



**Centers for Medicare & Medicaid  
Services**  
**Federally Facilitated Exchange**  
HHSM-500-2015-002446C

# **FFE UI Application Principles for Integration with FFE APIs**

---

**Version 3.20**  
June 25, 2020



consulting | technology | outsourcing

---

## Reviewers

| Name           | Date       | Team  |
|----------------|------------|-------|
| Dean Mohs      | 08/02/2017 | CCIIO |
| Jenny Song     | 08/08/2017 | NAVA  |
| Sabiheen Abdul | 08/09/2017 | MSI   |
|                |            |       |
|                |            |       |

## Document Information

|                                 |                                             |
|---------------------------------|---------------------------------------------|
| <b>SharePoint Location</b>      | <a href="#">CMS – FFE – Use Case Design</a> |
| <b>Application</b>              | FFE – SES                                   |
| <b>Team</b>                     | Accenture Federal Services – CMS FFE – E&E  |
| <b>Business / Process Owner</b> | CCIIO                                       |
| <b>Functional Designer</b>      | SES team                                    |

## Document Control

| Author                       | Version | Rev. date  | Summary of Changes             | Section    | Page |
|------------------------------|---------|------------|--------------------------------|------------|------|
| Cliffton Goh                 | 1.0     | 07/25/2017 | Initial DRAFT                  | All        | All  |
| Grace Arnold and Dena Nelson | 1.1     | 07/31/2017 | Provided comments              | All        | All  |
| Dean Mohs/CCIIO              | 1.1     | 08/02/2017 | Provided comments              | Appendix C | 21   |
| Jenny Song/NAVA              | 1.1     | 08/08/2017 | Provided comments              | All        | All  |
| Sabiheen Abdul/MSI           | 1.1     | 08/09/2017 | Provided comments              | All        | All  |
| Emily Shea                   | 2.0     | 08/22/2017 | Incorporated business feedback | All        | All  |

| Author      | Version | Rev. date  | Summary of Changes                                                                                                                                                                                                       | Section                 | Page          |
|-------------|---------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|---------------|
| AFS & CCIIO | 2.1     | 11/09/2017 | Added a summary of the supplemental documentation: Application Services Companion Guide & UI Question Companion Guide.<br>Included App 3.0 section flows and companion guides.<br>Added content for validation guidance. | All                     | All           |
| AFS         | 2.2     | 2/26/2018  | Updated the UI Question Companion Guide, Application Services Companion Guide, Configuration Data, and Exception Error Handling. Also added Section 3.3.6 to explain how the taxReturnFilingStatusType is set.           | 3.2-3.4 & 3.3.6 & 5.2.4 | 13-14 & 21-26 |
| AFS         | 3       | 4/18/2018  | Updated the UI Question Companion Guide (updates in blue font), Application Services Companion Guide, and Error Code list                                                                                                | 3.2-3.4 & 5.2.4         | 13-14 & 21-24 |
| AFS         | 3.1     | 5/16/2018  | Updated the UI Question Companion Guide (updates in red font)                                                                                                                                                            | 3.4                     | 14            |
| CCIIO       | 3.2     | 5/25/2018  | Updated the UI Question Companion Guide (updates in purple font)                                                                                                                                                         | 3.4                     | 14            |
| CCIIO       | 3.4     | 7/3/2018   | Updated the UI Question Companion Guide (updates in green font)                                                                                                                                                          | 3.4                     | 14            |
| CCIIO       | 3.5     | 8/23/2018  | Updated the UI Question Companion Guide (updates in pink font)                                                                                                                                                           | 3.5                     | 14            |
| CCIIO       | 3.6     | 9/12/2018  | Added updated Application Services Companion Guide                                                                                                                                                                       | 3.2                     | 13            |
| CCIIO       | 3.7     | 10/10/2018 | Added updated UI Question Companion Guide                                                                                                                                                                                | 3.3                     | 14            |
| CCIIO       | 3.8     | 2/1/2019   | Added updated UI Question Companion Guide                                                                                                                                                                                | 3.3                     | 14            |
| CCIIO       | 3.9     | 2/4/2019   | Added updated Application Services Companion Guide                                                                                                                                                                       | 3.2                     | 13            |
| CCIIO       | 3.10    | 2/8/2019   | Added updated UI Question Companion Guide                                                                                                                                                                                | 3.3                     | 14            |
| CCIIO       | 3.11    | 3/1/2019   | Added updated UI Question Companion Guide with revised user guides                                                                                                                                                       | 3.3                     | 14            |
| CCIIO       | 3.12    | 4/2/2019   | Added updated UI Question Companion Guide with revised user guides                                                                                                                                                       | 3.3                     | 14            |
| CCIIO       | 3.13    | 4/4/2019   | Added updated Application Services Companion Guide                                                                                                                                                                       | 3.2                     | 13            |
| CCIIO       | 3.14    | 6/24/2019  | Added updated UI Question Companion Guide (updates in green font)                                                                                                                                                        | 3.3                     | 15            |

| Author | Version | Rev. date | Summary of Changes                                                 | Section | Page |
|--------|---------|-----------|--------------------------------------------------------------------|---------|------|
| CCIIO  | 3.15    | 9/13/2019 | Added updated Application Services Companion Guide                 | 3.2     | 13   |
| CCIIO  | 3.15    | 9/13/2019 | Added updated UI Question Companion Guide (updates in orange font) | 3.3     | 15   |
| CCIIO  | 3.16    | 1/29/2020 | Added updated UI Question Companion Guide (updates in purple font) | 3.3     | 15   |
| CCIIO  | 3.17    | 2/7/2020  | Added updated Application Services Companion Guide                 | 3.2     | 14   |
| CCIIO  | 3.18    | 1/29/2020 | Added updated UI Question Companion Guide (updates in red font)    | 3.3     | 15   |
| CCIIO  | 3.19    | 3/4/2020  | Added updated Application Services Companion Guide                 | 3.2     | 14   |
| CCIIO  | 3.18    | 6/25/2020 | Added updated UI Question Companion Guide (updates in green font)  | 3.3     | 15   |
| CCIIO  | 3.19    | 6/25/2020 | Added updated Application Services Companion Guide                 | 3.2     | 14   |

---

## Contents

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| 1. Introduction.....                                                      | 6  |
| 1.1 Purpose.....                                                          | 6  |
| 1.2 Audience.....                                                         | 7  |
| 1.3 Assumptions/Constraints/Risks .....                                   | 7  |
| 1.3.1 Assumptions .....                                                   | 7  |
| 1.3.2 Constraints.....                                                    | 7  |
| 1.3.3 Risks.....                                                          | 7  |
| 2. Application Policy Principles.....                                     | 8  |
| 2.1 Overview.....                                                         | 8  |
| 2.2 Eligibility for All Insurance Affordability Programs .....            | 8  |
| 2.3 Limitations on Information Collected.....                             | 8  |
| 2.4 Accessibility and Disability Regulation.....                          | 8  |
| 2.5 Application Operational Assistance.....                               | 9  |
| 2.6 Real-time Eligibility Determinations.....                             | 9  |
| 3. System Build Requirements .....                                        | 9  |
| 3.1 Eligibility Event Dependencies.....                                   | 9  |
| 3.2 Application Services .....                                            | 13 |
| 3.3 Application Navigation.....                                           | 14 |
| 3.3.1 Branch for Financial Assistance Applications .....                  | 15 |
| 3.3.2 Dynamic Questions .....                                             | 15 |
| 3.3.3 Backend Interaction.....                                            | 15 |
| 3.3.4 Option for Consumer Not to Provide Information .....                | 16 |
| 3.3.5 Backward Navigation.....                                            | 16 |
| 3.3.6 Setting Tax Return Filing Status .....                              | 16 |
| 3.4 Application State and System Configuration .....                      | 17 |
| 4. Sample UI Interactions with Application Services.....                  | 17 |
| 4.1 Sample Interaction Diagrams.....                                      | 17 |
| 4.1.1 Online Use Case: Consumer Updates Application in Two Sessions ..... | 17 |
| 4.1.2 Online Use Case: Consumer Adds a Member.....                        | 18 |
| 4.2 App 3.0 Section Flows.....                                            | 19 |
| 5. Technology Principles .....                                            | 19 |
| 5.1 Overview.....                                                         | 19 |
| 5.2 Integration with FFE APIs .....                                       | 20 |
| 5.2.1 High-level Objectives.....                                          | 20 |
| 5.2.2 Client System Authentication and Authorization.....                 | 20 |
| 5.2.3 UI Application Technical Assumptions .....                          | 21 |
| 5.2.4 Exception Handling.....                                             | 22 |
| 6. Onboarding Process.....                                                | 25 |
| 7. Open Decisions .....                                                   | 25 |
| Appendix A: Acronyms .....                                                | 26 |
| Appendix B: FEPS Class and Attribute Definitions .....                    | 27 |
| Appendix C: Application Section Guidelines.....                           | 28 |
| Contact Information Section .....                                         | 28 |
| Designate Agent/Broker .....                                              | 28 |
| Help paying for Coverage Section.....                                     | 28 |

---

|                                                            |    |
|------------------------------------------------------------|----|
| SSNs, Citizenship, and Immigration Status Section.....     | 29 |
| Family and Household Section .....                         | 29 |
| Questions to screen for Medicaid eligibility Section ..... | 30 |
| Income Section .....                                       | 30 |
| Program Specific Questions Section.....                    | 30 |
| Special Enrollment Periods Section .....                   | 30 |
| Review, Sign, & Submit Section .....                       | 30 |
| Eligibility Results Section .....                          | 31 |

### Figures

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| Figure 3-1: SES Event Dependency with Attestation Groupings .....              | 13 |
| Figure 4-1: Online Use Case- Consumer Updates Application in Two Sessions..... | 18 |
| Figure 4-2: Online Use Case- Consumer Adds a Member .....                      | 19 |
| Figure 5-1: Authentication and Authorization Interaction.....                  | 21 |
| Figure 5-2: Validation Guidance Responses.....                                 | 23 |
| Figure 5-3: Detailed Error Code Messages.....                                  | 25 |

---

# 1. Introduction

## 1.1 Purpose

This document outlines the guiding principles of a user-interface (UI) application or system and its integration with the Federally Facilitated Exchange (FFE) Application Programming Interfaces (APIs) for eligibility services. The guidance covers both technical and policy requirements for integrating with the FFE APIs. Policy guidance includes explanations of relevant CMS regulations and sub-regulatory guidance governing the questions and functionality that must be included in any UI that produces eligibility results for Exchange coverage. The guidance in this document also includes clarification of back-end eligibility logic and when and how a UI requestor should plan to interact with APIs for eligibility services. Technical guidance includes technical objectives and assumptions, expected format of inputs, and error handling that is sent between UI requestors and the FFE APIs.

The technical and policy guidance aims to ensure accurate eligibility determinations and a high quality user experience while also providing flexibility to UI requestors. This document, as well as the additional documentation embedded in this guidance for more detailed clarifications, are in draft form and are subject to change. The business team welcomes feedback from UI requestors, especially EDE partners, on any major concerns. The guidance is organized into the following sections:

1. Application Policy Principles
  - a. This section includes high level guidelines for creating a single streamlined application in the FFE, such as eligibility for all insurance affordability programs, attestation collection regulations, and application operational assistance.
2. System Build Requirements
  - a. This section provides an overview of the eligibility flows and application services and guidance on how application questions are displayed based on interaction with the back-end.
  - b. This section includes the following companion guides and supporting documentation to support UI requestors. Note that the documentation is in draft form and is subject to change.
    - i. A **UI Question Companion Guide** provides policy background and requirements on how to collect attestations in the application.
    - ii. An **Application Service Companion Guide** includes information and validation logic for all data elements that are inputs to the eligibility flows and application services.
3. Sample UI Interactions with Application Services
  - a. This section includes **example application flows** to provide requestors with guidance on which application services to call in different consumer scenarios.
  - b. The **App 3.0 Section Flows** provide an example of application navigation and recommended calls to application services.

#### 4. Technology Principles

- a. This section provides technical guidelines on how to integrate with the FFE application APIs.

## 1.2 Audience

This document's target audience includes functional SMEs, business analysts, technical and solution architects, developers, development leads, security analysts, system integrators, and testers.

## 1.3 Assumptions/Constraints/Risks

### 1.3.1 Assumptions

Listed below are the assumptions that guided the identification and development of the design stated in this document. These assumptions are intended to promote mutual understanding between FFE partners, UI stakeholders, CMS stakeholders, and API service providers.

1. This document provides guidance for any application that calls FFE APIs for eligibility and other functions. The application requestor must adhere to the guidelines in this document in order to provide a correct eligibility determination for a consumer.
2. This document will evolve with additional detail as necessary.
3. All applications requesting to call FFE APIs to determine eligibility or other functions must be approved by CMS.
4. The requestor can process an in-progress application for completion or report a change on a submitted application. The FFE Update API can update an in-progress or submitted application.

### 1.3.2 Constraints

The following constraints apply to this document.

1. Only CMS approved applications will be authorized to call the FFE APIs.

### 1.3.3 Risks

The following risks apply to this document.

1. This document includes expectations and requirements for applications to ensure the consumer experience and accurate eligibility results. If the application does not ask the questions as specified in this document, the eligibility outcome could be incorrect.

---

## **2. Application Policy Principles**

### **2.1 Overview**

Per regulations at 42 CFR 435.916 and 45 CFR 155.405, the Marketplace must determine eligibility using a single streamlined application. As internal and external partners design their own user interfaces, it is important for entities to be aware of the expectations and minimum requirements for applications that are used to purchase health coverage on the Marketplace.

### **2.2 Eligibility for All Insurance Affordability Programs**

The law requires that one application be used for all programs under the Patient Protection and Affordable Care Act to ensure a proper adjudication of eligibility, minimize burden on applicants who may not know which program they are eligible for, and allow households with members eligible for different programs to apply using a single application. For applicants seeking financial assistance, the streamlined application must ask questions sufficient to provide an eligibility determination for all insurance affordability programs, including Advance Payments of the Premium Tax Credit (APTC), Cost-sharing Reductions (CSR), Medicaid, and the Children's Health Insurance Program (CHIP). The UI cannot choose to only ask questions for Marketplace eligibility and, in most cases, applicants may not be determined eligible for APTC unless they have been determined ineligible for Medicaid and CHIP.

### **2.3 Limitations on Information Collected**

Within the application UI, applicants can only be asked questions that are necessary for determining eligibility for coverage in a Qualified Health Plan (QHP) and all insurance affordability programs, or for the administration of these programs. DE entities must not ask questions that are not essential to these purposes or programs as part of the application UI. DE entities should not require consumers to enter the same information multiple times; however, if a DE entity prepopulates the response to an application question based on previously collected information, the DE entity must display the application question and provide the consumer the ability to edit that answer. Business logic is required to determine which questions to ask based on prior answers, as described in Section 3.3 of this document.

For example, in accordance with 42 C.F.R. § 435.907, states may not request citizenship and immigration information from “non-applicants,” which is the term for individuals who are identified on an application of an individual who is applying for coverage but who are themselves not applying for coverage. Certain requests for information must also be optional—as designated in the Application UI Companion Guide—and all optional fields must be clearly marked as such.

### **2.4 Accessibility and Disability Regulation**

The application will use plain language, be available in both Spanish and English and meet standards regarding accessibility to consumers with disabilities. The application will adhere to Section 508 of the Rehabilitation Act of 1973, as amended in 1998 (29 U.S.C. § 794 (d)), and the broader context of other related accessibility laws and policy.

---

## 2.5 Application Operational Assistance

The UI application or system will have pathways for consumers who have questions or experience technical issues and need assistance with the application process. The application user interface should also be built to support consumer technical needs, such as the ability to save and return later to the application, and the ability to review summaries and edit application information before submission.

## 2.6 Real-time Eligibility Determinations

Consumers will receive eligibility determinations in real time immediately following application submission. This means that the requestor may not wait and send applications in batches. If the FFE data verification sources are down, this may not be possible. A resubmission from the requestor will be required later.

# 3. System Build Requirements

This section provides an overview of the eligibility flows and application services and guidance on how application questions are displayed based on interaction with the back-end.

## 3.1 Eligibility Event Dependencies

The order in which attestations are collected on the application relies on the eligibility event dependencies. This section describes the impact of attestation groupings and eligibility events on downstream events.

There are specific eligibility events that impact downstream determination and verification events. A determination event provides an eligibility result such as QHP, APTC, Medicaid, and CHIP eligibility. A verification event provides a result of a verification to use as an input for a determination, such as SSN, citizenship, and income verification. The verification and determination flows are represented in boxes in Figure 3-1 below. A high-level description of each flow is as follows.

1. **Incarceration Verification:** Verifies a consumer's incarceration attestation with Social Security Administration (SSA).
2. **Citizenship Verification:** Verifies a consumer's citizenship attestation with SSA.
3. **SSN Verification:** Verifies a consumer's attested SSN with SSA.
4. **Immigration Verification:** Verifies a consumer's attested eligible immigration status with DHS.
5. **Derive Residency Status:** Derives a consumer's residency status based on attested home, mailing, or transient address.
6. **Determine Medicaid Standard:** Determines an applicant's Medicaid standard to use in income verification.
7. **Household Composition:** Builds a member's Medicaid and tax household to use in income calculation.

8. **Income Verification:** Calculates annual and current income, calls the hub to retrieve income data from Equifax, IRS and SSA, and verifies annual and current income for Medicaid/CHIP and APTC eligibility.
9. **Employer Sponsored Coverage (ESC) Minimum Essential Verification (MEC) Verification:** Verifies whether a consumer has access to ESC MEC for the purpose of determining APTC eligibility.
10. **Non-ESC MEC Verification:** Verifies whether a consumer has access to non-ESC MEC for the purpose of determining APTC eligibility.
11. **Determine Residency/QHP:** Verifies that an applicant's residency address is in the State in which the applicant is applying and determines eligibility for QHP.
12. **Preliminary Eligibility:** Determines an applicant's preliminary eligibility for Medicaid/CHIP and APTC which depends on residency, lawful presence, and income. Once preliminary eligibility is determined, the application asks program-specific questions to determine final Medicaid/CHIP or APTC eligibility.
13. **Determine Final Medicaid/CHIP:** Determines an applicant's final Medicaid or CHIP eligibility.
14. **Verify Indian Status:** Verifies an applicant's status as a member of a tribe for CSR eligibility.
15. **Determine Effective Date:** Calculates the QHP coverage effective date to determine when the applicant's QHP coverage can begin.
16. **Determine APTC Eligibility:** Determines eligibility for APTC.
17. **Determine CSR Eligibility:** Determines eligibility for CSRs.
18. **Determine Attestation Special Enrollment Period (SEP):** Determines eligibility for attestation SEPs, such as marriage SEP, loss of MEC SEP, release from incarceration SEP, and gain lawful presence SEP.
19. **Determine System SEP:** Determines eligibility for PTCCSR SEP.

The application needs to consider event dependencies when deciding the order in which attestations are collected to avoid unnecessary validation errors between the application and APIs. For example, in the figure below, income verification in swim lane 2 depends upon the application, income, and family attestation groupings so the application must collect the attestations in those groups before it can verify income and determine preliminary eligibility. The bubbles above each box represent the attestation groupings on which the flow depends. For example, Citizen/Immigration depends on 3) Member.Demographics and 6) Member.Lawfulpresence attestation groupings.

Below is a high-level description of the types of attestations in each attestation grouping.

## 1. Application

- a. Includes application-level attestations, such as legal agreements (formerly known as legal attestations), signature elements, contact method, and requesting financial assistance indicator.

- b. These attestations are used throughout all eligibility flows.
- 2. Household**
  - a. Includes application-level attestations, such as tax relationships, family relationships, and legal relationships.
  - b. These attestations are used to build the Medicaid and tax household to determine eligibility for Medicaid/CHIP and APTC/CSR.
- 3. Member.Demographics**
  - a. Includes member-level attestations, such as name, date of birth, sex, SSN, address (home, mailing, or transient), race/ethnicity, and marital status.
  - b. These attestations are used in hub calls and to derive and verify residency address.
- 4. Member.Income**
  - a. Includes member-level attestations, such as attested annual and current income amount, current income types and details for each income type, and income discrepancy attestations.
  - b. These attestations are used on the FA path for current and annual income verification flows.
- 5. Member.Family**
  - a. Includes member-level attestations, such as tax filing status, pregnancy attestation, student status, and foster care details.
  - b. These attestations are used to determine Medicaid standard, verify annual income, and determine prelim eligibility.
- 6. Member.Lawful Presence**
  - a. Includes member-level attestations, such as citizenship, immigration status, and immigration details.
  - b. These attestations are used to verify Citizenship and Immigration status.
- 7. Member.Medicaid CHIP**
  - a. Includes member-level attestations, such as the Medicaid denied attestation, dependent child covered, waiting period, and state health benefits attestation.
  - b. These attestations are used on the FA path to determine final Medicaid and CHIP eligibility. The attestations are collected depending on the result of the preliminary eligibility flow.
- 8. Member.Insurance Coverage**
  - a. Includes member-level attestations, such as other health coverage through Tricare or Peace Corp and employer sponsored coverage details.
  - b. These attestations are used on the FA path to verify Non-ESC MEC and ESC MEC for APTC/CSR eligibility. The attestations are collected depending on the result of the preliminary eligibility flow.

---

## 9. Member.Other

- a. Includes member-level attestations such as special enrollment period (SEP) attestations, American Indian/Alaskan Native details, and incarceration attestations.
- b. These attestations are used to determine SEP eligibility and verify Indian status and incarceration.

Figure 3-1 is a high-level diagram of the attestation and eligibility event dependencies. The diagram can be read in three ways as follows.

1. Bottom to top: To understand the dependencies on eligibility determination results, start at the bottom of the diagram.
  - a. For example, the outcome of APTC in swim lane 6 depends on the outcome of ESC MEC and Non-ESC MEC in swim lane 5.
2. Top to bottom: To understand the dependencies of verification events on other verification or determination events, start at the top of the diagram.
  - a. For example, Incarceration, Immigration and Citizenship, and residency status in swim lane 1 impact QHP eligibility in swim lane 4.
3. Double Arrow interaction: The double arrow interaction in swim lane 4 shows how an applicant can switch from preliminary Medicaid/CHIP eligible to APTC/QHP eligible. The diamond in swim lane 4 represents the application 'branch.'
  - a. If an applicant is determined preliminary Medicaid or CHIP eligible, the applicant is asked Medicaid/CHIP specific questions and Final Medicaid/CHIP is run. If the applicant answers the Medicaid/CHIP-specific questions in a way that makes the applicant ineligible for Medicaid/CHIP, Prelim Eligibility is run again to see if the applicant can be determined preliminary APTC eligible.

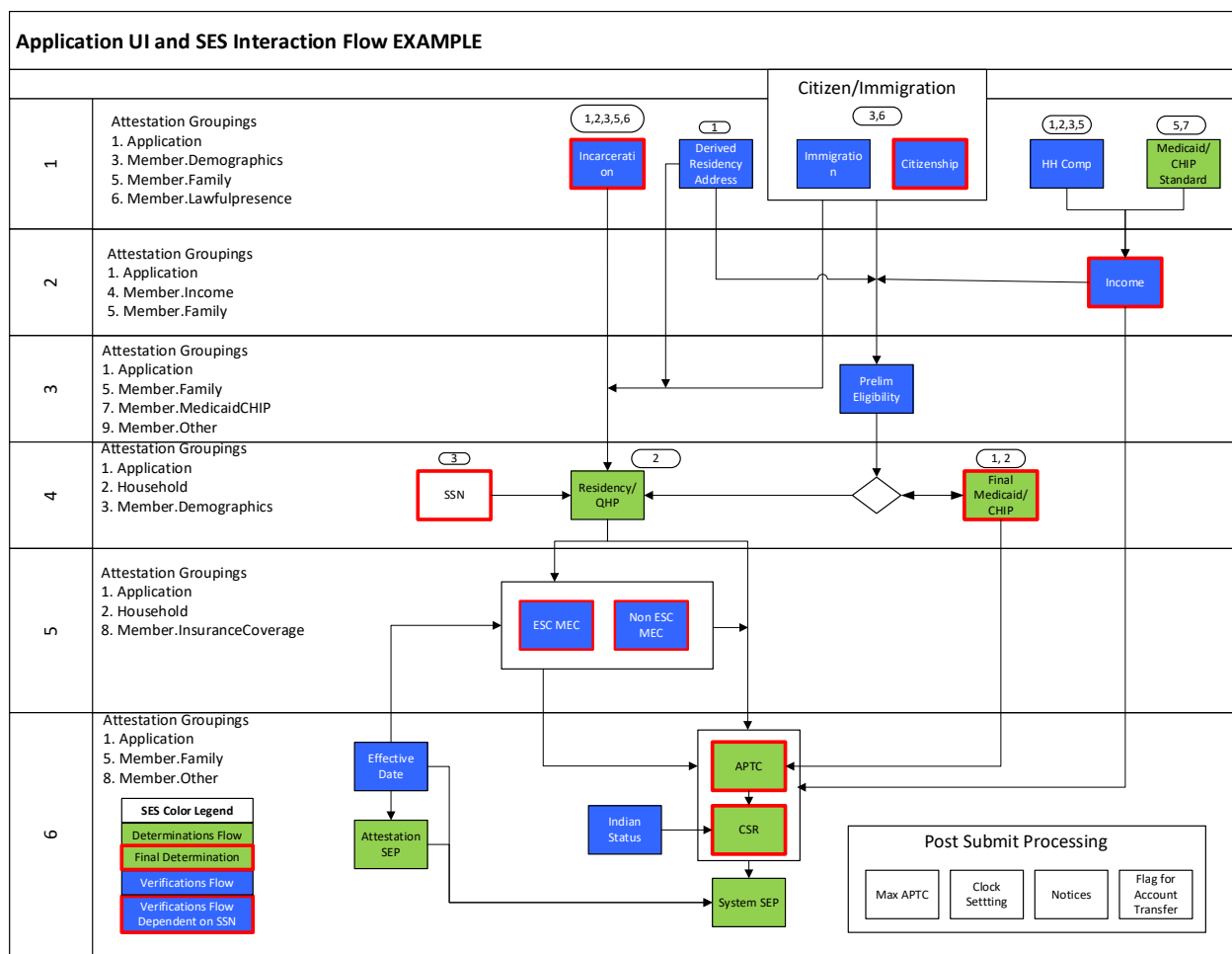


Figure 3-1: SES Event Dependency with Attestation Groupings

### 3.2 Application Services

The application will integrate with the FFE APIs to determine eligibility. The following lists the FFE APIs and a brief description of each:

1. Create Application
  - a. This API creates an initial application and an application from the prior coverage year.
2. Update Application
  - a. This API allows for modifications of an existing application based on the inputs provided by the requestor. This service updates in-progress and submitted applications. Based on the inputs to the service, specific verification and eligibility logic is re-computed.
3. Add and Remove Member
  - a. This API allows a requestor to add or remove a member from an application.
4. Submit Application

- 
- a. This API allows the requestor to finalize and submit the application. The service computes and/or re-computes events to ensure that all required fields are present and valid to submit the application. Upon successful submission, the service generates a submitted application that is immutable and can be used to process enrollment and downstream processes.
  5. Get Application
    - a. This API provides a detailed and summary view of a consumer's application.
  6. Delete Application
    - a. This API allows for a requestor to delete the latest in-progress application version or a submitted application version that is not tied to an active enrollment. There are certain restrictions to this API based on user role, system role, and operation access.
  7. Reference Data API
    - a. This API provides state and system configuration data for UI use only.

Each application API includes validations such that specific attestations are required to be collected prior to calling the API. For example, the minimum required fields to create an application include first name, last name, date of birth, and mailing address of the household contact. If the validations fail, the API will return an error message to the requestor.

The following Application Services Companion Guide includes information and validation logic for all data elements that are inputs to the eligibility flows and application services. The columns on the *Input Matrix* tab align with the Event Dependencies Diagram in Figure 3-1. Requestors can filter on the inputs for each column in the order of the Event Dependencies Diagram to identify which inputs are conditionally required based on the backend logic and/or which inputs overlap with other columns.



Application\_Services\_Companion\_Guide

### 3.3 Application Navigation

The application navigation section includes guidance for applications to provide dynamic interaction and the option for a consumer to not provide an answer to a question.

The application must be dynamic to support the following interaction types:

1. Accommodate when financial assistance is requested and when it is not
2. Question-level dynamic interaction and
3. Dynamic questions based on backend responses

The following UI Question Companion Guide provides all questions from the legacy application UI. Documentation on each application question includes: what data the question collects, when the question is displayed in the legacy application, why the question is asked (the policy

reasons), and how the question should be asked, if there is any flexibility or requirements around wording and question format.



ui\_question\_comp  
nion\_guide\_062520:

### 3.3.1 Branch for Financial Assistance Applications

The UI must accommodate applications that are requesting financial assistance and those that opt not to request financial assistance.

An application requesting financial assistance asks tax filing status questions and income questions in order to determine an applicant's preliminary eligibility for Medicaid/CHIP or APTC prior to asking additional program-specific questions. The point in the application in which an applicant is considered preliminary Medicaid/CHIP eligible or preliminary APTC eligible is known as the application "branch." The SES Update Application service will need to be called in order to determine preliminary eligibility so that only the appropriate program-specific questions can be asked. Refer to Section 4: Sample API Interactions for an example of the interaction between the application and SES Update Application.

For an application that is not requesting financial assistance, a DE entity must not ask tax filing status questions or income questions. A non-financial assistance application does not have a branch, so program-specific questions are not asked.

### 3.3.2 Dynamic Questions

The application should be dynamic such that subsequent questions are only asked when necessary in response to prior consumer attestations, including, but not limited to, attestations in reference to age, sex, citizenship, and income. For example, the pregnancy question should only be asked of females and questions regarding immigration status should only be asked of applicants who attest that they are not US citizens. However, under certain circumstances related to improving the holistic user experience and overall application flow, it may not always be the case that questions are only asked when necessary, though it is always preferable.

### 3.3.3 Backend Interaction

The application should ensure that there is dynamic interaction between the UI and backend logic. Eligibility logic verifies a consumer's attestation with data sources such as SSA, IRS, and DHS. The application should use information from the verification response to prompt appropriate follow-up questions. The application interaction opportunities between the UI and the backend are as follows:

1. **Social Security Number (SSN) retry** is needed when SSA has a partial match on data submitted for verification.
2. **Naturalized citizen follow-up attestation** is asked when SSA cannot verify citizenship for an attested citizen.

3. **Immigration SAVE Step 1a/1b follow-up questions** are requested by DHS. When an applicant attests to having eligible immigration status and provides immigration documentation, the backend calls the hub to verify the applicant's immigration status through the DHS SAVE Step 1 process. DHS can respond to the backend through the hub that they need more information to verify immigration status by returning an eligibility statement code of 32 (SEVIS ID) or 37 (retry). If the backend receives either code, the application asks applicants to enter their SEVIS ID or fix their document number or name. If the applicant enters the requested information, the backend calls the hub to verify the updating information through the DHS SAVE Step 1a or 1b process.
4. **Income discrepancy questions** are asked when income verification identifies a discrepancy between attested and external income.

### 3.3.4 Option for Consumer Not to Provide Information

There are a few situations in which the consumer can choose not to provide requested information that is required by the backend to build the Medicaid or tax household. In these cases, the UI should ask the question in a way that allows the consumer to indicate that they do not want to provide the answer and pass this answer to the Update Application API.

For example, when building the Medicaid household for a child whose non-custodial parent is their claiming tax filer, the consumer can choose not to provide information on their parents or siblings with whom the child lives. The backend Household Composition logic relies on an attestation about whether the consumer chose not to provide the information to determine whether the flow can build the member's Medicaid household. More examples and details will be provided in a separate document.

### 3.3.5 Backward Navigation

If the consumer changes attestations (e.g., through backward navigation) the client system must clear the answers to any question provided on a path that is not the final path by submitting the value 'null' in the Update Application API for those attestations. For example, if an applicant is preliminary Medicaid/CHIP eligible and answers the final Medicaid/CHIP attestations and then backward navigates and changes attestations such that they are now preliminary APTC eligible, the client system needs to 'null' the final Medicaid/CHIP attestations prior to calling the Update Application API.

### 3.3.6 Setting Tax Return Filing Status

When collecting tax information, the requester will need to return the tax return filing status of an application member as follows:

- **SINGLE\_FILER** – An unmarried application member who files taxes but does not claim dependents and is not claimed as a dependent.
- **HEAD\_OF\_HOUSEHOLD** – An unmarried application member who files taxes and claims dependents.

- MARRIED\_FILING\_SEPARATELY – A married application member who files taxes without their spouse
- MARRIED\_FILING\_JOINTLY – A married application member who files tax with their spouse.

### 3.4 Application State and System Configuration

There are specific eligibility factors or state options that FFE states can elect to cover. The application questions that may impact those state options should only be asked for an application in which the exchange state (the state in which the consumer is seeking coverage) elects the state option.

Some application questions are triggered based on SES system configuration values, such as the age in which to ask whether an applicant is pregnant or married. Those questions impacted by system configuration data should only display based on the system configuration values set by the FFE.

There are also system configuration values for date-specific reference data to which applications need to conform. For example, the UI should prevent a consumer from starting an application prior to the open enrollment start date for that coverage year.

A separate SES Reference Data API is available for external parties to request state and system configuration data.

## 4. Sample UI Interactions with Application Services

This section includes sample interactions between application requestors and the FFE APIs for different consumer scenarios and an example UI flow that interacts with the application services. As the application collects more attestations from the consumer, the application can request the Update Application API as many times as necessary. Once the application collects all attestations required up to the Submit API, the application can call Submit to receive the final eligibility.

More sample interactions are found in the document embedded below. Specific API description, functional and technical details, assumptions, and interactivity models are in APIary.



Example\_Flows\_SES  
\_Services - 07182017

### 4.1 Sample Interaction Diagrams

#### 4.1.1 Online Use Case: Consumer Updates Application in Two Sessions

The following diagram depicts the interaction between an online requestor and the FFE APIs in which the consumer starts an application, leaves half-way through, and then returns and calls the SES Update Application API to update the application. The consumer continues the application and calls the SES Submit Application API to submit.

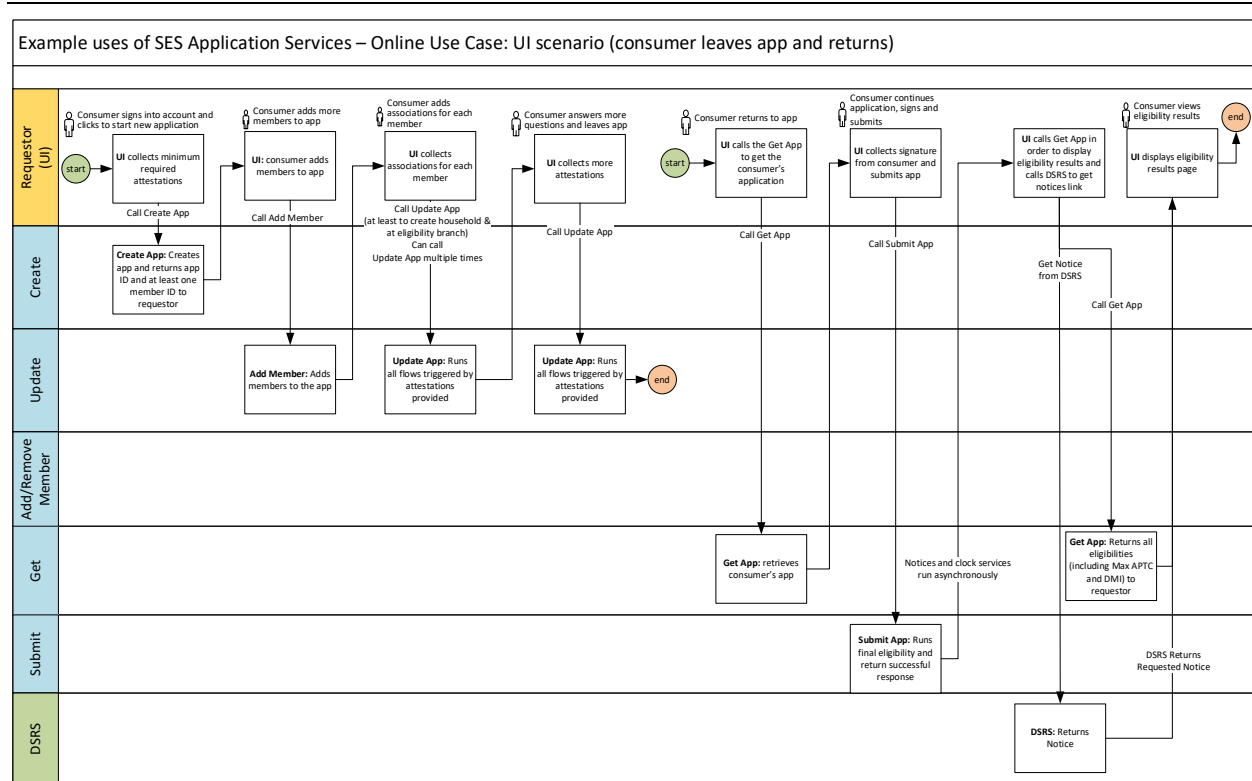


Figure 4-1: Online Use Case- Consumer Updates Application in Two Sessions

#### 4.1.2 Online Use Case: Consumer Adds a Member

The following diagram depicts the interaction between an online requestor and the FFE APIs in which a consumer calls the Add Member API to add a member, calls the Update Application API to update information for that new member, and later submits the application through the Submit Application API.

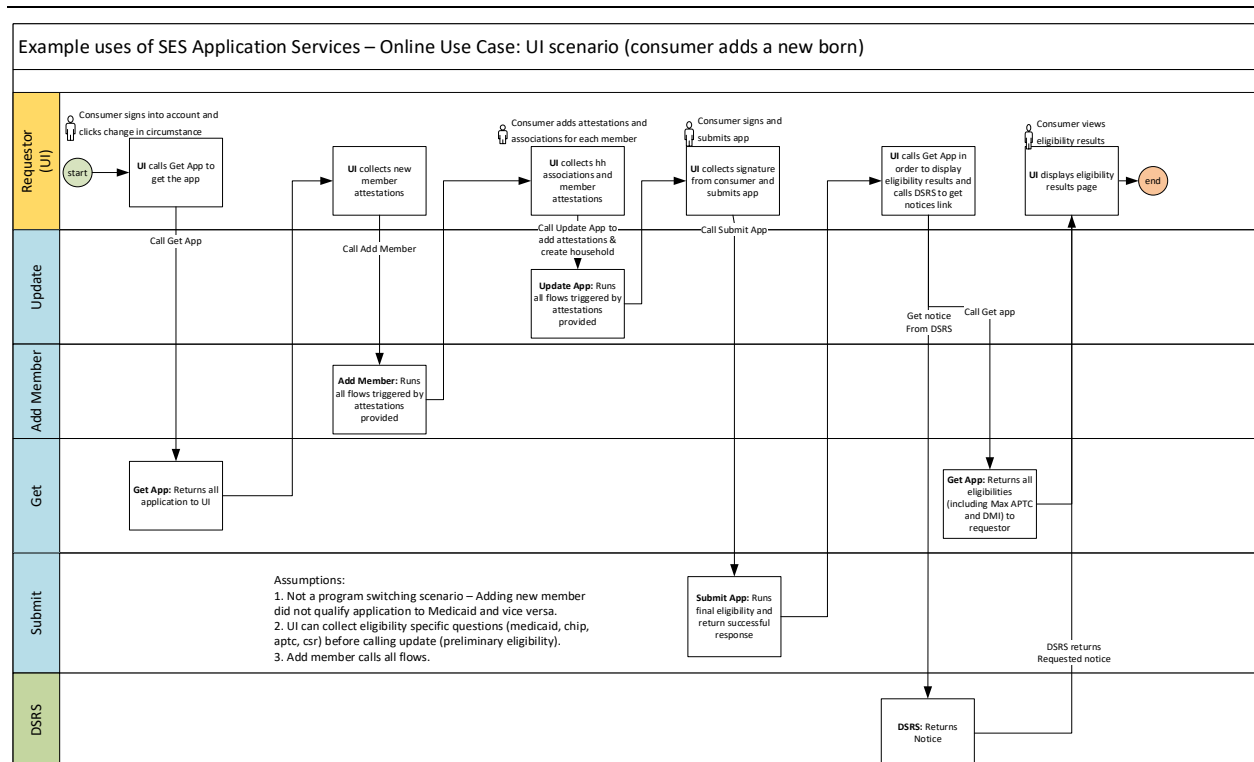


Figure 4-2: Online Use Case- Consumer Adds a Member

## 4.2 App 3.0 Section Flows

The following document provides an example of the front end sections of a UI application in the order they might appear to an applicant, and within that flow, when the calls to API services would be made. This is the model that the FFE UI on healthcare.gov (referred to here as App 3.0) plans to use for determining when to make calls to Create App, Add Member, Update App, and Submit App. CMS strongly recommends that EDE partners follow this same model as the technical and business teams agree that this sequencing will provide sufficient information for integrating relevant eligibility and verification information into the application flow to ensure that the appropriate questions are populated in subsequent sections of the application.



app\_3.0\_question\_f  
lows\_cutover\_launch

## 5. Technology Principles

### 5.1 Overview

To integrate with the FFE application APIs, the client system needs to follow a set of application principles. The APIs maintain the state of the application through the dialog between the client system and the application APIs. The APIs are stateless such that the APIs do not know where the consumer is in the application process. The principles below are not requirements but rather expectations of client system responsibilities and the integration model.

---

## 5.2 Integration with FFE APIs

FFE APIs are RESTful services. They are stateless, idempotent, and resource-based web services. Even though the functional capabilities are captured in different URIs (SES, DSRS, MCR, etc.), the general header information and protocol of communication are consistent.

### 5.2.1 High-level Objectives

The following are high level objectives of each of the FFE APIs.

1. Ensure the ease of use
  - a. Abstract and intuitive APIs that hides complex business logic from the client system.
  - b. Optimal client and service interactions.
  - c. Guidance to client systems on the application requirements and lifecycle.
  - d. Clear and concise API documentation.
2. Uphold data integrity
  - a. APIs that validate all inputs provided by the client system.
  - b. APIs that support all types of transaction so that no direct access to the database is needed.
3. Maintain auditability
  - a. Log all transactions made to each API.
  - b. Ensure sufficient information is logged for debugging purposes.
4. Safeguard security
  - a. Manage access to each API per client system and roles

### 5.2.2 Client System Authentication and Authorization

There are specific guidelines that should be followed in order to authenticate and authorize an application system to call the FFE APIs. For every request being made to the FFE APIs, an API key, a role id, and a source system name are expected to be included. The FFE APIs will use these to authenticate and authorize client requests. FFE defines authentication as “are you who you say you are?” and defines authorization as “are you allowed to run the requested operation?”

The following are authentication guiding principles:

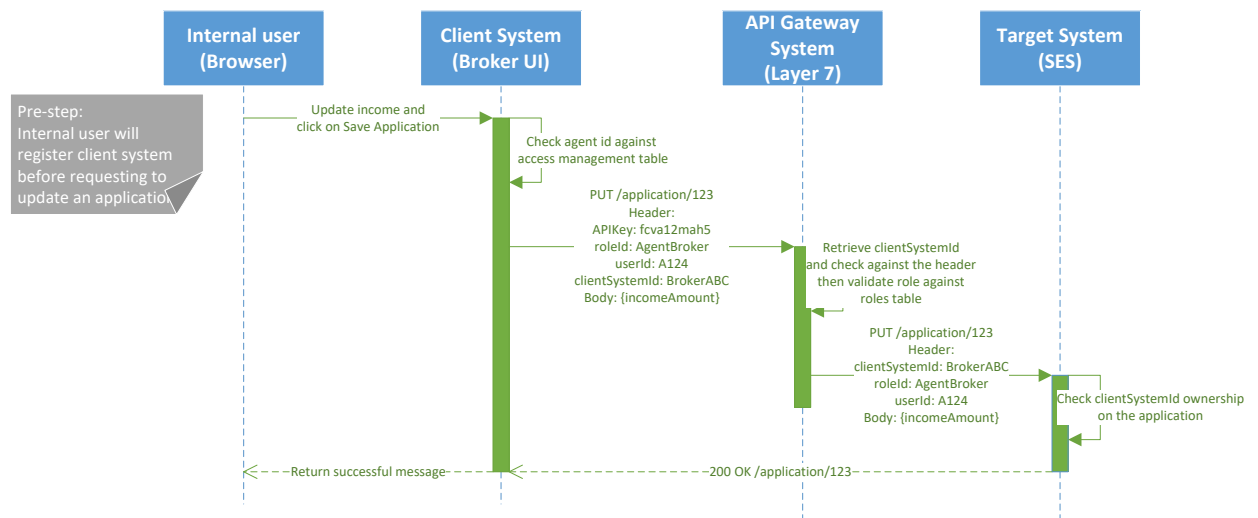
1. Client systems may use their own authentication provider or may leverage an FFE authentication provider service such as SLS or EIDM.
2. A review of the client system will be done to confirm that the client system conforms to the requirements before a secure connection between client applications and FFE APIs can be established. An API keys and enforced through the FFE API Gateway.
3. Each client system will have a unique API key assigned to them which the FFE API Gateway will translate to the client system identifier. A single client system may have users with different roles. The set of roles allowed for users of a given client system is pre-determined as part of the onboarding audit process. The FFE API Gateway will store

the list of roles allowed for request from that client system. Each request from the client system will include the role for the end user associated with the request. The FFE CA API Gateway will validate that the role in the request header is contained in table of valid roles for that client system.

The following are authorization guiding principles:

1. The client system is responsible to verify the consumer identity via the HUB's identity verification services or using a third-party service.
2. Even if the client system has full API access, depending on the specific FFE API, an ID-proofing flag can be used to authorize transactions. For example, the consumer will need to be ID proofed before SES Application Submission is called.
3. FFE APIs will allow or deny requests based on ownership of the application against the source system name, role id, and user id.

Below is a diagram of a sample interaction model for authentication and authorization:



**Figure 5-1: Authentication and Authorization Interaction**

### 5.2.3 UI Application Technical Assumptions

The following are key technical assumptions when implementing an application UI:

1. UI systems are responsible to keep session and state of the consumer interaction.
2. UI systems should have their own database to keep track of unique identifiers needed to execute business functions (e.g. application id, member id, etc.)
3. UI systems will support JSON request and response formats
4. UI systems will support both blocking and non-blocking API calls to the FFE APIs. Many of the FFE APIs relies on external data sources (e.g. SSA, IRS, etc.) to execute functionality. The complex dependencies between multiple external systems may degrade

site performance. Non-blocking calls are encouraged for business transactions that do not rely on FFE interactivity.

5. UI systems are expected to minimize the number of transactions to FFE APIs as much as possible. Many of the FFE APIs are built to handle partial updates but these APIs are not meant to be presentation APIs.
6. UI systems will log transactions to and from FFE APIs for auditability purposes.
7. UI systems will not use “guidance messages” from FFE APIs as logic to decide site navigation. These messages are additional details of possible issues and they are meant to be used by developers for triaging activities.
8. UI systems are expected to load reference data (e.g. state and system configuration) retrieved from FFE APIs during application startup. The eligibility reference data is large in volume. UI systems will need to cache these data and refresh it every month or quarter depending on the update frequency.

#### 5.2.4 Exception Handling

FFE APIs conform to the HTTP standards for error handling. The following are generic errors the APIs can return. Each API will have their own specific errors highlighted in APIary.

1. 20x
  - a. Reserved for successful transactions. All responses with 20x means that the request made it to the target application server. It does not signify any outcome of business functionalities. Within the response body, there is a STATUS field that represents business function. Possible values are SUCCESS and ERROR.
2. 40x
  - a. Reserved for error transactions. All responses with 40x means that the request format is malformed or does not have the appropriate access to execute the transaction.
3. 50x
  - a. FFE APIs should never return a 50x error on purpose. This signifies a severity-1 issue with the infrastructure hosting these services.

When validation errors are returned, requestors can use the `httpStatusCode`, `resultType`, `errorCode`, and `path` to determine how to triage the error. The table below provides the impact and next steps for the requestor based on some of the `httpStatusCodes` and `resultTypes` that can be returned by the API.

| API Response                                                                                                                 | Impact to the Requestor                                                         | Next Step for the Requestor                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li><code>httpStatusCode = 400</code></li> <li><code>resultType = error</code></li> </ul> | Stops the request, none of the inputs are saved, and none of the flows are run. | The path indicates which data element returned an error and the <code>errorCode</code> indicates the type of error. Correct the data element(s) that returned an error and re-provide all inputs in the |

|                                                                                                                      |                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                      |                                                                                                                                                      | following request. If the errorCode returns an apiMessage, then use the memberId in the apiMessage to correct the data element for the appropriate member.                                                                                                                                                                                                              |
| <ul style="list-style-type: none"> <li>• httpStatusCode = 500</li> <li>• resultType = error</li> </ul>               | Stops the request, none of the inputs are saved, and none of the flows are run.                                                                      | Report the incident to the FFM's service management team for triage.                                                                                                                                                                                                                                                                                                    |
| <ul style="list-style-type: none"> <li>• httpStatusCode = 200</li> <li>• resultType = success_with_errors</li> </ul> | Does not stop the request, the inputs are saved, and SES computes and saves the events from flows that are run and do not return a validation error. | The path indicates which data element returned an error and the errorCode indicates the type of error. Correct the data element(s) that returned an error in the following request. If the errorCode returns an apiMessage, then use the memberId in the apiMessage to correct the data element for the appropriate member.                                             |
| <ul style="list-style-type: none"> <li>• httpStatusCode = 409</li> <li>• resultType = error</li> </ul>               | Does not stop the request, the inputs are saved, and SES computes and saves the events from flows that are run and do not return a validation error. | The path indicates which data element returned an error and the errorCode indicates the type of error. Exit Submit App and return to Update App to correct the data element(s) that returned an error in the following request. If the errorCode returns an apiMessage, then use the memberId in the apiMessage to correct the data element for the appropriate member. |

Figure 5-2: Validation Guidance Responses

The API may also return the following errorCodes. The table below provides the next steps for the requestor based on the errorCode that is associated with data element that is specified in the path.

| errorCode             | Next Step for the Requestor                                                             |
|-----------------------|-----------------------------------------------------------------------------------------|
| required              | Add the data element that is specified in the path.                                     |
| invalid_field_format  | Correct the format of the data element in the path.                                     |
| invalid_enum          | Update the enum to match the list of allowable values for the data element in the path. |
| length_must_be_lesser | Reduce the length of the data element in the path.                                      |

| <b>errorCode</b>                                 | <b>Next Step for the Requestor</b>                                                                                                                                                                         |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| length_must_be_greater                           | Increase the length of the data element in the path.                                                                                                                                                       |
| value_must_be_lesser                             | Decrease the value of the data element in the path.                                                                                                                                                        |
| value_must_be_greater                            | Increase the value of the data element in the path.                                                                                                                                                        |
| invalid_coverage_year                            | Update the coverage year to be within the appropriate range of values.                                                                                                                                     |
| invalid_ssn                                      | Correct the format of the SSN for member ID in the path.                                                                                                                                                   |
| one_contact_member_required                      | Set householdContactIndicator = true for one of the application members.                                                                                                                                   |
| invalid_member_id                                | Update the contactMemberIdentifier to match one of the application members.                                                                                                                                |
| cannot_update_coverage_state                     | The coverage state for an existing application cannot be updated.                                                                                                                                          |
| invalid_income_type                              | Social Security Benefits or Unemployment cannot be selected as tribal income.                                                                                                                              |
| invalid_home_address                             | P.O Box is not a valid home address.                                                                                                                                                                       |
| provide_spouse                                   | Provide the relationship type that is specified in the errorCode for the member ID in the apiMessage.                                                                                                      |
| provide_dependent                                | Provide the relationship type that is specified in the errorCode for the member ID in the apiMessage.                                                                                                      |
| cannot_be_both_citizen_and_n<br>on_citizen       | Update either the citizenshipStatusIndicator or lawfulPresenceStatusIndicator for the member ID in the path.                                                                                               |
| naturalized_must_be_citizen                      | Update the citizenshipStatusIndicator = true for the member ID in the path because their naturalizedCitizenshipStatusIndicator = true.                                                                     |
| provide_most_recent_verifiabl<br>e_document_only | Only provide one SAVE-verifiable lawfulPresenceDocument for the member ID in the path.                                                                                                                     |
| datetime_must_be_in_the_past                     | Either the medicaidDeniedDate must be in the past, or the circumstanceChangeDate must be in the past if the circumstanceChangeType <> FUTURE LOSS_OF_MEC, for the member ID that is specified in the path. |
| datetime_must_be_in_the_futu<br>re               | The circumstanceChangeDate must be in the future if the circumstanceChangeType = FUTURE LOSS_OF_MEC for the member ID in the path.                                                                         |

| <b>errorCode</b>                         | <b>Next Step for the Requestor</b>                                                                                                                                                                         |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| datetime_must_be_after_birth_date        | The medicaidDeniedDate must be after the birth date, or the circumstanceChangeDate must be after the birth date if the circumstanceChangeType = ADOPTION or <> TAX_PENALTY, for the member ID in the path. |
| cannot_be_male_and_pregnant              | Update either the sex or pregnancyIndicator for the memberId in the path.                                                                                                                                  |
| eds_or_hub_delayed                       | If the HUB or EDS is down during submission, the requestor should use Update Application to collect a signature and completion date.                                                                       |
| invalid_countyFipsCode                   | Provide a valid FIPs code based on the state and county associated with the address.                                                                                                                       |
| cannot_provide_other_info_if_no_coverage | Either provide the insurance market type or remove the insurance policy member ID, policy number, or plan name.                                                                                            |
| cannot_provide_duplicate_market_types    | Remove duplicate insuranceMarketTypes within the enrolledCoverage segment that have the same enum value.                                                                                                   |

Figure 5-3: Detailed Error Code Messages

## 6. Onboarding Process

Additional documentation will be provided for the onboarding process.

## 7. Open Decisions

## Appendix A: Acronyms

Table 3: Acronyms

| Acronym     | Literal Translation                                     |
|-------------|---------------------------------------------------------|
| <b>API</b>  | Application Programming Interface                       |
| <b>APTC</b> | Advance Payments of the Premium Tax Credit              |
| <b>CCIO</b> | Center for Consumer Information and Insurance Oversight |
| <b>CHIP</b> | Children's Health Insurance Program                     |
| <b>CMCS</b> | Center for Medicaid and CHIP Services                   |
| <b>CMS</b>  | Centers for Medicare & Medicaid Services                |
| <b>CSR</b>  | Cost Sharing Reductions                                 |
| <b>DHS</b>  | Department of Homeland Security                         |
| <b>DSH</b>  | Data Services Hub                                       |
| <b>DSRS</b> | Document Storage and Retrieval System                   |
| <b>EDN</b>  | Eligibility Determination Notice                        |
| <b>EFT</b>  | Enhanced File Transfer                                  |
| <b>EIDM</b> | Enterprise Identity Management                          |
| <b>ESC</b>  | Employer Sponsored Coverage                             |
| <b>ESD</b>  | Eligibility Support Desktop                             |
| <b>ESW</b>  | Eligibility Support Worker                              |
| <b>FFE</b>  | Federally Facilitated Exchange                          |
| <b>IRS</b>  | Internal Revenue Service                                |

| Acronym     | Literal Translation               |
|-------------|-----------------------------------|
| <b>MEC</b>  | Minimum Essential Coverage        |
| <b>MCR</b>  | Marketplace Consumer Record       |
| <b>OC</b>   | Office of Communications          |
| <b>OHI</b>  | Offices of Hearings and Inquiries |
| <b>OTS</b>  | Office of Technology Solutions    |
| <b>QHP</b>  | Qualified Health Plan             |
| <b>RIDP</b> | Remote Identity Proofing          |
| <b>SEP</b>  | Special Enrollment Period         |
| <b>SES</b>  | Standalone Eligibility Service    |
| <b>SLS</b>  | Scalable Login System             |
| <b>SSA</b>  | Social Security Administration    |
| <b>SSN</b>  | Social Security Number            |
| <b>TWS</b>  | Tivoli Workload Scheduler         |
| <b>UI</b>   | User Interface                    |
| <b>VLP</b>  | Verify Lawful Presence            |
| <b>XML</b>  | Extensible Markup Language.       |

## Appendix B: FEPS Class and Attribute Definitions

The document below includes the class and attribute definitions used in the FFE APIs.



Class and Attribute  
Definitions.xlsx

## Appendix C: Application Section Guidelines

Each section of the single, streamlined application must collect the appropriate information to be able to map to the SES APIs and move forward the eligibility determination process. Embedded below is the application questionnaire based on the Paper Reduction Act (PRA) which includes all of the questions and rules to populate each question in the current FFE application. The questionnaire is included in this document to give an overall picture of the application questions; it is not to be used as a requirement document.

In addition, generic guidelines for each application section are documented below. Note this is not an exhaustive list and is not meant to be requirements for each section. Further requirements and guidelines for each question and application section are provided in the UI Question Companion Guide embedded above.



FFM Application  
Questionnaire.pdf

### Contact Information Section

1. Consumers must be given the option to either receive their notices by mail or electronically only (paperless)
2. Consumers must be given the opportunity to provide a mailing address different from their home address.

### Designate Agent/Broker

1. If an Agent/Broker is designated on the application, the Agent/Broker will be notified of application events via an event notification framework.

### Help paying for Coverage Section

3. Consumers must have the option to choose whether or not they would like to be considered for financial assistance. The UI may include a screener to help people decide whether or not to go down that path, but the screener should be optional. Consumers who don't want financial assistance should never be required to input income information to move forward in the application.

4. Consumers who do want to be considered for financial assistance must be considered for Medicaid, CHIP, APTC and CSRs. There is no option to opt-out of Medicaid and CHIP and be considered for APTC and CSRs only.

### **SSNs, Citizenship, and Immigration Status Section**

5. In accordance with 42 CFR 435.907, the collection of SSNs must include notice to consumers about how the SSN will be used and how to get an SSN if someone doesn't have one. Applicants who have an SSN are required to provide it, but there are applicants who may not have SSNs or not be eligible for SSNs and nevertheless they must be allowed to proceed with the application.
6. For "non-applicants," which is the term for individuals who are identified on an application of an individual who is applying for coverage but who are themselves not applying for coverage, collection of SSNs must be optional. In addition, the application may not request citizenship and immigration information from non-applicants.
7. The collection of immigration information is carefully formatted to be able to gather information that can be sent to the Department of Homeland Security (DHS) verification services and processed by the Data Services Hub (DSH).
8. In the citizenship and immigration sections of applications on healthcare.gov, additional questions for consumers about SSNs, citizenship, and immigration status are dynamically queued depending on the response from SSA and DHS through the hub. For example, if an initial verification of an immigration status fails, consumers should be given a second chance to fix their name in case there was a typo or provide their SEVIS ID for a specific immigration status. In addition, if an SSN is verified with SSA, but citizenship is not verified for an attested citizen, the application queues a question about whether the applicant is a naturalized citizen, and if so, proceeds to collect information from the applicant that may assist in verifying naturalized citizenship through DHS verification.

### **Family and Household Section**

9. The application must collect information on whether applicants plan to file federal income tax returns for the coverage year, and if married, whether they will be filing a joint return. The application should also collect information about whether consumers will be claimed as tax dependents by someone else, and whether applicants who are tax filers will claim dependents of their own. The SES eligibility logic needs full information about the tax household in order to determine eligibility for APTC.
10. In addition, in some cases additional information about family relationships and who is living together will be necessary for Medicaid and CHIP determinations. For example, for all children applying for coverage, the eligibility logic will need to know whether the child lives with their parents. If the child is claimed as a tax dependent by someone who is not their parent, or a non-custodial parent, the application may need to collect information both about the tax household and about the parents and siblings living with the child in order to process both Medicaid/CHIP and APTC eligibility.

---

## Questions to screen for Medicaid eligibility Section

11. Questions related to disability, assistance with daily living, foster care, and American Indian/Alaska Native status have been carefully worded to accurately assess potential Medicaid and CHIP eligibility and may not be substantively altered.

## Income Section

12. The application must collect information about both current monthly as well as annual income. Income amounts must be associated with a specific person on the application and a specific income type, rather than documented as general household income, because in some cases SES logic will use system logic to determine how income is counted and how it effects the eligibility determination.
13. The income section may be prepopulated with income information available through a trusted data source, such as Equifax data available through the Data Services Hub, as long as appropriate privacy and security safeguards are in place, such as identity proofing of the consumer whose income is displayed. However, the UI must always allow the consumer to edit or delete the prepopulated information.

## Program Specific Questions Section

14. Some questions on the application are only appropriate to ask consumers who appear eligible for APTC, and some questions are only appropriate to ask consumers who appear eligible for Medicaid and CHIP. The UI must call Update App before asking detailed questions about employer-sponsored coverage offers and only request information of applicants when relevant to their eligibility.
15. It is possible that an applicant who is preliminary Medicaid or CHIP is found ineligible after the program specific questions. In that case, the applicant may be asked the APTC specific questions.

## Special Enrollment Periods Section

16. Special Enrollment Period questions must be asked of all applicants who are QHP eligible, even during open enrollment.
17. For questions that may trigger special enrollment period verification issues (SVIs) the application UI must include language providing notice to the applicant that they may need to provide documents to verify their special enrollment period eligibility.

## Review, Sign, & Submit Section

18. Consumers should have an opportunity to review the information they have provided and make edits prior to submission.
19. Consumers are legally required to agree with a number of attestations in order to be found eligible for health coverage.
20. Some attestations are only relevant for Medicaid and/or CHIP and only should be displayed when consumers will be found eligible for those programs.

---

## **Eligibility Results Section**

21. It is critical that the eligibility results provide information on the programs each applicant is eligible for, as well as whether more information is needed in order to confirm eligibility or resolve data matching issues.
22. Consumers who are found ineligible for Medicaid should have the opportunity for their information to be sent to the Medicaid agency anyway, for the state to make a full determination.