indicate layers of applications. This generic model may not represent current TennCare architecture.

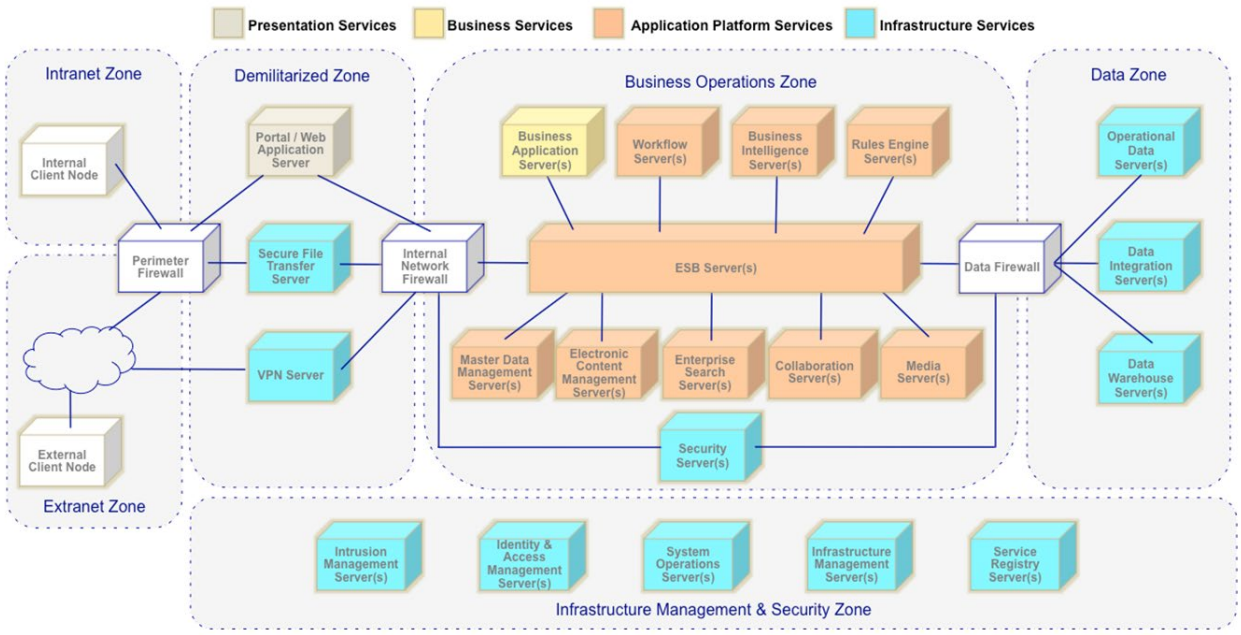

Figure 45: Sample Deployment Model

7.6.3. Extension: Capacity and Performance Plan

The Deployment Model may be enhanced with text or a chart explaining how the proposed technology would meet the non-functional requirements, especially for capacity and performance.

This plan will respond to the Volume and Performance Expectations.

7.7. Network Topology

Table 47: Network Topology

Network Topology		
Solution Design	Technology Architecture	Recommended See section 1.5
Summary	Shows the networking hardware and software that connects the technology infrastructure for a solution, within the enterprise network.	
Aliases	Network Diagram	
Format	Diagram	
Purpose	Developed to design changes to the enterprise network, to connect the new infrastructure of a solution. Shows how the network will support the solution's communication needs, meet the non-functional requirements, and securely integrate across organizations and technology environments. Provides a technical blueprint for maintaining the network.	
Related	The Deployment Model, Network Topology, and System Boundary Diagram are three aligned views on the same technology.	

7.7.1. Definition

The Network Topology is a diagram of network hardware and software that connects the servers and other nodes on the Deployment Model. The networking for a new solution will connect to or modify the existing enterprise network, so the solution's network should be modeled in the enterprise context.

On the topology diagram, use commonly recognized symbols to represent:

- Boundaries of secured areas of the network, including clouds, virtual private clouds (VPCs), subnets, data center environments
- The Internet

- Infrastructure nodes such as servers, user devices, data centers
- Network hardware and software nodes such as firewalls, routers, switches, gateways, load balancers, etc.
- Network connections between any of the nodes

Technology and security details may be shown on the diagram, such as IP addresses, subnets or port numbers.

7.7.2. Sample

Figure 46: Sample Network Topology is from the Conceptual Design of a solution. Once the Solution Design is complete, the physical network topology would specify actual rather than representative clouds and network equipment, with IP addresses and other details. This sample does not represent current TennCare architecture.

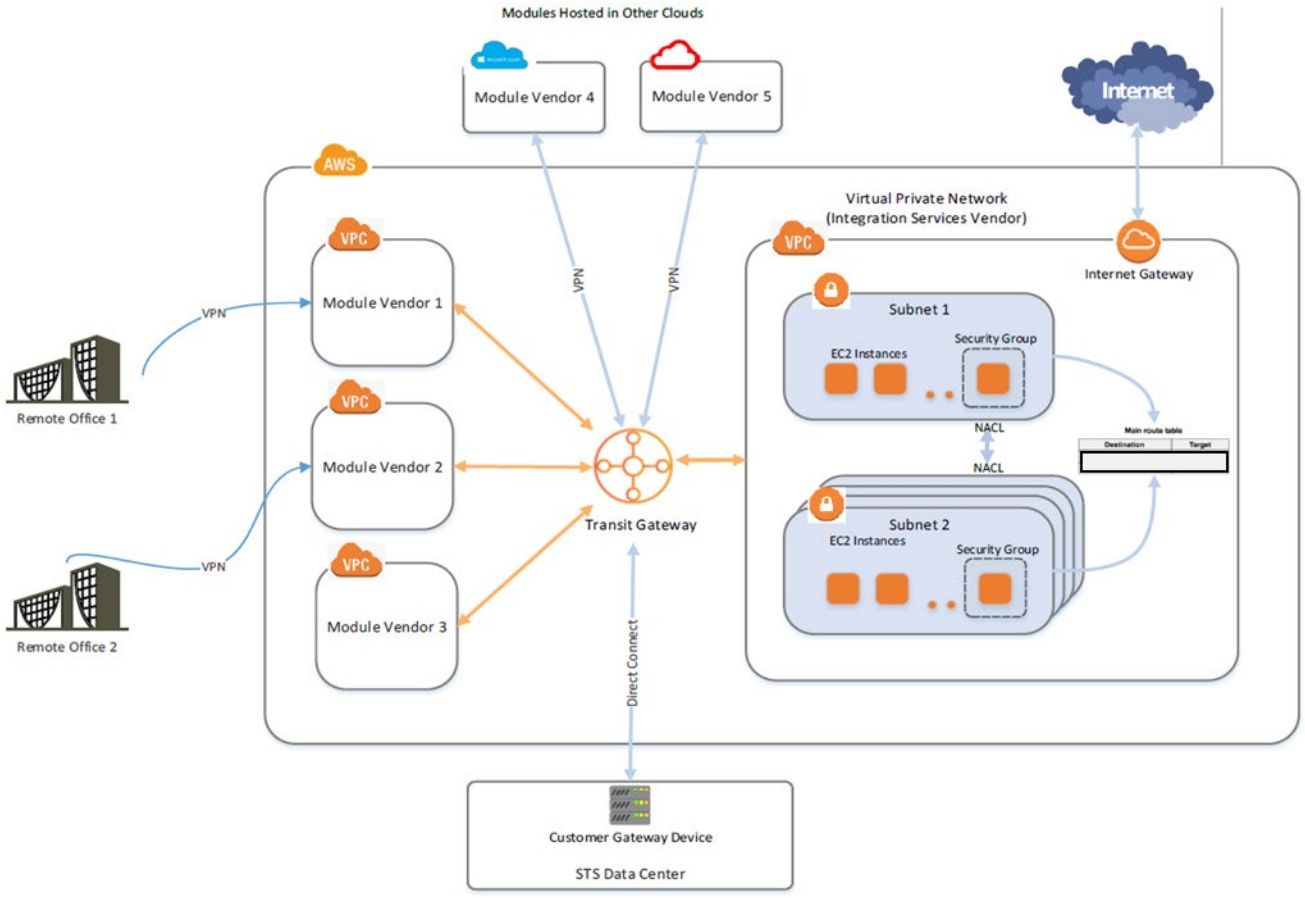

Figure 46: Sample Network Topology

8. Security Architecture Artifacts

8.1. Security in Other Domains

Security architecture is developed simultaneously with the other architectural domains, and it relies upon artifacts from other domains:

- Data sensitivity is noted in the data dictionaries of the Conceptual, Logical and Physical Data Models
- Security access rights are noted for application User Roles
- The technology architecture, including the Deployment Model and Network Topology, is designed to ensure security.

Contact TennCare IS Security for the template of the System Security Plan (SSP) deliverable, which includes extensions of the above artifacts.

Figure 47: Security in Other Domains illustrates the relationship between the Security artifact (defined below) and the Technology Architecture artifacts.

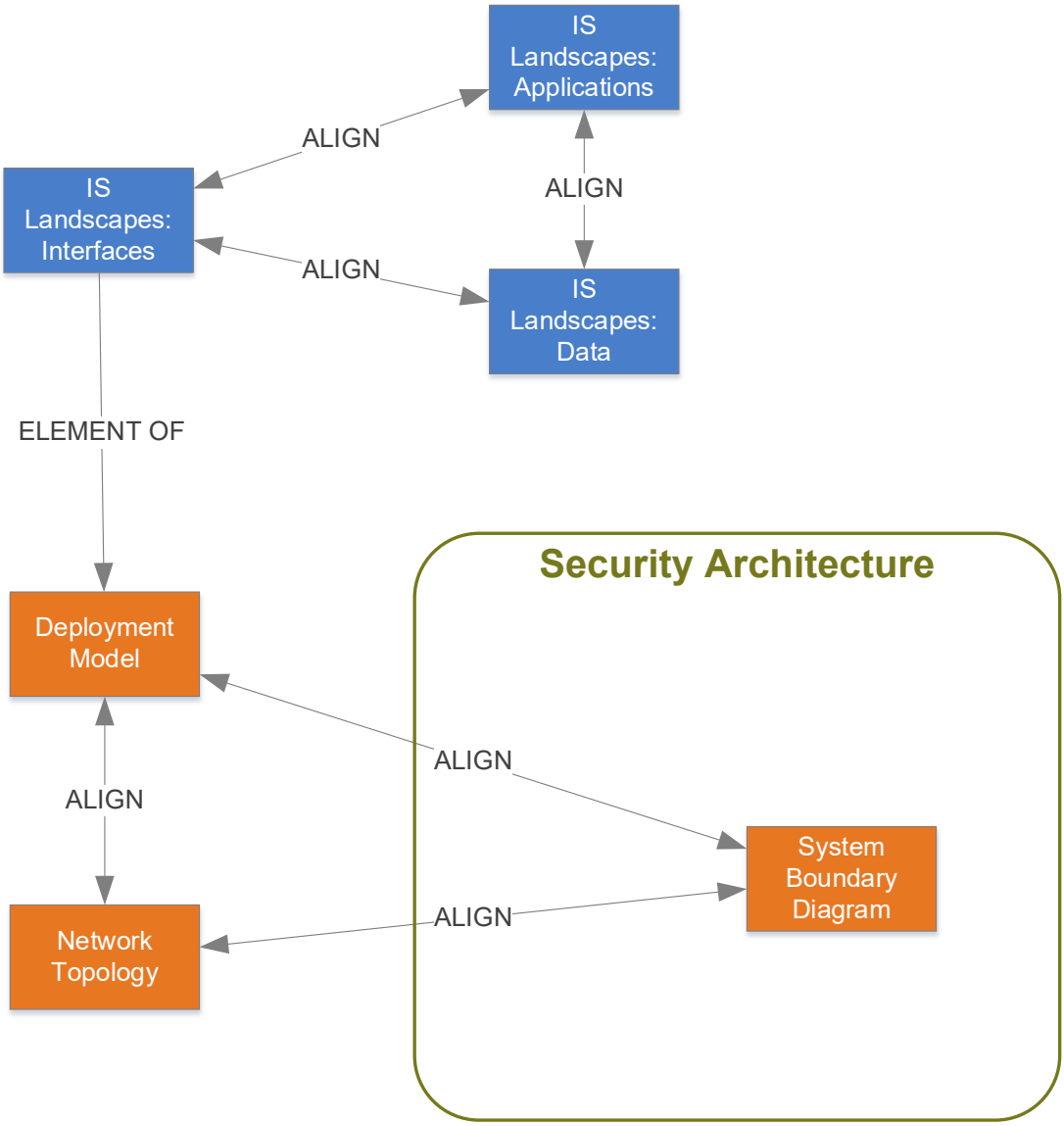

Figure 47: Security in Other Domains

8.2. System Boundary Diagram

Table 48: System Boundary Diagram

System Boundary Diagram		
Solution Design	Security Architecture	Recommended See section 1.5
Summary	An overview of a solution's subsystems and their interconnections with external systems, to depict security needs and controls. Shows security zones, servers, network hardware and software, and network connections.	
Aliases	Security Boundary Model	
Format	Overview diagram	
Purpose	Provides a blueprint for security design of the technology solution. Used to verify compliance to privacy and data protection requirements.	
Related	The Deployment Model, Network Topology, and System Boundary Diagram are three aligned views on the same technology. Data sensitivity is aligned with the Physical Data Model. This physical model should be traceable to the Conceptual Integration Architecture.	

8.2.1. Definition

The System Boundary Diagram is an overview of a solution's subsystems and their interconnections with external systems, to depict security needs and controls. The diagram shows security zones, servers, network hardware and software, and network connections.

More detailed guidance on this diagram is available from TennCare IS Security.

Appendix A: Glossary

Table 49: Glossary of Acronyms

Acronym	Meaning
BPMN	Business Process Modeling Notation
CMS	Centers for Medicare and Medicaid Services
COTS	Commercial Off-The-Shelf software
EA	Enterprise Architecture
IaaS	Infrastructure as a Service
IS	Information Systems (equivalent to Information Technology at TennCare)
PaaS	Platform as a Service
RACI	Responsible, Accountable, Consulted and Informed (a type of chart)
RFP	Request for Proposal
SaaS	Software as a Service
SILC	TennCare Solution Implementation Lifecycle Standard
TEDS	TennCare Eligibility Determination System
TOGAF	The Open Group Architecture Framework
UML	Unified Modeling Language