Seed Standard Definition

Jonathan Meyer

Version 0.0.1, 2016-06-20

Table of Contents

1. Introduction

Seed is a general standard to aid in the discovery and consumption of discrete units of work contained within a Docker image. While initially developed to support the Scale data processing system with job discovery, it is designed to be readily applied to other systems as well.

Seed compliant images must be named in a specific fashion due to the lack of label search capability on Docker Hub and Registry services. The suffix `-seed` must be used when naming images to enable discovery, prior to Hub or Registry push. This requirement will be deprecated as label search support is standardized across Docker registry services.

1.1. Format

Dockerfile snippet containing required label for Seed compliance:

```
FROM alpine

LABEL com.ngageoint.seed.manifest="{\"manifestVersion\":\"0.0.1\",\"jobs\":[]}"
```

The com.seed.manifest label contents must be serialized as a string-escaped JSON object. The following is a complete example including required and optional keys:

```
{
    "manifestVersion": "0.0.1",
    "jobs": [
        {
            "name": "my-algorithm",
            "version": "1.0.0",
            "title": "My first algorithm",
            "description": "Reads an HDF5 file and outputs two TIFF images, a CSV and
manifest containing cell_count",
            "tag": [
                "hdf5",
                "tiff",
                "csv",
                "image processing"
            ],
            "authorName": "John Doe",
            "authorUrl": "http://www.example.com",
            "timeout": 3600,
            "cpus": 10.0,
            "mem": 10240.0,
            "storage": 0.0,
            "interface": {
```

```
            "cmd": "/app/job.sh",
            "args": "${INPUT_FILE} ${JOB_OUTPUT_DIR}",
            "inputData": {
                "files": [
                    {
                        "name": "INPUT_FILE",
                        "required": true,
                        "mediaType": [
                            "image/x-hdf5-image"
                        ]
                    }
                ]
            },
            "outputData": {
                "files": [
                    {
                        "name": "output_file_tiffs",
                        "mediaType": "image/tiff",
                        "count": "2",
                        "pattern": "outfile*.tif"
                    },
                    {
                        "name": "output_file_csv",
                        "mediaType": "text/csv",
                        "pattern": "outfile*.csv"
                    }
                ],
                "json": [
                    {
                        "name": "cell_count",
                        "key": "cellCount",
                        "type": "integer"
                    }
                ]
            }
        },
        "errorMapping": [
            {
                "code": 1,
                "title": "Error Name",
                "description": "Error Description",
                "category": "system"
            },
            {
                "code": 2,
                "title": "Error Name",
                "description": "Error Description",
                "category": "data"
```

```
                },
                {
                    "code": 3,
                    "title": "Error Name",
                    "description": "Error Description",
                    "category": "algorithm"
                }
            ]
        }
    ]
}
```

1.2. Definitions

- GeoJSON, and the terms Geometry and Polygon are defined in IETF GeoJSON Draft-03
- Internet Assigned Numbers Authority (IANA), and the terms Media Types and MIME Types are defined in IETF RFC 6838
- ISO 8601 and the specifics of the date format are defined in IETF RFC 3339
- JavaScript Object Notation (JSON), and the terms object, name, value, array, and number, are defined in IETF RFC 4627.
- Semantic Versioning (SemVer), and the terms major, minor, and patch version are defined at http://semver.org/spec/v2.0.0.html
- 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 IETF RFC 2119.

2. Standard

The Seed standard is intended to provide a complete definition of the runtime processing, memory and storage requirements of one or more discrete units of work, in addition to the inputs, outputs and potential errors produced. Completeness is fundamental but the standard accommodates both simple and complex jobs by defining a minimal subset of REQUIRED properties. The following sections detail every possible REQUIRED and OPTIONAL manifest property in both root and child objects.

A complete Seed object contained within a com.ngageoint.seed.manifest label is always a string-escaped serialized object. In Seed, an object consists of a collection of name/value pairs — also called members. For each member, the name is always a string. Member values are either a string, number, object, array or one of the literals: true, false, and null. An array consists of elements where each element is a value as described above.

2.1. Seed Object

The Seed object is the root JSON object that MUST be placed within a com.ngageoint.seed.manifest Docker image label. At a minimum this object MUST define the `manifestVersion` and `jobs` names.

- The Seed object MUST have a member with the name `manifestVersion`. The member's value MUST be a string that conforms to the SemVer standard.
- The Seed object MUST have a member with the name `jobs`. The member's value is an array of `Job` objects and MUST contain at least 1 element. There is no other standard restriction on the array size.
- The Seed object MAY have any number of additional members.

```
{
  "manifestVersion": "0.0.1",
  "jobs": [ ... ]
}
```

2.1.1. Job Objects

The Job object is the core member for describing a single unit of work and the resources it requires.

- The Job object MUST have a member with the name `name`. The member's value MUST be a string of only lowercase alpha-numeric, dash or underscore characters (defined by the regex `[a-z0-9_-]+`).
- The Job object MUST have a member with the name `version`. The member's value MUST be a string that conforms to the SemVer standard.
- The Job object MUST have a member with the name `title`. The member's value MUST be a string and SHOULD contain a short descriptive title of the job.
- The Job object MUST have a member with the name `description`. The member's value MUST be a string and SHOULD contain a full job abstract.
- The Job object MAY have a member with the name `tag`. The member's value MUST be an array of strings and MAY contain any number of elements.
- The Job object MUST have a member with the name `authorName`. The member's value MUST be a string and SHOULD contain the authoring individual or organization.
- The Job object MAY have a member with the name `authorUrl`. The member's value MUST be a string and SHOULD contain a publicly accessible URL with further job detail.
- The Job object MUST have a member with the name `timeout`. This member's values MUST be a integer indicating a timeout period measured in seconds. Consuming systems MUST honor this value as a hard limit on job execution time.
- The Job object MUST have a member with the name `cpus`. This member's value MUST be a number indicating the whole or fractional CPUs allocated for the job.

- The Job object MUST have a member with the name `mem`. This member's value MUST be a number indicating the memory requirement in Mebibytes (MiB) for the job. Fractional MiB values MAY be used to indicate allocations below 1 MiB.
- The Job object MAY have a member with the name `storage`. This member's value MUST be a number indicating the runtime ephemeral storage requirements in Mebibytes (MiB) of the job. If omitted, the default of 0.0 may be assumed.
- The Job object MUST have a member with the name `interface`. This member's value MUST be an object as defined in `Interface`.
- The Job object MAY have a member with the name `errorMapping`. This member's value MUST be an array containing elements defined in
- The Job object MAY have any number of additional members.

The following annotated snippet provides quick reference to the use of Job object:

```
{
  "name": "my-algorithm", ①
  "version": "1.0.0", ②
  "title": "My first algorithm", ③
  "description": "Reads an HDF5 file and outputs two TIFF images, a CSV and manifest
containing cell_count", ④
  "authorName": "John Doe", ⑤
  "authorUrl": "http://www.example.com", ⑥
  "timeout": 3600,⑦
  "cpus": 10.0, ⑧
  "mem": 10240.0, ⑨
  "storage": 0.0, ⑩
  "interface": { ...  }, ⑪
  "errorMapping": [ ... ] ⑫
}
```

① Required string containing job identifier. Limited to regex `[a-z0-9_-]+`. `name` and `version` members combined should be unique system-wide.

② Required string containing job version identifier in SemVer format. `name` and `version` members combined should be unique system-wide.

③ Required string containing short job title.

④ Required string containing job abstract. Inline markup should be avoided, but not prohibited.

⑤ Required string containing job author name or organization.

⑥ Optional string containing URL to job website.

⑦ Required integer containing job timeout value in seconds.

⑧ Required number containing CPU needs of job. Whole or fractional values allowed.

⑨ Required number containing memory needs of job in Mebibytes (MiB). Whole or fractional values

allowed.

⑩ Optional number containing storage needs of job in Mebibytes (MiB). Whole or fractional values allowed.

⑪ Required [interface-section].

⑫ Optional array of [errormapping-section].

2.1.1.1. Interface Object

The Interface object is the primary member that describes the actual executable entrypoint, as well as inputs and outputs defined for a job.

- The Interface object MUST have a member with the name `cmd`. The member's value MUST be a string specifying the absolute path to the job executable within the Docker image.
- The Interface object MAY have a member with the name `args`. The member's value MUST be a string specifying the complete parameter string passed to the executable defined in `cmd` member. Based on the Linux shell, shell escaping of special characters MAY be required.
- The Interface object MAY have a member with the name `inputData`. The member's value is an array of `InputData` objects (see InputData Object) and MAY contain any number of elements.
- The Interface object MAY have a member with the name `outputData`. The member's value is an array of `OutputData` objects (see InputData Object) and MAY contain any number of elements. There is no other standard restriction on the array size.

The following annotated snippet provides quick reference to the use of Interface object:

```
{
  "cmd": "/app/job.sh", ①
  "args": "${INPUT_FILE} ${JOB_OUTPUT_DIR}", ②
  "inputData": { "files": [ { "name": "INPUT_FILE", ... }, ... ] }, ③
  "outputData": { ... } ④
}
```

① Required string indicating the job executable. It should be an absolute path for clarity.

② Optional string indicating the job arguments. Shell expansion may be used to inject existing environment variables and executable arguments. Linux shell escaping may be needed in the case of special characters.

③ Optional InputData object. This is the means to inject external data into the job container.

④ Optional OutputData object. This is the means to capture results from the job container.

2.1.1.1.1. InputData Object

The InputData object is the member responsible for indicating all mutable content available to the Seed image at runtime.

- The InputData object MAY have a member `files`. The member's value is an array of objects defined in Files Elements sub-section.
- The InputData object MAY have a member `json`. The member's value is an array of objects defined in JSON Elements sub-section.

Files Elements

- The Files object MUST have a member `name`. The member's value MUST be a string indicating the environment variable name that will be injected by the processing platform for job consumption.
- The Files object MUST have a member `mediaType`. The member's value MUST indicate the IANA Media type for the file being captured by OutputData.
- The Files object MAY have a member `required`. The member's value MUST be a boolean indicating whether this input value SHOULD always be expected. If omitted, the default value MUST be treated as true.

JSON Elements

- The JSON object MUST have a member `name`. The member's value MUST be a string indicating the environment variable name that will be injected by the processing platform for job consumption.
- The JSON object MUST have a member `type`. The member's value MUST be a string and indicate a valid JSON schema type.
- The JSON object MAY have a member `required`. The member's value MUST be a boolean indicating whether this input value SHOULD always be expected. If omitted, the default value MUST be treated as true.

The following annotated snippet provides quick reference to the use of InputData object:

```
{
  "files": [ ①
    {
      "name": "INPUT_FILE", ②
      "mediaType": [ "image/x-hdf5-image" ], ③
      "required": true ④
    }
  ]
  "json": [ ⑤
    {
      "name": "INPUT_STRING",
      "type": "string", ⑥
      "required": false ⑦
    }
  ]
}
```

① Optional array containing elements defined by Files Elements sub-section.

② Required string containing name used to inject data via environment variables.

③ Required array containing a list of accepted Media types.

④ Optional boolean indicating whether job requires this particular file. Default is `true`.

⑤ Optional array containing elements defined by JSON Elements sub-section.

⑥ Required string containing a valid JSON schema type for input validation.

⑦ Optional boolean indicating whether job requires this particular JSON input. Default is `true`.

2.1.1.1.2. OutputData Object

The OutputData object is the member responsible for indicating all output data and the means to capture that data following the execution of a Seed image. Data can be captured in two different forms: directly as a file or extracted JSON from a manifest. File type output is simply matched based on a standard glob pattern. JSON objects are expected to be gathered from a results manifest that by Seed standard convention MUST be provided either on container STDOUT or written at the root of the job output directory as `results_manifest.json`. The location of the job output directory is REQUIRED to be passed into the container at job execution time.

- The OutputData object MAY have a member `files`. The member's value is an array of objects defined in Files Elements sub-section.
- The OutputData object MAY have a member `json`. The member's value is an array of objects defined in JSON Elements sub-section.

Files Elements

- The Files object MUST have a member `name`. The member's value MUST be a string indicating the key the processing system will place the file name captured for downstream processing. variable name that will be injected by the processing platform for job consumption.
- The Files object MUST have a member `mediaType`. The member's value MUST indicate the IANA Media type for the file being captured by OutputData.
- The Files object MUST have a member `pattern`. The member's value MUST indicate a standard glob pattern for the capture of files.
- The Files object MAY have a member `count`. The member's value MUST be a string that accepts 2 possibilities: positive numeric values or a `*`. Numeric values indicate an explicit match expected for `pattern` while `*` indicates matching with no upper bound.
- The Files object MAY have a member `required`. The member's value MUST be a boolean indicating whether this input value SHOULD always be expected. If omitted, the default value is `true`.

JSON Elements

- The JSON object MUST have a member `name`. The member's value MUST be a string indicating the key the processing system will place the JSON member value in for downstream use. When `key` member is omitted, it must be a case-sensitive match of the member key in result manifest.
- The JSON object MUST have a member `type`. The member's value MUST be a string and indicate the

JSON schema type of the member being captured from the result manifest.

- The JSON object MAY have a member `key`. The member's value MUST be a string and indicate the case-sensitive result manifest member to capture. If omitted, the result member key is assumed to be a case-sensitive match for the above defined `name` member.
- The JSON object MAY have a member `required`. The member's value MUST be a boolean indicating whether this input value SHOULD always be expected. If omitted, the default value MUST be treated as true.

The following annotated snippets provides quick reference to the use of OutputData object:

Result Manifest

```
{
  "cellCount": 256,
  ...
}
```

Seed Manifest - OutputData object

```
"outputData": {
  "files": [ ①
    {
      "name": "OUTPUT_TIFFS", ②
      "mediaType": "image/tiff", ③
      "pattern": "outfile*.tif", ④
      "count": "2", ⑤
      "required": true ⑥
    }
  ],
  "json": [ ⑦
    {
      "name": "CELL_COUNT", ⑧
      "type": "integer", ⑨
      "key": "cellCount ⑩
    }
  ]
}
```

① Optional array containing elements defined by Files Elements sub-section.

② Required string containing output identifier.

③ Required string containing IANA Media type of file.

④ Required string containing glob expression for file capture. Processing system is expected to capture output relative to JOB_OUTPUT_DIR.

⑤ Optional string containing either a numeric count or `\*` for unbounded output. Default value is `1`.

⑥ Optional boolean indicating whether processing system should assume failure if output data is missing. Default value is true.

⑦ Optional array containing elements defined by JSON Elements sub-section.

⑧ Required string containing output identifier. MUST be used by processing framework to match member for capture from result manifest in absence of `key` member.

⑨ Required string containing JSON schema type of member extracted from result manifest.

⑩ Optional string containing key of result manifest member for extraction. This allows mapping from a result manifest member key that differs from the value of `name` member.

2.1.1.2. ErrorMapping Objects

The ErrorMapping objects allow for job developers to map arbitrary exit codes to meaningful textual descriptions. This is useful in passing information to the processing system to differentiate between data and algorithm errors.

- The ErrorMapping object MUST have a member `code`. The member's value MUST be an integer indicating the exit code of the executing job process.
- The ErrorMapping object MUST have a member `title`. The member's value MUST be a string indicating the short descriptive title of the error.
- The ErrorMapping object MAY have a member `description`. The member's value MUST be a string indicating the complete error description and possible causes.
- The ErrorMapping object MAY have a member `category`. If omitted, the default value is `algorithm`. The member's value MUST be a string containing one of the following values: `algorithm`, `data` or `system`.

The following annotated snippet provides quick reference to the use of ErrorMapping objects:

```
[
  {
    "code": 1, ①
    "title": "Error Name", ②
    "description": "Error Description", ③
    "category": "system" ④
  }
]
```

① Required integer indicating job process exit code.

② Required string containing human-friendly short name of error.

③ Optional string containing complete error code description.

④ Optional string containing the error type. This value MUST be either: `algorithm`, `data` or `system`. The default value is `algorithm`.

3. Usage

3.1. Implementing

A few requirements must be satisfied when implementing a system capable of executing Seed standardized images.

3.1.1. Environment Variables

Environment variable injection must be performed. These may be consumed by job directly as environment variables or shell variable expansion may be leveraged in the `interface.args` member.

The following minimum variables MUST be provided:

- JOB_OUTPUT_DIR: Root path all output products must be placed by job for processing system capture.
- ALLOCATED_CPUS: Value of `cpus` member.
- ALLOCATED_MEM: Value of `mem` member.
- ALLOCATED_STORAGE: Value of `storage` member.
- All `name` member values of `interface.inputData.files` elements MUST map to environment variables. Name case MUST map exactly from `name` value. It is the processing framework's responsibility to ensure data is mounted and to ensure the path is container resolvable.
- All `name` member values of `interface.inputData.json` elements MUST map to environment variables. Name case MUST map exactly from `name` value and `array`, `object` and `string` JSON types MUST be injected string encoded.

3.1.2. Output File metadata

There is often a need by the processing system to capture additional job extracted metadata on output files. The Seed standard allows for this through the use of side-car files. The side-car files must be named exactly as the file they describe, with the addition of the `.metadata.json` extension to the original file name (extension included). The file must be formatted according to the Seed Metadata Schema. This allows for both spatial and temporal metadata to be specified.

The following snippet demonstrates the optional values that may be specified:

Metadata JSON

```
{
    "geometry": { ①
        "type": "Polygon",
        "coordinates": [
            [ [ 100.0, 0.0 ], [ 101.0, 0.0 ], [ 101.0, 1.0 ], [ 100.0, 1.0 ], [ 100.0,
0.0 ] ]
        ]
    },
    "time": { ②
        "start": "2016-08-06T00:00:00.000Z", ③
        "end": "2016-08-06T00:00:00.000Z" ④
    }
}
```

① Optional GeoJSON Geometry object defining spatial extent of file.

② Optional Time object defining temporal extent of file.

③ Required string containing an ISO 8601 date indicating the start temporal extent of file. String must include full time down to minute precision.

④ Required string containing an ISO 8601 date indicating the end temporal extent of file. String must include full time down to minute precision. If the data is a single point-in-time the end date should match the start date.

3.2. Examples

The Seed standard is intended to support both simple and complex job packaging. To that end the standard allows for sensible defaults to take the place of a fully specified manifest. The following examples identify both a minimal Seed use and a more realistic, fully exercised standard.

3.2.1. Random Number Generator Job

Minimal manifest demonstrating the simplest possible Seed definition.

```
{
  "manifestVersion": "0.0.1",
  "jobs":
  [
    {
      "name": "random-number-gen",
      "version": "0.1.0",
      "title": "Random Number Generator",
      "description": "Generates a random number and outputs on stdout",
      "authorName": "John Doe",
      "timeout": 10,
      "cpus": 0.1,
      "mem": 1.0,
      "interface": {
        "cmd": "/app/job.sh"
      }
    }
  ]
}
```

Serialized as a label in a Dockerfile snippet:

```
FROM alpine

LABEL
com.ngageoint.seed.manifest="{\"manifestVersion\":\"0.0.1\",\"jobs\":[{\"name\":\"random-
number-gen\",\"version\":\"0.1.0\",\"title\":\"Random Number
Generator\",\"description\":\"Generates a random number and outputs on
stdout\",\"authorName\":\"John
Doe\",\"timeout\":10,\"cpus\":0.1,\"mem\":1.0,\"interface\":{\"cmd\":\"\/app\/job.sh\"}}]
}"
```

3.2.2. Image Watermark Job

Image watermark job taking a single image and returning with watermark applied using Seed definition.

```
{
    "manifestVersion": "0.0.1",
    "jobs": [
        {
            "name": "image-watermark",
            "version": "0.1.0",
            "title": "Image Watermarker",
            "description": "Processes an input PNG and outputs watermarked PNG.",
            "authorName": "John Doe",
            "timeout": 30,
            "cpus": 1.0,
            "mem": 64.0,
            "interface": {
                "cmd": "/app/watermark.py",
                "args": "${INPUT_IMAGE} ${JOB_OUTPUT_DIR}",
                "inputData": {
                    "files": [
                        {
                            "name": "INPUT_IMAGE",
                            "mediaType": [
                                "image/png"
                            ]
                        }
                    ]
                },
                "outputData": {
                    "files": [
                        {
                            "name": "OUTPUT_IMAGE",
                            "mediaType": "image/png",
                            "pattern": "*_watermark.png"
                        }
                    ]
                }
            },
            "errorMapping": [
                {
                    "code": 1,
                    "title": "Image Corrupt",
                    "description": "Image input is not recognized as a valid PNG.",
                    "category": "data"
                }
            ]
        }
    ]
}
```

Serialized as a label in a Dockerfile snippet:

```
FROM alpine

LABEL
com.ngageoint.seed.manifest="{\"manifestVersion\":\"0.0.1\",\"jobs\":[{\"name\":\"image-
watermark\",\"version\":\"0.1.0\",\"title\":\"Image
Watermarker\",\"description\":\"Processes an input PNG and outputs watermarked
PNG.\",\"authorName\":\"John
Doe\",\"timeout\":30,\"cpus\":1.0,\"mem\":64.0,\"interface\":{\"cmd\":\"\/app\/watermark.
py\",\"args\":\"${INPUT_IMAGE}
${JOB_OUTPUT_DIR}\",\"inputData\":{\"files\":[{\"name\":\"INPUT_IMAGE\",\"mediaType\":[\"
image\/png\"]}]},\"outputData\":{\"files\":[{\"name\":\"OUTPUT_IMAGE\",\"mediaType\":\"im
age\/png\",\"pattern\":\"*_watermark.png\"}]}},\"errorMapping\":[{\"code\":1,\"title\":\"
Image Corrupt\",\"description\":\"Image input is not recognized as a valid
PNG.\",\"category\":\"data\"}]}]}"
```

4. Discovery

A primary intention of this standard is for simple job discovery from public images hosted within either Docker Hub, Docker Trusted Registry or Docker Registry. There is significant fragmentation of APIs between the various Docker offerings and the following sections describe the steps that may be taken to access the labels defined by Seed.

None of the Docker registry services support label search in any fashion. This incurs the requirement of applying a secondary means to subset image results. The standard presently requires that all job image are named with the suffix `-seed`. This allows for quick filtering of results to a manageable set for discovery.

4.1. Docker Hub

Docker Hub stores Docker image manifest information in a readily accessible format only for Automated Builds. This enforces the need for all developers wishing to support simple discovery from Docker Hub to support Hub builds, as opposed to local image builds followed by a docker push. Given this caveat, a service such as ImageLayers can be used to quickly identify manifest content after discovering available images.

The following two steps may be taken to find and identify labels within Docker Hub:

- Perform HTTP GET to find Docker images:
  - URL: https://hub.docker.com/v2/search/repositories/?query=-seed
- Perform HTTP POST to get label details for images found in previous request:

- BODY: {"repos":[{"name":"myorg/myjob-seed","tag":"latest"}]}
- URL: https://imagelayers.io/registry/analyze

The ImageLayers service is a 3rd-party service by CenturyLink Labs, but the source code is openly available at ImageLayers and can be used as a reference implementation.

4.2. Docker Registry

Docker Registry does not natively support any type of search, but does provide a catalog API that allows for listing the entire registry contents. Using this along with tag and manifest inspection will allow label inspection.

The following steps may be taken to find and identify labels of Seed compliant images within Docker Registry:

All references to `{registry}`, `{image-id}` and `{tag}` in the following URLs should be replaced with your environment specific values.

- Perform HTTP GET against catalog endpoint to find `-seed` suffixed images:
  - URL: http://{registry}/v2/_catalog
- Perform HTTP GET against tags endpoint for each image matched:
  - http://{registry}/v2/{image-id}/tags/list
- Perform HTTP GET against manifests endpoint to retrieve labels per tag (extract labels from history JSON member):
  - http://{registry}/v2/{image-id}/manifests/{tag}

4.3. Docker Trusted Registry

There is a ticket in with Docker Trusted Registry team to natively support label search. Presently there is no API support to inspect hosted images for label metadata. Images must be pulled locally for inspection.

5. Schema

The following JSON Schema should be used to validate Seed manifests prior to label serialization into a Dockerfile for publish. It may be downloaded here: Seed Manifest Schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
```

```
    "properties": {
        "manifestVersion": {
            "type": "string",
            "pattern": "(0|[1-9][0-9]*)\\.(0|[1-9][0-9]*)\\.(0|[1-9][0-9]*)(-(0|[1-9][0-
9]*|[0-9]*[a-zA-Z-][0-9a-zA-Z-]*)(\\.(0|[1-9][0-9]*|[0-9]*[a-zA-Z-][0-9a-zA-Z-
]*))*)?(\\+[0-9a-zA-Z-]+(\\.[0-9a-zA-Z-]+)*)?"
        },
        "jobs": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "name": {
                        "type": "string",
                        "pattern": "[a-z0-9_-]+"
                    },
                    "version": {
                        "type": "string",
                        "pattern": "(0|[1-9][0-9]*)\\.(0|[1-9][0-9]*)\\.(0|[1-9][0-9]*)(-
(0|[1-9][0-9]*|[0-9]*[a-zA-Z-][0-9a-zA-Z-]*)(\\.(0|[1-9][0-9]*|[0-9]*[a-zA-Z-][0-9a-zA-Z-
]*))*)?(\\+[0-9a-zA-Z-]+(\\.[0-9a-zA-Z-]+)*)?"
                    },
                    "title": {
                        "type": "string"
                    },
                    "description": {
                        "type": "string"
                    },
                    "tag": {
                        "type": "array",
                        "items": {
                            "type": "string"
                        }
                    },
                    "authorName": {
                        "type": "string"
                    },
                    "authorUrl": {
                        "type": "string"
                    },
                    "timeout": {
                        "type": "integer"
                    },
                    "cpus": {
                        "type": "number"
                    },
                    "mem": {
                        "type": "number"
```

```
                    },
                    "storage": {
                        "type": "number",
                        "default": 0.0
                    },
                    "interface": {
                        "type": "object",
                        "properties": {
                            "cmd": {
                                "type": "string"
                            },
                            "args": {
                                "type": "string"
                            },
                            "inputData": {
                                "type": "object",
                                "properties": {
                                    "files": {
                                        "type": "array",
                                        "items": {
                                            "type": "object",
                                            "properties": {
                                                "name": {
                                                    "type": "string"
                                                },
                                                "required": {
                                                    "type": "boolean",
                                                    "default": true
                                                },
                                                "mediaType": {
                                                    "type": "array",
                                                    "items": {
                                                        "type": "string"
                                                    }
                                                }
                                            },
                                            "required": [
                                                "name",
                                                "mediaType"
                                            ]
                                        }
                                    },
                                    "json": {
                                        "type": "array",
                                        "items": {
                                            "type": "object",
                                            "properties": {
                                                "name": {
```

```
                                    "type": "string"
                                },
                                "required": {
                                    "type": "boolean",
                                    "default": true
                                },
                                "type": {
                                    "type": "string",
                                    "enum": [
                                        "array",
                                        "boolean",
                                        "integer",
                                        "number",
                                        "object",
                                        "string"
                                    ]
                                }
                            },
                            "required": [
                                "name",
                                "type"
                            ]
                        }
                    }
                }
            },
            "outputData": {
                "type": "object",
                "properties": {
                    "files": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "name": {
                                    "type": "string"
                                },
                                "mediaType": {
                                    "type": "string"
                                },
                                "pattern": {
                                    "type": "string"
                                },
                                "count": {
                                    "type": "string",
                                    "default": "1",
                                    "pattern": "([0-9]+|\\*)"
                                },
```

```
                                "required": {
                                    "type": "boolean",
                                    "default": true
                                }
                            },
                            "required": [
                                "name",
                                "mediaType",
                                "pattern"
                            ]
                        }
                    },
                    "json": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "name": {
                                    "type": "string"
                                },
                                "key": {
                                    "type": "string"
                                },
                                "type": {
                                    "type": "string",
                                    "enum": [
                                        "array",
                                        "boolean",
                                        "integer",
                                        "number",
                                        "object",
                                        "string"
                                    ]
                                },
                                "required": {
                                    "type": "boolean",
                                    "default": true
                                }
                            },
                            "required": [
                                "name",
                                "type"
                            ]
                        }
                    }
                }
            },
            "required": [
```

```
                            "cmd"
                        ]
                    }
                },
                "errorMapping": {
                    "type": "array",
                    "items": {
                        "type": "object",
                        "properties": {
                            "code": {
                                "type": "integer"
                            },
                            "title": {
                                "type": "string"
                            },
                            "description": {
                                "type": "string"
                            },
                            "category": {
                                "type": "string",
                                "default": "algorithm",
                                "enum": [
                                    "algorithm",
                                    "data",
                                    "system"
                                ]
                            }
                        },
                        "required": [
                            "code",
                            "title"
                        ]
                    }
                }
            },
            "required": [
                "name",
                "version",
                "title",
                "description",
                "authorName",
                "timeout",
                "cpus",
                "mem",
                "interface"
            ]
        }
    }
```

```
    },
    "required": [
        "manifestVersion",
        "jobs"
    ]
}
```