



MONARC Objects Sharing Platform

CASES Luxembourg

Version 2019-03-04

Table of Contents

1. Introduction	1
2. Objects involved in MOSP	1
2.1. JSON schema	1
2.1.1. Example of schemas	1
2.2. Objects	1
3. Using the platform	1
3.1. Creating a new object	1
3.2. Select a schema	2
3.3. Edit object definition	2
3.4. Exporting an object for use in MONARC	3
4. Using the API	4
4.1. Endpoint for Schemas	4
4.2. Endpoint for JSON objects	4
4.2.1. Getting the list of JSON objects	4
4.2.2. Creating a new object	5
4.3. More complex queries	6
5. Permissions	7
5.1. Schemas	7
5.2. Objects	7

1. Introduction

This document is intended to be a presentation and a user guide to [MOSP](#), a platform to create, edit and share valid JSON objects.

MOSP supports any kind of JSON objects, you just have to specify a [JSON schema](#). The instance [objects.monarc.lu](#), operated by [CASES](#), is dedicated to gather security related objects in the first place aimed to be used with [MONARC](#).

Objects officially provided by the MONARC project are available [here](#). Note that any **object** created with a **JSON schema** provided by the MONARC project will be compatible with the MONARC software. In other words, you can use our JSON schemas in order to create objects (referentials, mapping, etc.) for a usage in your own security analysis.

2. Objects involved in MOSP

The core concept of MOSP lies on the [JSON schemas](#) and [JSON Objects](#). Each objects you create within MOSP will conform to a selected schema.

2.1. JSON schema

A JSON schema describes and validates your existing data format(s). This ensures the quality of client submitted data. MOSP also uses JSON schemas for the generation of the object editor.



JSON schemas are always public.

For example, the JSON schema available [here](#) validates the Security referentials used in MONARC.

2.1.1. Example of schemas

- [Security referentials](#)
- [Security referentials mapping](#)

2.2. Objects

3. Using the platform

3.1. Creating a new object

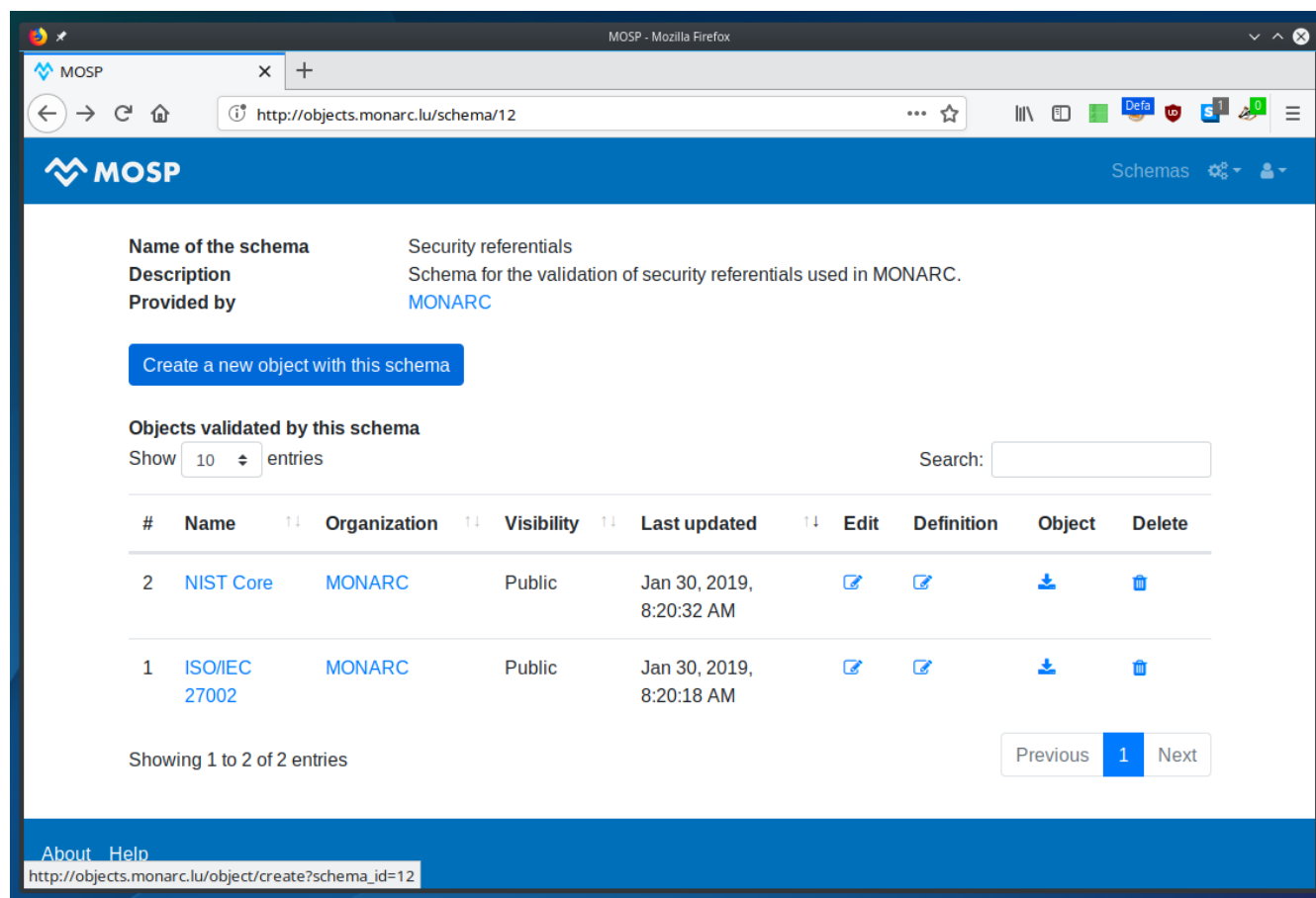
When creating a new object with MOSP you have to:

- choose a JSON schema from the ones available on the platform;
- specify a name and a brief description;
- choose an owning organization (a user account can be bound to one or multiple organizations);

- specify if the object will be public or private. Please limit the use of private objects. Our goal is to empower the sharing of information. Private objects can be useful for drafts.
- choose at least one license from the proposed.

You can use any available schemas in order to create shareable JSON objects.

3.2. Select a schema



Name of the schema Security referentials
Description Schema for the validation of security referentials used in MONARC.
Provided by [MONARC](#)

[Create a new object with this schema](#)

Objects validated by this schema
Show entries Search:

#	Name	Organization	Visibility	Last updated	Edit	Definition	Object	Delete
2	NIST Core	MONARC	Public	Jan 30, 2019, 8:20:32 AM	Edit	Definition	Object	Delete
1	ISO/IEC 27002	MONARC	Public	Jan 30, 2019, 8:20:18 AM	Edit	Definition	Object	Delete

Showing 1 to 2 of 2 entries Previous **1** Next

[About](#) [Help](#)
http://objects.monarc.lu/object/create?schema_id=12

3.3. Edit object definition

The screenshot shows the MOSP web interface for editing a NIST Core object. The interface is divided into three main sections: a form for editing the object, a JSON output preview, and a validation status.

Form Section:

- authors:** A list of authors. Item 1 is "The MONARC project".
- label:** A text field containing "NIST Core".
- measures:** A list of measures. Item 1 is "Asset Management (ID.AM)". Item 2 is "1_ID.AM-3". Item 3 is "Organizational communication and data flows are mapped".
- category:** A text field containing "Asset Management (ID.AM)".
- code:** A text field containing "1_ID.AM-3".
- label:** A text field containing "Organizational communication and data flows are mapped".
- uuid:** A text field containing "b0ceb168-a023-40af-ba24-e59bd4a45c90".

JSON Output Section:

You can also make changes to the JSON below.

```
{
  "authors": [
    "The MONARC project"
  ],
  "label": "NIST Core",
  "measures": [
    {
      "category": "Asset Management (ID.AM)",
      "code": "1_ID.AM-1",
      "label": "Physical devices and systems within the organization are inventoried",
      "uuid": "231fc2b1-88c2-458e-9d88-"
    }
  ]
}
```

Validation Section:

This will update whenever the form changes to show validation errors if there are any.

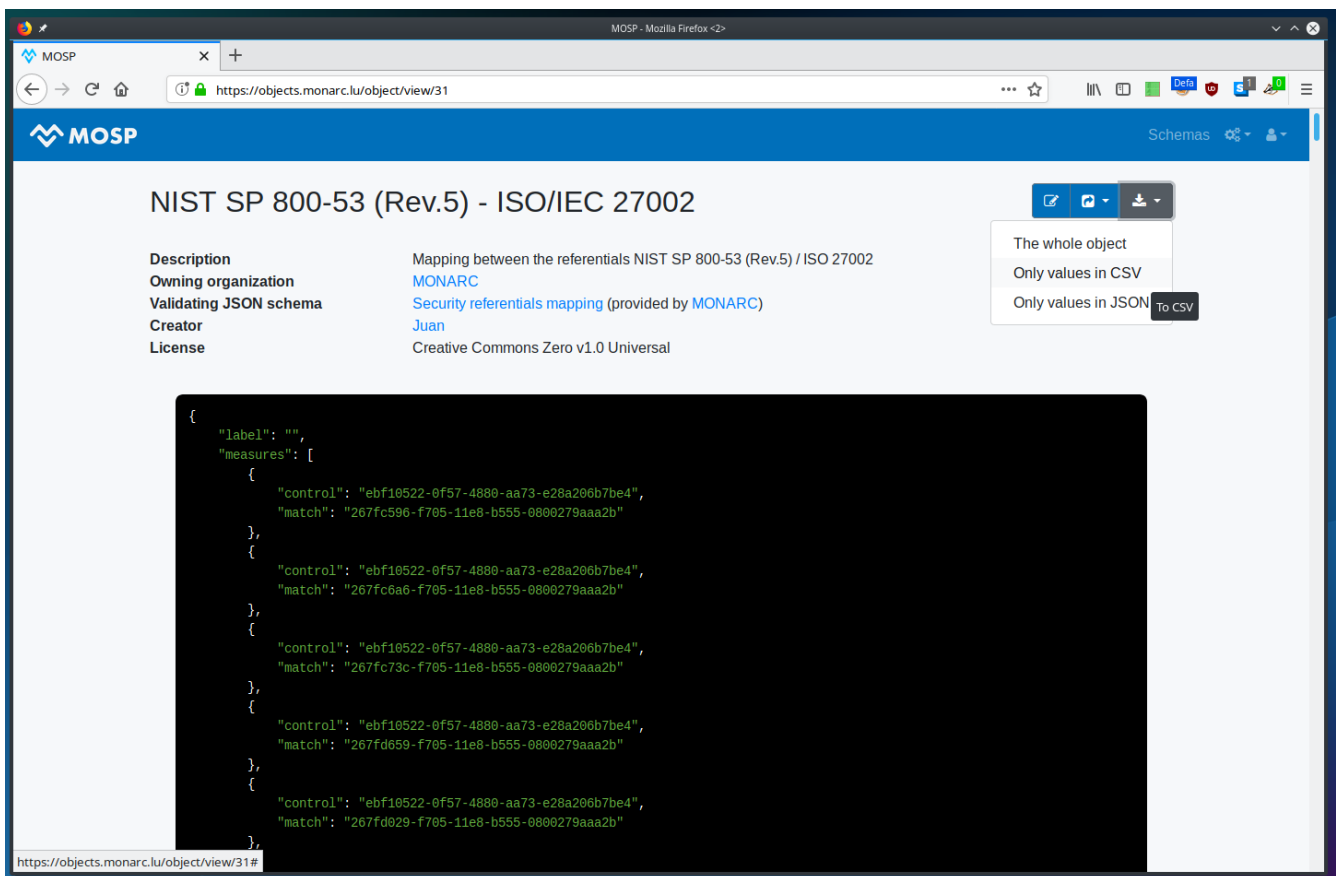
Object is valid.

The form above is generated thanks to the JSON schema. Using this form ensures that you will create valid objects.

If you update the schema related to an already defined object, eventual validation errors will be displayed in the gray box at the right of the interface.

3.4. Exporting an object for use in MONARC

Objects exported from MOSP can be imported in your projects via the [knowledge base](#) of MONARC.



MONARC will be able to handle the CSV or the JSON export.

4. Using the API

An API is provided in order to interact with the service programmatically.

4.1. Endpoint for Schemas

Getting the list of schemas:

```
https://objects.monarc.lu/api/v1/schema
```

Getting information about a schema:

```
https://objects.monarc.lu/api/v1/schema/13
```

4.2. Endpoint for JSON objects

4.2.1. Getting the list of JSON objects

```
https://objects.monarc.lu/api/v1/json_object
```

4.2.2. Creating a new object

```
$ json_object='{}'
$ curl -i --user username:password \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" -X POST \
  -d '{"name":"NIST","description":"Recommended Security Controls for Federal
Information Systems and Organizations. (Rev.5)"
,"org_id":2,"json_object":"$json_object"}' \
  https://objects.monarc.lu/api/v1/json_object
```

```
HTTP/1.0 400 BAD REQUEST
Content-Type: application/json
Content-Length: 85
Vary: Cookie
```

```
{
  "message": "You are not allowed to create/edit object from this organization."
}
```

```
$ json_object='{}'
$ curl -i --user username:password \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" -X POST \
  -d '{"name":"NIST","description":"Recommended Security Controls for Federal
Information Systems and Organizations. (Rev.5)"
,"org_id":3,"json_object":"$json_object"}' \
  https://objects.monarc.lu/api/v1/json_object
```

```
HTTP/1.0 400 BAD REQUEST
Content-Type: application/json
Content-Length: 85
Vary: Cookie
Date: Thu, 21 Feb 2019 09:07:26 GMT
```

```
{
  "message": "You must provide the id of a schema."
}
```

```
$ json_object='{}'
$ curl -i --user username:password \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" -X POST \
  -d '{"name":"NIST","description":"Recommended Security Controls for Federal
Information Systems and Organizations. (Rev.5)"
,"org_id":3,"schema_id":12,"json_object":"$json_object"}' \
  https://objects.monarc.lu/api/v1/json_object

HTTP/1.0 400 BAD REQUEST
Content-Type: application/json
Content-Length: 85
Vary: Cookie
Date: Thu, 21 Feb 2019 09:07:26 GMT

{
  "message": "The object submitted is not validated by the schema."
}
```

```
$ json_object='{ "label": "NIST SP 800-53", "measures": [ { "category": "Access
Control", "code": "AC-1", "label": "Access Control Policy and Procedures", "uuid":
"ebf10522-0f57-4880-aa73-e28a206b7be4" } ], "uuid": "cfd2cd50-95fa-4143-b0e5-
794249bacae1", "version": "5.0" }'
$ curl -i --user username:password \
  -H "Content-Type: application/json" \
  -H "Accept: application/json" -X POST \
  -d '{"name":"NIST","description":"Recommended Security Controls for Federal
Information Systems and Organizations. (Rev.5)"
,"org_id":3,"schema_id":12,"json_object":"$json_object"}' \
  https://objects.monarc.lu/api/v1/json_object

HTTP/1.0 201 CREATED
Content-Type: application/json
Content-Length: 2392
Location: https://objects.monarc.lu/api/v1/json_object/30
Vary: Accept, Cookie
Content-Type: application/json
Date: Thu, 21 Feb 2019 09:34:39 GMT
```

The content of the newly created object is also returned.

4.3. More complex queries

Getting all objects owned by the MONARC organization:


```
https://objects.monarc.lu/api/v1/json_object?q={"name":"organization","op":"has","val": {"name":"name","op":"eq","val": "MONARC"}}
```

Getting all schemas owned by the MONARC organization:

```
https://objects.monarc.lu/api/v1/schema?q={"filters":[{"name":"organization","op":"has","val":{"name":"name","op":"eq","val":"MONARC"}}]}
```

```
https://objects.monarc.lu/api/v1/json_object?q={"name":"schema","op":"has","val":{"name":"name","op":"eq","val": "Security referentials"}}
```

5. Permissions

This section details the management of permissions in MOSP.

5.1. Schemas

Schemas are always public.

You can only create an object in one of the organizations your account is linked to.

In order to edit a schema your account must be linked to the owning organization of the schema.

5.2. Objects

An object can be public or private (draft mode).

You can only create an object in one of the organizations your account is linked to. Since all schemas are public you can instantiate a new object with the schema of your choice.

In order to see a private object or to edit an object your account should be linked to the owning organization of this object.