# Enterprise Correspondence Management - Letters -- Integration Architecture

* **1** [**Introduction**](#EnterpriseCorrespondenceManagementLette)
  + **1.1** [**What Problem Needs to be Solved?**](#EnterpriseCorrespondenceManagementLette)
  + **1.2** [**Business Needs**](#EnterpriseCorrespondenceManagementLette)
    - **1.2.1** [**AWS Costing Considerations**](#EnterpriseCorrespondenceManagementLette)
    - **1.2.2** [**Chargeback Considerations**](#EnterpriseCorrespondenceManagementLette)
  + **1.3** [**Technical Needs**](#EnterpriseCorrespondenceManagementLette)
* **2** [**Solution Space and Discussion**](#EnterpriseCorrespondenceManagementLette)
  + **2.1** [**Use Case View**](#EnterpriseCorrespondenceManagementLette)
    - **2.1.1** [**Capability Viewpoint Vision (CV-1) (VASI Description)**](#EnterpriseCorrespondenceManagementLette)
    - **2.1.2** [**Capability Viewpoint Taxonomy (CV-2)**](#EnterpriseCorrespondenceManagementLette)
    - **2.1.3** [**Use Case**](#EnterpriseCorrespondenceManagementLette)
      * **2.1.3.1** [**Main Actors**](#EnterpriseCorrespondenceManagementLette)
      * **2.1.3.2** [**System/Tenant/Application Description**](#EnterpriseCorrespondenceManagementLette)
      * **2.1.3.3** [**Main User Functions (Use Cases)**](#EnterpriseCorrespondenceManagementLette)
    - **2.1.4** [**Primary Systems High Level Operational Viewpoint (OV-1)**](#EnterpriseCorrespondenceManagementLette)
  + **2.2** [**Logical View**](#EnterpriseCorrespondenceManagementLette)
    - **2.2.1** [**Systems Interface Description (SV-1)**](#EnterpriseCorrespondenceManagementLette)
    - **2.2.2** [**FTI Environment Example SV-1**](#EnterpriseCorrespondenceManagementLette)
    - **2.2.3** [**Services Context (SvcV-1) (VASI APIs)**](#EnterpriseCorrespondenceManagementLette)
    - **2.2.4** [**Services Resource Flow (SvcV-2)**](#EnterpriseCorrespondenceManagementLette)
    - **2.2.5** [**State Transitions (OV-6b)**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.1** [**Changeset Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.2** [**Connection Configuration Version Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.3** [**Workflow Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.4** [**Key Parameter Type Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.5** [**Resource Version Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.6** [**Template Version Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.7** [**Test Case Lifecycle**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.5.8** [**Letter Instance Lifecycle**](#EnterpriseCorrespondenceManagementLette)
    - **2.2.6** [**Data Models**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.6.1** [**Conceptual Data Models (DIV-1)**](#EnterpriseCorrespondenceManagementLette)
        + **2.2.6.1.1** [**Template and Changeset Monogamy**](#EnterpriseCorrespondenceManagementLette)
        + **2.2.6.1.2** [**Application, Workflow and Template Alignment**](#EnterpriseCorrespondenceManagementLette)
        + **2.2.6.1.3** [**Key Concepts**](#EnterpriseCorrespondenceManagementLette)
        + **2.2.6.1.4** [**Entity Relationship Diagram**](#EnterpriseCorrespondenceManagementLette)

**2.2.6.1.4.1** [**Resource Manager**](#EnterpriseCorrespondenceManagementLette)

**2.2.6.1.4.2** [**Template Manager**](#EnterpriseCorrespondenceManagementLette)

**2.2.6.1.4.3** [**Letter Manager**](#EnterpriseCorrespondenceManagementLette)

* + - * **2.2.6.2** [**Logical Data Viewpoint Model (DIV-2)**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.6.3** [**Legacy Component Transitioning - Conceptual Rosetta Stone**](#EnterpriseCorrespondenceManagementLette)
        + **2.2.6.3.1** [**High Level Rosetta Stone**](#EnterpriseCorrespondenceManagementLette)
        + **2.2.6.3.2** [**Lower Level Rosetta Stone**](#EnterpriseCorrespondenceManagementLette)
      * **2.2.6.4** [**Physical Data Viewpoint Model (DIV-3)**](#EnterpriseCorrespondenceManagementLette)
  + **2.3** [**Process View (Business Process Model)**](#EnterpriseCorrespondenceManagementLette)
    - **2.3.1** [**Main Business Process Model (OV-6d)**](#EnterpriseCorrespondenceManagementLette)
    - **2.3.2** [**Intake Processes**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.1** [**Line of Business Intake Process**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.2** [**Application Intake Process**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.3** [**Connection Configuration Intake Process**](#EnterpriseCorrespondenceManagementLette)
        + **2.3.2.3.1** [**Process Workflow for Connection Configuration Intake**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.4** [**Make Connection Available**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.5** [**Connection Decommissioning Process**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.6** [**Create Resource Version (\*\*awaiting final approval)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.7** [**Resource Version Creation Process (For already existing Resource Version)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.8** [**Create Workflow**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.9** [**Update Workflow**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.10** [**Retire Workflow**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.11** [**Overall Changeset Process (\*\*awaiting final approval)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.12** [**Create Template Version Subprocess (\*\*awaiting final approval)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.13** [**Template Version Retirement Subprocess (\*\*awaiting final approval)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.14** [**Template Version Testing Subprocess (\*\*awaiting final approval)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.15** [**Changeset Approval Process (\*\*awaiting final approval)**](#EnterpriseCorrespondenceManagementLette)
      * **2.3.2.16** [**Letter Finalization Process**](#EnterpriseCorrespondenceManagementLette)
        + **2.3.2.16.1** [**External Letter Finalization**](#EnterpriseCorrespondenceManagementLette)
        + **2.3.2.16.2**
    - **2.3.3** [**Process Sequence - Systems Event-Trace Description (SV-10c)**](#EnterpriseCorrespondenceManagementLette)
  + **2.4** [**Implementation View**](#EnterpriseCorrespondenceManagementLette)
  + **2.5** [**Deployment View**](#EnterpriseCorrespondenceManagementLette)
    - **2.5.1** [**Environment Mapping**](#EnterpriseCorrespondenceManagementLette)
    - **2.5.2** [**System Configuration**](#EnterpriseCorrespondenceManagementLette)
    - **2.5.3** [**System Monitoring and Metrics**](#EnterpriseCorrespondenceManagementLette)
    - **2.5.4** [**System Logging and Auditing**](#EnterpriseCorrespondenceManagementLette)
      * **2.5.4.1** [**Long-Term Provenance**](#EnterpriseCorrespondenceManagementLette)
    - **2.5.5** [**Security Strategy**](#EnterpriseCorrespondenceManagementLette)
  + **2.6** [**Analysis of Alternatives**](#EnterpriseCorrespondenceManagementLette)
  + **2.7** [**Recommendation**](#EnterpriseCorrespondenceManagementLette)
  + **2.8** [**Phases and Roadmap**](#EnterpriseCorrespondenceManagementLette)
    - **2.8.1** [**Phases**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.1** [**Phase 0: Foundations**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.2** [**Phase 1: Simple Inputs**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.3** [**Phase 2: Images and Attachments**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.4** [**Phase 3: Data Resources**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.5** [**Phase 4: Fragments, Conditional Attachments and Restrictions**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.6** [**Phase 5: Attachment Field Population**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.7** [**Phase 6: Bar Code Support**](#EnterpriseCorrespondenceManagementLette)
      * **2.8.1.8** [**Phase Map**](#EnterpriseCorrespondenceManagementLette)

**Solution Metadata**

|  |  |
| --- | --- |
| **Name** | **Value** |
| **Solution Status** | DRAFT |
| **Solution Type** | NEW TENANT |
| **Tenant** | 3000 DocGen |
| **Recommended Namespace** | ecml |
| **Recommended BIP Tenant App(s)** | bip-docgen-ecml-resource-manager  bip-docgen-ecml-resource-manager-mfe-ui  bip-docgen-ecml-letter-manager  bip-docgen-ecml-letter-manager-mfe-ui  bip-docgen-ecml-template-manager  bip-docgen-ecml-template-manager-mfe-ui  bip-docgen-ecml-svc  bip-docgen-ecml-mfe-root-ui |
| **Sponsoring organization** | OIT |
| **Number of users** | TBD |
| **Estimated Monthly Cost** | TBD |
| **Privacy** | Differs from system to system.  Resource Manager - None.  Template Manager - None.  Letter Manager - Both possible.  ECM-L-API - Both possible. |
| **Will need 508 Compliance** | Yes, all of the MFE UI's will require 508 compliance. |
| **Deployment date** | TBD |

# Introduction

This document is the design for Enterprise Correspondence Management-Letters (ECM-L), which enables business users to oversee the content and generation of letters. Currently, multiple Lines of Business (LoB) across the Veterans Affairs infrastructure use separate methods and applications to generate correspondence for their unique business needs. Each LoB relies on their own methods and applications to generate correspondence, resulting in operational inefficiencies and overreliance on IT support.

To address this challenge, Enterprise Correspondence Management-Letters (ECM-L) will serve as a centralized solution that enables LoBs to self-manage their correspondence needs within a unified infrastructure. Once provisioned into ECM-L, LoBs will gain autonomy to implement changes such as modifications to letter templates and resources independently of  IT resources, while still operating within a standardized environment. This approach will provide a central and controlled environment to produce, test, release, and store correspondence that is reliable across all LoBs.

## What Problem Needs to be Solved?

Individual Lines of Business (LoBs) within the Veterans Affairs Organization utilize disparate methods and applications to generate correspondence specific to their LoB. The absence of a centralized enterprise correspondence management system results in each LoB operating independently for letter templates and changes. Any substantial changes to templates or mandates rely on IT resources for those changes, often requiring customized development for each LoB. This dependency leads to resource constraints, inefficiencies, and a lack of historical validation, ultimately causing delays or inaccuracies in the correspondence delivered to our Veterans.

## Business Needs

|  |  |
| --- | --- |
| Is this a defect or an enhancement? | Enhancement |
| Who reported the need? | Office of Information & Technology (OIT) |
| When was the need noticed? | Q2 2022 |
| When did/will it become a problem? | Q2 2024 |
| Is there an existing Requirement? Strategic Objective? | TBD |
| What existing resources, tickets, emails, etc. might serve to better understand the problem? |  |
| What business processes are involved? | Document Generation, Document Storage and Retrieval |
| Who is the business owner for the impacted systems or processes? | * Jeffry Boutet * Melissa Morris * Brian Steege * Peter Graziatis |
| Who are the business decision makers for solving the problem? | Pam Devine |
| What is the historical context? How did we get where we are? | Historically, VA has used multiple applications to generate, manage, and edit letters. Correspondence Service has provided document generation functionality to the VBMS system, providing document generation for use in VBMS Core, Ratings, and Awards. With this architecture, any changes needed by the business specific letter requires large amounts of commitment and OIT resources. |
| What happens to the business stakeholders if nothing is done? | VA will have disjointed efforts and applications for correspondence and letters |
| Of the various aspects of this problem, which of them does the business consider as critical flaws? Which are nice-to-have changes? | * Need for improvements in Letters finalization * Need for centralized enterprise system that allows multiple LoBs to draft, generate, and finalize letters |

### AWS Costing Considerations

A total of fifteen LOBa will require the use of ECM-L letter generation and management functionality. It has been determined that C&P produces the largest number of letters, sitting at an average of 5000 letters per day. Utilizing this information as parameters, we developed a cost estimate based on an average file size of 400KB per letter. The table below illustrates the Standard S3 storage costs for C&P and LoBs smaller than C&P. According to AWS calculator, the cost would be under $20 per year.

The table below uses Compensation & Pension as the standard of 100% since it is a large LOB.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **%** | **Letters Per Day (LPD)** | **LPD Size (GB)** | **Cost Per Day ($)** | **Letters Per Month (LPM)** | **LPM Size (GB)** | **Cost Per Month ($)** | **Letters Per Year (LPY)** | **LPY Size (GB)** | **Cost Per Year ($)** |
| C & P | 5000 | 2 | 0.046 | 150000 | 60 | 1.38 | 1800000 | 720 | 16.56 |
| 50 | 2500 | 1 | 0.023 | 75000 | 30 | 0.69 | 900000 | 360 | 8.28 |
| 25 | 1250 | 0.5 | 0.0115 | 37500 | 15 | 0.345 | 450000 | 180 | 4.14 |
| 10 | 500 | 0.2 | 0.0046 | 15000 | 6 | 0.138 | 180000 | 72 | 1.656 |

### Chargeback Considerations

Each Line of Business (LoB), independent of one another, requires their own resources, templates, and letter instances in ECM-L; therefore, they can be easily separated financially by an LoB's application.

## Technical Needs

Historically, VA has used multiple applications to generate, manage, and edit letters. This causes numerous workflow inconsistencies, various vendors, and large amounts of OIT commitments and resources, none which seem to have a perfected process in place. The purpose of Enterprise Correspondence Management- Letters (ECM-L)  is to create a centralized, enterprise system, built around shared concepts, processes, and workflows to be leveraged by multiple Lines of Business (LoBs) in the VBA.  The system will be LoB oriented, each being responsible for their integration and usage, with BID only supporting, facilitating, and trouble-shooting as needed. Onboarding new LoB will not require development work, but use configurations and system activities.

 In order to prevent unintended interactions or dependencies, as well as support charge-back capabilities, the content, processes, and data of each LoB will be completely separate from those of another LoB. In addition, the ECM-L will be LoB agnostic, providing reasonable common denominators in concepts or processes among the LoBs, but ECM-L will not contain any proprietary business logic or concepts. Each LoB will be able to integrate and incorporate their own business logic or processes through predefined hooks, APIs, or other integration points. This will allow API access to Letter instances in data format as well as give LoBs the ability to configure data lookup "callback" to its own systems.

The implications of this enterprise system allow for reusing existing concepts in its approach, the opportunity to consolidate and simplify letter generation and management across VBA, and the ability to plan for complexity.

|  |  |
| --- | --- |
| Describe the problem in technical terms | We need an Enterprise level Line of Business agnostic tool to create and manage correspondence that can integrate with external API's without the need for a code change. |
| What are the architectural components in the problem's context? | New tenants: Resource Manager, Template Manager, Letter Manager, ECM-L Service. |
| What are the integration points that might be impacted by this problem? | Each of the new tenants will integrate with Line of Business specific API's on an as needed basis. |
| What technologies are involved? | Springboot, React, Kafka, iText, Tiny MCE. |
| Is this related to already-stated technical debt? | No. |
| If the problem is with a technology, what were the reasons for using this technology in the first place? | N/A |
| Has an attempt been made before to fix the problem? | No.  The existing components in the DocGen namespace were not designed to accomplish this goal / solve this problem and are too entwined with Line of Business specific logic to be repurposed to solve it. |
| Did it fail or did it just not finish? | N/A. |
| Who are the technical PoCs for that attempt? | N/A. |
| What components are impacted by this problem? | -All VA letters/correspondence  -Core, Ratings, Awards |
| Are there current mitigations that help alleviate the impact of the problem? | No. |
| What are the technical dependencies for this? What does this depend on? What depends on it? | The new tenants proposed will stand on their own and external Line of Business specific API's can be configured in as dependencies as needed. |
| Based on the SV-1 and SV-9 generated what is estimated monthly cost for this new service? Upload estimate of resources from the [AWS Pricing Calculator](https://calculator.aws/#/) | TBD |

# Solution Space and Discussion

The solution space describes the architectural context related to the problem/solution in a 4+1 style document. A 4+1 Design is a human-readable, standardized way of viewing an architecture from multiple perspectives. Each perspective is complementary, offering a different way to look at a particular architecture. Taken as a whole, a 4+1 Design is a comprehensive way to communicate an architecture.

## Use Case View

The Use-Case (or Business) View describes a high-level understanding of what the system must enable the users to do. This view closely ties to requirements documents and, in many ways, serves to restate needs formally listed there.

### Capability Viewpoint Vision (CV-1) (VASI Description)

The Enterprise Correspondence Management- Letter (ECM-L) service will reside on the Benefits Integration Platform as a Service (#2295) and is under the Veteran Benefits Management System (VBMS #1728) Application. ECM-L will  provide an enterprise level document system that is usable by multiple Lines of Business (LoBs) in VBA. The ECM-L will be built around concepts, processes, and workflows typically shared by VBA LoBs, while keeping the data agnostic. The ECM-L will allow each LoB to plug in, draft, and finalize their material while keeping the content, data, and processes of each, completely separate. This supports common processes while maintaining a LoB agnostic system.  In addition, the extensibility of ECM-L will allow LoBs to integrate and incorporate into their business logic through predefined hooks, APIs, or other integration points. The service provides a singular method for creating letters to be used by VBA systems.

### ****Capability Viewpoint Taxonomy (CV-2)****

The following diagram provides an architectural description of the overall, high-level vision and for the capability of this component/tenant.

|  |  |
| --- | --- |
| |  | | --- | |  | |

### Use Case

The following diagram visually represents the interactions between users, or actors, and the system. The diagram illustrates the functional requirements of the system.

|  |  |
| --- | --- |
| |  | | --- | |  | |

#### ****Main**** Actors

The following table describes the actors listed in the diagram above. An actor is user in the scope of this architecture and can be a human user or a "system actor" as appropriate.

| **Name** | **System** | **Description** |
| --- | --- | --- |
| Line of Business Administrator | Resource Manager, Template Manager | A User with the requisite roles and permissions to manage Resources for Templates.  A template is a configuration that specifies how a letter should be generated, including which data source to use, which fields to display, how to format the data, and the overall layout. User is also able to organize Templates into groups, i.e., assign a Template to a designated group(s). |
| Changeset Editor | Template Manager | A User responsible for managing Changesets, including creating, updating/modifying, testing, and cancelling Changesets, to facilitate the activation and/or retirement of a Template Version.  The User must be able to manage Templates in a Changeset, including the ability to create (specify a Template's content and configuration), read, update, cancel, and retire.  This user can also associate templates with a label. |
| Changeset Publisher | Template Manager | A User responsible for Approving, Scheduling, and Publishing Changesets. |
| Line of Business System Actor | ECM-L API | External system that interacts with ECM-L to conduct automated ECM-L operations that do not involve a human actor, including operations to Finalize a Letter, Generate Automated Letters, Get Letter Data, and Render a Letter Instance as a PDF. |
| Letter Finalizer | Letter Manager | A user with the ability to Finalize letters in the Letter Manager. |
| Letter Approver | Letter Manager | A user with the ability to Approve Letters in the Letter Manager. |
| Letter Editor | Letter Manager | A user with the ability to create, read, update, render, and delete letters in the Letter Manager. |

#### ****System/Tenant/Application Description****

The following table describes each system/tenant/application (rectangles) listed in the diagram above.

| **System Name** | **Description** |
| --- | --- |
| Resource Manager | An API that allows users to create and manage Resources and API Configurations.  The ECM-L subsystems such as the Template Manager and the ECM-L API are able to use these configured Resources and API Configurations to communicate with and gather contextual data from external Line of Business specific API's. |
| Template Manager | The tool used to manage templates with the ECM-L system. |
| ECM-L API | The ECM-L API is the outward face of the ECM-L system and allows for external system actors to interact with the ECM-L system in lieu of a human actor directly interacting with one of the individual components. |
| Letter Manager | The tool used to manage letter instances within the ECM-L system. |

#### Main User Functions (Use Cases)

The following table describes each of the use cases (ovals) in the diagram above.

| **Actor(s)** | **System** | **Ability** | **Details** |
| --- | --- | --- | --- |
| Line of Business Administrator | Resource Manager | Manage Resource | The Line of Business Administrator is able to create, modify, and delete Resources (Schema/File), for an Application. The Line of Business Administrator will need to be able to view the list of available Connection Configurations to create Resources. The Line of Business Administrator will need to align or match a given Resource with the Application. |
| Manage Workflows | The Line of Business Administrator is able to create, modify, and retire Workflows and to align Templates to Workflows. |
| Changeset Editor | Template Manager | Manage Templates | The Changeset Editor accesses the Template Manager to create, modify, and retire Templates.  This includes the ability to add or remove Resources for a Template. The Changeset Editor also has the ability to add Inputs (Data Entry Components) to a Template and format the values, as well as add conditional restrictions to a Template. |
| View List of Templates | A Changeset Editor is able to view a List of Templates. |
| Manage Labels | The Changeset Editor accesses the Template Manager system and is able to create, modify, and delete Template Labels, and align Templates to those Labels. |
| Manage Changesets | A Changeset Editor is able to create, test, and cancel Changesets. |
| Changeset Publisher | Template Manager | Approve Changesets | Once a Changeset is tested and ready for review a user with the Changeset Publisher role can approve the Changeset. |
| Schedule Changesets | The Changeset Publisher is able to  schedule publication of the Changeset immediately or in the future. |
| Line of Business System Actor | ECM-L API | Get List of Letters | A Line of Business System Actor is able to get Letter Instances from the ECM-L API based on Application and Key Parameters. |
| Get Letter Data | A Line of Business System Actor is able to collect letter data from the ECM-L API. |
| Fast Track Letter Instance(s) (auto Finalize) | A Line of Business System Actor is able to generate an automated letter within the ECM-L API.  If autoFinalize is true, the system actor can create and automatically finalize a Letter Instance(s). |
| Update Letter Instance(s) | A Line of Business System Actor has the ability to update a Letter Instance(s) within the ECM-L API. |
| Finalize Letter Instance(s) | A Line of Business System Actor has the ability to finalize a Letter Instance(s) within the ECM-L API. |
| Delete Letter Instance(s) | A Line of Business System Actor has the ability to delete a Letter Instance(s) within the ECM-L API. |
| Render Letter as PDF | A Line of Business System Actor is able to render a letter as a PDF from the ECM-L API. |
| Letter Finalizer | Letter Manager | Finalize Letter | Letter Finalizer has the ability to Finalize a Letter Instance.  Finalizing the Letter freezes its content and makes it immutable / unchanging. |
| Letter Approver | Letter Manager | Approve Letter | Some Applications will be configured to require Letters to be approved before they are finalized, and the Letter Approver role will be able to approve Letters to prepare them for Finalization. |
| Letter Editor | Letter Manager | Manage Letters | The Letter Editor has the ability to create, read, update, and delete a Letter Instance within the Letter Manager. |
| Render Letter as PDF | The Letter Editor is able to get a PDF rendering of a Letter Instance within the Letter Manager. |
| View List of Letters | The ability to view a List of available Letters within the Letter Manager. |

### Primary Systems High Level Operational Viewpoint (OV-1)

 The diagram below is a high level overview of the main architectural components of ECM-L and the functions they facilitate in the value stream.

The first section of the value stream, the intake process, starts as a Jira ticket. This section consists of mainly manual processes provisioning space and storage, establishing user access requirements, and aligning applications. At this point in the process Line of Business (LoB) data needs will be identified and API connections will be established to accommodate those needs.

The ECM-L LoB Business User Processes box consists of two main values; creating Resources and creating Templates. Resources are assets that will point to LoB data using the established API connections to populate Letter Instances as needed. The ability to create Resources is realized by the Resource Manager. Creating Templates, or Letter templates, is realized through the Template Manager; as well as the ability to test Resources.

The ECM-L LoB End User Processes box signifies the business processes involved in instantiating Letter Instances in the Letter Manager using Templates and Resources.

LoB System Processes encapsulates the ability to access, store, and send Letters as the LoB sees fit. Access to the Letters is capable through the ECM-L API.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |
| --- | --- | --- |
| **System (VASI)** | **System Item** | **Interaction** |
| ECM-L (3000) | Resource Manager | Uses the LoB connection to create and manage resources. |
| ECM-L (3000) | Template Manager | Creates and manages the LoB templates. |
| ECM-L (3000) | Letter Manager | Creates and manages the LoB letter instances. |
| ECM-L (3000) | ECM-L API | Accesses the LoB data and letters for the LoB. |
| VEFS (3023) | Claim Evidence | Stores the finalized PDF related to a claim. |
| PacMan (2993) | Package Manager | Creates a package for mail distribution. |
| LoB System | LoB System | Through the LoB API, uses its connection to establish resources, templates and populate letters with the appropriate data and access the finalized letters. |

## Logical View

The Logical View describes the conceptual view of the functionality and of the systems that will participate. This view defines what nouns are involved and the relationship between them. For example, it may decompose the design into sub-systems and packages, and for significant packages, the decomposition into classes and class utilities. This section may include the Business Object Model, states and transitions for key objects, and other key details about classes and attributes.

### Systems Interface Description (SV-1)

The following diagram provides a detailed view of system interfaces, depicting how systems and components interact with each other within the architecture.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |
| --- |
| **Legend** |
| Gray Rectangle = External to core system(s) |
| White = Internal to core system(s) |

**Component References Table**

| **Name** | **Description** | **Port** | **Protocol** |
| --- | --- | --- | --- |
| ECM-L Database | Database containing all ECM-L related data. | 1521 | TCP |
| Finalized Letter Repository | S3 Bucket containing Finalized Letter PDFs, that could potentially contain PII/PHI/etc. | 1521 | TCP |
| Resource Manager API | Internal API for the Resource Manager | 443 | HTTPS/REST |
| Template Manager API | Internal API for the Template Manager | 443 | HTTPS/REST |
| Letter Manager API | Internal API for the Letter Manager | 443 | HTTPS/REST |
| ECM-L API | The API for the umbrella service for the ECM-L system, integrated with all of the internal components. | 443 | HTTPS/REST |

### FTI Environment Example SV-1

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |
| --- |
| **Legend** |
| Gray Rectangle = External to core system(s) |
| White = Internal to core system(s) |

**Component References Table**

| **Name** | **Description** | **Port** | **Protocol** |
| --- | --- | --- | --- |
| ECM-L Content Database | Database containing "dry" non-contextual data, uncontaminated with PII/PHI/etc. | 1521 | TCP |
| ECM-L Letters Database | Database containing contextual data contaminated with PII/PHI/etc. |  |  |
| Finalized Letter Repository | S3 Bucket containing Finalized Letter PDFs, that could potentially contain PII/PHI/etc. | 1521 | TCP |
| Resource Manager API | Internal API for the Resource Manager | 443 | HTTPS/REST |
| Template Manager API | Internal API for the Template Manager | 443 | HTTPS/REST |
| Letter Manager API | Internal API for the Letter Manager | 443 | HTTPS/REST |
| ECM-L API | The API for the umbrella service for the ECM-L system, integrated with all of the internal components. | 443 | HTTPS/REST |

### Services Context (SvcV-1) (VASI APIs)

The following diagram describes service compositions and interactions, showing a given Application Components service and immediate service dependencies. The diagram highlights external system services that connect to the focal architecture.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |
| --- |
| **Legend** |
| Gray Rectangle = External to core system(s)  White = Internal to core system(s) |

|  |  |
| --- | --- |
| **Service / API** | **Description** |
| Resource Manager | MFE providing LoB with the ability to create, test, replace, and retire Resources. |
| Template Manager | MFE providing LoB with the ability to create, test and publish Changesets as well as create, test and retire Templates. |
| Letter Manager | MFE providing ability to create Letter Instances, the Letter Manager will grab the Template and populate using Resources. |
| ECM-L API | ECM-L API provides the LoB system with access to the Letter instances for sending and storing. |
| LoB API | LoB API is a placeholder for the connection that will be set up for resources to point to and access data from when populating a Template. |
| LoB UI | LoB UI will integrate connection to ECM-L MFE's for seamless workflow for the end user. |

### Services Resource Flow (SvcV-2)

The following diagram specifies the Resource Flows between Services and may also list the protocol stacks used in connections, providing a precise specification of a connection between Services.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |
| --- |
| **Legend** |
| Gray Rectangle = External to core system(s)  White = Internal to core system(s) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **HTTP Method** | **HTTP URL** | **Operation ID** | **Performing System** | **Description** |
| GET | LM/letterInstance/{uuid} | getSingleLetter | Letter Manager | ECM-L API request single Letter Instance |
| GET | LM/letterInstance | getList | Letter Manager | ECM-L API request of Letter Instances based on the Application and Filters (e.g., workflow, key parameters). |
| PUT | LM/letterInstance/finalization | finalizeLetter | Letter Manager | ECM-L API ascribes FINAL to an existing Letter Instance that is no longer editable |
| DELETE | Delete LM/letterInstance | deleteLetter | Letter Manager | ECM-L API deletes a Letter Instance |
| POST | LM/letterInstance?(autoFinalize=True) | createLetter | Letter Manager | ECM-L API creates a Letter Instance, if autoFinalize is true when the Letter is created it will automatically be Finalized. The autoFinalize enables the ability for LoB system actors to automaticallu create and then finalize a Letter |
| PUT | LM/letterInstance/{uuid} | updateLetter | Letter Manager | ECM-L API updates a Letter Instance |
| GET | TM/template?status=[DRAFT,COMPLETED,APPROVED,ACTIVE,RETIRED,Canceled] | listTemplates | Template Manager | ECM-L API requests list of Templates |
| GET | TM/template/{uuid} | getTemplate | Template Manager | ECM-L API requests Template |

### State Transitions (OV-6b)

The following diagrams map out the various states a specific object within the ECM-L architecture can occupy, and the triggers that move the object from one state to another.

#### Changeset Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DRAFT** | The Changeset has been created, named, and is undergoing planning. During this state, Templates are being created, added/removed to/from the Changeset, and submitted for review. During this state the Changeset remains editable. The DRAFT state is the only time when a Changeset can be modified.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ChangesetDraftCompletedEvent | READY TO TEST | Once all Templates have been submitted for review and testing- all Templates in the Changeset must be in either COMPLETED or ACTIVE state for the Changeset Draft to be complete. | | ChangesetCancellationProcessCompleteEvent | Canceled | The Changeset has been Canceled, but the included Templates remain in their existing state. | |
| **READY TO TEST** | The Changeset is fully assembled and all Templates are in COMPLETED state or ACTIVE state (for Templates slated for Retirement Action). This state involves creating and executing test cases to validate individual Templates and Template combinations/associations.  Activities that happen during this state:   * Create test cases to validate changes * Execute test activity for each test case * Manage Test Cases   **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ChangesetTestFailedEvent | DRAFT | All tests have been created, but one or more test cases have failed. Changeset sent back to DRAFT to make edits as needed. | | ChangesetTestsPassedEvent | TESTED | All tests have been created and have passed. Marking a test case as passed means the user has created a test case, executed it, viewed the results, and finally has verified that the Template changes are as expected within a test PDF. The user can create multiple cases with different data as needed, however all must be marked as passed before this transition can happen. | | ChangesetRecalledEvent | DRAFT | This Changeset has been recalled to allow for editing as needed. | |
| **READY FOR APPROVAL** | Every Template Version in the Changeset has at least one corresponding Test Case that has been marked as passed. For Changesets that do not contain a Template Version, such as Changesets that only contain a Retirement Action, no testing is required.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ChangesetTestAddedEvent | READY TO TEST | An additional test has been added to the Changeset, transitioning it back to READY TO TEST. | | ChangesetApprovalProcessCompletedEvent | APPROVED | The Changeset has completed the applicable approval process and been APPROVED.  ChangesetApprovalProcessCompletedEvent implies all Changeset Templates in the COMPLETED state (i.e., not targeted for retirement) have transitioned to the APPROVED state. | | ChangesetRecalledEvent | DRAFT | This Changeset has been recalled to allow for editing as needed. | | ChangesetCancellationProcessCompleteEvent | CANCELED | Changeset is no longer needed and has been CANCELED. | |
| **APPROVED** | Changeset validated and ready to be published (immediately or at a scheduled date/time).  Every Template in a Changeset is in an APPROVED state when the Changeset Approval Process is completed and the Changeset is in the APPROVED state.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ChangesetRecalledEvent | DRAFT | This Changeset needs to go back to the drawing board and incorporate more or different Templates. | | ChangesetCancellationProcessCompleteEvent | CANCELED | Changeset is no longer needed and has been CANCELED. | | ChangesetPublishingPlannedEvent | SCHEDULED | Changeset is SCHEDULED to be published immediately or in the future. | |
| **SCHEDULED** | Changeset is scheduled to be published in the future with date/time specified.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ChangesetRecalledEvent | DRAFT | This Changeset needs to go back to the drawing board and incorporate more or different Templates. | | ChangesetRescheduledEvent | SCHEDULED | The time set for publication of the Changeset has been rescheduled. | | ChangesetScheduledTimeArrivedEvent | PUBLISHED | The scheduled time set for publication has arrived and the Changeset is PUBLISHED and ready for use. This transition implies the transition of all new and included Templates to the ACTIVE state and any ACTIVE Templates targeted by a Retirement Action will transition to RETIRED. | |
| **PUBLISHED** | Changeset now live and effective. All related Templates are in either ACTIVE or RETIRED state in accordance with the included Templates and/or Retirement Actions.  ACTIVE Templates are available to create Letter Instances. |
| **CANCELED** | Changeset is no longer needed and has been CANCELED. |

#### Connection Configuration Version Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **INTAKE** | Intake process initiated by LoB, Change Request ticket created in Jira for configuring connection.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ConnectionConfigurationIntakeRequestRejectedEvent | REJECTED | Connection unable to be configured.  Note:   * Connection Configuration may be REJECTED due to failure to complete the CR process. | | ConnectionConfigurationIntakeProcessCompleteEvent | PROVISIONED | The systems or operations teams have completed what is necessary to make the Connection available and visible inside of the Resource Manager. | |
| **PROVISIONED** | Once in the PROVISIONED state, the Connection is visible to the LoB Administrator in the Resource Manager and the LoB Administrator can make the Connection AVAILABLE.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ConnectionConfigurationPublishedEvent | AVAILABLE | The LoB Administrator selects the PROVISIONED Connection from within the Resource Manager and makes it AVAILABLE. | |
| **AVAILABLE** | When the Connection is AVAILABLE, it can be used to access data/files by enabling the creation of a Resource in the Resource Manager.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ConnectionConfigurationRemovedEvent | DECOMMISSIONED | Connection Configuration removed and cannot provide access to any data/files. | |
| **DECOMMISSIONED** | The Configuration Connection is no longer AVAILABLE to be used to create a new Resource(s) and all existing Resources derived from the Connection Configuration will need to be evaluated. |
| **REJECTED** | The intake process was not completed.  Examples: if an LoB failed to provide necessary security information to establish a connection, LoB decided not to proceed. |

#### Workflow Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AVAILABLE** | A Workflow is in the AVAILABLE state.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | WorkflowRetiredEvent | RETIRED | Workflow is no longer needed and RETIRED. | |
| **RETIRED** | Workflow cannot be used for anything new.  No new Letter Instances can be created in this Workflow. |

#### Key Parameter Type Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AVAILABLE** | A Key Parameter Type is in the AVAILABLE state can be used in a Path.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | KeyParameterTypeRetiredEvent | RETIRED | Key Parameter Type is no longer needed and RETIRED. | |
| **RETIRED** | Key Parameter Type cannot be used in anything new. |

#### Resource Version Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DRAFT** | LoB Administrator started to create a Resource using an AVAILABLE Connection Configuration.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ResourceDeletedEvent | DELETED | Resource no longer needed and DELETED.  Note:   * If the Connection Configuration from which the DRAFT Resource is derived is DECOMISSIONED, a Resource in DRAFT state will be DELETED. | | ResourceDraftCompletedEvent | AVAILABLE | Resource is named, a path to a Connection Configuration is identified (path will have list of Key Parameter types, URL, and a Connection Configuration ID), and all data fields are identified. | |
| **AVAILABLE** | A Resource in the AVAILABLE state can be used to create a Template.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | ResourceRetiredEvent | RETIRED | Resource is no longer needed and RETIRED. | |
| **RETIRED** | Resource cannot be used to create a Template. |
| **DELETED** | Resource no longer needed and has been DELETED. |

#### Template Version Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DRAFT** | Template Version has been created, named, and is editable.   Entering this state, an incremented Version number is assigned.  Version numbers must be guaranteed unique from all other Version numbers for this Template regardless of state.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | TemplateDraftCompletedEvent | COMPLETED | No more edits need to made to Template Version and it is marked as COMPLETED. | | TemplateCanceledEvent | CANCELED | Template Version is no longer needed and has been CANCELED. | |
| **COMPLETED** | Template Version marked as COMPLETED. \* Test cases for a Changeset can only be created for Template Version in the COMPLETED state.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | TemplateChangesNeededEvent | DRAFT | Changes need to be made to the Template Version. | | TemplateApprovalProcessCompletedEvent | APPROVED | The Template Version has gone through the applicable approval process and been APPROVED.  Template Version included in a Changeset that is in APPROVED state. | | TemplateCanceledEvent | CANCELED | Template Version is no longer needed and has been CANCELED. | |
| **APPROVED** | Template Version has passed testing and been approved, no further action is necessary before the corresponding Changeset is PUBLISHED.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | TemplateActivatedEvent | ACTIVE | Template Version included in PUBLISHED Changeset. | | TemplateCanceledEvent | CANCELED | Template Version is no longer needed and has been CANCELED. | | TemplateRecalledEvents | DRAFT | Template Version has been Recalled for editing. | |
| **ACTIVE** | Template Version is live and available for instantiating Letter Instances.  Template Versions in the ACTIVE state are permanent and unalterable. Set in stone. Immutable. Final.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | TemplateRetiredEvent | RETIRED | Represents a request, within a Changeset, to RETIRE a Template Version.  Notes:   * Each Template Version requires a Retirement Action, i.e., you cannot retire multiple Template Version with one Retirement Action. * Retirement process that occurs during this transition may involve a multi-step process where a user is able to view all versions of a Template and select which version to RETIRE and after selecting a Template version to RETIRE, the user is provided a list of all Templates dependent on the Version selected for Retirement. * Retirement is effective when Changeset which includes the Template selected for Retirement is PUBLISHED. | |
| **CANCELED** | Template Version is no longer needed and has been CANCELED. |
| **RETIRED** | Template Version is no longer needed and has been RETIRED.  Notes:   * A RETIRED Template cannot be used to create a Letter Instance. * FINALIZED Letter Instances will be unaffected by RETIRED Templates. |

#### Test Case Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DRAFT** | A user has created a Test Case for a Template Version and is in the process of defining it.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | TestCaseDeletedEvent | DELETED | Test Case no longer needed and DELETED. | | TestCaseCompletedEvent | READY | Test Case is READY for use. | |
| **READY** | A Test Case is ready to go and can be used to evaluate the Template Version it was created to test.  **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | TestCasePassedEvent | PASSED | The Test Case has been run and the user has marked it as PASSED. | |
| **PASSED** | After evaluation by a user, the Test Case has been deemed acceptable. |
| **DELETED** | Test Case no longer needed and has been DELETED. |

#### Letter Instance Lifecycle

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DRAFT** | Editable Letter Instance has been created based on an ACTIVE Template. Upon entering and exiting from this state, all Resources included in the Template Version will call their respective URL and will render a result appropriately.  Note:   * DRAFT Letter Instances can be previewed as a PDF and will be rendered with a "DRAFT" watermark.   **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | LetterInstanceDeletedEvent | DELETED | Letter Instance no longer needed and has been DELETED. | | LetterInstanceDraftCompleteEvent | READY FOR REVIEW | Letter Instance based on ACTIVE Template Version created and marked READY FOR REVIEW. | |
| **READY FOR REVIEW** | Letter Instance contains all elements required by the Template Version (e.g., inputs, attachments, inclusions, etc.) and is READY FOR REVIEW by a user with permission to FINALIZE.  Note:   * Depending on individual LOB processes, the Drafting and Reviewing functions may be performed by the same or different roles. * READY FOR REVIEW Letter Instances can be previewed as a PDF and will be rendered with a "READY FOR REVIEW" watermark.   **Transitions from this state:**   |  |  |  | | --- | --- | --- | | **EVENT** | **TO** | **DESCRIPTION** | | LetterInstanceDraftRejectedEvent | DRAFT | Changes need to be made to the Letter Instance before it can pass review and start the finalization process. | | LetterInstanceRecreatedforNewVersionEvent | DRAFT | A new Template Version upon which the Letter Instance is based became ACTIVE while a Letter Instance is in the READY FOR REVIEW state.  User has decided to update the Letter Instance with the new Template Version.  Note:   * If the Template Version upon which the Letter Instance was based was RETIRED AND a new Template Version became ACTIVE while in this state, the user could continue to use the RETIRED Template Version, update the Letter Instance with the new ACTIVE Template Version  or DELETE the Letter Instance. | | LetterInstanceFinalizedEvent | FINALIZED | A Letter Instance based on an ACTIVE Template Version and data inputs, has been reviewed and approved.  Note:   * The FINALIZED Letter Instance is stored as data and will live in perpetuity in a data format and can be rendered as PDF, JSON, plain text, html, etc. * A FINALIZED Letter Instance is unalterable - Template Version locked at time of Finalization cannot be changed. | | LetterInstanceDeletedEvent | DELETED | Letter Instance is no longer needed and has been DELETED  Note:   * If the Template Version upon which the Letter Instance was based was RETIRED AND a new Template Version became ACTIVE while in this state, the user could continue to use the RETIRED Template Version, update the Letter Instance with the new ACTIVE Template Version  or DELETE the Letter Instance. | |
| **FINALIZED** | Letter Instance with all elements of an ACTIVE Template Version and Data Input.  A FINALIZED Letter Instance is immutable and will persist in data format, available for rendering, in perpetuity. |
| **DELETED** | Letter Instance is no longer needed and has been DELETED.  Note:   * Only Non-Final Letter Instances can be DELETED. |
| **ARCHIVED** | After a given period of time, the Letter Instance will be ARCHIVED as per applicable archival procedure. |

### Data Models

#### Conceptual Data Models (DIV-1)

The following diagrams capture high level data concepts of ECM-L.

##### Template and Changeset Monogamy

Templates can only ever exist in a single Changeset in an "Unfinished" state at a given time.  While a Changeset is in these states it is still being worked on and processed, and the Templates are essentially "Changeset Locked" and are prevented from being used in other Changesets.

**"Unfinished" Changeset States:**

* Draft
* Ready to Test
* Ready for Approval
* Approved
* Scheduled

After a Changeset is moved to a "Finished" state the Templates that it has "Locked" become "Unlocked" and can now be used in other Changesets again.

**"Finished" Changeset States:**

* Canceled
* Published

|  |  |
| --- | --- |
| |  | | --- | |  | |

##### Application, Workflow and Template Alignment

This diagram illustrates the relationship between workflows, labels, and templates within an application under the line of business. It consists of three main components: Workflow, Label, and Template.

* **Applications:**Applications are what own a Template.  LOB Admins can share a Template owned by one Application with another Application.
* **Templates**: Templates are user created Line of Business tailored Letter templates for a type of correspondence.
* **Workflows**: Workflows represent discrete tasks and sub processes within which end users of an Application may interact with the ECM-L.  Template Labels can be aligned to a Workflow so that Template Managers can create a curated and tailored list of Templates that should be presented to end users when they are sent to the Letter Manager while working within that Workflow.

A user can organize Templates by assigning them to workflows.

|  |  |
| --- | --- |
| |  | | --- | |  | |

The diagram above shows how Templates relate to Workflows.

|  |  |  |
| --- | --- | --- |
| **Application** | **Workflow** | **Available Templates** |
| A | A1 | A, B, C |
| A2 | C, D, E, F |
| A3 | None |
| No Workflow | All |
| B | B1 | H |
| No Workflow | None |

\*Special permission is required for users to have access to all Templates in this scenario.  Ordinary usage would require accessing the Letter Manager within the context of a Workflow.

##### Key Concepts

This is a diagram illustrating the basic high level concepts and how they relate to one another.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| **Key Term** | **Description** |
| **Line of Business** | An organizational segment of operational and strategic activities. |
| **Application** | The name of a product or service within a Line of Business that integrates with and interacts with the ECM-L components. |
| **Workflow** | An LOB Application-defined context in which the Letter Manager MFE is contained. |
| **Connection Configuration** | Mechanism for ECM-L to access another system for data/files to be included in Letters.  Enables connectivity and security. |
| **Resource** | Represents a web-based asset (data or file) on an LoB’s system. |
| **Data Resource** | Structured data resulting from a successful GET operation on an API. |
| **File Resource** | Static resource, such as an image or PDF form, accessible via URL. |
| **Template** | A versioned representation of a document type, usually containing template and Resource references necessary to manage and generate a Letter Instance. Templates can include other Templates, for example, using a header or a footer shared by multiple Templates. |
| **Fragment Template** | A Template that represents a piece of content can be included inside of another Template.  Fragment Templates can not become a Letter Instance on their own. |
| **Full Template** | A Template that represents the full content of a piece of correspondence, possibly including Fragment Templates. |
| **Template Version** | A specific version of a Template. |
| **Template Label** | Organizes Templates into groups. |
| **Letter Instance** | An instantiation of a Template Version. A Letter Instance can be manifested in the form of data, PDF, or other renderings. |
| **Changeset** | A vehicle for managing change with Templates, including creating new Template versions and retiring old Template versions. |
| **Retirement Action** | A representation of a user's desire to retire a specific version of a template and make it no longer available for letters. |

##### Entity Relationship Diagram

This Entity Relationship Diagram illustrates how the high level concepts of the ECM-L systems interact with and relate to one another.

|  |  |
| --- | --- |
| |  | | --- | |  | |

###### Resource Manager

**Line of Business**

A collection of Applications / services serving a specific customer need or business function. Applications, Resources, Key Parameter Types, and Connection Configurations all belong to a specific Line of Business.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |

**Application**

The product or service that belongs to a Line of Business which needs to interact with the ECM-L System.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Line of Business ID | The UUID of the Line of Business to which it belongs. |

**Workflow**

The workflow or sub process in which letters are created within a given Application.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Application ID | The UUID of the Application to which it belongs. |
| Label ID(s) | The UUID's of the labels aligned to this workflow. |

**Connection Configuration**

All of the configuration information necessary for the ECM-L to successfully interact and integrate with a specific external API. These are loaded into the Resource Manager via a configuration file and then are made available for use by Resources via the Resource Manager.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| URL | The environment-specific URL for the API. |
| Secret Path | The path to the secret in Vault. |
| Issuer Path | The path to the issuer in Vault. |
| Line of Business ID | The UUID of the Line of Business to which it belongs. |

**Resource**

The information necessary to gather content and data from outside of the ECM-L system that can be used within the documents that it creates and renders.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | The user friendly name of a Resource. |
| Type | The type of the resource. |

**Resource Version**

A specific version of a Resource.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Paths | A list of paths that can be used to access the Resource, depending on the given Key Parameter Type. |
| Connection Configuration ID | The ID of the Connection Configuration it uses. |

**Path**

The information related to a specific operation / API path that will allow the ECM-L System to access the information in a Resource Version.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Key Parameter Type(s) | The IDs of the Key Parameters required for this path. |
| URL | The path to the endpoint in the API represented by the chosen Connection Configuration. |

**Key Parameter Type**

A Line of Business and Application specific data type used to organize and contextualize correspondence and used as a parameter to gather contextual data from a Resource.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Line of Business ID | The UUID of the Line of Business to which it belongs. |

###### Template Manager

**Template Label**

A label used to organize Templates and to align them to different Workflows.  There are two different types of labels - Full Template Labels and Fragment Template Labels.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |

**Template**

A collection of versioned configuration information and content required to manage a specific type of correspondence.  There are two different types of templates - Full Templates and Fragment Templates.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Labels | The labels to which this Template has been aligned. |

**Template Version**

The versioned configuration information and content required to manage a specific type of correspondence.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| Template ID | The immutable UUID of the Template this is versioning. |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Inputs | A collection of inputs that can be used to hydrate the content of the Template with user created data. |
| Resource IDs | The UUID's of any resources that are aligned to this Template, that can be used to hydrate the content of the Template with contextual data. |
| Content | The actual FreeMarker and textual content of the Template. |
| Restriction IDs | The UUID's of any restrictions that should be applied to correspondence created based on this Template. |
| Conditional Attachment IDs | The UUID's of any attachments that should be conditionally added to correspondence created based on this Template. |

**Conditional Attachment**

Files like PDF's that are attached to correspondence if certain conditions are met.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Conditions | A list of conditions that once met will result in it being attached to the correspondence. |
| Resource ID | The UUID of the Resource that provides the file to attach. |

**Condition**

A comparison or check used in Restrictions and Conditional Attachments.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Type | The condition type, such as a string-match or a boolean condition. |

**Restriction**

A condition based restriction on creating correspondence related to a specific Template Version.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Conditions | A list of the UUIDs for conditions that if met result in a Template Version being restricted. |
| Reason | A user-friendly description of the reason for this restriction. |
| Type | Determines the impact of this restriction - such as hiding it from the list of available templates, or disabling it and indicating why using the given Reason. |

**Input**

A way to collect data from a user that can be used to hydrate the content of a Template.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Name | Its user-friendly name. |
| Type | Determines the type of input, which informs things like the formatting options in the Content Editor of the Template Manager and which form elements to present to the user in the Enterprise Letter Virtual Interview Screen (ELVIS). |
| Default | The optional default value of the Input. |

###### Letter Manager

**Context**

The circumstances and the subject of the correspondence, beneath which all of the instanced letters and correspondence are organized.

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| ID | Its immutable UUID. |
| Workflow ID | The UUID of the Workflow for this context. |
| Key Parameter Type(s) | The type(s) of Key Parameter given for the context. |
| Key Parameters Value(s) | The actual unique Key Parameter value(s) that combined with the Workflow make up the unique context in which our correspondence can be organized. |

**Letter Instance**

| **Attribute Name** | **Description** |
| --- | --- |
| ID | Its immutable UUID. |
| Template Version ID | The UUID of the Template Version for which this instance was created and will be rendered against. |
| Context ID | The unique identifier related to the specific context within which this letter was instanced. |
| Created Date | The date this instance was created. |
| Input Values | A collection of the data collected from users based on the Inputs defined in the Template Version. |
| Data Resource Values | A collection of the data collected from data resources based on the Resources defined in the Template Version. |

#### Logical Data Viewpoint Model (DIV-2)

The following diagram defines relationships between data elements, providing a structured representation of data flows, entities, and attributes.

|  |  |
| --- | --- |
| |  | | --- | |  | |

#### Legacy Component Transitioning - Conceptual Rosetta Stone

The following diagrams outline how we have re-imagined existing functionality in a new agnostic and enterprise way, allowing for people to translate concepts from the legacy tools into their modern counterparts.

##### High Level Rosetta Stone

|  |  |
| --- | --- |
| |  | | --- | |  | |

##### Lower Level Rosetta Stone

|  |  |
| --- | --- |
| |  | | --- | |  | |

#### Physical Data Viewpoint Model (DIV-3)

N/A

## Process View (Business Process Model)

The Process View describes the behavior of the systems, the sequence of events, and how various tasks combine to the functionality described in the Use-Case View. It decomposes the systems into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). The Process View describes how systems interact, detailing the timing of messages or events.

### Main Business Process Model (OV-6d)

### Intake Processes

The following diagrams delineate the key processes and workflows within ECM-L, mapping the sequence of activities and information flow that drive operations.

#### Line of Business Intake Process

The intake process for a new Line of Business involves an ECM-Line of Business Intake Form, which outlines how the new Line of Business interacts with ECM-L system and its components, and identifies the Application(s) and any Connection Configurations the Line of Business wants to include in the intake process.  The Line of Business will submit the Line of Business Intake Form, and Connection Configuration Intake Form if needed, and create a Jira ticket. After submission of the Intake Form(s), there will be an Integration Meeting with stakeholders to introduce ECM-L capabilities, integration points and connection configurations.

[**Enterprise Correspondence Management - Line of Business Intake Form**](file:///\\UXENSVR\display\VAExternal\Enterprise+Correspondence+Management+-+Line+of+Business+Intake+Form)

#### Application Intake Process

The intake process for a new Application is performed by configuration outside of the ECM-L user interface.  It involves Service Tickets and System Teams and successful processing results in an Application becoming usable in ECM-L.

[Enterprise Correspondence Management - Application Intake Form](file:///\\UXENSVR\display\VAExternal\Enterprise+Correspondence+Management+-+Application+Intake+Form)

#### Connection Configuration Intake Process

The connection configuration intake process results in a successful integration between the ECM-L applications and an external Line of Business API.  This process begins with a Connection Configuration Intake Form filled out by the Line of Business, which results in a Change Request that loads the necessary secret and issuer key pair into the external API and into the individual ECM-L applications, and ultimately a Connection Configuration available in the Resource Manager in the Provisioned status.

The Connection Configuration intake form for a new LoB requesting connection to ECM-L will require the following:

* Point of Contact
* Line of Business
* Connection Name
* API Registration
* Vault Location
* OAS URL
* Certificates and Secrets
* API Permissions and Policies
* Swagger or OAS Specification
* Input Parameters
* Security Needs
* Transport Layers
* JWT
* JSON Path

Here is an example Change Request ticket that can be cloned in Jira:  [BPS-66120](https://jira.devops.va.gov/browse/BPS-66120) - Getting issue details... STATUS .

##### Process Workflow for Connection Configuration Intake

This process guides the intake and integration of new connections. It includes submission, approval, configuration, and confirmation across multiple systems.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| **Task** | **Description** |
| Submit Connection Configuration Intake Form | Line of Business submits an intake form to request a new connection configuration. |
| Create Change Request for External API Integration | Dev Team creates a change request to initiate the API integration process. |
| Submit Change Request to Release Management | Dev Team submits the change request for formal approval through release management. |
| Approve Change Request | Approval is given to proceed with the connection changes. |
| Mark Change Request as In Progress | The change request status is updated to indicate active work. |
| Create and Add Secret/Issuer Values in ECM-L Vault and External Vault | DevOps team populates secret and issuer values needed for integration in ECM-L Vault and External Vault. |
| Merge PR to Provision Connection Configuration | Pull request is merged to apply the new configuration. |
| Confirm Successful Integration (Resource Manager) | Confirm that the integration is successful in the Resource Manager. |
| Confirm Successful Integration (Letter Manager) | Verify that Letter Manager has been successfully integrated. |
| Confirm Successful Integration (ECM-L API) | Ensure that ECM-L API reports successful integration. |
| Mark Change Request as Done | Dev Team marks the request as completed once all validations pass. |
| Process the Provisioned Connection Configuration in Resource Manager | Resource Manager finalizes configuration. |
| Make Corrections to Change Request | If issues arise, the Dev Team adjusts the change request and may re-submit it. |
| Mark Change Request as Draft | If changes are incomplete, the request is returned to draft status for revision. |

#### Make Connection Available

The Line of Business Administrator is able to make a Connection Configuration Available.

|  |  |
| --- | --- |
| |  | | --- | |  | |

| **Task** | **Description** |
| --- | --- |
| Process the PROVISIONED Connection | Process starts with a PROVISIONED Connection. |
| View List of Connection Configuration(s) | LOB Admin views the list of connection configurations assigned to their Line of Business (LOB). |
| Select appropriate PROVISIONED Connection | The LOB Admin selects a connection with a PROVISIONED status to make available. |
| Test PROVISIONED Connection by checking API | The Admin tests the connection’s API to ensure it works. |
| Connection Successful? | Decision point: was the API test successful? |
| If Yes → Select Appropriate PROVISIONED Connection | If the test is successful, the connection status updates to AVAILABLE. |
| Connection Configuration Published Event | Event trigger, confirming Connection is AVAILABLE. |
| → End | Process ends, connection is AVAILABLE for Resources. |
| If No → Connection Configuration Intake Process | If the test fails, the Admin must return to the Connection Configuration Intake process. |

#### Connection Decommissioning Process

This flow handles retiring a connection while considering replacements and dependencies. It ensures resource and template clean-up before removal.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| **Task** | **Description** |
| Confirm Decommissioning | Initiate the process by confirming the Connection Configuration should be decommissioned. |
| Replacement Connection? | Check if a replacement Connection exists for the one being decommissioned. |
| Evaluate Replacement Connection | If a replacement Connection exists, that Connection must be evaluated for readiness and availability. |
| Replacement Connection State | Assess the state of the replacement Connection: AVALIABLE, PROVISIONED, or DECOMMISSIONED. |
| Wait for ConnectionPublishedEvent of replacement | If the state of the replacement Connection is PROVISIONED, pause the process until the replacement Connection is fully published (i.e., AVAILABLE). |
| Cancel with Error | If the state of the replacement Connection is DECOMMISSIONED, the process is Canceled due to error because a DECOMMISSIONED Connection cannot be used as replacement. |
| Get all Resource versions derived from this Connection | If there is no replacement Connection OR if the replacement Connection is AVAILABLE, retrieve all Resource Versions derived from the Connection being replaced. |
| Resource version Retirement for each Resource | Retire all Resource Versions that use the Connection being DECOMMISSIONED.  This is a subprocess that happens for each Resource Version being retired. See model detailing the process for Retiring a Resource Version. |
| Get all Template versions using the retired Resources | All ACTIVE Template Versions using the Resource Version being RETIRED are identified. |
| Template version Retirement for each Template | Retire all Template Versions using the to-be RETIRED Resource Version. This subprocess is detailed in the process model for Retiring a Template Version. |
| Connection Removed Event | Trigger an event confirming successful removal of the connection. |

#### Create Resource Version (\*\*awaiting final approval)

|  |  |
| --- | --- |
| |  | | --- | |  | |

\*Table Needed

#### Resource Version Creation Process (For already existing Resource Version)

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| **Task** | **Description** |
| Select Resource version draft | If a draft of the Resource version already exists it is selected |
| Edit draft if needed | If the draft selected is not ready to be tested or failed a previous test it will be edited |
| Create new Resource version draft | If no draft exists of the new Resource version, a new draft will be created |
| Test Resource version | Resource version is tested to see if a call to the api can be completed |
| Publish Resource version | The Resource version passes testing and is made AVAILABLE |
| Resource Available Event | Making the new Resource version AVAILABLE notifies Templates using the previous Resource version of a new version being made AVAILABLE |

#### Create Workflow

This diagram outlines the steps the LOB Admin follows to create a new workflow.

|  |  |
| --- | --- |
| |  | | --- | |  | |

| **Task** | **Description** |
| --- | --- |
| Log in to Resource Manager | The LOB Admin logs into the Resource Manager application. |
| View Application List | The user views the list of available applications. |
| Select Application | The user selects a specific application to create a workflow for. |
| Create New Workflow | The user initiates the creation of a new workflow. |
| Add Template? | A decision point to determine whether to add one or more templates. |
| Select One or More Templates | If yes, the user selects the relevant templates for the workflow. |
| Workflow Created | The workflow is successfully created and saved in the system. |

#### Update Workflow

This diagram outlines the steps the LOB Admin follows to update a workflow.

|  |  |
| --- | --- |
| |  | | --- | |  | |

| **Task** | **Description** |
| --- | --- |
| Log in to Resource Manager | The LOB Admin logs into the Resource Manager application. |
| View Application List | The user views the list of all available applications. |
| Select Application | The user selects an application to view associated workflows. |
| View Workflow | The user accesses the list of workflows under the selected application. |
| Select Workflow | The user selects a specific workflow to update. |
| Edit Workflow | The user edits the workflow by changing its name or adding/removing templates. |
| Workflow Updated | The system saves the changes, and the workflow is successfully updated. |

#### Retire Workflow

This diagram outlines the steps the LOB Admin follows to retire a workflow.

|  |  |
| --- | --- |
| |  | | --- | |  | |

| **Task** | **Description** |
| --- | --- |
| Log in to Resource Manager | The LOB Admin logs into the Resource Manager application. |
| View Application List | The user views a list of all available applications. |
| Select Application | The user selects a specific application containing the workflow to delete. |
| View Workflow | The user views workflows associated with the selected application. |
| Select Workflow | The user selects the specific workflow they want to retire. |
| Retire Workflow | The user initiates the retirement process. A notification alerts them of the consequences of this action. |
| Workflow Retired | The selected workflow is retired from the system. |

#### Overall Changeset Process (\*\*awaiting final approval)

This diagram outlines the complete lifecycle of a changeset from draft creation to publishing. It illustrates key states, decision points, and transitions across testing, approval, and scheduling.

|  |  |
| --- | --- |
| |  | | --- | |  | |

\*Table Needed

#### Create Template Version Subprocess (\*\*awaiting final approval)

|  |  |
| --- | --- |
| |  | | --- | |  | |

\*Table Needed

#### Template Version Retirement Subprocess (\*\*awaiting final approval)

This diagram details how different template states are handled during retirement. Template Version Retirement is a stand-alone process, but also integral to the Connection Decommissioning process.

|  |  |
| --- | --- |
| |  | | --- | |  | |

\*Table Needed

#### Template Version Testing Subprocess (\*\*awaiting final approval)

Before a Changeset is Scheduled and Published we want to make sure the new versions of the Templates in it reflect the changes we want, so we check that by creating and using Test Cases that let us view rendered PDF's for those templates, using sample data.  Test Cases are created for specific Template Versions, and are used inside of Changesets to test them, but live outside of the Changeset, and can be re-used by future Changesets to regression test changes to those templates.

|  |  |
| --- | --- |
| |  | | --- | |  | |

\*Table Needed

#### Changeset Approval Process (\*\*awaiting final approval)

This flow governs the approval of tested templates within a changeset. It includes subprocesses for template approval, validation, and cancellation handling.

|  |  |
| --- | --- |
| |  | | --- | |  | |

\*Table Needed

#### Letter Finalization Process

This swimlane diagram illustrates how letters are reviewed, approved and finalized.

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| **Task** | **Template** |
| Letter marked as 'Ready for Review' | The letter is flagged in the system as ready for review by a Letter Editor |
| Select letter Instance to Review | Letter Finalizer selects the letter instance that is ready for review. |
| Access Saved Data | Retrieve relevant letter data from the Letter Database (DB). |
| Get Template | Template Manager retrieves the appropriate template or template for the letter. |
| Populate Data in Template | Letter Manager injects retrieved data into the letter template template. |
| Display Rendered Letter | Show the rendered PDF to the LoB Admin for approval review. |
| Letter Good to Approve? | Decision point- if the letter is acceptable, proceed to finalize; otherwise return it to draft. |
| Return Letter to Draft State | If not approved, the letter is sent back to draft state for revision. |
| Finalize Letter | If approved, the letter is finalized and saved as the final version. |
| Call ECM-L API | Trigger the ECM-L API to initiate PDF rendering. |
| Render PDF | ECM-L API renders the letter into a PDF format and makes it available to the LoB system/System. |
| View Final PDF | Allow the finalized PDF to be viewed by the approver or system. |

##### External Letter Finalization

This diagram outlines the automated generation and storage of final letters via ECM-L API. It integrates multiple services to render and store the final PDF, followed by a success alert.

##### 

|  |  |
| --- | --- |
| |  | | --- | |  | |

|  |  |
| --- | --- |
| **Task** | **Description** |
| Generate Final Letter Workflow | Start the external letter finalization process. |
| Call ECM-L API with Letter UUID | Trigger ECM-L API using the unique identifier of the letter. |
| Receive Request | ECM-L API receives the request to process the final letter. |
| Access Saved/Frozen Letter Data | Letter Manager API retrieves frozen data from the Letter Manager DB. |
| Retrieve Template | If the letter template is marked 'Active', it's retrieved from the ECM-L DB. |
| Populate Data in Template | Data from the letter is populated into the retrieved template for rendering. |
| Render Final Letter as PDF | ECM-L API composes and renders the final version of the letter in PDF format. |
| View Final PDF | Allow users to view the generated PDF before it's finalized. |
| Send Final PDF to Storage | Push the finalized PDF to a storage system accessible to downstream applications. |
| Store Final PDF | LoB Repository stores the finalized letter document. |
| Display Success Alert Message | User is notified that the final PDF was successfully created and stored. |
| View Success Alert Message | User confirms the success message to close the workflow. |

### Process Sequence - Systems Event-Trace Description (SV-10c)

The following diagrams illustrate chronological sequences of events and interactions within different workflows of ECM-L, providing a dynamic view of system behavior over time.

**Letter Creation Sequence (SV-10c)**

|  |  |
| --- | --- |
| |  | | --- | |  | |

## Implementation View

The Implementation View (sometimes called the Development View) will be further completed alongside the Development of ECM-L.

## Deployment View

| **Purpose** | This section is to document any deviations or unique conditions that are not outlined by the standard practices of the BIP Platform which may be found in the Deployment View section of BIP Architecture Documentation (2001AM). |
| --- | --- |
| **Acceptance Criteria** | * Environments * System Configuration * System Monitoring and Metrics * System Logging and Auditing * Security Strategy |
| **Diagram Type** | UML Component or Block if needed |

### Environment Mapping

| **Environment** | **Description** |
| --- | --- |
| DEV | Development Environment |
| TEST | Development testing environment |
| INT | Development Integration testing environment |
| IV&V/SQA | Independent Verification & Validation / System Quality Assurance |
| UAT | User Acceptance Testing |
| PAT | Pre-Acceptance Testing |
| PROD Development Test (PDT) | Environment used by Development to Test changes and patches |
| DEMO | Demonstrations |
| PREPROD | Environment used to test new releases prior to deploying to PROD |
| PERF | Performance testing |
| Cost of Living Adjustment (COLA) | Environment used to assist in determining Cost of Living changes |
| PROD Test (PRODTEST) | Environment to test PROD changes and patches prior to deploying to PROD |
| PROD | Production |

### System Configuration

No deltas from BIP standard practices.

### System Monitoring and Metrics

No deltas from BIP standard practices.

### System Logging and Auditing

No deltas from BIP standard practices.

#### ****Long-Term Provenance****

Letter Instance Provenance refers to the ability to chronicle how a Letter Instance was created, i.e., by whom, and when, while also tracing the origins of every element, including configured content as well as all data values input. A Letter Instance is derived from a Template Version, thus maintaining a comprehensive record of the Template Version, from which it was derived, including the Template Version’s unique content and the Resource Version(s) it used, will ensure traceability of all the configured content.  Chronicling the data input ensures the verifiable origin of the data necessary for auditing and validation purposes.  For all data fields, the record will log the data/time of input, the source of data (e.g., the Resource Version from which the data originated (API or URL) or user, if manually input), and the actual data values entered in the Letter Instance.

The diagram below illustrates how each data element depicted in the Letter Instance (Orange, Red, Yellow) can be traced to its origin.  For example, the Data Structure input in the Letter Instance can be traced to the Orange Resource (configured in the Template Version) which used a Connection Configuration to API #1 to access the data used to populate the Letter Instance. The Data File can be traced to the Red Resource (configured in the Template Version) which accessed the Data File from API #2 via the Connection Configuration and Other Data can be traced to the User who manually input the data in the Letter Instance.

|  |  |
| --- | --- |
| |  | | --- | |  | |

### Security Strategy

ECM-L is heavily dependent on policies concerning authentication and authorization and will utilize the **B**enefits **A**pplication **A**uthori**Z**ation framework (BAAZ) as the propagation mechanism for an LOB’s authorization policies.  BAAZ is a policy framework that uses Open Policy Agent (OPA) with a central policy store where LOBs can manage their own policies and any application can create its own policies and apply policies that are managed by a central security authority and appropriately audited. (See [Benefit Applications Authorization Services](https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fconfluence.devops.va.gov%2Fdisplay%2FVAExternal%2FBenefit%2BApplications%2BAuthorization%2BServices&data=05%7C02%7C%7C0bd70409f48049c1fce308dd7186b54a%7Ce95f1b23abaf45ee821db7ab251ab3bf%7C0%7C0%7C638791545566800716%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=1xdT4%2B8uCLMuyoE4Rzc1r414W9DEsGsqIn1MpOTGKio%3D&reserved=0)). As an enterprise system, LOBs will have self-service capabilities for maintaining their own governance and security policies.  An important part of the Intake Process will be for each LOB to establish a set of policies for that LOB, including policies for access to Applications and Workflows.

Recommendation

## Analysis of Alternatives

Within the VA, individual Lines of Business (LOBs) use various applications to generate and manage letters, with each application tailored to the specific LOB business needs.

When devising an Analysis of Alternatives (AOA) for a transition to Enterprise Content Manager for Letters (ECM-L), each LOB must consider the cost, risk and benefit of continued use of their current application for letter generation versus implementing ECM-L. Since ECM-L will be implemented in a phased approach, scaling from simple letters to more complex letters, a LOB must also consider the complexity of the letters they need to generate to determine the appropriate transition course. If the complexity of a LOB-generated letter exceeds the capability of ECM-L, an interim solution may be required.

## Recommendation

## Phases and Roadmap

### ****Phases****

A breakdown of the pieces of functionality supported by each phase of development.

#### Phase 0: Foundations

The basic foundations of the ECM-L systems, allowing for users to generate simple letters with no dynamic or contextual content.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Line of Business Management | Create Workflow |  |  |  |  |
| Create Key Parameter Type |  |  |  |  |
| Changeset and Template Management | Create Changeset |  |  |  |  |
| Create, Update, Retire Template |  |  |  |  |
| Create and Run Test Cases |  |  |  |  |
| Schedule Changeset |  |  |  |  |
| Create, Align Template Label |  |  |  |  |
| Get Template List |  |  |  |  |
| Letter Management | Create, Edit, Delete Letter |  |  |  |  |
| Render Letter |  |  |  |  |
| Finalize Letter |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~2.5 PIs

#### Phase 1: Simple Inputs

Simple inputs that do not use any external resources are supported.  No images or attachments, and no gathered data, just what the User Provides.  These can be formatted and inserted into Template Content in the Template Manager.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Simple Input Support | Text Inputs |  |  |  |  |
| Number Inputs |  |  |  |  |
| Boolean Inputs |  |  |  |  |
| Date Inputs |  |  |  |  |
| Physical Address Inputs |  |  |  |  |
| Option Selection Inputs |  |  |  |  |
| Image Resource Support | Images in Content |  |  |  |  |
| Storage and Distribution | Claim Evidence |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~2 PIs

#### Phase 2: Images and Attachments

Images and Attachments can now show up in a Letter.  A new Attachment Input lets users choose which attachments should be attached to a Letter.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Complex Input Support | Attachment Selection Inputs |  |  |  |  |
| Recipient Selection Input |  |  |  |  |
| Storage and Distribution | Package Manager |  |  |  |  |
| Attachment Resource Support | Attachments |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~1.5 PIs

#### Phase 3: Data Resources

Data Resource fields can now be used in a Letter, filling the Template with contextual data.  These fields can be formatted and inserted into Template Content in the Template Manager. Inputs can be defaulted to manually inputted data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Data Resource Support | Text Fields |  |  |  |  |
| Number Fields |  |  |  |  |
| Boolean Fields |  |  |  |  |
| Date Fields |  |  |  |  |
| Physical Address Fields |  |  |  |  |
| Field Groups |  |  |  |  |
| Changeset and Template Management | Label Creation |  |  |  |  |
| Complex Input Support | Inputs with Data Resource Defaults |  |  |  |  |
| Input Groups |  |  |  |  |
| List Builder Inputs |  |  |  |  |
| Option Select Sub-Inputs |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~1.5 PIs

#### Phase 4: Fragments, Conditional Attachments and Restrictions

Allows users to insert and include one template into another template. Adds in the ability to create Conditional Attachments and add Restrictions to Templates.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Fragments | Include Fragment Templates in a Template |  |  |  |  |
| Fragment Selection Input |  |  |  |  |
| Conditional Fragments |  |  |  |  |
| Conditional Attachments | Conditional Attachments |  |  |  |  |
| Restrictions | Letter Restrictions for Single and Existing Letters |  |  |  |  |
| String-match Restrictions |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~1.5 PIs

#### Phase 5: Attachment Field Population

Attachments with form field elements can be populated using Data Resource information.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Attachment Resource Support | Attachment Field Population |  |  |  |  |
| Child Letters | Child Letters |  |  |  |  |
| Conditional Child Letters |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~1.5 PI

#### Phase 6: Bar Code Support

Allows for child letters to be created and managed in the Letter Manager. Also allows for different types of bar codes and QR codes to be inserted into template content, generated based on Data Resource information.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **RM** | **TM** | **LM** | **API** |
| Bar Code Support | QR Codes |  |  |  |  |
| Bar Codes |  |  |  |  |
| Address Bar Codes |  |  |  |  |

Letters Supported:

<TBD>

Estimated LOE:

~1 PI

#### Phase Map

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Functionality Group** | **Functionality** | **P0** | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** |
| Line of Business Management | Create Workflow |  |  |  |  |  |  |  |
| Create Key Parameter Type |  |  |  |  |  |  |  |
| Changeset and Template Management | Create Changeset |  |  |  |  |  |  |  |
| Create, Update, Retire Template |  |  |  |  |  |  |  |
| Create and Run Test Cases |  |  |  |  |  |  |  |
| Schedule Changeset |  |  |  |  |  |  |  |
| Create, Align Template Label |  |  |  |  |  |  |  |
| Get Template List |  |  |  |  |  |  |  |
| Template Labels |  |  |  |  |  |  |  |
| Letter Management | Create Letter |  |  |  |  |  |  |  |
| Render Letter |  |  |  |  |  |  |  |
| Delete Letter |  |  |  |  |  |  |  |
| Finalize Letter |  |  |  |  |  |  |  |
| Simple Input Support | Text Inputs |  |  |  |  |  |  |  |
| Number Inputs |  |  |  |  |  |  |  |
| Boolean Inputs |  |  |  |  |  |  |  |
| Date Inputs |  |  |  |  |  |  |  |
| Physical Address Inputs |  |  |  |  |  |  |  |
| Option Selection Inputs |  |  |  |  |  |  |  |
| Complex Input Support | Input Groups |  |  |  |  |  |  |  |
| Attachment Selection Inputs |  |  |  |  |  |  |  |
| List Builder Inputs |  |  |  |  |  |  |  |
| Option Selection Sub-Inputs |  |  |  |  |  |  |  |
| Recipient Selection Input |  |  |  |  |  |  |  |
| Inputs with Data Resource Defaults |  |  |  |  |  |  |  |
| Image Resource Support | Images in Content |  |  |  |  |  |  |  |
| Attachment Resource Support | Attachments |  |  |  |  |  |  |  |
| Conditional Attachments |  |  |  |  |  |  |  |
| Attachment Field Population |  |  |  |  |  |  |  |
| Letter Restrictions | Single Draft Only |  |  |  |  |  |  |  |
| Existing Draft Restrictions |  |  |  |  |  |  |  |
| String Match |  |  |  |  |  |  |  |
| Data Resource Support | Text Fields |  |  |  |  |  |  |  |
| Number Fields |  |  |  |  |  |  |  |
| Boolean Fields |  |  |  |  |  |  |  |
| Date Fields |  |  |  |  |  |  |  |
| Physical Address Fields |  |  |  |  |  |  |  |
| Field Groups |  |  |  |  |  |  |  |
| Fragments | Fragment Templates |  |  |  |  |  |  |  |
| Fragment Selection Input |  |  |  |  |  |  |  |
| Conditional Fragments |  |  |  |  |  |  |  |
| Child Letters | Child Letters |  |  |  |  |  |  |  |
| Conditional Child Letters |  |  |  |  |  |  |  |
| Bar Code Support | QR Codes |  |  |  |  |  |  |  |
| Bar Codes |  |  |  |  |  |  |  |
| Address Bar Codes |  |  |  |  |  |  |  |
| Storage and Distribution | Claim Evidence Support |  |  |  |  |  |  |  |
| Package Manager Support |  |  |  |  |  |  |  |