PROJECT POPLIN

MEDICAID REFERENCE ARCHITECTURE

SERVICE DEFINITION TEMPLATE

MITA Governance Board
Poplin Working Group

1. Introduction

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

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

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

2. Standards and Frameworks

Development of Poplin is distributed across disparate organizations, each with local and unique modeling and engineering standards and practices. Adoption of a common set of standards will act to connect and unify respective work efforts through common language and techniques.

2.1 Poplin Goals

The Poplin reference architecture has the following goals:

- Enable fine-grain modularity and microservice deployments
- Promote reusability of services
- Promote interoperability
- Promote innovation
- Provide a migration path for legacy systems
- Establish trust framework for nationwide information exchange

2.2 Poplin Principles

A primary goal of Poplin is to enable DevOps development practice and microservice deployments, utilizing Systems Thinking and Lean Thinking principles for analysis and design. There are a variety of interpretations and case-studies of implantation of these principles available in business literature. For the purposes of Poplin development, they are simplified below:

1) Systems Thinking
    a. *Holism.* Consider the context of system of interest.
    b. *Fractal.* Understand patterns between micro and macro perspective.
    c. *Synergy.* Understand that collaborative systems have greater value-production capability than aggregate of constituent parts.
    d. *Learn.* Engineer systems with feedback functionality for evolutionary, iterative design.

2) Lean Thinking
    a. *Value.* Understand value from the perspective of the customer and how value is delivered to the customer as a value stream.
    b. *Flow.* Design systems that identify and eliminate obstacles for delivery of value to customer.
    c. *Pull.* Design systems that execute as a function of customer demand.
    d. *Perfection.* Develop systems that evolve towards improved outcomes. Perfection is a direction, not a destination.

3) DevOps
    a. *Client Focus.* Develop solutions from a client perspective.
    b. *Collaborate.* Organize efforts in cross-functional, cross-domain teams.
    c. *Monitor Everything.* Continuously measure performance towards better insight.
    d. *Learn.* Continuously analyze and understand performance to refine existing and develop new techniques and technologies towards better performance.
    e. *Continuous Delivery.* Employ techniques that enable short cycle-times to drive by learning.

4) MicroService

a. *Domain Driven.* Understand commonalities in the problem space.

b. *Encapsulation.* Hide data and implementation complexities behind standard (public) interface.

c. *Failure Isolation.* Create systems which are adaptable and fault tolerant.

d. *Decentralization.* Build autonomy and distributed intelligence into systems.

e. *Continuous Delivery.* Utilize DevOps principles to enable continuous evolution of automation systems.

3. Standards and Frameworks

1) Unified Modeling Language 2.5 (UML) – Employ UML as means of diagram notation to describe Activity, Class, Object, State, Use Case, Sequence etc.

   http://www.omg.org/spec/UML/

2) Systems Modeling Language 1.5 (SysML) – Employ SysML extensions to UML to elaborate Requirements, Block, Internal Block, etc.

   http://www.omg.org/spec/SysML/

3) Unified Architecture Framework 1.0b1 (UAF) – Employ UAF as viewpoint framework to ensure comprehensive understanding of systems structure and function is described.

   http://www.omg.org/spec/UAF/

4) Medicaid Information Technology Architecture 3.0 (MITA) – Utilize MITA to relate RA to Medicaid Enterprise processes and resources.

   https://www.medicaid.gov/medicaid/data-and-systems/mita/mita-30/index.html

4. Architectural Development Process

Poplin is defining minimal functionality of services for MES implementations. These service definitions are encapsulations and abstractions of the underlying complexity of specific application service implementations, distilling necessary functionality as Service Specifications.

Development of the Service Specifications starts from analysis of the MITA business processes for which it will be consumed. Through understanding of processes' individual actions, information flow between actions, and stakeholder responsibilities to perform actions, in combination with subject matter expertise and historical context of process implementations and other solutions, opportunities for automation can be identified (see Figure 1: Process analysis steps).

Such automation opportunities can then be elaborated by understanding automation performer responsibilities across processes and synthesizing performer actions as ServiceFunctions (see Figure 3: Specification meta-model). Analysis of process flow can then be used to develop understanding of related resources and functional parameters.

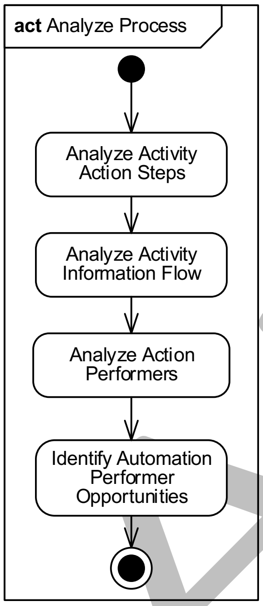
act Analyze Process
Analyze Activity Action Steps
Analyze Activity Information Flow
Analyze Action Performers
Identify Automation Performer Opportunities

Figure 1: Process analysis steps

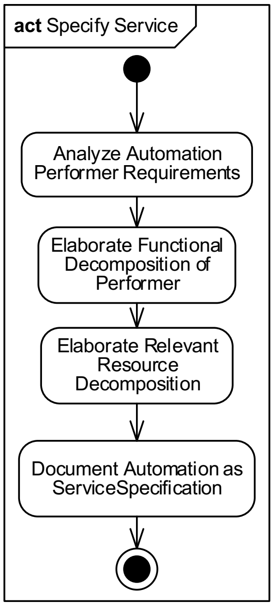
act Specify Service
Analyze Automation Performer Requirements
Elaborate Functional Decomposition of Performer
Elaborate Relevant Resource Decomposition
Document Automation as ServiceSpecification

Figure 2: Service specification steps

5. Service Definition Architectural Artifacts

Service definitions for Poplin should contain the following four artifacts for each service:

1) Business Process Models
2) Object / Resource Diagrams
3) Object / Resource Descriptions
4) API Descriptions

5.1 Business Process Models

Business Process Models are behavior diagrams and descriptions that represent the functionality and flow of a system and its processes.

a. Sys/UML Diagrams

   i. Activity
   ii. UseCase
   iii. Sequence
   iv. Communication/Information Flow
   v. State
   vi. Class/Block

5.2 Object and Resource Models

Object and Resource Models are structural diagrams and descriptions that represent the entity composition and relationships of a system

a. Sys/UML Diagrams

   i. Class/Block/Object
   ii. CompositeStructure/InternalBlock

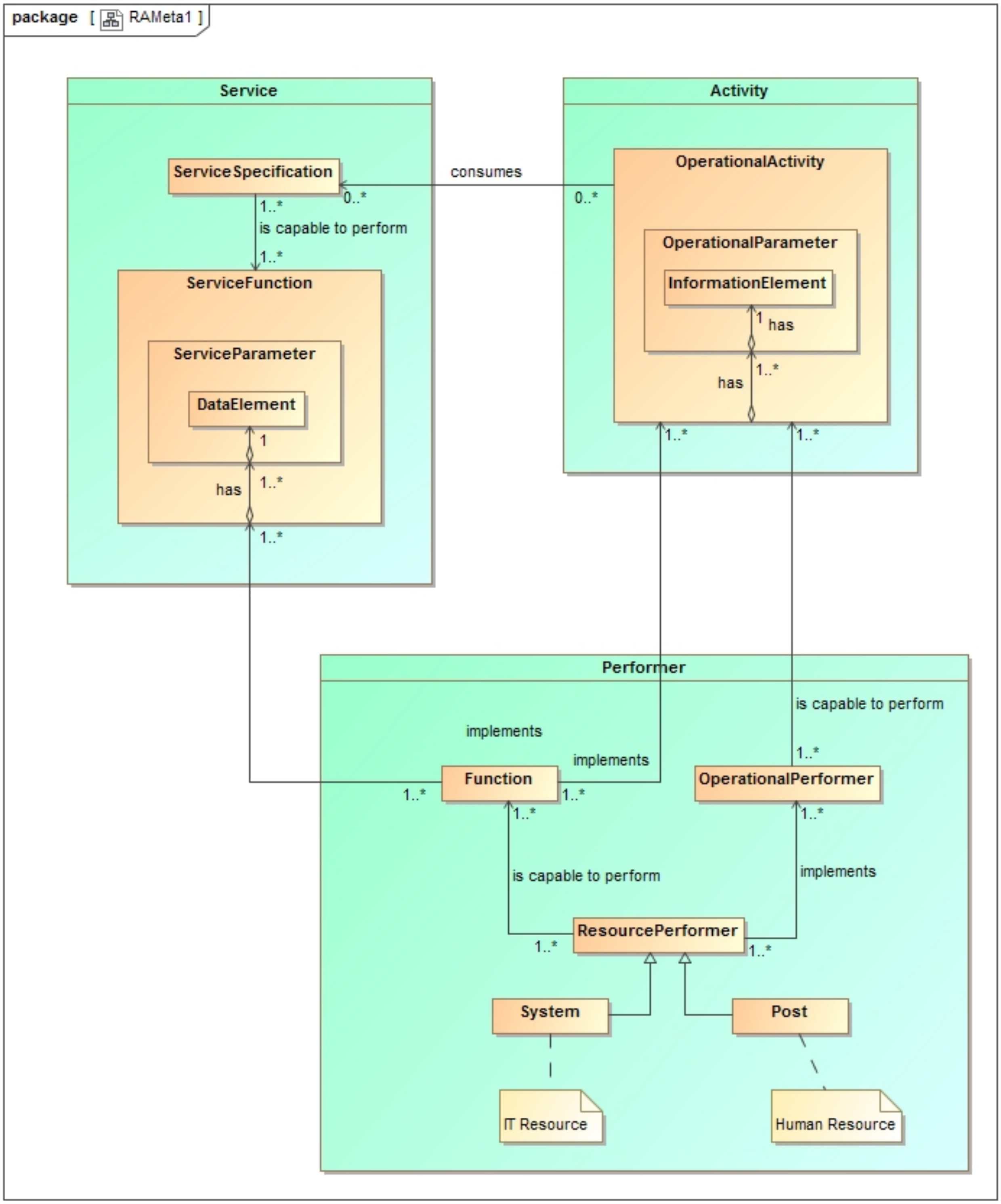

Figure 3: Specification Meta-Model

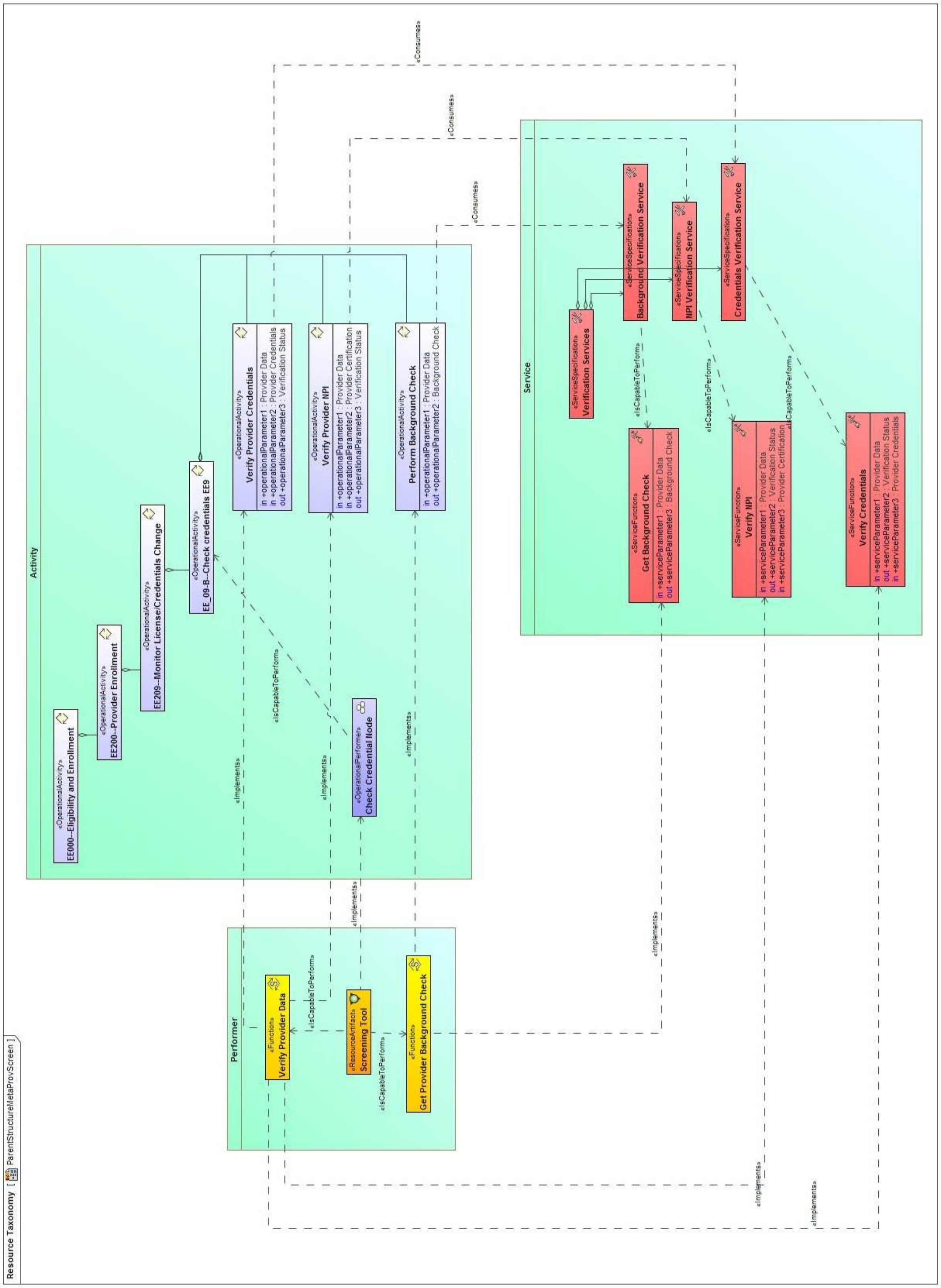
Resource Taxonomy [ParentStructureMetaProvScreen]
Activity
«OperationalActivity»
EE000--Eligibility and Enrollment
«OperationalActivity»
EE200--Provider Enrollment
«OperationalActivity»
EE209--Monitor License/Credentials Change
«OperationalActivity»
EE_09-B--Check credentials EE9
«OperationalActivity»
Verify Provider Credentials
in +operationalParameter1 : Provider Data
in +operationalParameter2 : Provider Credentials
out +operationalParameter3 : Verification Status
«OperationalActivity»
Verify Provider NPI
in +operationalParameter1 : Provider Data
in +operationalParameter2 : Provider Certification
out +operationalParameter3 : Verification Status
«OperationalActivity»
Perform Background Check
in +operationalParameter1 : Provider Data
out +operationalParameter2 : Background Check
«OperationalPerformer»
Check Credential Node
«IsCapableToPerform»
«Implements»
«Consumes»
Service
«ServiceSpecification»
Verification Services
«ServiceSpecification»
Background Verification Service
«ServiceSpecification»
NPI Verification Service
«ServiceSpecification»
Credentials Verification Service
«ServiceFunction»
Get Background Check
in +serviceParameter1 : Provider Data
out +serviceParameter3 : Background Check
«ServiceFunction»
Verify NPI
in +serviceParameter1 : Provider Data
out +serviceParameter2 : Verification Status
in +serviceParameter3 : Provider Certification
«ServiceFunction»
Verify Credentials
in +serviceParameter1 : Provider Data
out +serviceParameter2 : Verification Status
in +serviceParameter3 : Provider Credentials
Performer
«Function»
Verify Provider Data
«ResourceArtifact»
Screening Tool
«Function»
Get Provider Background Check

Figure 4: Provider Screening Structural Diagram Example

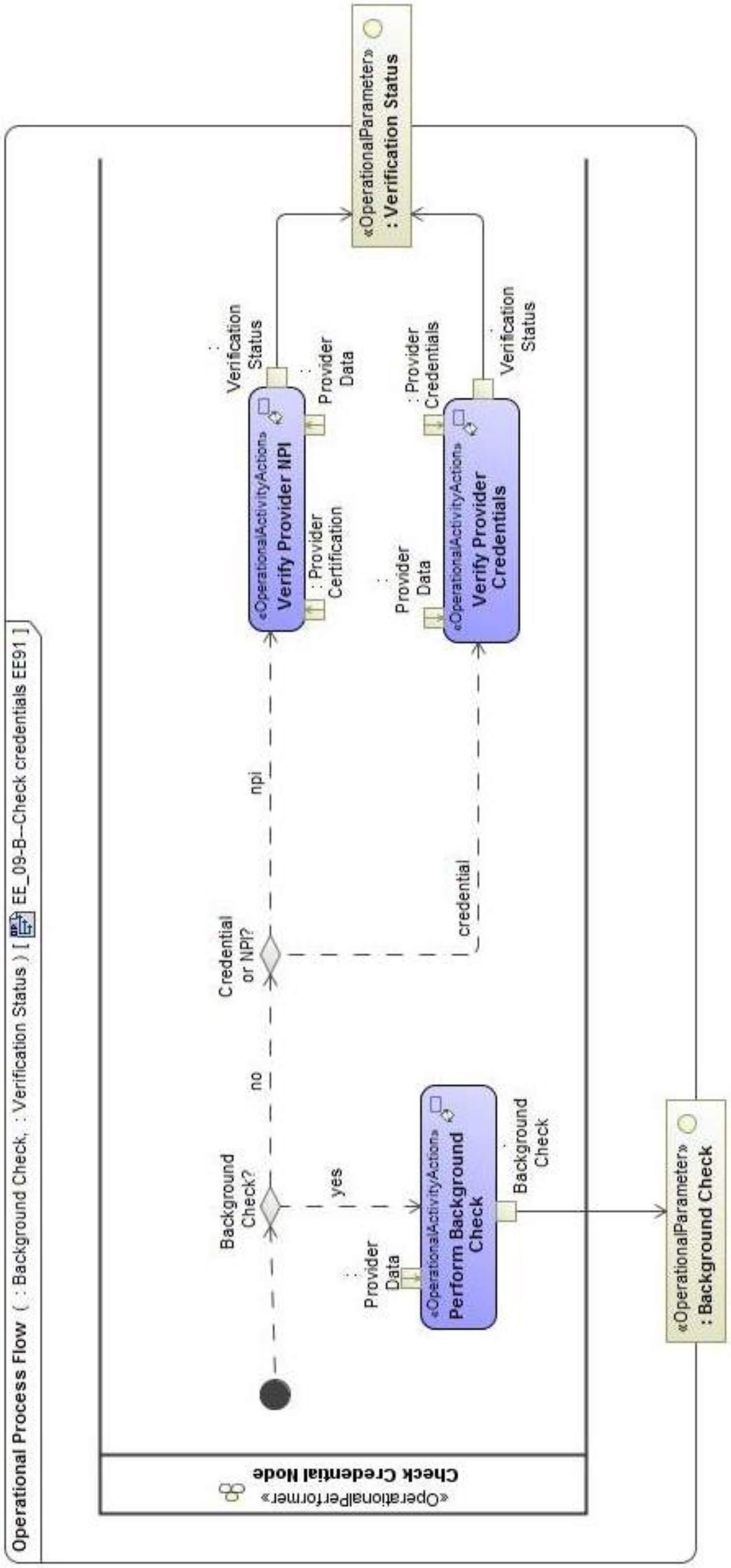

Figure 5: Provider Screening Behavioral Diagram Example

5.3 Object and Resource Definitions

Object / resource definitions specify the nature of the data that can be exchanged with a service, what fields are included with each object, the name and type of each field, whether the field is optional or required, and type of relationships with other objects (one-to-one, 0 or more, 1 or more, etc).

Specifically, the object / resource definition should have the following form:

<Name of Object/Resource>
    <Based on parent Object if inherited>
<Description of Object/Resource>
<List of fields in Object/Resource>
    <Name and type of field>   <optional or required, relationship>   <Description of field>
    <Name and type of field>   <optional or required, relationship>   <Description of field>
    …

5.3.1 Examples

The following example has been taken from the Standard Health Record project, an open source initiative to standardize health records across Electronic Health Record systems and currently under development. The full specification is available at standardhealthrecord.org.

In this example, several objects are presented below within an inheritance tree around Practitioners, along with the fields for each object. Each field is described and specified whether the value is required, optional, or part of a relationship (0 or more, 1 or more, etc). The example contains many links which provide some context.

Person

A person. In practical terms, people relevant to the health or social situation of the subject (including the person of record). C0027361

Value: HumanName	A name used by a human being, written as it would be typically expressed. May include a breakdown of the various elements

		of the name (family name, given name, etc.).
`Anonymized`	optional	Flag indicating if personally identifiable information has been withheld.
`FictionalPerson`	optional	Flag indicating if this record represents a fictional (synthetic, not real) person.
`DateOfBirth`	optional	A date of birth or approximate year or period (year or date range), if estimated.
`AdministrativeGender`	optional	A gender classification used for administrative purposes. Administrative gender is not necessarily the same as a biological description or a gender identity. This attribute does not include terms related to clinical gender.
`AddressUsed`	0 or more	An address used by for dwelling, work, or other purposes. May be a physical location or an address used only for deliveries, such as a PO box.
`Telecom`	0 or more	An electronic means of contacting an organization or individual.
`LanguageUsed`	0 or more	Language used for communication by a human, either the subject of record, parent, or other involved person.
`Deceased`	optional	An indication that the person is no longer living, given by a date, time of death, or a boolean value which, when true, indicates the person is deceased.

Practitioner

A person who practices a healing art. C2220264

Based On Person

Value: HumanName		A name used by a human being, written as it would be typically expressed. May include a breakdown of the various elements of the name (family name, given name, etc.).
Affiliation	0 or more	Membership, association, or connection to an organization.
NationalProviderIdentifier	optional	A unique 10-digit number to allow providers to identify themselves in a standard way throughout their industry.
MedicalSpecialty	0 or more	A branch of medicine practiced by the provider.
Anonymized	optional	Flag indicating if personally identifiable information has been withheld.
FictionalPerson	optional	Flag indicating if this record represents a fictional (synthetic, not real) person.
DateOfBirth	optional	A date of birth or approximate year or period (year or date range), if estimated.
AdministrativeGender	optional	A gender classification used for administrative purposes. Administrative gender is not necessarily the same as a biological description or a gender identity. This attribute does not include terms related to clinical gender.
AddressUsed	0 or more	An address used by for dwelling, work, or other purposes. May be a physical location or an address used only for deliveries, such as a PO box.

Telecom	0 or more	An electronic means of contacting an organization or individual.
LanguageUsed	0 or more	Language used for communication by a human, either the subject of record, parent, or other involved person.
Deceased	optional	An indication that the person is no longer living, given by a date, time of death, or a boolean value which, when true, indicates the person is deceased.

PrimaryCareProvider [Entry]

The medical professional responsible for first-line care of an individual. C2735026

Based On ProviderRelationship

HealthcareRole	is C2735026	required	A type of service provided to the subject.
NonOccurrenceModifier		not used	When true, indicates either that the event or action documented in the entry did not occur, or the the thing documented is absent or does not exist.
Provider		required	A person or organization who provides services. This includes healthcare and relevant non-healthcare services, including personal services

		such as transportation.
GeneralizedTemporalContext	optional	A generalized indicator of a dateTime or an age when an event happened.
Priority	optional	An ordinal numbered priority where 1=highest priority, 2=second highest priority, etc.

Provider

A person or organization who provides services. This includes healthcare and relevant non-healthcare services, including personal services such as transportation.C1138603

Value: Choice		
| RelatedPerson	required	A person, other than a practitioner or the patient (the person of record) who is relevant to the health or social situation of the subject.
| Practitioner	required	A person who practices a healing art.
| Organization	required	A social or legal structure formed by human beings.

ProviderRelationship [Entry]

A person or organization responsible for providing a service to the person of record. Such a relationship can exist without a specific encounter or services being rendered; for example, an emergency contact relationship can exist without ever calling upon that person. The entry can also be used to state that a provider relationship does not exist, e.g., patient does not have a primary care provider. C1314939

Provider	optional	A person or organization who provides services. This includes healthcare and relevant non-healthcare

		services, including personal services such as transportation.
`HealthcareRole`	required	A type of service provided to the subject.
`NonOccurrenceModifier`	optional	When true, indicates either that the event or action documented in the entry did not occur, or the the thing documented is absent or does not exist.
`GeneralizedTemporalContext`	optional	A generalized indicator of a dateTime or an age when an event happened.
`Priority`	optional	An ordinal numbered priority where 1=highest priority, 2=second highest priority, etc.

5.4 API Definitions

API definitions specify the verbs and nouns used to communicate with a service. In the case of a RESTful interface, the verbs would be GET, PUT, POST, and DELETE. The nouns would be the resources or representations, for example, “widget”.

The API definition should provide all of the information needed to successfully call the service endpoints, decipher the results, and respond meaningfully to errors.

<API Overview>

    <General description of the purpose of the API and notes that should be applied generally by clients when accessing the API.>

<List of API endpoints>

    <Verb for the endpoint (e.g. GET, PUT, POST, DELETE for RESTful services) followed by the relative URL for the endpoint>

    <Supported data formats>

    <Any header fields that are required>

    <List of parameters that are required>

        <Description and type>

        <List or range of valid values, if appropriate>

<Optional or required>

<List of return values>

<Examples of a call in the supported formats, return value formats, and common errors>

5.4.1 Examples

The following example is borrowed from Project Cedar, an open source tool to validate Quality Reporting Document Architecture (QRDA) collection systems. The resource in this example is test_execution, which is used to manage the retrieval, creation, and deletion of QRDA tests executed by a particular user.

- Cedar v1

Overview¶ ↑

Cedar is a tool for testing the strength of Electronic Clinical Quality Measure (eCQM) collection systems that receive Quality Reporting Document Architecture (QRDA) files.

Registrations and Sessions are for user account control. Test executions create QRDA data, which are stored in Documents. Documents are also used to check your collection system's accuracy.

Follows the JSON-API specification. In addition to formatting responses, JSON-API compliance means requests must be made with 'application/vnd.api+json' accept and content-type headers.

- Cedar v1 / Test executions / index

GET /api/v1/test_executions

Get all tests of current user

Supported Formats

json

Headers

Header name	Description
X-API-EMAIL required	user's email

Header name	Description
X-API-TOKEN required	user's current authentication token

Examples

```
GET /api/v1/test_executions
200
{
  "data": [
    {
      "type": "test_executions",
      "id": "5797a2973bb752280873d7cd",
      "attributes": {
        "state": "incomplete",
        "qrda_progress": 0.0,
        "name": "first test",
        "description": "",
        "reporting_period": "2016",
        "file_path": "",
        "qrda_type": "3",
        "validations": [
          "duplicate_population_ids"
        ],
        "measures": [
          "CMS109v4"
        ]
      },
      "relationships": {
        "documents": {
          "data": []
        }
      },
      "links": {
        "self": "http://127.0.0.1:3000/api/v1/test_executions/5797a2973bb752280873d7cd
"
      }
    }
  ]
}
```

- Cedar v1 / Test executions / create

POST /api/v1/test_executions

create new test

Supported Formats

json

Params

Param name	**Description**
data required	Root level hash **Validations:** - Must be a Hash
data[attributes] required	Properties to set **Validations:** - Must be a Hash
data[attributes][name] required	Name of test **Validations:** - Must be String
data[attributes][description] optional	Description for this test **Validations:** - Must be String
data[attributes][reporting_period] required	Which year measures will be from **Validations:** - Must be one of: `2016`, `2015`, `2014`.
data[attributes][qrda_type] required	“1” or “3” for category 1 and category 3 QRDA data, measures, and validations, respectively **Validations:** - Must be one of: `1`, `3`.
data[attributes][validations] required	Validation selection. Validations are referenced individually by their code/id, and collectively through tags. Cedar will attempt to filter out incompatible validations, and a message will be sent in the meta hash of the response if validations are removed. **Validations:** - Must be a Hash

Param name	Description
data[attributes][validations][tags] optional	Compatible validations with these tags will be included **Validations:** - Must be an array of String
data[attributes][validations][include] optional	Individual validations to include **Validations:** - Must be an array of String
data[attributes][validations][exclude] optional	Individual validations to be excluded (applied last) **Validations:** - Must be an array of String
data[attributes][validations][all] optional	Set true to include all compatible validations. Excluded validations are still excluded **Validations:** - Must be Boolean
data[attributes][measures] required	Measure selection. Measures are referenced individually by their cms_id or hqmf_id, and collectively through tags. Cedar will attempt to filter out incompatible measures, and a message will be sent in the meta hash of the response if measures are removed. **Validations:** - Must be a Hash
data[attributes][measures][tags] optional	Compatible measures with these tags will be included **Validations:** - Must be an array of String
data[attributes][measures][include] optional	Individual measures to include **Validations:** - Must be an array of String
data[attributes][measures][exclude] optional	Individual measures to be excluded (applied last) **Validations:**

Param name	Description
	- Must be an array of String
data[attributes][measures][all] optional	Set true to include all compatible measures. Excluded measures are still excluded **Validations:** - Must be Boolean

Headers

Header name	Description
X-API-EMAIL required	user's email
X-API-TOKEN required	user's current authentication token

Examples

```
POST /api/v1/test_executions
{
  "data": {
    "attributes": {
      "name": "incompatible measure and validation",
      "reporting_period": "2016",
      "qrda_type": "3",
      "measures": {
        "include": [
          "CMS55v4"
        ]
      },
      "validations": {
        "include": [
          "numer_greater_than_denom"
        ]
      }
    }
  }
}
400
{
  "errors": [
    {
      "title": "Incompatible Validations and Measures",
      "detail": "All selected measures are continuous, but validations
numer_greater_than_denom are only for discrete measures"
    }
  ]
```

```
}
POST /api/v1/test_executions
{
  "data": {
    "attributes": {
      "name": "first",
      "reporting_period": "2016",
      "qrda_type": "1",
      "measures": {
        "tags": [
          "Hospital"
        ],
        "include": [
          "40280381-4B9A-3825-014B-C2730E6F088C",
          "CMS117v4",
          "CMS100V2"
        ],
        "exclude": [
          "CMS9v4",
          "CMS91v5"
        ]
      },
      "validations": {
        "tags": [
          "Schema"
        ],
        "include": [
          "discharge_after_upload",
          "numer_greater_than_denom"
        ]
      }
    }
  }
}
202
{
  "data": {
    "type": "test_executions",
    "id": "5797a2953bb752280873d7bc",
    "attributes": {
      "state": "incomplete",
      "qrda_progress": 0.0,
      "name": "first",
      "description": "",
      "reporting_period": "2016",
      "file_path": "",
      "qrda_type": "1",
      "validations": [
        "unfinished_file",
        "discharge_after_upload"
      ],
      "measures": [
        "CMS100v4",
        "CMS102v4",
        "CMS104v4",
        "CMS105v4",
        "CMS107v4",
        "CMS108v4",
        "CMS109v4",
        "CMS110v4",
        "CMS111v4",
        "CMS113v4",
        "CMS114v4",
        "CMS171v5",
        "CMS172v5",
        "CMS178v5",
        "CMS185v4",
        "CMS188v5",
        "CMS190v4",
        "CMS26v3",
        "CMS30v5",
        "CMS31v4",
        "CMS32v5",
        "CMS53v4",
        "CMS55v4",
        "CMS60v4",
        "CMS71v5",
        "CMS72v4",
        "CMS73v4",
        "CMS117v4",
        "CMS75v4"
      ]
    },
    "relationships": {
      "documents": {
        "data": []
      }
    },
    "links": {
      "self": "http://127.0.0.1:3000/api/v1/test_executions/5797a2953bb752280873d7bc"
    }
  },
  "meta": {
    "filtered": [
      {
        "removed": "CMS100v2",
        "reason": "Incorrect Measure Reporting Period",
        "conflict": "2016"
      },
      {
        "removed": "numer_greater_than_denom",
        "reason": "Incorrect QRDA Type",
        "conflict": "QRDA cat 1"
      }
    ]
  }
}
POST /api/v1/test_executions
{
  "data": {
    "attributes": {
      "name": "all",
      "reporting_period": "2016",
      "qrda_type": "1",
      "measures": {
        "all": true
      },
      "validations": {
        "all": true
      }
    }
  }
```

```
}
202
{
  "data": {
    "type": "test_executions",
    "id": "5797a2963bb752280873d7c2",
    "attributes": {
      "state": "incomplete",
      "qrda_progress": 0.0,
      "name": "all",
      "description": "",
      "reporting_period": "2016",
      "file_path": "",
      "qrda_type": "1",
      "validations": [
        "discharge_after_upload",
        "discharge_before_admission",
        "incorrect_code_system",
        "invalid_code",
        "invalid_value_set",
        "value_set_without_code_system",
        "inconsistent_time_formats",
        "invalid_measure_id",
        "reporting_period",
        "unfinished_file"
      ],
      "measures": [
        "CMS100v4",
        "CMS102v4",
        "CMS104v4",
        "CMS105v4",
        "CMS107v4",
        "CMS108v4",
        "CMS109v4",
        "CMS110v4",
        "CMS111v4",
        "CMS113v4",
        "CMS114v4",
        "CMS171v5",
        "CMS172v5",
        "CMS178v5",
        "CMS185v4",
        "CMS188v5",
        "CMS190v4",
        "CMS26v3",
        "CMS30v5",
        "CMS31v4",
        "CMS32v5",
        "CMS53v4",
        "CMS55v4",
        "CMS60v4",
        "CMS71v5",
        "CMS72v4",
        "CMS73v4",
        "CMS91v5",
        "CMS9v4",
        "CMS117v4",
        "CMS122v4",
        "CMS123v4",
        "CMS124v4",
        "CMS125v4",
        "CMS126v4",
        "CMS127v4",
        "CMS128v4",
        "CMS129v5",
        "CMS130v4",
        "CMS131v4",
        "CMS132v4",
        "CMS133v4",
        "CMS134v4",
        "CMS135v4",
        "CMS136v5",
        "CMS137v4",
        "CMS138v4",
        "CMS139v4",
        "CMS140v4",
        "CMS141v5",
        "CMS142v4",
        "CMS143v4",
        "CMS144v4",
        "CMS145v4",
        "CMS146v4",
        "CMS147v5",
        "CMS148v4",
        "CMS149v4",
        "CMS153v4",
        "CMS154v4",
        "CMS155v4",
        "CMS156v4",
        "CMS157v4",
        "CMS158v4",
        "CMS159v4",
        "CMS160v4",
        "CMS161v4",
        "CMS163v4",
        "CMS164v4",
        "CMS165v4",
        "CMS166v5",
        "CMS167v4",
        "CMS169v4",
        "CMS177v4",
        "CMS179v4",
        "CMS182v5",
        "CMS22v4",
        "CMS2v5",
        "CMS50v4",
        "CMS52v4",
        "CMS56v4",
        "CMS61v5",
        "CMS62v4",
        "CMS64v5",
        "CMS65v5",
        "CMS66v4",
        "CMS68v5",
        "CMS69v4",
        "CMS74v5",
        "CMS75v4",
        "CMS77v4",
        "CMS82v3",
        "CMS90v5"
      ]
    },
    "relationships": {
      "documents": {
        "data": []
      }
    },
    "links": {
      "self": "http://127.0.0.1:3000/api/v1/test_executions/5797a2963bb752280873d7c2
    }
  }
}
```

Appendix A: Definition of stereotypes

Name	Definition
DataElement	A formalized representation of data that is managed by or exchanged between systems
Function	An Activity which is specified in context of the ResourcePerformer (human or machine) that IsCapableOf Performing it.
InformationElement	An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities
OperationalActivity	An Activity that captures a logical process, specified independently of how the process is carried out
OperationalParameter	An element that represents inputs and outputs of an OperationalActivity
OperationalPerformer	Denotes a logical entity that Is Capable To Perform Operational Activities which produce, consume and process resources.
Post	A type of job title or position that a person can fill
ResourcePerformer	An element that can perform Functions
ServiceFunction	An Activity that describes the abstract behavior of ServiceSpecifications, regardless of the actual implementation.
ServiceParameter	An element that represents inputs and outputs of a ServiceFunction, represents inputs and outputs of a ServiceSpecification
ServiceSpecification	The specification of a set of functionalities provided by one element for the use of others
System	A System is an artificial artifact consisting of blocks that pursue a common goal that cannot be achieved by the system's individual elements
consumes	An abstraction relationship that asserts that a service in someway contributes or assists in the execution of an OperationalActivity
Has (aggregation)	A binary association between a property and one or more composite objects which group together a set of instances
implements	A component definition that is not intended to have a specification itself. Rather, it is an implementation for a separate «specification» to which it has a Dependency
is capable to perform	An Abstraction relationship defining the traceability between the CapableElements to the Activities that they can perform