**PROJECT POPLIN**

**MEDICAID REFERENCE ARCHITECTURE**

**CARE/CASE SERVICE DEFINITIONS**

**MITA Governance Board**

**Poplin Working Group**

**State of Vermont**

**2018.02.15**

# POPLIN Project Background

The Centers for Medicare and Medicaid Services (CMS) seeks to develop a robust module marketplace for Medicaid to promote competition of solutions, reduce risk, streamline certification, and better serve the needs of individual states’ Medicaid Enterprise Systems (MES) through standards and conditions for its funding of state implementations. Such conditions include application of modular design, development, and deployment methodologies and technologies for state-specific MES components and reuse of those components when possible.

To help states and vendors prepare for and use a module marketplace for Medicaid, CMS has sponsored efforts with MITRE Corporation through its Federally Funded Research and Development Center’s (FFRDC) CMS Alliance to Modernize Healthcare (CAMH) to develop a reference architecture (Project Poplin) than can extend the Medicaid Information Technology Architecture (MITA) technical architecture.

Poplin seeks to provide standard definitions for application services that enable MES implementations across a diversity of state requirements so that vendors can develop robust technologies with broad application. This Service Definition Template defines the components required in a service definition document.

# Modularity

There are many level of modularity that the MES’s could be abstracted to. Each module fulfills a business need of the overall MES. These are the initial 10 Business Areas as defined by MITA are:

* Business Relationship Management
* Care Management
* Contractor Management
* Eligibility and Enrollment Management
* Financial Management
* Member Management
* Operations Management
* Performance Management
* Plan Management
* Provider Management

Many business processes cross over the above abstractions. This can make it difficult when attributing a process to a predefined business area.

# Introduction

This document breaks down Care Management business processes with the goal of identifying the dependencies between all systems in a modern and modular Medicaid Enterprise.

The core of the Care Management business area is coordinated care management. This is accomplished generally through a Case Management system. A case management system enables an organization to perform many more activities than just care management.

We will be taking a focus of Case Management and how it fits into the MITA architecture, processes it enables, and the standard interfaces that need to be available for a more connected enterprise.

# Communications

Before we begin identifying minimum functionality of services for MES implementations we want to identify what means should exist to allow systems to interact.

There are basically three ways systems can communicate with each other

1. Batch Interface  
   Best suited to exchange reference data that does not change frequently. One example would be a list of ICD-10 codes. This method is not preferred.
2. Real-Time Interface  
   This type of interface is necessary when requesting time sensitive information or the most accurate data in an ever-changing environment. One example of this would be a request to see if a client is currently enrolled, or eligible, to receive a service.
3. Event Driven Interface  
   This is a key component of any enterprise. When data is changing in multiple systems there is a need for notifying others of changes that could affect their activities. One example would be if a provider was no longer eligible to perform services we would want the Case system to be aware of this as soon as possible. In this type of environment, a message would be generated, and the Case solution would see the message and process it as needed.

Solutions can expose functionality in multiple ways to handle these types of communications. Publishing Application Programming Interfaces (APIs), Service Oriented Architecture (SOA) Services, and a Messaging Service are a few of the desirable ways. These methods are really a layer of abstraction away from the underlying complexity that allows multiple systems to communicate in a standard way. This also enables the realization of a more modular enterprise.

# System Approach

We will be taking the approach of using an integration layer to enable real-time and event driven communication between systems. Two key components of integration layer are an API Gateway and a Messaging system.

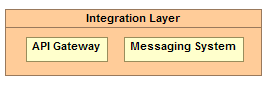


Figure 1 - Integration Layer

These are the basic systems in a modern enterprise needed to enable integration/communications between systems.

The API Gateway is used to expose API’s as defined in this document that are needed to support real-time access to information and processes. The API’s define how systems will communicate, the format of the request, what the expected response will be, as well as the protocol used to access the API.

The Messaging System is another component that allows for the real-time processing of information. Messaging enables event processing. A system could for example take action based on an event, such as the approval of an application, and triggering additional processes. The same detail as described for API’s is necessary for the use of standard messaging.

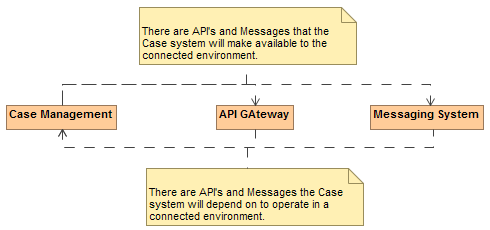


Figure 2 - API and Message Connectivity

Although we will mention API’s and Messaging that the case system depends on we will only be defining the details for those that the case system provides. Any system that adheres to the specifications to follow will be able to use the APIs and Standard Messaging to communicate with the case system.

# Assumptions

## System Assumptions

The following are assumptions being made as to the modularity of the systems involved and where responsibilities lie for the maintenance of information. The systems expected by the Case Management system are:

1. Master Person Index (MPI)  
   This system acts as a phone book. It contains only the basic information needed to identify a person across all systems which enables analysis across the whole enterprise. The information should be minimal to avoid security issues. No medical information is captured here. Every Client and Provider would have their basic Person information here. No other system stores information in the MPI. Adding a person requires specific governance to maintain integrity. A system may request a person and if the person does not exist a temporary entry may be created which will be followed up on by the MPI Data Steward.
2. Master Client Index (MCI)  
   A person listed in the master client index is receiving, or has received, services from the State. This system maintains the client data, health data including care plans, and may also include client specific data related to other human service programs. Master Client links to a person in the Master Person Index.
3. Human Service Program Management  
   This system is the focal point of plan information management. The services offered, the rates allowed, the benefits provided and other information that governs the program is maintained here.
4. Provider Management  
   This system is responsible for the list of providers, what roles that can fill, and what services they can perform.
5. Consent Management  
   This system is responsible for maintaining the entry and analysis of consent forms and data indicating the client’s preferences when it comes to making data available to users of various systems.

## API Assumptions

API’s will be defined as RESTful. The use of the PATCH operation will not be included in the definitions below. POST, GET, PUT, and DELETE operations will be defined as appropriate.

# System Dependencies

When you look at Case Management processes you find that there are many dependencies required within the enterprise. We will break down these individual dependencies where they apply and outline the necessary services required to allow these components to communicate.

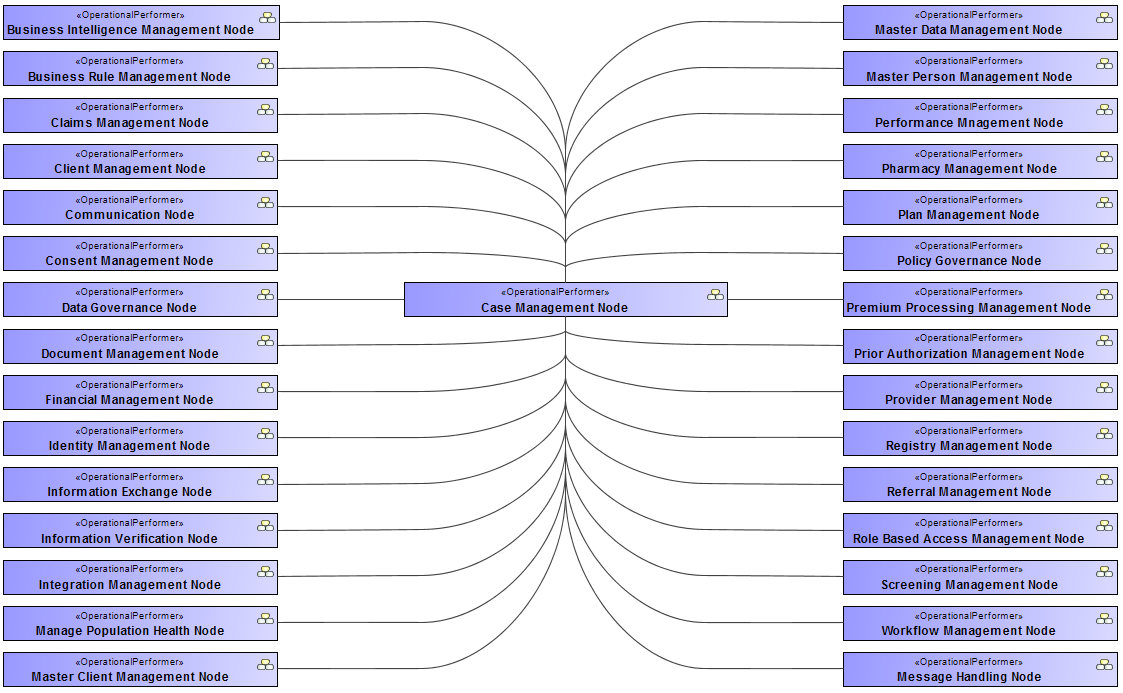


Figure 3 – High Level Care and Case Dependencies

From the image above its easy to see how interconnected the systems that make up Human Services can be. It is essential that the proper sub-systems are in place to make the migration to a common modular future state possible, progress in a logical fashion, and reusable.

The remaining documentation describes the communications between systems and how the this is accomplished using the systems described in the Systems Approach section.

# Case Activities

As we focus more specifically on case management we find many standards that identify case processes. A generally accepted list of high-level case functionality includes:

| **High Level Case Activities Table** |
| --- |

| **Name** | **Documentation** |
| --- | --- |
| Initiate | This covers the processing of incoming referrals and other information that are the basis for case creation. |
| Engage | Successful engagement is the basis of effective case management. Engagement establishes the relationship and sets the ground rules which are enhanced and reviewed throughout the phases of case management. Engagement begins prior to the initial contact between the case manager and the person requiring services, their family and/or career. |
| Assess | Assessment is a dynamic and ongoing process where information is gathered from a range of sources about a person, including their life situation, and formal and informal supports. The range of information is then considered in the context of information and advice from the person with a disability, their family and/or careers and informal supports. Information may be obtained from a range of sources, including written reports or records, verbal reports and observations and impressions of the case manager. |
| Plan | The goals, needs and wishes of the client form the basis of a plan. A plan is a map of actions that documents the issues, methods, responsibilities and timeframes needed to meet the identified goals. |
| Implement | Implementation is the process of putting into action the plan developed by the case manager, together with the person, their family and other supporting team members. The aim of implementation is to help the person achieve their goals and desired outcomes, identified in the assessment and planning phases. |
| Monitor | Monitoring is an active and ongoing process where aspects of the planning and implementation phases are reviewed. It identifies the effectiveness and relevance of planned goals, focusing on the timeliness and success of strategies being used to achieve these goals. Additionally, it provides the opportunity to adjust the plan to address any unanticipated problems. |
| Review | Case management practice involves regular formal and informal review processes. The review phase is important to help ensure that outcomes for people are relevant to their needs and include a focus on community inclusion and participation. The review process should be driven by, and inclusive of, the client and how they feel the process is working. |
| Transition | Transition can occur for many reasons and involves creating a plan with the clients and other supports to insure they receive the care needed. This could happen when a client has reached the goals applicable to the services being rendered or is no longer eligible. In these cases, clients are generally referred to other resources for continued assistance. |
| Close | Cessation of involvement by a case manager or the closure of a case may be influenced by many factors. These factors may relate to organizational or agency requirements in which a limited time frame is identified for the case manager involvement with any one client. This may raise significant concerns or questions as to the appropriateness of terminating contact. The decision as to whether to cease involvement or close the relationship can be influenced by a number of factors. |

Adding to the case activities we also have two outliers that cannot be overlooked.

| **Supporting High Level Case Activities Table** |
| --- |

| **Name** | **Documentation** |
| --- | --- |
| Analyze | Analyze data to determine where evidence shows changes are needed to improve outcomes, to detect fraud, and to verify utilization of services. |
| Support | There are many supporting functions necessary for enabling the care and case functionality. Managing master data, reference data, security, auditing, and historical data are just a few areas of concern. |

The sections below will explore each of these areas to identify the following:

1. Identify the process(s)
2. Dependencies on other systems / modules
3. Identify the Activities involved at the dependency boundary
4. Identify the objects necessary to support these activities
5. Identify the APIs the case module depends on being available
6. Identify the APIs the case module will make available
7. Identify Messaging the case module can act on
8. Identify Messaging the case will generate
9. Define the data objects needed to satisfy the API and Messaging defined
10. Define the resources needed to implement the activities, APIs, and messaging

Also note that there are additional activities that are required for the management and maintenance of the Case System that are listed in the Supporting Activities sub-section.

The next two figures we have traced the high-level case processes to the applicable MITA Business Processes. The results are shown in the following two figures.

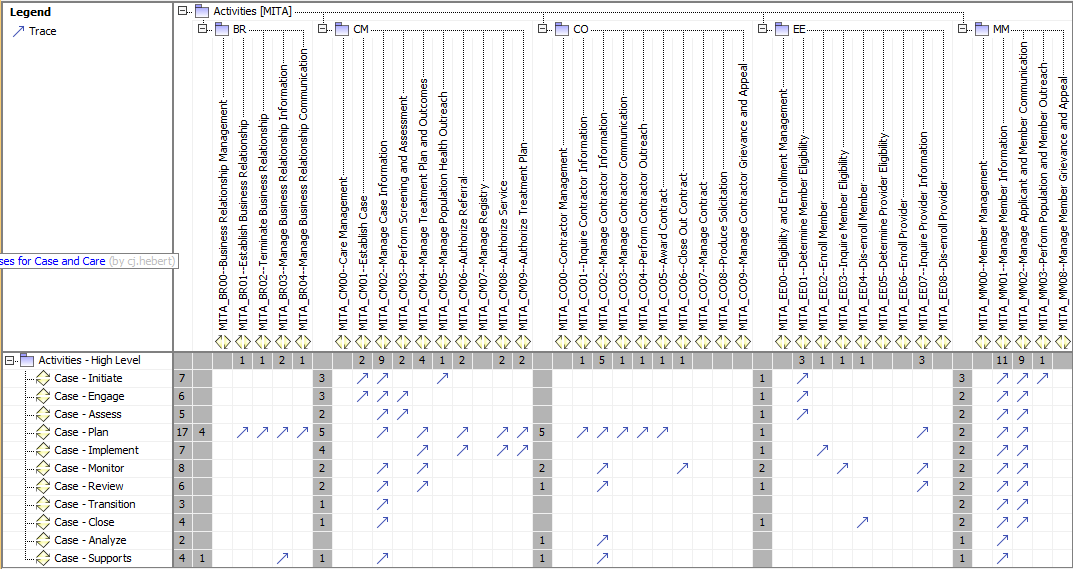


Figure 4 - Case to MITA Business Processes A

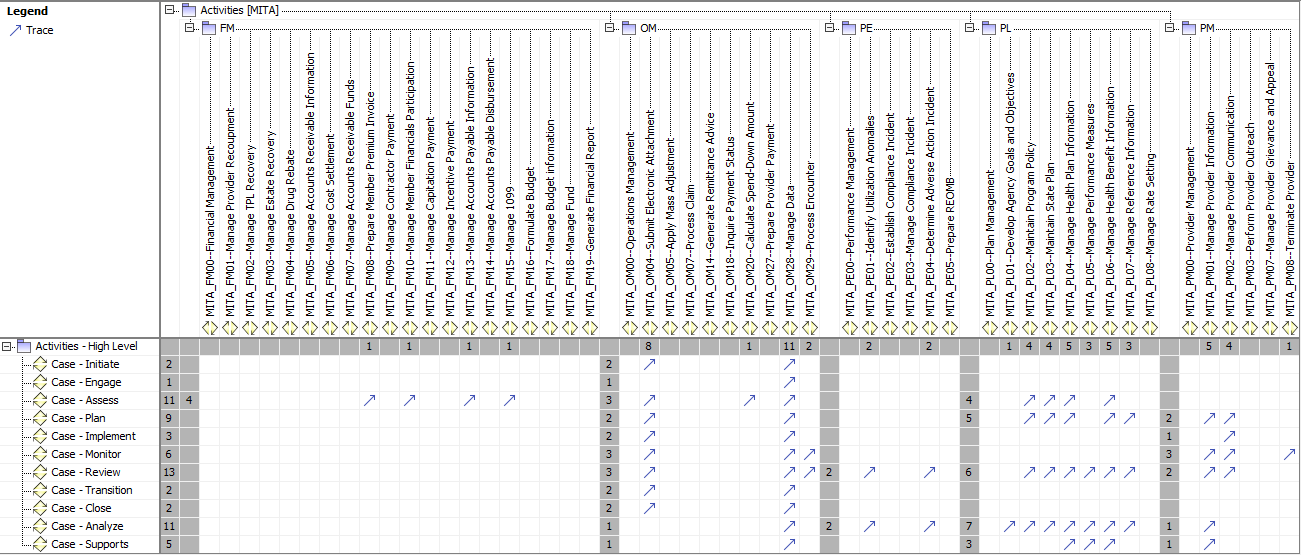


Figure 5 - Case to MITA Business Processes B

# Services

As shown in the dependencies described above, there is a reliance on many systems to retrieve information from and many others that rely on the information produced during case processing.

This section outlines the processes in more detail along with the services needed to support them.

## Case Services

### Post Case

Case will be responsible for exposing a standard API to allow other systems to create a case.

For the purposes of expediting processes, a request to create a case by access the Post Case API should create a case if there is enough information available to know who the person involved is and what the issue is the case is needed to handle.

The process of case initiation will rely on Get Person and Get Client at a minimum.

As shown below, Post Case is being used by the Client Application Management.

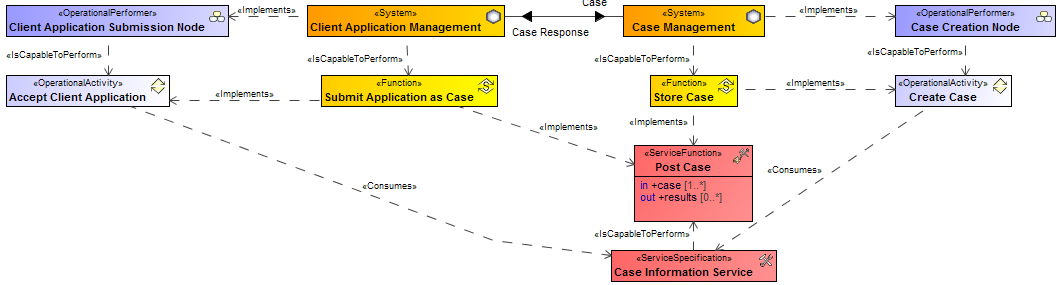


Figure 6 - Post Case

Many systems may access the Post Case API and some of the expected ones are shown below:

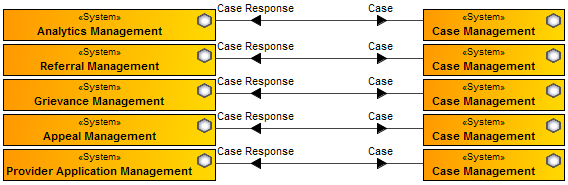


Figure 7 - Other Systems Accessing Case Creation

### Get Case Exists

Case will be responsible for exposing a standard API to allow other systems to verify if a case of a specified type exists for a known person.

A link to the specific case will be returned for a given person and case type. The return will show if there is a past case or a active case.

As shown below, Get Case Exists is being used by Client Application Management to check if a case exists for a client application:

### Get Case Status

Case will be responsible for exposing a standard API to allow other systems to verify the status of a case.

The status may be returned given a person and a case type, or by a case identifier.

As shown below, Get Case Status is being used by Client Application Management to check the status of an application:

## Security Services

The following security services are necessary to enable controlled access to systems and data.

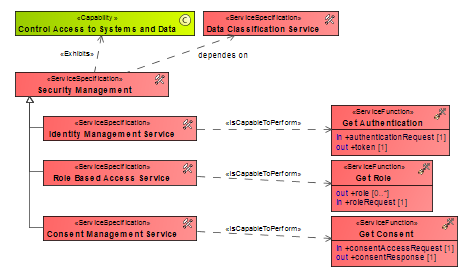


Figure 8 - Security Services

### Get Authentication

The Identity Management Module will be responsible for Authentication. All systems within the enterprise will rely on the Identity Management module for making available the Get Authentication API.

This service will be defined in further detail in the Identity Management Service Definitions document.

For case purposes, we should be able to get a token from the service by supplying login information. This token should be a required argument for all API calls within the MES Enterprise. This is not used for external B2B communications as the servers that enable external communications rely on other forms of authentication.

Pictured here a conceptual representation of accessing the Identity Management API to retrieve an identity token.

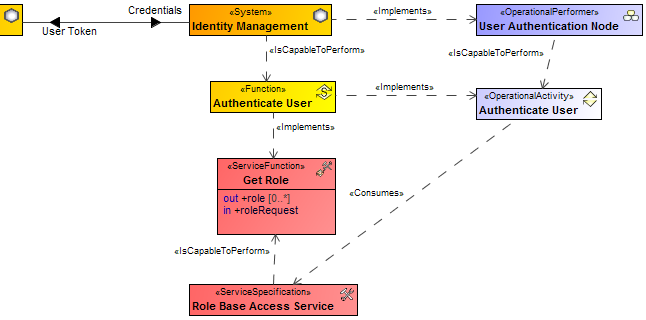


Figure 9 - Get Authentication

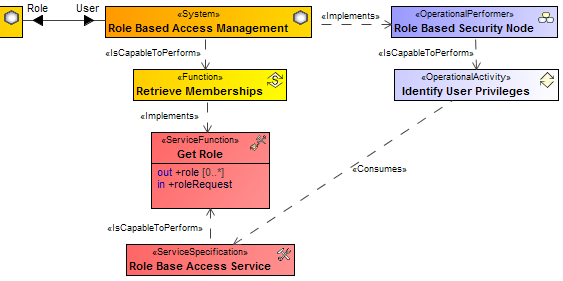
### Get Roles

The Access Management Module will be responsible for determining the role of the user which determines what systems and functions they can access. All systems within the enterprise will rely on the Access Management module for making available the Get Role API.

This service will be defined in further detail in the Access Management Service Definitions document.

For case purposes, we should be able to pass, for example, the user identity and department and receive a list of valid roles for the user.

Pictured here is conceptual representation of accessing the Access Management API to retrieve the user roles.



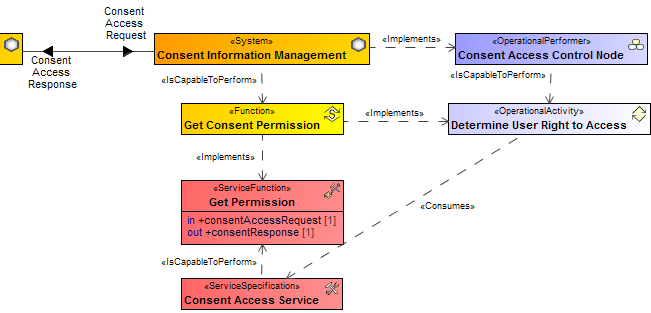
### Get Permission (Consent)

The Consent Management Module will be responsible for determining the user’s ability to access sensitive client data. All systems within the healthcare enterprise will rely on the Consent Management module for making available the Get Permission API.

This service will be defined in further detail in the Consent Management Service Definitions document.

For case purposes, we should be able to pass, for example, the user identity, roles, and the type of attributes of the data or system to be accessed and receive a response indicating whether access is allowed based on an individual’s consent preferences.

Pictured here is conceptual representation of accessing the Access Management API to retrieve the user roles.



## Other Supporting Services Required

The services listed in this section are services that the Case System will be dependent on. It may be possible that the case system would implement some of these functions itself but in a fully modular approach these services would be provided by other systems. The services list below are mentioned here for the purpose of identifying the necessary services to support a modular case system.

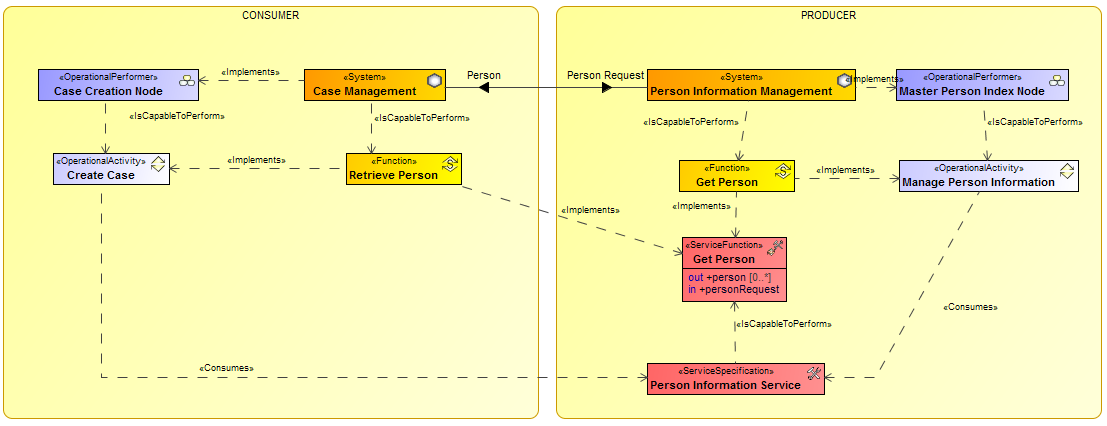
### Get Person

Case will rely on a Master Person Index. This module will be responsible for making available the service to get the basic information for a person.

This service will be defined in further detail in the Master Person Service Definitions document.

For case purposes, we should be able to get a person by a unique identifier, by information supplied such as name, address, phone, or by various other identification types.

Case would also expect if a person did not exist it should be created by the Master Person Index and flagged for review, so the case processes can carry on without interruption.



The service to Get Person should return:

1. A single person as requested
2. A list of possible matching persons as requested

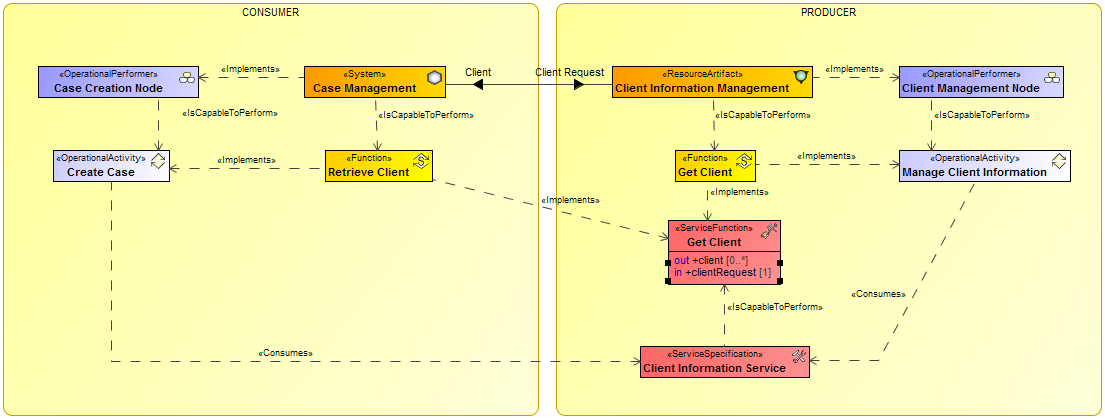
### Get Client

Case will rely on a Master Client Index. This module will be responsible for making available the services to get the basic information for a client.

This service will be defined in further detail in the Master Client Service Definitions document.

For case purposes, we should be able to get a client by a unique identifier, by information supplied such as name, address, phone, or by various other identification types.

Case would also expect if a client did not exist it should be created by the Master Client Index and flagged for review, so the case processes can carry on without interruption.



The service to Get Client should return:

1. A single client as requested
2. A list of possible matching clients as requested

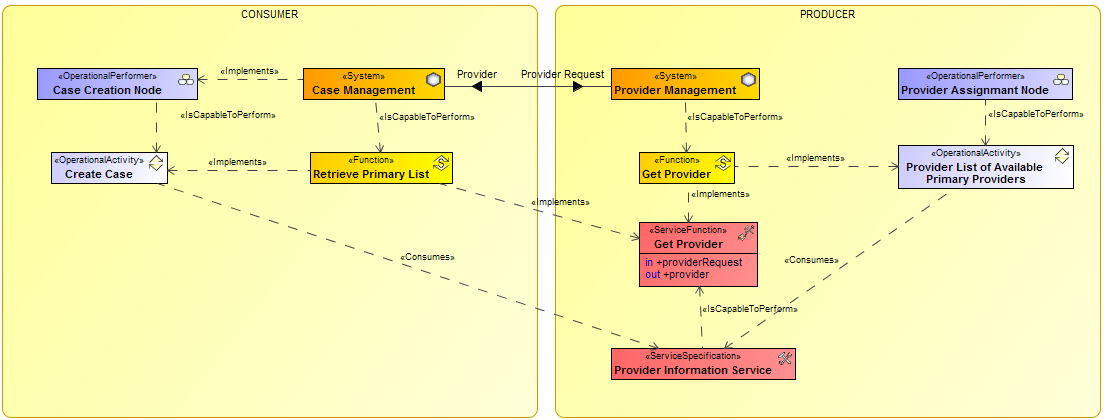
### Get Provider

Case will rely on a Master Provider Index. This module will be responsible for making available the services to get the basic information for a provider.

This service will be defined in further detail in the Master Provider Service Definition document.

For case purposes, we should be able to get a provider by a unique identifier, by information supplied such as name, address, phone, by role, by specialty, by geographical area, or by various other identification types.

There are no expectations on the Get service other than the stated above.



The service to Get Provider should return:

1. A single provider as requested
2. A list of possible matching providers as requested

### POST, GET, PUT, DELETE Plan

### POST, GET, PUT, DELETE Household

# Messaging

## Person Messages

Messaging would be expected from a Person system. Specifically, if a person was merged with another person record, the person address change, or some other change that could affect the possible delivery of services to a person.

## Client Messages

Messaging would be expected from the Client system. Specifically, if a client moved or some action was taken to indicate they may no longer qualify for services.

## Provider Messages

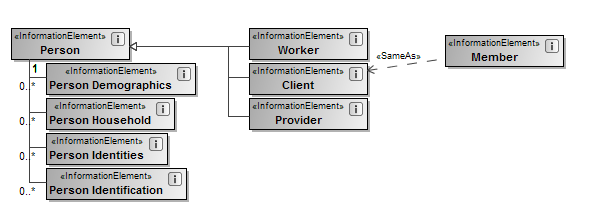
Messaging would be expected from a Provider system. Specifically, if a provider was terminated, specialty removed, or provider facility moved or was closed.

# Additional Object Information

## Person

The Person object is key for all systems. It holds the basic information of a “Person” and every other representation of a person (such as a client) starts with the basic person element.

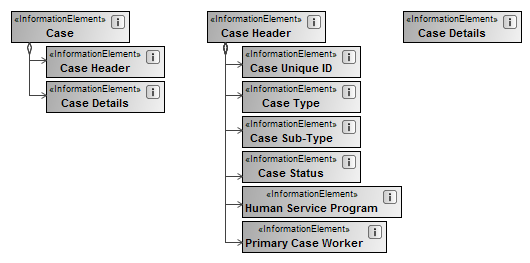
As shown below worker, client, provider are specializations of person. That is, they are a “person” with some additional information.



## Client

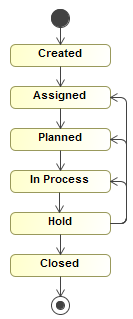
## Case

The basic structure



## Object States

## Case States

A case goes through many states during its life as shown here:  


# Appendix A - MITA Business Process Reference

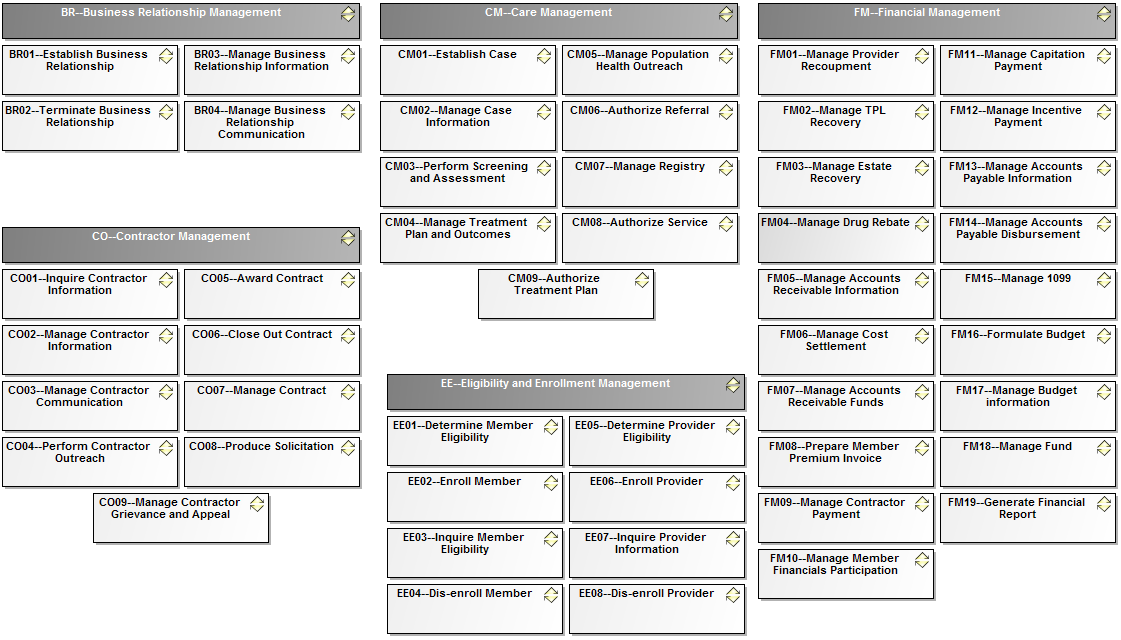


Figure 10 - MITA Business Processes A

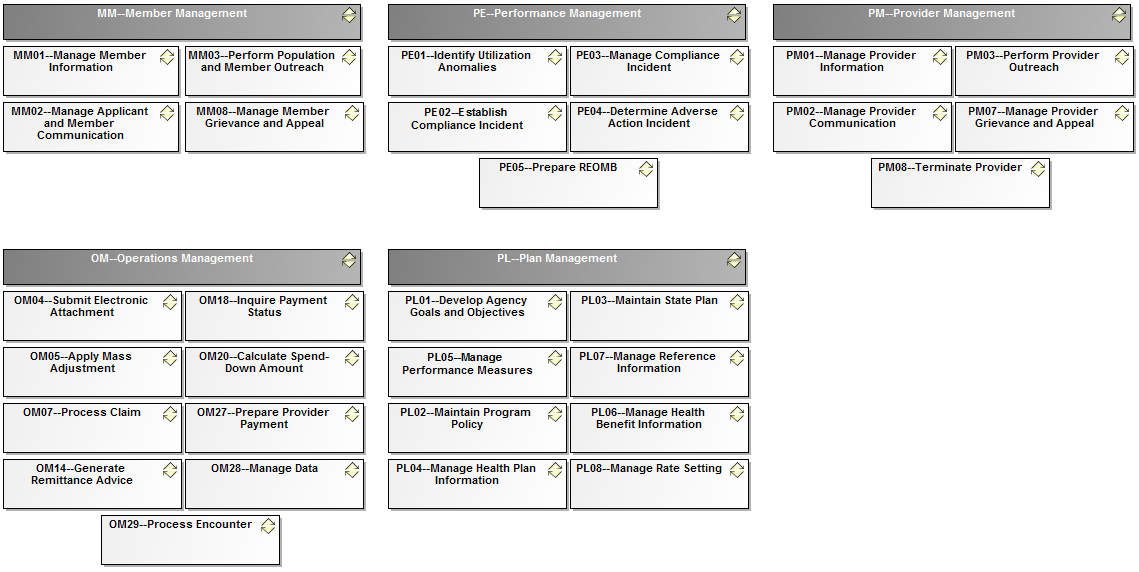


Figure 11 - MITA Business Processes B

# Appendix B - MITA Care Management Business Process Dependencies

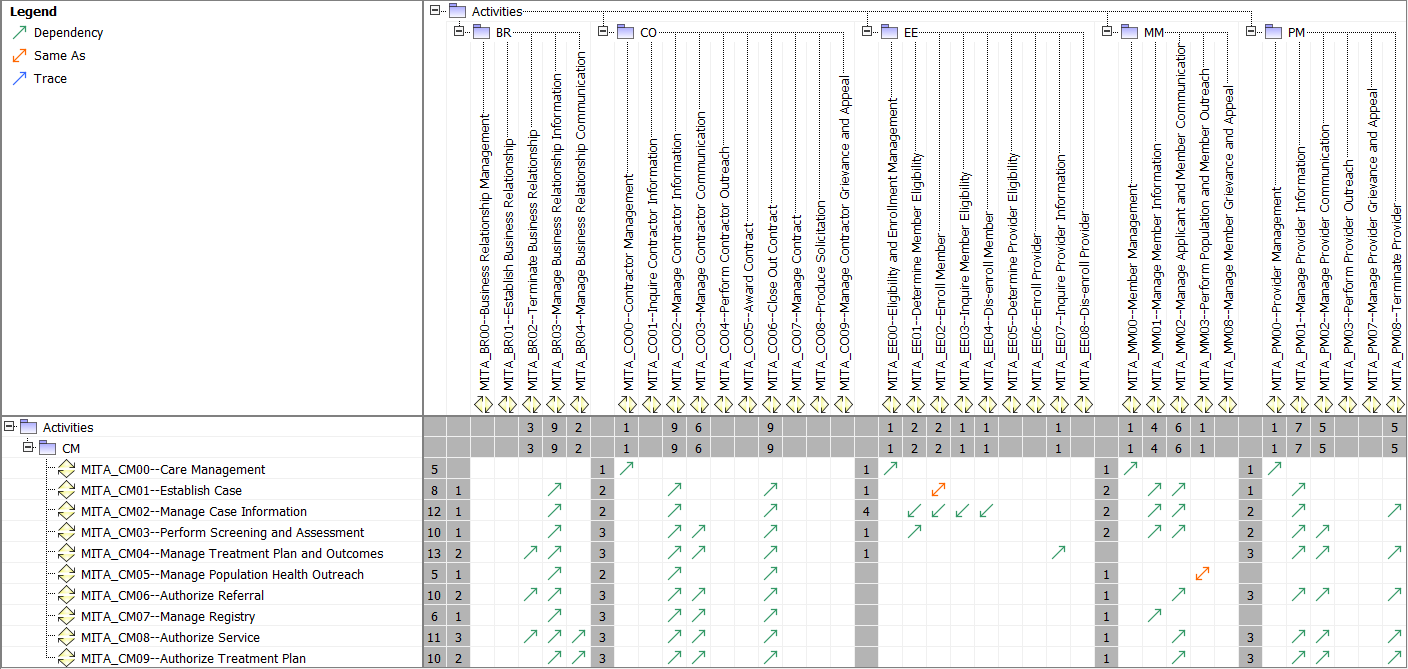


Figure 12 - MITA CM Dependencies A

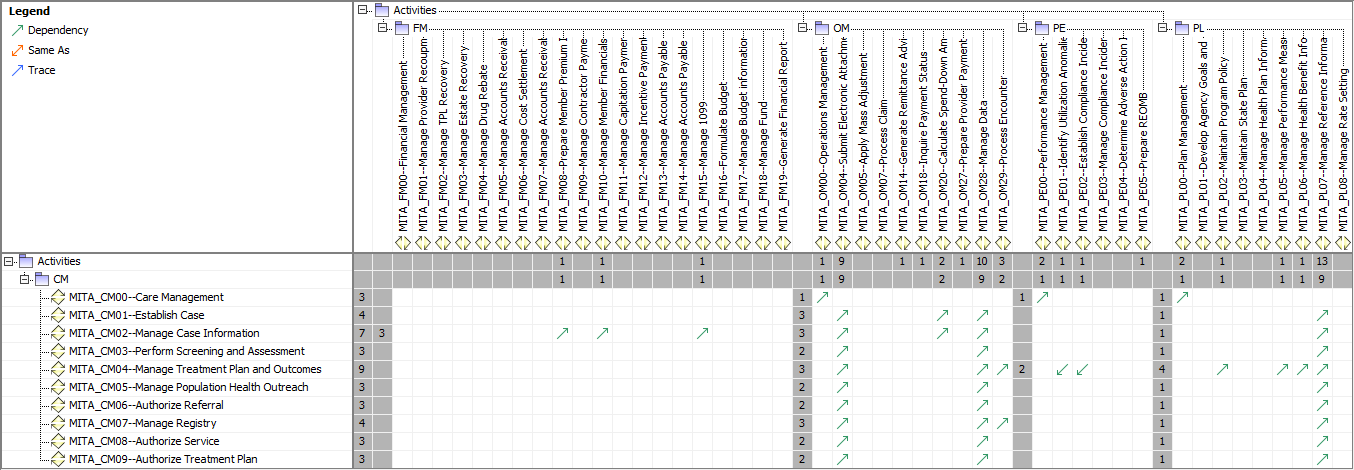


Figure 13 - MITA CM Dependencies B