

Deferrals MFE & API Solution Document

Name	Value
Solution Status	DRAFT
Solution Type	NEW Tennant
Recommended BIP Tenant Namespace	VBMS CORE
Recommended System Acronym	
Recommended BIP Tenant App(s)	bip-vbms-core
Sponsoring organization	VBA
Number of users	
Estimated Monthly Cost	
Privacy	PHI and PII
Will need 508 Compliance	Yes
Deployment date	
Solution and Support Links	Solutioning: Deferrals Modernization 4+1

- [Introduction](#)
 - [Problem Statement](#)
 - [What Problem Needs to be Solved?](#)
 - [Business Needs](#)
 - [Technical Needs](#)
- [Recommendation](#)
- [Use Case View](#)
 - [Capability Viewpoint Vision \(CV-1\) \(VASI Description\)](#)
 - [Capability Viewpoint Taxonomy \(CV-2\)](#)
 - [Primary Systems High Level Operational Viewpoint \(OV-1\)](#)
 - [Key Scenarios / Use Cases: Ideal workflows and primary system interactions](#)
- [Logical View](#)
 - [Systems Interface Description \(SV-1\)](#)
 - [State Transitions \(OV-6b\)](#)
 - [Data Models](#)
- [Process View](#)
 - [Current State Process Model](#)
 - [Future State Process Model](#)
 - [Proposed Main Business Process Model \(OV-6d\)](#)
 - [Process Sequence - System Event-Trace Description \(SV-10c\)](#)
- [Implementation View](#)
 - [Primary System Technologies \(SV-9\) \(VASI Technology Components\)](#)
- [Deployment View](#)
 - [Environment Mapping](#)
 - [System Configuration](#)
 - [System Monitoring and Metrics](#)
 - [System Logging and Auditing](#)
 - [Security Strategy](#)

Introduction

Problem Statement

We recommend migrating all business logic relating to Deferrals, and related actions on Veteran Claims processing, to a new Deferrals application utilizing a single micro front end (MFE). The MFE then has the capability to be embedded in the UI of VBMS-Core, VBMS-Ratings, VBMS-Awards, and any other end-user applications added to the VBMS application stack during the modernization of Claims processing for the VA.

Deferrals information captures work on a claim that is incomplete or otherwise needs to be corrected, causing a claim to flow backward in the Claim Lifecycle so it can be reworked. The Deferrals process is complex, crossing multiple workflows within VBMS-Ratings and VBMS-Core and impacting (1) the downstream VBMS-Awards process and (2) assignment of claims to VA employees via the National Work Queue (NWQ). For details on the Deferrals process and workflow view Deferrals in VBMS-Core Reference Guide.

For the modernization of VBMS applications we are utilizing the Micro Frontend architecture. Micro frontends are small applications that manage the user interface of a small, reusable subset of functionality in the system. This effort will focus on decoupling the Deferrals process from VBMS-Core and using the Micro Frontend (MFE) architecture for Deferrals. To support the modernization and evolution of Deferrals application functionality, in addition to resolving known 508 and UI/UX compliance issues, a new Deferrals user interface is necessary to provide a stable and compliant foundation for future application growth.

By utilizing a micro frontend approach, teams can focus on smaller areas of overall user workflows which ultimately means more rapid delivery. This approach also allows sections of user interfaces to adopt newer technologies and ideas without forcing global dependencies, and since each micro frontend has a separate build and deployment process, micro frontends can be deployed more efficiently without affecting other micro frontends.

Link to Jira Initiative:

- VBMSR-32096 - Capability to interface with Deferrals in a MFE
- VBMSR-32140 - Capability to Provide Deferrals Data via Deferrals API

What Problem Needs to be Solved?

Currently, Deferrals service code resides in the business logic layer of the legacy VBMS-Core application structure. The mission-critical Deferrals process:



1. Is reliant on outdated systems that involve switching from VBMS applications.
2. Presents unnecessary challenges to developers currently working on Modernization.
3. Is counter to the VA's current VBMS Modernization efforts to move away from Core's monolithic application architecture.

Deferrals currently resides in CORE where users have to 'swivel-chair' to another system (Ratings, Awards). In decoupling Deferrals from CORE into an MFE we are preparing for future state that will allow Deferrals to be mounted into vbms-home. This is a intermediary phase which will lead to the streamlining process to remove the 'swivel-chair' process that the users are currently facing. By creating this MFE we are setting up an environment that will allow for integration of deferrals into other systems (Ratings and Awards) in the future.

Reflecting industry best practices and standards, today the VA favors a more modular approach to application architecture.

We determined an intermediary phase is necessary to develop a Deferrals MFE and a Deferrals API first, in order to address any downstream streamlining of workflows in other VBMS applications.

Business Needs

Is this a defect or an enhancement?	Enhancement
Who reported the need?	Joshua Will and Mary Kate Abler in Theme BPT-3396,
When was the need noticed?	Unknown
When did/will it become a problem?	Unknown
Is there an existing Requirement? Strategic Objective?	Strategic Objective: BPT-3396, 1. Theme:  CPPLM-425Deferral Modernization: Implementation a. Initiatives: VBMSR-32096 2. Theme:  CPPLM-425Deferral Modernization: Implementation a. Initiative: VBMSR-32140
What existing resources, tickets, emails, etc. might serve to better understand the problem?	Solutioning: Deferrals Modernization 4+1, Deferrals in VBMS-Core Reference Guide
What business processes are involved?	Deferrals
Who is the business owner for the impacted systems or processes?	OBA

Who are the business decision makers for solving the problem?	<ul style="list-style-type: none"> • OBA: Primary: Jacqueline Simpson (Jacqueline.Simpson@va.gov), Senior Management and Program Analyst • OBA: Secondary: Michael Palmer (Michael.Palmer5@va.gov), Program Analyst • Compensation Services - Pepe Cervantes • Office of Administrative Review - Lauren Sylvia or Autumn Paschke • Pension & Fiduciary - Michael Hart or Bartolo Reyes • National Work Queue - Kelli Gorley
What is the historical context? How did we get where we are?	Prior to the involvement of the Requirements and Solutioning ARTs, the IT response to this request was that addressing this business need was not feasible due to the siloed architecture of VBMS-R and VBMS-Core, and that an entire rewrite of the deferral logic would be needed. SOPs and work-arounds that have become the accepted practice for handling deferrals will persist if nothing is done to address the gap in functionality.
What happens to the business stakeholders if nothing is done?	The SOPs and work-arounds will continued to be used for handling deferrals.
Of the various aspects of this problem, which of them does the business consider as critical flaws? Which are nice-to-have changes?	the siloed architecture of VBMS-R and VBMS-Core, and that an entire rewrite of the deferral logic would be needed

Technical Needs

Describe the problem in technical terms	The current deferrals service code only lives in the monolith vbms-core and is not accessible to other legacy monoliths of vbms modern applications.
What are the architectural components in the problem's context?	Java Spring Boot monoliths are the primary architectural components that are in the problem's context. MFEs and an external API will be the new solution context within vbms-home.
What are the integration points that might be impacted by this problem?	<p>Most immediately:</p> <ul style="list-style-type: none"> • legacy workflows via: <ul style="list-style-type: none"> ◦ vbms-core ◦ vbms-rating • modern workflows via: <ul style="list-style-type: none"> ◦ vbms-home
What technologies are involved?	<ul style="list-style-type: none"> • Single Spa MFE • React • Spring Boot
Is this related to already-stated technical debt?	These are technologies utilized across modernization initiatives to address technical debt and reusability.
If the problem is with a technology, what were the reasons for using this technology in the first place?	vbms-core monolith most likely was originally intended to house all vbms applications, however ratings was built as a separate monolith which has created technical debt (i.e. not having access to deferrals service logic from within ratings).
Has an attempt been made before to fix the problem?	Not that I'm aware of.
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?	vbms-rating is the primary driver for this problem resolution, however it affects any deferrals data vbms consumer.
Are there current mitigations that help alleviate the impact of the problem?	Currently, users navigate to vbms-core for deferrals, but this doesn't mitigate the problem; it is the only path for the user currently.
What are the technical dependencies for this? What does this depend on? What depends on it?	The Deferrals MFE depends on supporting claim context from future MFE framework infrastructure such as the future Claim Navbar MFE, Participant Subheader MFE, and Veteran Summary Panel MFE.
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	TBD.

Recommendation

The possible solutions to this would be: a legacy implementation where the MFE is embedded into legacy applications (legacy embed); or creation of the MFE and storage of it within VBMS-home where it can be accessed by any application (modern embed). Historically there has only ever been two integration options standalone or mounting into legacy core, but now there is the option of mounting into vbms-home.

This modern embed is a solution proposing to mount Deferrals MFE into Vbms-home. In embedding the Deferrals MFE into VBMS-Home we allow the MFE to be accessible by all external applications. So after the creation of the MFE in VBMS-home, it can be accessed by any application (Ratings, Awards, Core) by utilizing the MFE connector. This means that instead of developing separate embeded solutions in each application, each application can instead have a connector that leads them to Deferrals MFE with VBMS-Home. This aligns with current modernization practices and does not make the Modernized solution reliant on outdated systems.

The legacy embed solution would have to be completed for each VBMS environment separately (Ratings, Awards, Core). This solution is a possibility but relies on embedding BCL MFE into the React a business logic layer of the legacy VBMS-Core application structure. This does not align with the VA's current VBMS Modernization efforts to move away from Core's monolithic application architecture, and presents unnecessary challenges to developers currently working on Modernization. There are ARTs currently working on Ratings Modernization concurrently to our work in Deferrals Modernization.

Our Recommendation is to go forward with the modern embed solution versus mounting Deferrals MFE into the legacy monoliths of VBMS-Core or VBMS-Ratings.

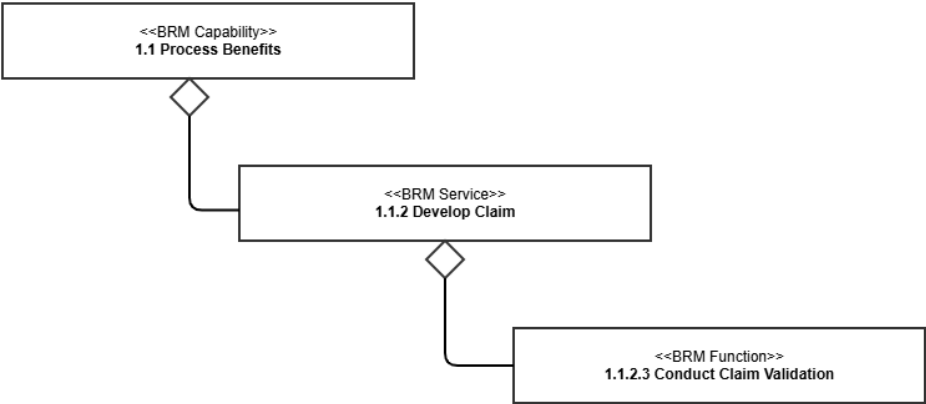
Use Case View

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

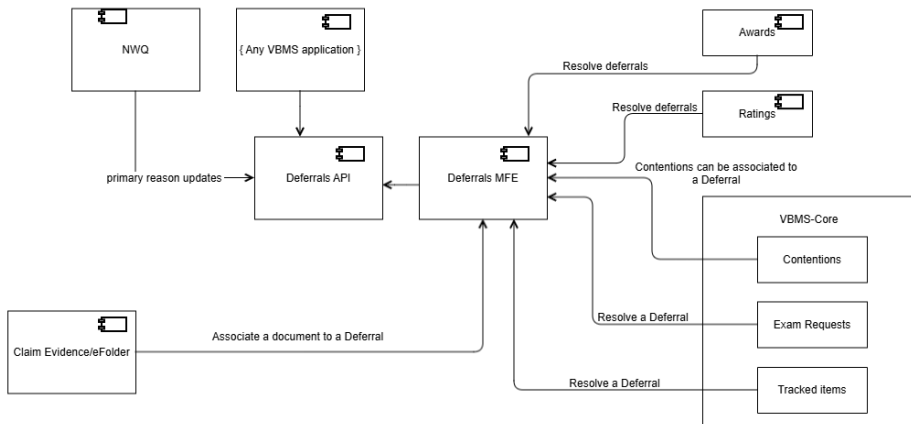
TBD

Presently, VBMS users must navigate to VBMS-Core to interface with deferrals. There is a business need to modernize deferrals making the system more accessible to other VBMS applications. This initiative proposes to migrate all business logic relating to deferrals to a new freestanding Deferrals application utilizing a single micro front end (MFE) and a Deferrals API. The MFE design affords flexibility for architecture integration. A Deferrals MFE will make accessible and standardize the deferral user experience for VBMS applications.

Capability Viewpoint Taxonomy (CV-2)



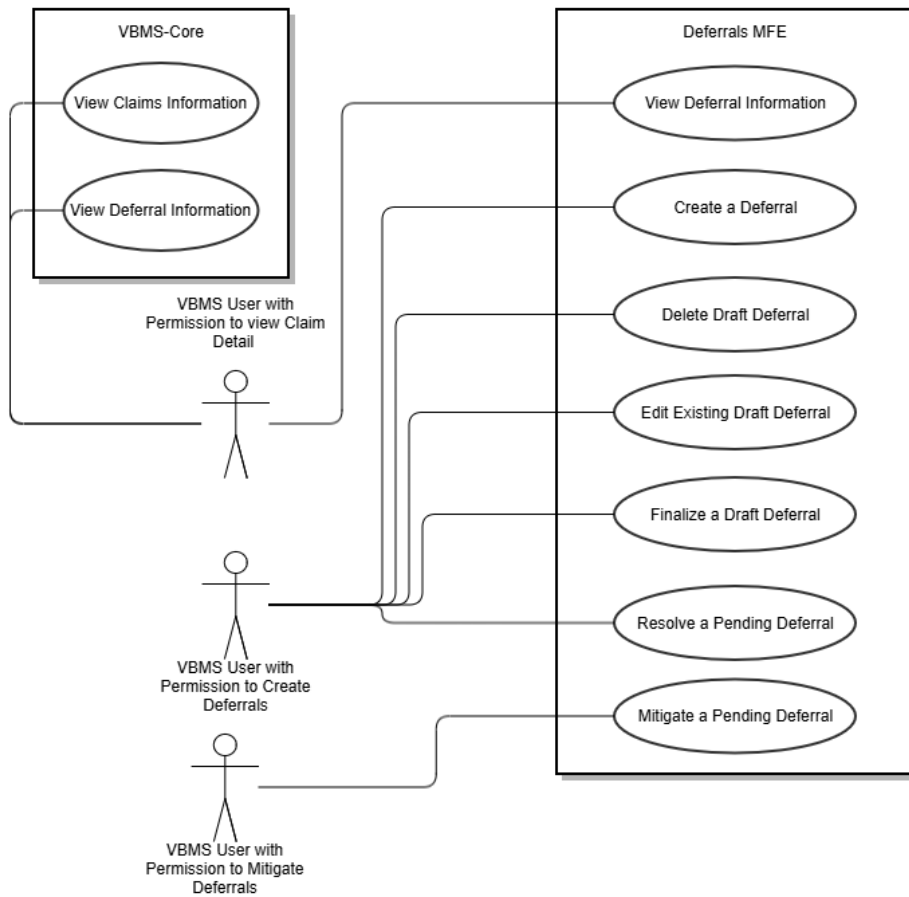
Primary Systems High Level Operational Viewpoint (OV-1)



For any VBMS applications that need to access the Deferrals MFE, those applications would either look to embed the Deferrals MFE into their root or more simply those applications would look to link out to the Deferrals MFE located in vbms-home. The Deferrals API is intended to primarily serve the Deferrals MFE for CRUD operations, but it can also be called directly by any potential consumers in their application.

Component	Interaction Description
Claim Evidence/eFolder and Deferrals MFE	Associates a document to a deferral in Deferrals MFE
NWQ and Deferrals API	Primary reason updates
Any VBMS App and Deferrals API	Any application can retrieve deferral information from Deferrals API
Awards and Deferrals MFE	Awards interacts with Deferrals MFE to perform CRUD operations
Ratings and Deferrals MFE	Ratings interacts with Deferrals MFE to perform CRUD operations
Contentions and Deferrals MFE	Contentions can be associated to a deferral in Deferrals MFE
Exam Requests and Deferrals MFE	Exam requests interacts with Deferrals MFE to perform CRUD operations - the creation of an Exam Request will change the state of a Deferral from Draft to Resolved or Pending to resolved.
Tracked items and Deferrals MFE	Tracked Items interacts with Deferrals MFE to perform CRUD operations - the creation of an Tracked Item will change the state of a Deferral from Draft to Resolved or Pending to resolved.

Key Scenarios / Use Cases: Ideal workflows and primary system interactions



Main Actor

Name	Description
VBMS User with Permission to Create Deferrals	Allow users to create a claim deferral
VBMS User with Permission to view Claim details	Allows the user to view detailed information about a claim, including the end product code, date of claim.
VBMS User with Permission to mitigate Deferrals	Allows users to manually resolve a claim deferral

System/Tenant/Application Description

System Name	Description
VBMS-Core	System in which most Claims are developed and processed.
Deferrals MFE	Micro front end used to create, read, edit, and delete deferrals to claims.

Main User Functions (Use Cases)

Actor	System	Ability	Details	
			Pre conditions	Post Conditions

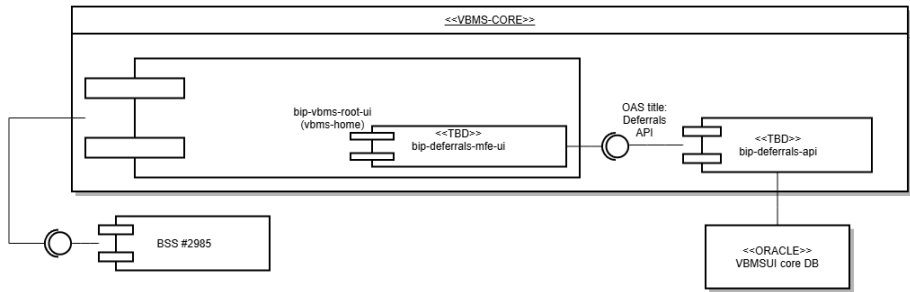
VBMS User with Permission to Create Deferrals	Deferrals MFE	Create Deferral	<ul style="list-style-type: none"> Claim is currently assigned to the user User has the PDD permission "Create Claim Deferral" Claims is not in Initial Development phase There is a tracked item other than the Automated 5103 Notice Required Claim is not cancelled or closed There is not an existing draft deferral on the claim (unless it is in Pending Authorization) 	<ul style="list-style-type: none"> Upon creating the claim is put into "Draft" Status Deferral is set as "avoidable" or "unavoidable" based on combination of Primary Reason and deferral reason If draft deferral is not finalized within 10 days it will be removed and a record of it stored
VBMS User with Permission to view Claim details	Deferrals MFE /VBMS-Core	View Deferral Information	User is currently authorized to view deferrals	User will be able to view deferral information on claims after they are closed as long as the deferral status is pending or resolved
VBMS User with Permission to Create Deferrals	Deferrals MFE	Resolve a Pending Deferral	<ul style="list-style-type: none"> to manually resolve the user must have the permission "Manually Resolve Deferral" <ul style="list-style-type: none"> User must also select a reason as to why the deferral is being resolved manually To automatically resolve the deferral the user must complete certain actions based on the claim lifecycle 	<ul style="list-style-type: none"> Deferral status is updated from "Pending" to "Resolved" If manually resolved the following is saved: <ul style="list-style-type: none"> timestamp of when the resolution occurred the name and RO of the user who resolved the deferral selected resolution reason free text comment VBMS-Core work item is automatically completed
VBMS User with Permission to Create Deferrals	Deferrals MFE	Edit Existing Draft Deferral	<ul style="list-style-type: none"> Draft deferral can only be edited by the user assigned to the claim Can only edit a draft deferral from the history if it has not been finalized Users can switch a resolved or mitigated deferral from avoidable to unavoidable 	<ul style="list-style-type: none"> Deferral will be edited to have a new primary reason and/or deferral reason Deferral will have changed from avoidable to unavoidable or vice versa
VBMS User with Permission to Create Deferrals	Deferrals MFE	Delete a draft deferral	<ul style="list-style-type: none"> User has permission to create deferrals The deferral is still in draft stage and not finalized 	<ul style="list-style-type: none"> Draft deferral will be deleted from history No record of the deleted draft deferral will be kept
VBMS User with Permission to Create Deferrals	Deferrals MFE	Finalize a draft deferral	<ul style="list-style-type: none"> Claim is assigned to current user User has the PDD Permission for "Create Claim Deferral" All documents in the Veteran's eFolder with a deferral bookmark are associated to a deferral 	<ul style="list-style-type: none"> All pending deferrals are finalized at once Deferral(s) are changed from "draft" to "Pending" The Claim's lifecycle and status are updated Claim's work item is completed and the claim is removed from the assigned user's WQ Deferral is set to Active if the current claim lifecycle phase is the one in which the rework for the deferral is to be done
VBMS User with Permission to Mitigate Deferrals	Deferrals MFE	Mitigate a pending deferral	<ul style="list-style-type: none"> To manually resolve the user must have the permission "Manually Resolve Claim Deferral" <ul style="list-style-type: none"> User must also select a reason as to why the deferral is being resolved manually 	<ul style="list-style-type: none"> Deferral status is updated from "Pending" to "Resolved" If manually resolved the following is saved: <ul style="list-style-type: none"> timestamp of when the resolution occurred the name and RO of the user who resolved the deferral selected resolution reason free text comment VBMS-Core work item is automatically completed The claim status and the lifecycle status are automatically changed to the values that were assigned to the claim when the deferral was created A deferral cannot be created with the same reasons as a previously mitigated deferral User will receive an error when attempting to create another deferral for a previously mitigated deferral

Logical View

Components and Connectivity

When a deferral is needed, currently a VSR creates the Deferral in Core. RVSRs who usually use Ratings must "swivel-chair" to Core for this process. Core processes the form entered by a VSR/RVSR and handles the Deferral entity using business logic stored within a Deferrals java bean, and NWQ handles assignments to individual users based on pre-existing rules specific to Deferrals.

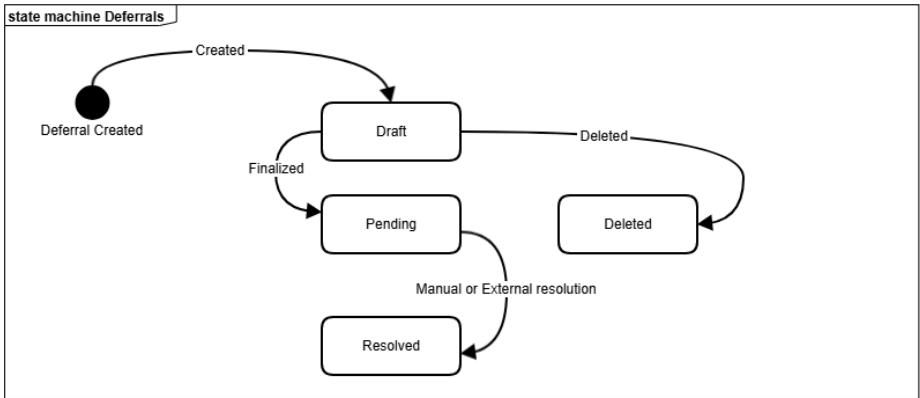
Systems Interface Description (SV-1)



Component References Table

Name	Description	Ports	Protocols
BSS #2985	An authentication service designated for operations-based applications. The SSOi service connects to VA AD to validate users' credentials from desktop session or Kerberos token and uses Federation to support external cloud providers while also utilizing HSPD-12 trust services to authenticate internal VA PIV users.	443	https
VBMS Home	VBMS Home serves as the central entry point for all microfrontends, providing a unified interface where users can access and interact with different MFEs including Deferrals. This component is intended to serve as the hub, or root, housing and organizing various MFEs.	443	https
Deferrals MFE UI	Deferrals is a MFE within VBMS Home that provides an interface for users to create, edit, delete, finalize, resolve, and mitigate deferrals for a specific claim. This MFE serves as the primary tool for interacting with deferral data, offering a streamlined, user-friendly experience for managing various aspects of deferrals.	443	https
Deferrals MFE API	Deferrals API is a Spring-based REST service designed to manage and interact with Deferrals data. It provides a set of endpoints for performing operations such as fetching, creating, updating, and deleting deferral records. The API ensures all data interactions between the Deferrals MFE and Core DB are efficient, secure, and consistent.	443	https
Core Database	The database contains all the critical information related to Deferrals. It stores data across different tables Deferrals relies on to operate, providing a reliable, centralized location for storing data as needed.	443	https

State Transitions (OV-6b)



In order to pass successfully from Ratings to Core (and potentially back to Ratings, then through Awards and potentially the DRA tool), certain data points must be passed and updated within a system of record. We recommend a freestanding Deferrals micro frontend to work through the existing user flow without losing claims.

You can think of a deferral essentially as a flag indicating additional information or action is needed to make an informed, final decision on a claim. Deferrals move through three statuses: draft, pending, and resolved.

Draft

- **Description:** This is the initial state of a created deferral where everything needed to "rework the claim" is identified and documented. A draft deferral can be edited or deleted by the assigned user of the claim, and supervisors are able to reassign that claim. If it isn't finalized (i.e. all necessary requirements haven't been outlined) before 10 business days, the draft deferral will expire and that action will be logged.
 - **Transition from Draft:** A deferral transitions from the draft to pending state, also referred to as the deferral status, upon finalizing. The claim status and lifecycle phase are automatically updated.

Pending

- **Description:** Pending is the state after a deferral has been finalized, but before it has been resolved. Think of it as a holding period, where everything previously identified for that final claim decision is *pending* completion. In this state, the deferral cannot be deleted or edited. A pending deferral may or may not be "inactive". Pending, inactive deferrals allow the associated claim to continue moving through the claim process until it can't—that is, the only thing left to complete is the deferral. On the other hand, if a deferral is active then the claim is prevented from moving forward until the deferral is resolved first (*Note: Whether a pending deferral is considered active or inactive depends on the "deferral rework phase", which is determined by the "claim lifecycle phase"*).
 - **Transition from Pending:** The state transitions from pending to resolved when a deferral has been resolved or mitigated by manual action or external system interactions.

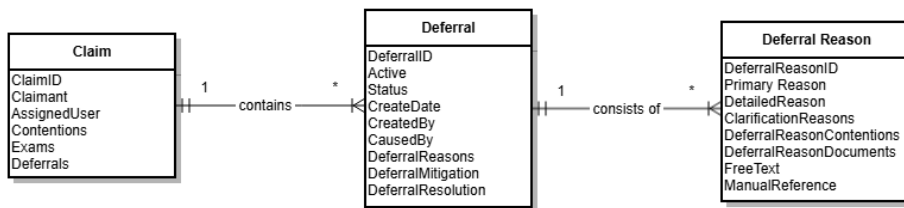
Resolved

- **Description:** A deferral in this state is considered closed by way of the resolve or mitigate action. When a deferral is marked as "resolved," it means that all the necessary actions or information required to complete the claim have been received and addressed by an authorized user or system outside of VBMS. Alternatively, a deferral can be "mitigated," which means it was closed for another reason, such as being deemed invalid by a supervisor.
 - **Transitions from Resolved:** No further transitions; this is the final state of a deferral.

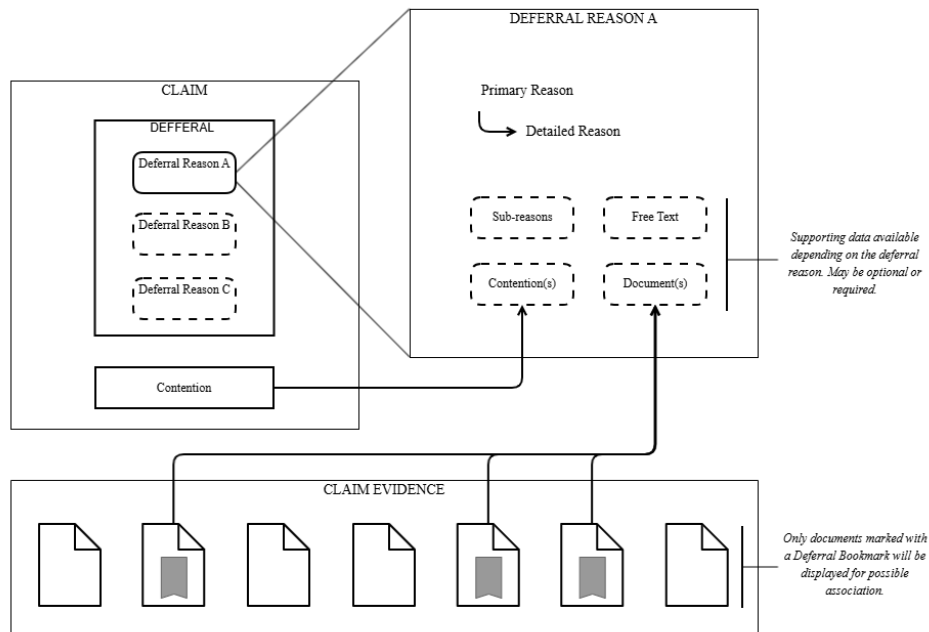
Data Models

Conceptual Data Viewpoint Model (DIV-1)

The relationship between claims and deferrals is as follows: each claim can contain many deferrals, and each of these deferrals can have multiple deferral reasons. Deferral reasons are made up of a primary reason and a detailed reason combination. Supporting documents and information (contentions, clarification reasons, free text, etc.) can be associated to each primary and detailed reason combination, depending on the combination of primary and deferral reason with the claim lifecycle phase.



To expand upon this please refer to the chart below showcasing a claim (large rectangle). This claim has a deferral associated to it (smaller box). Within this deferral we can see that there are multiple deferral reasons. Looking into 'Deferral Reason A' we can see there is a primary and detailed reason, which are required for a deferral. All of the supporting information for this primary and detailed reason combination are shown in the dotted smaller rectangles indicating they're not always mandatory based on the selected primary and detailed reason combination. When referring back to the deferral we see there is also a "Deferral Reason B" and "Deferral Reason C". These are other deferral reasons associated with the deferral, with each having their unique primary and detailed reason combination.



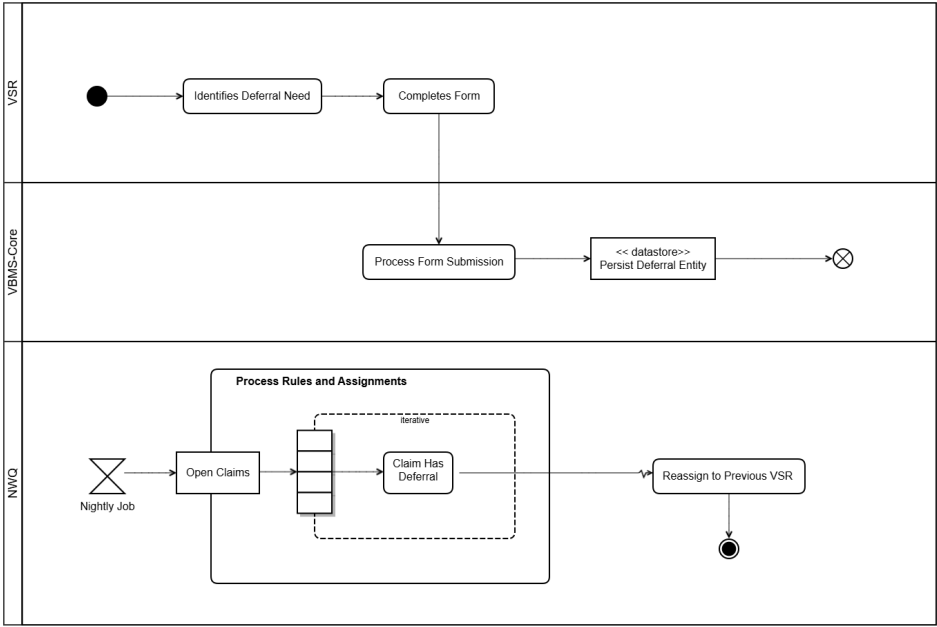
Claim				
Attribute Name	Data Type	Multiplicity	Constraints	Description
ClaimID	Number	1	Unique, Required	Unique identifier for each claim
Claimant	Claimant	1	Required	The recipient of the submitted claim
AssignedUser	VbmsUser	0..1	Is Authorized	The authorized user currently assigned to the claim
Contentions	Contention	0..*	Optional	A list of contentions associated with the claim
Exams	Exam	0..*	Optional	A list of exams associated with the claim
Deferrals	Deferral	0..*	Optional	A list of deferrals associated with the claim, indicating additional info or action(s) required to make informed, final decisions for claims

Deferral				
Attribute Name	Data Type	Multiplicity	Constraints	Description
DeferralID	Number	1	Unique, Required	Unique identifier for each deferral
Active	Yes/No	1	Required	Indicates whether a deferral is active or inactive
Status	DeferralStatus	1	Required	Current status of the deferral: draft, pending, resolved
CreateDate	Date	1	Required	Date a deferral is created which the expiration date of a draft deferral depends on (equal to 10 business days from the create date)
CreatedBy	VbmsUser	1	Required	The user who created the deferral
CausedBy	VbmsUser	0..1	Optional	The last user who updated the deferral
DeferralReasons	DeferralReason	1..*	Required	A list of unique primary and detailed reason combinations with other possible info indicating why a deferral is needed for the claim
DeferralMitigation	Mitigation	0..1	Optional	Indicates the deferral was closed, including details on who performed the mitigation and the reason why (e.g. invalidity)
DeferralResolution	Resolution	0..1	Optional	Indicates the deferral requirements have been satisfied, marking it as closed, with details on who or what (i.e., external system) closed it and reason for resolution

The primary focus of the current effort is creation of a Deferral when the Claim is in 'Ready for Decision' or 'Secondary Ready for Decision' Claim Status. This requires enhancement of the Deferral experience within VBMS-R. However, the Solution could optionally unify the creation of Deferrals across VBMS Core and VBMS-R. The immediate need is to simplify the workflow of RVSRs who need to defer one or more Issues on a Claim, while moving others through the Rating process.

Note the repeated context switching between applications in the Requirements Team's current state Business Process Model. Currently, users log into individual systems and input data that is maintained locally. Deferral data that affects the Claim Status and NWQ is only accessible within VBMS Core and data that impacts the Rating of Issues is accessible only within VBMS-R. In future, users must have access to the Deferrals user interface from either application, which will necessitate passing data from VBMS-R to the system of record for Deferrals (currently VBMS Core).

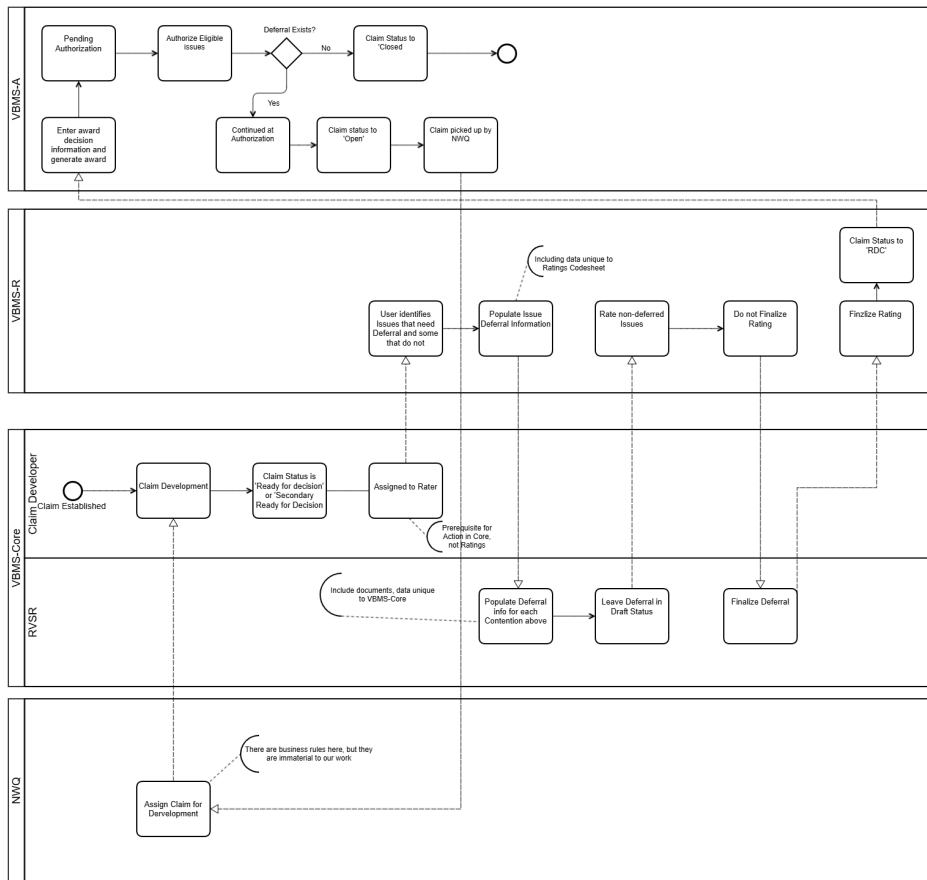
The existing process model from the Deferrals Overview provides an excellent simplified view of the current process in Core:



Current State Process Model

Main Business Process Model

The Current State Process Model indicating "swivel-chair" operation by RVSR, working in both VBMS-Ratings and VBMS-Core, was created by the Requirements Team. This use case is specifically needed to manage deferred contention(s) on a claim that has both contentions ready for rating and contentions that need additional information/rework.



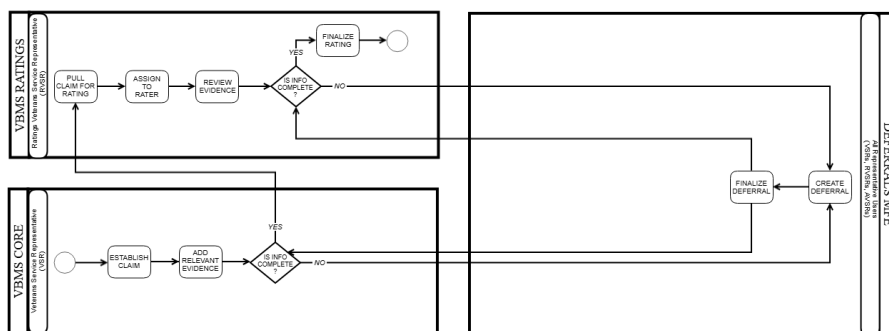
Future State Process Model

Proposed Main Business Process Model (OV-6d)

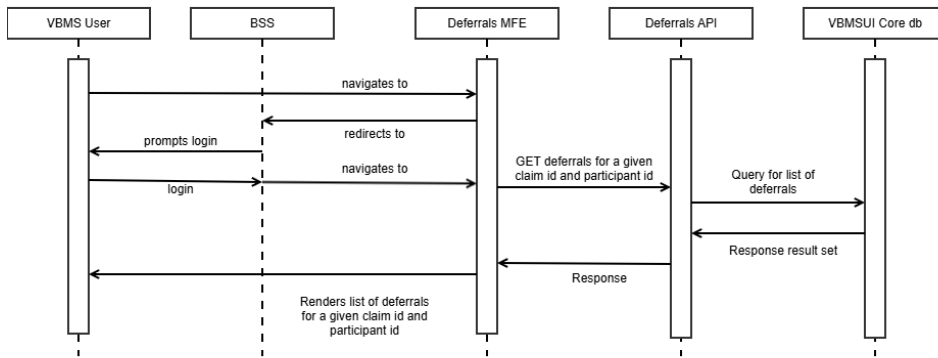
This process model represents a simple, high level overview of the proposed future Deferrals state, demonstrating the flow between VBMS Core, VBMS Ratings, and the standalone Deferrals MFE.

For simplicity this diagram is only showing the Deferral process from the perspective of Core and Ratings, this same flow would occur with Awards as well.

Please note this diagram is subject to evolve.



Process Sequence - System Event-Trace Description (SV-10c)



Alternate Flows & Exceptions

- Preventing a user from submitting with incomplete information
- Deferrals API is unavailable
- User cancels deferral submission
- User creates a deferral with the same reason on a draft deferral
- User creates a deferral with same information as a mitigated deferral
- Unable to save deferral reason. The deferral has moved beyond DRAFT status
- A deferral cannot be completed: No work item exists for this rework phase
- User creates a Exam + primary Reason "Insufficient Exam" or "Clarification Needed" deferral without a DBQ as supporting documentation
- User tries to complete a work item while there is a pending deferral for a reason other than "Ratings"
- User tries to finalize draft deferral but there are unassociated documents with deferrals bookmarks
- Deferral is automatically deleted when associated contentions are deleted
- Deferral is automatically deleted when associated documents are deleted

Implementation View

Primary System Technologies (SV-9) (VASI Technology Components)

For standard tools see [VBMS SV-9 Master Table](#).

Software Type	Software Vendor	Software Name	Software Version	TRM Link	TRM ID
Development Tool	Meta	React.js	18.2.0	Link	10254
Development Tool	Meta	React Developer Tools	6.1.1	Link	14251
Development Tool	N/A - Open Source	single-spa	6.0.1	N/A	N/A
Development Tool	N/A - Open Source	single-spa Inspector	0.5.0	N/A	N/A
Development Tool	VA	BIP Component Library (BCL)	5.3.0	N/A	N/A
Development Tool	Oracle	Java	8	Link	22
Development Tool	Broadcom	Spring Boot	2.1.4	Link	8508
Development Tool	Microsoft	Visual Studio (VS) Code	1.98.2	Link	8371
Development Tool	Bruno	Bruno	1.28.0	Link	16783
Development Tool	GitHub	GitHub Enterprise	-	Link	9533
Development Tool	GitHub	CodeQL	2.19.2	N/A	N/A
Middleware: System Management Tool	Kubernetes	Kubernetes	Client: 1.30.0 Kustomize: 5.0.4-0.20230601165947-6ce0bf390ce3 Server: 1.29.12-eks-2d5f260	Link	11815
Development Tool	Argo Project	Argo Continuous Deployment (CD)	-	Link	16051

Development Tool	Jenkins	Jenkins Continuous Integration Server	2.426.3	Link	6397
Development Tool	SonaType	SonaType Nexus Repository	3.61.0	Link	6411
Middleware: System Management Tool	HashiCorp	Vault	1.18.0	Link	11551
Middleware: System Management Tool	HashiCorp	Consul	1.13.9	Link	9895

Deployment View

Environment Mapping

Cluster	Environment	PII?	Link	Description	Managed By
DEV	DEV	No	TBD	Used by development teams for debugging and testing initial integration of the software and may be unstable.	Development
	INT	No	TBD	An integration testing area where developers or users can experiment with new features, configurations, and data without affecting other environments or systems.	Development
	TEST	No	TBD	Used by development teams to validate software against acceptance criteria, simulating realistic use cases for QAs to execute tests with goals to identify potential bugs and ensure software functions as expected across different conditions. It should be reasonably stable.	Development
STAGE	UAT (User Acceptance Testing)	No	TBD	Where end-users or stakeholders test the software in conditions that closely mimic the production environment, ensuring user needs and business requirements are met before going live.	DevOps
	IVV (SQA)	No	TBD	Used by the QA team to independently verify and validate software.	DevOps
	PDT (Production Development Test)	No	TBD	Snapshot of production codebase for development teams to support triage efforts and used to create patches.	DevOps
	PAT (Partner Acceptance Testing)	No	TBD	Snapshot of production codebase for development teams to support triage efforts and used to create patches.	DevOps
	DEMO	No	TBD	Used for showcasing software to clients, stakeholders, and internal teams often pre-populated with sample data used to demonstrate key features and functionality, intended to provide a smooth, polished experience without risks.	DevOps
PREPROD	BAT	No	TBD	Used for testing by the education group (EDU) in support of the Digital GI Bill.	DevOps
	COLA (Cost of Living Adjustments)	Yes	TBD	Used by the VA to test adjustments to Veteran's benefits.	DevOps
	PERF	Yes	TBD	Used for verifying the software functions properly at scale.	DevOps
	PREPROD (Pre-Production)	Yes	TBD	Used for verifying the software deploys and operates with live data.	DevOps
	PRODTEST	Yes	TBD	A daily copy of production where ProdOps team can triage production-level software and data.	DevOps
PROD	PROD (Production)	Yes	TBD	The most critical, live, and fully functional environment where software is utilized by end-users (VA employees across ROs). Any issues here can directly impact clients, with performance, security, and availability being top priorities.	DevOps

System Configuration

Follows BIP Platform standards.

System Monitoring and Metrics

Follows BIP Platform standards.

System Logging and Auditing

Follows BIP Platform standards.

Security Strategy

Follows BIP Platform standards.