Baseline API and Data Model Version 1.0 - Part 2

Project Specification Draft 01

17 April 2024

This stage:

https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/psd01/baseline-api-v1.0-psd01-part2.md (Authoritative)

https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/psd01/baseline-api-v1.0-psd01-part2.html

https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/psd01/baseline-api-v1.0-psd01-part2.pdf

Previous stage:

N/A

Latest stage:

https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/baseline-api-v1.0-part2.md (Authoritative)

https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/baseline-api-v1.0-part2.html

https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/baseline-api-v1.0-part2.pdf

Open Project:

Baseline, part of the Ethereum Community Projects

Project Chair:

Dr. Andreas Freund (a.freundhaskel@gmail.com)

Former Chair:

John Wolpert (john.wolpert@mesh.xyz), ConsenSys Mesh

Editors:

Kyle Thomas (kyle@provide.services), Provide

Andreas Freund (a.freundhaskel@gmail.com)

Yoav Bittan (yoav.bittan@mesh.xyz), ConsenSys Mesh

Keith Salzman (keith.salzman@mesh.xyz), ConsenSys Mesh

Chaals Nevile (chaals@entethalliance.org), EEA

Related work:

This specification is related to:

[baseline-core-v1.0] *Baseline Core Specification Version 1.0 - Part 1* Edited by Dr. Andreas Freund, Yoav Bittan, Keith Salzman, Chaal Neville, Anais Ofranc and Kyle Thomas. 17 April 2024. Project Specification Draft 01. https://docs.oasis-open.org/ethereum/baseline/baseline-core/v1.0/psd01/baseline-core-v1.0-psd01-part1.md . Latest stage: ps://docs.oasis-open.org/ethereum/baseline/baseline-core/v1.0/baseline-core-v1-part1.0.md.

[baseline-ccsm-v1.0] *Baseline CCSM Requirements Version 1.0 - Part 3* Edited by Dr. Andreas Freund, Yoav Bittan, Keith Salzman, Chaal Neville, Anais Ofranc and Kyle Thomas. 17 April 2024. Project Specification Draft 01. https://docs.oasis-open.org/ethereum/baseline/baseline-core/v1.0/psd01/baseline-core-v1.0-psd01-part3.md . Latest stage: https://docs.oasis-open.org/ethereum/baseline/baseline-core/v1.0/baseline-core-v1.0-part3.md .

Abstract:

This document describes the Baseline programming interface and expected behaviors of all instances of this interface together with the required programming interface data model.

Status:

This document is a Project Specification Draft and is no longer under active development.

Comments on this work can be provided by opening issues in the project repository or by sending email to the project's public comment lisbaseline@lists.oasis-open-projects.org.

Citation format:

When referencing this specification the following citation format should be used:

[baseline-api-v1.0] *Baseline API and Data Model Version 1.0 - Part 2* Edited by Dr. Andreas Freund, Yoav Bittan, Keith Salzman, Chaal Neville, Anais Ofranc and Kyle Thomas. 17 April 2024. Project Specification Draft 01. https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/psd01/baseline-api-v1.0-psd01-part2.md . Latest stage: https://docs.oasis-open.org/ethereum/baseline/baseline-api/v1.0/baseline-api-v1.0-part2.md .

Notices

Table of Contents

1 Introduction

The Baseline Protocol is an open-source initiative that combines advances in cryptography, messaging, and Consensus Controlled State Machines (CCSMs) often referred to as blockchains or distributed ledger technology (DLT) to deliver secure and private business processes at low cost – event ordering, data consistency, and workflow integrity. The Baseline Protocol provides a framework that allows Baseline Protocol Implementations (BPIs) to establish a common (business) frame of reference enabling confidential and complex (business) collaborations between enterprises without moving any sensitive data between traditional Systems of Record. The work is an Ethereum Community Project, which is managed by OASIS.

1.0 IPR Policy

The Baseline Protocol API Specification is provided under the RF on Limited Terms Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/legalruleml/ipr.php).

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

1.2 Normative References

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels" RFC 2119, DOI: 10.17487/RFC2119 March 1997https://www.rfc-editor.org/rfc/rfc2119.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005https://www.rfc-editor.org/info/rfc4122.

2 API Overview

1. Baseline
   - Description: Baseline Core provides internal integration middleware interfaces for baselining systems of record.
2. Consensus Controlled State Machine (CCSM)
   - Description: Baseline Core CCSM API provides interfaces for general interaction with the underlying CCSM.
3. Privacy
   - Description: Baseline Core Privacy provides interfaces supporting general consistency, zero-knowledge cryptography protocols and secure multi-party computation (MPC).
4. Registry
   - Description: Utilities for resolving Unique Identifiers such as DIDs to DID documents for arbitrary BPI Subjects.
5. Vault
   - Description: Baseline Core Vault API provides tools and methods for managing digital authentication credentials for BPI Subjects based on roles and BPI Workgroup instances.

3 Baseline

URL	Methods	Resource Type
{api-root}/bpi_accounts/	Get, Post	Objects
{api-root}/bpi_accounts/{id}	Get	State
{api-root}/protocol_messages/	Post	Objects
{api-root}/subjects/	Get, Post	Objects
{api-root}/subjects/{id}	Get, Put	Objects
{api-root}/subjects/{id}/accounts	Get, Post	Objects
{api-root}/subjects/{id}/accounts/{id}	Get	Objects
{api-root}/workflows/	Get, Post	Objects
{api-root}/workflows/{id}	Get	State
{api-root}/workflows/{id}/worksteps	Get	State
{api-root}/workflows/{id}/worksteps/{workstep_id}	Get	State
{api-root}/workgroups/	Get, Post	Objects
{api-root}/workgroups/{id}	Get, Put	Objects
{api-root}/workgroups/{id}/subjects	Get, Post	Objects

Testability Statement for all specification requirements: API endpoints can be tested using API testing platforms such asPostman. Since each API schema given in the requirements below specifies API responses using RFC 2616 HTMl error codes, and since each API schema has a given test fixture, all API specification requirements are testable.

3.1 BPI Accounts

Returns a list of BPI Account objects in the authorized scope.

[R1]

```
/bpi_accounts:
  get:
    deprecated: false
    description: Returns a list of `BPIAccount` objects in the authorized scope; _normative_
    operationId: listBPIAccounts
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/BPIAccount'
              type: array
        description: The request was successful and a list of BPI account objects has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: List BPI account objects
```

[R2]

Creates a BPI Account using the parameters provided in the request body.

```
  post:
    deprecated: false
    description: 'Creates a `BPIAccount` using the parameters provided in the request body; _normative_'
    operationId: createBPIAccount
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/BPIAccount'
          example:
            owners:
              - 'did:elem:7cb23e2b-07ed-4562-8afb-73955f8f17c5'
            security_policies:
              - type: AuthenticationPolicy
                reference: 'https://example.com/policies/authentication-policy.json'
            nonce: 4114
            workflows:
              $ref: '#/components/schemas/WorkflowInstance'
      required: true
    responses:
      '201':
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/BPIAccount'
        description: The request was successful and a new BPI account has been created.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Create BPI account
    tags:
      - Baseline
```

[R3]

Retrieves details for a specific BPI Account.

```
/bpi_accounts/{id}:
  get:
    deprecated: false
    description: Retrieves details for a specific `BPIAccount`; _normative_
    operationId: getBPIAccountDetails
    parameters:
      - description: ID of a specific `BPIAccount`; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
        example: '563cb9ed-01cb-49ec-9690-d779ea20f55f'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/BPIAccount'
              type: array
        description: The request was successful and detailed information about the specified BPI account has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Get BPI account details
    tags:
      - Baseline
```

3.2 Protocol Messages

[R4]

Creates a Baseline protocol message per the parameters provided in the request body.

```
/protocol_messages:
  post:
    summary: Create a Baseline protocol message
    description: Creates a Baseline protocol message per the parameters provided in the request body; _normative_
```

```
      operationId: createProtocolMessage
      parameters: []
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ProtocolMessagePayload'
        required: true
      responses:
        '201':
          description: The request was successful and a new protocol message was created.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ProtocolMessage'
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      deprecated: false
      servers:
        - url: http://localhost:8080
      tags:
        - Baseline
```

3.4 BPI Subjects

[R5]

Returns a list of BPI Subject instances in the authorized scope.

```
  /subjects:
    get:
      deprecated: false
      description: 'Returns a list of BPI `Subject` instances in the authorized scope; _normative_'
      operationId: listSubjects
      parameters:
        - $ref: '#/components/parameters/page'
        - $ref: '#/components/parameters/results-per-page'
      responses:
        '200':
          content:
            application/json:
              schema:
                items:
                  $ref: '#/components/schemas/Subject'
                type: array
          description: The request was successful and a list of subjects has been returned.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      servers:
        - url: http://localhost:8080
      summary: List subjects
      tags:
        - Baseline
```

[R6]

Creates a BPI Subject using the parameters provided in the request body.

```
    post:
      deprecated: false
      description: 'Creates a BPI `Subject` using the parameters provided in the request body; _normative_'
      operationId: createSubject
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Subject'
            example:
              wallet_id: '99c404e9-fe10-4ca7-b787-d5943d03591c'
              credentials: []
              description: Organization for testing
              metadata: {}
              name: ACME Inc.
              type: Organization
        required: true
      responses:
        '201':
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Subject'
          description: The request was successful and a new Subject has been created.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      servers:
        - url: http://localhost:8080
      summary: Create subject
      tags:
        - Baseline
```

[R7]

Retrieves details for a specified BPI `Subject.

```
/subjects/{id}:
  get:
    deprecated: false
    description: 'Retrieves details for a specified BPI `Subject`; _normative_'
    operationId: getSubjectDetails
    parameters:
      - description: 'ID of a specific `Subject` for which detailed information is desired'
        in: path
        name: id
        required: true
        schema:
          type: string
        example: 'did:elem:331156fb-fa5c-4557-ac39-69ae34c72c18'
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Subject'
        description: The request was successful and the details of the specified subject have been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Get subject details
    tags:
      - Baseline
```

[R8]

Updates a specified BPI `Subject` using the parameters provided in the request body.

```
/subjects/{id}:
  put:
    deprecated: false
    description: 'Updates a specified BPI `Subject` using the parameters provided in the request body; _normative_'
    operationId: updateSubjectDetails
    parameters:
      - description: 'ID of a specific `Subject` to be updated; _UUID as specified by RFC4122_'
        in: path
        name: id
        required: true
        schema:
          type: string
        example: 'did:elem:99c404e9-fe10-4ca7-b787-d5943d03591c'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Subject'
          example:
            wallet_id: '99c404e9-fe10-4ca7-b787-d5943d03591c'
            credentials: []
            description: Organization for testing
            metadata: {}
            name: ACME Inc.
            type: Orgnization
      required: true
    responses:
      '204':
        description: The request was successful and the specified subject has been updated.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Update subject details
    tags:
      - Baseline
```

[R9]

Returns a list of BPI SubjectAccount for a specified BPI Subject.

```
/subjects/{id}/accounts:
  get:
    deprecated: false
    description: 'Returns a list of BPI `SubjectAccount` for a specified BPI `Subject`; _normative_'
    operationId: listSubjectAccounts
    parameters:
      - description: 'ID of a specific `Subject` for which a list of  `SubjectAccount` objects is desired; _UUID as specified by RFC4122_'
        in: path
        name: id
        required: true
        schema:
          type: string
        example: 'did:elem:331156fb-fa5c-4557-ac39-69ae34c72c18'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/SubjectAccount'
              type: array
        description: The request was successful and a list of subject context objects has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: List subject contexts
    tags:
      - Baseline
```

[R10]

Creates a BPI SubjectAccount for a specified BPI Subject.

```
/subjects/{id}/accounts:
  post:
    deprecated: false
    description: 'Creates a BPI `SubjectAccount` for a specified BPI `Subject`; _normative_'
    operationId: createSubjectAccount
    parameters:
      - description: 'ID of a specific `Subject` to which the `SubjectAccount` is to be associated'
        in: path
        name: id
        required: true
        schema:
          type: string
        example: 'did:elem:331156fb-fa5c-4557-ac39-69ae34c72c18'
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/SubjectAccount'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/SubjectAccount'
        description: The request was successful and a list of subject context objects has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: List subject contexts
    tags:
      - Baseline
```

[R11]

Returns details of the BPI SubjectAccount when given a specified BPI Subject.

```
/subjects/{subject_id}/accounts/{id}:
  get:
    deprecated: false
    description: 'Returns details for a specific BPI `SubjectAccount` for a specified BPI `Subject`; _normative_'
    operationId: getSubjectAccountDetails
    parameters:
      - description: 'ID of a specific `SubjectAccount` for which a list of  `SubjectAccount` objects is desired'
        in: path
        name: id
        required: true
        schema:
          type: string
        example: 'did:elem:648142fb-fa5c-4557-ac39-69ae34c72c18'
      - description: 'ID of a specific `Subject` for which a list of `SubjectAccount` objects is desired'
        in: path
        name: subject_id
        required: true
        schema:
          type: string
        example: 'did:elem:331156fb-fa5c-4557-ac39-69ae34c72c18'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/SubjectAccount'
              type: object
        description: The request was successful and detailed information about the specified subject context has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Get subject context details
    tags:
      - Baseline
```

3.3 Workflows & Worksteps

[R12]

Returns a list of BPI Workflow instances in the authorized scope.

```
/workflows:
  get:
    description: Returns a list of BPI `Workflow` instances in the authorized scope; _normative_
    operationId: listWorkflows
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/WorkflowInstance'
              type: array
        description: The request was successful and a list of workflow instances has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: List workflow instances
    tags:
      - Baseline
```

[R13]

Creates a BPI WorkflowInstance using the parameters provided in the request body.

```
/workflows:
  post:
    description: Creates a BPI `WorkflowInstance` using the parameters provided in the request body; _normative_
    operationId: createWorkflow
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/WorkflowInstance'
          example:
            name: Procure to Pay
            type: procure_to_pay
    responses:
      '201':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/WorkflowInstance'
              type: array
        description: The request was successful and a new workflow instance has been created.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Create workflow instance
    tags:
      - Baseline
```

[R14]

Retrieves details for a specific BPI WorkflowInstance.

```
/workflows/{id}:
  get:
    description: Retrieves details for a specific BPI `WorkflowInstance`; _normative_
    operationId: getWorkflowDetails
    parameters:
      - description: ID of a specific `WorkflowInstance`; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
        example: '8def7853-245b-4743-b702-984417a8ba2f'
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/WorkflowInstance'
        description: The request was successful and detailed information about the specified `WorkflowInstance` has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Get workflow instance details
    tags:
      - Baseline
```

[R15]

Returns a list of BPI Workstep instances in a specific Workflow instance.

```
/workflows/{id}/worksteps:
  get:
    description: Returns a list of BPI `Workstep` instances in a specific `Workflow` instance; _normative_
    operationId: listWorksteps
    parameters:
      - description: ID of a specific `WorkflowInstance`; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/WorkstepInstance'
              type: array
        description: The request was successful and a list of workstep instances has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: List workstep instances
    tags:
      - Baseline
```

[R16]

Retrieves details for a specific BPI workstep instance in a specific BPIWorkflowInstance.

```
/workflows/{id}/worksteps/{workstep_id}:
  get:
    description: Retrieves details for a specific BPI workstep instance in a specific BPI `WorkflowInstance`; _normative_
    operationId: getWorkstepDetails
    parameters:
      - description: ID of a specific `WorkflowInstance`; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
        example: '8def7853-245b-4743-b702-984417a8ba2f'
      - description: ID of a specific `WorkstepInstance`; _UUID as specified by RFC4122_
        in: path
        name: workstep_id
        required: true
        schema:
          type: string
        example: '9299e803-5ed7-4618-b5d1-6a133dc67790'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/WorkstepInstance'
              type: object
            example:
              created_at: '2021-04-21T02:12:05.000Z'
              id: '2fffec13-590e-41eb-9a84-69ed881e0036'
              prover_id: 'dd615561-83e6-4f53-ac0f-aaf1749df139'
              require_finality: true
              workflow_id: 'f9227803-5541-4c13-a554-f6980f03362a'
              type: 'purchase_order'
        description: The request was successful and detailed information about the specified instance has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Get workstep instance details
    tags:
      - Baseline
```

3.4 Workgroups

[R17]

Returns a list of BPI Workgroup instances in the authorized scope.

```
/workgroups:
  get:
    deprecated: false
    description: Returns a list of BPI `Workgroup` instances in the authorized scope; _normative_
    operationId: listWorkgroups
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/Workgroup'
              type: array
        description: The request was successful and a list of workgroup instances has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: List workgroups
    tags:
      - Baseline
```

[R18]

Creates a BPI Workgroup using the parameters provided in the request body.

```
/workgroups:
  post:
    deprecated: false
    description: Creates a BPI `Workgroup` using the parameters provided in the request body; _normative_
    operationId: createWorkgroup
    requestBody:
      content:
        application/json:
          example:
            subject_id: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
            description: An example of the request body for workgroup creation
            name: Example workgroup
            network_id: 07102258-5e49-480e-86af-6d0c3260827d
            type: baseline
            security_policies: []
            admins:
              - 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
          schema:
            $ref: '#/components/schemas/Workgroup'
      required: true
    responses:
      '201':
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Workgroup'
        description: The request was successful and a new workgroup has been created.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Create workgroup
    tags:
      - Baseline
```

[R19]

Retrieves details for a specific BPI Workgroup.

```
/workgroups/{id}:
  get:
    description: Retrieves details for a specific BPI `Workgroup`; _normative_
    operationId: getWorkgroupDetails
    parameters:
      - description: ID of a specific `Workgroup` for which detailed information is desired; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Workgroup'
        description: The request was successful and detailed information about the specified workgroup has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Get workgroup details
    tags:
      - Baseline
```

[R20]

Updates a specific Workgroup using the parameters provided in the request body.

```
/workgroups/{id}:
  put:
    deprecated: false
    description: Updates a specific `Workgroup` using the parameters provided in the request body; _normative_
    operationId: updateWorkgroup
    parameters:
      - description: ID of a specific `Workgroup` to be updated; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
    requestBody:
      content:
        application/json:
          example:
            subject_id: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
            description: An example of the request body for updating a workgroup
            name: Example workgroup
            network_id: '07102258-5e49-480e-86af-6d0c3260827d'
            type: baseline
            security_policies: []
            admins:
              - 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
          schema:
            $ref: '#/components/schemas/Workgroup'
    responses:
      '204':
        description: The request was successful and the specified workgroup has been updated.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Update workgroup
    tags:
      - Baseline
```

[R21]

Returns a list of BPI Subject instances associated with a specific BPI Workgroup.

```
/workgroups/{id}/subjects:
  get:
    deprecated: false
    description: Returns a list of BPI `Subject` instances associated with a specific BPI `Workgroup`; _normative_
    operationId: listWorkgroupSubjects
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
      - description: ID of a specific `Workgroup`; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
        example: '41e89661-d1f0-48e0-9270-3df7a413bdf9'
    responses:
      '200':
        content:
          application/json:
            schema:
              items:
                $ref: '#/components/schemas/Subject'
              type: array
        description: The request was successful and a list of Subjects has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: List workgroup subjects
    tags:
      - Baseline
```

[R22]

Associates a specified BPI `Subject` with a specific BPI `Workgroup`. Note that the authorization should be scoped to the BPI `Workgroup` in which the BPI `Subject` is to be included.

```
/workgroups/{id}/subjects:
  post:
    deprecated: false
    description: |-
      Associates a specified BPI `Subject` with a specific BPI `Workgroup`; authorization should be scoped to the BPI `Workgroup` in which the BPI `Subject` is to be included. _normative_
    operationId: associateWorkgroupSubject
    parameters:
      - description: ID of a specific `Workgroup` for which a specific subject is to be associated; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
        example: '41e89661-d1f0-48e0-9270-3df7a413bdf9'
    requestBody:
      content:
        application/json:
          example:
            subject_id: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
    responses:
      '201':
        description: The request was successful and the specified subject has been associated with the specified workgroup.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    servers:
      - url: http://localhost:8080
    summary: Associate workgroup subject
    tags:
      - Baseline
```

4 Consensus Controlled State Machine (CCSM)

The CCSM interfaces enable interaction of the BPI with an underlying CCSM.

URL	Methods	Resource Type
{api-root}/contracts/	Get, Post	Objects

URL	Methods	Resource Type
{api-root}/contracts/{id}	Get	State
{api-root}/networks	Get	Object
{api-root}/networks/{id}	Get	Object
{api-root}/networks/{id}/status	Get	Object
{api-root}/transactions	Get, Post	Objects
{api-root}/transactions/{id}	Get	State
{api-root}/wallets	Get, Post	Objects
{api-root}/wallets/{id}/accounts	Get	State

4.1 Contracts

[O1]

Returns a list of smart contracts in the authorized scope.

```
/contracts:
  get:
    deprecated: false
    description: Returns a list of smart contracts in the authorized scope; _non-normative_
    operationId: listContracts
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
      - description: 'Flag to indicate if token contracts should be filtered from the response; default is true'
        explode: true
        in: query
        name: filter_tokens
        required: false
        schema:
          type: boolean
      - description: 'Indicates how the response should be sorted; currently, recent is the only supported option and returns by accessed_at descending.'
        explode: true
        in: query
        name: sort
        required: false
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              items:
                $ref: '#/components/schemas/Contract'
              type: array
        description: The request was successful and a list of smart contracts has been returned.
        headers:
          X-Total-Results-Count:
            $ref: '#/components/headers/X-Total-Results-Count'
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
      default:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error'
        description: unexpected error
    summary: List contracts
    tags:
      - CCSM
```

[O2]

Deploys a contract using the parameters provided in the request body.

```
/contracts:
  post:
    deprecated: false
    description: 'Deploys a contract using the parameters provided in the request body; _non-normative_'
    operationId: deployContract
    parameters: []
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Contract'
      required: true
    responses:
      '201':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/Contract'
        description: The request was successful and the specified contract has been deployed.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Deploy contract
    tags:
      - CCSM
```

[O3]

Retrieves details for a specified Contract.

```
  /contracts/{id}:
    get:
      deprecated: false
      description: 'Retrieves details for a specified `Contract`; _non-normative_'
      operationId: getContractDetail
      parameters:
        - description: 'ID of a specific `Contract` for which detailed information is desired; _UUID as specified by RFC4122_'
          in: path
          name: id
          required: true
          schema:
            type: string
      responses:
        '200':
          content:
            application/json; charset=UTF-8:
              schema:
                $ref: '#/components/schemas/Contract'
          description: The request was successful and detailed information about the specified contract has been returned.
          headers:
            X-Total-Results-Count:
              $ref: '#/components/headers/X-Total-Results-Count'
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
        default:
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Error'
          description: unexpected error
      summary: Get contract detail
      tags:
        - CCSM
```

4.2 Networks

[O4]

Returns a list of available peer-to-peer CCSM Network instances.

```
  /networks:
    get:
      deprecated: false
      description: Returns a list of available peer-to-peer CCSM `Network` instances; _non-normative_
      operationId: listNetworks
      parameters:
        - $ref: '#/components/parameters/page'
        - $ref: '#/components/parameters/results-per-page'
        - $ref: '#/components/parameters/public'
      responses:
        '200':
          content:
            application/json; charset=UTF-8:
              schema:
                items:
                  $ref: '#/components/schemas/Network'
                type: array
          description: The request was successful and a list of CCSM networks has been returned.
          headers:
            X-Total-Results-Count:
              $ref: '#/components/headers/X-Total-Results-Count'
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: List CCSM networks
      tags:
        - CCSM
```

[O5]

Retrieves the syncing status and, if fully-synced, real-time metrics and metadata of a specified CCSMNetwork.

```
/networks/{id}/status:
  get:
    deprecated: false
    description: 'Retrieves the syncing status and, if fully-synced, real-time metrics and metadata of a specified CCSM `Network`; _non-normative_'
    operationId: getNetworkStatus
    parameters:
      - description: 'ID of a specific CCSM `Network` for which the status is desired; _UUID as specified by RFC4122_'
        in: path
        name: id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/NetworkStatus'
        description: The request was successful and the status of the specified CCSM network has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
      default:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Error'
        description: unexpected error
    summary: Get CCSM network status
    tags:
      - CCSM
```

4.3 Transactions

[R23]

Returns a list of BPI Transaction instances in the authorized scope.

```
/transactions:
  get:
    deprecated: false
    description: Returns a list of BPI `Transaction` instances in the authorized scope; _normative_
    operationId: listTransactions
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
      - description: When true, contract creation transactions are filtered
        explode: true
        in: query
        name: filter_contract_creations
        required: false
        schema:
          type: boolean
      - description: Comma-delimited list of statuses by which the transactions response is filtered
        explode: true
        in: query
        name: status
        required: false
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              items:
                $ref: '#/components/schemas/Transaction'
              type: array
        description: The request was successful and a list of transactions has been returned.
        headers:
          X-Total-Results-Count:
            $ref: '#/components/headers/X-Total-Results-Count'
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: List transactions
    tags:
      - CCSM
```

[R24]

Creates and broadcast a BPI transaction in a chain- and protocol-agnostic manner using the parameters provided in the request body.

```
/transactions:
  post:
    deprecated: false
    description: >
      Creates and broadcast a BPI transaction in a chain- and protocol-agnostic manner using the parameters provided in the request body; _normative_
    operationId: createTransaction
    parameters: []
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Transaction'
      required: true
    responses:
      '201':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/Transaction'
        description: The request was successful and a new transaction has been created.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Create transaction
    tags:
      - CCSM
```

[R25]

Retrieves details for a specified BPI `Transaction`.

```
/transactions/{id}:
  get:
    deprecated: false
    description: Retrieves details for a specified BPI `Transaction`; _normative_
    operationId: getTransactionDetails
    parameters:
      - description: ID of a specific `Transaction` for which detailed information is desired; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/Transaction'
        description: The request was successful and detailed information about the specified transaction has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Get transaction details
    tags:
      - CCSM
```

4.4 Wallets

[O6]

Returns a list of Wallet instances.

```
  /wallets:
    get:
      deprecated: false
      description: Returns a list of `Wallet` instances; _non-normative_
      operationId: listWallets
      parameters:
        - $ref: '#/components/parameters/page'
        - $ref: '#/components/parameters/results-per-page'
      responses:
        '200':
          content:
            application/json; charset=UTF-8:
              schema:
                items:
                  $ref: '#/components/schemas/Wallet'
                type: array
          description: The request was successful and a list of wallets has been returned.
          headers:
            X-Total-Results-Count:
              $ref: '#/components/headers/X-Total-Results-Count'
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: List wallets
      tags:
        - CCSM
```

[O7]

Creates a Wallet using the parameters provided in the request body. AWallet may be setup as custodial or non-custodial. If the Wallet is custodial then the platform will derive addresses and securely persist an BPI Account for each of those derived addresses.

```
  /wallets:
    post:
      deprecated: false
      description: |
        Creates a `Wallet` using the parameters provided in the request body. A `Wallet` may be setup as custodial or non-custodial. If the Wallet is custodial then the platform will derive addresses and securely persist an `Account` for each of those derived addresses. _non-normative_
      operationId: createWallet
      parameters: []
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Wallet'
        required: true
      responses:
        '201':
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Wallet'
          description: The request was successful and a new wallet has been created.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: Create wallet
      tags:
        - CCSM
```

[O8]

Returns a list of Account instances for a specific Wallet.

```
/wallets/{id}/accounts:
  get:
    deprecated: false
    description: Returns a list of `Account` instances for a specific `Wallet`; _non-normative_
    operationId: listWalletAccounts
    parameters:
      - $ref: '#/components/parameters/results-per-page'
      - description: ID of a specific `Wallet` containing `Account` instances; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              items:
                $ref: '#/components/schemas/BPIAccount'
              type: array
        description: The request was successful and a list of accounts has been returned.
        headers:
          X-Total-Results-Count:
            $ref: '#/components/headers/X-Total-Results-Count'
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: List wallet accounts
    tags:
      - CCSM
```

5 Privacy

URL	Methods	Resource Type
{api-root}/provers/	Get, Post	Objects
{api-root}/provers/{id}	Get	State
{api-root}/provers/{id}/prove	Post	Object
{api-root}/provers/{id}/verify	Post	Object

5.1 Provers

[R26]

Returns a list of (cryptographic) Prover instances in the authorized scope.

```
/provers:
  get:
    deprecated: false
    description: 'Returns a list of `Prover` instances in the authorized scope; _normative_'
    operationId: listProvers
    parameters:
      - description: 'Filter the response by elliptic curve'
        explode: false
        in: query
        name: curve
        required: false
        schema:
          type: string
      - description: 'Filter the response by identifier'
        explode: false
        in: query
        name: identifier
        required: false
        schema:
          type: string
      - description: 'Filter the response by provider'
        explode: false
        in: query
        name: provider
        required: false
        schema:
          type: string
      - description: 'Filter the response by proving scheme'
        explode: false
        in: query
        name: proving_scheme
```

```
          required: false
          schema:
            type: string
        - description: 'Filter the response by the status'
          explode: false
          in: query
          name: status
          required: false
          schema:
            type: string
      responses:
        '200':
          content:
            application/json; charset=UTF-8:
              schema:
                items:
                  $ref: '#/components/schemas/Prover'
                type: array
          description: The request was successful and a list of provers has been returned.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: List provers
      tags:
        - Privacy
```

[R27]

Creates a BPI Prover using the parameters provided in the request body.

Creating a prover entails compiling the prover from source, performing the appropriate setup or multiparty key ceremony and securely persisting resulting artifacts.

A Prover must be provisioned before it can be used to generate and verify proofs. The foregoing holds true regardless of which provider, proving_scheme, curve and prover identifier you specify (or source, if the raw source code of the Prover is provided instead of an identifier).

A persistent store must be implicitly initialized upon the creation of a new prover unless an existing identifier (store_id) is provided and the referenced store is valid in the context of the prover and authorized scope.

The parameters required to provision a Prover vary across providers and proving schemes.

```
  /provers:
    post:
      deprecated: false
      description: |-
        Creates a BPI `Prover` using the parameters provided in the request body.

        Creating a prover entails compiling the prover from source, performing the appropriate setup or multiparty key ceremony and securely persisting resulting artifacts.

        A `Prover` must be provisioned before it can be used to generate and verify proofs. The foregoing holds true regardless of which provider, proving_scheme, curve and prover identifier you specify (or source, if the raw source code of the `Prover` is provided instead of an identifier).

        A persistent store must be implicitly initialized upon the creation of a new prover unless an existing identifier (store_id) is provided and the referenced store is valid in the context of the prover and authorized scope.

        The parameters required to provision a `Prover` vary across providers and proving schemes. _normative_
      operationId: createProver
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Prover'
        required: true
      responses:
        '201':
          description: The request was successful and a new prover has been created.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: Create prover
      tags:
        - Privacy
```

[R28]

Retrieves details for a specific Prover.

```
  /provers/{id}:
    get:
      deprecated: false
      description: 'Retrieves details for a specific `Prover`; _normative_'
      operationId: getProverDetails
      parameters:
        - description: 'ID of a specific `Prover`; _UUID as specified by RFC4122_'
          in: path
          name: id
          required: true
          schema:
            format: uuid
            type: string
      responses:
        '200':
          content:
            application/json; charset=UTF-8:
              schema:
                $ref: '#/components/schemas/Prover'
          description: The request was successful and detailed information about the specified prover has been returned.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: Get prover details
      tags:
        - Privacy
```

[R29]

Generates a cryptographic proof given a valid witness parameter; calling this API has an implicit side-effect of writing to the configured Store.

```
  /provers/{id}/prove:
    post:
      deprecated: false
      description: |-
        Generates a cryptographic proof given a valid witness parameter; calling this API has an implicit side-effect of writing to the configured `Store`. _normative_
      operationId: generateProof
      parameters:
        - description: ID of a specific `Prover`; _UUID as specified by RFC4122_
          in: path
          name: id
          required: true
          schema:
            type: string
      requestBody:
        content:
          application/json:
            schema:
              properties:
                identifier:
                  type: string
                proving_scheme:
                  description: Proving scheme to be used
                  example: groth16
                  readOnly: false
                  type: string
                curve:
                  description: Pairing-friendly elliptic curve
                  example: BN254
                  readOnly: false
                  type: string
                provider:
                  description: Circuit provider
                  example: gnark
                  readOnly: false
                  type: string
                name:
                  type: string
                  description: 'User-defined name for proof'
                store_id:
                  description: 'ID of `Store` that will accept the generate proof'
                  format: uuid
                  type: string
                witness:
                  $ref: '#/components/schemas/Witness'
              type: object
        required: true
      responses:
        '200':
          description: The request was successful and a new proof has been created.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
    '422':
     description: The API returns a 422 status code and human-readable error message(s) if (i) the witness parameters is not provided, (ii) required prover arguments (i.e., fields) are not included within the specific witness parameter or (iii) when Prover constraints are not satisfied.
    '429':
     $ref: '#/components/responses/TooManyRequests'
    '500':
     $ref: '#/components/responses/InternalServerError'
    '503':
     $ref: '#/components/responses/ServiceUnavailable'
   summary: Generate proof
   tags:
    - Privacy
```

[R30]

Verifies a Proof using the Witness parameters provided in the request body.

```
 /provers/{id}/verify:
  post:
   deprecated: false
   description: Verifies a `Proof` using the `Witness` parameters provided in the request body; _normative_
   operationId: verify
   parameters:
    - description: ID of a specific `Prover`; _UUID as specified by RFC4122_
     in: path
     name: id
     required: true
     schema:
      type: string
   requestBody:
    content:
     application/json:
      schema:
       $ref: '#/components/schemas/StateProof'
    required: true
   responses:
    '201':
     description: The request was successful and a verification result has been returned.
    '401':
     $ref: '#/components/responses/Unauthorized'
    '403':
     $ref: '#/components/responses/Forbidden'
    '404':
     $ref: '#/components/responses/NotFound'
    '429':
     $ref: '#/components/responses/TooManyRequests'
    '500':
     $ref: '#/components/responses/InternalServerError'
    '503':
     $ref: '#/components/responses/ServiceUnavailable'
   summary: Verify proof
   tags:
    - Privacy
```

6 Registry

URL	Methods	Resource Type
{api-root}/resolve/{did}	Get	object

6.1 Resolve Identifier

[O9]

Endpoint to resolve an identifier, DID, to it associated key material document, DID document, forSubject instances.

```
 /resolve/{did}:
  get:
   description: Endpoint to resolve an identifier, DID, to it associated key material document, DID document, for `Subject` instances; _non-normative_
   operationId: resolveDID
   parameters:
    - description: ID of a specific DID to be resolved
     in: path
     name: did
     required: true
     schema:
      type: string
     example: 'did:elem:8def7853-245b-4743-b702-984417a8ba2f'
   responses:
    '200':
     description: A resolved DID
     content:
      application/json; charset=UTF-8:
       schema:
        type: object
        properties:
         '@context':
          type: array
          description: List of @contexts
          items: {}
         id:
          type: string
          description: ID of DID
         verificationMethod:
          type: array
          description: Details regarding the verification method used to resolve DID
          items:
           type: object
           properties:
            controller:
             type: string
             description: ID of DID controller
            id:
             type: string
             description: ID of DID
            publicKeyJWK:
             type: object
             description: Public key for JWK
             properties:
              kid:
               type: string
              n:
               type: string
              e:
               type: string
              fingerprint:
               type: string
            public_key:
             type: string
         type:
          type: string
          description: Type of DID context
         assertionMethod:
          type: string
          description: Assertion method used
         authentication:
          type: string
         capabilityInvocation:
          type: string
         capabilityDelegation:
          type: string
         keyAgreement:
          type: string
       example:
        '@context':
         - 'https://www.w3.org/ns/did/v1'
         - 'https://w3id.org/security/suites/jws-2020/v1'
        id: 'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a'
        verificationMethod:
         - controller: 'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a'
          id: 'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a#a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
          publicKeyJwk:
           kid: 'a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
           'n': >-

86148399721108856798667922918184805408960463864134465179962941121913234801627871709210212868113561138032675209893636335349544406320216202967352355777378204802411

           e: '10001'
           fingerprint: 'a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
           public_key: |-
            -----BEGIN PUBLIC KEY-----
            MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA0yqdJMBej1X8WwMMkMZw
            DV6zZzup4RHLcln0xfGSm6dMPBDM1G96fuHhOwH5+uU5MQHJP7RqW71Bu5dLIG8Z
            RX+XyUtb0sxCV/7X27Nm/bKpDysaSWQ36reAmw5wVaB1SoFeN519FY5rhoCWmH3W
            auBAHTzpjg57p7uR0XynYXf8NSGXlysWHppkppqwrPH64G6UZaB7SMl1PFfkJeqZ
            zJpzBGYWsixdF1EjXn+Yz0mhUZO2OSPWifOuN7cpn3BuNqegg4iVdz5HDoQhJW7N
            uRhf3buKd/mjat8XA3e2Rkrr2h835GloScJkj7I4BZUNkzKQuEK6C9xW/zJtbPqQ
            RYEq84A1hMfSZ3G5HFe2JkqiyvXkFwS3qMc5Pur8tZSzBj6AYMoJJso/aOdphpR8
            6MaaWXWTwvwfpZbMRqehOcsmQcNLF2gLJPuHzR5WtVCnWrDgvjsWyeDD1WISKusi
            aOeHxZjS3Bjl4Imq48l1wi2eI/11F/Xg70F4FJaMYLVHJA2nsmBuuQ9UDYHHq876
            clKvIvgIItzJcv9lnmjl1Jks1DwCUF3qF2ugYcs9A3EoEcNzhMgZNJ2j5OUzfx1E
            bzVKkqoC9MQpZWXgqV0KQqKK4I3rMY+1hLqk4S4eF9ZAVlT33qfMzlf0qWTOcP1Z
            i2dsm0fy4NxWxknlEn5/LhMCAwEAAQ==
            -----END PUBLIC KEY-----
        type: JsonWebKey2020
        assertionMethod:
         'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a#a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
        authentication:
         'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a#a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
        capabilityInvocation:
         'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a#a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
        capabilityDelegation:
         'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a#a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
        keyAgreement:
         'did:elem:54dc7f20-f2ae-49ff-93ab-c3857c64600a#a3:c2:e5:17:c2:19:26:3c:fa:a2:2c:38:7b:ca:0c:60'
   summary: Retrieve DID document
   tags:
    - Registry
```

7 Vault

URL	Methods	Resource Type
{api-root}/unseal	Post	Object
{api-root}/vaults	Get, Post	Objects

URL	Methods	Resource Type
{api-root}/vaults/{id}/keys	Get, Post	Objects
{api-root}/vaults/{id}/keys/{key_id}	Delete	Object
{api-root}/vaults/{id}/keys/{key_id}/derive	Post	Object
{api-root}/vaults/{id}/secrets	Get, Post	Objects
{api-root}/vaults/{id}/secrets/{secret_id}	Get, Delete	Objects

7.1 Vault Management

[O10]

Unseals a Vault and enables additional interaction within the authorized scope.

```
/unseal:
  post:
    deprecated: false
    description: >
      Unseals a `Vault` and enables additional interaction within the authorized scope. _non-normative_
    operationId: unsealVault
    parameters: []
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/UnsealVault'
      required: true
    responses:
      '204':
        description: The request was successful and the corresponding vault has been unsealed.
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Unseal vault
    tags:
      - Vault
```

[O11]

Returns a list of Vault instances in the authorized scope.

```
/vaults:
  get:
    deprecated: false
    description: Returns a list of `Vault` instances in the authorized scope; _non-normative_
    operationId: listVaults
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              items:
                $ref: '#/components/schemas/Vault'
              type: array
        description: The request was successful and a list of vaults has been returned.
        headers:
          X-Total-Results-Count:
            $ref: '#/components/headers/X-Total-Results-Count'
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: List vaults
    tags:
      - Vault
```

[O12]

Creates a Vault instance using the parameters provided in the request body.

```
/vaults:
  post:
    deprecated: false
```

```
      description: Creates a `Vault` instance using the parameters provided in the request body
      operationId: createVault
      parameters: []
      requestBody:
        content:
          application/json:
            example:
              description: Example vault description
              name: Example vault name
            schema:
              $ref: '#/components/schemas/Vault'
        required: true
      responses:
        '201':
          content:
            application/json; charset=UTF-8:
              schema:
                $ref: '#/components/schemas/Vault'
          description: The request was successful and a new vault has been created.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: Create vault
      tags:
        - Vault
```

[O13]

Returns a list of keys in a specific Vault.

```
  /vaults/{id}/keys:
    get:
      deprecated: false
      description: Returns a list of keys in a specific `Vault`; _non-normative_
      operationId: listKeys
      parameters:
        - $ref: '#/components/parameters/page'
        - $ref: '#/components/parameters/results-per-page'
        - description: ID of a specific `Vault` that contains the desired keys ; _UUID as specified by RFC4122_
          in: path
          name: id
          required: true
          schema:
            type: string
      responses:
        '200':
          content:
            application/json; charset=UTF-8:
              schema:
                items:
                  $ref: '#/components/schemas/Key'
                type: array
          description: The request was successful and a list of keys for the specified vault has been returned.
          headers:
            X-Total-Results-Count:
              $ref: '#/components/headers/X-Total-Results-Count'
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: List keys
      tags:
        - Vault
```

[O14]

Creates a Key in a specific Vault.

```
  /vaults/{id}/keys:
    post:
      deprecated: false
      description: Creates a `Key` in a specific `Vault`; _non-normative_
      operationId: createKey
      parameters:
        - description: ID of a specific `Vault` in which a desired `Key` is to be created; _UUID as specified by RFC4122_
          in: path
          name: id
          required: true
          schema:
            type: string
      requestBody:
        content:
          application/json:
            example:
              description: Secure channel for messaging
              name: Private chat
              spec: ChaCha20
              type: symmetric
              usage: encrypt/decrypt
              vault_id: ae3c07d9-8a41-4cd3-92c0-e6358cbd516c
            schema:
              $ref: '#/components/schemas/Key'
        required: true
      responses:
        '201':
          content:
            application/json; charset=UTF-8:
              schema:
                $ref: '#/components/schemas/Key'
          description: The request was successful and a new key has been created.
        '401':
          $ref: '#/components/responses/Unauthorized'
        '403':
          $ref: '#/components/responses/Forbidden'
        '404':
          $ref: '#/components/responses/NotFound'
        '429':
          $ref: '#/components/responses/TooManyRequests'
        '500':
          $ref: '#/components/responses/InternalServerError'
        '503':
          $ref: '#/components/responses/ServiceUnavailable'
      summary: Create a key
      tags:
        - Vault
```

[O15]

Deletes a specific `Key` in a specific `Vault`.

```
/vaults/{id}/keys/{key_id}:
  delete:
    deprecated: false
    description: Deletes a specific `Key` in a specific `Vault`; _non-normative_
    operationId: deleteKey
    parameters:
      - description: ID of a specific `Vault` that contains a specific `Key` to be deleted; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
      - description: ID of a specific `Key` to be deleted; _UUID as specified by RFC4122_
        in: path
        name: key_id
        required: true
        schema:
          type: string
    responses:
      '204':
        description: The request was successful and the specified key has been deleted.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Delete a key
    tags:
      - Vault
```

[O16]

Creates a derived key from a specific `Key` in a specific `Vault`.

```
/vaults/{id}/keys/{key_id}/derive:
  post:
    deprecated: false
    description: Creates a derived key from a specific `Key` in a specific `Vault`; _non-normative_
    operationId: deriveKeyRequest
    parameters:
      - description: ID of a specific `Vault` in which a derived key is to be created; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
      - description: ID of a specific `Key` to be used to create derived key; _UUID as specified by RFC4122_
        in: path
        name: key_id
        required: true
        schema:
          type: string
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/DerivedKey'
      required: true
    responses:
      '201':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/DerivedKey'
        description: The request was successful and a new key has been derived.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Creates a derived key
    tags:
      - Vault
```

[O17]

Returns a list of secrets stored in a specific Vault.

```
/vaults/{id}/secrets:
  get:
    deprecated: false
    description: Returns a list of secrets stored in a specific `Vault`; _non-normative_
    operationId: listSecrets
    parameters:
      - $ref: '#/components/parameters/page'
      - $ref: '#/components/parameters/results-per-page'
      - description: ID of a specific `Vault` in which the desired secrets are stored; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              items:
                $ref: '#/components/schemas/Vault'
              type: array
        description: The request was successful and a list of secrets has been returned.
        headers:
          X-Total-Results-Count:
            $ref: '#/components/headers/X-Total-Results-Count'
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: List secrets
    tags:
      - Vault
```

[O18]

Stores a Secret in a specific Vault.

```
/vaults/{id}/secrets:
  post:
    deprecated: false
    description: Stores a `Secret` in a specific `Vault`; _non-normative_
    operationId: storeSecret
    parameters:
      - description: ID of a specific `Vault` in which to store a specific `Secret`; _UUID as specified by RFC4122_
```

```
        in: path
        name: id
        required: true
        schema:
          type: string
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Secret'
      required: true
    responses:
      '201':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/Secret'
        description: The request was successful and the secret has been stored.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Store secret
    tags:
      - Vault
```

[O19]

Deletes a specific Secret from a specific Vault.

```
/vaults/{id}/secrets/{secret_id}:
  delete:
    deprecated: false
    description: Deletes a specific `Secret` from a specific `Vault`; _non-normative_
    operationId: deleteSecret
    parameters:
      - description: ID of a specific `Vault` that contains a `Secret` to be deleted; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
      - description: ID of a specific `Secret` to be deleted; _UUID as specified by RFC4122_
        in: path
        name: secret_id
        required: true
        schema:
          type: string
    responses:
      '204':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/Secret'
        description: The request was successful and the specified secret has been deleted.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Delete secret
    tags:
      - Vault
```

[O20]

Retrieves a specific Secret from a specific Vault.

```
/vaults/{id}/secrets/{secret_id}:
  get:
    deprecated: false
    description: Retrieves a specific `Secret` from a specific `Vault`; _non-normative_
    operationId: retrieveSecret
    parameters:
      - description: ID of a specific `Vault` in which a specific `Secret` is stored; _UUID as specified by RFC4122_
        in: path
        name: id
        required: true
        schema:
          type: string
      - description: ID of a specific `Secret` to be retrieved; _UUID as specified by RFC4122_
        in: path
        name: secret_id
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json; charset=UTF-8:
            schema:
              $ref: '#/components/schemas/Secret'
        description: The request was successful and the specified secret has been returned.
      '401':
        $ref: '#/components/responses/Unauthorized'
      '403':
        $ref: '#/components/responses/Forbidden'
      '404':
        $ref: '#/components/responses/NotFound'
      '429':
        $ref: '#/components/responses/TooManyRequests'
      '500':
        $ref: '#/components/responses/InternalServerError'
      '503':
        $ref: '#/components/responses/ServiceUnavailable'
    summary: Retrieve secret
    tags:
      - Vault
```

8 API Components

```
components:
  headers:
    X-Total-Results-Count:
      schema:
        description: Number of total query hits across all pages
        example: 14
        type: integer
  parameters:
    page:
      description: Limits resulting response to the page number specified
      in: query
      name: page
      required: false
      schema:
        type: integer
    public:
      description: When `TRUE`, limits results to non-private records
      in: query
      name: public
      required: false
      schema:
        type: boolean
    results-per-page:
      description: Sets the number of results included per page
      in: query
      name: rpp
      required: false
      schema:
        type: integer
  responses:
    Accepted:
      description: The request has been accepted for processing, but the processing has not been completed
    Forbidden:
      content:
        application/json:
          example:
            code: 403
            message: The supplied API Authorization header does not have permission to access the requested resource. Please verify that you have sufficient permissions and try again
          schema:
            properties:
              code:
                type: integer
              message:
                type: string
            type: object
      description: Forbidden
    InternalServerError:
      content:
        application/json:
          example:
            code: 500
            message: The requested resource is not implemented by the platform
          schema:
            properties:
              code:
                type: integer
              message:
                type: string
            type: object
      description: Not Implemented
    NoContent:
```

```
    description: The request was successful but did not return a response
  NotFound:
    content:
      application/json:
        example:
          code: 404
          message: Platform did not find the requested resource
        schema:
          properties:
            code:
              type: integer
            message:
              type: string
          type: object
    description: Not Found
  OK:
    description: The request was successful
  ServiceUnavailable:
    content:
      application/json:
        example:
          code: 503
          message: The request cannot be fulfilled due to temporary unavailability of a backend service
        schema:
          properties:
            code:
              type: integer
            message:
              type: string
          type: object
    description: Service Unavailable
  Success:
    description: The request was successful and a new entity was created
  TooManyRequests:
    content:
      application/json:
        schema:
          example:
            code: 429
            message: The request was not accepted due to exceeding the rate limit
          properties:
            code:
              type: integer
            message:
              type: string
          type: object
    description: Too many requests
  Unauthorized:
    content:
      application/json:
        example:
          code: 401
          message: The request required an API Authorization header, or one was provided which could not be authenticated
        schema:
          properties:
            code:
              type: integer
            message:
              type: string
          type: object
    description: Unauthorized
```

9 API Data Schemas

9.1 BPI Account

[R31]

The account associated with the (commercial) state of an agreement between counterparties, which must only be changed as a result of a valid BPI transaction (i.e., a successful BPI Workstep execution) by way of a StateClaimas the new (commercial) state.

```
  BPIAccount:
    description: The account associated with the (commercial) state of an agreement between counterparties, which must only be changed as a result of a valid BPI transaction (i.e., a successful 
BPI `Workstep` execution) by way of a `StateClaim`as the new (commercial) state; _normative_
    properties:
      '@context':
        description: Context reference file for SubjectAccount schema validation
        example: []
        type: object
        readOnly: false
      balances:
        description: 'Balances for the specified `BPIAccount`'
        type: object
        example: {}
        readOnly: true
      created_at:
        description: Date and time of `BPIAccount` creation
        example: '2021-02-27T14:26:06.864Z'
        format: date-time
        readOnly: true
        type: string
      owners:
        description: Must be an array of either a DID or another resolvable unique identifier of the `Subject`
        example:
          - 'did:elem:7cb23e2b-07ed-4562-8afb-73955f8f17c5'
        type: array
        items:
          type: string
          properties:
            subject_id:
              description: ID of a specific `Subject`
              type: string
        readOnly: false
      id:
        description: ID of the `BPIAccount`; _UUID as specified by RFC4122_
        example: 6bb23e2b-07ed-4562-8afb-73955f8f17c5
        format: uuid
        readOnly: true
        type: string
      metadata:
        description: Metadata useful for BPI operations and UI implementations
        example: {}
        type: object
        readOnly: false
      nonce:
        description: incrementing deterministic counter starting at zero
        example: 4512
        format: uint64
        readOnly: false
        type: integer
      security_policies:
        description: Security policies attached to a `BPIAccount` typically connected to a role but may be independent
        example:
          type: 'SecurityPolicy'
          reference: ''
        type: array
        items:
            type: object
            required:
              - type
              - reference
            properties:
              type:
                description: Type of security policy
                example: 'AuthenticationPolicy'
                type: string
              reference:
                description: Resolvable reference link to the security policy object, avoids duplicating policy objects
                example: 'https://example.com/policies/authentication-policy.json'
                type: string
                format: URI
        readOnly: false
      state_claims:
        description: '`StateClaim` instances associated with the `BPIAccount'
        example:
        readOnly: false
        type: array
        items:
          $ref: "#/components/schemas/StateClaim"
      workflows:
        description: List of `WorkflowInstance` objects related to the specified `BPIAccount`
        example: []
        readOnly: false
        type: array
        items:
          $ref: "#/components/schemas/WorkflowInstance"
    required:
      - owners
      - nonce
    title: BPI Account
    type: object
```

9.2 Contract

[O31]

Smart contract object.

```
    Contract:
      description: Smart contract object; _non-normative_
      example:
        compiled_artifact: {}
        accessed_at: '2021-02-28T00:41:44.506+0000'
        address: '0xD3F14c012a8717FF7b5D5AC08C287de58C38D8ba'
        created_at: '2021-02-27T14:26:06.864Z'
        id: 84a91108-70db-43c5-9f2f-e1655b9e0b03
        name: Shuttle
        network_id: 07102258-5e49-480e-86af-6d0c3260827d
        params:
          abi:
            anonymous: false
            inputs:
              indexed: false
              internal_type: bytes
              name: message
              type: bytes
            name: Ekho
            type: event
          wallet_id: df83bcbb-ff5b-4c4c-b1f5-8ca6a61fe460
          bytecode: 0x
          contract_name: Ekho
          outputs: []
          source: 0x
          state_mutability: nonpayable
          pubsub_prefix: 57478604c134ef20b94dff84e52c0fd562ddae56cd50431339766819a7523d09
          transaction_id: 4c388cca-f11c-4818-96a0-e8ad0b30553d
        workgroup_id: dc767890-0bbc-4c1d-bc04-f5842330018f
        type: registry
      properties:
        type:
          description: Type of contract deployed
          example: 'RegistrySmartContract'
          readOnly: true
          type: string
        compiled_artifact:
          description: Object for the deployed contract; implementation specific based on chosen contract_type
          example: {}
          readOnly: true
          type: object
        params:
          type: object
          description: Parameters associated with the `Contract`
          example: {}
          readOnly: false
      required:
          - type
          - compiled_artifact
      title: Contract
      type: object
```

9.3 Derived Keys

[O32]

A cryptographic Key generated from a password or master key.

```
DerivedKey:
  description: A cryptographic `Key` generated from a password or master key; _non-normative_
  example:
    context: channel-6852386c-8a3d-41c6-aa0e-766a31a8faaf
    description: this is a secure channel
    name: private chat
    nonce: 4512
  properties:
    context:
      description: Machine-readable string describing the key derivation context
      example: ''
      readOnly: false
      type: string
    description:
      description: User-defined description of derived `Key`
      example: Example description for example name
      readOnly: false
      type: string
    name:
      description: User-defined name of derived `Key`
      example: Example name
      readOnly: false
      type: string
    nonce:
      description: 'Random 32-bit integer or incrementing counter which must only be used once to avoid exposing the underlying secret; if not provided, a random 32-bit integer is used.'
      example: 4512
      format: int32
      readOnly: false
      type: integer
  required:
    - context
    - description
    - name
  title: Derived key
  type: object
```

9.4 Errors

[R31]

A human-readable description of an error.

```
Error:
  example:
    code: 400
    message: The request was successful
  properties:
    code:
      description: Number corresponding to a specific error
      example: 404
      readOnly: true
      type: integer
    message:
      description: A human-readable description of an error
      example: The request was successful
      readOnly: true
      type: string
  required:
    - code
    - message
  title: Error
  type: object
```

9.5 Keys

[O33]

A string of data that is used to lock or unlock cryptographic functions.

```
Key:
  description: A string of data that is used to lock or unlock cryptographic functions; _non-normative_
  example:
    created_at: '2021-04-17T21:07:16.098Z'
    description: Secure channel for messaging
    id: 02e81595-ef57-4e71-8e27-1c8b2dfdefbf
    name: Private chat
    spec: ChaCha20
    type: symmetric
    usage: encrypt/decrypt
    vault_id: 'ae3c07d9-8a41-4cd3-92c0-e6358cbd516c'
  properties:
    description:
      description: User-defined description of `Key` to be generated
      example: Example description
      readOnly: false
      type: string
    id:
      description: 'ID of a specific `Key`. _UUID as specified by RFC4122_'
      example: 6bb23e2b-07ed-4562-8afb-73955f8f17c5
      format: uuid
      maxLength: 36
      minLength: 1
      readOnly: true
      type: string
    name:
      description: User-defined name of `Key` to be generated
      example: Example name
      readOnly: false
      type: string
    public_key:
      description: The public key for a specific `Key`
      readOnly: true
      type: string
    spec:
      description: Specification to be used for `Key` creation
      example: BIP39
      readOnly: false
      type: string
    type:
      description: Type of `Key` being created
      enum:
        - asymmetric
        - symmetric
      example: symmetric
      readOnly: false
      type: string
    usage:
      description: Purpose for `Key` being created
      enum:
        - encrypt/decrypt
        - sign/verify
      example: encrypt/decrypt
      readOnly: false
      type: string
    vault_id:
      description: ID of a specific `Vault` in which a specific `Key` is stored. _UUID as specified by RFC4122_
      example: acb91fd4-b377-4732-9c5c-220c2305e665
      format: uuid
      maxLength: 36
      minLength: 1
      readOnly: false
      type: string
  readOnly: true
  required:
    - name
    - spec
  title: Key
  type: object
```

9.5 Network

[O34]

A CCSM Network object.

```
Network:
  description: CCSM Network object; _non-normative_
  example:
    type: Provide Network
    specification: ''
    chain_id: 8aaf1468-9cc1-4735-8975-aa3909178482
    config: {}
    block_explorer_url: https://etherscan.io
    chainspec_url: https://gist.githubusercontent.com/kthomas/3ac2e29ee1b2fb22d501ae7b52884c24/raw/161c6a9de91db7044fb93852aed7b0fa0e78e55f/mainnet.chainspec.json
    engine_id: ethash
    is_ethereum_network: true
    json_rpc_url: https://node.example.com
    native_currency: ETH
    network_id: 324cf211-6d67-4cae-b26c-c424308eabfa
    platform: evm
    protocol_id: pow
    security:
      egress: '*'
      ingress:
        0.0.0.0/0:
          tcp:
            - 8050
            - 8051
            - 30300
          udp:
            - 30300
    websocket_url: wss://node.example.com
    created_at: '2018-01-13T22:00:47.947907Z'
    description: Ethereum mainnet
    enabled: true
    id: deca2436-21ba-4ff5-b225-ad1b0b2f5c59
  required:
    - type
    - specification
  properties:
    type:
      description: Type of CCSM `Network`
      type: string
      example: ''
      readOnly: false
    specification:
      description: Specification used for CCSM `Network`
      type: string
      example: ''
      readOnly: false
  title: Network
  type: object
```

[O35]

Status of a CCSM network.

```
NetworkStatus:
  example:
    block: 10051293
    chain_id: '3'
    last_block_at: 1618585516000
    meta:
      average_blocktime: 0x
      blocktimes:
        - '55'
        - '24'
        - '8'
        - '1'
        - '3'
        - '20'
        - '7'
      last_block_hash: '698465416514684984piouhygiuytfruytdcghbkjhbvjyhdc'
      last_block_header: {}
  properties: {}
  title: Network Status
  type: object
```

9.6 Protocol Messages

[R32]

Baseline protocol message internal to a BPI.

```
ProtocolMessage:
  description: 'Baseline protocol message internal to a BPI; _normative_'
  required:
    - identifier
    - payload
    - sender
    - recipients
    - signature
  properties:
    workflowinstance_id:
      description: 'ID of associated workflow instance; _UUID as specified by RFC4122_'
      format: uuid
      readOnly: false
      type: string
    identifier:
      description: unique identifier of a message
      format: uuid
      readOnly: true
      type: string
    messagethread_id:
      description: unique identifier of a message thread in order to trace causally connected messages
      format: uuid
      readOnly: true
```

```
        type: string
      opcode:
        description: Type of BPI operation to be executed by the BPI
        type: string
      payload:
        $ref: '#/components/schemas/ProtocolMessagePayload'
      recipients:
        type: array
        items:
          type: string
      sender:
        type: string
      signature:
        type: string
      type:
        type: string
    title: 'Protocol Message'
    type: object
```

[R33]

Baseline protocol message payload.

```
  ProtocolMessagePayload:
    title: Protocol Message Payload
    description: Baseline protocol message payload; _normative_
    properties:
      proof:
        type: string
        description: Proof to verify
      type:
        type: string
        description: 'Type of payload'
      witness:
        $ref: '#/components/schemas/Witness'
      Prover:
        description: |-
          A state synchronization and consistency mechanism for completing on- and off-chain state transitions.

          **Lifecycle**
          __________

          |Status|Description   |
          |------|-----------------------|
          |init|default status value for all newly-initialized provers|
          |compiling|prover is currently being compiled|
          |compiled|prover has been successfully compiled|
          |pending_setup|trusted setup will begin asynchronously|
          |running_setup|trusted setup is running asynchronously|
          |deploying_artifacts|on-chain shield/verifier contract artifacts are being deployed|
          |provisioned|prover is ready for use|; ; _normative_
    example:
      created_at: '2021-02-27T14:26:06.864Z'
      curve: 'BN254'
      description: 'Example circuit-type prover'
      id: '584877fd-6a18-4950-9917-1dd0da26e249'
      identifier: 'Cubic'
      name: 'Purchase Order'
      provider: 'gnark'
      proving_scheme: 'cubic'
      store_id: 'f39a4b7a-2e4f-496b-a4f1-8a9635da10e7'
      type: 'circuit'
    properties:
      created_at:
        description: 'Date and time of `Prover` creation'
        example: '2021-02-27T14:26:06.864Z'
        format: date-time
        readOnly: true
      description:
        description: 'User-defined description of the `Prover`'
        example: 'Prover example description'
        readOnly: false
        type: string
      id:
        description: 'ID for a specific `Prover`; _UUID as specified by RFC4122_'
        example: 'ce0319c8-ddc8-4d46-a16c-c506f5dae9a0'
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: true
        type: string
      identifier:
        description: 'Prover identifier'
        example: 'Cubic'
        type: string
      name:
        description: 'User-defined name of `Prover`'
        example: 'Prover example name'
        readOnly: false
        type: string
      status:
        description: 'Status of `Prover`'
        enum:
          - init
```

```
              compiling
              compiled
              pending_setup
              running_setup
              deploying_artifacts
              provisioned
          type: string
        store_id:
          description: 'ID of the persistent store instance; _UUID as specified by RFC4122_'
          readOnly: false
          type: string
        type:
          description: 'Type of `Prover`'
          example: circuit
          readOnly: false
          type: string
        vault_id:
          description: ID of host `Vault`; _UUID as specified by RFC4122_
          example: 0ca9331a-d3fb-401c-b00b-5ada557c1056
          format: uuid
          readOnly: false
          type: string
      required:
        - name
        - identifier
        - provider
        - proving_scheme
        - curve
        - type
      title: Prover
      type: object
```

9.7 Recovery Policies

[R34]

Recovery policy attached to a BPI SubjectAccount typically connected to a role; may be independent.

```
    RecoveryPolicy:
      description: 'Recovery policy attached to a BPI `SubjectAccount` typically connected to a role; may be independent; _normative_'
      example:
        type: 'recoveryKeyPolicy'
        reference: ''
      properties:
        type:
          description: type of recovery policy
          example: 'RecoveryKeyPolicy'
          type: string
        reference:
          description: resolvable reference link to the recovery policy object, avoids duplicating policy objects
          example: 'https://example.com/policies/recovery-key-policy.json'
          type: string
          format: URI
      readOnly: false
      required:
        - type
        - reference
      title: Recovery Policy
      type: object
```

9.7 Cryptographic Secret

[O36]

Cryptographic secret.

```
    Secret:
      description: 'Cryptographic secret; _non-normative_'
      example:
        created_at: '2021-04-17T20:35:34.062+0000'
        description: 'This secret is being stored for demonstration purposes'
        id: 'e9138dd7-b429-410a-bfa8-2a506d9e4297'
        name: 'Example secret'
        type: 'ed25519'
        value: '0x1'
        vault_id: 22e48752-7bff-4daa-89b3-91053bd509f2
      properties:
        created_at:
          description: Date-time of `Secret` creation
          format: date-time
          readOnly: true
          type: string
        description:
          description: User-defined descriptor of specific `Secret`
          readOnly: false
          type: string
        id:
          description: ID of a specific `Secret`; _UUID as specified by RFC4122_
          format: uuid
          maxLength: 36
          minLength: 1
          readOnly: true
          type: string
        name:
          description: User-defined name of `Secret`
          readOnly: false
          type: string
        type:
          description: Type of `Secret`; can be an arbitrary string as of v1.0.0
          readOnly: false
          type: string
        value:
          readOnly: false
          type: string
        vault_id:
          description: ID of `Vault` in which a specific `Secret` is stored; _UUID as specified by RFC4122_
          readOnly: false
          type: string
      required:
        - description
        - name
        - type
        - vault_id
        - value
      title: Secret
      type: object
```

9.8 BPI State

[R35]

Representation of a valid state as claimed by a workgroup participant.

```
  StateClaim:
    description: Representation of a valid state as claimed by a workgroup participant; _normative_
    example:
      cardinality: '7'
      path: '0111111'
      root: '1fbd47cd5872d379b9846973c2f4d9ec9f9a5da929f0bf212c57de477848c82525c8487847de572c21bff029a95d9a9fecd9f4c2736984b979d37258cd47bd1f'
      values:
        - 1d9224378d77925d612c9f926eb9fb92850e6551def8328011b6a972323298d5
        - 64633297396b6eb9d95ada5d79090c49499d58966aeb44df28b683d7b9632d24
        - fc94366912a846ca63ecfcffb586a856072eb3238ba90665b6a55b58b6f9aa47
        - c525c32e7cea02e3f36d7fa6cc2977da7eba1fd85b16261198c879271b2554a9
        - 616901d6bd3c54537b51672f7b0022b7e6136ba4225a350ebfc3937b6f728f35
        - 1f64767072db8634f0d9561455682172f3ae9919dc085e45c8a7a27bdf7612a7
        - c2263710add40de071af658495f4ca12c2703e1bd086012eb5a58e35825fcdf8
    properties:
      cardinality:
        description: Cardinality of the claim
        format: uint256
        type: string
      path:
        description: Path containing the siblings needed to reconstruct the root at the specified cardinality
        type: string
      root:
        description: Merkle root of the claimed state
        type: string
      values:
        description: List of hashed proofs corresponding to the values at cardinality and each sibling path
        items:
          type: string
        type: array
    title: State Claim
    type: object
```

[R36]

Proposed (commercial) state transition of a BPI State Object via a BPI Transaction.

```
  StateProof:
    description: 'Proposed (commercial) state transition of a BPI State Object via a BPI Transaction; _normative_'
    example:
      proof: |-
        cd50a3d7436610d3ba8e60d7e1be147a
        65f915326c4debed565fa9f58eac399b
        36871252fec8852f579ab24bb7e8f35e
        6e9b14d69ba7c2fa893c4c890d2d68ff
        fad6ac9a54e291b24b9d86d590d096df
      store: true
      witness: {}
    properties:
      proof:
        description: 'The hex-encoded proof to verify'
        example: |-
          cd50a3d7436610d3ba8e60d7e1be147a
          65f915326c4debed565fa9f58eac399b
          36871252fec8852f579ab24bb7e8f35e
          6e9b14d69ba7c2fa893c4c890d2d68ff
          fad6ac9a54e291b24b9d86d590d096df
        readOnly: false
        type: string
      store:
        description: 'When true, the proof hash is written to the configured prover store'
        example: true
        readOnly: false
        type: boolean
      witness:
        $ref: '#/components/schemas/Witness'
    required:
      - proof
      - witness
    title: 'State Proof'
    type: object
```

9.9 BPI Subject & Accounts

[R37]

Abstract identifier representing, e.g., an organization or user; a BPI Subject interacts with its BPIAccount instances by way of a BPI SubjectAccount.

```
  Subject:
    example:
      created_at: '2021-04-16T15:05:23.130Z'
      description: 'Organization for testing'
      id: 'did:elem:a7e165dd-2d91-4f1e-b025-6ef2c5514603'
      metadata: {}
      name: 'ACME Inc.'
      type: 'Organization'
    description: |-
      'Abstract identifier representing, e.g., an organization or user; a BPI `Subject` interacts with its `BPIAccount` instances by way of a BPI `SubjectAccount`; _normative_'
    type: object
    properties:
      '@context':
        description: Context reference file for BPI `Subject` schema validation
        example: []
        type: object
        readOnly: false
      created_at:
        description: Date and time of BPI `Subject` creation
        example: '2021-02-27T14:26:06.864Z'
        format: date-time
        readOnly: true
        type: string
      description:
        description: 'User-defined description of BPI `Subject`'
        example: Organization for testing
        readOnly: false
        type: string
      id:
        description: 'ID of a specific BPI `Subject`'
        example: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
        readOnly: true
        type: string
      metadata:
        nullable: true
        type: object
        properties:
          address:
            description: '`Wallet` address of associated BPI `Subject`'
            example: '0x2f90A6D021db21e1B2A077c5a37B3C7E75D15b7e'
            type: string
          api_endpoint:
            description: 'URL of the baseline API, which may not be exposed to the WAN'
            example: 'https://api_endpoint/api/v1/endpoint'
            type: string
          messaging_endpoint:
            description: 'Publicly-accessible TCP endpoint where the `Subject` receives inbound protocol messages'
            example: 'https://api_endpoint/api/v1/messaging'
            type: string
          interface_metadata:
            description: 'Arbitrary data as needed for additional external interfaces'
            type: object
      name:
        description: 'Name of `Subject`'
        example: 'ACME Inc.'
        readOnly: false
        type: string
      subject_id:
        description: '`Subject` ID of `Subject` creator'
        example:
        readOnly: false
        type: string
      type:
        description: Type of `Subject`
        example: 'Organization'
        type: string
        readOnly: false
    title: Subject
```

[R38]

A BPI SubjectAccount is the context of a Subject.

```
  SubjectAccount:
    description: |-
      A BPI `SubjectAccount` is the context of a `Subject`. _normative_
    title: Subject Account
    type: object
    required:
        - credentials
        - id
        - security_policies
        - recovery_policy
        - subject_id
        - role
        - bpi_account_ids
    properties:
      '@context':
        description: Context reference file for BPI `SubjectAccount` schema validation
        example: []
        type: object
        readOnly: false
      id:
        description: Identifier of the specific BPI `SubjectAccount`
        example: 'did:elem:4A252b7e30f6cD6a85d20476Bb291Ef7f3a05293'
        type: string
        readOnly: true
      bpi_account_ids:
        type: array
        description: IDs of `BPIAccount` instances associated with the specified BPI `SubjectAccount`
        items:
          type: string
        example:
          - 6bb23e2b-07ed-4562-8afb-73955f8f17c5
          - 7cb11a1a-01ca-3421-6fbd-42651c1a32a1
      created_at:
        description: Date and time of BPI `SubjectAccount` creation
        example: '2021-02-27T14:26:06.864Z'
        format: date-time
        readOnly: true
        type: string
      credentials:
        description: A list of cryptographically verifiable credentials establishing legal identity
        example:
          credential_type: 'JWS'
          credentials:
            id: 'https://example.com/issuer/123#ovsDKYBjFemIy8DVhc-w2LSi8CvXMw2AYDzHj04yxkc'
            type: JsonWebKey2020
            controller: 'https://example.com/issuer/123'
            publicKeyJwk:
              kty: OKP
              crv: Ed25519
              x: CV-aGlld3nVdgnhoZK0D36Wk-9aIMlZjZOK2XhPMnkQ
        type: array
        items:
          type: object
          required:
            - type
            - credential
          properties:
            type:
              description: Type of credential associated with account
              example: 'JWS'
              type: string
            credential:
              description: Credential associated with the BPI `SubjectAccount`. Should be a W3C Verifiable Credential
              type: object
        readOnly: false
      metadata:
        description: Metadata useful for BPI operations and UI implementations such as names or API and messaging endpoints
        example: {}
        type: object
        readOnly: false
      type:
        description: Type of BPI `SubjectAccount`
        example: 'ProvideSubjectAccount'
        type: string
        readOnly: false
      recovery_policy:
          $ref: "#/components/schemas/RecoveryPolicy"
      role:
        description: Role of the `Subject`; implementation specific
        example:
          name: Organization
          reference: 'https://example.com/roles/organization.json'
        type: object
        required:
          - name
          - reference
        properties:
            name:
              description: Name of role
              example: 'Organization'
              type: string
            reference:
              description: Resolvable reference link to the role object, avoids duplicating role objects
              example: 'https://example.com/roles/organization.json'
              type: string
              format: URI
        readOnly: false
      subject_id:
        description: Must be either a DID or another resolvable unique identifier of the `Subject`
        example: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
        maxLength: 1024
        type: string
        format: URI
        readOnly: false
      security_policies:
        description: Security policies attached to a `Subject` typically connected to a role but may be independent
        example:
          type: 'AuthenticationPolicy'
          reference: ''
        type: array
        items:
            type: object
            required:
              - type
              - reference
            properties:
              type:
                description: Type of security policy
                example: 'AuthenticationPolicy'
                type: string
              reference:
                description: Resolvable reference link to the security policy object, avoids duplicating policy objects
                example: 'https://example.com/policies/authentication-policy.json'
                type: string
                format: URI
        readOnly: false
```

9.10 BPI Transactions

[R39]

Defines a BPI transaction between two or more subjects to be submitted to a CCSMNetwork.

```
Transaction:
  description:  BPI transaction between two or more subjects to be submitted to a CCSM `Network`; _normative_
  example:
    context: {workstep_id: '88b50dd4-73ea-46d1-a434-b6a2a908bc51'}
    created_at: '2021-04-21T02:12:11.158Z'
    id: '99a61ee5-73ea-46d1-a434-b6a2a908bc51'
    data:
      block: 10079406
      block_timestamp: '2021-04-21T02:12:05.000Z'
      broadcast_at: '2021-04-21T02:12:05.000Z'
      finalized_at: '2021-04-21T02:12:14.277Z'
      hash: '84f8525882f3bb24cd2d3cbdf0264115e03932b927f928f9e70b72bf90dd54f1'
      hd_derivation_path: m/44'/60'/0'/0/0
      network_id: '66d44f30-9092-4182-a3c4-bc02736d6ae5'
      status: 'success'
      to: '7c8fe6f1-38c3-4da1-b4b7-7591c6d0ca7c'
      value: 0
      wallet_id: '0fda2f96-d957-4f65-9e92-fa5092c51751'
  properties:
    context:
      description: 'Establishes the context of the transaction either within a BPI `Workstep` or `Workflow` instance'
      type: object
      items:
        properties:
          workflow_id:
            type: string
          workstep_id:
            type: string
        anyOf:
          - required:
              - workflow_id
              - workstep_id
    created_at:
      description: 'Date and time for the creation of a specific `Transaction`'
      example: '2021-04-21T02:12:11.158Z'
      format: date-time
      readOnly: true
      type: string
    data:
      description: 'Transaction data'
      nullable: true
      type: object
    description:
      description: 'A user-defined summary of the transaction'
      nullable: true
      type: object
    id:
      description: 'ID for a specific `Transaction`; _UUID as specified by RFC4122_'
      example: 'bb0e8038-8e72-46db-a08d-204f804a24c1'
      format: uuid
      maxLength: 36
      minLength: 1
      readOnly: true
      type: string
  title: 'Transaction'
```

9.11 Vault Management

[O37]

Unseals a specified Vault.

```
UnsealVault:
  description: 'Unseals a specified `Vault`; _non-normative_'
  example:
    key: 'quantum mirror summer general keep west promote satisfy limit butter mother oyster lazy whip earn spot ticket protect mean play script peace parent intact'
  properties:
    key:
      description: 'Private key required for unsealing a specific `Vault`'
      type: string
  required:
    - key
  title: 'Unseal Vault'
  type: object
```

[O38]

Provides key management with a focus on providing advanced privacy and messaging capabilities (i.e., zero-knowledge proofs, SNARK-friendly hash functions, double-ratchet algorithm, etc.).

```
  Vault:
    description: 'Provides key management with a focus on providing advanced privacy and messaging capabilities (i.e., zero-knowledge proofs, SNARK-friendly hash functions, double-ratchet algorithm, etc.); _non-normative_'
    example:
      created_at: '2021-08-23T05:40:11.047741Z'
      description: 'Example Vault for testing and documentation purpose'
      id: '02093d35-61fd-42c3-b70e-44f5312f5187'
      name: Example vault
    properties:
      created_at:
        description: 'Date and time of `Vault` creation'
        example: '2021-04-21T02:12:11.158Z'
        format: date-time
        readOnly: true
        type: string
      description:
        description: 'User-defined descriptor for a specific `Vault` instance'
        example: 'Sample vault description'
        readOnly: false
        type: string
      id:
        description: 'ID for a specific `Vault` instance; _UUID as specified by RFC4122_'
        example: 'f1f29322-b8fe-4adf-95ad-af1585e823c5'
        format: uuid
        readOnly: true
        type: string
      name:
        description: 'User-defined name for a specific `Vault` instance'
        example: 'Example vault name'
        readOnly: false
        type: string
    title: Vault
    type: object
```

9.12 Wallet Management

[O39]

Defines a wallet structure.

```
  Wallet:
    description: 'Defines a wallet structure; _non-normative_'
    example:
      created_at: '2021-03-30T02:50:17.155Z'
      id: 'bbc1195e-d131-4b2f-b1a8-1b6f34b55d3e'
      key_id: 'db9e6e21-23b0-497d-a02c-8a8813f8bf2d'
      public_key: 'xpub661MyMwAqRbcGdYXwwnwcnrH51AxyF3kkTeEt6iZfNDXsg4MLPgV7bEZ2v4uCx9djAboy6vSv3VbHPc6hf4Do8wb7FpJCuG7aMBwH2QgLJu'
      purpose: 44
      subject_id: 'did:elem:7c8fe6f1-38c3-4da1-b4b7-7591c6d0ca7c'
      vault_id: '190822c9-62f5-4caf-a419-df735a793b2f'
    properties:
      created_at:
        description: Date and time of `Wallet` creation
        example: '2021-01-27T03:19:33.349Z'
        readOnly: true
        type: string
      id:
        description: 'ID for a specific `Wallet`; _UUID as specified by RFC4122_'
        example: '4ee831ae-327f-4a79-bd74-7e9f56ce40fa'
        format: uuid
        readOnly: true
        type: string
      key_id:
        description: 'ID of a specific private key used to sign a specific `Wallet`; _UUID as specified by RFC4122_'
        example: '5c928a43-9897-417a-9851-768e19b3cecd'
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: true
        type: string
      network_id:
        description: 'ID for a specific `Network` for which a specific `Wallet` is intended; _UUID as specified by RFC4122_'
        example: '84a139c0-2e20-476c-91d0-618f7332189e'
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: false
        type: string
      public_key:
        description: 'Public signing key of a specific `Wallet`'
        example: 'xpub661MyMwAqRbcGdYXwwnwcnrH51AxyF3kkTeEt6iZfNDXsg4MLPgV7bEZ2v4uCx9djAboy6vSv3VbHPc6hf4Do8wb7FpJCuG7aMBwH2QgLJu'
        readOnly: true
        type: string
      purpose:
        description: 'Purpose is a constant set to 44 (or 0x8000002C) following the BIP43 recommendation. It indicates that the subtree of this node is used according to this specification. Hardened derivation is used at this level'
        example: 44
        format: string
        readOnly: false
        type: integer
      subject_id:
        description: 'ID for a specific `Subject` that created a specific `Wallet`'
        example: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
        maxLength: 1024
        minLength: 1
        readOnly: true
        type: string
      vault_id:
        description: 'ID for a specific `Vault` in which a specific `Wallet` is stored; _UUID as specified by RFC4122_'
        example: 'aa1b60d4-4f77-438d-8fca-cb514596e6c3'
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: true
        type: string
    title: 'Wallet'
    type: object
```

9.13 Cryptographic Witness

[R40]

Public and private inputs used to generate the Proof. Payload can be delivered as a JSON object or chunked and streamed.

```
  Witness:
    description: 'Public and private inputs used to generate the `Proof`. Payload can be delivered as a JSON object or chunked and streamed. Example is dependent on `Prover` parameters. _normative_'
    example: {}
    properties: {}
    title: Witness
    type: object
  WorkflowInstance:
    description: 'A BPI `WorkflowInstance` consisting of a set of stepwise BPI workstep instances; _normative_ '
    example:
      id: 'ed509f33-cbe4-4e3d-86ff-e508e6496f85'
      name: Procure to Pay
      type: procure_to_pay
      workstep_ids:
        - 'ed509f33-cbe4-4e3d-86ff-e508e6496f85'
        - 'eg922g62-ghr6-1c2c-09cc-f601e5715g28'
    properties:
      id:
        description: 'ID of a specific baseline `WorkflowInstance`; _UUID as specified by RFC4122_'
        example: ed509f33-cbe4-4e3d-86ff-e508e6496f85
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: True
        type: string
      name:
        description: 'User-defined name for a specific `WorkflowInstance`'
        example: 'Test workflow'
        type: string
      type:
        description: 'Type of `WorkflowInstance'
        example: Purchase order
        readOnly: false
        type: string
      workstep_ids:
        description: List of `WorkstepInstance` ids included in the `WorkflowInstance`
        example:
          - 'ed509f33-cbe4-4e3d-86ff-e508e6496f85'
        readOnly: true
        type: array
        items:
          type: string
    required:
      - type
    title: Workflow Instance
    type: object
```

9.14 Workgroup Management

[R41]

A BPI Workgroup represents a logical collection of resources provisioned within a BPI.

```
  Workgroup:
    description: |-
      'A BPI `Workgroup` represents a logical collection of resources provisioned within a BPI. _normative_'
    example:
      subject_id: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
      config:
        baselined: true
        webhook_secret: '1dfd34519a06420d8a6ed18afdffe932'
      created_at: '2021-05-21T12:36:52.977Z'
      description: 'Workgroup intended for demonstration purposes'
```

```
        id: 'did:elem:bdf94fc0-63ad-4ee9-ac75-7eb63365d0f6'
        name: 'Example Workgroup'
        network_id: '07102258-5e49-480e-86af-6d0c3260827d'
        type: 'baseline'
        security_policies: []
        admins:
          - 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
      properties:
        config:
          description: BPI-specific workgroup data
          example: {}
          type: object
        created_at:
          description: 'Date and time of `Workgroup` creation'
          example: '2021-05-21T12:36:52.977Z'
          format: date-time
          minLength: 1
          readOnly: true
          type: string
        description:
          description: 'User-defined `Workgroup` descriptor'
          example: 'Example workgroup description'
          maxLength: 255
          type: string
        id:
          description: 'ID for a specific `Workgroup`'
          example: 'did:elem:bdf94fc0-63ad-4ee9-ac75-7eb63365d0f6'
          minLength: 1
          readOnly: true
          type: string
        name:
          description: 'User-defined name of the `Workgroup`'
          maxLength: 255
          minLength: 1
          type: string
        network_id:
          description: 'ID for a specific `Network` host of a specific `Workgroup`; _UUID as specified by RFC4122_'
          example: '07102258-5e49-480e-86af-6d0c3260827d'
          format: uuid
          maxLength: 36
          minLength: 1
          type: string
        type:
          default: baseline
          description: 'Designates `Workgroup` type. In a Baseline `Workgroup`, will always be baseline'
          example: baseline
          minLength: 1
          type: string
        participants:
          description: '`Subject` identifiers that are `Workgroup` participants'
          minLength: 1
          readOnly: false
          type: array
          items:
            $ref: "#/components/schemas/Subject"
          example:
            - 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
        security_policies:
          description: 'Security policies attached to a Workgroup'
          example:
            type: AuthenticationPolicy
            reference: https://example.com/policies/authentication-policy.json
          minLength: 1
          type: array
          items:
              type: object
              required:
                - type
                - reference
              properties:
                type:
                  description: type of security policy
                  example: 'AuthenticationPolicy'
                  type: string
                reference:
                  description: resolvable reference link to the security policy object, avoids duplicating policy objects
                  example: 'https://example.com/policies/authentication-policy.json'
                  type: string
                  format: URI
          readOnly: false
        subject_id:
          description: ID of `Subject` that created `Workgroup`
          example: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
          type: string
        admins:
          description: 'IDs of `Subject` instances with administrative control of `Workgroup`'
          example:
          type: array
          minLength: 1
          items:
            type: string
      required:
        - name
        - type
        - subject_id
        - security_policies
        - admins
      title: Workgroup
      type: object
```

9.16 Workstep Management

[R42]

One stepwise state transition in a WorkflowInstance represented by a Prover instance. A Workstep may or may not be required to reach on-chain finality prior to the execution of the next Workstep in a Workflow.

```
  WorkstepInstance:
    description: 'One stepwise state transition in a `WorkflowInstance` represented by a `Prover` instance. A `Workstep` may or may not be required to reach on-chain finality prior to the execution of the next `Workstep` in a `Workflow`. _normative_'
    example:
      created_at: '2021-04-21T02:12:05.000Z'
      id: '2fffec13-590e-41eb-9a84-69ed881e0036'
      prover_id: 'dd615561-83e6-4f53-ac0f-aaf1749df139'
      require_finality: true
      workflow_id: 'f9227803-5541-4c13-a554-f6980f03362a'
      type: 'purchase_order'
      participants: []
      subject_id: 'did:elem:93229a14-5e13-4c45-8352-3ad9948b8ae3'
    properties:
      id:
        description: 'ID of a specific `WorkstepInstance`; _UUID as specified by RFC4122_'
        example: '2fffec13-590e-41eb-9a84-69ed881e0036'
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: true
        type: string
      created_at:
        description: 'Date and time of `WorkstepInstance` creation'
        example: '2021-05-21T12:36:52.977Z'
        format: date-time
        minLength: 1
        readOnly: true
        type: string
      participants:
        description: 'List of `Subject` instances that are associated with the `WorkstepInstance`'
        type: array
        items:
          $ref: '#/components/schemas/Subject'
      prover_id:
        description: 'ID of the a specific `Prover` instance that is represented in a specific `WorkstepInstance`; _UUID as specified by RFC4122_'
        example: 'dd615561-83e6-4f53-ac0f-aaf1749df139'
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: false
        type: string
      require_finality:
        default: false
        description: 'When true, the resulting proof from the referenced `WorkstepInstance` will be written to the underlying mainnet.'
        example: true
        type: boolean
      state_claims:
        items:
          $ref: '#/components/schemas/StateClaim'
        type: array
      subject_id:
        description: 'ID of associated `Subject'
        type: string
        readOnly: false
        example:
      type:
        description: 'Type of workstep'
        example: 'purchase_order'
        type: string
        readOnly: false
      workflow_id:
        description: 'ID of a specific `WorkflowInstance` that a specific `WorkstepInstance` is part of; _UUID as specified by RFC4122_'
        example: f9227803-5541-4c13-a554-f6980f03362a
        format: uuid
        maxLength: 36
        minLength: 1
        readOnly: false
        type: string
    required:
      - subject_id
      - participants
      - workflow_id
    title: Workstep
    type: object
```

10 API Security

This section gives an example of how to programmatically integrate API security using the example of the well-known OAuth 2.0 authentication standard in a YAML formulation.

In accordance with the OAuth 2.0 specification, when a Subject is authorized and the requested scope includes offline_access, a refresh token is vended and returned on behalf of the caller. This refresh token is long-lived and can be used to authorize short-lived access tokens using the refresh_tokengrant type on subsequent authorization requests. This pattern is useful for machine-to-machine applications; a secure practice is to store the long-lived refresh token in a Vault instance (i.e., as a secret), read it into application memory during container initialization and then use it to authorize a short-lived access token. If the container remains running long enough for the access token to expire, the refresh token should once again be used to seamlessly authorize a new access token.

```
securitySchemes:
  OAuth2:
    description: |-
      In accordance with the OAuth 2.0 specification, when a `Subject` is authorized and the requested `scope` includes `offline_access`, a refresh token is vended and returned on behalf of the caller. This refresh token is long-lived and can be used to authorize short-lived access tokens using the `refresh_token `grant type on subsequent authorization requests. This pattern is useful for machine-to-machine applications; a secure practice is to store the
      long-lived refresh token in a Vault instance (i.e., as a secret), read it into application memory during container initialization and then use it to authorize a short-lived access token. If the container remains running long enough for the access token to expire, the refresh token should once again be used to seamlessly authorize a new access token. _non-normative_
    flows:
      implicit:
        authorizationUrl: https://ca.example.com/api/v1/oauth/authorize
        scopes:
          email: email and password
          offline_access: access/refresh token authorization model
    type: oauth2
  bearerAuth:
    bearerFormat: JWT
    description: |-
      'The presence of a `bearer` API token is required to authorize most Baseline API calls. A `bearer` API token is an encoded JWT which contains a subject claim (`sub`) which references an authorized `Subject`. The authorized entity uses a signed `bearer` authorization Token xto access one or more resources for which the Token was authorized.  Unless otherwise noted, all API requests must include a header such as `Authorization: bearer <jwt>` _normative_'
    scheme: bearer
    type: http
externalDocs:
description: Privacy Policy
url: https://example.com/privacy-policy
security:
- bearerAuth: []
- OAuth2: []
servers:
- description: Baseline Operator
  url: https://baseline.provide.network
- description: Local Baseline Integration Stack
  url: http://localhost:8080
- description: Local BRI-1 Stack
  url: http://localhost:9999
```

This section is non-normative.

11 Conformance

This section describes the conformance clauses and tests required to achieve an implementation that is provably conformant with the requirements in this document.

9.1 Conformance Targets

This document does not yet define a standardized set of test-fixtures with test inputs for all MUST [R], SHOULD [D], and MAY [O] requirements with conditional MUST [CR] or SHOULD [CD] requirements.

A standardized set of test-fixtures with test inputs for all MUST, SHOULD, and MAY requirements with conditional MUST or SHOULD requirements is intended to be published with the next version of the standard.

9.2 Conformance Levels

This section specifies the conformance levels of this standard. The conformance levels aim to enable implementers several levels of conformance to establish competitive differentiation.

This document defines the conformance levels of a BPI as follows:

- **Level 1:** All MUST requirements are fulfilled by a specific implementation as proven by a test report that proves in an easily understandable manner the implementation's conformance with each requirement based on implementation-specific test-fixtures with implementation-specific test-fixture inputs.
- **Level 2:** All MUST and SHOULD requirements are fulfilled by a specific implementation as proven by a test report that proves in an easily understandable manner the implementation's conformance with each requirement based on implementation-specific test-fixtures with implementation-specific test-fixture inputs.
- **Level 3:** All MUST, SHOULD, and MAY requirements with conditional MUST or SHOULD requirements are fulfilled by a specific implementation as proven by a test report that proves in an easily understandable manner the implementation's conformance with each requirement based on implementation-specific test-fixtures with implementation-specific test-fixture inputs.
- **Level 4:** All MUST requirements are fulfilled by a specific implementation as proven by the test report defined in this document that proves the implementation's conformance with each requirement based on test-fixtures with test-fixture inputs as defined in this document. **This conformance level cannot yet be achieved since there is not yet a defined set of standardized test-fixtures and test-inputs**.
- **Level 5:** All MUST and SHOULD requirements are fulfilled by a specific implementation as proven by the test report defined in this document that proves the implementation's conformance with each requirement based on test-fixtures with test-fixture inputs as defined in this document. **This conformance level cannot yet be achieved since there is not yet a defined set of standardized test-fixtures and test-inputs**.
- **Level 6:** All MUST, SHOULD, and MAY requirements with conditional MUST or SHOULD requirements are fulfilled by a specific implementation as proven by the test report defined in this document that proves the implementation's conformance with each requirement based on test-fixtures with test-fixture inputs as defined in this document. **This conformance level cannot yet be achieved since there is not yet a defined set of standardized test-fixtures and test-inputs**.

[D1]

A claim that a Baseline Protocol API implementation conforms to this specification SHOULD describe a testing procedure carried out for each requirement to which conformance is claimed, that justifies the claim with respect to that requirement.

[D1] testability: Since each of the non-conformance-target requirements in this documents is testable, so must be the totality of the requirements in this document. Therefore, conformance tests for all requirements can exist, and can be described as required in [D1].

[R43]

A claim that a Baseline API implementation conforms to this specification at**Level 2** or higher MUST describe the testing procedure carried out for each requirement a**Level 2** or higher, that justifies the claim to that requirement.

Appendix A - Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged.

Participants:

Andreas Freund

Anais Ofranc, Consianimis

Gage Mondok, Chainlink

Kyle Thomas, Provide

Daven Jones, Provide

Mehran Shakeri, SAP

Alessandro Gasch, SAP

John Wolpert, ConsenSys

Sam Stokes, ConsenSys

Nick Kritikos, ConsenSys

Yoav Bittan, ConsenSys

Keith Salzman, ConsenSys

Chaals Nevile, EEA

Appendix B. Revision History

Revisions made since the initial stage of this numbered Version of this document have been tracked on Github.

Appendix C. Non-Normative References

[OAuth-2.0]

Aaron Parecki, (2020), “OAuth 2.0 Simplified”, ISBN-13: 978-1387751518.

C.1 Internationalization and Localization Reference

The standard encourages implementers to follow the W3C “Strings on the Web: Language and Direction Metadata” best practices guide for identifying language and base direction for strings used on the Web wherever appropriate.

Appendix D - Notices