Appeals Case Management System Project

Technical Approach Analysis Report

Version 4.0, December 8, 2014

Health and Human Services Agency, Office of Systems Integration

Revision History

REVISION HISTORY			
REVISION/WORK SITE #	**DATE OF RELEASE**	**OWNER**	**SUMMARY OF CHANGES**
Final Version 4.0	12/08/14		Final for approval

Approvals

NAME	ROLE	DATE
Rick Murphy	Project Manager	

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Purpose

The purpose of the Technical Approach Analysis Report (TAAR) is to provide the following:

- Options Analysis – Analysis of technology options for the Appeals Case Management System (ACMS).
- Solution Evaluation Approach – Approach for evaluating ACMS solution proposals obtained in the vendor solicitation process, with respect to ACMS Functional and Non-Functional Requirements.
- Suitability Criteria – Criteria for assessing the fitness level of solutions considered for the ACMS.

The document is a work product resulting from Task 6.1 "ACMS Technical Approach Analysis " as described in The Agreement number 71531186:

> Analyze the various technology options for meeting the ACMS requirements and determine the best technical solution.

Given that this document is created in the pre-solicitation phase of the ACMS Project, the document focuses on evaluation constraints and criteria for ACMS proposed solutions expected to be received by the parties responding to ACMS Request for Proposal (RFP). The technical approach as presented in this document is intended to drive specification of architectural and technical constraints as presented in the solicitation documents for ACMS, including the RFP.

1.2 Scope

The scope for this document is delimited by the following:

- ACMS Functional requirements as elaborated to date and available in the ACMS Project's pre-solicitation.
- ACMS Non-functional requirements, such as performance, scalability, and availability requirements.
- Architectural and technological constraints and preferences as identified by standards applicable to ACMS, including Medicaid Information Technology Architecture (MITA) 3.0.

In accordance with Task 6.1 "ACMS Technical Approach Analysis " as described in The Agreement number 71531186, architectural and technical considerations are expected to include trade-offs pertaining to a number of various facets of the solution:

- The extent of the target functionality for ACMS available over time, as applicable to Commercial Off the Shelf (COTS), Modified Off the Shelf (MOTS),

leveraging an existing solution (also referred to as "transfer[1] system"), or custom solution
- The level of modifiability of the solution, different in each basic types of choices listed above
- The type of solution deployment, including a traditional data center-type deployment or deployment in the cloud (with a number of choices available in that category).

1.3 Approach

The approach adopted in this document is guided by the principle that all potential solutions for ACMS must be evaluated using a single set of criteria, consistently applied to evaluate candidate solutions, regardless of the type or source of the solution being evaluated. The approach is informed by the following observations:

- All (or virtually all) facets of a proposed solution to be evaluated for ACMS have impact on one or more components of the Total Cost of Ownership (TCO) for that solution.
- Different technical approaches to the solution for ACMS have different impacts on various components of the TCO.
- Different technical approaches to the solution for ACMS are very likely to have different impacts on the related departmental IT projects that are undertaken after the delivery of ACMS to Production.

The technical approach presented has been developed from analyses of the types of solutions available for Case Management System, ranging from solutions specific to legal domain to generic Business Process Management (BPM) solutions. In the process, a number of distinctions and taxonomies have been developed to provide a basis for evaluating technical approaches for ACMS and, consequently, to guide creation of solicitation documentation for ACMS and subsequent evaluation and scoring of proposed solutions:

- When realization of functional requirements is considered, the guiding distinction adopted identifies the following elements:
    - ACMS-specific building blocks (such as ACMS-specific information structures or ACMS-specific scheduling/calendaring)
    - Generic business building blocks (such as workflow or business rules execution and management)
    - Infrastructure elements (such as application integration or security infrastructure)

[1] For purposes of this document "Transfer" system is a generic term, however, the specific Transfer system studied for the ACMS solution is the ACCMS solution at the AOC.

- When adopting an approach to assessing suitability of technical solutions, the proposed approach is based on business alignment, involving the following:

    - Consideration of costs and benefits over the solutions full lifecycle (i.e., from acquisition to retirement of the solution)
    - Consideration of how specific types of solutions affect TCO and/or ROI (when the level of functional fit is assumed equal).

For discussion of the approach outlined above, please refer to Section 4, "Adopted Approach Overview".

1.4 References

The following table provides a description of references used in the document.

Table 1-1 Artifacts or Sources Referenced in the Document

ID	Description
TAAR	Technical Approach Analysis Report (this document)
ACMS-BRM	Business Reference Model for ACMS; see ACMS Project SharePoint portal
ACMS-BPM	Business Process Model for ACMS; see ACMS Project SharePoint portal
CEAF-RAs	CEAF 2.0 Reference Architectures: http://www.cio.ca.gov/wiki/Enterprise%20Architecture%20Documents.ashx
OSI-BP	OSI's Best Practices website, http://www.bestpractices.cahwnet.gov

NOTE: The document convention adopted to indicate a reference uses square brackets around the reference ID (e.g., OSI-BP). The reference ID may be followed by more specific information, such as page number (e.g., TAAR page 7), section number (e.g., TAAR sec 1.4), or similar.

1.5 Abbreviations and Acronyms

The following table summarizes abbreviations and acronyms used in the document.

Table 1-2 Abbreviations and Acronyms

Entry	Description
ACMS	Appeals Case Management System
BP	Business Process (BPs when plural)
BPE	Business Process Engine

Entry	Description
BPM	Business Process Model
BR	Business Rule (BRs when plural)
BRE	Business Rule Engine
BRM	Business Reference Model
CDSS	California Department of Social Services
COTS	Commercial Off the Shelf
EA	Enterprise Architecture
FSR	Feasibility Study Report
IADP	Implementation Advance Planning Document
IEEE	Institute of Electrical and Electronics Engineers
IVR	Interactive Voice Recognition
MOTS	Modified Off the Shelf
OSI	Office of Systems Integration
RA	Reference Architecture (RAs when plural)
SOA	Service Oriented Architecture

1.6 Document Organization

The following table summarizes organization of this document.

Table 1-3 Document Organization

Section #	Section Heading	Section Summary
2.	Executive Summary	Provides a summary of goals, approach, and outcomes.
3.	ACMS Functional Background	Provides relevant background information about the functional elements of ACMS and approach used to elicit and capture these elements.
4.	Adopted Technical Approach Overview	Summarizes the technical approach adopted in the document.

Section #	Section Heading	Section Summary
5.	Technology Options for ACMS	Identifies and discusses technical choices as specified by the ACMS Feasibility Study Report (FSR) and options aligned with CMS/CEAF/FEA.
6.	Proposed Technical Approach for ACMS	Formulates preferences and recommended choices for ACMS
7.	Proposed Technical Solution for ACMS – Supplemental Analysis	Introduces the comparisons and scoring for the main evaluation options available for ACMS. Focuses on comparing MOTS/Composite and Transfer System options with an emphasis on the ACCMS Transfer system at the AOC.
8. to 10.	Appendices	Provides reference material directly applicable to the technical approach.

2. EXECUTIVE SUMMARY

This TAAR seeks to meet two critical goals of the ACMS Project Acquisition Phase:

1. To clearly identify the required technical aspects of the ACMS.
2. To support the identification of the vendor/proposal most likely to meet those needs with lowest Total Cost of Ownership (TCO).

The recommended technical approach for ACMS can be summarized as follows and applies to all modes of provisioning as identified in the ACMS FSR (COTS, MOTS, Transfer, and Custom). The recommendations also apply to hybrid scenarios, in which different components are provisioned in a different way (some COTS, some MOTS, some Transfer, and some Custom in the most complex case). In the hybrid scenario, it is still preferred for all the components to be deployed in the same way.

The section "Proposed technology Solution – Supplemental Analysis" illustrates the presented approach in an expanded analysis that takes into consideration additional factors with respect to the technical approach discussed in the document. The analysis shows that utilizing a MOTS Software option based on Composite and Service Oriented Architecture (SOA) architectural principles best supports the CDSS current and future needs for an ACMS. In addition to the analysis detailed below, we have included additional evaluation criteria that also support this recommendation based on specific quantitative analysis.

A. Choose MOTS over COTS, Transfer or Custom solutions because:

- In contrast to COTS, MOTS are designed to be extensively modifiable, especially when the functional modifications of the product can be achieved using configuration rather than coding. This has significant positive impact on time-to-implementation and future sustainability of the solution.
- In contrast to Transfer solutions, MOTS are designed to be deployed and customized in various target environments rather than reflect specific needs of the single environment in which they grew, as is typically the case with Transfer candidates. Also, MOTS provides for better knowledge transfer and training than is usually the case with Transfer candidates.
- In contrast to Custom solutions, in the area of BPM, MOTS generally or Case Management more narrowly already offer significant functionality out-of-the box.

B. Choose Composite over Monolithic/Legacy Solutions because:

- Componentized solutions tend to have lower TCO over the lifecycle of the solution. They are more resilient to technology drift, as they make it possible to replace a single solution component (e.g., a reporting engine) with minimal impact on remaining solution components.

- Componentized solutions are easier to maintain, because their mutual interactions are visible and standardized (by implemented APIs) rather than hidden. Plus, the mutual interactions take place using a standardized mechanism (or a small number of such mechanisms).
- Componentized solutions are easier to show MITA- and CEAF compliance when compared to Monolithic solutions.

C. Choose a solution that makes it possible for user configuration rather than building software in order to address missing or changing functional requirements because:

- Configuration is less costly than software construction and requires less technical resources.
- Configuration provides for faster implementation of required functional requirements.
- Configuration provides for faster application modification in the face of changing requirements.
- The ability to maintain separation between functions and components (e.g., a component that provides for execution of business rules as contrasted with the set of ACMS-specific business rules) increases maintainability and sustainability of the ACMS solution.
- The separation capability has the potential to significantly increase the usable lifespan of the system compared to systems in which there is no such separation.

D. Choose a solution that is designed to make use of or integrate with infrastructure building blocks (such as Security Services or Application Integration Services) or directly provides infrastructure components as reusable components. Doing so decreases the costs related to ACMS maintenance and, longer term, for other related projects in the same organization (follow-on projects in the department, or agency).

E. Choose solutions that can be deployed to the cloud, if there is no cloud provider lock-in. Utilizing the cloud has a number of advantages over the traditional application hosting, as discussed in the section 5.1.3 on cloud deployments.

3. ACMS FUNCTIONAL BACKGROUND

This section provides background information about the functional elements of ACMS that are relevant to this document. The topics presented in this section include the following:

- Business objectives for ACMS and its expected business impacts
- Business-aligned approach to requirement elicitation for ACMS solicitation.

Business objectives have direct impact on formulation of functional requirements. They also have an impact on the overall technical approach (as introduced in the section "Adopted Approach Overview") in that only the candidate solutions for ACMS that are demonstrably capable of supporting realization of these objectives are to be considered, and the candidate solutions that do not should be discarded.

3.1 Business Objectives for ACMS and Its Expected Business Impacts

Providing claimants online, real-time access to their case information in ACMS is expected to improve case timeliness and mitigate penalties for late hearing decisions as result of the following:

- Decreasing hearing delays and associated penalties
- Reducing the staff workload related to phone calls from claimants for case status information
- Providing better information to claimants:
    - Leading to an increase in prehearing resolution of disputes without need for hearing
    - Leading to improved use of the hearing to focus on issues in dispute.

The ACMS IADP (section 7.6) lists the following Business Objectives for ACMS:

Table 3-1 ACMS Business Objectives

#	Objective
1	Reduce the average life cycle of an open Appeals Case, from receipt of the Hearing Request to release of the decision, by 14%, from 105 days to 90 days after one year of implementation.
2	Ensure 100% of notifications to the public are available in English and 12 other languages by first month of implementation.
3	30 days after implementation the three sub-systems/functions identified as Sound Recording App, Audio Transfer & Upload Log Database, and 100% of the functionality associated with them, will be available in a single consolidated process within the ACMS reducing processing time by 66% and freeing staff to perform other necessary duties.
4	6 months after implementation, reduce the amount of time spent by State Hearings Division (SHD) staff on a monthly basis specifically for the manual calculation and

#	Objective
	review of penalties due to untimely release of decisions from 65 hours to 20 hours, a decrease of 69%.
5	6 months after implementation the three sub-systems/functions identified as Decision System, Decision Archive and Decision Release, and 100% of the functionality associated with them, will be available in a single consolidated workflow process all within the ACMS, reducing the average decision processing time by 33%.

When considering ramifications of the above objectives on non-functional requirements, the following items stand out:

- The need to provide an *integrated workflow* solution in order to meet the overall automation, timeliness, and flexibility goals (objectives #3 and #5)
    - The "integrated" part in "integrated workflow" means that the integration is apparent to the user of the solution, and consequently, integrating separate building blocks as a way of providing for the end-user integration of business functions is not excluded from consideration
    - The "workflow" part in "integrated workflow" means that the core preoccupation for ACMS is ability to automate orchestration of repeatable patterns of business activity, which in turn is typically represented as sequences of operations or tasks and assigned to specific roles (groups or individuals) and referred to as "Business Process" definition.
- The need to provide for a flexible localization of generated documents to the public (such as Acknowledgement Letters) in 12 languages – with the understanding that the list of supported languages can change over time (objective #2)
- The need to gather key (definable) execution metrics during the execution of business functions and to report them in the form of dashboards and standard pre-defined reports (all objectives)
- The need to integrate ACMS with external components in the environment, such as Interactive Voice Recognition (IVR), audio recording facilities, e-mail servers, web servers, etc. (objective #3)

The above elements can be considered mandatory when evaluating competing solutions for ACMS. Absence of any of them in a proposed solution for ACMS undermines accomplishment of the related business goal(s) for ACMS.

3.2 Approach to Requirement Elicitation for ACMS Solicitation

For the sake of this document, Functional Requirements (as elaborated in Business Reference Model (BRM), Business Process Model (BPM) and related specifications) can be classified into a small number of groups of related items, as shown in the following table.

Table 3-2 Groups of ACMS Functional Requirements

Group	Examples

Maintenance of Case-Related Information	• Maintaining Case Lifecycle, including:
	  ○ Relevant information elements
	  ○ History of changes of the information
	  ○ Storing and archiving of closed/resolved Cases
	• Supporting annotations/notes for Cases and their elements
	• Integration of Cases with supporting digital documents
	• Linking related Cases

Calendaring of Hearings	• (Re-)Scheduling of hearings in conformance with applicable patterns of activity, resource availability and related business rules • Assigning resources to hearing on the day of hearing • Notifying all parties involved in a scheduled hearing while observing applicable rules (e.g., the minimal advance notice time)
Support for bi-directional communication with the Claimants	• Supporting interactions using phone calls and the IVR facilities whenever applicable (e.g., automated claim status queries) • Providing on-line access for Claimants using a Web site/portal, including user account for Claimants and personalization with applicable information (e.g., addresses, e-mails, designated representatives) • Creation of documents (such as Notices of Action) in any of the supported 12 languages
Monitoring, Tracking, and Performance Measurement	• Gathering business process/workflow executing performance data • Gathering vital statistics and metrics • Tracking of work item assignments and re-assignments • Generating notifications about delays in work item processing • Reporting on measurements and metrics
Business Process/ Workflow Capabilities	• Supporting work queues • Assigning work items to workers • Presenting status information in Dashboards (for County Workers, Judges, Management) • Generating notifications about completed steps • Generating warnings about impending delays

Reporting	• Generating pre-defined reports • Supporting definition and execution of ad hoc reports • Allowing for modifications of templates used for report generation
Group	**Examples**
Security-Related	• Authentication of system users • Access control to information elements, business processes and tasks within processes • Creating audit log of relevant user activities as per applicable security policy
Interoperability-Related	Interfacing (in Phase 2 of ACMS Project) with other systems of relevance to ACMS, including the following: • SAWS, • CalHEERS, • DHCS' SURGE system, • DSS applications.

The grouping of Functional Requirements as summarized above is further referenced in the section "Approach to Realization of Functional Requirements" later in the document.

4. ADOPTED APPROACH OVERVIEW

This section provides an overview of the technical approach adopted in this document, including the following facets:

- Driving principles for the TAAR
- Applicable background concepts and distinctions
- Importance of the way Functional Requirements for ACMS are *realized* for the TAAR.

The above areas are discussed in the subsections that follow.

4.1 Driving Principles

The approach adopted in the document is guided by the following principles:

- The evaluation of proposed solutions for ACMS must be open and not constrained by any pre-conceived or arbitrary exclusions.

- All potential solutions for ACMS must be evaluated using a single and consistent set of criteria. The challenge is to identify the set of criteria that can be consistently applied to candidate solutions for ACMS.

There are also a number of observations or lessons learned from large IT projects that inform the approach, as follows.

- All potential ACMS solutions will unavoidably incur costs, regardless of their type or source. Although the types of costs, their relative amounts and their distribution over time will vary from solution to solution.

- All (or virtually all) facets of a considered solution to be evaluated for ACMS have impact on one or more components of the Total Cost of Ownership (TCO) for that solution.

- Although specific ACMS solutions can have very different applicable TCO structures, it is still possible to group the solutions based on similar impacts of the architectural/technical approach they embody on specific elements of the TCO.

It should be noted that the provisioning options as identified in the ACMS FSR (COTS, MOTS, Transfer, and Custom) have impacts on TCO and Time-to-Implementation (TTI), but they are independent of the technical approach choices discussed in the document. In other words, the provisioning options do not determine the technical approach adopted in them and are not sufficient to formulate a technical approach to the ACMS solution.

As reference, with respect to technical/architectural groupings, the main useful distinctions are as follows:

- Monolithic vs. Composite Solutions
- Business vs. Infrastructure building blocks or components
- Business-Generic vs. Business-Specific building blocks or components

The above distinctions, as well as a small number of related concepts and terms, are described in the following subsection. The presented approach uses a number of concepts and distinctions used later in the document to further elaborate the topic. Some are based on standard distinctions as used in the industry, and some are developed through analyses in this document (such as analysis of technical solution types applicable for ACMS). The following subsections provide the details.

4.2 Functional vs. Non-Functional Requirements

The approach to elicitation of functional requirements (sometimes referred to as "Business Requirements") adopted in the ACMS project activities uses standard distinction between Functional Requirements and Non-Functional Requirements:

- Functional requirements are taken to represent the what part of the target system (ACMS)
- Non-Functional requirements are taken to represent the "how" part for ACMS - that is, how Functional Requirements are to be provided for and what are the constraints applicable to candidate solutions.

In ACMS Project, the functional requirements are expressed using the following models:

- Business Reference Model (BRM) which provides static view of requirements groups and their structure.
- Business Process Model (BPM) that provides dynamic, workflow-like view of requirements, including assignment of functional elements to specific Actors (participants in the processes supported by the ACMS).

BRM and BPM for ACMS (documentation for which is available from the ACMS Project SharePoint) have been developed with the following goals in mind:

- For inclusion in the solicitation documentation (including the RFP for ACMS)
- To be used as reference material for evaluating the level of fit for functional requirements in the solution options to be assessed
- To be used as a starting point for elaboration of detailed requirements in the ACMS Project after the solicitation phase.

Of the above goals, the first two are relevant to this document, solicitation and assessment of proposed solutions. The last one (elaboration of detailed Functional Requirements in their entirety) is not, at least beyond considering *representative* detailed requirements that substantiate the need for a given technical capability or capabilities in the technical solution to be evaluated.

4.3 Monolithic vs. Composite Solutions

Yet another distinction that is useful for evaluating proposed solutions for ACMS (and which can be treated as yet another pitfall pertaining to realization of functional requirements) is the distinction between two fundamentally different ways of architecting, designing, or building a solution:

- The approach resulting in Monolithic solutions
- The approach resulting on Composite solutions

The above approaches are characterized in the table below.

Table 4-1 Monolithic vs. Composite Solutions

Facet/Approach	Monolithic	Composite
Partitioning of the solution	Solution is not partitioned, as available as a single, *indivisible* application or system	A solution is *composed of* a number of discrete and visible parts (components or building blocks)
Implied concept of realization of requirements	All requirements are provided for (to the extent they are supported) by the solution as a whole	All requirements are provided for through *interaction* among relevant components.
External visibility of the internal processing	No visibility: no internal Application Programming Interfaces (APIs) (even if they exist) are externally visible	Interactions among discrete components are visible and typically standardized (e.g., adhering to explicit APIs)
Substitutability of internal components	Generally - not possible, or possible with high cost, long time and/or risk	Possible at the level of discrete components while limiting or removing the impact on other components, when the substitution conforms to the prescribed API.

The following diagram illustrates the above differences:

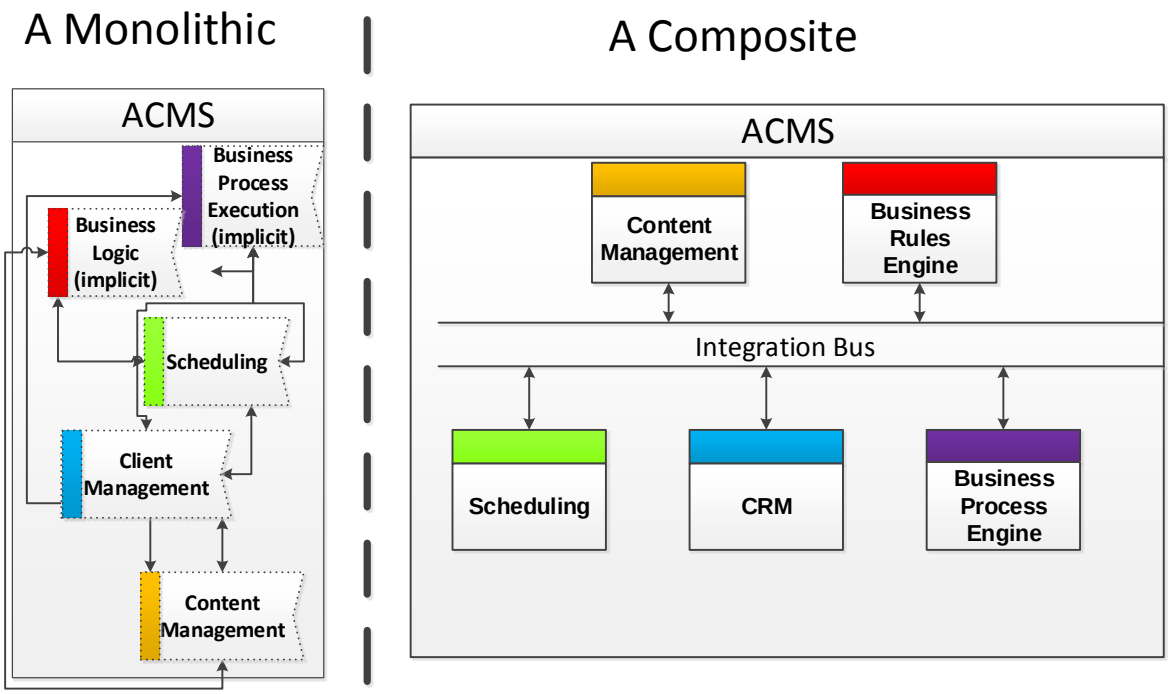

Figure 4-1 Monolithic vs. Composite Solution

The differences between Monolithic and Composite solutions are likely to manifest themselves when evaluating and comparing competing solutions for ACMS. A given Monolithic Solution and a Composite Solution may be equivalent with respect to a given set of functional requirements at the moment of acquiring them, but their other characteristics are likely to be quite different. For example, their respective abilities to meet changing requirements over time (and related costs, time, and risks) are bound to be different. The impacts of those differences on the TCO are discussed further in the document.

4.4 Business vs. Infrastructure Building Blocks

The distinction between business and non-business applies to building blocks of an application or system. All building blocks (including platforms, components and services) can be divided into the following:

- Building blocks that directly support realization of core business requirements, such as business process/workflow execution, support for nosiness rules, etc., further referred to as “Business” building blocks.

- Building blocks that enable the actual functioning of the Business building blocks, such as integration mechanisms for interactions among components or applications, or security services; they are further referred to as “Infrastructure” building blocks.

4.5 Business-Generic vs. Business-Specific Building Blocks

Within the Business building blocks, a further distinction can be made:

- Building blocks that support business functions generically, such as workflow engines or business rule repositories, further referred to as "Business-Generic" building blocks

- Building blocks that are specific to and reflective of the details of the functional requirements supported by the system, such as specific business processes definitions, specific business rules definitions, or dedicated business components, such as Scheduling/Calendaring component in case of ACMS.

For further discussion of the distinction in context of ACMS, please refer to the section "Approach to Realization of Functional Requirements in ACMS".

4.6 Realization of Functional Requirements

"Realization" of Functional Requirements indicates how the requirements in question can be fulfilled by a given technical solution. Realization of requirements is a key concept for the technical approach and satisfaction of requirements, as depicted in the following diagram:

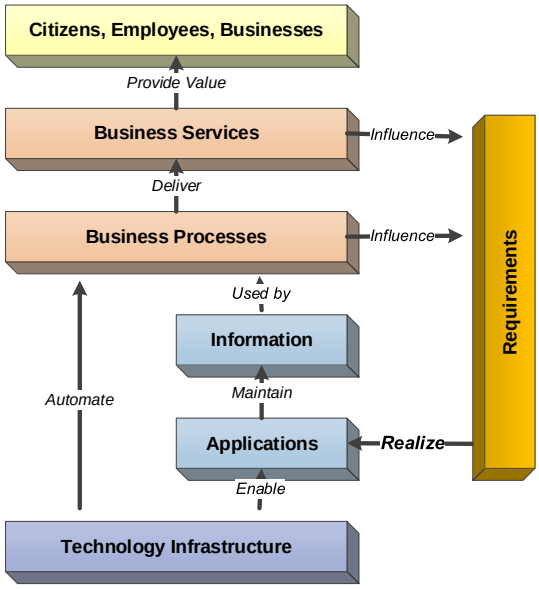

Figure 4-2 Architectural Context: Applications Realizing Requirements

The diagram above shows an overview of key domains as usually represented in Enterprise Architecture frameworks, such as MITA or CEAF. The diagram illustrates the place of "Applications" (such as an ACMS application) in context of the Requirements (that are realized by Applications) and between Technology Infrastructure on one hand and Business Process and Services on the other hand.

The discussion of realization of given requirements involves a number of challenges and related pitfalls. One challenge is that there are typically a number of valid ways of realizing a given functional requirement; for example, reporting requirements for ACMS can be realized using product A, product B, or through interactions involving some components X and Y. The related pitfall is the expectation that Functional Requirements alone that can positively identify or pick up specific valid technical realizations. This is not the case; rather, it is specific technical realizations that need to be examined for their ability to support given requirements, including Functional Requirements.

Another challenge is that some different ways of realizing given requirements may not be distinguishable one from another based solely on a facet of the external behavior of a component or application realizing the requirement in question. For example, response times as observed in product A when creating a given report and compared to product B may be equal for all practical intents and purposes. The related pitfall is to consider two realizations of a given requirement that are equivalent in some respect (such as response time or number of reports supported in the example above) to be generally equivalent and equally desirable in the adopted solution. This is an obvious mistake: even though the products A and B can be equivalent in a number of respects (e.g., functions supported), there are other respects in which they are not equivalent – for example, when it comes to respective licensing and maintenance costs.

The lessons learned from the items described above are reflected in the adopted approach and its insistence to identify all key aspects of realization that affect not only functionality of the solution, but also the TCO related to its adoption, directly or indirectly.

4.7 Approach to Realization of Functional Requirements in ACMS

In order to provide a common grounding for discussion of different types of realizations of functional requirements for ACMS and of their respective merits and disadvantages, the discussion of building blocks is kept at logical level – that is, presented in a way that is product- and vendor- neutral. The following *types* of *logical* building blocks can be distinguished as useful for the elaboration of technical approach to ACMS:

- ***Application-specific*** elements (such application-specific information structures or application-specific building blocks – in case of ACMS, the scheduling/calendaring component)
- ***Generic Business*** building blocks (such as workflow or business rules execution and management)
- ***Generic Infrastructure*** building blocks (such as application integration or security infrastructure)

It is important to note that both distinctions above (logical building blocks vs. technical components on the one hand, and application vs. business vs. infrastructure components) are applicable to IT applications in general, including MOTS, COTS, Componentized, or Hybrid solutions. What make these types of solutions different from one another are

factors such as which building blocks are provided at the acquisition time and the extent of modifiability of the provided building blocks.

4.8 The Role of Reference Architectures (RAs) and CEAF

Reference Architectures (RAs) are architectural-level specifications that provide architectural and logical solution blueprints, in the form of logical architectures and architectural patterns. These blueprints are of significant help in the solicitation process:

- To the solution requester, when mapping functional requirements with a technical approach and architectural blueprints
- To the solution provider, when mapping technical capabilities of particular solution to the logical architecture
- To the solution customer, to communicate what technical capabilities are needed in the solution and why.

The current 2.0 version of the California Enterprise Architecture Framework (CEAF) provides a number of Reference Architectures (for the current versions of CEAF Reference Architectures, please see:

http://www.cio.ca.gov/wiki/Enterprise%20Architecture%20Documents.ashx.)

CEAF 2.0 provides currently eight Reference Architectures. In varying degrees, all of the areas covered by the above Reference Architectures apply to ACMS. The following table summarizes which Reference Architecture covers specific capabilities group.

Table 4-2 ACMS-Related Capabilities as Covered by Reference Architectures

RA Moniker	RA Name	Capabilities Covered in the RA
BI	Business Intelligence RA	Reporting as subset of communication with ACMS users
CC	Cloud Computing RA	Cloud-based operation, management and scalability
EAI	Enterprise Application Integration RA	Integration Capabilities
ECM	Enterprise Content Management RA	Lifecycle of digital assets made available to ACMS users
eGov	eGovernment RA	Web-based interactions with ACMS users, including personalization
IdAM	Identity and Access Management RA	Security-related capabilities

RA Moniker	RA Name	Capabilities Covered in the RA
MDM	Master Data Management RA	Information and systems of record
SOA	Service-Oriented Architecture RA	Business Process execution and orchestration of services to support required functionality.

Even though RAs provide solution blueprints in the form of logical architectures, they take into consideration a number of relevant capabilities. For example, typical considerations of Security capabilities in context of a turn-key system are limited to authentication, authorization, and possibly Single Sign-On (SSO) capabilities. The corresponding CEAF RA is Identity and Access Management RA, and it considers also the following:

- Centralized life cycle management of a user's digital identity and attributes (including User and Group administration),
- Centralized administration of user entitlements/permissions
- Defining and enforcing Access Policies
- Provisioning of security for existing applications, etc.

All of the above capabilities are relevant for an acceptable ACMS solution, regardless of whether these security-related capabilities are to be provided by the ACMS application, the infrastructure facilities, or some mixture of the two.

4.9 Approach to Assessing Technical Suitability of Solutions for ACMS

When assessing suitability of various technical solutions for realization of a given set of functional requirements, the challenge is to provide a common ground that makes it possible to apply consistent, and non-arbitrary evaluation criteria. The proposed basis for that assessment is business alignment, involving the following:

- Consideration of costs and benefits over the full lifecycle of the solution considered (i.e., from the original provisioning of the system to its eventual retirement)
- Consideration of how specific types of solutions affect specific elements of TCO and/or ROI (when the level of functional fit is assumed equal).

4.10 Approach to Establishing Evaluation Criteria for ACMS

A three-step approach is recommended for the establishment of the evaluation criteria for the ACMS.

Step I Requirements Definition

Establish detailed list of the functional and non-functional requirements. The list of requirements documented will be reviewed and weighted according to the specific needs of CDSS for use as evaluation criteria in rating the responses. Specific evaluation criteria will be documented in the RFP.

Key actions:

- Identify unique functional and non-functional requirements
  Develop the unique business and system requirements for the ACMS. These requirements will be used in Step II as a basis for determining which tool provides CDSS the best fit to their requirements.
- Establish Critical Success Factors
  Meetings focused on communicating and affirming issues and critical success factors, understanding specific project expectations, and identifying how these will impact CDSS's organization and the selection project.

Functional and non-functional requirements coupled with the critical success factors will serve as a detailed checklist to guide and facilitate vendor evaluation.

Step II Request for Proposal Development and Execution

RFP will be sent to interested bidders, responses will be compiled and analyzed and a demo list of 2-3 vendors will be created. Step II will be completed by facilitating the vendor demonstration process, scoring and compiling of results. Specific evaluation criteria will be documented in the RFP.

Suggested Evaluation Process

- Develop RFP/Scorecard and Solicit Vendor Bids
  Vendors will be contacted and bids solicited from them.
- Facilitate Comparative Analysis of RFP Responses
  Review requirements and institute a ranking system to evaluate the vendor proposals. Collect vendor RFP responses and prepare a comparative analysis report. Utilize comparative analysis report to further condense candidate list to 2-3 vendors for demonstration.
- Facilitate Vendor Demonstration Sessions
  Invite top candidates to CDSS to present their system and to answer/clarify specific questions related to their RFP response. Develop demo scripts to evaluate vendors.

Step III Final Analysis and Recommendation

During the final step of the evaluation process, the project team finalizes the selection process as documented in the RFP, presents the ACMS system recommendation to executive management.

5. TECHNOLOGY OPTIONS FOR ACMS

Building on distinctions and concepts introduced in the previous sections, this section identifies and analyzes technology options for ACMS using the following steps:

- The discussion of options as identified in the ACMS FSR, namely COTS, MOTS, Transfer, and Custom.
- Identification of the types of Case Management-Related solutions available in the market and determining their characteristics with respect to flexibility in Business Process specification (and by extension, Business Rules specification).
- Discussing the perspective of generic vs. specialized business components in the context of ACMS.

5.1 FSR Technology Options

The ACMS FSR references a number of applicable ACMS solution options, including COTS, MOTS, custom, and "transfer" solutions (further referred to collectively as the "FSR Options"). The ACMS FSR also references two (2) applicable deployment options: cloud-based and traditional system deployments.

The types of solutions referred to in the FSR Options actually represent expectations or assumptions, primarily as to the level of available functionality in the system over time, rather than indicate or determine the types of technology (technical stack, architectural approach, etc.) that underwrite a given type of solution.

The FSR technology options are further referred to as "provisioning modes", because the differences between them are primarily related to the way the respective solutions are provisioned, with ramifications for licensing and ownership of source code. The provisioning modes are independent of the technical approach choices discussed in the document. In other words, any technical approach can potentially be provisioned as COTS, MOTS, Transfer, or Custom. Moreover, hybrid scenarios are also possible, where different components of the target solution are provisioned in a different way (e.g., some as COTS, some as MOTS, some as Transfer components).

Although the COTS/MOTS/Custom/Transfer distinction does not determine the underlying technological solution, it can still help us compare ACMS solutions from the perspective of the expected time-to-implementation or distribution of costs over time. The sections that follow discuss these issues.

The following tables provide a comparison of COTS/MOTS/Custom/Transfer provisioning modes: Table 5-1 shows main similarities and Table 5-2 shows differences among the options. The cloud/non-cloud option is treated separately, as it can be used in any of the COTS/MOTS/Custom/Transfer options.

Table 5-1 Typical Similarities among FSR Provisioning Modes

Option vs. Option	COTS	MOTS	Transfer
MOTS	• Proprietary products with licensing ramifications • Designed for deployment in various environments • Significant part of the target functionality already implemented		
Transfer	• Significant part of the target functionality already implemented	• Significant part of the target functionality already implemented	
Custom	None	None	• Source code available • Non-proprietary licensing

Table 5-2 Typical Differences among FSR Provisioning Modes

Option vs. Option	COTS	MOTS	Transfer
MOTS	• Support for configurability		
Transfer	• Transfer options are typically *not* designed for transfer, COTS are. Licensing	• Transfer options are typically *not* designed for transfer, MOTS are. Licensing	

Option vs. Option	COTS	MOTS	Transfer
Custom	• Extent of functionality available up-front. Extent of control over functionality. Extent of control over source code.	• Extent of functionality available up-front. Extent of control over functionality. Extent of control over source code.	• Extent of functionality available up-front

5.1.1 COTS vs. MOTS

For clarity, this document uses the following definitions of the terms "COTS" and "MOTS":

- Commercial-Off-The-Shelf (COTS) – Used to describe any pre-built solution in which significant part of the required system functionality is already present without new development. These systems typically incur their main costs early.

- Modified-Off-The-Shelf (MOTS) - Used to describe any pre-built solution in which the majority of the required system functionality is already present, and the remaining functionality is to be added via new development over time. These systems typically incur their costs over time, with most of the cost up-front and the remainder financing the required modifications.

The common features of COTS and MOTS include the following:

- Both types are typically commercial solutions, with ramifications in terms of acquisition costs (including the initial licensing costs), and on-going costs (license renewal costs, dedicated support costs, know-how transfer costs, etc.).

- Both types are typically delivered as closed-source solutions. This means they provide no access to their source code, and often with license terms precluding any form of reverse engineering of the product. On occasion it is possible to negotiate with the commercial vendor to have access to the sources, but this more an exception than the rule. Access to the source code significantly increases maintainability and sustainability of the system.

The following diagram depicts one difference between COTS and MOTS:

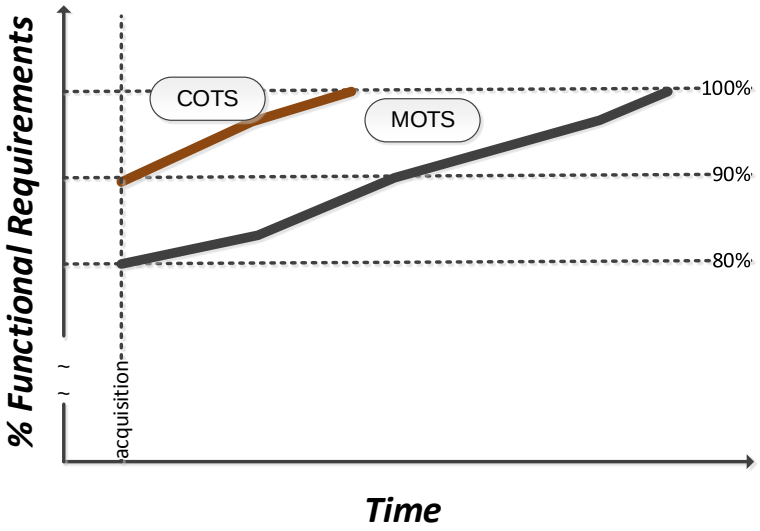

Figure 5-1 Functionality vs. Time in COTS vs. MOTS

The diagram above shows different levels (expressed as percentages for illustration purposes only) of expected fit between the functional requirements of the target system and the product, when classified as "COTS" or "MOTS". Even though the distinction is not informative with respect to the underlying technology choices, it indicates important business considerations, as follows:

- The time needed and the cost to the project until the system is deployed in production and fully operational.

- The (potentially indirect) cost to the business of delaying the production stage of the system in question, with the delay attributed to the time needed to provide for the requirements that are not supported out of the box.

5.1.2 Transfer vs. Custom

For clarity, this document uses the following definitions for the terms "Custom" and "Transfer":

- Custom – Custom development designates an option of building the ACMS solution from scratch, in-house or by a vendor.

- Transfer – The "Transfer" option indicates a scenario of re-using a similar system already in production in another department or agency within the state, with expected high level of functional overlap between it and ACMS.

The main difference between "Custom" and "Transfer" options is Time-to-Implement. Due primarily to the time required to develop from scratch, "Custom" solutions are expected to take longer to reach a usable level of functionality than "Transfer" solutions.

The following diagram portrays the difference in Transfer vs. Custom options:

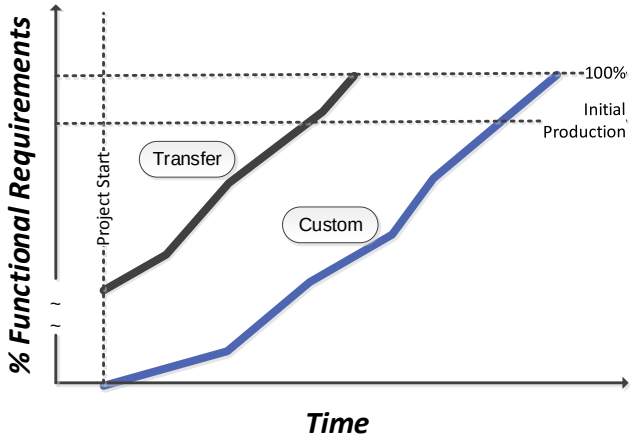

Figure 5-2 Functionality vs. Time in Transfer vs. Custom Options

The advantage that the Transfer option has over the Custom option is that typically the solution to be transferred is a custom solution that has already been built and deployed to production. This mitigates the risks and makes it possible to shorten the time frames.

The Transfer and Custom options share the following features:

- Licensing costs – Both types of solutions are expected to have very low or near-zero licensing costs, with possible exception of the components that need to be bought or added to satisfy the entirety of the requirements for the target system.

- Both types of solutions are expected to provide full access to the source code, often as a result of the organization currently using the solution already owning the sources (rather than solely the vendor owning the sources).

- Neither type of solution is typically designed to be transferred (in contrast to COTS/MOTS, which are designed for multiple deployments in varying environments in the first place).

5.1.3 Traditional vs. Cloud-based Deployment

The last pair of options mentioned in the ACMS FSR involves deploying to "the cloud" (or making use of "cloud computing") as opposed to deploying to the traditional data centers with dedicated hardware infrastructure and dedicated software, including system and application software.

Following the definition from the US National Institute of Standards and Technology (NIST), "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." Following NIST and the CEAF Cloud Computing Reference Architecture, the essential characteristics of cloud computing include the following:

- On-demand self-service, which allows consumers to unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each Service Provider.
- Broad network access, which means that cloud's capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.
- Resource pooling, which means that the provider's computing resources (CPUs, memory, storage, network bandwidth) are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.
- Measured Service that provides a metering capability at a level of aggregation that is appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts) in the cloud; this allows service consumers to pay only for what they use.
- Rapid elasticity allows for rapid provisioning of computing resources and subsequent releasing of them when no longer needed, in some cases automatically, to scale with demand.
- Multi-Tenancy allows for one instance of application serving multiple customers at the same time, while sharing cloud's resources. This sharing increases operational efficiency and decreases operation costs.

Cloud Computing offers a number of service models, depending on which elements in the hosted stack are transparently shared by users of the cloud:
- In Infrastructure-as-a-Service (IaaS) hardware resources can be shared.
- In Platform-as-a-Service (PaaS) processing and data can be shared.
- In Software-as-a-Service (SaaS) software architecture and application instances can be shared.

In general, any of the COTS/MOTS/Custom/Transfer options that can be deployed using the traditional hosting model can also be deployed in the cloud, at least in the IaaS service model. Both types of deployment satisfy basic requirements, such as the following:

- Access and data security (in case of the cloud and ACMS, public cloud installations are excluded, but private and governmental clouds are acceptable)
- Accessibility and recoverability
- Ability to cost multiple environments (Production, Development, UIT, Testing/QA)
- Operations monitoring

However, the cloud offers a number of advantages to the application owner compared to the traditional data center hosting. Most of these advantages reflect lower capital and operational costs, in case of the cloud, as follows:

- Cloud-based solutions are expected to have lower costs in the areas of capital expenses on infrastructure (cooling, power, or buildings) and hardware/software components, than traditional methods of deployment in non-cloud data centers.
- Given economies of scale, cloud-based solutions have lower operational costs than analogous dedicated Data Center operations.
- In cloud-based solutions, utilization can be measured and reported at a granular level. Consequently, the operational costs can reflect the actual utilization of the resources rather than a contracted amount independent of the actual utilization, as it is typically the case with traditional non-cloud deployments.

Moreover, the cloud makes it possible to achieve the following:

- To quickly meet increased computing power, storage, and/or network bandwidth needs and with easily determinable cost. Non-cloud deployments are likely to require more time and cost structure in order to meet the new demand.
- To deploy system/application versions quicker than in traditional deployment solutions by utilizing the standardized and automated deployment processes and tools.
- Depending on the type of cloud services provided for software components required by the application in question (such as databases, application servers), further economies of scale are achievable, when the cloud provider can provide SaaS for the application.

In case of solutions that can be deployed in the traditional Data Center and to the cloud, the preference should be given to the deployment in the cloud, given the advantages pointed out above. ACMS does not have any identified concerns (such as security concerns, reliability or access times) that preclude it from being successfully deployed in the cloud.

In a case where the ACMS solution can be used *only* when deployed to the cloud, the important decision factor is a viable exit strategy. If the solution requires a specific type of cloud creating long-term dependency on the service provider (e.g., because of technically or contractually difficult porting of the application to a different cloud provider), then the advantages of such a solution for the business should be weighed against the long term ramifications of the dependency.

5.2 Advantages and Disadvantages of COTS/MOTS/Custom/Transfer Options

The preceding subsections provide characterization of COTS/MOTS/Custom/Transfer and point out their main similarities and differences. The following table points out main advantages and disadvantages of the discussed options.

Table 5-3 Main Advantages and Disadvantages of COTS/MOTS/Custom/Transfer Options for ACMS

Option	Main Advantages	Main Disadvantages
COTS	• High expected functional fit • Designed for installation in various environments • Maturity of commercial product • Support availability	• Adding missing functionality requires programming • Adjusting to changing requirements can be costly • Internal architecture and design can be documented but not fully accessible or exposed • Typically proprietary, closed implementation • Proprietary licensing – ramifications to costs and reuse
MOTS	• High expected functional fit • Improvement over COTS with respect to configurability • Designed for installation in various environments • Maturity of commercial product • Support availability	• The extent of configurability has limits and may not be sufficient for the requirements at hand, resulting in potentially complex programming • Internal architecture and design can be documented but not fully accessible or exposed • Proprietary, closed implementation • Proprietary licensing – ramifications to costs and reuse
Transfer	• High expected functional fit • Low acquisition costs • Pre-existing and potentially accessible user base	• Not designed for installation in various environments, and may require reverse engineering and documentation efforts • Limited support available • Challenging synchronization of updates with the organization that initially created the solution • Details of licensing and ownership make take time to finalize.

Custom	• High expected functional fit • Simplified ownership • Usually, no licensing fees *directly* associated with Custom code (fees and licenses may be still associated with *components required* by the solution) • The extent of control over source code	• Problematic extent of functionality available up-front, depending on re-used components if any • Time-to-Implementation (TTI) – typically longer than in COTS or MOTS

The following table provides a more detailed comparison of several facets characterizing COTS/MOTS/Custom/Transfer options.

Table 5-4 Characterization of COTS/MOTS/Custom/Transfer Options for ACMS

Option/Facet	COTS	MOTS	Transfer	Custom
Expected level of maturity of the application	High	High	Varies, but not as high as COTS/MOTS	Low (depending on what is reused), increasing over time
Expected level of functional fit at the time of acquisition or project start	Very high, preferably 90% or more	High, preferably 80% or more	High, preferably 80% or more	Low to none (depending on what is reused)
Type of solution license	Proprietary	Proprietary	State-Internal	State-Internal
Expected cost and time needed to provide the remaining functionality	High: extensions to the existing product typically require software	Medium: some functions may be obtained by configuring the product,	Depends on the architecture/design of the system	Depends on the architecture/design of the system

Option/Facet	COTS	MOTS	Transfer	Custom
	construction	other may still require software construction		
Expected cost and time needed to accommodate future changes to business processes and rules	High: extensions to the existing product typically require software construction	Medium: some functions may be obtained by configuring the product, other may still require software construction	Depends on the architecture/design of the system	Depends on the architecture/design of the system
Has been designed for deployment in different organizations and environments	Yes	Yes	Typically, no	Typically, no
Has been architected for compliance with MITA	Varies but not very likely in case of mature products	Varies but more likely than COTS	Varies but not likely. Making it compliant can be cost-prohibitive	Can be architected to be compliant
Potential for the application or its components to be reused	Small, limited by internal design and/or licensing issues	Small, limited by internal design and/or licensing issues	The potential exists but is limited by existing the design	The potential exists and it can be enhanced by appropriate design
Requires license fees for the product	Yes	Yes	No	No
Requires license fees for required components (e.g., database, application server)	Yes	Yes	Yes	Yes

Option/Facet	COTS	MOTS	Transfer	Custom
Available Documentation for Maintainers or Programmers	Good (typically)	Good (typically)	Varies, may become good over long period of time	Varies, may become good over long period of time
Access to the application source code	No (typically)	No (typically)	Yes	Yes
Cost directly related to application upgrades	Depends on licensing, non-negligible	Depends on licensing, non-negligible	M&O costs only – no licensing ramifications and access to source code	M&O costs only – no licensing ramifications and access to source code
Cost indirectly related to application upgrades	Not easily foreseen nor controlled - breakage of application specific extensions is possible	Not easily foreseen nor controlled - breakage of application specific extensions is possible	Foreseeable and controllable	Foreseeable and controllable

The following table summarizes typical risks associated with the COTS/MOTS/Transfer/Custom options.

Table 5-5 Representative Risks in COTS, MOTS, Transfer, and Custom Options

#	Risk	Applicable To	Consequences	Mitigation
R1	Inaccurate assessment of technical capabilities of the proposed solution	All	• Significant TCO increase • Problematic compliance with standards (MITA, CEAF) affecting budget in some cases	• Gaining insight into the architecture of the solution, if possible • Formulating RFP questions to clarify the internals
R2	Inaccurate assessment of the functional fit with ACMS	COTS and MOTS	The expected time-to-production and overall cost grow, possibly beyond the justification of having the COTS/MOTS in the first place	• Gaining insight into the architecture of the solution, if possible • Formulating RFP questions to clarify the internals
R3	Inaccurate assessment of the functional and environmental fit with ACMS	Transfer	Increase in the time-to-production and cost reducing the benefit of the low to zero licensing costs.	• Examination of available documentation • Analysis of source code
R4	Inaccurate assessment of the flexibility to meet changing business requirements	All	• High cost of providing for realization of changing requirements • Long term, a risk of the solution becoming economically unviable	• Gaining insight into the architecture of the solution, if possible • Formulating RFP questions to clarify the internals
R5	Insufficient organization (including skills) to support maintenance and operations	All	• Problematic availability and sustainability of the solution • Long term, a risk of the solution becoming economically unviable	• Considering required organizational change as part of the assessment of choices • Executing a roadmap to provide required organizational capabilities.

5.3 Types of Case Management-Related Solutions

Given that the COTS/MOTS/Transfer/Custom options do not provide sufficient basis for analyzing or prescribing technology approach to ACMS, a limited survey of the existing applicable solutions for ACMS currently available has been performed, using the following sources:

- Responses to the ACMS RFI.
- Market research, including materials published by organizations (such as Gartner or Forrester).
- Interviews with technical staff of similar systems in the State.

The objective of the above efforts was to identify available types of Case Management-related solutions rather than to assess specific products. The following groups of Case Management solutions can be identified:

- *Domain-specific Case Management Solutions* (D-CMS), which support requirements specific to a given domain, such as Legal or Social/Medical Services- centered domains.
- *Generic* (i.e., not domain-specific) *Case Management Solutions* (G-CMS), which aim at supporting typical case-based business processing scenarios and typically provide some level of customization within the pre-defined scenarios.
- *Generic Business Process Management Solutions* (G-BPM), which are designed for automation of most types of business processes, rather than specifically case-centered variations of it.

The types of solutions listed above differ in their focus (the type of problems the respective solutions attempt to solve) and ability to address changing requirements, as summarized in the following table.

Table 5-6 Main Focus of CMS-Related Solution Types

Facet/Solution Type	Domain CMS (D-CMS)	Generic CMS (G-CMS)	Generic BPM (G-BPM)
(Legal) Domain Focus	Yes	No	No
Sample Products or Solution	ACCMS (transfer system)	Microsoft Dynamics CRM	RedHat BPM Suite
Case Management Focus	Typically limited to the understanding of Case specific to the domain or, more narrowly, in the application	Yes, delimited by the supported case scenarios or processes	No, but can support Case Management when the appropriate workflows are defined

Facet/Solution Type	Domain CMS (D-CMS)	Generic CMS (G-CMS)	Generic BPM (G-BPM)
Focus on implementation of specific Business Processes and Business Rules	Possibly, but the process and the rules are typically enclosed in the black box	Yes, with the extent limited by configurability as provided by the grey box	Yes, white box visibility[2]
Focus on declarative implementation of flexible *Business Processes and Business Rules*	No	Limited	Yes

Note that the sample solutions or products mentioned under "Sample Products or Solution" are provided solely for illustration purposes rather than as an indication of their fitness for purpose (which remains to be established).

The above table introduces a basic grouping of CMS-related solution types. Further survey of the market offerings and materials published by product survey companies (Gartner, Forrester) expand this basic grouping in the following way:

- There is an increased visibility of "flexible" Business Process Management solutions, which are characterized with growing emphasis on configurability (as contrasted with customization requiring programming) of business processes and business rules.

- There are products that overlap categories (e.g., have some features of CMS-centric solution and some features of more general BPM-centric solutions) the main strengths of which are based on their past evolution, e.g., from Content Management Systems or Customer Relationship Management (CRM) systems.

Based on research published by Gartner in 2014, the following features characterize modern CMS:

[2] The difference indicated is in terms of external visibility of how Business Processes and Business Rules are conceptualized and implemented in the system: "black box" means that the internal is not externally visible and consequently not open to configuration. The "grey box" means that some but not all of the aspects are externally visible, and consequently some but not all aspects are open to configuration. The "white box" means that the implementation is externally visible and configurable at least in principle, typically as a consequence of a design decision to make them so, which is absent from the previous two options.

Table 5-7 Features of Contemporary CMS

#	Feature	Description	ACMS FSR
F1	Supports a broad range of content types and content interaction services	Case management requires a broad range of data types, from highly structured data (such as an individual's name, address or ID) to highly unstructured data (such as scanned images, faxes, email communications, and audio or video files).	Deploy an IVR that provides 24 hours/7 days telephone access to benefit applicants/recipients.
F2	Support broad range of collaboration services to facilitate individual and group interactions among all case participants	Supporting collaboration interactions as part of the audit trail of a case. Multiple technologies are typically integrated to support collaborative interactions, including scheduling, email, chat, text messaging, social media and e-rooms.	Provide online web data input, review, or case status by benefit applicants/recipients, Authorized Representatives and other stakeholders.
F3	Interoperates well with other external content and process services	Case handling is often constrained by rules, which may be managed in an external rule engine and thus shared with the content management framework, as well as with other applications.	The new ACMS will have the capacity to interoperate with other systems including CalHEERS, SAWS and DHCS' SURGE system.

#	Feature	Description	ACMS FSR
F4	Provides vertical- and horizontal-specific data models, nomenclature, hierarchies and case life cycle management	Architecture of specific data models, nomenclature, hierarchies and case life cycle management impacts the implementation time.	A best of suite approach will allow CDSS to implement the leader in their field, rather than try to encourage a solution provider to develop functionality that is not part of their current solution. A best of suite solution was identified through a market analysis that demonstrated clear leaders whose products delivered most, if not all, components of the ACMS functionality.
F5	Provides application adapters to industry- and domain-specific environments, including legacy and Web data	Solution must integrate with critical systems of record, including legacy, industry, Web and social data sources.	Provide the capacity for secure interfaces with CalHEERS, SAWS Consortia and DHCS SURGE and HHS systems.
F6	Provides comprehensive and highly configurable role-based user experiences	Support for role-based user interfaces or workbenches focus and simplify case handling is a critical step toward productivity gains. Simplifying the interface between case workers, the content in cases and the managers who make decisions based on the work in progress is a crucial factor for success.	1. Provide adequate information security controls and role based access. 2. Improve reporting functions through the expansion of case identification parameters and a Management Reporting module.
F7	Leverages models for easy adaptability of the solution	Solution should enable business and technical roles to easily adjust their solutions as needed.	The solution should be sustainable long term.

#	Feature	Description	ACMS FSR
F8	Provides business-role-friendly dashboards, metrics and reporting	Solution should provide open and easy access to a database of case execution history, as well as appropriate dashboards, models, visualizations, reports and other tools to monitor, analyze and report on case handling.	Reduction of processing time as a business goal requires gathering of metrics and reporting for process improvements.
F9	Supports a broad range of case orchestration, from highly structured to highly unstructured flows	Solution must provide case orchestration for policy driven and structured (predictable sequences of activities, usually represented in a flow model) workflows.	Implement cohesive and intuitive workflows. Decision System, Decision Archive and Decision Release, and 100% of the functionality associated with them, will be available in single consolidated workflow process all within the ACMS.
F10	Has been **proven** in deployments with 100,000 cases or more annually	Case management solution should support large-scale case handling.	Have adequate capacity to process projected hearing volumes. Address the consolidation of the SHD main case management database (HWDC) with 22 ad-hoc systems into one comprehensive case management system. Capture all necessary information to perform SHD business.

#	Feature	Description	ACMS FSR
F11	Provides intelligent and versatile on-ramps and off-ramps	Solution must have ability to deliver various inbound content objects to a case folder with the ability to generate outbound content (e.g., acknowledgement letters with multi-language support) to customers, such as reports, pre-composed letters, correspondence or statements. Also important is the ability to export case data using a variety of outbound file formats.	1. Federal Affordable Care Act "No wrong door" policy. 2. Achieve language requirement compliance

The surveyed types of solutions differ in the extent of flexibility to meet changing requirements and in the level of the functional fit expected "out-of-the-box" for a given solution type, as shown in the diagram below.

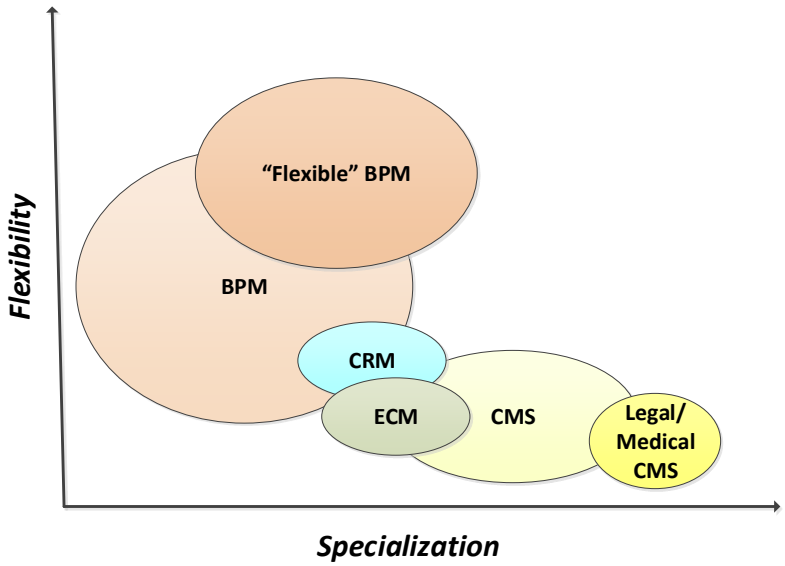

Figure 5-3 Specialization vs. Flexibility in Types of Case Management Solutions

The following table provides a description of the solution types portrayed in the above diagram and provides examples of products illustrating the classification. The examples included are solely for illustrative purposes.

Table 5-8 Sub-Types of Case Management Solutions in the Marketplace and Examples

#	Type	Sub-Type	Description	Example(s)
1.	D-CMS	L-CMS	*Legal domain-specific* case management systems	OpenJustitia
2.	G-CMS		Generic case management systems	IBM Curam
3.	G-CMS	CMS-ECM	Case management systems based on or evolved from ECM solutions	IBM Case Manager (FileNet-based)
4.	G-CMS	CMS-CRM	Case management systems based on or evolved from CRM solutions	Microsoft CRM Dynamics
5.	BPM		Generic BPM systems allowing for easy definition and modification of business process specifications	Apache Activiti
6.	BPM	FBPM	Generic BPM with emphasis on declarative specification of business processes, workflows, and business logic.	RedHat BPM Suite

5.4 CMS Solution Types and Expected Level of Fit for ACMS

The types of the CMS solutions identified above provide a high-level grouping of potential choices for ACMS. These groups are further characterized with respect to features that affect the expected level of functional fit with ACMS functional requirements, and with the (architectural) constraints on the solution introduced by the conformance with MITA. The following table provides a summary of the facets for each of the groups identified.

Table 5-9 BP/BR Flexibility in CMS-Related Solution Types

Facet/Solution Type	Domain CMS (D-CMS)	Generic CMS (G-CMS)	Generic BPM (G-BPM)
Flexibility with respect to supported information structures	Low for specialized solutions	Varies depending on the extent of customization allowed in the product	High
Expected level of fit of information structures relevant to ACMS	High, but varies in practice	Moderate to low, varies in practice	Low to none, the structures need building
Flexibility with respect to supported workflow and processes	Low for specialized solutions	High within constraints of predefined case types, otherwise low	High
Expected level of fit of the supported business process with ACMS	High, but varies in practice	Moderate to low, varies in practice	Low to none, the specific process needs defining
Flexibility with respect to supported business rules	Low for specialized solutions	Medium to High, but in practice varies depending on the adopted approach to Business Rules	High, but Business Rules need defining
Expected level of fit of the business rules with ACMS	High, but varies in practice	Moderate to low, varies in practice	Low to none, the specific Business Rules need defining

The above table identifies technical/architectural facets for evaluating proposed solutions for ACMS.

6. PROPOSED TECHNICAL APPROACH TO ACMS

The discussion of COTS/MOTS/Transfer/Custom options (presented in section 5.1 "FSR Technology Options") evaluated provided these high level recommendations:

1. Select a composite solution.
2. Select a solution that is designed to make use of or integrate with infrastructure building blocks.
3. Select a solution that makes it possible for user configuration rather than building software in order to address missing or changing functional requirements.
4. Select a solution that can be deployed to the cloud if there is no cloud provider lock-in.

6.1 Summary of Recommendations

The recommended technical approach for ACMS can be summarized as follows:

A. Choose MOTS over COTS, Transfer or Custom solutions because:

- In contrast to COTS, MOTS are designed to be extensively modifiable, and especially when the functional modifications of the product can be achieved using configuration rather than coding, this has significant positive impact on time-to-Production and future sustainability of the solution.
- In contrast to Custom solutions, MOTS in the area of Business Process Management (BPM) generally or Case Management more narrowly already offer significant functionality out-of-the box.
- In contrast to Transfer solutions, MOTS are designed to be deployed and customized in various target environments rather than – as it is typically the case of Transfer candidates – reflect specific needs of the single environment in which they grew. Also, MOTS provide for much better knowledge transfer and training than it is usually the case with Transfer candidates.

B. Choose Composite over Monolithic Solutions because:

- Componentized solutions tend to have lower TCO over lifecycle of the solution – they are more resilient to technology drift, as they make it possible to replace a single component of the solution (e.g., a reporting engine) with minimal impact on remaining components of the solution.
- Componentized solutions are easier to maintain, because their mutual interactions are visible and standardized (by implemented APIs) rather than hidden, and because they take place using a standardized mechanism (or a small number of such mechanisms).
- Componentized solutions are easier to show to be MITA- and CEAF-compliant than Monolithic ones.

C. Choose a solution that makes it possible to user configuration rather than building software in order to address missing or changing functional requirements because:

- Configuration is cheaper than software construction and requires less technical resources.
- It provides for faster implementation of required functional requirements and for faster modification of the existing application in face of changing requirements.
- Ability to maintain the two separately (for example, a component that provides for execution of business rules as contrasted with the set of ACMS-specific business rules) increases maintainability and sustainability of the solution for ACMS.
- The separation has the potential to significantly increase the usable lifespan of the system implementing it compared to systems in which there is no such separation.

D. Choose a solution that is designed to make use of or integrate with infrastructure building blocks (such as Security Services, Application Integration Services) or directly provides an infrastructure components as a reusable component because doing so decreases the costs related to maintenance in case of ACMS and, longer term, for other related projects in the same organization (follow-on projects, department, or agency)

E. Choose solutions that can be deployed to the cloud if there is no cloud provider lock-in. Using the cloud has a number of advantages over the traditional application hosting, as discussed in the section 5.1.3 on cloud deployments.

Using the above approach, and the comparisons and analysis presented in section "Proposed technology Solution – Supplemental Analysis" show that utilizing a MOTS option based on Composite and SOA principles best supports the CDSS current and future needs for an ACMS. In addition to the analysis detailed below, we have included additional evaluation criteria that also support this recommendation based on specific quantitative analysis
The subsections that follow discuss the background of the above recommendation in context of ACMS.

Additional information detailing the recommendation is listed below,

- Monolithic vs. Composite solutions (as introduced in section 4.3 "Monolithic vs. Composite Solutions")
- Generic vs. Specialized building blocks (as introduced in section 4.4 "Business vs. Infrastructure Building Blocks")
- Business vs. Infrastructure building blocks (as introduced in the same section).

6.2 TCO and Monolithic vs. Composite Perspective on ACMS

Total Cost of Ownership of any solution for ACMS involves a number of costs beyond the initial costs of acquiring the solution. The following figure (based on a study entitled

"Total Cost of Ownership "and published by the Government of New Zealand) identifies main elements of the cost:

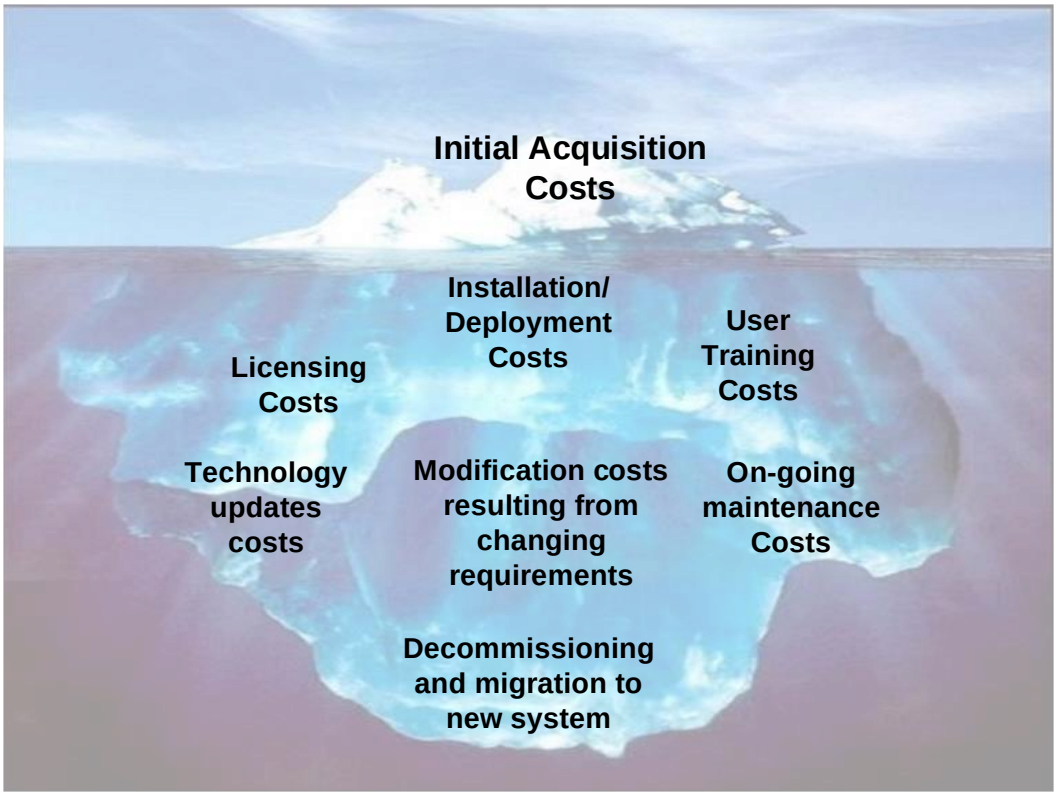

Figure 6-1 Elements of TCO for an IT Solution

The figure shows the following main components of the TCO in case of an IT system:

- Initial acquisition costs
- Installation and deployment costs
- Licensing costs
- User training costs
- Costs resulting from technology updates
- On-going maintenance costs
- Costs resulting from changing requirements (e.g., changing laws and regulations, or Business Process evolution)
- Costs related to decommissioning of the system and migrating to a new system

Various provisioning options (COTS/MOTS/Transfer/Custom) typically involve different mixes of the portrayed costs: in one type of solution, the cost of acquisition may be low compared to other solutions, but at the same time the cost of its installation and user training may be much higher than in other choices. For an evaluation of different provisioning options, please see Section 7, Supplemental Analysis. This section is concerned with the relationship between two types of technical approaches and relevant

elements of the TCO. The types of solutions used for discussion have been introduced in Section 4.3 "Monolithic vs. Composite Solutions", and are as follows:

- Monolithic solution, as exemplified by the traditional, proprietary, turn-key systems
- Composite solution, as exemplified by contemporary service- and component-based systems.

The aspects of TCO directly affected by the above technical approaches include the following:

- **Maintainability**, which represents solutions ability to meet new requirements and address defects expeditiously
- **Leveragability** and re-usability, which represents the potential of decreasing the cost and time-to-market for future systems requiring some of the functions already realized in the current system
- **Integrateability**, which describes ability to interact with other systems, components, or services
- **Sustainability**, which determines economically viable lifespan of the system.

In all above areas, composite solutions typically produce better TCO than functionally equivalent monolithic solutions. In all these areas, the root cause for this is the same: ability to orderly decompose potentially complex systems into smaller components, with all interactions regulated by an explicit API. Composite solutions make this possible, whereas monolithic solutions do not.

The above problematic is further discussed in Section 5 "Technology Options for ACMS".

6.3 Specific vs. Generic Component Perspective on ACMS

Following the distinction of specific generic infrastructure building blocks as described in the preceding subsection, this section discusses such building blocks in context of ACMS. The following diagram provides an illustration of ACMS based on the grouping of functional requirements as shown in "Table 3-2 Groups of ACMS Functional Requirements" in the section "ACMS Functional Background" and the discussion of ACMS-relevant solution types as presented in the preceding subsections.

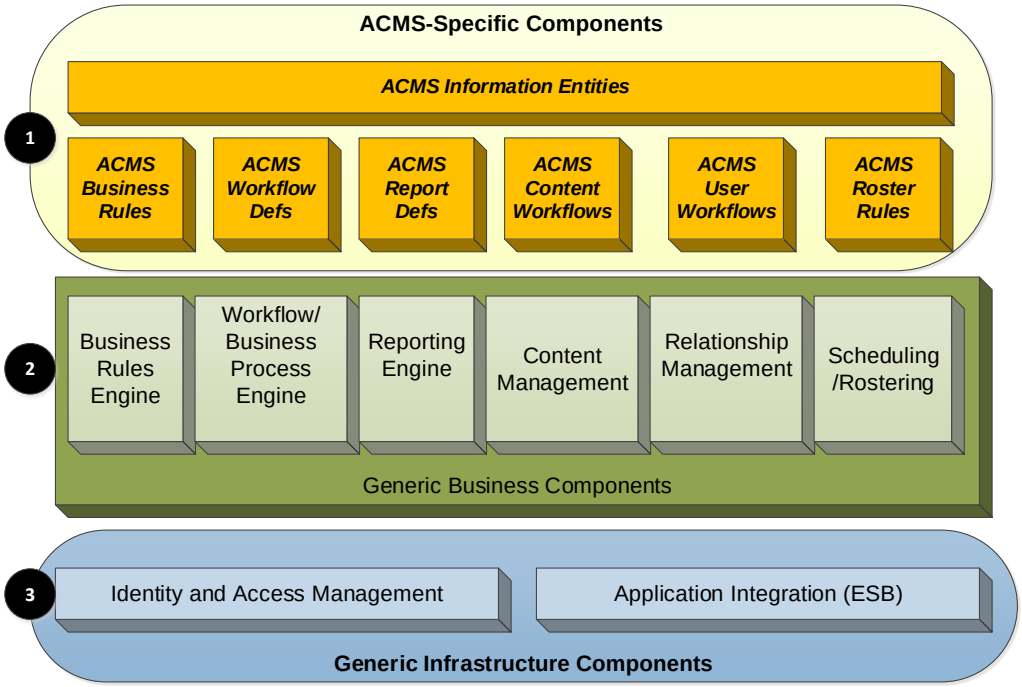

Figure 6-2 Infrastructure, Generic Business, and ACMS-Specific Components

The following table provides a description of the numbered elements shown in the diagram.

Table 6-1 Groups of ACMS Functional Requirements

Marker	Description
(1)	The group of ACMS-specific components (i.e., such that are not likely to be realized by any *generic* business components), including the following: a. Definition of ACMS business processes and workflows b. Definition of ACMS business rules (as applied in the processes) c. Definition of ACMS-specific scheduling patterns and business rules affecting calendaring and roster (re)generation. d. ACMS-specific Information Entities (aka "Object Model") – shown in "Figure 6-3 ACMS-Specific Information Model and Their Relationships" below.
(2)	The group of Generic Business Components, including the following: a. Business Process Engine (capable of executing ACMS business processes) b. Business Rule Engine (capable of executing ACMS business rules) c. Reporting Engine (capable of producing ACMS-defined reports against ACMS data structures)
(3)	The group of Infrastructure Components, including the following: *a.* Identity and Access Management *b.* Application Integration (ESB)

The following table provides an overview of characterization of monolithic vs. componentized solutions in terms of the types of building blocks used and their organization.

Table 6-2 Building Blocks in Monolithic and Componentized Solutions

Facet/Solution Type	Monolithic	Componentized
Architecture has *logical* building blocks	Varies	Yes
Solution has *physical* building blocks	No (typically)	Yes
Provides generic business building blocks	No (typically)	Yes (typically)
Provides generic infrastructure building blocks	No (typically)	Flexible in providing the infrastructure building blocks or in interfacing to the environment.

6.4 Information Structures in ACMS

Following well-established software development methodologies (including elaboration of Technical Architectures as presented in MITA 3.0), it is important to emphasize the importance of the solution being able to support specific information structures. The term "Information Structures" means here logical level data structures that are required in ACMS to support lifecycle of Claim information and all related information elements. An initial analysis of BRM and BPM for ACMS suggests the following types of structures:

- Entities (such as Claimant, Case, Party, etc.) which represent bundles of data pieces that belong together, exist independently of other entities and which are processed as such.
- Groupings of Entities, using kind-of relationship (e.g., "Withdrawal" and "Decision" entities can be thought of as a kind-of "Outcome" entity).
- Associations among Entities including their type (one-to-one, one-to-many associations, etc.).

The above structures are illustrated in the following Unified Modeling Language (UML) diagram:

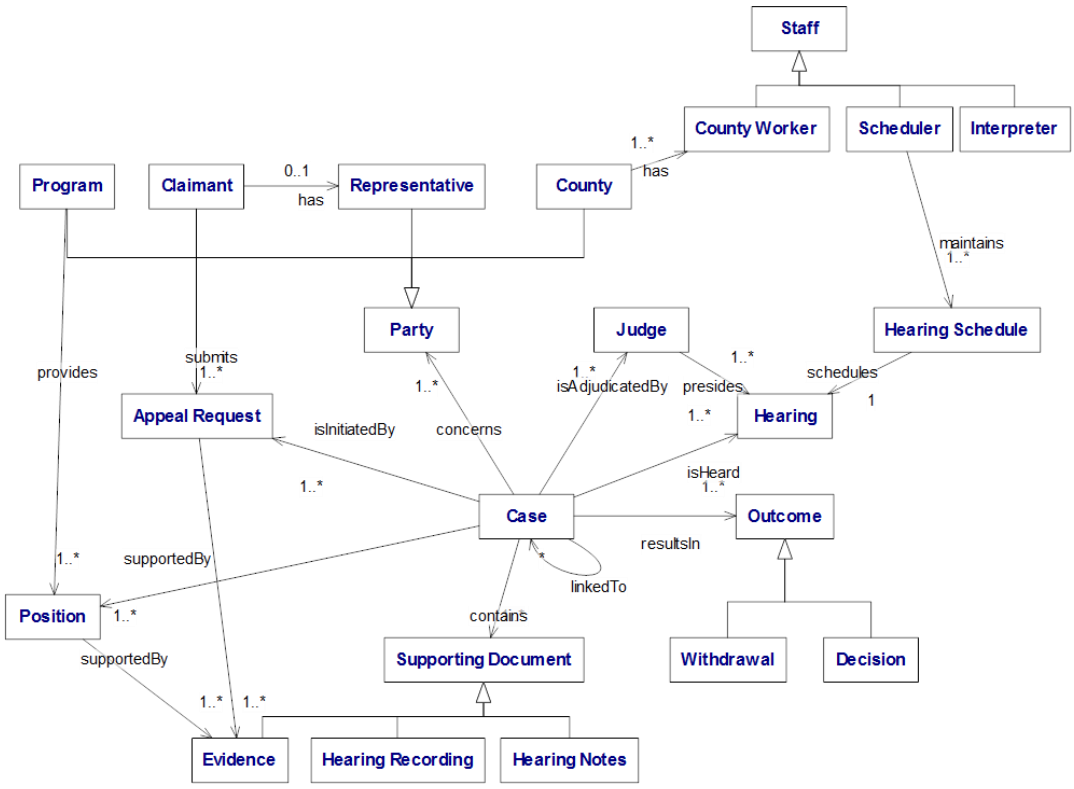

Figure 6-3 ACMS-Specific Information Model and Their Relationships

Even though the information structure represented in the above diagram is likely to undergo modifications, it is an important piece to consider when assessing the level of fit between the information entities and their relationships as supported in a candidate solution for ACMS. Potential mismatches, and especially inability to express some of the required relationships (or reasonable workaround for that inability), spell protracted customization and high cost of adoption of the solution in question.

6.5 Technical Capabilities Required for ACMS

In addition to applying the distinction between ACMS-specific and generic logical building blocks for ACMS as shown in the subsections above, the adopted approach makes use of Technical Capabilities required for realization of functional requirements for ACMS.

Technical Capabilities of a solution determine if that solution can support a group of functional requirements or business process or processes at all, and if yes, to what degree this is possible. For example, if a given solution does not have a capability to support a workflow execution, then automation of manual tasks is not likely to be satisfactory without incurring significant cost. If a solution implements a workflow, but does it in a hard-coded way, then matching the existing practices with the workflow in the solution is problematic, and so is adjusting of the workflow to new requirements; both shortcomings can be usually addressed, but at a cost. The following diagram shows groups of technical capabilities that are required for realization of functional requirements for ACMS:

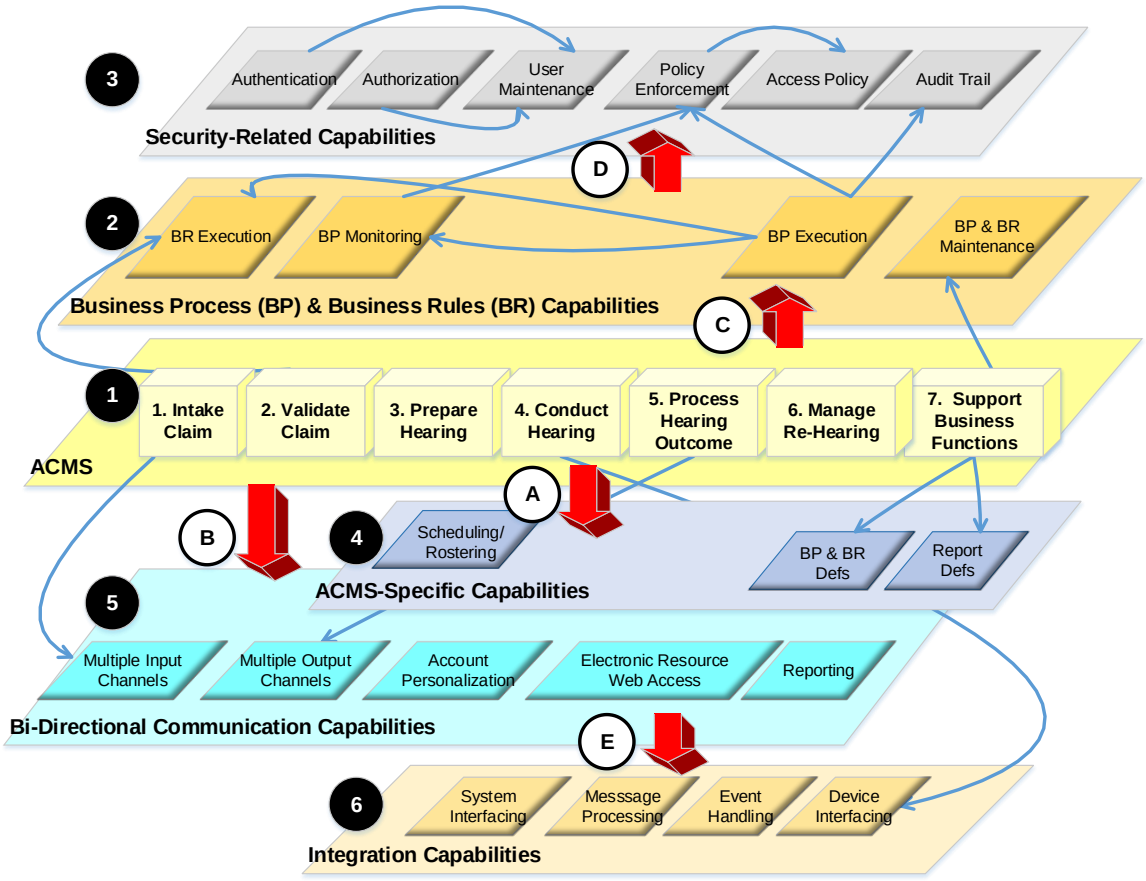

Figure 6-4 Groups of Capabilities for ACMS

The items identified by digits (1 to 6) are described in the following table. The arrows marked by letters (A to E) indicate dependency relationship between two groups of capabilities. Note that the order of markers (digits or letters) is not intended to suggest any form of sequence in time.

Table 6-3 Groups of Capabilities for ACMS and Representative Functional Requirements

#	Capability Group	Representative Corresponding Functional Requirements
1	ACMS as a whole	Top-Level functional components as specified by the ACMS BRM and BPM
2	Business Process and Business Rules-Related Capabilities	• ACMS shall use explicit Business Rules when executing Business Processes, including data validations, deciding about flow, or exception handling • ACMS shall provide for management of Business Rules, including creating, modifying, and retiring Business Rules • ACMS shall allow for declarative definition of sequencing of tasks/operations in order to increase maintainability of the system

3	Security-Related Capabilities	• ACMS shall provide for identification of all users requesting access to the system • ACMS shall protect access to resources and information based on user privileges, including HIPPA-compliant access control
4	ACMS-Specific Capabilities	• ACMS shall support full lifecycle of Claim-related Information Entities. • ACMS shall provide Scheduling and Roster capabilities • ACMS shall support ACMS-specific definitions and configurations of workflow, business rules, and reports
5	Bi-Directional Communication Capabilities	ACMS shall support interacting with Claimants using a number of channels, including the following: • Web Page submission (electronic document(s)) • E-mail • Letter • Phone call • Request/meeting with Intake Worker • Electronic request from another Application/System
6	Integration Capabilities	ACMS shall support integration and interaction with other application or systems, including electronic transmission of relevant information, in synchronous, asynchronous, or batch modes as appropriate for the given interaction.

7. PROPOSED TECHNOLOGY SOLUTION – SUPPLEMENTAL ANALYSIS

This section provides a practical illustration of the approach elaborated in the document and provides an evaluation of **two types** of options that are of interest to CDSS: a MOTS/composite-based solution and a transfer option, as exemplified by the Appellate Court Case Management System (ACCMS). The ACCMS is only an example of a transfer system option. There are other Transfer systems in production, but the ACCMS is the one for which we obtained documentation required for this analysis. The evaluation is made in context of the overall key business priorities for ACMS, as follows:

- Meeting Business needs – now and into the future
- Supporting workflow/business process-based organization of work
- Capable of sustaining the target expected workloads
- Supporting collection of metrics in order to measure performance and introduce process improvements
- Supporting no wrong door policy (including IVR)
- Compliant with applicable standards (such as Section 508, CEAF, CMS)
- Interoperable with external systems (such as CalHEERS or SAWS)
- Supporting digital content (such as digital documents, audio, image, etc.)
- Supporting calendaring of hearings (collaborative/flexible)
- Multi-Language (supporting 13 total languages when generating specific documents)
- Sustainable (state staff)

This evaluation includes elements beyond the domain of the technical. The following figure shows facets taken into consideration for the proposed technology solution evaluation:

Figure 7-1 Context for Evaluation of the Proposed Solution

In this evaluation, the term "MOTS/Composite" combines two elements endorsed by the technical approach analysis performed in the main document:

- A MOTS (Modifiable-Off-the-Shelf) product is typically a commercial off-the-shelf software product that is architected and built to provide a level of customization, either by configuring the product or by coding customizations using prescribed APIs.
- Composite architectural approach (as discussed in Sections 4.3 and 6.2) that emphasizes creating the solution for ACMS using discrete building blocks with well-specified APIs. Note that SOA is a form of the Composite approach, in which the component is viewed as a provider of services and potentially consumer of other services as provided by other components.

7.1 Evaluation Process

The presented evaluation process involved the steps as follows:

1. Establishing a set of evaluation criteria and, as appropriate, dividing the criteria among a set of categories.
2. Determining a scheme for scoring products against the evaluation criteria.
3. Providing a set of numerical weights to determine the relative importance of the criteria and evaluation categories.
4. Computing the overall score for each product.

The subsections that follow provide further description.

7.1.1 Evaluation Areas and Criteria

In the evaluation of the proposed solution, the areas taken for consideration and as a source of applicable evaluation criteria are shown in the following figure:

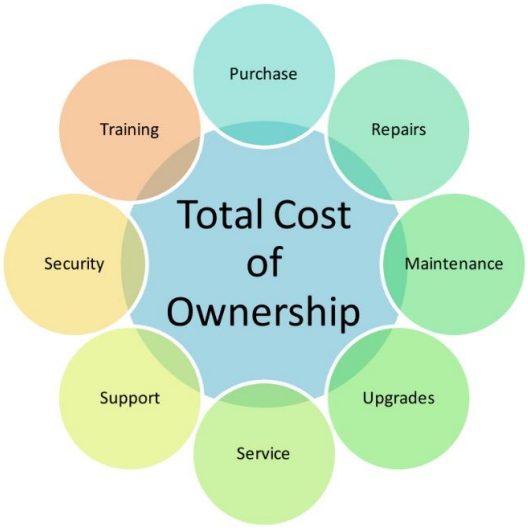

Figure 7-2 Proposed Solution Evaluation Areas

The corresponding evaluation criteria include the following TCO-related elements:

- Cost of acquisition ("Purchase"), including cost of licenses and fees.
- Cost of Maintenance that involves costs related to maintaining the product in operation without changing its functional requirements.
- Cost of Upgrades, including direct licensing costs, and indirect costs resulting from the need to upgrade other components dependent on the product being upgraded.
- Cost of Support, either provided directly by the vendor of the product or by a contracted third party.
- Cost of Security, including making the product compliant with the applicable security policies and testing of the outcome.
- Cost of Training, including accessing relevant training materials, courses or expert know-how if needed.

7.1.2 Scoring Scheme

The adopted scoring scheme used a multi-attribute utility (MAU) analysis – it enables straightforward, rigorous, and consistent decision making. In this type of analysis, by convention, any scoring function is normalized so that the scores for individual areas being scored fall in the range from 0 to 1. The simplest practical evaluation scale in this approach uses the following values:

- 0 if a product does not meet evaluation criteria.
- .5 if a product partially meets evaluation criteria.
- 1 if a product fully meets evaluation criteria.

These are not the only possible scale values. It is possible to use a larger discrete set or a continuous set of values, as long as they are between 0 and 1. However, using a finer scale often does not change the overall outcome of the comparison.

7.1.3 Evaluation Weighting Factors

All evaluation criteria fall into one of the five main evaluation areas, as shown in the following figure:

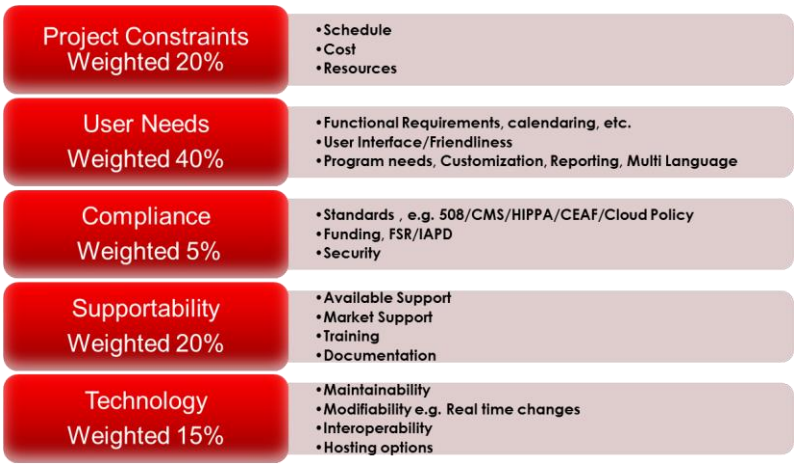

Figure 7-3 Evaluation Weighting Factors

The specific weights can be adjusted if needed, although the proposed weighting reflects reasonable hierarchy of priorities in a project like ACMS.

7.2 Comparing MOTS/Composite and ACCMS

This section describes comparisons applicable to MOTS/Composite vs. ACCMS that provide the basis for the actual numerical scoring of the competing choices. The comparisons include discussion of the following areas:

- Comparison of key advantages and disadvantages
- Comparison of key risks

The above comparisons are presented in the subsections that follow.

7.2.1 Key Advantages and Disadvantages

The following table summarizes respective key advantages and disadvantages for the two compared solutions.

Table 7-1 MOTS/Composite and ACCMS Key Advantages and Disadvantages

Solution	Advantages	Disadvantages
MOTS/ Composite	• Current technology that's compliant with CMS principles and CA IT Policy. • Supports automation required by DSS workflows and business rules. • Long term sustainability of the solution • Can meet all current CDSS stated FSR & functional	• MOTS product is not a turn-key system • Licensing costs and license legal terms may have adverse ramifications for flexibility in deployment and in reusing solution's components in related projects.

Solution	Advantages	Disadvantages
	requirements. • Case management framework. Out-of-the-box design patterns and workflows	
ACCMS	• Solution has been proven in production at the AOC. • Solution has overlaps in basic functionality with requirements identified for DSS. • User interface design in ACCMS in use for years by a sizeable user community. • Configurability of classifications and tags when handling Case-related information.	• Does not support automation using workflows for stated performance goals. • Does not *currently* support the levels of load required. (The current processed loads are less than 1% of ACMS). • Does not support access to the system for the Public. • Lacks support for multiple languages. • Developed as an in-house tool *specific* to Appellate Court Requirements rather than for transfer to other environments.

7.2.2 MOTS/Composite and ACCMS Key Risks

The following table summarizes respective key risks for the two compared solutions. Note that even though adoption of the MOTS/Composite solution is not free of risk, the number of risk-related entries is substantially lower than for ACCMS.

Table 7-2 MOTS/Composite and ACCMS Key Risks

MOTS/Composite	ACCMS
• Initial complexity: most MOTS solutions are complex and may require significant high level expertise and manufacturer support to configure and to start using the product efficiently • Availability of relevant skills	• Technical Knowledge Transfer – Some of the time saved by leveraging an existing system will be lost due to knowledge transfer required to the new project team. (Core reason – not designed to be a transfer solution) • Solution **does not meet** Federal architecture/California Enterprise Architecture standards and may place project schedule and funding at risk due to justification and exemptions that may be required to proceed with the ACCMS option. • Solution **does not have core functionality required by the FSR**, such as workflow

MOTS/Composite	ACCMS
	capability, multi–language support, IVR integration, etc. • Even understanding of the actual implementation of ACCMS (**Reverse Engineer)** (including its business logic and business model) requires significant effort. • Modifying ACCMS to add core required functionality is likely to **require re-architecting and re-designing the system**, and is at risk to be more costly and take more time than creating a custom solution. • Availability of relevant skills: ACCMS uses ColdFusion templates, a dated technology, and its business logic is expressed using Oracle's stored procedures (PL/SQL). Although PL/SQL skills are available, the market for PL/SQL applied to business logic is small. • Scope Definition – The project scope will need to be more clearly defined up front to help set expectations and clarify the desired feature set; this requires more upfront detailed project planning.

7.3 Scores for MOTS/Composite and ACCMS

The tables in this section provide individual and weighted scores for the MOTS/Composite and ACCMS options, against other potential choices as identified in the RFI process for ACMS.

The following table provides final scores for all considered options for ACMS.

Table 7-3 Summary of Scores

	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
Total Weighted Score	**3**	**2.9**	**1.7**	**1.575**	**3.5**	**1.975**	**1.9**

The following table provides detailed scores for all considered options for ACMS.

Table 7-4 Detailed Scores

#	Evaluation Criteria	Description of Criterion	Weight	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
1	**Project Constraints**		**20%**	**0.1**	**0.1**	**0.1**	**0.1**	**0.3**	**0.4**	**0.2**
1.1	Budget	Evaluation of Total Cost of Ownership, including the licensing costs, implementation costs, and maintenance and hosting operational costs.		0.5	0.5	0	0	1	1	1

#	Evaluation Criteria	Description of Criterion	Weight	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
1.2	Project Implementation Schedule	Evaluation of the level of effort to complete the Design, Development, and Implementation Phase of the solution by 10/2017.		0	0	0.5	0.5	0.5	1	0
2	**User Related**		**40%**	**1.2**	**1.2**	**0.8**	**0.6**	**1.6**	**1.4**	**0.4**
2.1	Functional Requirements	Evaluation of the best fit solution for the identified functionality required to meet program need. This criterion refers to evaluating how well the system meets the predefined functional requirements		1	1	0.5	0	1	1	0
2.2	User Friendliness	Evaluation of the user interface and the adaptability of users to the new system.		1	1	0.5	0.5	1	1	0.5
2.3	Customization Requirements	Evaluation of the level of customization that would be required to accommodate the business need.		0	0	0.5	0.5	1	0.5	0
2.4	Reporting and Analysis Capabilities	Evaluation of the reporting and analysis capabilities that the solution offers.		1	1	0.5	0.5	1	1	0.5

#	Evaluation Criteria	Description of Criterion	Weight	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
3	**Compliance**		**5%**	**0.1**	**0.1**	**0.075**	**0.05**	**0.1**	**0.075**	**0.05**
3.1	Compliant with Technology Standards	Identification of the standards the solution is required to be in compliance and comparing the conformance.		1	1	0.5	0.5	1	0.5	0.5
3.2	Compliant with Legal and Regulatory policies	Evaluation of application support of legal and regulatory policy.		1	1	1	0.5	1	1	0.5
4	**Technology**		**15%**	**0.9**	**0.9**	**0.525**	**0.525**	**0.9**	**0.45**	**0.45**
4.1	Maintenance	Evaluation of the factors that impact maintainability, including, hardware and software platform support, code and configuration management support and knowledge base.		1	1	0.5	0	1	0.5	0.5

#	Evaluation Criteria	Description of Criterion	Weight	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
4.2	System Requirements	Evaluation of the system requirements as they relate to the risks and tradeoffs reflected in architectural decisions relating to nonfunctional requirements, including performance, scalability, reliability, security, and technology maturity.		1	1	0.5	0.5	1	0.5	0.5
4.3	Real Time Changes	Evaluation of the ability to make changes to the application, specifically the business rules.		1	1	0.5	0.5	1	0	0.5
4.4	Modifiability	Evaluation of the ability to make changes to a system quickly and cost-effectively.		1	1	0.5	1	1	0.5	0.5
4.5	Inter-operability	Evaluation of approach to address disparate technology and data formatting requirements to interoperate.		1	1	0.5	0.5	1	0.5	0.5
4.6	Hosting & Back-up System Options	Evaluation of hosting and back up platform options to address replication of servers, firewalls, proxies, and load balancers.		1	1	1	1	1	1	0.5

#	Evaluation Criteria	Description of Criterion	Weight	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
5	**Support Related**		**20%**	**0.7**	**0.6**	**0.2**	**0.3**	**0.4**	**0.5**	**0.2**
5.1	Supportability	Evaluation of the supportability of the solution, satisfying any necessary needs or requirements, but also the provision of equipment, support infrastructure, additional software, facilities, manpower, or any other resource required to maintain the software operational and capable of satisfying its function. Including operational aspects associated to the installation, loading (or unloading), configuration, error recovery and execution of the software, and modification (often mistakenly called Software Maintenance) related to the evolution of the software due to the need of fixing bugs, or adding/changing functionality due to changing user needs.		1	0.5	0.5	0	0.5	0.5	0.5

#	Evaluation Criteria	Description of Criterion	Weight	Custom Build - J2EE	Custom Build - .NET	COTS - JACS	COTS - Salesforce	MOTS - Modifiable/ Composite	Transfer System - ACCMS	Transfer System - HCART
5.2	Market Support	Evaluation of market resources available to support the solution. Including resource skills availability, software forums, software API's.		1	1	0	0.5	0.5	1	0.5
5.3	Training Availability	Evaluation of market/vendor training options. Including instructor led, web based, self-paced training options, third party training channels, etc.		0.5	0.5	0	0.5	0.5	0.5	0
5.4	Documentation	Evaluation of documentation of the solution. Including requirements documentation, architecture/design documentation, technical documentation, and user and administrator documentation.		1	1	0.5	0.5	0.5	0.5	0
	Scores	***% Weighted Average***		**3**	**2.9**	**1.7**	**1.575**	**3.3**	**2.825**	**1.3**

8. APPENDIX A: APPLICABLE STANDARDS AND GUIDELINES

The ACMS-applicable standards and guidelines are identified below. They are grouped based on whether they are international, federal, or state-based standards or guidelines. The tables below identify applicable standards based on this grouping.

8.1 Applicable International Standards and Guidelines

Table 8-1 Applicable International Standards and Guidelines

Standard/ Guideline	Area of Application	Description
ISO/IEC 25051	QA, Requirements	”Software product Quality Requirements and Evaluation”, which defines quality requirements for COTS software products; it replaces the ISO/IEC 12119
ISO/IEC/IEEE 42010	Architectural Specifications	“Systems and software engineering”, which standardizes architectural descriptions of software systems; it replaces the widely known IEEE 1471.
ISO/IEC 25010	Assessment of product quality and of quality-in-use	Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models; replaces ISO/IEC 9126-1:2001

8.2 Applicable Federal Standards and Guidelines

Table 8-2 Applicable Federal Standards and Guidelines

Standard/ Guideline	Area of Application	Description
CMS	Solicitation, Provisioning, EA	CMS Seven Conditions and Standards (MITS-11-01-v1.0) version 1.0 as of April 2011
MITA 3.0	EA, esp. Business Processes	Medicaid Information Technology Architecture (MITA), version 3.0 as of February 2012
FEAF	EA, Provisioning	Federal Enterprise Architecture Framework

8.3 Applicable State-Level Standards and Guidelines

Table 8-3 Applicable State-Level Standards and Guidelines

Standard/ Guideline	Area of Application	Description
CEAF 2.0	EA	California Enterprise Architecture Framework, version 2.0 as of November, 2013.
CDSS ISP	Information Security	CDSS Information Security Program
SAM 5300	Information Security	The state's information security program guide developed by the California Office of Information Security

9. APPENDIX B: ISO 25010 TECHNICAL CHARACTERISTICS

The following table describes facets of technical characteristics as identified by the ISO 25010 standard.

Table 9-1 ISO 25010 Technical Characteristics

Attribute	Sub-Attribute	ISO Description
Functional Suitability		Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions
Functional Suitability	***Functional Completeness***	Degree to which the set of functions covers all the specified tasks and user objectives.
Functional Suitability	***Functional Correctness***	Degree to which a product or system provides the correct results with the needed Degree of precision
Functional Suitability	***Functional Appropriateness***	Degree to which the functions facilitate the accomplishment of specified tasks and objectives.
Performance Efficiency		Performance relative to the amount of resources used under stated conditions.
Performance Efficiency	***Time behavior***	Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements.
Performance Efficiency	***Resource utilization***	Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.
Performance Efficiency	***Capacity***	Degree to which the maximum limits of a product or system parameter meet requirements.
Compatibility		Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.
Compatibility	***Co-existence***	Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product

Compatibility	***Interoperability***	Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.
Usability		Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.
Usability	***Appropriateness recognizability***	Degree to which users can recognize whether a product or system is appropriate for their needs.
Usability	***Learnability***	Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use
Usability	***Operability***	Degree to which a product or system has attributes that make it easy to operate and control.
Usability	***User error protection***	Degree to which a system protects users against making errors.
Usability	***User interface aesthetics***	Degree to which a user interface enables pleasing and satisfying interaction for the user.
Usability	***Accessibility***	Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.
Reliability		Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. Limitations in reliability are due to faults in requirements, design and implementation, or due to contextual changes.
Reliability	***Maturity***	Degree to which a system, product or component meets needs for reliability under normal operation.
Reliability	***Availability***	Degree to which a system, product or component is operational and accessible when required for use.

Reliability	***Fault tolerance***	Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.
Reliability	***Recoverability***	Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system.
Security		Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.
Security	***Confidentiality***	Degree to which a product or system ensures that data are accessible only to those authorized to have access.
Security	***Integrity***	Degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.
Security	***Non-repudiation***	Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.
Security	***Accountability***	Degree to which the actions of an entity can be traced uniquely to the entity.
Security	***Authenticity***	Degree to which the identity of a subject or resource can be proved to be the one claimed.
Maintainability		Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.
Maintainability	***Modularity***	Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.
Maintainability	***Reusability***	Degree to which an asset can be used in more than one system, or in building other assets.

Maintainability	***Analyzability***	Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.
Maintainability	***Modifiability***	Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.
Maintainability	***Testability***	Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.
Portability		Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.
Portability	***Adaptability***	Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.
Portability	***Installability***	Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment.
Portability	***Replaceability***	Degree to which a product can replace another specified software product for the same purpose in the same environment.

10. APPENDIX C: SAMPLE TECHNICAL SCORING CRITERIA

This section provides sample scoring criteria for characteristics described in Appendix B. Even though the samples uses scale from 1 to 4, the score values can be easily mapped onto the 0-1 range as used in the evaluation section.

Table 10-1 End-User Response Performance Scoring Criteria

Score	Basis	Notes
1	No data provided for configuration of the target system to meet user response targets	
2	Configuration meeting the user response targets is known for less than 50% concurrent users	
3	Configuration meeting the user response targets is known for 50 to 75% concurrent users or more	
4	Configuration meeting the user response targets is known for more than 75% concurrent users	

Table 10-2 Approved Operating Systems Compatibility Scoring Criteria

Score	Basis	Notes
1	No information provided or does not use any of the approved operating systems	
2	Requires one from the approved operating system	
3	Can be deployed on two different approved operating systems	
4	Can be deployed on more than two different approved operating systems	

Table 10-3 Supported Databases Compatibility Scoring Criteria

Score	Basis	Notes
1	No information provided or requires a database that is not on the approved list	
2	Requires a specific database listed as approved	
3	Capable of configuring to work with any two of the databases listed as approved	
4	Capable of configuring to work with more than two of the databases listed as approved	

Table 10-4 Types of exchanges with other systems Interoperability Scoring Criteria

Score	Basis	Notes
1	No information provided	
2	Batch-based ETL only	
3	Supports standard messaging API (e.g., JMS) with Message Oriented Middleware (MOM) products	
4	As in #3 and supports web-services API	

Table 10-5 Solution as a consumer of external services Interoperability Scoring Criteria

Score	Basis	Notes
1	No information provided	
2	Requires building proprietary adaptors to become a consumer of external services	
3	Requires building standard messaging API adaptors or web services clients to become a consumer of external services	
4	Already has at least one messaging API-based client for external services, and at least one web service client	

Table 10-6 Solution as a provider of services Interoperability Scoring Criteria

Score	Basis	Notes
1	No information provided	
2	Requires building proprietary components to become a service provider	
3	Allows for exposing services using standard APIs (messaging API or web services)	
4	Already exposes services for external consumption using standard APIs (messaging API or web services)	

Table 10-7 Input Channels Available Usability Scoring Criteria

Score	Basis	Notes
1	No information provided	
2	Indirect (telephone or letter) access only	
3	As in #2, and on-Line access	
4	As in #3 and IVR access for designated functions (such as checking the appeal status)	

Table 10-8 Output Channels Available Usability Scoring Criteria

Score	Basis	Notes
1	No information provided	
2	Hardcopy documents only (printed, mailed)	
3	As #2 above and on-line querying facilities for claimants, including checking appeal status and access to applicable forms	
4	As #3 above and support for outgoing emails, automated phone calls	

Table 10-9 User Perspective Integration Scoring Criteria

Score	Basis	Notes
1	No information provided	
2	Different roles must use different user interfaces or applications to access the system	
3	There is a single user interface for all business roles in the system or the solution supports Single Sign On (SSO)	
4	As in #3, and the user interface in the solution can be configured in function of the user role and/or special requirements of the user	

Table 10-10 User Workflow Scoring Criteria

Score	Basis	Notes
1	No information provided or there is no support for workflows	
2	Some support for workflows exists, but it requires coding to modify it	
3	Extensive support for workflows exists, but not all of the workflows can be modified through changing configuration rather than programming	
4	Support for workflows exists and workflows can be modified through changing configuration rather than programming	

Table 10-11 Multiple Languages Support Scoring Criteria

Score	Basis	Notes
1	No information provided or there is no support for languages other than English	
2	Adding support for a new language requires programming	
3	Adding support for a new language requires some programming and some configuration changes	
4	Adding support for a new language does not require coding and can be achieved by modifying configuration	

Table 10-12 Contextual Help Learnability Scoring Criteria

Score	Basis	Notes
1	No information provided or no contextual help	
2	Some contextual on-line help in English	
3	Complete contextual on-line help in English	
4	Complete contextual on-line help with ability to support multi-language help for Claimants	

Table 10-13 Beginner Personalization Learnability Scoring Criteria

Score	Basis	Notes
1	No information provided or no support	
2	Incomplete but still covering main usage scenarios	
3	Yes – *either* covering 80% or more of usage scenarios *or* configurable – allowing non-programmers to expand the coverage of on-line help	
4	Yes –covering 80% or more of usage scenarios *and* configurable – allowing non-programmers to expand the coverage of on-line help	

Table 10-14 Learning Environment Availability Scoring Criteria

Score	Basis	Notes
1	No information provided or no learning environment	
2	Yes, but covering less than 80% of usage scenarios	
3	Yes: *either* covering at least 80% of usage scenarios *or* configurable with respect to usage scenario coverage and training data	
4	Yes: covering at least 80% of usage scenarios *and* configurable with respect to usage scenario coverage and training data	

Table 10-15 Cue Availability Operability Scoring Criteria

Score	Basis	Notes
1	No information provided or no identifiable cues	
2	Cues – such as breadcrumbs – are available for some steps in some processes	
3	Cues – such as breadcrumbs – are available for all steps in some processes	
4	Cues are available for all steps in all processes and their presentation is configurable	

Table 10-16 Work Queue Operability Scoring Criteria

Score	Basis	Notes
1	No information provided or does not provide	
2	Provides a view for some roles but not all	
3	Provides the view for all roles	
4	Provides configurable view for all roles	

Table 10-17 Defective Inputs Error Protection Scoring Criteria

Score	Basis	Notes
1	No information provided or no identifiable visual cues on defective inputs	
2	Partial validation for required data types (date, phone, SSN, etc.) or the presentation of the cues is not configurable	
3	Presentation of the cues is configurable	
4	As #3 above and complete validation of required data types (date, phone, SSN, etc.)	

Table 10-18 Application Error Messaging Scoring Criteria

Score	Basis	Notes
1	No information provided or no communication	
2	Application produces a visual message to signal error condition to users	
3	As in #2, and the application writes to a server-side log	
4	As in #3 above, and the application notifies support/help desk with data identifying the event	

Table 10-19 UI Consistency Support Scoring Criteria

Score	Basis	Notes
1	No information provided or Inconsistent presentation, no UI guidelines/standards, and no mechanism for enforcing consistency in the UI (such as support for CCS)	
2	Inconsistent presentation or no UI guidelines, but has a mechanism for enforcing consistency in the UI (such as support for CCS)	
3	Has a mechanism for enforcing consistency in the UI (such as support for CCS) and consistent presentation, but no UI guidelines/standards	
4	The UI is consistent, configurable, and supported by existing UI guidelines/standards.	

Table 10-20 Unit Testing Testability Scoring Criteria

Score	Basis	Notes
1	No information provided or no unit tests available	
2	Unit tests are present but for some components only	
3	Unit tests are present for all components	
4	Unit tests are present for all components and they have measured coverage	

Table 10-21 Error Condition Sharing Testability Scoring Criteria

Score	Basis	Notes
1	No information provided or no communication support in the application for the user to share error information with the help desk	
2	By phone	
3	By emailing elements captured on the workstation	
4	By automatic submission on error condition or when submission is requested by the user	