Republic of Estonia
Ministry of Economic Affairs and Communications

Nordic Council of Ministers

USER GUIDE

Description of eCMR Index Registry and Tested Use Case Example by Fitek Team

Development funded by
DIGINNO-Proto

Manual Version 1.07

Contents

eCMR index registry and technology

Hyperledger Fabric platform used as the backbone of the eCMR indexing system. It is an enterprise grade permissioned distributed ledger platform originally developed by IBM. It provides modular architecture with fine-grained customization possibility across different network nodes and ensures data exchange security, non-repudiation, and trust.

This platform perfect fit for eCMR indexing solution as its enterprise-grade functionality allows to implement various data exchange scenarios with production-grade security and transparency measures.

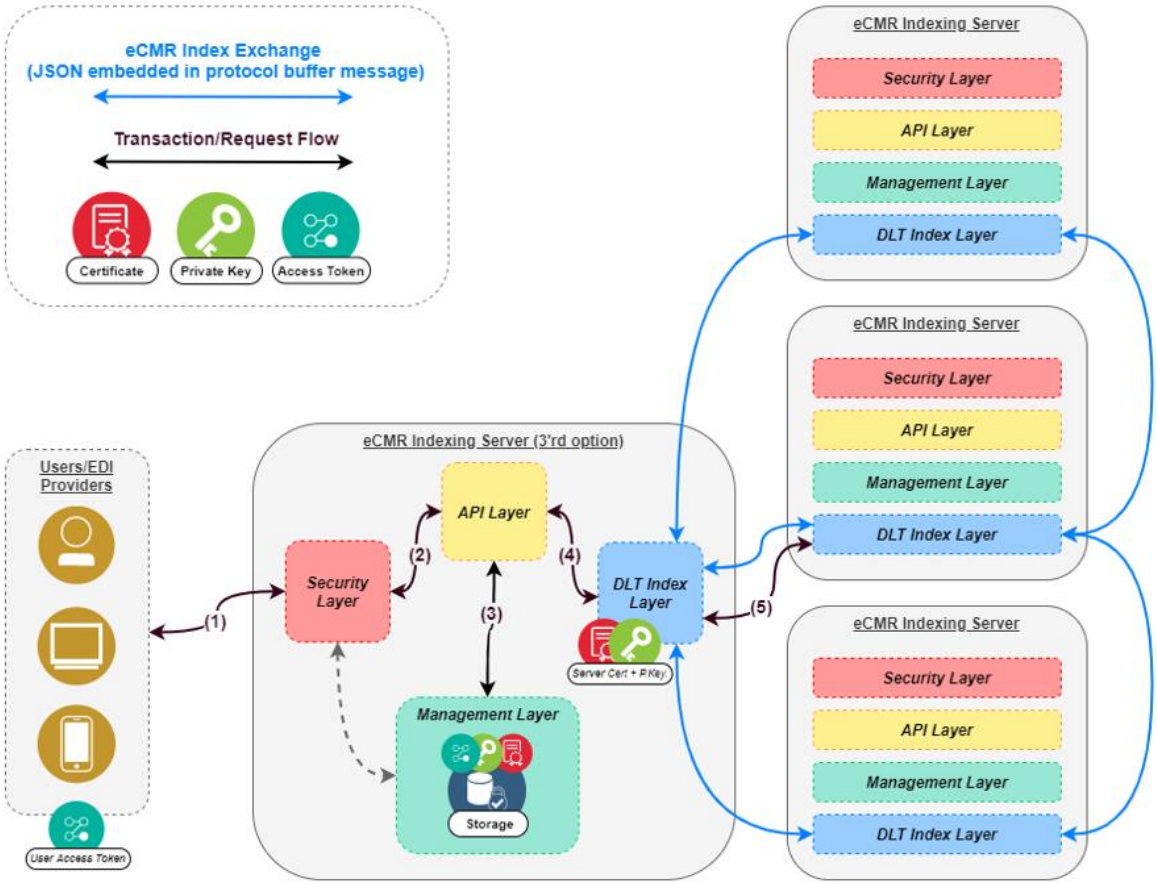


Figure 1. eCMR indexing solution API request scenario illustration.

The figure above illustrates the main workflow scenario of the system. As shown in illustration, the users will be provided access tokens. With an access token, the user will be able to send request to eCMR indexing server API that will allow to add or query eCMR indexing data if

the access token and the linked certificate in the server meets the defined rules and requirements (as an example, specific users might be allowed only to query the eCMR indexing data based on his certificate attributes). The combination of security and management layers will provide user management functionality and permissioned system access functionality.

Users exchange exchange data with eCMR index over the API on base JSON (JavaScript Object Notation).

Smart-contract input/output structure

The input and output data format for smart-contract/-s use JSON format as it will be aligned with internal data exchange meta-structure of the system, have more structured and readable input format, is supported by document-based databases and has possibility to validate input based on already implemented standards and solutions.

The smart-contract/-s used in developed in a way, that they expect to receive one argument as JSON parameter, since it allow easier and more flexing input data validation. Input data of smart-contract is going to be validated using either JSON Schema (which is preferred) or internal programming language validation functionality in specific cases.

Internal indexing server search and storage engine

To allow flexible and fast access to the exchanged data between different countries, each country must use *CouchDB* as search and storage engine. It allow in search exchange data by specific data fields while also ensuring search performance and flexibility.

API

eCMR indexing server use API that abstracts the complexity of communication with data exchange layer. It based on current RESTful API standards and allow user add or query eCMR indexing data.

The following aspects define API implementation:

- **Architectural style:** REST

- **Request/data format:** JSON
- **Security, authentication and authorization:** OAuth2.0 / OpenID
- **Documentation:** OpenAPI 3.0 (OAS 3.0)

Credential storage

To streamline adoption of this solution, provide easy and flexible access to the system, ensure security and transparency of the system, from the governmental perspective - the private keys and public keys of the users (clients, EDI Providers, customs, etc…) will be stored in the governmental eCMR indexing server. This allow to have fine-grained and strict control over the users credentials and also streamline the adoption of the system as it will allow users to connect to a system via simplified API while also preserving the security aspects of the system. This scenario is illustrated in the section above (Figure 1).

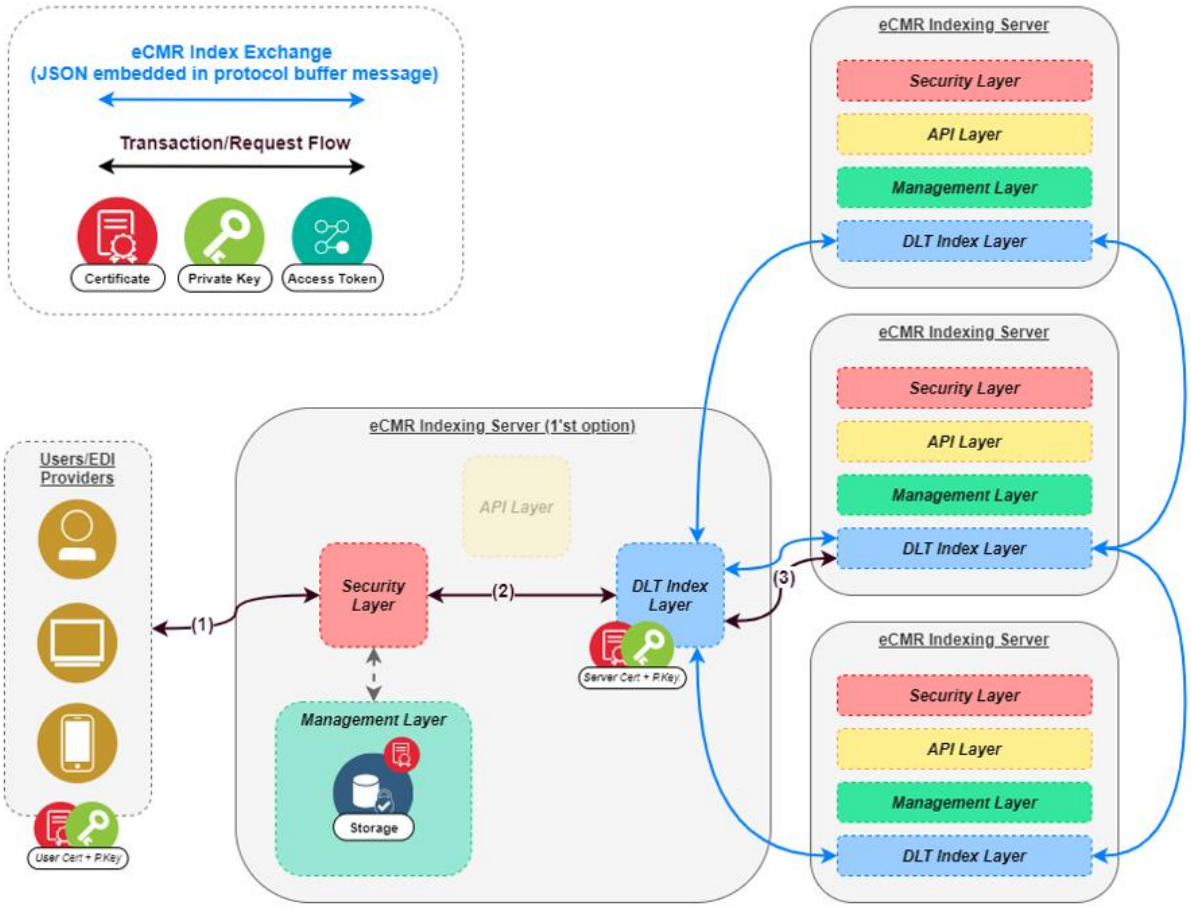


Figure 2. eCMR direct communication with the DLT layer scenario.

NOTE: *If connecting client will not be satisfied with the provided security of API, he will be able to send registration request (CSR) to the eCMR indexing server and securely obtain his certificate while not exposing his private keys to governmental indexing server. This allow to have maximum security both from the user perspective and government perspective, however it will reduce the simplicity the system as the API will not be used in this scenario and user will have to directly communicate with the system using different protocol (scenario is illustrated in Figure 2).*

Optional direct communication with DLT indexing layer

If the user is not satisfied with the provided API solution, he will be able to obtain certificate without having to store his privately generated private key in governmental eCMR indexing server (as describe above in. *Credential storage*). It means that the user will bypass API layer and communicate directly with DLT indexing layer and as a result, he will have to use additional software development kit, which enables communication with the indexing layer directly

Semantic data model of the API

Description for Index server project (version 4.01)

ID	Level	Cardinality	Business Term	Description	Usage Note	TDED	Data type	BRS ID	Sample value
BG-1	1	1..1	documentIdentifier	The unique identifier of this exchanged document (OID)	participantIdentifier: documentIdentifier.value		Complex		
BG-1.2	2	1..1	participantIdentifier		• ISO/IEC 6523 • https://en.wikipedia.org/wiki/ISO/IEC_6523 • https://docs.peppol.eu/poacc/billing/3.0/codelist/ICD/ We propose to use similar rules that PEPPOL use for eInvoicing		Complex		
BT-1.2-1	3	1..1	scheme				A(4)		200
BT-1.2-2	3	1..1	value				A(9)		125677598
BG-1.3	2	1..1	scheme		Id:value		Complex		
BT-1.3-1	3	1..1	Id		Busdox		A()		
BT-1.3-2	3	1..1	value				A()		eCMR v.1.0
BT-1-1	2	1..1	value				A()		iso6523-actorid-upis:0200:125677598-7357bf9f95ez
BG-2	1	1..1	document				Complex		
BT-2-1	2	1..1	Country code	The location where this exchanged document has been issued. Identifier Country Code Definition: The unique identifier of a country for this logistics location.	Country code in ISO3166-1alpha-2 system (https://www.iban.com/country-codes)	UDT0000019 UN01004598	A(2)	R-6	LT
BG-2.1	2	1..N	transport		categoryCode: countryCode registrationNumber		Complex		
BT-2.1-1	3	1..1	Category code	The code specifying the type of logistics means of transport	Information about trucks and trailers (urn:un:unece:uncefact:codelist:standard:UNECE: TransportEquipmentCategoryCode:D19A TE=Trailer).	UN02000087 UN01003767	A(3)		vehicle
BT-2.1-2	3	1..1	Country code	The identifier of the country for this government registration.	https://en.wikipedia.org/wiki/International_vehicle_registration_code	UDT0000011 UN01011036	A(3)		EE
BT-2.1-3	3	1..1	Vehicle registration number	An identifier of this logistics means of transport, such as the		UDT0000011 UN01003822	An(9)		800ABC

				road transport registration plate number					
BT-2-2	2	1..1	uri	The unique Uniform Resource Identifier (URI) for this specified binary file.	Location of source document in the address space of certain EDI provider	UDT000002 UN01006015	A()	R-4	https://lt.ecmr4.eu/documents/iso6523-actorid-upis:0200:125677598-7357bf9f95e5
BT-2-3	2	1..1	Document fingerprint	The number of the checksum for this document.		UDT000002 UN00002653 UN00006066	A()	R-7	14b55a3a797d1ad449a74d0c6020f05caaf58b02627eb88f3a86bf9eac1ca1a8
BT-3	1	1..1	Creation DateTime				Timestamp	R-1	Filled in by the system
BT-4	1	0..1	Update DateTime				Timestamp	R-1	Filled in by the system
BT-5	1	0..1	Expire DateTime	Information of date and time when consignment was closed		UDT000008 UN01001275	Timestamp	R-1	

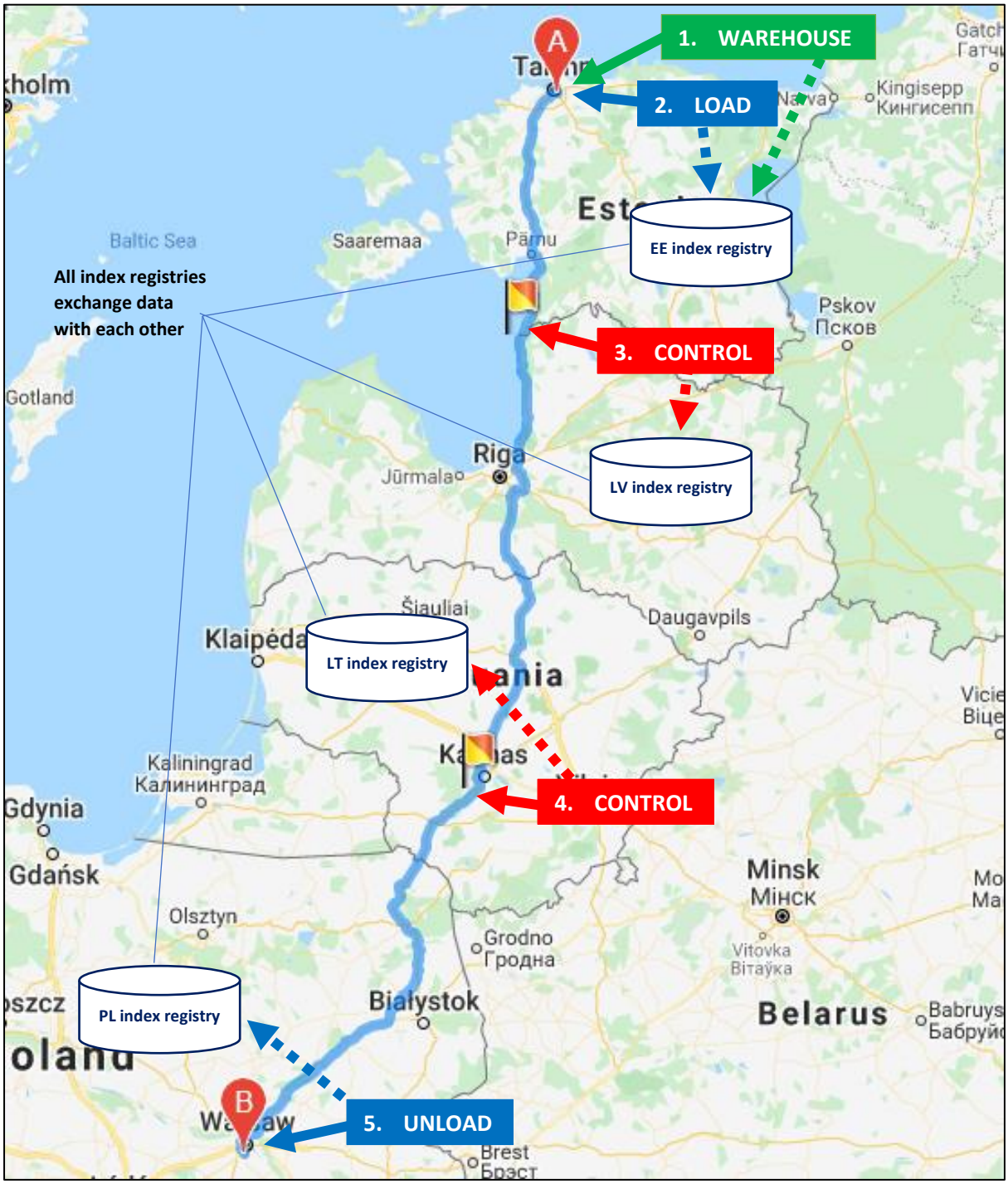
- Reference to UN/CEFACL CCL: https://www.unece.org/fileadmin/DAM/uncefact/CCL/CCL20A.zip
- Reference to UN/CEFACT eCMR:
  - https://www.unece.org/fileadmin/DAM/cefact/Standards/eCMR/05_eCMR_CCL_Structure_16FEB18.pdf
  - https://www.unece.org/fileadmin/DAM/cefact/Standards/eCMR/01_BRS_eCMR_v1.pdf

- UN/CEFACT multimodal transport: http://www.smdg.org/assets/assets/0-SMDG74/Pres/DAD3-SMDG-74-20190919-UNCEFACT-SP.pdf
- Message Standards in the EU: https://ipcsa.international/epsca-message-ref-guide

Business Requirements Specification

BRS ID	Description
R-1	expire date / time / time zone of the eCMR document (optional) a. expire date / time / time zone must validate from now to the future b. If expire date not defined, then document in open status in the eCMR index system
R-4	Truck registration number (required, if haulier)
R-5	Trailer/tank registration number (required, if haulier and goods loaded to trailer) Possible connect with goods just truck and also truck train, examples with truck & trailer(s): Example 1. https://upload.wikimedia.org/wikipedia/commons/2/29/Alle_Varianten_LZV.jpg?1587496423282 Example 2. http://www.transportoversize.eu/files/Main/docs/Dimensions%20of%20vehicle.bmp Registration number > isoCode - country code, example in Ukraine "UA" Registration number > regNumber - vehicle registrated plate number Look also https://en.wikipedia.org/wiki/Vehicle_registration_plates_of_Europe
R-6	eCMR index number (required) a. eCMR index structure described in section “STRUCTURE OF eCMR INDEX”
R-7	Hash of binary e-CMR document, fingerprint (required) Attribute "type" connected with "fingerprint" define what technology used when system create it (use SHA256).

Visual view of use case process



© Google MAP

If you use in test examples below, don't forget update inside source code vehicle plate numbers and unique number of eCMR index ID.

Case 1. Warehouse (Estonia)

Participant:	Warehouse
Vehicle plate numbers:	**(EE) 137ABC** [TRUCK], **(EE) QI99** [TRAILER]
Activities:	I as warehouse worker create eCMR document and add document link to national eCMR index registry
Process:	1. Warehouse add eCMR metadata and document location to national eCMR index registry.
User rights group:	Business
Pre requirements:	1. Created user account (link) 2. Account added to group “Business” (link) 3. User has downloaded it’s certificate for API call (link)

Process 1 example code (POST)

Role:	Warehouse
Business name:	Ospentos International OÜ
Register code:	11306955

```
{
    "participantIdentifier": {
        "scheme": "iso6523-actorid-upis",
        "value": "0191:11306955"
    },
    "documentIdentifier": {
        "scheme": {
            "id": "busdox-docid-qns",
            "value": "eCMR v.1.0"
        },
        "value": "iso6523-actorid-upis:0191:11306955-QXxUQqbe3jqc"
    },
    "processIdentifier": {
        "scheme": "cenbii-procid-ubl",
        "value": "urn:www.cenbii.eu:profile:ecmr:ver1.0"
    },
    "document": {
        "countryCode": "EE",
        "uri": "https://www.ospentos.ee/documents.php?id=QXxUQqbe3jqc",
        "fingerprint": "14b55a3a797d1ad449a74d0c6020f05caaf58b02627eb88f3a86bf9eac1ca1a8"
    }
}
```

Warehouse share with driver eCMR QR Code

Case 2. Load (Estonia)

Participant:	Truck driver
Vehicle plate numbers:	**(EE) 137ABC** [TRUCK], **(EE) QI99** [TRAILER]
Activities:	1. I as driver load goods to trailer and download eCMR documents by unique ID throw EDI provider system 2. I as driver accept eCMR documents. Also change eCMR document location and add truck and trailer plate numbers
Process:	1. I as EDI provider load eCMR document location from national eCMR index registry 2. I as EDI provider load eCMR document by document URI 3. I as EDI provider add eCMR updated metadata and document location to national eCMR index registry.
User rights group:	Business
Pre requirements:	1. Created user account (link) 2. Account added to group “Business” (link) 3. User has downloaded it’s certificate for API call (link) 4. eCMR was created on previous step and documentIdentifier is known to user

Process 1 example code (GET)

```
{{API_URL}}/index/iso6523-actorid-upis:0191:14258742-QXxUQqbe3jqc
```

Index registry answer


```
[{
    "participantIdentifier": {
        "scheme": "iso6523-actorid-upis",
        "value": "0191:11306955"
    },
    "documentIdentifier": {
        "scheme": {
            "id": "busdox-docid-qns",
            "value": "eCMR v.1.0"
        },
        "value": "iso6523-actorid-upis:0191:11306955-QXxUQqbe3jqc"
    },
    "processIdentifier": {
        "scheme": "cenbii-procid-ubl",
        "value": "urn:www.cenbii.eu:profile:ecmr:ver1.0"
    },
    "document": {
        "countryCode": "EE",
        "createdDateTime": "2019-10-12T14:20:50.52+07:00",
```

```
        "updatedDateTime": "2019-10-12T14:20:50.52+07:00",
        "uri": "https://www.ospentos.ee/documents.php?id=QXxUQqbe3jqc",
        "fingerprint": "14b55a3a797d1ad449a74d0c6020f05caaf58b02627eb88f3a86bf9eac1ca1a8"
    }
}]
```

Role:	EDI provider
Business name:	Intepia OÜ
Register code:	14258742

Process 3 example code (PUT)

Changes marked with yellow colour!

```
{
    "participantIdentifier": {
        "scheme": "iso6523-actorid-upis",
        "value": "0191:14258742"
    },
    "documentIdentifier": {
        "scheme": {
            "id": "busdox-docid-qns",
            "value": "eCMR v.1.0"
        },
        "value": "iso6523-actorid-upis:0191:11306955-QXxUQqbe3jqc"
    },
    "processIdentifier": {
        "scheme": "cenbii-procid-ubl",
        "value": "urn:www.cenbii.eu:profile:ecmr:ver1.0"
    },
    "document": {
        "countryCode": "EE",
        "transport": [
            {
                "categoryCode": "3 39",
                "countryCode": "EE",
                "registrationNumber": "137ABC"
            },
            {
                "categoryCode": "3 TE",
                "countryCode": "EE",
                "registrationNumber": "QI99"
            }
        ],
        "uri": "https://www.intepia.com/documents/?id=QXxUQqbe3jqc",
        "fingerprint": "85475814b55a3a797d1ad449a74d0c6020f05caaf58b02627eb88f3a86bf9eac"
    }
}
```

Case 3. Control (Latvia)

Participant:	Police
Vehicle plate numbers:	**(EE) 137ABC** [TRUCK], **(EE) QI99** [TRAILER]
Activities:	1. Police patrol car video camera scan from distance the plate number of the TRUCK moving on the road and send scanned number info throw internet to national central data system. 2. Police central data system make request to national eCMR index server and get back all eCMR documents connected with TRUCK number. 3. Police system download eCMR document by document URI 4. Police information system analyze data return summed results to computer display on the patrol car.
Process description:	1. Control institution central data system make info request from eCMR index server by TRUCK registration number. 2. eCMR index server return package of eCMR documents indexes connected with truck number.
User rights group:	Government
Pre requirements:	1. Created user account (link) 2. Account added to group “Government” (link) 3. User has downloaded it’s certificate for API call (link) 4. Truck is on Latvian territory 5. At least one plate number is known to user

Process 1 example code (GET)

```
{{API_URL}}/index?transport_reg_number=137ABC&transport_country_code=EE
```

Index registry answer


```
[{
    {
        "participantIdentifier": {
            "scheme": "iso6523-actorid-upis",
            "value": "0191:14258742"
        },
        "documentIdentifier": {
            "scheme": {
                "id": "busdox-docid-qns",
                "value": "eCMR v.1.0"
            },
          "value": "iso6523-actorid-upis:0191:11306955-QXxUQqbe3jqc"
        },
        "processIdentifier": {
```

```
            "scheme": "cenbii-procid-ubl",
            "value": "urn:www.cenbii.eu:profile:ecmr:ver1.0"
        },
        "document": {
            "countryCode": "EE",
            "transport": [
                {
                    "categoryCode": "3 39",
                    "countryCode": "EE",
                    "registrationNumber": "137ABC"
                },
                {
                    "categoryCode": "3 TE",
                    "countryCode": "EE",
                    "registrationNumber": "QI99"
                }
            ],
            "createdDateTime": "2019-10-12T14:20:50.52+07:00",
            "updatedDateTime": "2019-10-14T14:20:50.52+07:00",
            "uri": "https://www.intepia.com/documents/?id=QXxUQqbe3jqc",
            "fingerprint": "85475814b55a3a797d1ad449a74d0c6020f05caaf58b02627eb88f3a86bf9eac"
        }
    }
}]
```

Case 4. Control (Lithuania)

Participant:	Police
Vehicle plate numbers:	**(EE) 137ABC** [TRUCK], **(EE) QI99** [TRAILER]
Activities:	1. Police patrol car video camera scan from distance the plate number of the TRAILER moving on the road and send scanned number info throw internet to national central data system. 2. Police central data system make request to national eCMR index server and get back all eCMR documents connected with TRAILER number. 3. Police system download eCMR document by document URI 4. Police information system analyze data return summed results to computer display on the patrol car.
Process description:	1. Control institution central data system make info request from eCMR index server by TRAILER plate number. 2. eCMR index server return package of eCMR documents indexes connected with TRAILER number.
User rights group:	Government
Pre requirements:	1. Created user account (link) 2. Account added to group “Government” (link) 3. User has downloaded it’s certificate for API call (link) 4. Truck is on the Lithuanian territory 5. At least one plate number is known to user

Process 1 example code (GET)

```
{{API_URL}}/index?transport_reg_number=QI99&transport_country_code=EE
```

Index registry answer

```
Same as Case 4 answer
```

Case 5. Unload (Poland)

Participant:	Truck driver
Vehicle plate numbers:	**(EE) 137ABC** [TRUCK], **(EE) A-123** [TRAILER]
Activities:	I as driver unload goods from trailer and confirm in mobile device that goods are unloaded
Process:	1. I as EDI provider confirm that eCMR document expired (not connected any more with truck and trailer).
User rights group:	Business
Pre requirements:	1. Created user account (link) 2. Account added to group “Business” (link) 3. User has downloaded it’s certificate for API call (link) 4. DocumentIdentifier is known to user 5. Truck is on the Poland territory

Process 1 example code (PUT)

```
{
    "participantIdentifier": {
        "scheme": "iso6523-actorid-upis",
        "value": "0191:125677598"
    },
    "documentIdentifier": {
        "scheme": {
            "id": "busdox-docid-qns",
            "value": "eCMR v.1.0"
        },
        "value": "iso6523-actorid-upis:0191:11306955-QXxUQqbe3jqc"
    },
    "processIdentifier": {
        "scheme": "cenbii-procid-ubl",
        "value": "urn:www.cenbii.eu:profile:ecmr:ver1.0"
    },
    "document": {
        "countryCode": "EE",
        "transport": [
            {
                "categoryCode": "3 39",
                "countryCode": "EE",
                "registrationNumber": "137ABC"
            },
            {
                "categoryCode": "3 TE",
                "countryCode": "EE",
```

```
                "registrationNumber": "QI99"
            }
        ],
        "expiredDateTime": "2019-10-12T14:20:50.52+07:00",
        "uri": "https://www.intepia.com/documents/?id=QXxUQqbe3jqc",
        "fingerprint": "14b55a3a797d1ad449a74d0c6020f05caaf58b02627eb88f3a86bf9eac1ca1a8"
    }
}
```

Questions and answers

Question 1: If the EU generates ~ 470 million CMRs annually, will the eCMR registrers on base DLT be able to handle such a large amount of transactions?

Answer: Such a system can make up to 20 000 transactions per second per one server.

Simplified calculation:

A workday has ~ 8 hours and per year ~ 253 workdays.

8 hours * 3600secounds (seconds in hour) * 235 workdays * 20 000 transactions = ~ 135 billion transactions per year per one server.

As we see, the software is more than enough to handle a big amount of transactions.

Question 2: Can I control the history of changes made with the eCMR index?

Answer: Yes, it is possible. In this case, the GET query looks like **/index/{id}?history=true**

Question 3: Why does a company have to add "expiredDateTime" to the eCMR index registry after unloading the goods?

Answer: expiredDateTime very important add to eCMR index registry when goods are unloaded. If police control the truck after unload, then eCMR index registry does not show any more current index connected with truck or trailer plate number.

Question 4: We as government institution cannot find several vehicles in the system by vehicle number. Why?

Answer: If a truck with a trailer loads the cargo, then warehouse or the truck company info system add index to the eCMR index registry.

If the truck unloads the goods, then info system add update and mark that current index expired (expiredDateTime = now).

If police control make requests by truck or by trailer number between load and unload, then the result will be the previously mentioned index.

If police control after unload by truck or by trailer number, then this cargo not connected any more with the vehicle and registry gives you an empty result (it means that trailer empty).

After unloading possible find this cargo in eCMR index registry only by index ID.

Question 5: How I can find to what category belong my truck or trailer?

Answer: Very easy to find right category to your truck or trailer. Definition of category base on the UN/CEFACT standard.

Syntax: "CategoryCode": "{{CATEGORY1}} {{CATEGORY2}}"

Explanation:

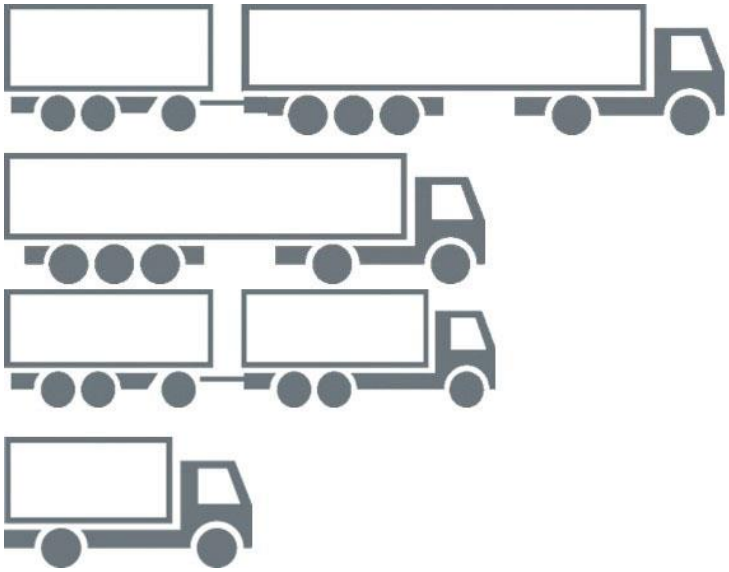
- {{CATEGORY 1}} Road transport (3)
- {{CATEGORY 2}} Category of transport

Example: 3 31

#	Type of transport equipment	{{CATEGORY1}}	{{CATEGORY2}}
1	Truck	3	31
2	Truck with loading place	3	34
3	Truck with loading place (refrigerator)	3	39
4	Truck with loading place (isoterm)	3	39
5	Truck with loading place (cistern)	3	32
6	Truck with loading place (platform)	3	36
7	Truck with loading place (tipper)	3	35
8	semi trailer	3	SM
9	cistern	3	TA
10	tipper	3	TE
11	platform	3	TE
12	isoterm	3	TE
13	refrigerator	3	TE
14	grain	3	TE
15	Boxcar	3	BX
16	Container	3	CN

Question 6: How many trailers can I add to the index?

Answer: The number of trailers is not limited!



Question 7: If our company use road train how define it in eCMR index registry?

Answer: eCMR index registry API allow very easy define more than one trailer. Code example with 1 truck with loading place and number plate 137ABC pluss 2 trailers with number plates QI99 and QR88:

```
...
        "transport": [
            {
                "categoryCode": "3 39",
                "countryCode": "EE",
                "registrationNumber": "137ABC"
            },
            {
                "categoryCode": "3 TE",
                "countryCode": "EE",
                "registrationNumber": "QI99"
            },
            {
                "categoryCode": "3 TE",
                "countryCode": "EE",
                "registrationNumber": "QR88"
            }
        ],
...
```

Question 8: From where I find my eCMR index ID?

Answer: Look added example below.

Question 8: Based on which information I must put together the index ID?

Answer: Look added example and explanation below.

ISO 6523 ICD list: https://docs.peppol.eu/poacc/billing/3.0/codelist/ICD/

Question 9: If I as a truck company download from warehouse system eCMR document and start share eCMR document in my server (change document URI and fingerprint), then what more I must change in the index registry?

Answer: You MUST keep only index ID value, but change all other data fields by your company.

Question 10: On base question 6. If I as truck company change during the eCMR document lifecycle index ID, then what happen?

Answer: You can change index ID, but it's mean that you create new lifecycle of changes. In this case very important that you close eCMR document previous lifecycle with field "expiredDateTime".

Question 11: Is it possible to see who viewed or changed the eCMR index?

Answer: Yes, it's possible. Please check under the technical documentation section "Get index access log".

Question 12: Is it possible add with index CMR document but in PDF format and how in this way looks like PUT or POST source code?

Answer: Changes in source code in POST or PUT if eCMR in XML format (market with yellow):

```
{
   . . .
    "documentIdentifier": {
        "scheme": {
            "id": "busdox-docid-qns",
            "value": "ecmr v.1.0"
        },
        "value": "iso6523-actorid-upis:0200:125677598-PDJRawMov7Ld"
    },
    "processIdentifier": {
        "scheme": "cenbii-procid-ubl",
        "value": "urn:www.cenbii.eu:profile:ecmr:ver1.0"
    },
   . . .
}
```

Changes in source code in POST or PUT if CMR in PDF format:

```
{
   . . .
    "documentIdentifier": {
        "scheme": {
            "id": "busdox-docid-qns",
            "value": "pdf v.1.0"
        },
        "value": "iso6523-actorid-upis:0200:125677598-PDJRawMov7Ld"
    },
    "processIdentifier": {
        "scheme": "cenbii-procid-ubl",
        "value": "urn:www.cenbii.eu:profile:pdf:ver1.0"
    },
       . . .
}
```