MCCF

Technical Manual

**TAS 01.00**



December 2017

Department of Veterans Affairs

Office of Information and Technology (OI&T)

Revision History

**NOTE:** The revision history cycle begins once changes or enhancements are requested after the document has been baselined.

| Date | Revision | Description | Author |
| --- | --- | --- | --- |
| 10/19/2017 | 1.0 | Initial document | Patrick Whalen |
| 11/7/2017 |  | Added interface information | Patrick Whalen |
| 12/19/2017 |  | Removed build number from doc | Lee Benhart |
| 12/20/2017 |  | Added security data transmission | Patrick Whalen |

A Technical Manual is a required end-user document for all OI&T software releases. The intended audience for this document is local IT support, management, and development personnel for nationally released software. It provides sufficient technical information about the software for developers and technical personnel to operate and maintain the software with only minimal assistance from Product Support staff.

**Table of Contents**

[1. Introduction 1](#_Toc499117639)

[1.1. Purpose 1](#_Toc499117640)

[1.2. System Overview 1](#_Toc499117641)

[1.3. Document Orientation 2](#_Toc499117642)

[2. Implementation and Maintenance 3](#_Toc499117643)

[2.1. System Requirements 3](#_Toc499117644)

[2.2. System Setup and Configuration 3](#_Toc499117645)

[3. Files 4](#_Toc499117646)

[4. Routines 5](#_Toc499117647)

[5. Exported Options 5](#_Toc499117648)

[6. Mail Groups, Alerts, and Bulletins 5](#_Toc499117649)

[6.1. Build Notifications 5](#_Toc499117650)

[6.2. Application Notifications 5](#_Toc499117651)

[7. Public Interfaces 5](#_Toc499117652)

[7.1. Integration Control Registrations 6](#_Toc499117653)

[7.2. Application Programming Interfaces 6](#_Toc499117654)

[7.3. Remote Procedure Calls 6](#_Toc499117655)

[7.4. HL7 Messaging 7](#_Toc499117656)

[7.5. Web Services 9](#_Toc499117657)

[8. Standards and Conventions Exemptions 9](#_Toc499117658)

[8.1. Internal Relationships 9](#_Toc499117659)

[8.2. Software-wide Variables 9](#_Toc499117660)

[9. Security 9](#_Toc499117661)

[9.1. Security Menus and Options 9](#_Toc499117662)

[9.2. Security Keys and Roles 10](#_Toc499117663)

[9.3. File Security 10](#_Toc499117664)

[9.4. Electronic Signatures 10](#_Toc499117665)

[9.5. Secure Data Transmission 10](#_Toc499117666)

[10. Archiving 10](#_Toc499117667)

[11. Non-Standard Cross-References 10](#_Toc499117668)

[12. Troubleshooting 10](#_Toc499117669)

[12.1. Special Instructions for Error Correction 10](#_Toc499117670)

[12.2. National Service Desk and Organizational Contacts 10](#_Toc499117671)

[13. Acronyms and Abbreviations 10](#_Toc499117672)

**Table of Figures**

[Figure 1 2](#_Toc499117673)

[Figure 2 6](#_Toc499117674)

**Table of Tables**

[Table 1: MCCF Base Code Directory Structure 4](#_Toc499117675)

# Introduction

The Medical Care Collection Fund System (MCCF) automates and modernizes the five VA work streams: eBilling, ePharmacy, eInsurance, ePayments, and eAdmin.

## Purpose

This Technical Manual is designed to provide the developers and technical personnel with information necessary to install, maintain, and troubleshoot MCCF.

## System Overview

**WORK STREAM MODULES**

eBilling

* Claims Tracking, Encounter Form Utilities, and Insurance Data Capture.

ePharmacy

* Provides a method for managing the medications given to Veterans who have visited a clinic or who have received prescriptions upon discharge from the hospital
* Automatically generates prescription labels, and prints refill request forms
* Medication histories are kept online to permit checks for potential interactions
* Profiles can be generated to assist the clinician in managing the patient's medication regimen
* Management reports aid the pharmacy in controlling inventory and costs

eInsurance

* Determination of eligibility for claimed insurance (Verification)

ePayments

* Billing, Collections, Patient Account Profiling, EDI Lockbox

eAdmin

* Content Management

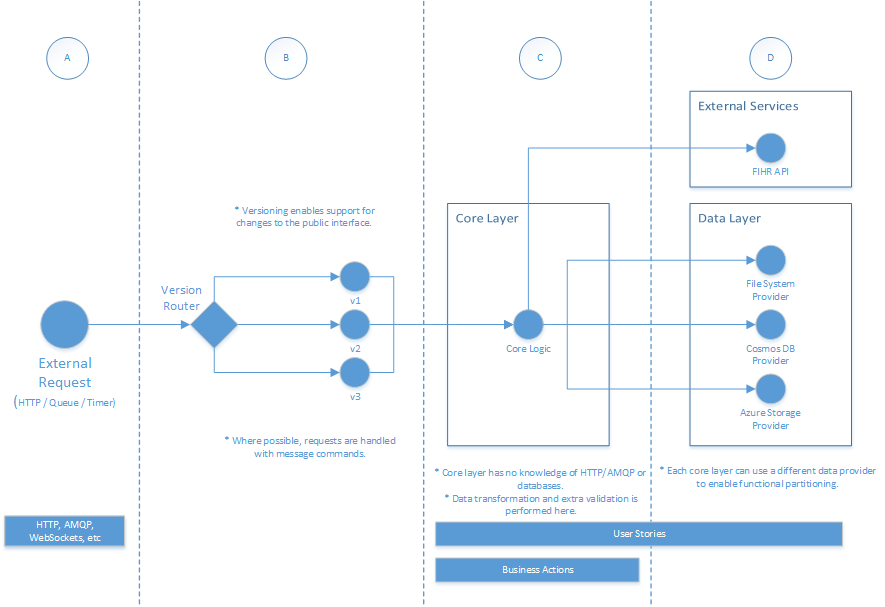
**ARCHITECTURE**

MCCF is a web-based system built on the Angular 2(Version 4) framework and Node,js. Angular is a TypeScript-based open-source front-end web application platform for building complex web applications. Node.js is a JavaScript run-time environment for executing JavaScript code server-side.

TAS backend services are driven by Node v8.

The following diagram lays out the general architecture of a service.

Figure



## Document Orientation

The MCCF Technical manual is intended for technical personnel, software developers and computer system managers. Technical personnel should be familiar with software development procedures, software development tools, and Unix operating system. Software developers should be familiar with Angular, Node.js, JavaScript (ES6), TypeScript, CSS, and HTML

The MCCF Technical Manual is divided into major sections for general clarity and simplification of the material being presented. The Implementation and Maintenance Section provides information on any aspect of the package that is site configurable.

### Disclaimers

#### Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed and/or modified freely provided that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

#### Documentation Disclaimer

The appearance of external hyperlink references in this manual does not constitute endorsement by the Department of Veterans Affairs (VA) of these Web sites or the information, products, or services contained therein. The VA does not exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of the VA.

### References

# Implementation and Maintenance

For development MCCF requires the installation of Node.js and Angular2 client.

## System Requirements

CentOS Linux v7

### Hardware Requirements

N/A

### Software Requirements

Angular (version 4)

Node.js v8

MCCF requires a browser to run the application: Chrome v60, Internet Explorer v11.

### Database Requirements

Azure Cosmos DB and Azure SQL Database

## System Setup and Configuration

The client-side code for MCCF is written in Angular TypeScript. A native Angular Client program, which is installed on a software developer’s computer, installs the code with the command npm install. The code is then compiled into JavaScript and HTML bundles with the command npm build. These bundles are “served” by a web server to the browser user interface. A software developer can run the client-side application locally with the command npm start and entering localhost:4200 into the browser URL field.

The server-side code for MCCF is written in Node v6 ES6. The application is installed with a standard npm install command. Tests are performed with the mocha command. A software developer can run the application locally with the grunt command accessing port 8193.

# Files

The MCCF base code directory structure is inserted below:

Table 1: MCCF Base Code Directory Structure

/mccf\_tas\_core  
/src  
 /app  
 /assets  
 /environments  
 index.html  
 polyfills.ts  
 test.ts  
 tsconfig.spec.json  
 favicon.ico  
 main.ts  
 styles.css  
 tsconfig.app.json  
 typings.d.ts  
/scripts  
.angular-cli.json  
.bowerrc  
.jazzignore  
inventory  
protractor.conf.js  
tsconfig.json  
.jshintrc  
Jenkinsfile  
README.md  
tslint.json  
.travis.yml  
/karma  
requirements.yml  
Vagrantfile  
ansible.cfg  
karma.conf.js  
runFortifyScan.sh\*  
editorconfig  
bower.json  
LICENSE  
/.git  
devmachine.yml  
package.json  
site.yml  
.gitignore  
/e2e  
package-lock.json

DEVELOPMENT AND PRODUCTION ENVIRONMENT DIFFERENCES

When Angular code is compiled on a developer’s computer, supporting module libraries are downloaded from the NPM (Node Package Manager) public repository. In production, MCCF does not allow this because the system is behind a VA firewall. When MCCF is in production, Angular uses the Sinopia service for installing modules.

# Routines

N/A

# Exported Options

N/A

# Mail Groups, Alerts, and Bulletins

## Build Notifications

* Jenkins build success [email]
* Jenkins build failure [email]
* MCCF\_TAS deployment notification [email]

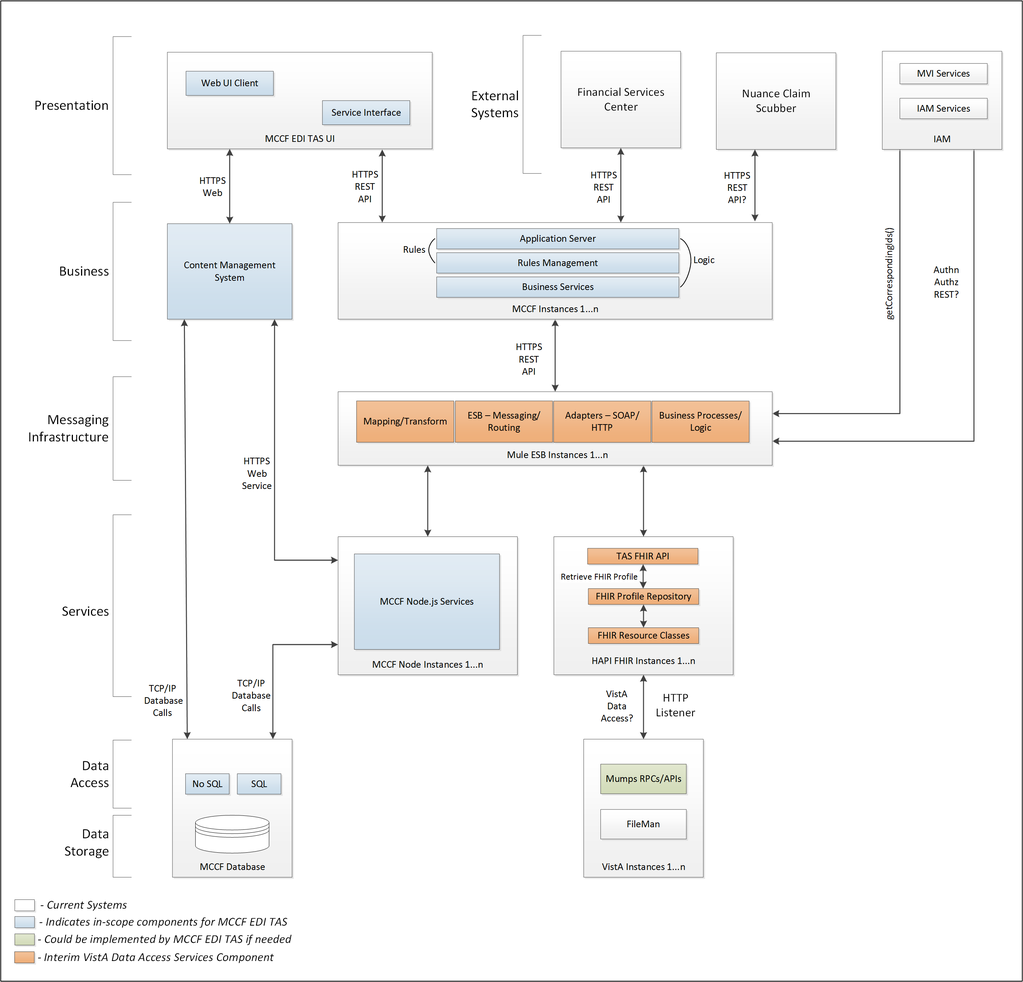
## Application Notifications

* [no email defined at this time]

# Public Interfaces

MCCF is a “reporting and controlling” application and interfaces with numerous data sources. These data sources include MCCF App Databases, FHIR, FSC, IAM, Mule, Vista, and the Scrubber. The following is a graphical representation of MCCF application interfaces.

Figure



## Integration Control Registrations

N/A

## Application Programming Interfaces

N/A

## Remote Procedure Calls

N/A

## HL7 Messaging

### HL7 version 2

* Originally created in 1989, the HL7 version 2 standard (also known as Pipehat) supports hospital workflows.
* HL7 v2.x messages use a non-XML encoding syntax based on segments (lines) and one-character delimiters.

### HL7 version 3

* The HL7 version 3 standard developed in 2005 also supports all healthcare workflows. The version 3 standard, unlike version 2, is based on a formal methodology and object-oriented principles.
* The HL7 version 3 messaging standard defines a series of Secure Text messages called interactions to support all healthcare workflows.
* HL7 v3 messages are based on an XML encoding syntax, as shown in this example:

1. <POLB\_IN224200 ITSVersion="XML\_1.0" xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <id root="2.16.840.1.113883.19.1122.7" extension="CNTRL-3456"/><creationTime value="200202150930-0400"/>

### FHIR is a standard and an API

* FHIR is the next generation healthcare messaging standard   
  created by HL7 and is designed to enable   
  exchange of health information through   
  RESTful services using HTTP.
* Data representation can be in JSON or XML. JSON and XML are the most standard formats for representing data.
* FHIR is built on previous HL7 standards   
  like HL7 2.x, 3.x, CDA etc.
* See <https://www.hl7.org/fhir/> for   
  more information

{

"resourceType": "Patient",

"id": "pat3",

"text": {

"status": "generated",

},

"identifier": [

{

"use": "usual",

"type": {

"coding": [

{

"system": "http://hl7.org/fhir/v2/0203",

"code": "MR"

}

]

},

"system": "urn:oid:0.1.2.3.4.5.6.7",

"value": "123457"

}

],

"active": true,

"name": [

{

"use": "official",

"family": [

"Notsowell"

],

"given": [

"Simon"

]

}

],

"gender": "male",

"birthDate": "1982-01-23",

"deceasedDateTime": "2015-02-14T13:42:00+10:00",

"managingOrganization": {

"reference": "Organization/1",

"display": "ACME Healthcare, Inc"

}

}

## Web Services

The MCCF application front-end uses Angular HTTP and WebSockets for communicating with data services. These are standard communication protocols for browser-based web applications.

# Standards and Conventions Exemptions

## Internal Relationships

N/A

## Software-wide Variables

N/A

# Security

## Security Menus and Options

N/A

## Security Keys and Roles

N/A

## File Security

N/A

## Electronic Signatures

N/A

## Secure Data Transmission

The application uses the Javascript library **crypto-js** for encoding and decoding data using protocols Base64, SHA256, and other standard internet encoding algorithms.

# Archiving

N/A

# Non-Standard Cross-References

N/A

# Troubleshooting

## Special Instructions for Error Correction

N/A

## National Service Desk and Organizational Contacts

N/A

# Acronyms and Abbreviations

MCCF – Medical Care Collection Fund

EDI – Electronic Data Interchange

TAS – Transaction Application Suite

NPM – Node Package Manager

Template Revision History

| **D****ate** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| July 2016 | 1.1 | Updated instructional text to simplify content. | OI&T Documentation Standards Committee |
| June 2016 | 1.0 | Initial version | OI&T Documentation Standards Committee |