TRE-FX

Technical Documentation - Five Safes RO-Crate

- **Title**: TRE-FX Technical Documentation - Five Safes RO-Crate
- **Date**: 2023-12-14
- **Authors**: Stian Soiland-Reyes, Stuart Wheater, Tom Giles, Carole Goble, Philip Quinlan
- **Cite as**: https://doi.org/10.5281/zenodo.10376350
- **Abstract**: This report documents Five Safes RO-Crate, a FAIR representation of software execution requests used by the TRE-FX architecture. A Five Safes RO-Crate represents a unit of computational access to sensitive information which is managed in accordance with a set of principles conforming to the 5 safe framework. The aim is to enable trusted workflow execution in a Trusted Research Environment (TRE), from an authenticated workflow run request, through approval and review processes to a completed workflow execution.

The Five Safes RO-Crate profile

A Five Safes RO-Crate represents a **unit of computational workflow-based access to sensitive information** which is managed in accordance with a set of principles conforming to the Five Safe framework, a well-established model for managing access to confidential or sensitive data. The aim is to enable trusted workflow execution in a Trusted Research Environment (TRE), from an authenticated workflow run request, through approval and review processes to a completed workflow execution. The profile has been developed for the purpose of TRE-FX implementation of workflow execution in a distributed TRE [Giles 2023]. The Five Safes RO-Crate is a specialised profile of RO-Crate, whereby encapsulated elements and metadata provide the necessary context for evaluating the safety and appropriateness of both data access and analysis.

Note:

A crate that is compliant to the Five Safes RO-Crate profile is not inherently *safe* - its role is to streamline the flow of information by standardising the metadata it collects and carries. That metadata is used to support the Five Safes processes of the TREs and their issuing/receiving clients.

A Five Safes RO-Crate operates in a pre-determined and controlled context: (i) the workflow that is to be executed within the TRE to answer a request has already pre-approved by the TRE and will be executed in a secure deployment and (ii) the services to implement the crate phases (figure 1) are secure and adhere to the Five Safes.

RO-Crate is a community-based specification for packaging and describing research outputs, based on FAIR linked data standards. The approach has been adopted by a variety of research domains [Soiland-Reyes 2022] with specialisation in different *profiles* to combine generic and domain-specific metadata. Recently, the Workflow Run Crate profiles have been developed and are being implemented by more than 6 workflow engines including CWL and Galaxy [De Geest 2022; Leo 2023].

The Five Safes model provides a structured approach to managing confidential or sensitive data through five dimensions: Safe Data, Safe Projects, Safe People, Safe Settings, and Safe Outputs. For data controllers operating Trusted Research Environments (TREs), ensuring compliance with data governance and legal frameworks is critical, especially in the context of federated analytics.

The Five Safes RO-Crate aims to provide a mechanism to encapsulate data, workflows, and provenance with extensible metadata in a standardised, compliant package, and hence improve the flow of the metadata, queries and results necessary to streamline TRE operations, enable rigorous compliance, and enhance data integrity and security.

The initial crate with a workflow run request references a pre-approved workflow and project details for manual and automated assessment according to the TRE's agreement policy for the sensitive dataset. The crate goes through multiple phases internal to the TRE, including validation, sign-off, workflow execution and disclosure control. At this later stage the crate is also conforming to the Workflow Run Crate profile for return to the user, and a derived public version (possibly redacted) can be published in data use registers to document the analysis.

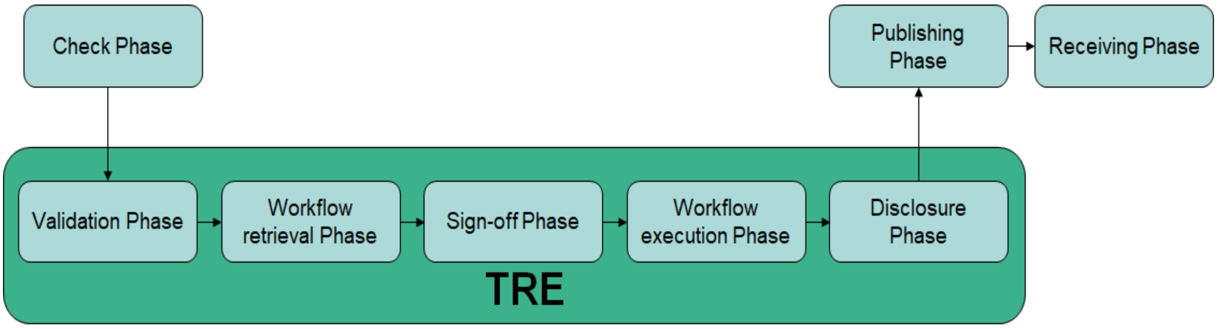

Figure 1: Phases the metadata of a Five Safes RO-Crate is expected to support

The structured format of the Five Safes RO-Crate allows for a standardised audit trail, facilitating regulatory compliance checks and adherence to industry standards like BagIt and the RO-Crate specifications further enhances interoperability. A mapping of the Five Safes dimensions to the Crate metadata is given in Table 1.

Five Safes Principle	Five Safes RO-Crate handling
Safe projects Is this use of the data appropriate?	Embedded reference to the Project Embedded reference to the pre-approved workflow to be run, documented workflows and associated AssessActions ensure each analysis step is traceable and complies with pre-approved standards.
Safe people Can the users be trusted to use it in an appropriate manner?	Embedded reference to the Person requesting as well as who are reviewing. Sign-off phase verifies appropriate use of workflow and workflow parameters for a given TRE. Metadata attributes like 'publisher' and 'agent' define authorised roles for crate execution or modification.
Safe settings Does the access facility limit unauthorised use?	The full request can be audited and traced ensuring that TREs are provided with a mechanism to validate a request is for a project they have authorised, from a person they have authorised for the given project. The crate has been demonstrated to work within TREs whilst allowing the TRE to maintain oversight and control at all times. Embedded reference to the pre-approved workflow or

	included ad-hoc approved workflow definition. BagIt payload manifests and checksums ensure workflow integrity, confirming unaltered data from trusted sources.
Safe data Is there a disclosure risk in the data itself?	The embedded project and person attributes ensure the Crate can be directed by the TRE to the correct environment where the data has been provisioned for the project. The data will have been specifically provisioned and assessed against the Safe Data principle before the Crate is allowed to be run against the data. Workflow RO-Crates describing the workflows are registered in the WorkflowHub for transparency.
Safe outputs Are the statistical results non-disclosive?	Pre-approval to workflow assesses non-disclosiveness in general as only pre-defined outputs are returned. Embedded outputs from the workflow are additionally checked by the Disclosure phase. Documented workflows and associated AssessActions ensure each analysis step is traceable and complies with pre-approved standards. Workflow RO-Crates describing the workflows are registered in the WorkflowHub for transparency. Only pre-approved workflows which are fully open to scrutiny are run, resulting in predictable outputs that are contained within the Crate and open for human inspection before being released from the TRE. Disclosure phase may redact sensitive data from the crate, replaced with anonymized references. BagIt payload manifests and checksums ensure workflow integrity, confirming unaltered data from trusted sources.

Table 1 Mapping Five Safes to Five Safes RO-Crate, including embedded references and metadata elements to provide the context needed for data access and analysis evaluation.

The next section is a snapshot of the Five Safes RO-Crate profile published as version 0.4.

Five Safes RO-Crate profile

- Permalink: https://w3id.org/5s-crate/0.4
- Version: 0.4

- Release notes: https://github.com/trefx/5s-crate/releases/tag/0.4.0
- Published: 2023-09-15

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Note: All references to schema.org types/properties/instances use the prefix `http://schema.org/` (not `https`) to correspond with their official JSON-LD context.

Overview

A Five Safes RO-Crate represents a unit of computational access to sensitive information which is managed in accordance with a set of principles conforming to the 5 safe framework. The aim is to enable trusted workflow execution in a Trusted Research Environment (TRE), from an authenticated workflow run request, through approval and review processes to a completed workflow execution.

Archive serialisation

A compliant Five Safes RO-Crate SHOULD be stored and transferred as an ZIP archive containing a single BagIt directory (*bag*) of an arbitrary name, which payload folder data/ contains the RO-Crate Metadata File `ro-crate-metadata.json` and any required data files (e.g. inputs).

Internally a processing TRE MAY choose to unpack a ZIP file to a local file store, taking necessary security and performance precautions (see Security considerations).

The BagIt payload manifest [RFC8493] MUST be present using sha-512 checksums, and the tag manifest SHOULD be included as sha-512 [FIPS 180-4]. Payload and tag manifests using other checksums MAY be included, taking care to exclude `tagmanifest-*` files from their checksums.

Example:

```
query-12389/
  |   bagit.txt                 # MUST indicate BagIt 1.0 or later
  |   bag-info.txt              # As per BagIt specification
  |   manifest-sha512.txt       # As per BagIt specification
  |   tagmanifest-sha512.txt    # As per BagIt specification
  |   fetch.txt                 # Optional, per BagIt Specification
  |   data/                     # Payload: RO-Crate root directory
      |   ro-crate-metadata.json          # RO-Crate Metadata File MUST be present
      |   [payload files and directories]  # 1 or more SHOULD be present
```

BagIt expectations

The RO-Crate BagIt expectations for *Adding RO-Crate to Bagit* MUST be followed. The `bag-info.txt` file MUST include a generated `External-Identifier:` field, which SHOULD be a UUID URN [rfc4122], e.g.:

```
External-Identifier: urn:uuid:9796155a-fe44-4614-89b8-71945f718ffb
```

The identifier of the External-Identifier represents this crate as a request and subsequent response, and SHOULD be freshly generated for each request. It is RECOMMENDED to *not* modify this identifier as the Five Safes RO-Crate progresses through the distributed TRE processing, unless it is recognized as an previous execution.

Note that as the ro-crate-metadata.json establishes payload directory `data/` as the RO-Crate Root it can only reference files and directories there within, the RO-Crate MUST NOT reference tag files like `../fetch.txt` or other relative paths outside the Bag (see Security considerations).

Zip expectations

The ZIP archive MUST only contain a single top-level entry for the bag directory, identified by the `bagit.txt` marker. For interoperability in terms of ZIP features, implementations SHOULD follow guidance for an OSF ZIP Container (ignoring *OCF Abstract Container* requirements).

Metadata file expectations

The RO-Crate Metadata File MUST conform to RO-Crate 1.2 (or later minor version).

For the purpose of this specification, this Metadata file skeleton is assumed, with all subsequent JSON elements shown added to the `@graph` array:

```
{ "@context": "https://w3id.org/ro/crate/1.2-DRAFT/context",
  "@graph": [

  ]
}
```

The compliant RO-Crate version MUST be declared in the metadata file descriptor:

```
{
  "@type": "CreativeWork",
  "@id": "ro-crate-metadata.json",
  "about": {"@id": "./"},
  "conformsTo": {"@id": "https://w3id.org/ro/crate/1.2-DRAFT"}
}
```

The root data entity of a Five Safes RO-Crate MUST have the `@id` equal `"./"` (as it is stored within the BagIt ZIP archive). It MAY have an additional identifier.

Profile conformance

Crates conforming to this profile specification SHOULD indicate this on the Root Data Entity dataset, using its own `conformsTo`:

```
{
  "@id": "./",
  "@type": "Dataset",
  "conformsTo": {"@id": "https://w3id.org/5s-crate/0.4"},
  "hasPart": [
  ],
  "mainEntity": {"@id": "https://workflowhub.eu/workflows/289?version=1"},
  "mentions": {"@id": "#query-37252371-c937-43bd-a0a7-3680b48c0538"},
  "sourceOrganization":
    {"@id": "#project-be6ffb55-4f5a-4c14-b60e-47e0951090c70"}
},
{
  "@id": "https://w3id.org/5s-crate/0.4",
  "@type": "Profile",
  "name": "Five Safes RO-Crate profile"
}
```

Note that `licence` and `datePublished` is not required in a submitted crate, but SHOULD be included for a published crate (see Publishing Phase).

Referencing a Workflow Crate

The metadata file MUST reference a Workflow RO-Crate `Dataset` as its `mainEntity`, indirectly indicating the workflow to execute.

The identifier SHOULD be a permalink or versioned URL (e.g. https://workflowhub.eu/workflows/289?version=1) or MAY be a nested directory within the BagIt payload directory (e.g. `"workflow289.1/"`).

Note: unlike in the Workflow Run profile, the programming language of the workflow and its other metadata are not expressed in this RO-Crate, but within the referenced Workflow RO-Crate. The `programmingLanguage` inside the Workflow RO-Crate SHOULD be either `https://w3id.org/workflowhub/workflow-ro-crate#cwl` or `https://w3id.org/workflowhub/workflow-ro-crate#nextflow`.

See security considerations for workflow referencing and execution.

Finding the RO-Crate archive

If the `identifier` is a URI, an URL to the downloadable Workflow RO-Crate ZIP archive SHOULD be included with distribution, otherwise clients SHOULD [use Signposting](https://signposting.org/) to find the link to the RO-Crate by looking for the link with `rel="item" type="application/zip" profile="https://w3id.org/ro/crate"` – for instance:

```
curl -I "https://workflowhub.eu/workflows/289?version=1"
```

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Link: <https://workflowhub.eu/workflows/289/ro_crate?version=1> ;
      rel="item" ;
      type="application/zip" ;
      profile="https://w3id.org/ro/crate"
```

Example:

```
{
  "@id": "https://workflowhub.eu/workflows/289?version=1",
  "@type": "Dataset",
  "name": "CWL Protein MD Setup tutorial with mutations",
  "conformsTo": {"@id": "https://w3id.org/workflowhub/workflow-ro-crate/1.0"},
  "distribution": {
    "@id": "https://workflowhub.eu/workflows/289/ro_crate?version=1"
  }
},
{
  "@id": "https://workflowhub.eu/workflows/289/ro_crate?version=1",
  "@type": "DataDownload",
  "conformsTo": {"@id": "https://w3id.org/ro/crate"},
  "encodingFormat": "application/zip"
}
```

In the above example, the `mainEntity` points to a `Dataset` that conforms to the Workflow RO-Crate Profile and references the ZIP download URI using `distribution`, therefore the client can download the workflow directly without needing to follow Signposting headers.

Requested Workflow Run

The metadata file MUST include a [`CreateAction`](http://schema.org/CreateAction), which MUST be referenced from `mentions` of the root entity. The `identifier` SHOULD be based on a UUID (different from the BagIt `External-Identifier`).

The `CreateAction` MUST reference the Workflow Crate using `instrument`.

Example:

```
{
  "@id": "#query-37252371-c937-43bd-a0a7-3680b48c0538",
  "@type": "CreateAction",
  "actionStatus": "http://schema.org/PotentialActionStatus",
  "agent": {"@id": "https://orcid.org/0000-0001-9842-9718"},
  "instrument": {"@id": "https://workflowhub.eu/workflows/289?version=1"},
  "name": "Execute query 12389 on workflow ",
  "object": [
    {"@id": "input1.txt"}
  ]
}
```

The CreateAction’s `actionStatus` will change during execution.

Requesting Agent

The individual person who is requesting the run MUST be indicated as an agent from the `CreateAction`, which SHOULD have an `affiliation` to the organisation they are representing for access control purposes.

```
{
  "@id": "https://orcid.org/0000-0001-9842-9718",
  "@type": "Person",
  "name": "Stian Soiland-Reyes",
  "affiliation": { "@id": "https://ror.org/027m9bs27"},
  "memberOf": [
    {"@id": "#project-be6ffb55-4f5a-4c14-b60e-47e0951090c70"}
  ]
},
{
  "@id": "https://ror.org/027m9bs27",
  "@type": "Organization",
  "name": "The University of Manchester"
}
```

Note: The organisation under `affiliation` is typically the employing organisation, e.g. a university or hospital. Virtual organisations such as research projects MAY additionally be listed using `memberOf` (see also Responsible Project below).

Responsible Project

The project that the request is sent on behalf of, typically related to permission to use a TRE, MUST be indicated from the root dataset using `sourceOrganization` to a Project. The responsible project SHOULD be referenced from the requesting agent's `memberOf`.

Note: The responsible project is not necessarily a `ResearchProject` corresponding to a funded grant, but may be more specific studies within a funded project. Various TREs may have different granularity and identifiers for the responsible projects. A project `Grant` MAY be referenced using `funding` from the responsible project.

It is RECOMMENDED to include TRE-specific ids under `identifier` (which MAY be an array). If the identifier is not globally unique (e.g. an integer rather than an UUID or URI), it is RECOMMENDED to add a repository-specific identifier and provide the local identifier as value of a `PropertyValue` entity. Multiple repository-specific identifiers MAY be included for different TREs from a single `Project` entity.

The project MAY indicate the `member` organisations, in which case one of them SHOULD match the `affiliation` of the Requesting Agent with a `memberOf` to this project.

Example

```
{
  "@id": "#project-be6ffb55-4f5a-4c14-b60e-47e0951090c70",
  "@type": "Project",
  "name": "Investigation of cancer (TRE72 project 81)",
  "identifier": [
     {"@id": "_:localid:tre72:project81"}
  ],
  "funding": {
      "@id": "https://gtr.ukri.org/projects?ref=10038961"
  },
  "member": [
    {"@id": "https://ror.org/027m9bs27"},
    {"@id": "https://ror.org/01ee9ar58"}
  ]
},
{
  "@id": "_:localid:tre72:project81",
  "@type": "PropertyValue",
  "name": "tre72",
  "value": "project81"
},
{
  "@id": "https://gtr.ukri.org/projects?ref=10038961",
  "@type": "Grant",
  "name": "EOSC4Cancer"
}
```

Inputs

Requested inputs SHOULD be set on the `CreateAction` using the `object` property:

```
{
  "@id": "#query-37252371-c937-43bd-a0a7-3680b48c0538",
  "@type": "CreateAction",
  "object": [
    {"@id": "input1.txt"},
    {"@id": "#enableFastMode"}
  ],
  "…": {}
}
```

Each input MUST have a corresponding *data entity*, which SHOULD have a `exampleOfWork` reference to a corresponding `FormalParameter`:

```
{
  "@id": "input1.txt",
  "@type": "File",
  "name": "input1",
  "exampleOfWork": { "@id": "#sequence"}
},
{
  "@id": "#enableFastMode",
  "@type": "PropertyValue",
  "name": "--fast-mode",
  "value": "True",
  "exampleOfWork": {"@id": "#fast"}
},
{
  "@id": "#sequence",
  "@type": "FormalParameter",
  "name": "input-sequence"
},
{
  "@id": "#fast",
  "@type": "FormalParameter",
  "name": "fast-mode"
}
```

Tip: While the FormalParameter SHOULD match the definitions within the Workflow Crate referenced from `mainEntity`, the only requirement from this profile is that their name is programmatically recognized by the workflow engine for binding input parameters of the particular workflow.

Outputs

If the workflow has successfully executed, that is the `CreateAction` has `actionStatus` set to `http://schema.org/CompletedActionStatus`, the output data entities SHOULD be referenced from the `results` array.

Output entities MUST be described as in the Workflow Run Crate profile, with type SHOULD be either `File`, `Dataset`, `Collection`, `DigitalDocument` or `PropertyValue`.

Implementations MAY include the outputs within the Crate BagIt archive, in which case it is RECOMMENDED to use the folder `outputs/` to avoid conflict with other files in the crate.

```
{
  "@id": "#query-37252371-c937-43bd-a0a7-3680b48c0538",
  "@type": "CreateAction",
  "result": [
    {"@id": "outputs/table.csv"},
    {"@id": "outputs/diagrams/"}
  ],
  "…": {}
},
{
  "@id": "outputs/qa.csv",
  "@type": "File",
  "encodingFormat": "text/csv",
  "name": "Tabular listing of quality assessment"
},
{
  "@id": "outputs/diagrams/",
  "@type": "Dataset",
  "name": "Diagrams of regions of interest"
}
```

Tip: Implementations may need to inspect the `FormalParameter` of the referenced Workflow Crate to propagate a human readable name and `encodingFormat` file format of the inputs and output in this crate.

Sensitive data

Outputs MAY include references to sensitive data that is only accessible from within the TRE or through URIs that require authentication. The requirement for permission SHOULD be indicated by typing the data entity as a [DigitalDocument]() that use `hasDigitalDocumentPermission` to reference the [DigitalDocumentPermission]() entity, typically assigning `http://schema.org/ReadPermission` with grantee to only to the Responsible Project.

```
{
  "@id": "urn:uuid:07b81e0f-7ac4-5428-9940-878b241e2397",
  "@type": "DigitalDocument",
  "encodingFormat": "text/csv",
  "name": "Patient measurement 07b81e0f-7ac4-5428-9940-878b241e2397",
  "hasDigitalDocumentPermission": {"@id": "#permissions-07b81e0f"}
},
{
  "@id": "#permissions-07b81e0f",
  "@type": "DigitalDocumentPermission",
  "permissionType": "http://schema.org/ReadPermission",
  "grantee": { "@id": "#project-be6ffb55-4f5a-4c14-b60e-47e0951090c70"}
}
```

Review process

The Five Safes RO-Crate may face several reviews both before and after workflow execution, automated and manual. To record that such review will or has taken place, a series of additional Action contextual entities SHOULD be related to the root data entity using mentions.

It is RECOMMENDED that the first step after authentication is a syntactic validation step that verifies the RO-Crate validity according to this profile and system expectations. This step SHOULD remove `mentions` references to any end-user-provided AssessAction (as defined in this profile) from the submitted crate, in order to ensure only assessment endorsements by the particular TRE are considered in the subsequent internal processing.

Assessment actions SHOULD indicate an actionStatus to reflect the outcome or pending nature of the assessment. Each assessment SHOULD have the root data entity (typically `{"@id": "./"}`) listed under `object`, and MAY include additional entities that were assessed.

The phase of the review process is indicated using subclasses of `Action` and more accurately with `additionalType` using terms from the Safe Haven Provenance (SHP) ontology.

The name of the action MUST provide a human readable name of the type of check and its outcome, but SHOULD NOT be consulted by software for decision making (rather they should check `actionStatus` and `additionalType`).

Each completed action SHOULD have a timestamp using `endTime` that follows the ISO-8601 syntax of RFC 3339 (including timezone or Z). `startTime` MAY be included for active, failed and completed actions.

The main actor performing the assessment SHOULD be listed under `agent` and refer to either a `Organization` (e.g. an TRE helpdesk), `Person` (manual check) or a `SoftwareApplication` (automated check). A `SoftwareApplication` acting on behalf of a TRE MUST include a reference to the TRE Organization using `provider`. There may be multiple actors appearing as `agent` for different actions, each of which should be listed as contextual entities with at least `name`.

```
{ "@id": "https://tre72.example.com/#crate-validator",
  "@type": "SoftwareApplication",
  "name": "RO-Crate validator at TRE72",
  "provider": {"@id": "https://tre72.example.com/"}
},
{ "@id": "https://tre72.example.com/",
  "@type": "Organization",
  "name": "TRE 72 trusted research environment at The University of Manchester",
  "parentOrganization": {"@id": "https://ror.org/027m9bs27"}
}
```

Check phase

Before any further processing, the content of a submitted crate SHOULD be checked for integrity and completeness against the BagIt payload manifest and tag manifest, considering at least the SHA-512 algorithm. This phase MAY also check any cryptographic signatures.

Example:

```
{
  "@id": "#check-f33fe90c-0c22-4c72-b299-de509028410e",
  "type": "AssessAction",
  "additionalType": {"@id": "https://w3id.org/shp#CheckValue"},
  "name": "BagIt checksum of Crate: OK",
  "endTime": "2023-04-18T12:11:45+01:00",
  "object": {"@id": "./"},
  "instrument": {
      "@id": "https://www.iana.org/assignments/named-information#sha-512"},
  "agent": {"@id": "#validator-a4a66c63-2fe0-4c57-830d-268a40718313"},
  "actionStatus": "http://schema.org/CompletedActionStatus"
},
{
  "@id": "https://www.iana.org/assignments/named-information#sha-512",
  "@type": "DefinedTerm",
  "name": "sha-512 algorithm"
}
```

Note that subsequent modifications to the submitted crate by the TRE will necessarily mean checksums become out of date. It is RECOMMENDED to update the BagIt manifest following crate modifications if further TRE phases require checksum (e.g. after network transfer), however any subsequent internal checksum validations SHOULD NOT be recorded as an `AssessAction`. Checksums of the final crate MUST be updated by the Publishing phase and recorded accordingly.

The check phase MAY perform any additional file-level security checks required by the particular TRE, e.g. maximum file size of crate, valid characters in filenames or use of symbolic links.

Validation phase

A crate that has been validated according to RO-Crate specifications and this profile SHOULD mention an `AssessAction` which `instrument` refers to the profile entity, and an `additionalType` referring to `https://w3id.org/shp#ValidationCheck`

Example:

```
{
  "@id": "#validate-1146f640-819e-4c86-b029-b763a0040896",
  "type": "AssessAction",
  "additionalType": {"@id": "https://w3id.org/shp#ValidationCheck"},
  "name": "Validation against Five Safes RO-Crate profile: approved",
  "startTime": "2023-04-18T12:11:46+01:00",
  "endTime": "2023-04-18T12:11:49+01:00",
  "object": {"@id": "./"},
  "instrument": {"@id": "https://w3id.org/5s-crate/0.4"},
  "agent": {"@id": "#validator-a4a66c63-2fe0-4c57-830d-268a40718313"},
  "actionStatus": "http://schema.org/CompletedActionStatus"
}
```

The validation phase MAY perform any additional syntactic or semantic checks required by the particular TRE and workflow, e.g. correspondence between provided and expected input parameters, in which case this should be reflected by adding such entities to `object` (checked) and `instrument` (expected) as arrays.

Workflow retrieval phase

The referenced workflow crate may be retrieved by the TRE before Sign-off or Workflow Execution, potentially using a local proxy (see Finding the RO-Crate archive and Security considerations). In this case, the retrieval SHOULD be indicated by a `DownloadAction` with the Workflow RO-Crate's distribution as object (indicating the URL that was downloaded from, potentially following Signposting).

Implementations MAY choose to unpack and add the Workflow Crate folder to the Bagit, in which case it SHOULD be indicated as an additional data entity referenced as result, which reference the `mainEntity` from `sameAs` and the download location in `distribution`.

Example:

```
{
  "@id": "#download-8b51bf57-6b29-44da-b24b-638c8df91639",
  "type": "DownloadAction",
  "name": "Downloaded workflow RO-Crate via proxy",
  "startTime": "2023-04-18T12:11:50+01:00",
  "endTime": "2023-04-18T12:11:52+01:00",
  "object": {"@id": "https://workflowhub.eu/workflows/289/ro_crate?version=1"},
  "result": {"@id": "workflow/289/"},
  "agent": {"@id": "http://proxy.example.com/"},
  "actionStatus": "http://schema.org/CompletedActionStatus"
},
{
  "@id": "workflow/289/",
  "sameAs": { "@id": "https://workflowhub.eu/workflows/289?version=1" },
  "@type": "Dataset",
  "name": "CWL Protein MD Setup tutorial with mutations",
  "conformsTo": {"@id": "https://w3id.org/workflowhub/workflow-ro-crate/1.0"},
  "distribution": {
    "@id": "https://workflowhub.eu/workflows/289/ro_crate?version=1"
  }
}
```

Sign-off phase

Before executing a Five Safes RO-Crate, the TRE SHOULD check if the requesting agent is permitted to execute the particular workflow on behalf of the responsible project. This SHOULD include checks against the Agreement policy data maintained by the TRE. This may be a manual and/or automated check, as indicated by `agent`. The `object` SHOULD additionally reference the workflow and responsible project (unless these were not part of the sign-off checks).

Example:

```
{
  "@id": "#signoff-3b741265-cfef-49ea-8138-a2fa149bf2f0",
  "type": "AssessAction",
  "additionalType": {"@id": "https://w3id.org/shp#SignOff"},
  "name": "Sign-off of execution according to Agreement policy: approved",
  "endTime": "2023-04-19T17:15:12+01:00",
  "object": [
      {"@id": "./"},
      {"@id": "https://workflowhub.eu/workflows/289?version=1"},
      {"@id": "#project-be6ffb55-4f5a-4c14-b60e-47e0951090c70"}
  ],
  "instrument": {"@id": "https://tre72.example.com/agreement-policy/81"},
  "agent": {"@id": "https://orcid.org/0000-0002-1825-0097"},
  "actionStatus": "http://schema.org/CompletedActionStatus"
},
{
  "@id": "https://tre72.example.com/agreement-policy/81",
  "@type": "CreativeWork",
  "name": "Agreement policy for TRE72 for project 81"
}
```

Workflow execution phase

In this phase, the approved workflow execution is performed within the TRE. The `CreateAction` of the workflow execution will use its `actionStatus` and acquire a `startTime` and `endTime`.

When the execution is in `http://schema.org/CompletedActionStatus` or `http://schema.org/FailedActionStatus`, the crate SHOULD also follow the Provenance Crate profile, e.g. the workflow outputs data entities will be listed as `result`.

Execution states

The states of the Five Safes RO-Crate is indicated by the `actionStatus` of this main action being one of the following string values:

- `http://schema.org/PotentialActionStatus` -— The request is queued to be executed
- `http://schema.org/ActiveActionStatus` -— The request is currently executing
- `http://schema.org/CompletedActionStatus` -— The request has finished successfully
- `http://schema.org/FailedActionStatus` —- The request failed, but may have partial results or logs.

Example:

```
{
  "@id": "#query-37252371-c937-43bd-a0a7-3680b48c0538",
  "@type": "CreateAction",
  "actionStatus": "http://schema.org/CompletetedActionStatus",
  "startTime": "2023-04-18T13:52:19+01:00",
  "endTime": "2023-04-18T14:00:19+01:00",
  "agent": {"@id": "https://orcid.org/0000-0001-9842-9718"},
  "instrument": {"@id": "https://workflowhub.eu/workflows/289?version=1"},
  "name": "Execute query 12389 on workflow ",
  "object": [
    {"@id": "input1.txt"}
  ],
  "result": [
    { "@id": "outputs/matches.txt"}
    { "@id": "outputs/stats.txt"}
  ]
}
```

Note that even if the workflow execution action is in `http://schema.org/CompletedActionStatus`, two additional phases are required to pass before the Five Safes RO-Crate can be considered “finished”, see the following subsections.

Disclosure phase

Before workflow results are returned from the TRE, a disclosure check SHOULD be performed, e.g. to verify the workflow execution has not revealed sensitive data. Depending on the workflow and TRE data this may be automated and/or manual as indicated by `agent`.

In the example below, a person has been assigned, while the `actionStatus` indicates the disclosure check is pending (`startTime` in this case indicating predicted waiting time into the future).

```
{
  "@id": "#disclosure-b16c1f0a-ae7f-4582-9b28-7d9df3313e27",
  "type": "AssessAction",
  "additionalType": {"@id": "https://w3id.org/shp#DisclosureCheck"},
  "name": "Disclosure check of workflow results: pending (estimate: 1 week)",
  "startTime": "2023-04-25T16:00:00+01:00",
  "object": {"@id": "./"},
  "agent": {"@id": "https://orcid.org/0000-0002-1825-0097"},
  "actionStatus": "http://schema.org/PotentialActionStatus"
}
```

If a crate fails the disclosure phase, its content such as workflow results MUST NOT be included in the returned crate returned to the user. Likewise, its workflow execution `CreateAction` and corresponding output data entities SHOULD be removed from the metadata file.

Publishing phase

Before a disclosure-approved Five Safes RO-Crate is published to the requesting user (or archived in a repository), some housekeeping tasks are to be completed.

The root data entity's `datePublished` SHOULD be updated to the time the manifest is last written. The `publisher` SHOULD be updated to reflect the executing TRE. The `mentions` MUST be expanded to include all the `AssessActions` recorded by the TRE. `hasPart` MUST include (possibly through intermediate folders as `Dataset`) any `result` data entities now referred to from the workflow execution's `CreateAction`.

The `licence` SHOULD be included to describe the licence of the workflow output data, either an open licence such as Creative Commons, or a restrictive (typically TRE-specific) conditions of access.

```
{
  "@id": "./",
  "@type": "Dataset",
  "conformsTo": {"@id": "https://w3id.org/5s-crate/0.4"},
  "datePublished": "2023-04-29T11:01:04+01:00",
  "publisher": {"@id": "https://tre72.example.com/"},
  "licence": {"@id": "http://spdx.org/licenses/CC-BY-4.0"},
  "hasPart": [{"…":""}],
  "mainEntity": {"@id": "https://workflowhub.eu/workflows/289?version=1"},
  "mentions": [{"…":""}],
  "sourceOrganization": {
    "@id": "#project-be6ffb55-4f5a-4c14-b60e-47e0951090c70"
  }
},
{
  "@id": "https://spdx.org/licenses/CC-BY-4.0",
  "@type": "CreativeWork",
  "name": "Creative Commons Attribution 4.0 International",
  "identifier": "CC-BY-4.0"
}
```

TRE implementations MAY additionally record changes to the RO-Crate as it has progressed through the execution, by associating a CreateAction and subsequent UpdateAction to the root data entity as object.

Following the final update of the RO-Crate Metadata file and content, the BagIt payload manifests and tag manifests MUST be updated (as a minimum because the `ro-crate-metadata.json` has been modified). The result entity is NOT recorded, as `../manifest-sha512.txt` would have escaped the RO-Crate root. The agent SHOULD delete manifest files it can't re-generate. After the checksum calculation, the TRE SHOULD not do any further changes to the crate or BagIt files.

```
{
  "@id": "#bagit-ce785c0b-c988-4043-8cbd-1489dcebc14f",
  "type": "UpdateAction",
  "startTime": "2023-04-29T12:12:25+01:00",
  "additionalType": {"@id": "https://w3id.org/shp#GenerateCheckValue"},
  "name": "BagIt manifests of Crate updated",
  "object": {"@id": "./"},
  "instrument": {
     "@id": "https://www.iana.org/assignments/named-information#sha-512"},
  "agent": {"@id": "#validator-a4a66c63-2fe0-4c57-830d-268a40718313"},
  "actionStatus": "http://schema.org/CompletedActionStatus"
},
{
  "@id": "https://www.iana.org/assignments/named-information#sha-512",
  "@type": "DefinedTerm",
  "name": "sha-512 algorithm"
}
```

Note: This action must be written to the RO-Crate Metadata File *before* calculating the payload manifest, and therefore can't include the correct `endTime`. The `actionStatus` should nevertheless reflect the status as if it has already completed. Likewise, the payload manifest must be calculated before updating the tag manifest, as it includes the checksum of the payload manifest.

Receiving phase

Clients receiving a Five Safes RO-Crate SHOULD check the BagIt manifest checksums similar to the Check phase, as well as the status of all the actions specified in this profile before further processing.

It is NOT sufficient for clients to check the publishing `AssessAction`, as TRE implementations are permitted to expose partial crates which have failed approval phases or which are in pending/execution state.

Clients MAY add additional post-processing data and/or metadata not specified in this profile to the crate (e.g. ReceiveAction), in which case they SHOULD maintain the BagIt manifests accordingly. Manifest checksums can be used to detect accidental local changes in post-processing.

Security considerations

It is RECOMMENDED that implementers apply strong access control before accepting a Five Safes RO-Crate.

Allowing execution of any Workflow Crate effectively allows execution of arbitrary code. It is RECOMMENDED to check against a list of pre-approved workflows (see Sign-off phase), e.g. using file checksums or cryptographic signatures.

Clients parsing and unpacking ZIP files, JSON metadata, workflow definitions and BagIt manifests SHOULD apply reasonable security measures to limit the possibility of an attacker to consume excessive disk, CPU or memory resources, as well as escaping any file directory or execution container jails. For instance, clients should check for invalid file path characters, relative paths or symbolic links escaping the crate, as well guard against *zip bomb* attacks.

Pre-approved workflows could be exploited by a malicious attacker if they, the workflow engine or their tools themselves have security vulnerabilities, e.g. by using hand-crafted input parameters that by-passes command line escapes. Workflow executions are not guaranteed to complete in a given timescale; sufficient timeouts and resource usage restrictions SHOULD therefore be applied by the workflow engine.

It is currently out of scope for this specification how to verify that a Five Safes RO-Crate was requested by the given person, or how to verify if the person has access to a particular TRE according to their *Agreement policies*. It is therefore RECOMMENDED that implementers check authentication and authorization of a submitted query and use strong encryption. Implementers SHOULD check that the `@id` and `affiliation` of the *Requesting Agent* and *Responsible Project* corresponds to the authentication, and MAY inject/overwrite client-submitted data.

Malicious clients submitting a Five Safes RO-Crate may have included additional entities, properties and types, which may cause security concerns in an implementation. Implementers SHOULD sanity check inputs, including ensuring that all paths are relative within the bag or absolute URIs, and MUST remove references to any client-submitted `AssessActions`, as these could be used to bypass the TRE compliance process.

Malicious clients MAY attempt to reference URLs or IP addresses that are only accessible within a TRE. Implementers MUST perform any URL downloads (such as Workflow RO-Crates or container images) in a way that does not access the secured TRE network, e.g. from a *Demilitarized Zone* (DMZ) with a network firewall restricting access to the TRE, or through a proxying repository controlled by TRE administrators.

As an executed Five Safes RO-Crate may be intended for publishing (possibly following an embargo period), it SHOULD NOT include sensitive data or security tokens within the metadata file or the BagIt archive (e.g. in configuration or log files); TREs SHOULD verify this in the Disclosure Phase. It is RECOMMENDED to use keychain services or time-limited security access tokens that can be assured to be expired before the Crate is published.

The crate MAY include references (e.g. S3 URIs) to sensitive data, in which case the implementation and executed workflow SHOULD protect against divulging sensitive information (directly or indirectly) in the `File` identifiers, use UUID v5 hashing [RFC4122] to hide sensitive identifiers. Note: predicable identifiers like `patient-456` would still be vulnerable in such hashing due to iteration attacks.

Media type and profiles

When transferring a Five Safes RO-Crate using HTTP, implementations SHOULD use the following HTTP headers for content-type and profile:

```
GET http://example.com/crates/42.zip HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/zip
Link: <https://w3id.org/ro/crate>; rel="profile"
```

HTML landing pages that reference a Five Safes RO-Crate SHOULD include Signposting using HTTP `Link` headers that refer to the Crate's ZIP download and the RO-Crate profile:

```
HEAD http://example.com/crates/42.html HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: text/html
Link: <https://example.com/query-12389.zip>; rel="item", type="application/zip"
Link: <https://w3id.org/ro/crate>; rel="profile"; type="application/zip";
   anchor="https://example.com/query-12389.zip"
```

Implementations MAY also provide direct public access to the RO-Crate metadata file, in which case they SHOULD follow the RO-Crate media type recommendations for JSON-LD, in which case it is RECOMMENDED to convert the metadata file to Detached RO-Crate by establishing a base URI based on the BagIt External-Identifier UUID (e.g. `arcp://uuid,9796155a-fe44-4614-89b8-71945f718ffb/`).

References

[De Geest 2022] Enhancing RDM in Galaxy by integrating RO-Crate https://doi.org/10.3897/rio.8.e95164

[FIPS 180-4] Secure Hash Standard (SHS) https://doi.org/10.6028/NIST.FIPS.180-4

[Giles 2023] TRE-FX: Delivering a federated network of trusted research environments to enable safe data analytics https://doi.org/10.5281/zenodo.10055354

[Leo 2023] Recording provenance of workflow runs with RO-Crate https://doi.org/10.48550/arXiv.2312.07852

[Provenance Run Crate] Provenance Run Crate profile https://w3id.org/ro/wfrun/provenance/0.2

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels https://doi.org/10.17487/RFC2119

[RFC3339] Date and Time on the Internet: Timestamps https://doi.org/10.17487/RFC3339

[RFC4122] A Universally Unique IDentifier (UUID) URN Namespace https://doi.org/10.17487/rfc4122

[RFC8174] Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words https://doi.org/10.17487/RFC8174

[RFC8493] The BagIt File Packaging Format (V1.0) https://doi.org/10.17487/rfc8493

[RO-Crate 1.1] RO-Crate Metadata Specification 1.1 https://w3id.org/ro/crate/1.1

[RO-Crate] Research Object Crate (RO-Crate) https://w3id.org/ro/crate

[schema] Schema.org vocabulary http://schema.org/

[schema Action] Schema.org potential action http://schema.org/docs/actions.html

[Signposting] Signposting the Scholarly Web https://signposting.org/

[Soiland-Reyes 2022] Packaging research artefacts with RO-Crate https://doi.org/10.3233/DS-210053

[Workflow RO-Crate] Workflow RO-Crate profile https://w3id.org/workflowhub/workflow-ro-crate/1.0

[Workflow Run Crate] Workflow Run Crate profile https://w3id.org/ro/wfrun/workflow/0.2

[ZIP] APPNOTE.TXT - .ZIP File Format Specification 6.3.8 http://www.pkware.com/documents/casestudies/APPNOTE.TXT