

EMREG: EMREX Registry Specification

Table of Contents

1.	Introduction
2.	EMREG Architecture
3.	Data Models
o	3.1. EMP (EMREX Provider) Data Model
o	3.2. EMC (EMREX Client) Data Model
o	3.3. Supporting Data Models
4.	Registration Process
o	4.1. EMP Registration
o	4.2. EMC Registration
o	4.3. Approval Workflow
o	4.4. Activation Schedule
5.	API Endpoints
6.	Security Considerations
7.	Governance

Introduction

The **EMREX Registry (EMREG)** serves as the central metadata repository for the EMREX network, enabling secure and interoperable exchange of educational results between institutions across Europe. EMREG acts as the **single source of truth** for:

- **EMP (EMREX Provider)** metadata: Institutions that provide educational results (e.g., universities, national education agencies).
- **EMC (EMREX Client)** metadata: Institutions that request educational results (e.g., home universities).
- **Public keys:** Used for JWT signature validation and response signing.
- **Supported data formats:** ELMO XML/JSON, PDF, EIDAS UserInfo, etc.

EMREG ensures **trust, interoperability, and compliance** with GDPR and eIDAS regulations by maintaining accurate and up-to-date metadata for all participants in the EMREX network.

EMREG Architecture

EMREG follows a **centralized registry model** with the following components:

1. **Registry Database:** Stores metadata for EMPs and EMCs.
2. **Admin Web Portal:** For managing the register.
3. **Self-Service Portal:** For EMP/EMC registration and updates.
4. **REST API:** For programmatic access to registry data (used by EMCs/EMP during OAuth2 flows).

Data Models

3.1. EMP (EMREX Provider) Data Model

An **EMP** represents an institution or service that provides educational results (e.g., grades, diplomas).

Attribute	Type	Description	Required	Example
name	String	Official name of the EMP.	Yes	"DUO (Dienst Uitvoering Onderwijs)"
adminEmail	String	Contact email for administrative purposes.	Yes	admin@duo.nl
website	URL	Official website of the EMP.	Yes	https://www.duo.nl
publicKey	String (PEM)	Public key for JWT signature validation and response signing.	Yes	-----BEGIN PUBLIC KEY-----...
emreg2Props	Object	EMREX 2.0-specific properties (see below).	Yes	
institutions	Institution[]	List of institutions represented by this EMP.	No	
country	Country	Country where the EMP is based (see3.3).	Yes	

emreg2Props Object

Attribute	Type	Description	Required	Example	Note
supportedDataFormats	DataFormat[]	List of supported data formats (e.g., ELMO XML, ELM JSON).	Yes		@TODO: HOW TO MANAGE DATA FORMATS
availableEvidenceTypes	EvidenceType[]	Types of evidence supported (e.g., diplomas, transcripts).	No		@TODO: NEEDS DISCUSSION
authorizationUrl	URL	OAuth2 authorization endpoint (RFC 6749).	Yes	https://emp.duo.nl/oauth2/authorize	
tokenUrl	URL	OAuth2 token endpoint (RFC 6749).	Yes	https://emp.duo.nl/oauth2/token	
resourceUrl	URL	Resource server endpoint for fetching results.	Yes	https://emp.duo.nl/results	
requiredTrustLevel	String	Minimum trust level required for EMCs (e.g., "substantial", "high").	No	substantial	@TODO: NEEDS DISCUSSION
extensionOpenapiSpec	URL	Optional url to an openapi spec where EMP describes additional resources	No	https://emp.duo.nl/results/api-docs.yaml	
pingUrl	URL	Endpoint for health checks.	No	https://emp.duo.nl/ping	

Example EMP JSON

```
{
  "name": "DUO (Dienst Uitvoering Onderwijs)",
  "adminEmail": "admin@duo.nl",
  "website": "https://www.duo.nl",
  "publicKey": "-----BEGIN PUBLIC KEY-----...",
  "emreg2Props": {
    "supportedDataFormats": [
    ],
    "authorizationUrl": "https://emp.duo.nl/oauth2/authorize",
    "tokenUrl": "https://emp.duo.nl/oauth2/token",
    "resourceUrl": "https://emp.duo.nl/results",
    "requiredTrustLevel": "substantial",
    "pingUrl": "https://emp.duo.nl/ping"
  },
  "institutions":,
  "country": {
    "isoCode": "NL",
    "singleFetch": false
  }
}
```

3.2. EMC (EMREX Client) Data Model

An **EMC** represents an institution that requests educational results (e.g., a home university).

Attribute	Type	Description	Required	Example

clientId Attribute	String Type	Unique identifier for the EMC. Description	Yes Required	emc-nl-uva Example
redirectUri	URL	OAuth2 redirect URI (must match exactly during validation).	Yes	https://emc.uva.nl/callback
name	LocalizedString	Localized name of the EMC (supports multiple languages).	Yes	{ "en": "UVA", "nl": "UvA" }
logo	URL	URL to the EMC's logo.	No	https://emc.uva.nl/logo.png
country	String	Country where EMC is based (ISO Country code)	Yes	NL
trustLevel	String	Trust level assigned by EMREG (e.g., "substantial", "high").	Yes	substantial
email	String	Contact email for administrative purposes.	Yes	emrex@uva.nl
website	URL	Official website of the EMC.	No	https://www.uva.nl
publicKey	String (PEM)	Public key for JWT signature validation.	Yes	-----BEGIN PUBLIC KEY----- ...

Example EMC JSON

```
{
  "clientId": "emc-nl-uva",
  "redirectUri": "https://emc.uva.nl/callback",
  "name": {
    "en": "University of Amsterdam",
    "nl": "Universiteit van Amsterdam"
  },
  "logo": "https://emc.uva.nl/logo.png",
  "trustLevel": "substantial",
  "email": "emrex@uva.nl",
  "website": "https://www.uva.nl",
  "publicKey": "-----BEGIN PUBLIC KEY-----..."
}
```

–

3.3. Supporting Data Models

DataFormat

Attribute	Type	Description	Required	Example
name	String	Name of the data format.	Yes	elmo
version	String	Version of the format.	No	2.1
schemaUrl	URL	URL to the schema definition.	No	https://emrex.eu/elmo.xsd
allowedQueryParams	String	Query Parameters to be used to 'query' the resource	No	withAttachments

EvidenceType

Attribute	Type	Description	Required	Example
name	String	Name of the evidence type.	Yes	diploma

Institution

Attribute	Type	Description	Required	Example
name	String	Name of the institution.	Yes	University of Amsterdam

Country

Attribute	Type	Description	Required	Example
isoCode	String	ISO 3166-1 alpha-2 country code.	Yes	NL
singleFetch	Boolean	Whether single-fetch is supported.	No	false

Registration Processes

EMREG supports **two distinct flows** for managing EMP/EMC metadata:

1. **Initial Registration Flow:** For new EMPs/EMCs that are not yet in the registry.
2. **Update Flow:** For existing EMPs/EMCs that need to modify their metadata.

Both flows use the **same data models** but differ in **approval requirements** and the use of `activeByDateTimeUtc`.

4.1. Initial Registration Flow

Steps

1. **Account Creation:**
 - EMP/EMC administrator creates an account on the **EMREG Self-Service Portal**.
 - Account is verified via email.
2. **Form Submission:**
 - Administrator fills out the registration form with all required fields.
 - For EMPs: Includes `emreg2Props`, `institutions`, and `country`.
 - For EMCs: Includes `clientId`, `redirectUri`, and `publicKey`.
 - `activeByDateTimeUtc` is **not required** for initial registrations (activation is immediate upon approval).
3. **Review and Approval:**
 - **EMP Registrations:**
 - Sent to the **EMREX Board** for approval.
 - Board verifies legitimacy and compliance with EMREX policies.
 - **EMC Registrations:**
 - Reviewed by an **EMREG Administrator**.
 - Administrator verifies technical details (e.g., `redirectUri`, public key format).
 - Approval criteria:
 - For EMPs: Must be a recognized institution (e.g., national agency or university).
 - For EMCs: Must be a legitimate educational institution.
 - The reviewer may contact the applicant for additional verification (e.g., phone call, documentation).
4. **Approval/Rejection:**
 - If approved, the EMP/EMC is **immediately activated** in the registry.
 - If rejected, the applicant receives a notification with the reason.

4.2. Update Flow

Steps

1. **Login:**
 - EMP/EMC administrator logs in to the **EMREG Self-Service Portal** using existing credentials.
2. **Update Submission:**
 - Administrator submits an update to the existing metadata.
 - `activeByDateTimeUtc` is **required** for updates:
 - Specifies when the changes should take effect.
 - Allows institutions to schedule updates (e.g., for maintenance windows).
 - Updates can include:
 - New endpoints (e.g., `authorizationUrl`, `tokenUrl`).

- Updated public keys.
- Changes to supported data formats or trust levels.

3. Review and Approval:

- **EMP Updates:**
 - Sent to the **EMREX Board** for approval if the update includes significant changes (e.g., new institutions, trust level changes).
 - Minor updates (e.g., endpoint URLs) may be auto-approved.
- **EMC Updates:**
 - Reviewed by an **EMREG Administrator** if the update includes critical changes (e.g., `redirectUri`, `publicKey`).
 - Minor updates may be auto-approved.
- Approval criteria:
 - Changes must comply with EMREX policies.
 - Updates must not break existing integrations.

4. Approval/Rejection:

- If approved, the update is **marked as pending** in the registry.
- If rejected, the applicant receives a notification with the reason.

5. Scheduled Activation:

- The update becomes **active at the specified `activeByDateTimeUtc`**.
- Until then:
 - The **old metadata** remains in effect.
 - The update is **not visible** in the public API.
- EMREG runs a **scheduled job** to activate pending updates at the specified time.

4.3. Approval Workflow Comparison

Aspect	Initial Registration	Update Flow
Purpose	Add a new EMP/EMC to the registry.	Modify existing EMP/EMC metadata.
<code>activeByDateTimeUtc</code>	Not required (activation is immediate upon approval).	Required (changes take effect at the specified time).
Approval Process	Always requires manual review (EMREX Board for EMPs, EMREG Admin for EMCs).	Manual review only for significant changes; minor updates may be auto-approved.
Activation	Immediate upon approval.	Scheduled for the specified <code>activeByDateTimeUtc</code> .
Impact of Rejection	Applicant must resubmit with corrections.	Changes are not applied; old metadata remains in effect.
Use Case	Onboarding new institutions.	Maintaining existing registrations (e.g., endpoint changes, key rotations, new data formats).

4.4. Activation Schedule

- **Initial Registrations:**
 - Activated **immediately upon approval**.
- **Updates:**
 - Require an `activeByDateTimeUtc` field to specify when changes take effect.
 - Allows institutions to:
 - Schedule updates during maintenance windows.
 - Coordinate changes with other systems.
 - Test updates in a staging environment before production activation.
 - EMREG runs a **daily scheduled job** to activate pending updates at their specified times.

API Endpoints

EMREG provides a **REST API** for programmatic access to registry data. The following endpoints are available:

Endpoint	Method	Description	Authentication	Example Request
<code>/emp-list</code>	GET	Returns a list of active EMPs.	None	GET <code>/emp-list</code>
<code>/emp/{empId}</code>	GET	Returns metadata for a specific EMP.	None	GET <code>/emp/duo-n1</code>
<code>/clients/{clientId}</code>	GET	Returns metadata for a specific EMC.	Bearer Token (EMP)	GET <code>/clients/emc-n1-uva</code>

Endpoint	Method	Description	Authentication	Example Request
/health	GET	Health check endpoint.	None	GET /health

Security Considerations

- HTTPS Enforcement:**
 - All communications with EMREG **must** use HTTPS (RFC 2818).
 - Endpoints using HTTP are rejected.
- Public Key Validation :**
 - EMREG validates the format of public keys during registration and updates.
 - Keys must be in **PEM format** and use **RS256** (RFC 7518).
- Trust Levels:**
 - EMPs can specify a `requiredTrustLevel` for EMCs.
 - EMCs are assigned a `trustLevel` by EMREG.
 - Token requests are rejected if the EMC's `trustLevel` is below the EMP's `requiredTrustLevel`.
- Rate Limiting:**
 - EMREG API endpoints enforce rate limiting to prevent abuse.
- Data Validation :**
 - All fields are validated against the data models.
 - Invalid submissions are rejected with detailed error messages.
- Audit Logging:**
 - All registration changes and API accesses are logged for auditing.
- Scheduled Activations:**
 - Updates are only activated at the specified `activeByDateTimeUtc`, reducing the risk of unintended disruptions.

Governance

Roles and Responsibilities

Role	Responsibilities
EMREX Board	Approves/rejects new EMP registrations and significant EMP updates.
EMREG Administrator	Reviews and approves new EMC registrations and significant EMC updates; manages the registry.
EMP Administrator	Submits and maintains EMP metadata; requests updates.
EMC Administrator	Submits and maintains EMC metadata; requests updates.

Compliance

- GDPR:** EMREG complies with GDPR for data processing and storage.
- eIDAS:** Supports eIDAS trust levels and electronic identification.
- OAuth2:** Follows RFC 6749, RFC 7523, and RFC 7636 for security.

Dispute Resolution

- Disputes regarding registrations are escalated to the **EMREX Board**.
- Technical issues are handled by the **EMREG support team** (`support@emreg.eu`).