

# HROS JEDx API Proposal

<b>Overview.....</b>	<b>3</b>
What is JEDx?.....	3
JEDx API Development Approach.....	3
Data Standards.....	3
Technology Standards.....	4
The Overall Model.....	4
This Proposal.....	5
<b>JEDx System Context, Component and Interfaces.....</b>	<b>6</b>
The T3 Initiative JEDx Context.....	6
System Context.....	6
Components & Interfaces.....	11
Organization Manager.....	12
Worker Manager.....	13
Worker Compensation Report Manager.....	14
Worker Paid Hours Report Manager.....	15
Component Data Model.....	17
Component Collaborations.....	17
Manage Organization (Employer).....	18
Register Organization.....	18
Retrieve a Single or Multiple Organizations.....	19
Maintain a Single Organization.....	20
Maintain Multiple Organizations.....	21
Register and Maintain Worker.....	22
Register Worker.....	22
Retrieve a Single or Multiple Workers.....	23
Maintain a Single Worker.....	24
Maintain Multiple Workers.....	25
Manage Worker Compensation Report.....	26
Post Worker Compensation Report.....	26
Retrieve a Single or Multiple Worker Compensation Reports.....	27
Maintaining a Single Worker Compensation Report.....	28
Maintain Multiple Worker Compensation Reports.....	29
Manage Worker Paid Hours Report.....	30
Post Worker Paid Hours Report.....	30

Retrieve a Single or Multiple Worker Paid Hours Reports.....	31
Maintaining a Single Worker Paid Hours Report.....	32
Maintain Multiple Worker Paid Hours Reports.....	33
<b>API Serializations.....</b>	<b>34</b>
HR-Open API Serializations.....	35
JEDx API Serializations.....	35
<b>Appendices.....</b>	<b>36</b>
References.....	36
Document Map of this Proposal.....	36
JEDx System Context, Components and Interfaces.....	36
API Serializations.....	36
HR-Open API Serialization.....	36
JEDx API Serialization.....	37

# Overview

This is a brief description of the proposal and its intent and what it includes and does not include.

## What is JEDx?

The Jobs and Employment Data Exchange ([JEDx](#))<sup>1</sup> is a public-private, standards-based initiative to promote the consistent sharing and use of data on jobs and employment.

JEDx aims to standardize and improve the sharing and use of employment data through public-private collaboration. It involves data and technology standards, particularly Application Programmable Interfaces ([API](#)) standards<sup>2</sup>, for better data sharing and protection.

This project's current focus is on improving employer data reporting to government agencies, starting with state unemployment insurance reporting, and enhancing job description and posting data sharing for improved job searches.

Future phases will include empowering workers and learners to use their Learning and Employment Records (LER) for job applications and government programs, supporting lifelong learning and workforce retraining.

A comprehensive data dictionary has been developed, and a roadmap for JEDx's implementation has been launched, including partnerships with a number of states.

The project seeks to use existing standards where possible to avoid creating redundant systems. The initial focus is on government reporting due to its complexity and the need for a standardized approach across multiple states and entities. It will emphasize privacy and plans to build an infrastructure that supports current and future data exchange needs, including employment records which are part of learning and employment records issued as verifiable credentials, without limiting to predefined objects.

This proposal is provided for for public comment (July 2024), with a candidate release by scheduled in late-summer (August/September 2024), followed by a proof of concept in the fall (October/November 2024).

## JEDx API Development Approach

### Data Standards

Data standards provide a data model and definitions (e.g., JEDx Data Dictionary) for the consistent organization and specification of jobs and employment data.

---

<sup>1</sup> <https://www.uschamberfoundation.org/solutions/workforce-development-and-training/jedx>

<sup>2</sup> <https://aws.amazon.com/what-is/api/>

- **Data vocabulary/dictionary.** Standard meanings and allowed values for data elements. A standard data dictionary solves the problem of understanding the data. When the source system, collection system, and reporting and analytics systems all utilize a standard data dictionary, it prevents misinterpretation of meaning and enhances data quality. .

A sample list Data Elements included in the standard data dictionary include the following:

- Employer
  - Work location
  - Hours Worked
  - Direct Compensation
  - Indirect Compensation
  - Job Description
  - Job Role
  - Job Duties
- **Data model.** Entity definitions and entity relationships. For example, a data model could show relationships between a worker, employer, job, and job competencies logically modeled in the data. A standard data model defines a common way of organizing data. It defines entities, such as an employee, and determines which data elements are considered properties of the “employee”, such as the first name.

## Technology Standards

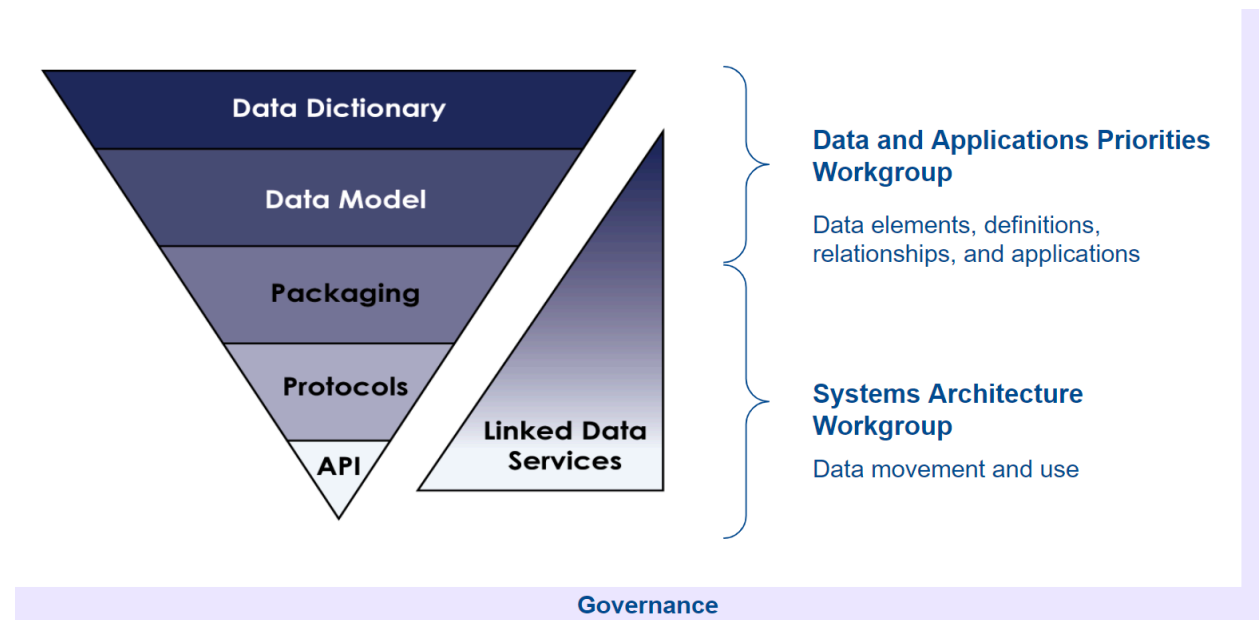
Technology standards (e.g., API standards) provide the basis for improved data sharing, management, and protection while improving access and use.

## The Overall Model

JEDx promotes the development and use of data and technology standards to improve how employers report data to government agencies, starting with:

- State Unemployment Insurance (UI) reporting, and the access and utilization of this data;
- Enhancing the sharing and usage of job description data to track and manage career pathways through education and training partnerships;
- Improving the posting and application of job data to facilitate more effective job searches; and
- Empowering workers and learners to utilize their own Learning and Employment Records (LER) when applying for jobs and government programs.

This is the abstract model for the tiers of standards and structure being used to frame this project:



## This Proposal

This proposal focuses on the Application Programming Interface (API) and Protocol standards needed to move and interoperate the data and information needed to empower the business and use cases being taken on.

This proposal will also lay out the role that APIs play in modernizing data exchange and how API standards can be developed and pilot-tested to modernize how employers report data to government agencies.

There are several components that will make up this proposal:

1. This overview document
2. A technical document laying out the use cases, component models, and the explanation of the serialization paths.
3. Technical specifications that outline two pathways for serialization that HR Open intends to use for the development of the standard.
  - a. Serialization is how data is structured and designed to move over the wires between one computer and application and another computer and application.

This standard will also provide input into the development of an improved occupational autocoding process and system that can be tested and implemented.

# JEDx System Context, Component and Interfaces

## The T3 Initiative JEDx Context

The JEDx API project<sup>3</sup> run by the [T3 Initiative Network](#), and sponsored by the US Chamber of Commerce, provides the operational framework for the provision and consumption of JEDx APIs that are scoped to the specific project at hand. The JEDx initiative identified several initial projects. The context outlined here, is scoped to the first project, Improving Federal and State Reporting. In this project, states, federal agencies, employers, HR technology service providers, workforce analytics providers, and other public and private partners will collaborate to reduce costs and create greater value for employer reporting to federal and state agencies. This project will start with the enhancement of state unemployment insurance (UI) reporting. The project's primary objectives include:

- Reducing federal and state reporting costs for employers and government agencies
- Improving data quality and timeliness
- Providing better data for public and private workforce analytics and program administration applications

Secondary objectives, while taking into account the other JEDx projects, aim to ensure that employers and HR service providers adopt a consistent approach in:

- Sharing and using job description data in career pathways (Project 2)
- Improving job posting data for search (Project 3)
- Providing workers/learners with their own LERs for job applications and government programs and benefits (Project 4)

## System Context

The System Context diagram, presented below, provides a descriptive representation of the JEDx API Context. More generally, the purpose of any System Context diagram is to provide a focused view of three things:

- the system boundary,
- the external entities that interact with the system, and
- the nature of the interaction or communication between entities and the system.

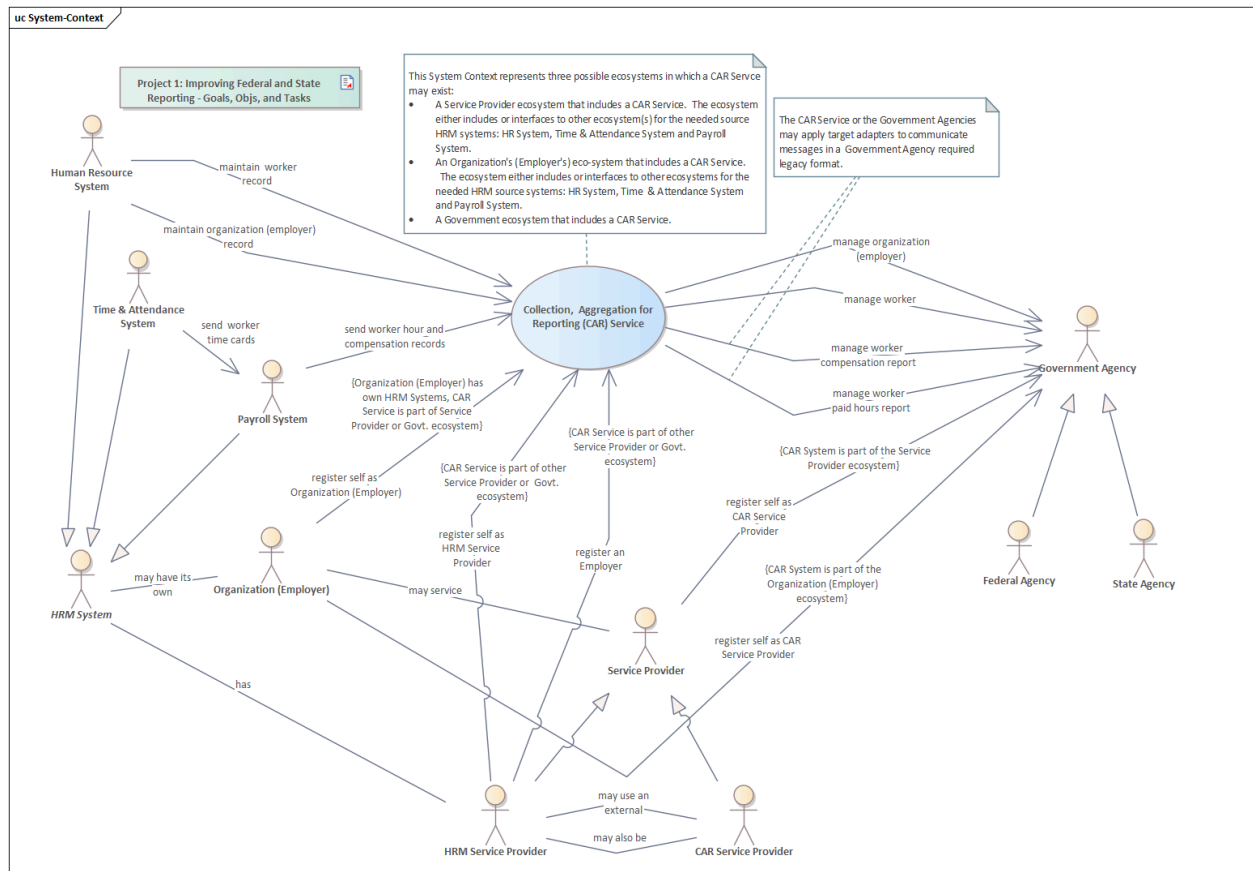
The system is represented as a big circle. Surrounding the system are the external entities, represented as stick figures. These external entities could be either systems or individuals. The directed lines connecting the external entities and the system signify an interaction and/or information exchange between them; they imply and could potentially map to use cases.<sup>4</sup> The

---

<sup>3</sup> <https://www.uschamberfoundation.org/solutions/workforce-development-and-training/jedx>

<sup>4</sup> (Armour et al.) Armour, F., Miller, G. Advanced Use Case Modeling. Addison-Wesley. 2001.

arrow indicates the direction or flow of the interaction and/or information, from the initiator to the recipient. The non-directed lines between two external entities describe a relationship between them.



This System, that is the subject of the System Context diagram, is referred to as the **Collection, Aggregation for Reporting (CAR) Service**. The System Context diagram considers and represents three possible ecosystems in which a CAR Service may exist:

- A **Service Provider ecosystem** that includes a CAR Service. The ecosystem either includes or interfaces to other networks for the required source Human Resource Management (HRM) systems: Human Resource (HR) System, Time & Attendance (TA) System and Payroll (PR) System.
- An **Organization's (Employer's) eco-system** that includes a CAR Service. The ecosystem either includes or interfaces to other networks for the needed HRM source systems: Human Resource (HR) System, Time & Attendance (TA) System and Payroll (PR) System.
- A **Government ecosystem** that includes a CAR Service.

The interactions initiated from the external entities to the CAR Service include:

- **Human Resource System** manages employer, worker, work relationship and work assignment.
  - *Example: Maintain worker record* - This record includes worker-related information the person, organization (employer) identification for the worker, work relationship, and work assignment information needed to satisfy requirements on the WorkerCompensationReport and the WorkerPaidHoursReport.
  - *Example: Maintain organization (employer) record* - This record includes organization (employer)-related information needed to satisfy requirements on the WorkerCompensationReport and the WorkerPaidHoursReport.
- **Time & Attendance System** manages schedules, hours worked and time off; it is critical for payroll accuracy and compliance.
  - *Example: Send worker time cards* - This includes worker time card information needed to satisfy requirements on the WorkerPaidHoursReport.
- **Payroll System** processes payroll including compensating employees for their work. It involves calculating total wage earnings, withholding deductions, filing payroll taxes and delivering payment.
  - *Example: Send worker hour and compensation records* - These records include worker earnings needed to satisfy data requirements for the worker compensation and worker paid hours records.
- **Organization (Employer)** is a legal organization that controls and directs a worker under an express or implied work agreement and pays the worker a salary or wages (e.g., the Employer of an Employee).
  - *Example: Register self as Organization (Employer)* -
    - Preconditions:
      1. The CAR Service is part of the Service Provider or Govt. ecosystem.
      2. Organization (Employer) has its own HRM System that registers itself with the CAR Service, specifically as an Employer. Note, the Employer may already be registered with another Service Provider or Govt. Agency. The CAR Service must know the Organizations (Employers) consuming its services. Also, the CAR Service must know the Organization (Employer) population on which it must provide government reporting.
  - *Example: Register self as CAR Service Provider* -
    - Preconditions:
      1. The CAR Service is incorporated within a Service Provider ecosystem.

A Service Provider, which could be either an HRM Service Provider or a dedicated CAR Service Provider, registers itself with a Govt. Agency, specifically as a Service Provider and as a CAR Service Provider. The registration must also include the CAR Service services (i.e., applications) that will be communicating with the Government Agency.
- **Service Provider** - A Service Provider works on-behalf of its clients (Organizations (Employers) or other Service Providers). A Service Provider may or may not include a



CAR Service. There are two types of Service Providers: **HRM Service Provide** and **CAR Service Provider**.

- *Example: Register self as CAR Service Provider -*
  - Preconditions:
    1. The CAR Service is part of the Service Provider ecosystem. A Service Provider, which could be either an HRM Service Provider or a dedicated CAR Service Provider, registers itself with a Govt. Agency, specifically as a CAR Service Provider. The registration must also include the CAR Service services (i.e., applications) that will be communicating with the Government Agency.

The interactions initiated from the CAR Service to the Government Agency include:

- **Government Agency** - A Government Agency within a government has responsibility for administering and enforcing a certain domain(Wikipedia). In this context, the term 'Government Agency' is used to refer to those agencies that have state UI reporting obligations. There are two types of Government Agencies: **Federal Agency** and **State Agency**.
  - *Example: Register and maintain organization (employer) -*
    - Preconditions:
      1. The CAR Service that manages the Organization is registered. The CAR Service registers the Organization for which it will be providing worker reports. The CAR Service ensures the Organization's information is up to date.
  - *Example: Register and maintain worker -*
    - Preconditions:
      1. The Organization (Employer) is registered. The CAR Service registers the Worker for which it will be providing worker records. The CAR Service ensures the Worker information is up to date.
  - *Example: Manage worker compensation report -*
    - Preconditions:
      1. The Organization (Employer) is registered.
      2. The Worker is registered.The CAR Service sends the worker compensation report, in accordance with the stipulated time period reporting requirements.
  - *Example: Manage worker paid hours report -*
    - Preconditions:
      1. The Organization (Employer) is registered.
      2. The Worker is registered.The CAR Service sends the worker paid hours report, in accordance with the stipulated time period reporting requirements.

The relationships between two external entities include:

- An **Organization (Employer)** *may have its own HRM System(s)* A **Service Provider** *may service or provide services to an Organization (Employer)* An **HRM Service Provider** operates and manages an **HRM System**.
- An **HRM Service Provider** *might engage an external CAR Service Provider*.
- An **HRM Service Provider** *may also be CAR Service Provider*.
- **Third Party Agents** may be included in this process involving any of these entities. They could also function as a **CAR Service Provider**.

## Components & Interfaces

This JEDx API Proposal provides API Specifications that address a subset of the interactions described in the System Context, specifically, the interactions initiated from the CAR Service to the Government Agency:

- *Register and maintain organization (employer)*
- *Register and maintain worker*
- *Manage worker compensation report*
- *Manage worker paid hours report*

This section provides a Component Specification that outlines four functional components and the interfaces they provide to facilitate these information exchanges. A Component Specification diagram is shown for each of the four functional components and their interfaces. In each of the diagrams, the classifier with the “Mgr” suffix represents the functional component; the classifier with the “Mgt” suffix represents the component’s offered interface. Since these interfaces are to be realized as RESTful Web APIs, the naming of the interfaces’ operations is in accord with the corresponding HTTP methods, specifically:

- The POST method indicates the create operation.
- The PUT method indicates the update operation.
- The DELETE method indicates the delete operation.
- The GET method indicates the read operation.

The following format describes the interface operation signatures:

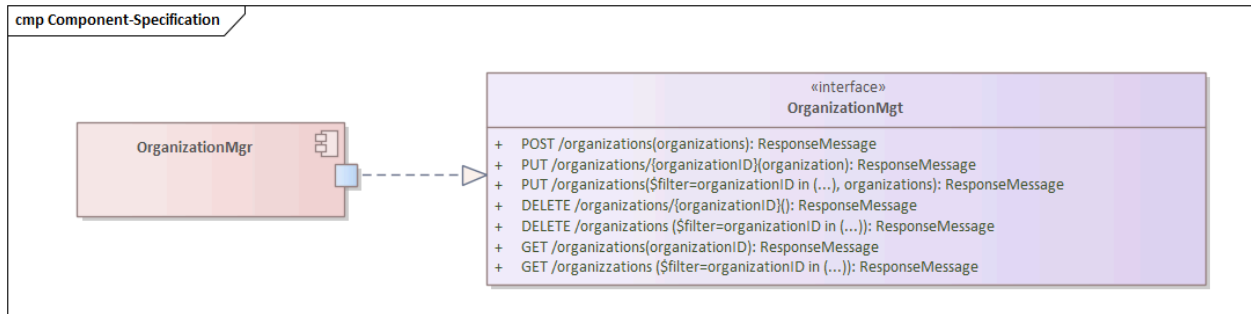
**OperationSignature = *OperationName* *ClassParameter* “:” *ReturnType***  
*where,*  
***OperationName* = *HTTPMethod* *ResourcePath* [“?” *ResourceQuery* ]**  
***HTTPMethod* = “DELETE” | “GET” | “HEAD” | “OPTIONS” | “PATCH”**  
**| “POST” | “PUT”**  
***ResourcePath* = \*(“/” *PathSegment* “/”) | [“{” *templateExpression* “}”] |**  
**\*(“/” *PathSegment* “/”)**  
***ClassParameter* = (“ *ClassParameterName* “)”**

For example, the operation signature: “POST /organizations/{organizationID}(organization): ResponseMessage” realizes the format as follows:

- ***HTTPMethod*** = “POST”
- ***ResourcePath*** = “/organizations/{organizationID}”
- ***ClassParameter*** = (organization)
- ***ReturnType*** = “ResponseMessage”

## Organization Manager

The OrganizationMgr component has responsibility for managing the organization-related data. It offers an OrganizationMgt interface for external entities to submit requests to create, update and delete organizations.



The interface comprises four operations (in order shown in the diagram, above):

- A POST operation on the organizations collection resource creates one or more organization instance resources in the collection.
- A PUT operation on an organization instance resource updates (replaces) that organization instance resource in the collection.
- A PUT operation on the organizations collection resource updates (replaces) all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being replaced to a subset of the collection. In the former case, when all the instance resources in the collection are updated, the absence of an instance resource implies a deletion of that resource.
- A DELETE operation on an organization instance resource removes that organization instance resource in the collection.
- A DELETE operation on the organizations collection resource may delete all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.
- A GET operation on an organization instance resource retrieves that organization instance resource in the collection.
- A GET operation on the organizations collection resource may retrieve all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.

Any of the operations may return a response message that includes:

- The HTTP status-line (HTTP version number, response status code, response status description)
- Message headers,

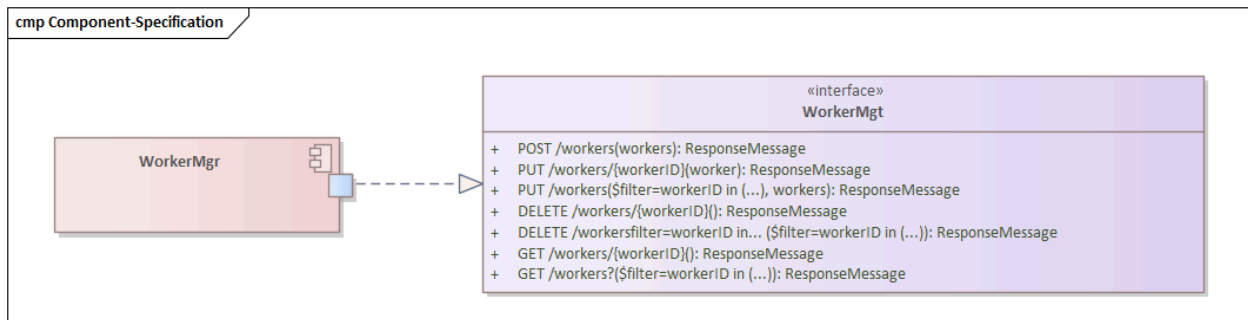
- An optional Message-body which may include a Resource Representation and/or a Confirmation Message to convey success, warning and error messages.

## Notes:

- OrganizationID is assigned by GovernmentAgency and returned to the submitter.

## Worker Manager

The WorkerMgr component has responsibility for managing the worker-related data. It offers an WorkerMgt interface for external entities to submit requests to create, update and delete workers.



The interface comprises five operations (in order shown in the diagram, above):

- A POST operation on the workers collection resource creates one or more worker instance resources in the collection.
- A PUT operation on a worker instance resource updates (replaces) that worker instance resource in the collection.
- A PUT operation on the workers collection resource updates (replaces) all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being replaced to a subset of the collection. In the former case, when all the instance resources in the collection are updated, the absence of an instance resource implies a deletion of that resource.
- A DELETE operation on a worker instance resource removes that worker instance resource in the collection.
- A DELETE operation on the workers collection resource may delete all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.
- A GET operation on a worker instance resource retrieves that organization instance resource in the collection.
- A GET operation on the worker collection resource may retrieve all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.

Any of the operations may return a response message that includes:

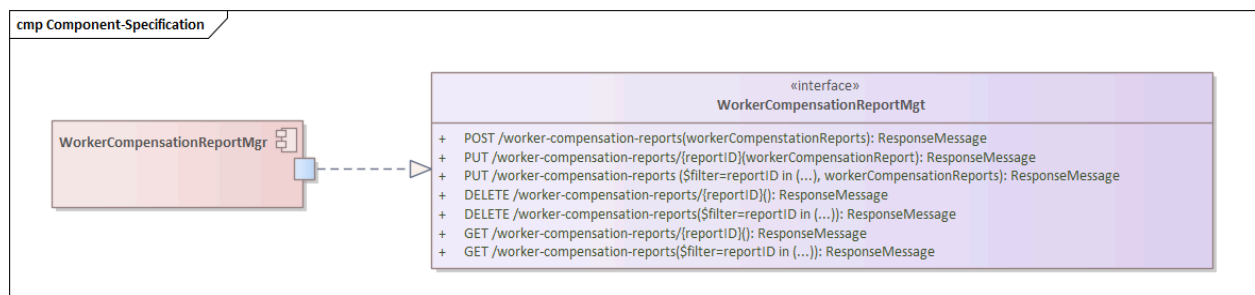
- The HTTP status-line (HTTP version number, response status code, response status description)
- Message headers,
- An optional Message-body which may include a Resource Representation and/or a Confirmation Message to convey success, warning and error messages.

Notes:

- WorkerID is assigned by GovernmentAgency and returned to the submitter.
- OrganizationID(s) is/are included in the Worker instance.

## Worker Compensation Report Manager

The WorkerCompensationReportMgr component has responsibility for managing the worker compensation report-related data. It offers an WorkerCompensationReportMgt interface for external entities to submit requests to create, update and delete worker compensation reports.



The interface comprises five operations (in order shown in the diagram, above):

- A POST operation on the worker-compensation-reports collection resource creates one or more worker-compensation-report instance resources in the collection.
- A PUT operation on an worker-compensation-report instance resource updates (replaces) that worker-compensation-report instance resource in the collection.
- A PUT operation on the worker-compensation-reports collection resource updates (replaces) all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being replaced to a subset of the collection. In the former case, when all the instance resources in the collection are updated, the absence of an instance resource implies a deletion of that resource.
- A DELETE operation on an worker-compensation-report instance resource removes that worker-compensation-report instance resource in the collection.
- A DELETE operation on the worker-compensation-report collection resource may delete all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.

- A GET operation on a worker-compensation-reports instance resource retrieves that worker-compensation-reports instance resource in the collection.
- A GET operation on the worker-compensation-reports collection resource may retrieve all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.

Any of the operations may return a response message that includes:

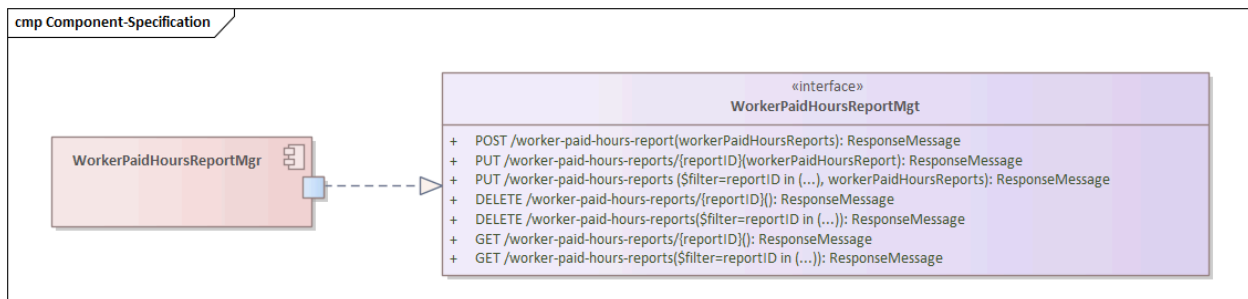
- The HTTP status-line (HTTP version number, response status code, response status description)
- Message headers,
- An optional Message-body which may include a Resource Representation and/or a Confirmation Message to convey success, warning and error messages.

Notes:

- WorkerCompensationReportID is assigned by the GovernmentAgency and returned to the submitter.
- WorkerID is included in the Report instance.

## Worker Paid Hours Report Manager

The WorkerPaidHoursReportMgr component has responsibility for managing the worker paid hours report-related data. It offers an WorkerPaidHoursReportMgt interface for external entities to submit requests to create, update and delete worker compensation reports.



The interface comprises five operations (in order shown in the diagram, above):

- A POST operation on the worker-paid-hours-reports collection resource creates one or more worker-compensation-report instance resources in the collection.
- A PUT operation on an worker-paid-hours-report instance resource updates (replaces) that worker-paid-hours-report instance resource in the collection.
- A PUT operation on the worker-paid-hours-report collection resource updates (replaces) all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being replaced to a subset of the collection. In the former case, when all the instance

resources in the collection are updated, the absence of an instance resource implies a deletion of that resource.

- A DELETE operation on an worker-paid-hours-report instance resource removes that worker-paid-hours-report instance resource in the collection.
- A DELETE operation on the worker-paid-hours-report collection resource may delete all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.
- A GET operation on a worker-paid-hours-reports instance resource retrieves that worker-compensation-reports instance resource in the collection.
- A GET operation on the worker-paid-hours-reports collection resource may retrieve all instance resources in the collection; it is used in the request message of a bulk operation. A URI query component may serve to limit the instance resources being deleted to a subset of the collection.

Any of the operations may return a response message that includes:

- The HTTP status-line (HTTP version number, response status code, response status description)
- Message headers,
- An optional Message-body which may include a Resource Representation and/or a Confirmation Message to convey success, warning and error messages.

Notes:

- WorkerPaidHoursReportID is assigned by the GovernmentAgency and returned to the submitter.
- WorkerID is included in the Report instance.



## Component Data Model

A logical data model that underlies the component interface information models can be accessed here:

[https://drive.google.com/file/d/1aO3ArxOGwSae3Vy0Lzr8uEmkgYnALs4U/view?usp=drive\\_link](https://drive.google.com/file/d/1aO3ArxOGwSae3Vy0Lzr8uEmkgYnALs4U/view?usp=drive_link)

## Component Collaborations

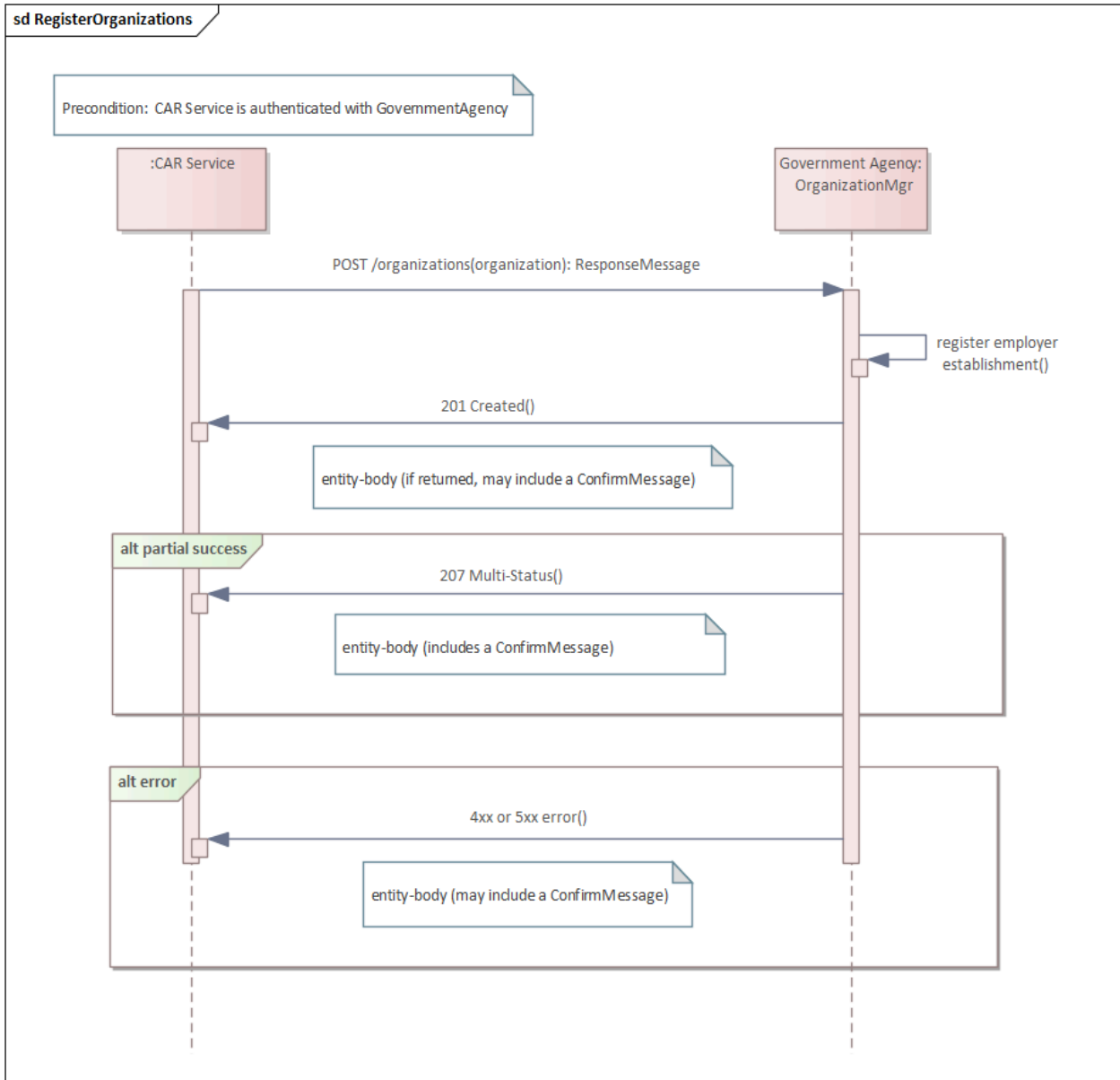
Component collaborations illustrate the message exchange and sequencing between collaborating components to realize interactions identified in the System Context. As noted in the previous section, this proposal addresses a subset of those interactions specifically, the interactions initiated from the CAR Service to the Government Agency:

- *Register and maintain organization (employer)*
- *Register and maintain worker*
- *Manage worker compensation report*
- *Manage worker paid hours report*

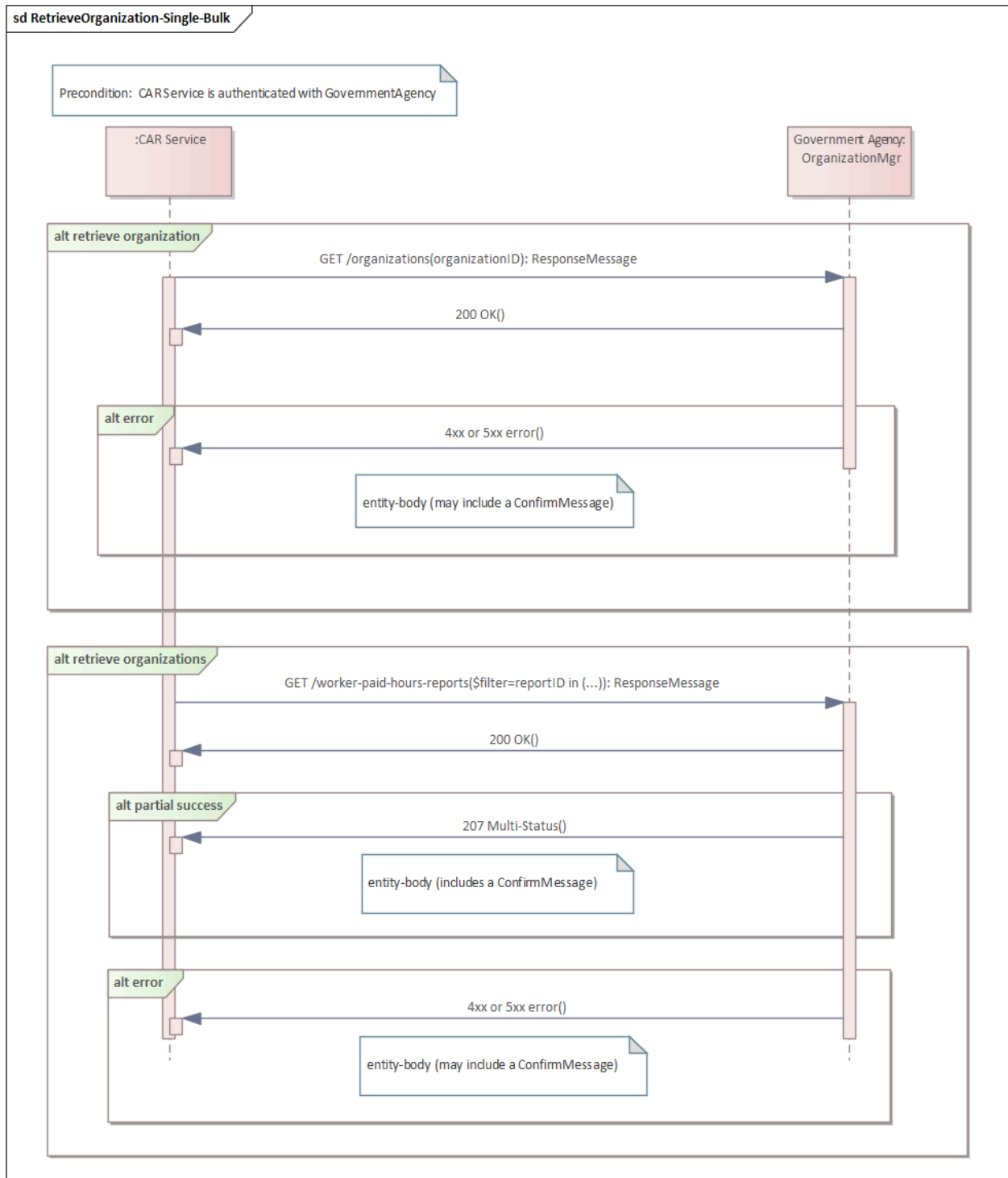
The following collaborations leverage the components and interfaces as defined in the previous section.

## Manage Organization (Employer)

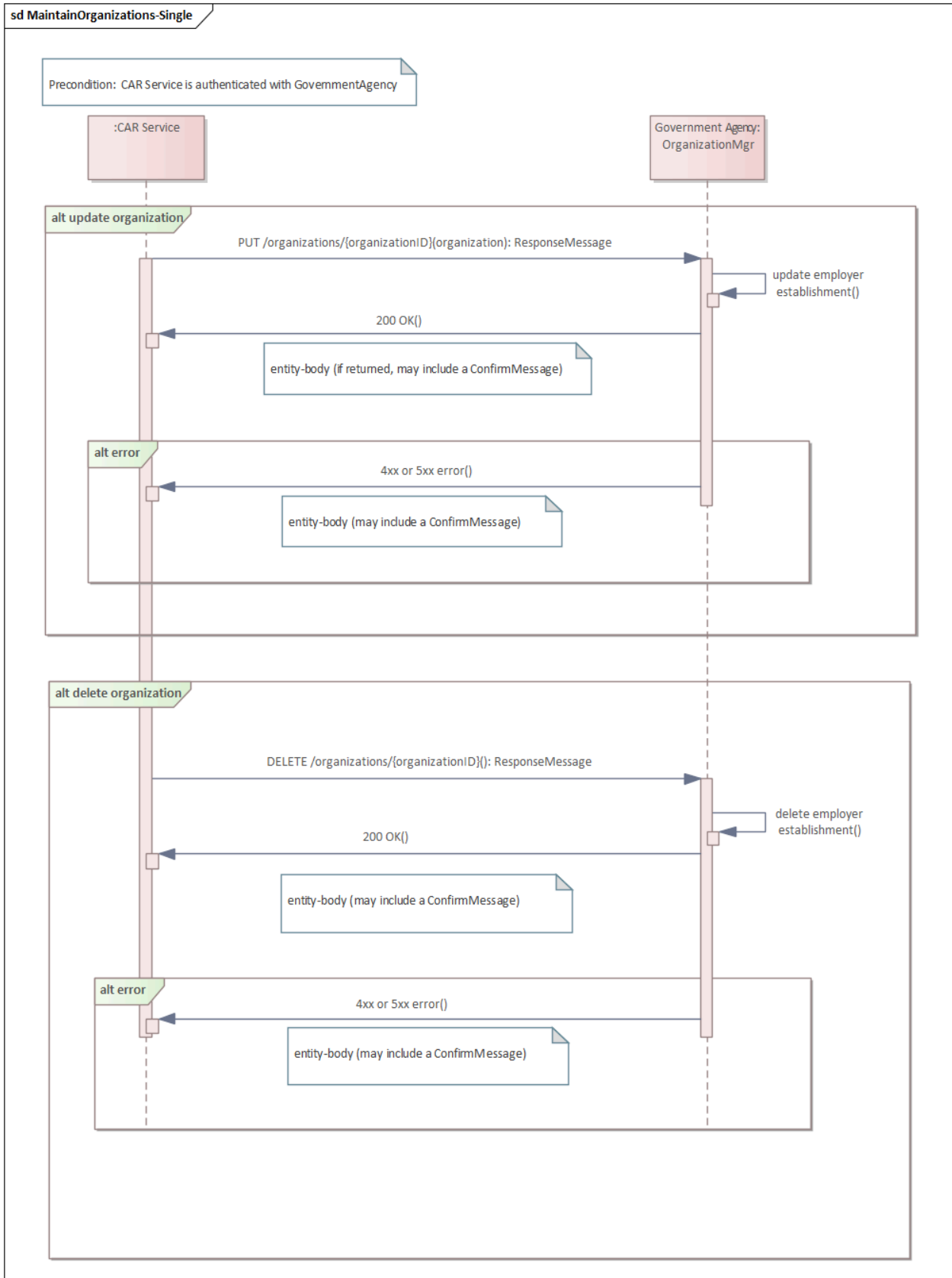
### Register Organization



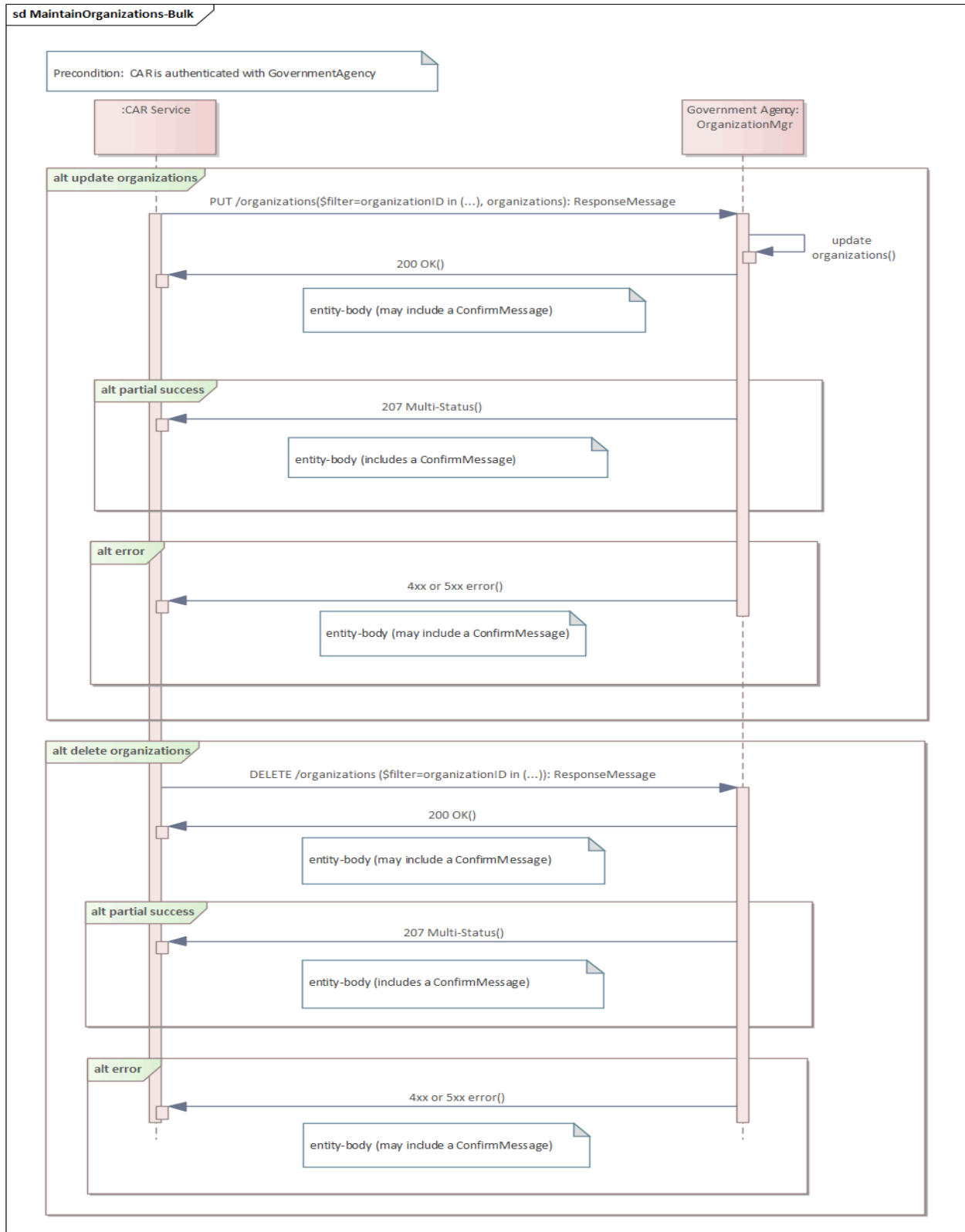
## Retrieve a Single or Multiple Organizations



## Maintain a Single Organization

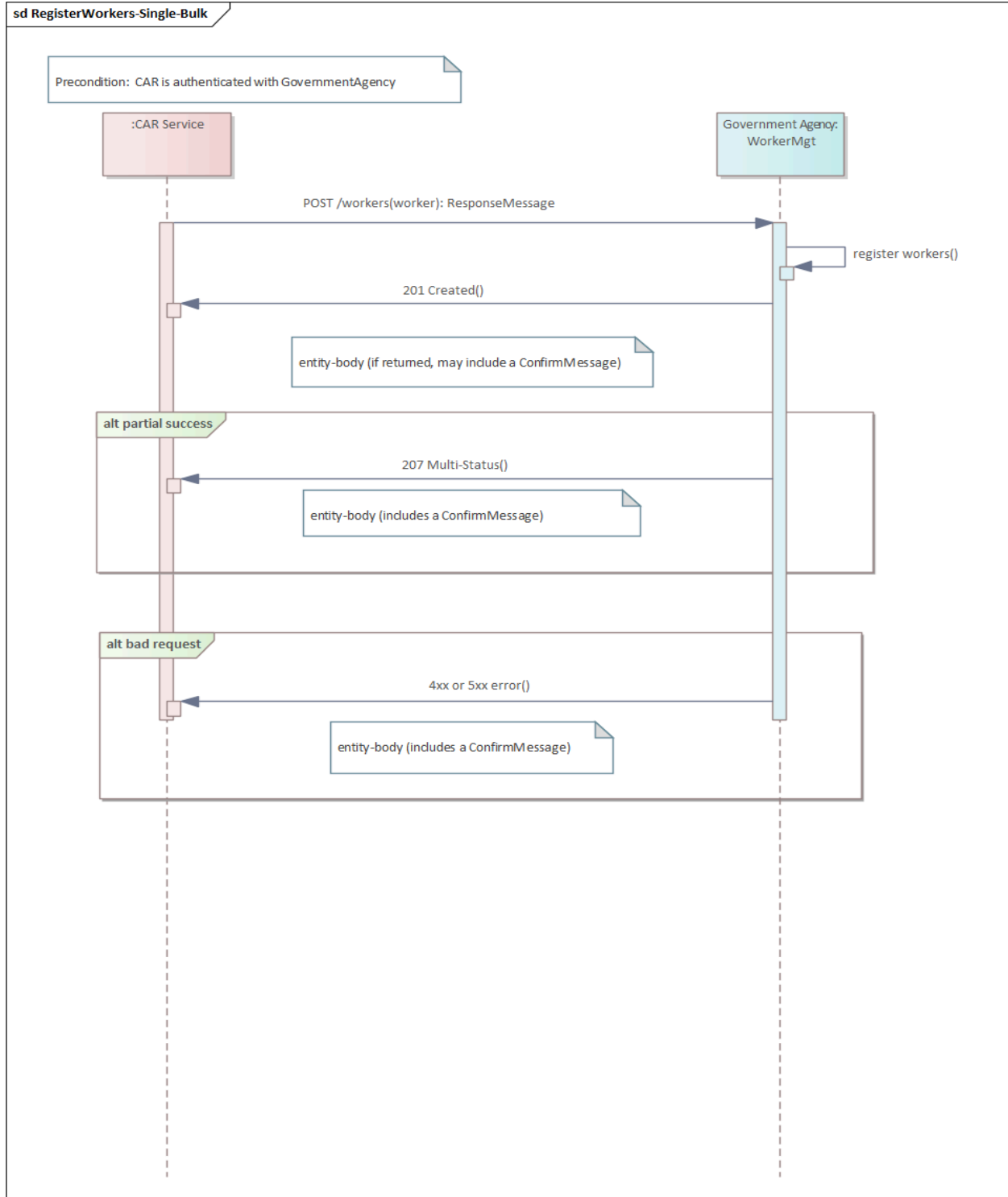


## Maintain Multiple Organizations

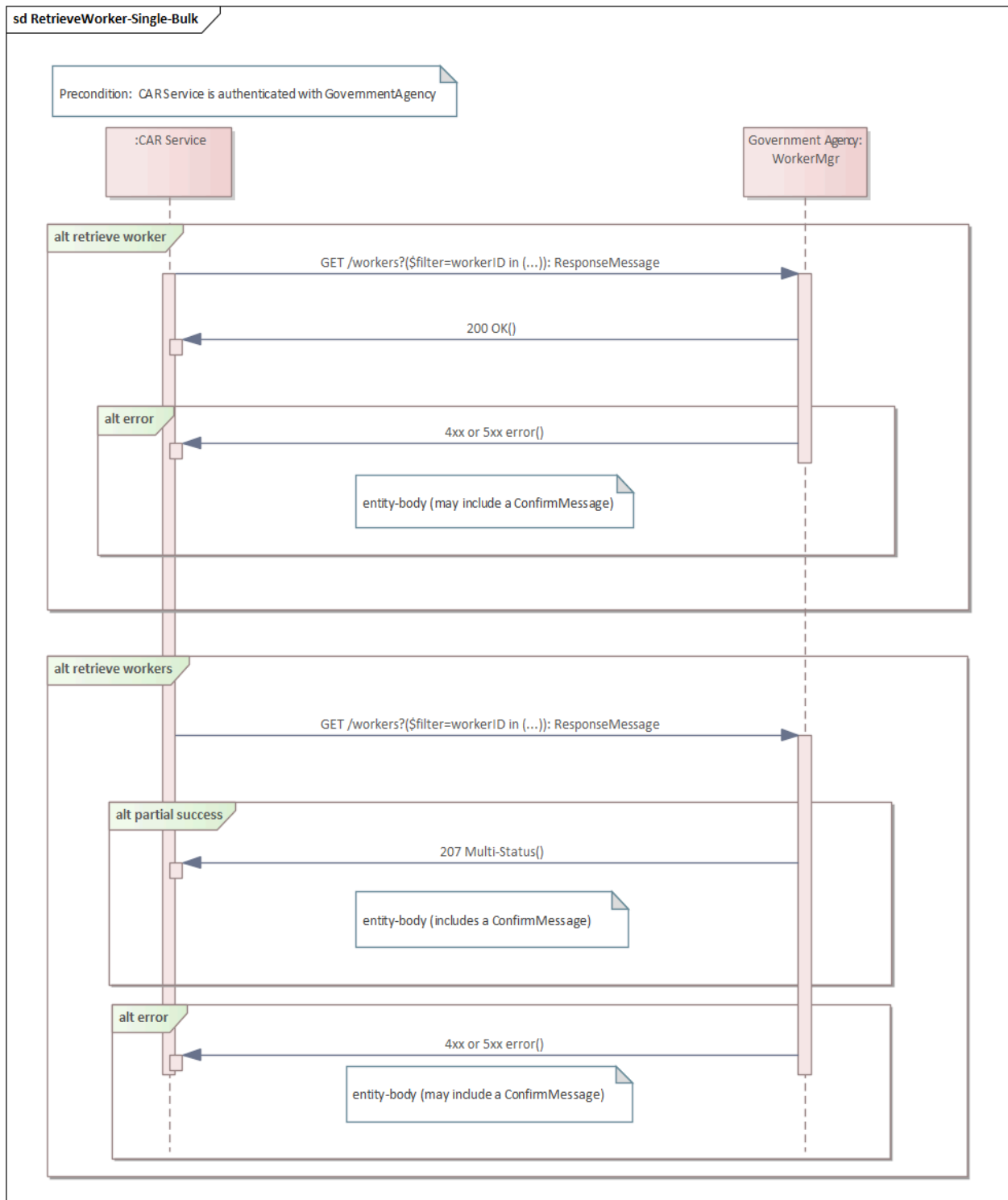


## Register and Maintain Worker

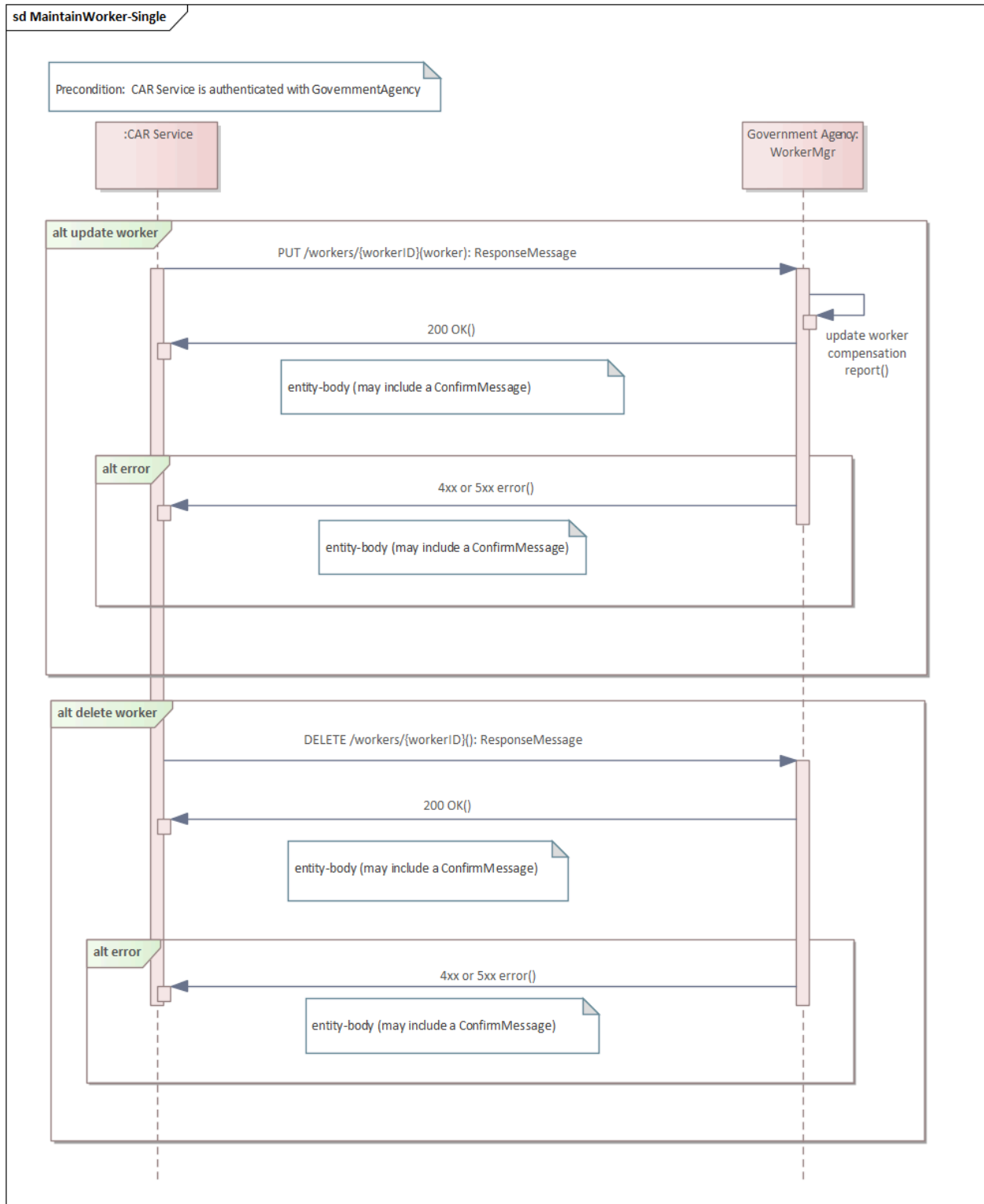
### Register Worker



## Retrieve a Single or Multiple Workers

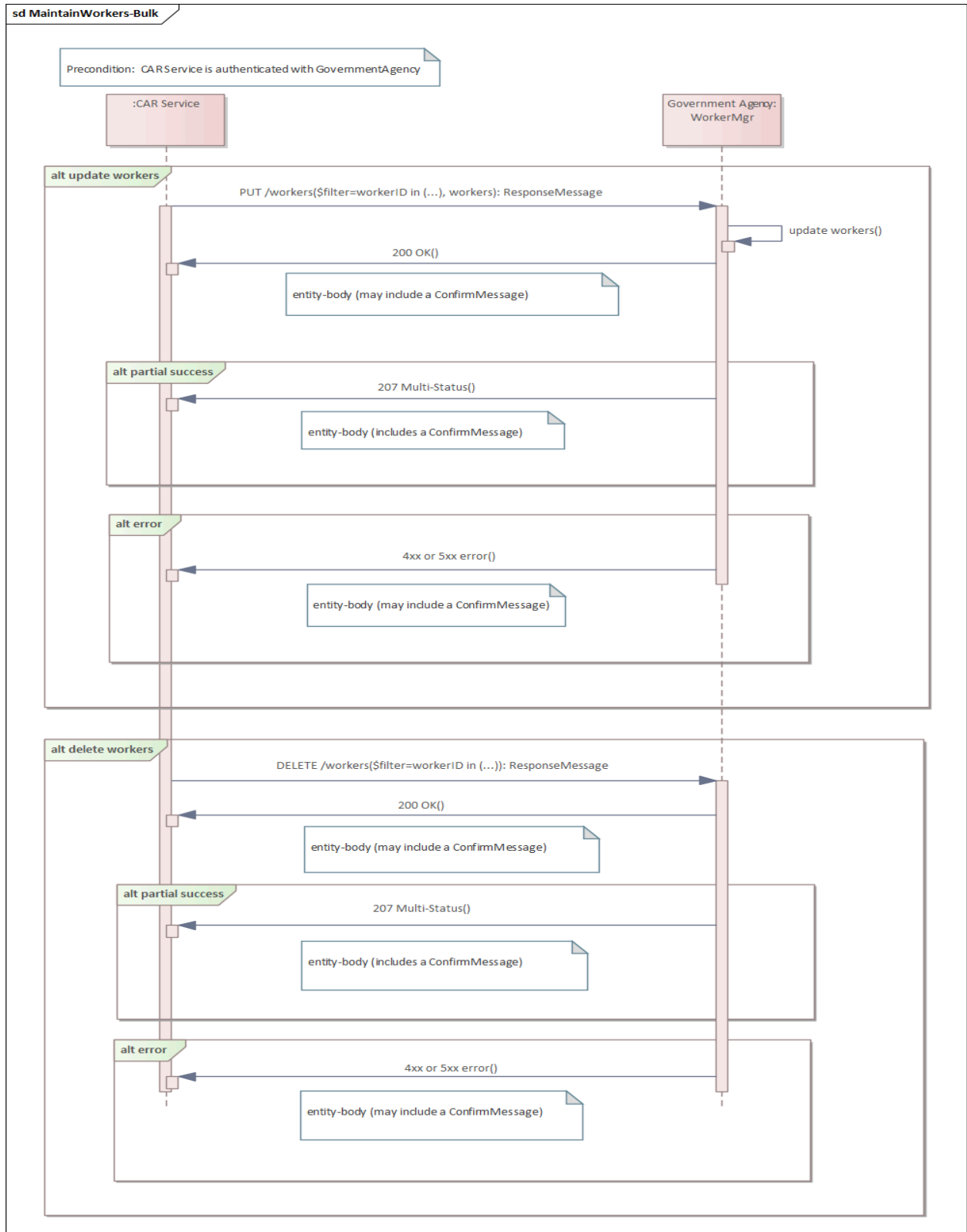


## Maintain a Single Worker



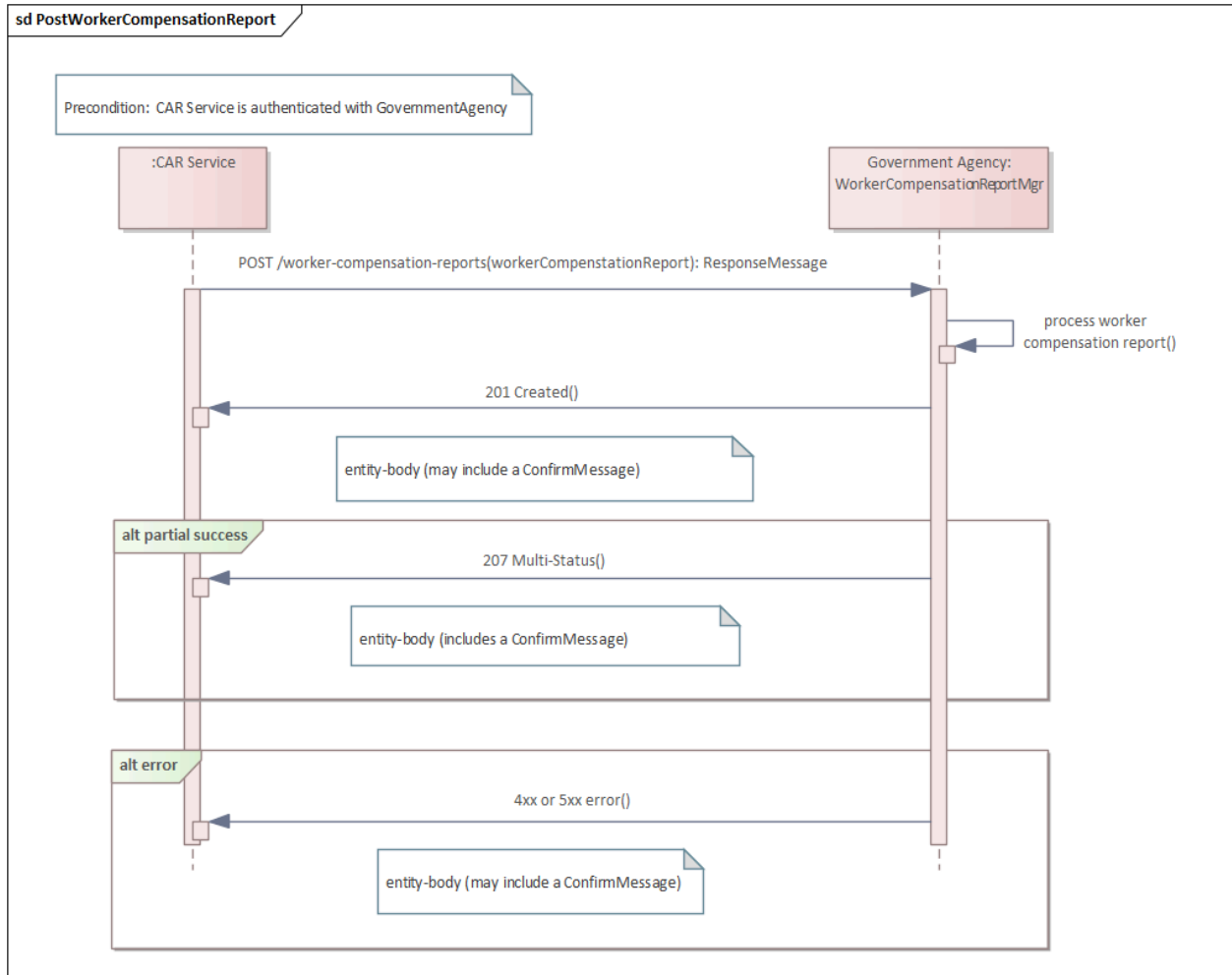


## Maintain Multiple Workers

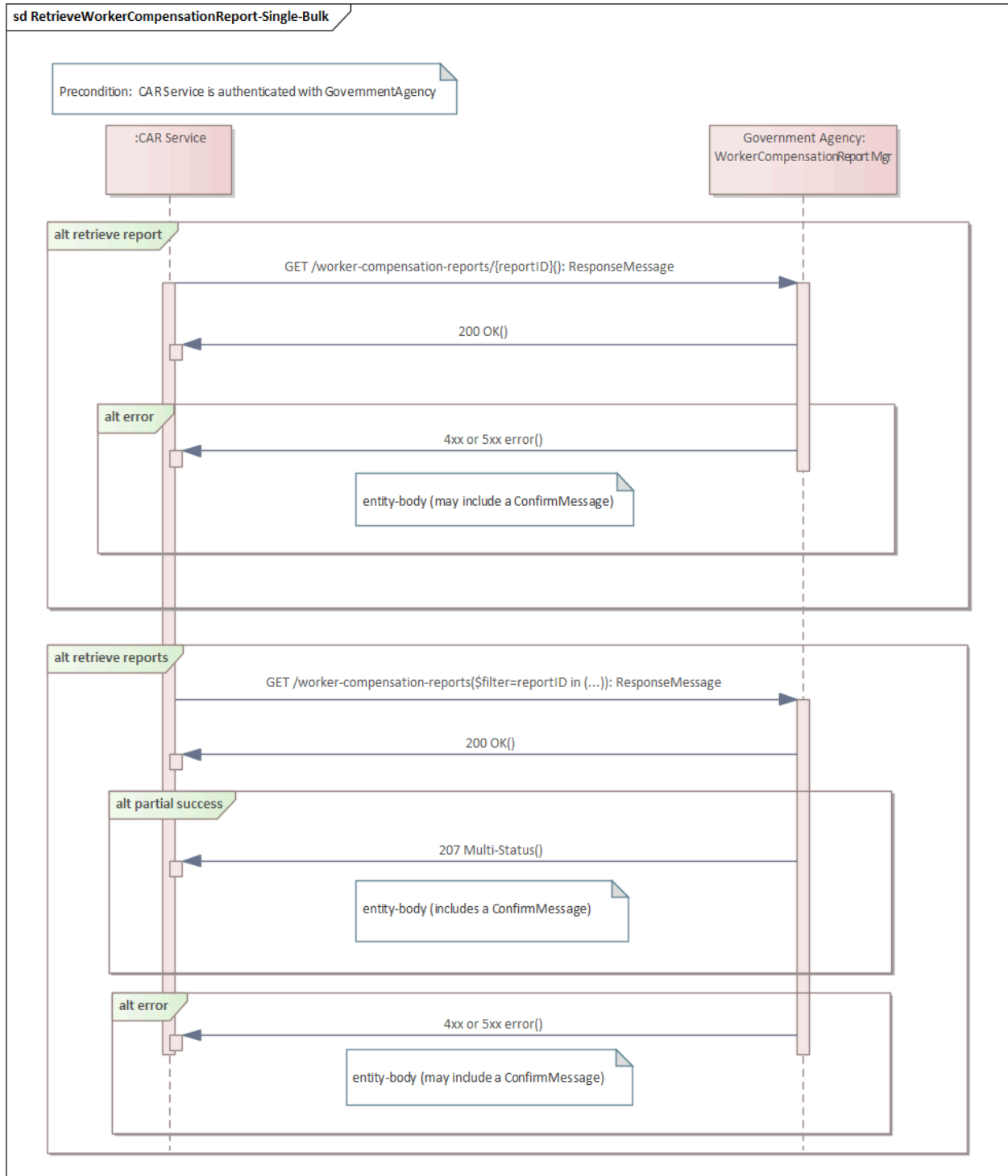


# Manage Worker Compensation Report

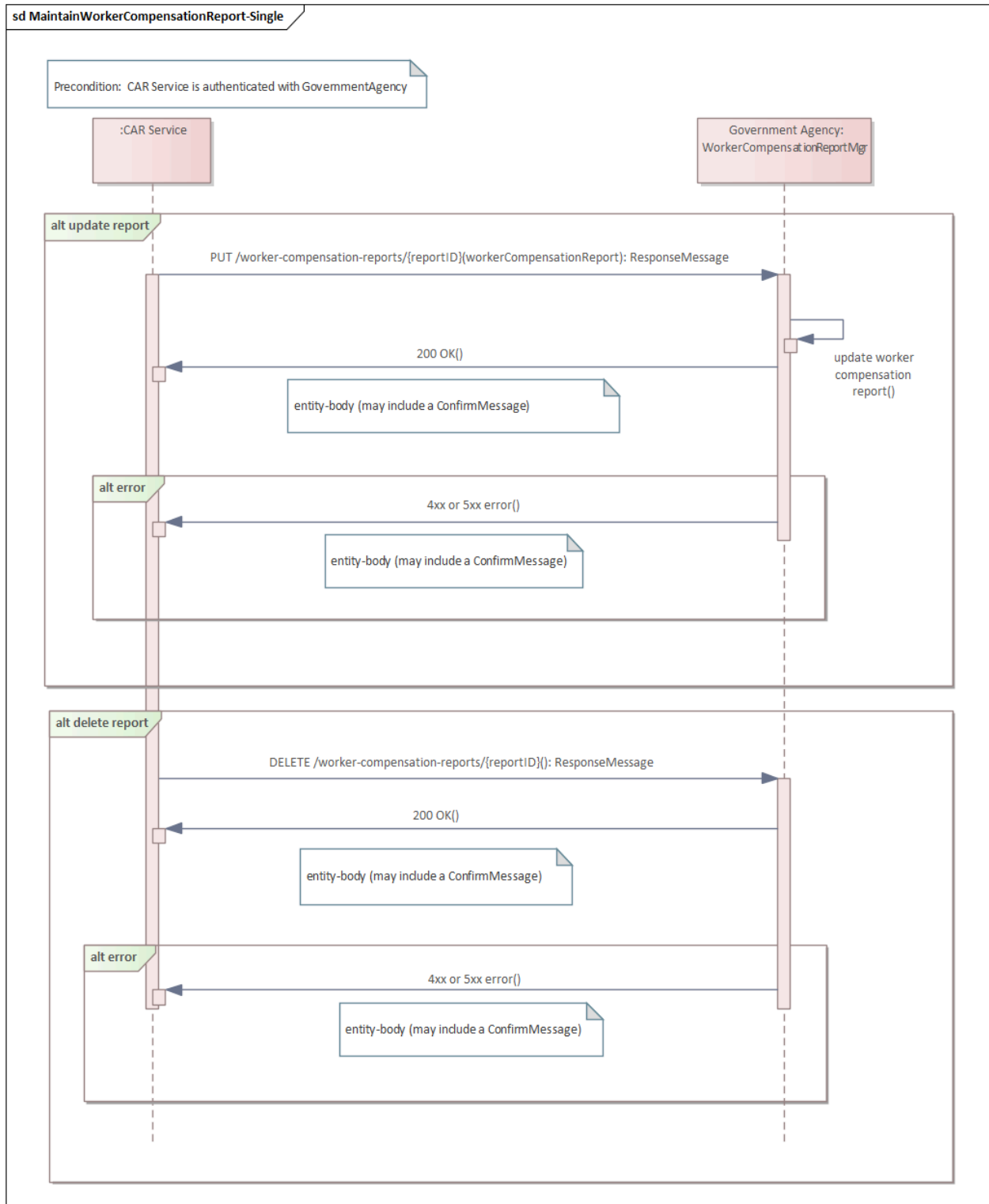
## Post Worker Compensation Report



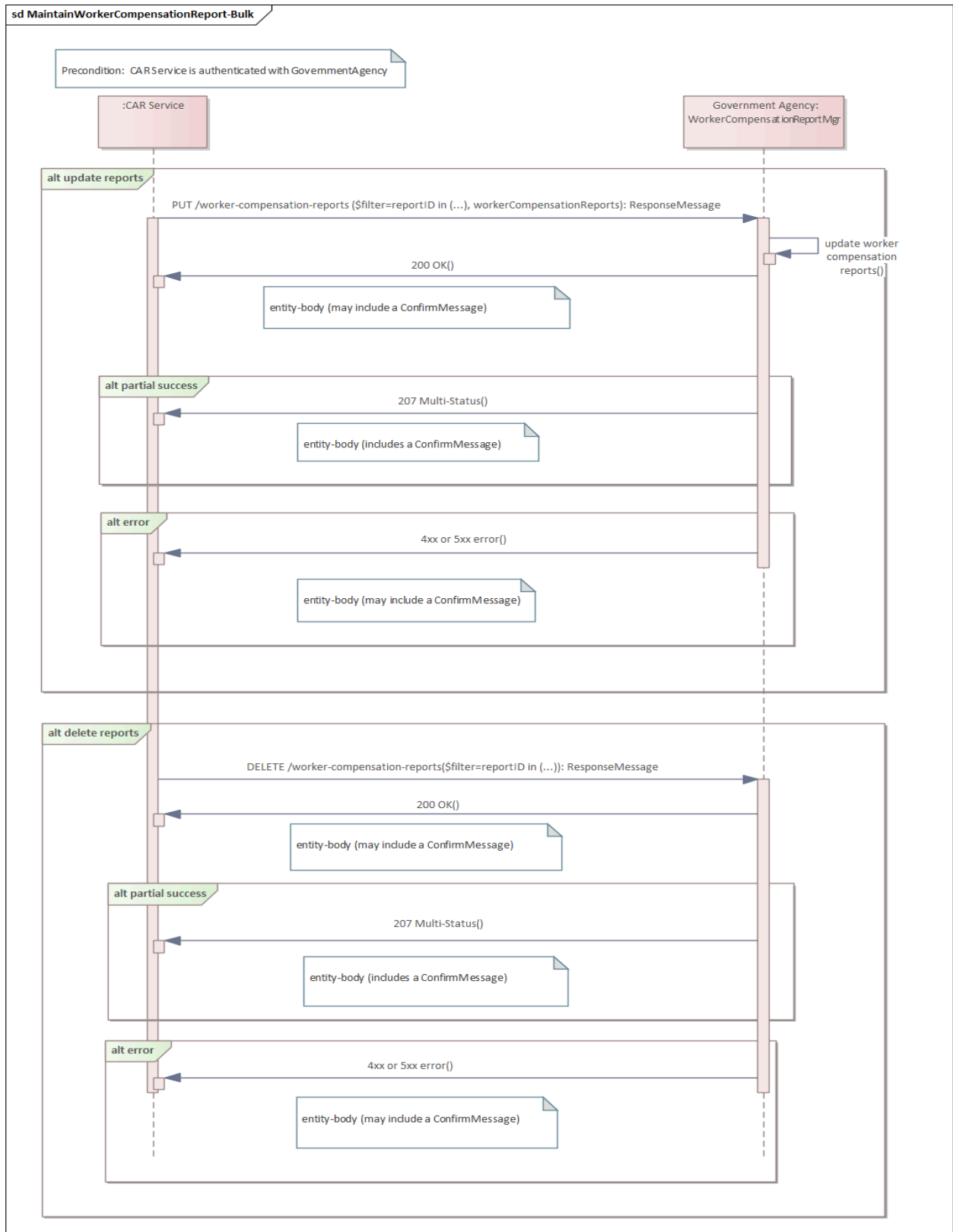
## Retrieve a Single or Multiple Worker Compensation Reports



## Maintaining a Single Worker Compensation Report

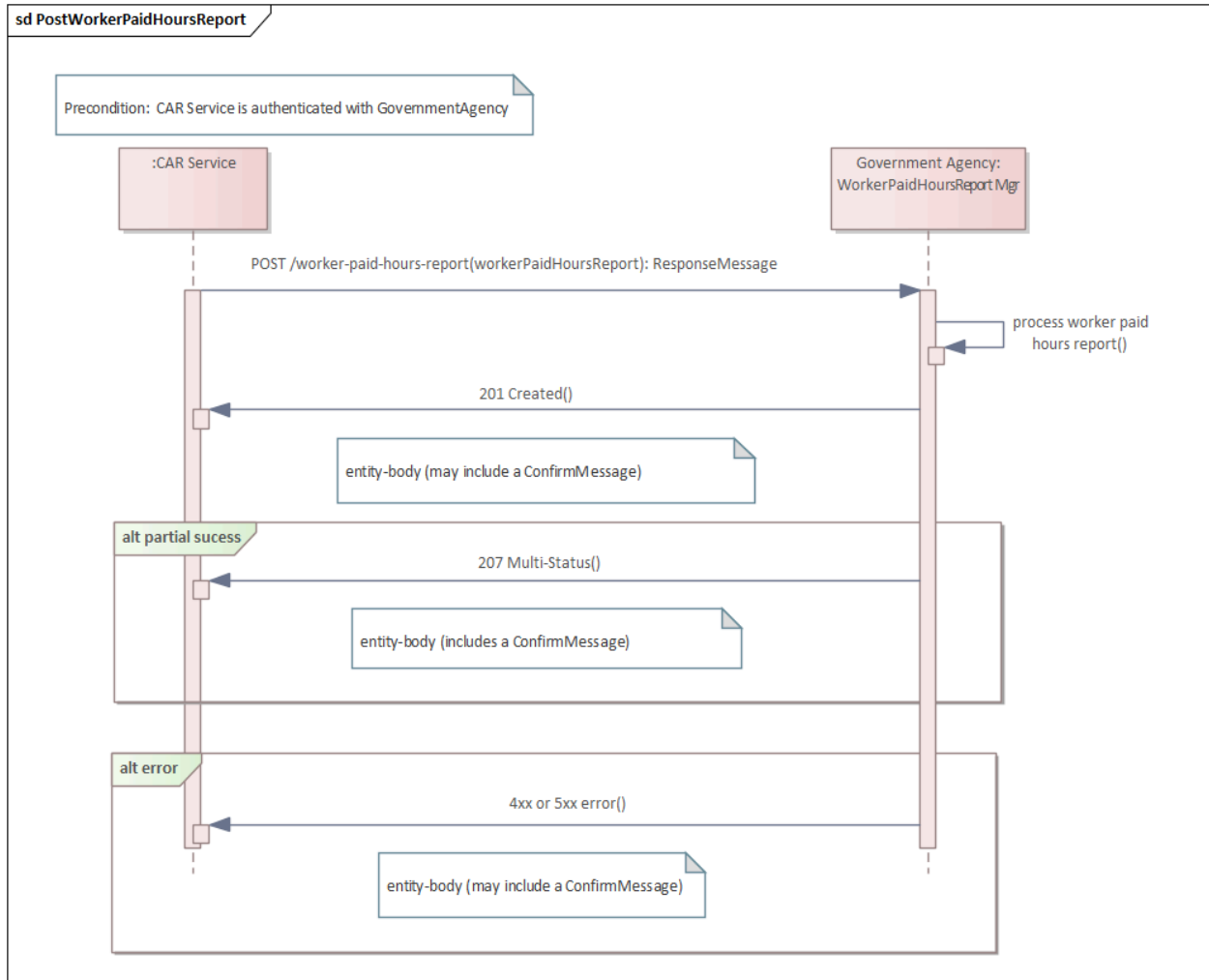


## Maintain Multiple Worker Compensation Reports

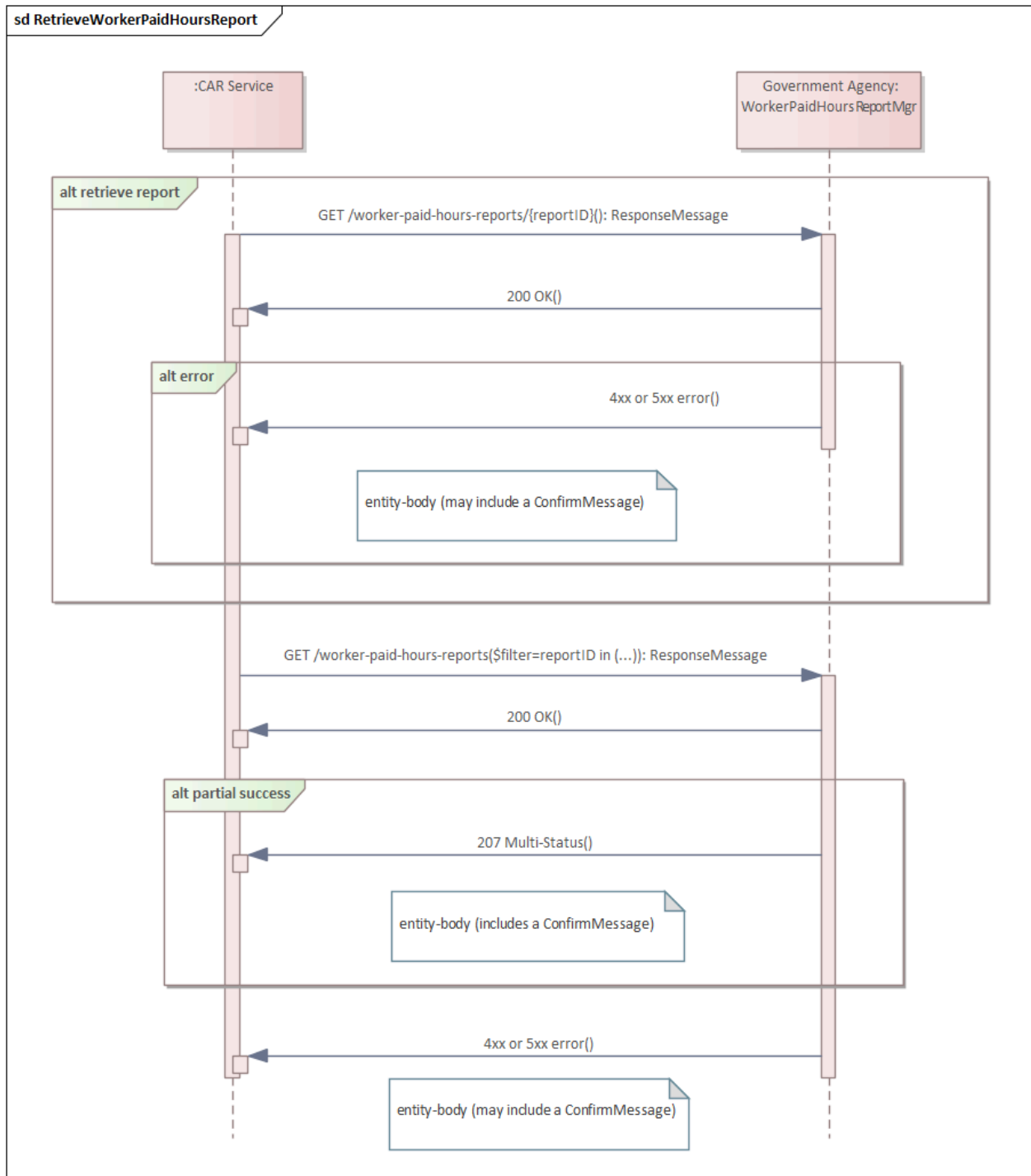


# Manage Worker Paid Hours Report

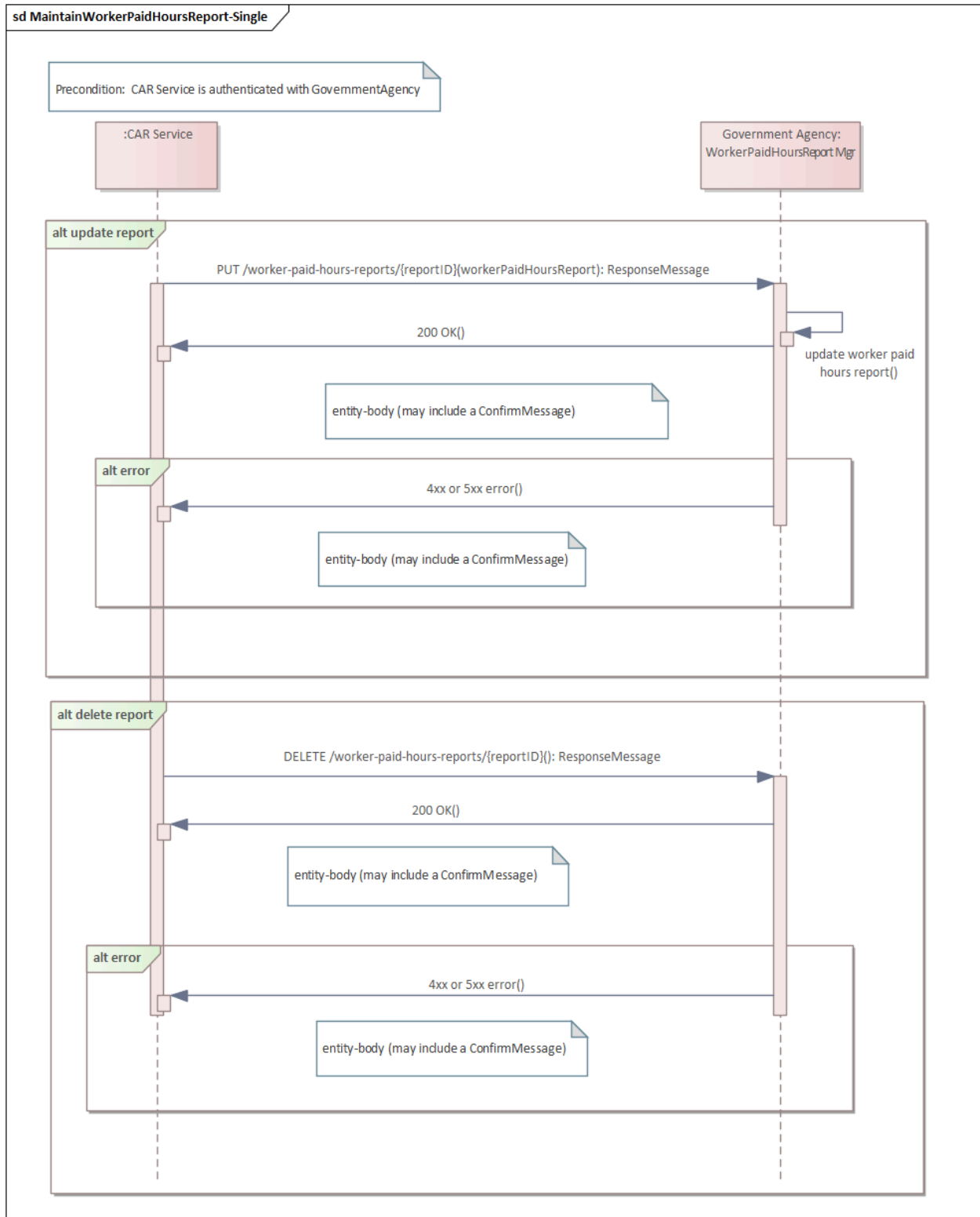
## Post Worker Paid Hours Report



## Retrieve a Single or Multiple Worker Paid Hours Reports

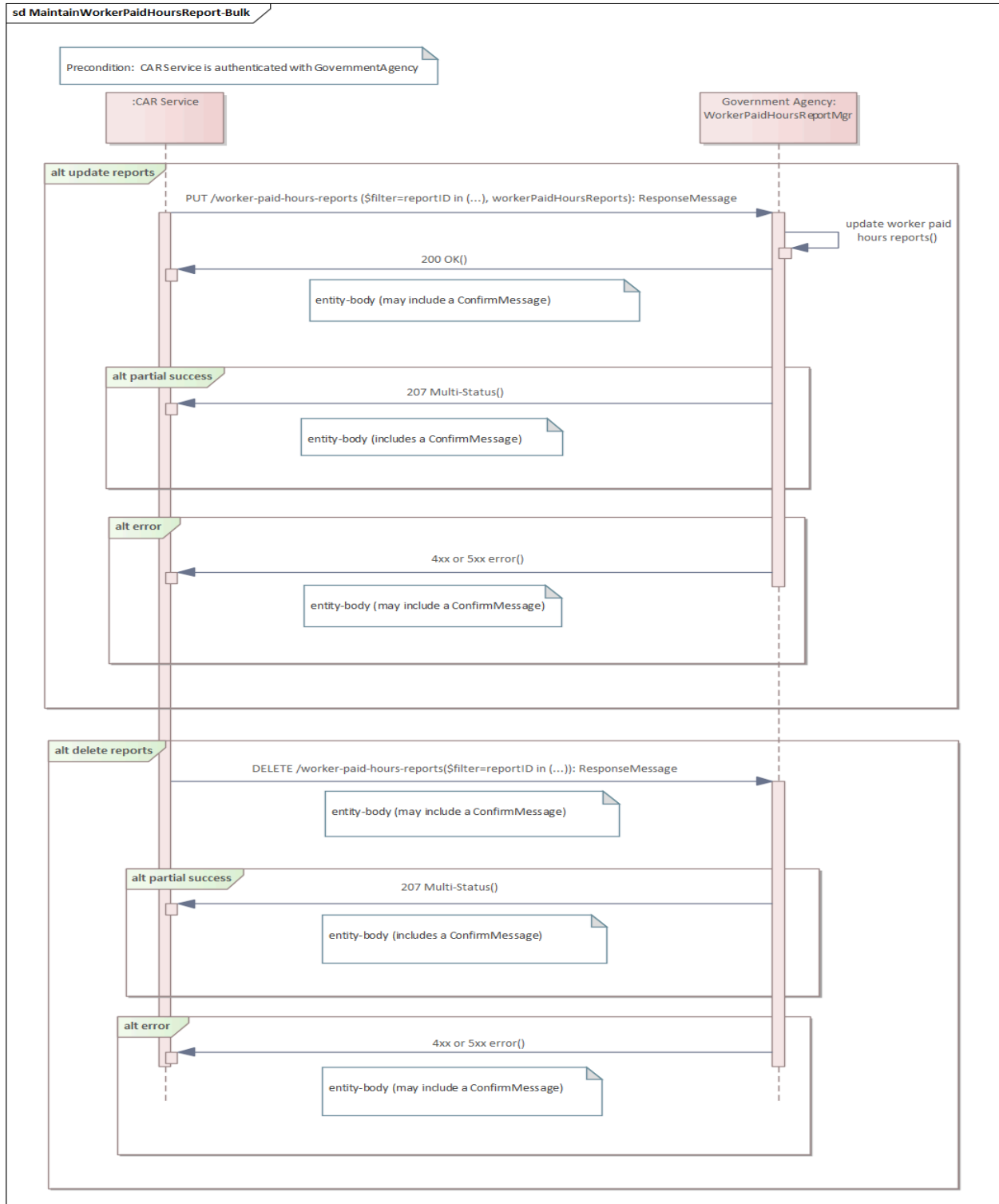


## Maintaining a Single Worker Paid Hours Report





## Maintain Multiple Worker Paid Hours Reports

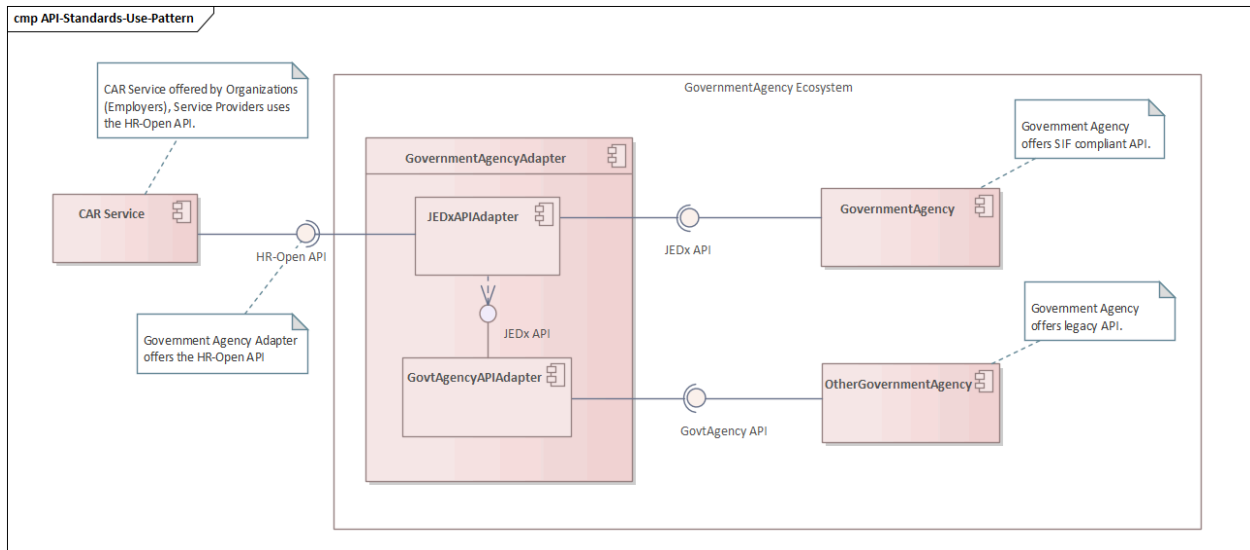


# API Serializations

API Serializations describe each of the component interfaces, presented above. The API Serializations, also referred to as API Documents, leverage the [OpenAPI Specification \(OAS\)](#)<sup>5</sup>. The OpenAPI Specification is an open standard that provides a consistent language for defining APIs. The following sections provide two types of API Serializations: the first is in conformance with HR-Open API standard(s) and the second is in conformance with JEDx API standard(s). More specifically, HR-Open adopted the Open Application Group's (OAGi) [RESTful Web API Design](#) specification<sup>6</sup> as normative for its API Documents. The JEDx API standard(s) adopted the [Systems Interoperability Framework \(SIF\) Infrastructure specification](#)<sup>7</sup> as normative for its API Documents. Note that although the two serializations conform to different RESTful Web API Design standards, both serializations use the information model (i.e., JSON schema), as defined by HR-Open.

Given that this standard supports two API Serializations, the question arises about how the two API Serializations will coexist and collaborate. The following diagram addresses this question.

Externally, the Government Agency ecosystem will offer the HR-Open APIs; the CAR Service will use this API. The JEDx API Adapter will transform the HR-Open API messages to JEDx API messages. Internally, the Government Agency ecosystem will support JEDx APIs and existing, legacy Government Agency APIs. In the case that a Government Agency (shown as OtherGovernmentAgency in the diagram) offers an existing, legacy API, then the Government Agency API Adapter will transform the JEDx API messages to the Government Agency-specific API messages.



<sup>5</sup> <https://www.openapis.org/what-is-openapi>

<sup>6</sup> [https://oagi-org.s3.amazonaws.com/OAGi\\_RESTful\\_Web\\_API\\_Design\\_EN\\_US\\_Letter.pdf](https://oagi-org.s3.amazonaws.com/OAGi_RESTful_Web_API_Design_EN_US_Letter.pdf)

<sup>7</sup> <https://data.a4l.org/sif-infrastructure/>

## HR-Open API Serializations

Link: <https://github.com/HROpen/APISpecifications/tree/jedx-project-1>

Summary: This is a ReDoc render of the OpenAPI resources comprising the relevant HR Open Standards 4.4. The specification may be uploaded and view here:

<https://redocly.github.io/redoc/>

The API Specifications can all be found [here](#), in the APISpecifications repo, **jedx-project-1** branch. See the readme for an explanation of the organization of the content in the repo.

Also, as a result of the data model and schema refactoring work, several schema changes were made. All these changes are in a single repo, [here](#), in the Common repo, **jedx-project-1** branch,

## JEDx API Serializations

Link: <http://specification.sifassociation.org/JEDx/ServiceDocs/>

Summary: This is a ReDoc render of the OpenAPI resources that include the relevant HR Open Standards 4.3 over SIF Infrastructure 3.4. At the intersection of these two standards, meet further discussion will define: unique IDs for addressing individual objects, grouping tags for operations involving multiple objects, the removal or inclusion of XML (where defined), and if this should be migrated to a Swagger render. Moreover, enhancements are anticipated to be made to the Data Model to fully address the use cases being piloted.

# Appendices

## References

(Armour et al.) Armour, F., Miller, G. Advanced Use Case Modeling. Addison-Wesley. 2001.  
(JEDx 2021) US Chamber of Commerce Foundation

## Document Map of this Proposal

**Source Format:** Google Document

**Description:** A description of the proposal and its intent and what it includes and does not include

## JEDx System Context, Components and Interfaces

**Source Format:**

1. Google Document with images from PowerPoint (PPT) and Enterprise Architect (EA) included
2. Source files (EA and PPT)

**Description:** This document would lay out:

- the problem space
- a system diagram of the ecosystem and an explanation of the ecosystem
- the use cases covered in the scope of the proposal
- component diagrams of initial objects
- sequence diagrams
- List of supported API actions

## API Serializations

- description of the two serializations supported

## HR-Open API Serialization

**Source Format:**

1. Google Document with images from PowerPoint (PPT) and Enterprise Architect (EA) included and references to deeper OAGI technical information and Swagger documentation.
2. Source files (EA and PPT)
3. Open API Swagger documentation

**Description:** The main document would lay out:

- All the API actions (per use case)
- Links back to the Context document
- Elaborations on the expected choreographies and sequence diagrams
- Links to the Open API Swagger documents for each action

## JEDx API Serialization

### **Source Format:**

1. Google Document with images from PowerPoint (PPT) and Enterprise Architect (EA) included and references to deeper A4L technical information and Swagger documentation (Example original: [JEDx API White Paper](#))
2. Source files (EA and PPT)
3. Open API Swagger documentation

### **Description:** The main document would lay out:

- All the API actions (per use case)
- Links back to the Context document
- Links to the Open API Swagger documents for each action