

# AGENDA

1. Process overview
2. High-level architecture
3. Bledger Network Token
- 4. Blockchain benefits and transparency**
5. Verify data on-chain
6. Bledger screenshots



# BLOCKCHAIN BENEFITS AND TRANSPARENCY

## Benefits of using Blockchain



### Easy Integration

- Polygon provides developer-friendly APIs and SDKs, making it simple to integrate blockchain functionality into existing systems
- Compatibility with Ethereum tools and smart contracts accelerates development and reduces setup time



### Non-Repudiation

- Every transaction is cryptographically signed and permanently stored on-chain
- This guarantees that recorded actions cannot be denied or altered later, ensuring data integrity and traceability



### Security

- Polygon inherits Ethereum's strong security model while offering optional scaling solutions
- The decentralized network architecture minimizes single points of failure and protects against tampering or unauthorized access



### Transparency

- All transactions and smart contract interactions are publicly verifiable
- This builds trust among stakeholders and simplifies auditing, compliance, and accountability



### Scalability and cost efficiency

- Polygon's Layer-2 approach offers faster transactions and lower gas fees without sacrificing decentralization
- Ideal for applications that need both speed and trust



# VERIFY DATA ON-CHAIN

---

Step by step guide for on  
chain data verification



# VERIFY DATA ON-CHAIN

PolygonScan

The platform enables the **verification of transactions** executed on the blockchain through the block explorer.

The following sections outline the various types of operations that can be verified and detail the procedures required to perform on-chain verification.


The screenshot displays the PolygonScan interface for the BLedger Network (BLN). The top navigation bar includes 'Home', 'Blockchain', 'Tokens', 'NFTs', and 'More'. The main content area is divided into three columns: 'Overview' (TOKENID: 7, TOTAL TRANSFERS: -), 'Market' (ONCHAIN MARKET CAP: -, CIRCULATING SUPPLY MARKET CAP: -), and 'Other Info' (TOKEN CONTRACT: 0x5480c3310a23993978fee5b40fa029c63f11a88f). Below these columns, there are tabs for 'Transfers' and 'Contract'. The 'Contract' tab is active, showing options for 'Code', 'Read Contract', and 'Write Contract'. A 'Connect to Web3' button is visible, along with a 'Read Contract Information' section. The 'Read Contract Information' section shows a list of contract functions, including '1. ADMIN\_ROLE (0x75b238fc)'. The interface also includes a search bar for 'TokenID:7' and a 'Connect to Web3' button.





# VERIFY DATA ON-CHAIN – READING CONTRACT (1/5)



Reading contract

The following are the steps to perform to verify the contract.

**Network B** 

+ Add auditor Diagram  Edit 

Clients	Level	Scope 1 e 2
No elements in table		

Suppliers	Level	q / t	Scope 1 e 2
 Company 3	2	q: - t: -	-----
 Company 2	2	q: - t: -	-----

- **Network Details**

On the *Network Detail* page, it's possible to click the Polygon icon next to the network name to see the list of the network's transactions



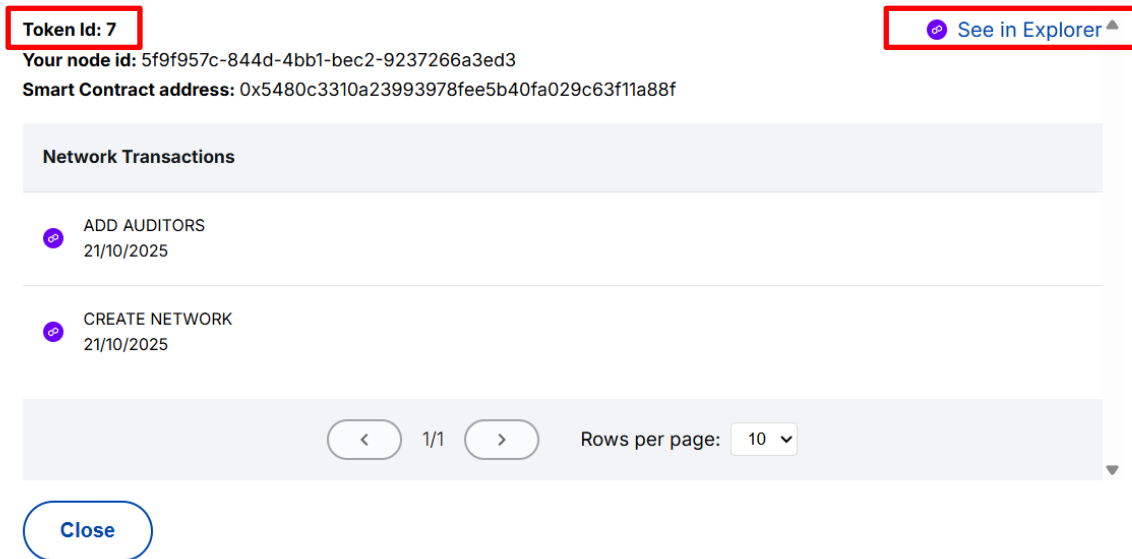
# VERIFY DATA ON-CHAIN – READING CONTRACT (2/5)

Reading contract

- **View network transactions**

A modal window appears displaying the list of network transactions. At the top, the following details are shown:

- the on-chain **token ID**
- the **“See in Explorer”** link



The screenshot shows a modal window with a close button (X) in the top right corner. At the top left, the text "Token Id: 7" is highlighted with a red box. To the right, the text "See in Explorer" with a purple icon and an upward arrow is also highlighted with a red box. Below this, the following information is displayed:

- Your node id: 5f9f957c-844d-4bb1-bec2-9237266a3ed3
- Smart Contract address: 0x5480c3310a23993978fee5b40fa029c63f11a88f

The main content area is titled "Network Transactions" and contains a list of two transactions:

Transaction	Date
ADD AUDITORS	21/10/2025
CREATE NETWORK	21/10/2025

At the bottom of the list, there are navigation controls: a left arrow, "1/1", a right arrow, and a "Rows per page: 10" dropdown menu. A "Close" button is located at the bottom left of the modal.



# VERIFY DATA ON-CHAIN – READING CONTRACT (3/5)

Reading contract

- **Explorer**

By clicking **“See in Explorer”**, the blockchain explorer opens directly in the **Read Contract** section.

**polygonscan** Home Blockchain Tokens NFTs More

BLedger Network (BLN) </> API

**Overview**

TOKENID  
7

TOTAL TRANSFERS  
-

**Market**

ONCHAIN MARKET CAP  
-

CIRCULATING SUPPLY MARKET CAP  
-

**Other Info**

TOKEN CONTRACT  
0x5480c3310a23993978fee5b40fa029c63f11a88f

Transfers **Contract** TokenID:7

Code **Read Contract** Write Contract

Connect to Web3

Read Contract Information [Expand all] [Reset]

1. ADMIN_ROLE (0x75b238fc)	🔗	🔗	→
2. AUDITOR_ROLE (0x6e1d616e)	🔗	🔗	→
3. COMPANY_ROLE (0x1c9ee11a)	🔗	🔗	→
4. authority (0xb7e214f)	🔗	🔗	→
5. balanceOf (0x70a08231)	🔗	🔗	→



# VERIFY DATA ON-CHAIN – READING CONTRACT (4/5)

## Reading contract

- **Browse available functions**

In **Read Contract** section, it's possible to find all readable functions of the contract

- For example, to fetch the current metadata JSON:

- ✓ Open the **tokenURI** function
- ✓ Enter the current **tokenId** in the input field
- ✓ Click **Query**
- ✓ The call returns the URL to the **metadata.json** for that token

Transfers Contract TokenID: 7

Code Read Contract Write Contract

Connect to Web3

Read Contract Information [Expand all] [Reset]

1. ADMIN_ROLE (0x75b238c)	⊖ ⊞ →
2. AUDITOR_ROLE (0x6e1d516e)	⊖ ⊞ →
3. COMPANY_ROLE (0x1c9ee11a)	⊖ ⊞ →
15. tokenContentHash (0x28dc758)	⊖ ⊞ →
16. tokenURI (0xc87b56dd)	⊖ ⊞ →

The **metadata.json** contains the network's hierarchical organization and the related documents. The link to the metadata.json corresponds to the tokenURI stored within the token.

16. tokenURI (0xc87b56dd)

tokenId (uint256)

7

Query

↳ string

[ tokenURI(uint256) method Response ]

>> string : <https://bledger.blockchainreply.it/api/v1/public/network/metadata/796d8d8e75a01597868519d078e4b3a52a7ed73947fa83b0b2abbe49c091dc37>



# VERIFY DATA ON-CHAIN – READING CONTRACT (5/5)

Reading contract

- Browse available functions

It is also possible to view the deployed on-chain contract by navigating to the “Contract” → “Code” section.

The screenshot shows the Bledger Network (BLN) contract page. The top section displays contract details:

- Overview:** TOKENID: 7, TOTAL TRANSFERS: -
- Market:** ONCHAIN MARKET CAP: -, CIRCULATING SUPPLY MARKET CAP: -
- Other Info:** TOKEN CONTRACT: 0x5480c3310a23993978fee5b40fa029c63f11a88f

The middle section shows the contract source code, with a "Contract" tab selected. The code is Solidity and includes the following details:

- Contract Name:** BledgerNetwork
- Optimization Enabled:** Yes with 200 runs
- Compiler Version:** v0.8.29+commit.ab5807c
- Other Settings:** cancon EvmVersion, MIT license

The source code is displayed in a code editor with the following content:

```
4242 contract BledgerNetwork is ERC721, ERC721Pausable, AccessManaged {
4243     // Roles (must match IdentityManager)
4244     uint64 public constant ADMIN_ROLE = type(uint64).min;
4245     uint64 public constant COMPANY_ROLE = 1;
4246     uint64 public constant AUDITOR_ROLE = 2;
4247
4248     // Token state
4249     uint256 private _nextTokenId;
4250     mapping(uint256 => string) private _tokenUri;
4251     mapping(uint256 => bytes32) private _tokenContentHash;
4252     mapping(uint256 => mapping(address => bool)) private _auditorOf; // auditors per tokenId
4253
4254     // Events
4255     enum UpdateReason { Generic, Upload, Audit, RefusedInvitation }
4256
4257     event NetworkCreated(uint256 indexed tokenId, address indexed to, string uri, bytes32 contentHash);
4258     event NetworkUpdated(uint256 indexed tokenId, address indexed updater, string newUri, bytes32 contentHash, UpdateReason reason);
4259     event AuditorSet(uint256 indexed tokenId, address indexed auditor, bool active);
4260     event MetadataUpdate(uint256 _tokenId); // EIP-4906
4261
4262     // Errors
4263     error NonexistentToken();
4264     error EmptyBatch();
4265     error MismatchedArrayLengths();
4266 }
```



# VERIFY DATA ON-CHAIN – VERIFY METADATA

Verifying integrity of the metadata.json

The following are the steps to perform to verify the integrity of the file *metadata.json*

- Copy the link returned from **tokenUri** function tab (*from the previous step – verify contract*) and open it in a new browser tab. The result is the *metadata.json*
- Copy the JSON displayed in 'step a' into any Keccak256 hashing tool (e.g., [https://emn178.github.io/online-tools/keccak\\_256.html](https://emn178.github.io/online-tools/keccak_256.html))
- The resulting hash should match the final part of the link. For example: <https://bledger.blockchainreply.it/api/v1/public/network/metadata/796d8d8e75a01597868519d078e4b3a52a7ed73947fa83b0b2abbe49c091dc37>

```
Input

{"networkId": "dd3e907d-bfd9-4d36-84df-43daacda63f7", "networkCreated": "2025-10-21T14:12:00.954692Z", "networkUpdated": "2025-10-21T14:21:23.281446Z", "metadataCreated": "2025-10-21T14:21:23.293128623Z", "merkleHash": "284918d7be8f1e7ac1de32e7ea219a200f18f6ccebef795c14b326cb12d1553a", "membersCount": 5, "uploadedCount": 1, "auditedCount": 0, "docs": [{"id": "fd217c2b-9aa7-44fb-b389-31338d3c477d", "nodeId": "5f9f957c-844d-4bb1-bec2-bec2-9237266a3ed3", "type": "GLOBAL_SCOPE_FILE", "documentHash": "9a4683f671f98cd890a3ec60ad1415f50faa4766fe61a89930dbf1dff8f62248"}, {"id": "b211be40-c2dd-468d-a2f4-af78a62e9711", "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3", "type": "GRANULAR_SCOPE_1", "documentHash": "09cb1db4d69054d360b0ef9386053efff61aacefa37c3a67c4ada82ac9133a357"}, {"id": "9bc94dfa-9ab6-44ae-95fa-cf5f70deeb9e", "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3", "type": "GRANULAR_SCOPE_2", "documentHash": "13cb47db1de53a256df0889813e282c624835d92e6dd799984338f42089614ec"}]}
```

```
Output

796d8d8e75a01597868519d078e4b3a52a7ed73947fa83b0b2abbe49c091dc37
```

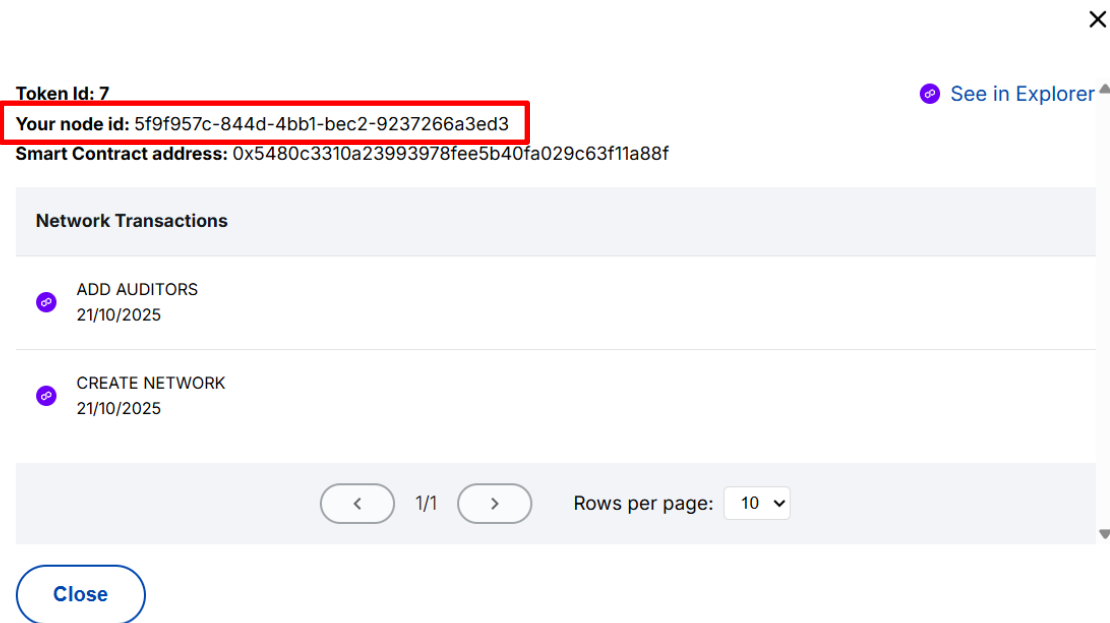


# VERIFY DATA ON-CHAIN – UPLOADED DOCS (1/3)

Verifying integrity of uploaded documents

The following are the steps to perform to verify the integrity of the uploaded documents.

- Copy the **node id** from the network transaction tab



The screenshot shows a modal window with a close button (X) in the top right corner. The main content area displays the following information:

- Token Id:** 7
- Your node id:** 5f9f957c-844d-4bb1-bec2-9237266a3ed3 (This text is highlighted with a red rectangular box in the original image)
- Smart Contract address:** 0x5480c3310a23993978fee5b40fa029c63f11a88f
- See in Explorer** (with a purple icon)

Below this information is a section titled **Network Transactions** containing a list of two transactions:

- ADD AUDITORS** (with a purple icon) on 21/10/2025
- CREATE NETWORK** (with a purple icon) on 21/10/2025

At the bottom of the transaction list, there is a pagination control showing "< 1/1 >" and a "Rows per page: 10" dropdown menu. A "Close" button is located at the bottom left of the modal.



## VERIFY DATA ON-CHAIN – UPLOADED DOCS (2/3)

Verifying integrity of uploaded documents

- Search the *node id* in the metadata json (e.g., 5f9f957c-844d-4bb1-bec2-9237266a3ed3)
- There should be three records under **docs** corresponding to the *node id* – one for each uploaded file (the Scope 1 & 2 document and the documents for the emissions details for Scope 1 and 2)

```
"docs": [  
  {  
    "id": "fd217c2b-9aa7-44fb-b389-31338d3c477d",  
    "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",  
    "type": "GLOBAL_SCOPE_FILE",  
    "documentHash": "9a4683f671f98cd890a3ec60ad1415f50faa4766fe61a89930dbf1dff8f62248"  
  },  
  {  
    "id": "b211be40-c2dd-468d-a2f4-af78a62e9711",  
    "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",  
    "type": "GRANULAR_SCOPE_1",  
    "documentHash": "09cb1db4d69054d360b0ef9386053eff61aacefa37c3a67c4ada82ac9133a357"  
  },  
  {  
    "id": "9bc94dfa-9ab6-44ae-95fa-cf5f70deeb9e",  
    "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",  
    "type": "GRANULAR_SCOPE_2",  
    "documentHash": "13cb47db1de53a256df0889813e282c624835d92e6dd799984338f42089614ec"  
  }  
]
```



# VERIFY DATA ON-CHAIN – UPLOADED DOCS (3/3)

Verifying integrity of uploaded documents

- Using a Keccak256 file hashing tool (e.g., [https://emn178.github.io/online-tools/keccak\\_256\\_checksum.html](https://emn178.github.io/online-tools/keccak_256_checksum.html)), it is possible to verify that the files uploaded to the platform match the hashes recorded in the **metadata.json**:
  - Upload the **CSV file** to the tool (e.g., *Scope 1*)
  - Verify that the resulting hash matches the one stored in **metadata.json**.

The screenshot displays a web-based Keccak256 checksum tool interface. It is divided into an 'Input' section at the top and an 'Output' section at the bottom. In the 'Input' section, a file icon is shown with the filename 'granular\_scope\_1\_data\_facts.csv'. To the right of the input area, a JSON object is displayed within a green-bordered box, representing the metadata for the uploaded file. The JSON object contains the following fields: 'id', 'nodeId', 'type', and 'documentHash'. The 'documentHash' value is highlighted in green. A red arrow points from the 'documentHash' value in the JSON output to the 'documentHash' value in the 'Output' section of the tool, which is also highlighted in green. The 'Output' section shows the resulting hash: '09cb1db4d69054d360b0ef9386053eff61aacefa37c3a67c4ada82ac9133a357'.

```
{
  "id": "b211be40-c2dd-468d-a2f4-af78a62e9711",
  "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
  "type": "GRANULAR_SCOPE_1",
  "documentHash": "09cb1db4d69054d360b0ef9386053eff61aacefa37c3a67c4ada82ac9133a357"
},
```

```
09cb1db4d69054d360b0ef9386053eff61aacefa37c3a67c4ada82ac9133a357
```

# VERIFY DATA ON-CHAIN – VERIFIED DOCS (1/2)

Verifying documents verified by the auditor

- When the auditor has verified the documents, it is possible to view the on-chain transaction (*NetworkUpdated - reason 2*)

The screenshot displays a blockchain explorer interface. At the top, there is a navigation bar with a back arrow and the text "All networks", and the word "AUDITOR" on the right. Below this is a large heading "Network B" with a small purple icon. A table below the heading has three columns: "Request from", "Level", and "Verification Request". The first row shows "Company 3" under "Request from", "2" under "Level", and "Verified 07/11/2025" under "Verification Request". A red arrow points from the "Verified" text in the table to the "Address" field in the "Transaction Receipt Event Logs" section below. The "Transaction Receipt Event Logs" section has tabs for "Overview", "Logs (5)", and "State". The "Logs (5)" tab is active. The log entry shows an "Address" field with a long hexadecimal string, a "Name" field with the text "NetworkUpdated (index\_topic\_1 uint256 tokenId, index\_topic\_2 address updater, string newUri, bytes32 contentHash, uint8 reason) View Source", and "Topics" with two entries: "0" and "2". The "Data" field contains a URL and a content hash, with "Dec" and "Hex" buttons to the right.

Request from	Level	Verification Request
Company 3	2	Verified 07/11/2025

Transaction Receipt Event Logs

Address: `0x5480c3310a23993978fee5b40fa029c63f11a88f`

Name: `NetworkUpdated (index_topic_1 uint256 tokenId, index_topic_2 address updater, string newUri, bytes32 contentHash, uint8 reason)` View Source

Topics: 0: `0x7459fa1b2e2ea892d4bd729f7809d501976cc87ca1fc1e3dcb07b7f26063d13c`

1: tokenId: Dec → 7

2: updater: Dec → `0xc82c465EaA04880f7c58d352beb63516212bd80c`

Data: `new... https://bledger.blockchainreply.it/api/v1/public/network/metadata/48c6f7c864ce9bfe7d7f253a130a15c8dcf07a8bdc688fc16a3c8cc7a35e08de` contentHash: `48C6F7C864CE9BFE7D7F253A130A15C8DCF07A8B8DC688FC16A3C8CC7A35E08DE` reason: 2



# VERIFY DATA ON-CHAIN – VERIFIED DOCS (2/2)

Verifying documents verified by the auditor

- By copying the link provided in the transaction data (*from the previous step*) and opening it in a new browser tab, the metadata will be displayed, showing whether the document has been verified (*audited=true*)

```
Data
new... https://bledger.blockchainreply.it/api/v1/public/network/metadata/48c6f7c864ce9bfe7d7f253a130a15c8dcf07a8
bdc688fc16a3c8cc7a35e08de
contentHash: 48C6F7C864CE9BFE7D7F253A130A15C8DCF07A8BDC688FC16A3C8CC7A35E08DE
reason: 2
```

← → ↻ 🔍 bledger.blockchainreply.it/api/v1/public/network/metadata/48c6f7c864ce9bfe7d7f253a130a15c8dcf07a8bdc688fc16a3c8cc7a35e08de

```
{
  "networkId": "dd3e907d-bfd9-4d36-84df-43daacda63f7",
  "networkCreated": "2025-11-20T14:12:00.954692Z",
  "networkUpdated": "2025-11-07T16:13:41.134991Z",
  "metadataCreated": "2025-11-07T16:13:41.143788141Z",
  "merkleHash": "284918d7be8f1e7ac1de32e7ea219a200f18f6ccebef795c14b326cb12d1553a",
  "membersCount": 5,
  "uploadedCount": 2,
  "auditedCount": 2,
  "docs": [
    [
      {
        "id": "fd217c2b-9aa7-44fb-b389-31338d3c477d",
        "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
        "type": "GLOBAL_SCOPE_FILE",
        "documentHash": "9a4683f671f98cd890a3ec60ad1415f50faa4766fe61a89930dbf1dff8f62248",
        "audited": true,
        "id": "b211be40-c2dd-468d-a2f4-af78a62e9711",
        "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
        "type": "GRANULAR_SCOPE_1",
        "documentHash": "09cb1db4d69054d360b0e9f386053eff61aacfa37c3a67c4ada82ac9133a357",
        "audited": true,
        "id": "9bc94dfa-9ab6-44ae-95fa-cf5f70deeb9e",
        "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
        "type": "GRANULAR_SCOPE_2",
        "documentHash": "13cb47db1de53a256df0889813e282c624835d92e6dd799984338f42089614ec",
        "audited": true,
        "id": "e3bfe628-0ae2-4cf2-ae38-0fad5b384310",
        "nodeId": "417f8e17-44e5-44fe-84a9-c16b98b2eddf4",
        "type": "GLOBAL_SCOPE_FILE",
        "documentHash": "8d4bba7d0ba4e6189e29b60597828f9d9a1414be5450ca0dd82af74cfc6f882",
        "audited": true,
        "id": "e7443544-7f8-44d4-a331-9ed3a25332ed",
        "nodeId": "417f8e17-44e5-44fe-84a9-c16b98b2eddf4",
        "type": "GRANULAR_SCOPE_1",
        "documentHash": "40f0d7260d8d80d611d60a7b9b43d9fbcab5fb2539d94abe535215a43d9e2b",
        "audited": true,
        "id": "6cdbf588-6d4f-4ad4-84c3-ac5eb0b51bb7",
        "nodeId": "417f8e17-44e5-44fe-84a9-c16b98b2eddf4",
        "type": "GRANULAR_SCOPE_2",
        "documentHash": "841c86657cf9118527e62daff7d0ef3ba21f81ea3e8c36da2d2f5ce65c1a137",
        "audited": true
      }
    ]
  ]
}
```



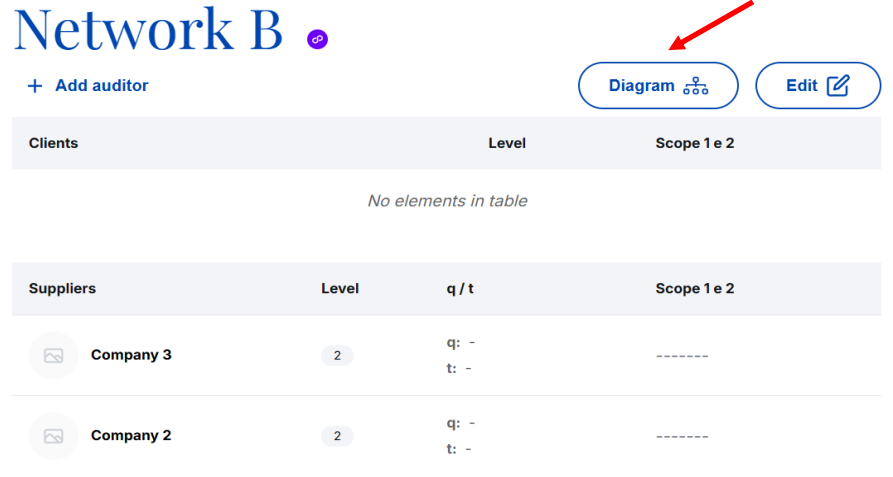
# VERIFY DATA ON-CHAIN – NETWORK TREE (1/3)

Verifying integrity of network tree structure

Since only **Business Network Admin** has access to the entire tree structure, he is the only one who can do the verification of the tree network structure.

The following are the steps to perform to verify the integrity of the network tree structure.



- Open Network tree from “Organigram Button”



The screenshot shows the 'Network B' interface. At the top left, there is a '+ Add auditor' button. To the right, there are two buttons: 'Diagram' (with a diagram icon) and 'Edit' (with a pencil icon). A red arrow points to the 'Diagram' button. Below the buttons, there are two tables. The first table is for 'Clients' and is currently empty, with the text 'No elements in table' below it. The second table is for 'Suppliers' and contains two rows of data.

Clients	Level	Scope 1 e 2
No elements in table		

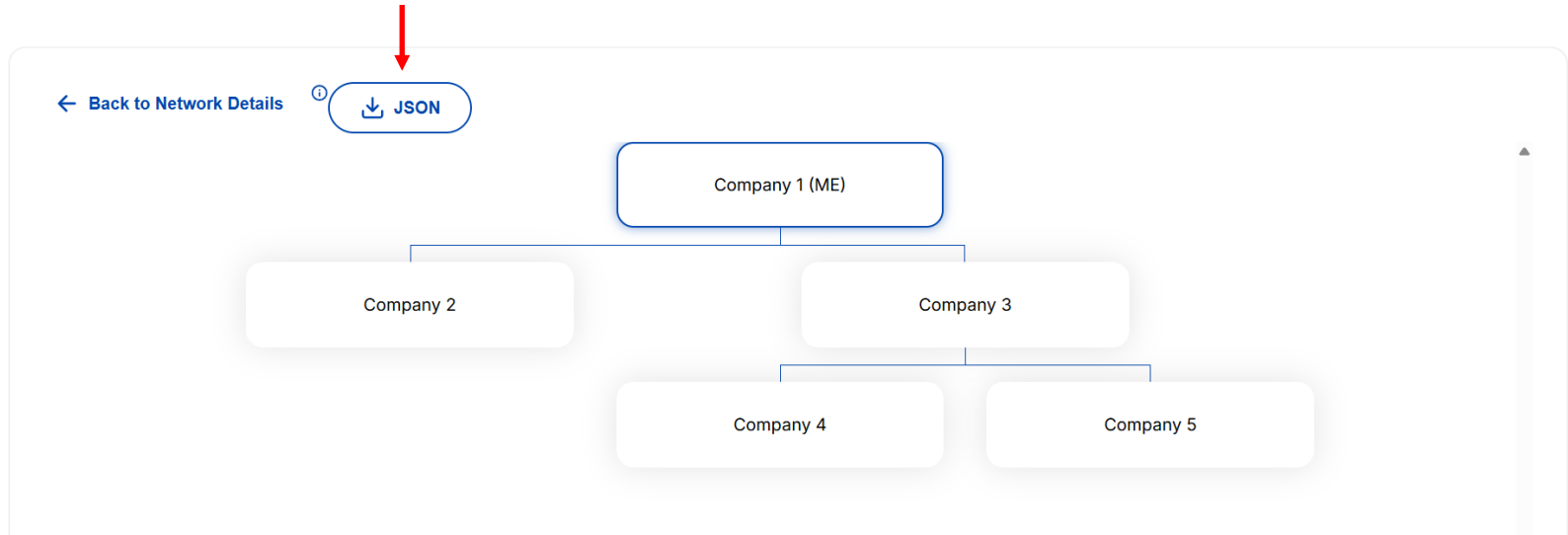
Suppliers	Level	q / t	Scope 1 e 2
 Company 3	2	q: - t: -	-----
 Company 2	2	q: - t: -	-----



# VERIFY DATA ON-CHAIN – NETWORK TREE (2/3)

Verifying integrity of network tree structure

- Download the tree json from “JSON” button (*organigram.json*)



# VERIFY DATA ON-CHAIN – NETWORK TREE (3/3)

Verifying integrity of network tree structure

- Using the endpoint `/public/network/merkle-hash`, it is possible to send a POST API request with the downloaded JSON as the body (*in the previous slide*). The API returns the Merkle root hash, which should match the `merkleHash` field in the metadata.json

The screenshot shows a web browser interface for a REST client. The top part displays a POST request to the endpoint `https://bledger.blockchainreply.it/api/v1/public/network/merkle-hash`. The request body is a JSON tree structure representing a network tree. The bottom part displays the response body, which is a JSON metadata object. Both the `merkleHash` field in the request and the `hash` field in the response are highlighted with red boxes, indicating they match.

```
POST https://bledger.blockchainreply.it/api/v1/public/network/merkle-hash

Params Authorization Headers (11) Body Scripts Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2   "companyId": "d8ecfdbb-fd12-4fb5-b0ba-c6ce42ac3b45",
3   "companyName": "Company 1",
4   "children": [
5     {
6       "companyId": "eaa92704-312e-40c5-aae1-8891e090f9bd",
7       "companyName": "Company 2",
8       "children": []
9     },
10    {
11      "companyId": "76cc409f-c66a-4663-add9-9001921009ea",
12      "companyName": "Company 3",
13      "children": [
14        {
15          "companyId": "3a18f13c-2cb2-4d6a-e400-f82712d88751",
16          "companyName": "Company 4",
17          "children": []
18        }
19      ]
20    }
21  ]
22 }

"networkId": "dd3e907d-bfd9-4d36-84df-43daacda63f7",
"networkCreated": "2025-10-21T14:12:00.954692Z",
"networkUpdated": "2025-10-21T14:21:23.281446Z",
"metadataCreated": "2025-10-21T14:21:23.281446Z",
"merkleHash": "284918d7be8f1e7ac1de32e7ea219a200f18f6ccebef795c14b326cb12d1553a",
"membersCount": 5,
"uploadedCount": 1,
"auditedCount": 0,
"docs": [
  {
    "id": "fd217c2b-9aa7-44fb-b389-31338d3c477d",
    "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
    "type": "GLOBAL_SCOPE_FILE",
    "documentHash": "9a4683f671f98cd890a3ec60ad1415f50faa4766fe61a89930dbf1dff8f62248"
  },
  {
    "id": "b211be40-c2dd-468d-a2f4-af78a62e9711",
    "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
    "type": "GRANULAR_SCOPE_1",
    "documentHash": "09cb1db4d69054d360b0ef9386053eff61aacefa37c3a67c4ada82ac9133a357"
  },
  {
    "id": "9bc94dfa-9ab6-44ae-95fa-cf5f70deeb9e",
    "nodeId": "5f9f957c-844d-4bb1-bec2-9237266a3ed3",
    "type": "GRANULAR_SCOPE_2",
    "documentHash": "13cb47db1de53a256df0889813e282c624835d92e6dd799984338f42089614ec"
  }
]
}
```



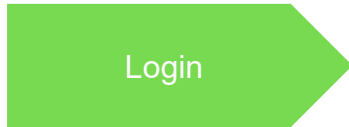
# BLEDGER SCREENSHOTS

---



# BLEDGER PLATFORM

- The platform can be accessed at the following link: <https://bledger.blockchainreply.it/>
- The following slides present screenshots of the various features offered by the platform



*B-Ledger.demo*

## B-Ledger

Digital platform for collaborative data-collection in business networks

Role  
Select role

Company  
Select company

Login

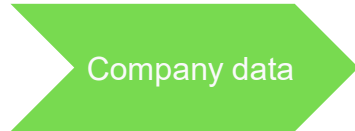
B-Ledger @ 2025. All rights reserved.




# BLEDGER PLATFORM

Company profile


- **Overview** section provides information related to the company



B-Ledger.demo [Overview](#) [Networks](#) [Logout \(Company 1\) ↗](#)



**Company 1**

Digital Wallet   
0xe1819260719b1d9136b34758ad1968309398f796

**Company data**

Nation	Company Name
ITALY	Company 1
ID Type	ID Number
ID_CARD	12345678

**Representative data**

Name	Surname
Mario	Rossi
E-mail	
company1@company.com	

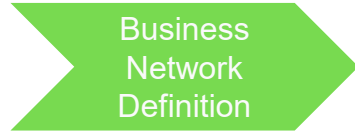
B-Ledger @ 2025. All rights reserved.



# BLEDGER PLATFORM

## Business Network Definition – 1/3

- This section presents the step-by-step process for creating a network



## Network name Step 1/2

Write the name you want to associate to your new network.

Select the Governance Model

Select the Data Model

Select the Auditor for this network

The auditor will be responsible for validating Scope 1 & 2 data.

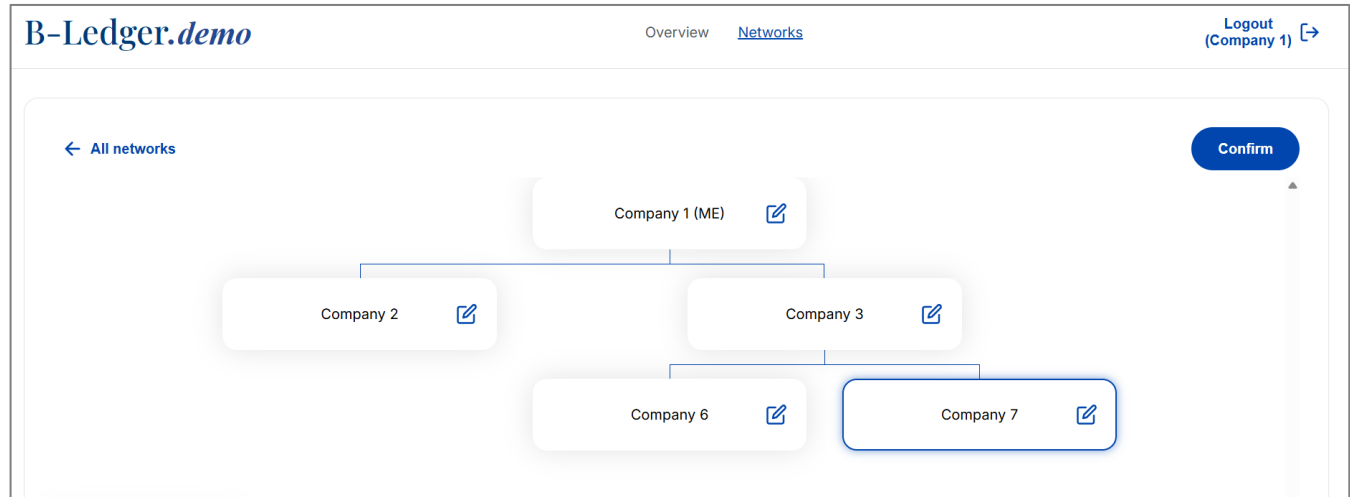
Company 8 ×



# BLEDGER PLATFORM

Business Network Definition – 2/3

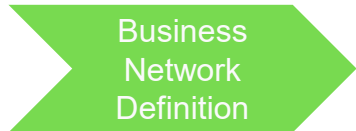
Business  
Network  
Definition



# BLEDGER PLATFORM

Business Network Definition – 3/3

- The company may choose to accept or decline the invitation to join a network



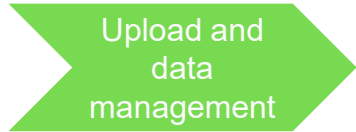
A screenshot of the BLEDGER PLATFORM interface. At the top, there are navigation links for "Overview" and "Networks", and a "Logout (Company 3)" button with an external link icon. Below this, there is a section for "Network test" with a "6" indicator. A pagination bar shows "&lt; 1/1 &gt;" and "Rows per page: 10". Underneath is an "Invitations" section. The first invitation is for "Network C", which has two buttons: "Accept" (blue) and "Decline" (red). These two buttons are enclosed in a red rectangular box. At the bottom, another pagination bar shows "&lt; 1/1 &gt;" and "Rows per page: 10".



# BLEDDER PLATFORM

Data management – upload data – 1/4


- To calculate the **Emission Intensity**, a network member should upload its own Scope 1 and 2 files, must upload the **q** and **t** values of each supplier, and its suppliers should calculate their **Emission Intensity**



### Upload documents

**1. Scope 1 & 2 documents**  
This document must contain the data regarding Scope 1 & 2.

[Download template](#)




[Choose a file](#)  
Or drop it here

Supported file: **.CSV**  
Max. dimension: **5mb**

**2. Emission details (SCOPE 1)**  
This document must contain the explanation of data contained in Scope 1 & 2 document

[Download template](#)




[Choose a file](#)  
Or drop it here

Supported file: **.CSV**  
Max. dimension: **5mb**

**3. Emission details (SCOPE 2)**  
This document must contain the explanation of data contained in Scope 1 & 2 document

[Download template](#)



[Choose a file](#)  
Or drop it here

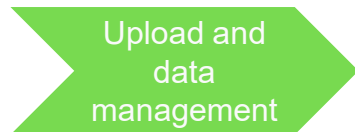
Supported file: **.CSV**  
Max. dimension: **5mb**

[Cancel](#) [Next Step](#)















# BLEDGER PLATFORM

Data management – upload data – 2/4



**Upload documents**

 File name: scope_1_2_data_facts.csv 0.3kb	 File name: granular_scope_1_data_facts.csv 0.5kb	 File name: granular_scope_2_data_facts.csv 0.3kb
Verifying 	Verifying 	Verifying 
Uploading 	Uploading 	Uploading 
Data extraction 	Data extraction 	Data extraction 

[Delete Data](#)

**Extracted data**

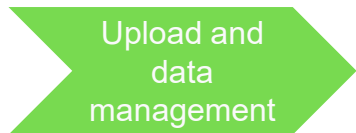
<b>Scope 1</b>	<b>47,700.94 tCO<sub>2</sub>e</b>
Stationary combustion	7,683.00 tCO <sub>2</sub> e
Mobile combustion	15.67 tCO <sub>2</sub> e
Fugitive emission	10,256.00 tCO <sub>2</sub> e
Business process emission	42.27 tCO <sub>2</sub> e
Other direct emission	29,704.00 tCO <sub>2</sub> e
<b>Scope 2</b>	<b>96,152.00 tCO<sub>2</sub>e</b>
Scope 2 electricity emission	34,288.00 tCO <sub>2</sub> e
Scope 2 other energy emission	61,864.00 tCO <sub>2</sub> e
<b>Tot.</b>	<b>143,852.94 tCO<sub>2</sub>e</b>

[Confirm data](#)



# BLEDGER PLATFORM

Data management – upload data – 3/4




2025

## My Scope 1 e 2 file

In this section you can access your uploaded Scope1 and Scope2 documents, review the data, and request verification from an auditor


EMISSION DETAILS

 **scope\_1\_2\_data\_facts.csv**

Uploaded 23/10/2025


[Request verification](#)

EMISSION DETAILS

 **granular\_scope\_1\_data\_facts.csv**

Uploaded 23/10/2025

EMISSION DETAILS

 **granular\_scope\_2\_data\_facts.csv**

Uploaded 23/10/2025

[Download All](#)

<b>Scope 1</b>	<b>47,700.94</b> tCO <sub>2</sub> e
<b>Scope 2</b>	<b>96,152.00</b> tCO <sub>2</sub> e
<b>Tot.</b>	<b>143,852.94</b> tCO <sub>2</sub> e

[Read More](#)

## Compute your Scope 3

Use your network data to compute Emission Intensity and Scope 3 emissions.

**33.33%**

Some suppliers have not yet calculated their Emission Intensity or are missing quantity or transport emissions.

**1 Compute Emission Intensity**

Make sure you've set the quantity and transportation emissions for all of your suppliers

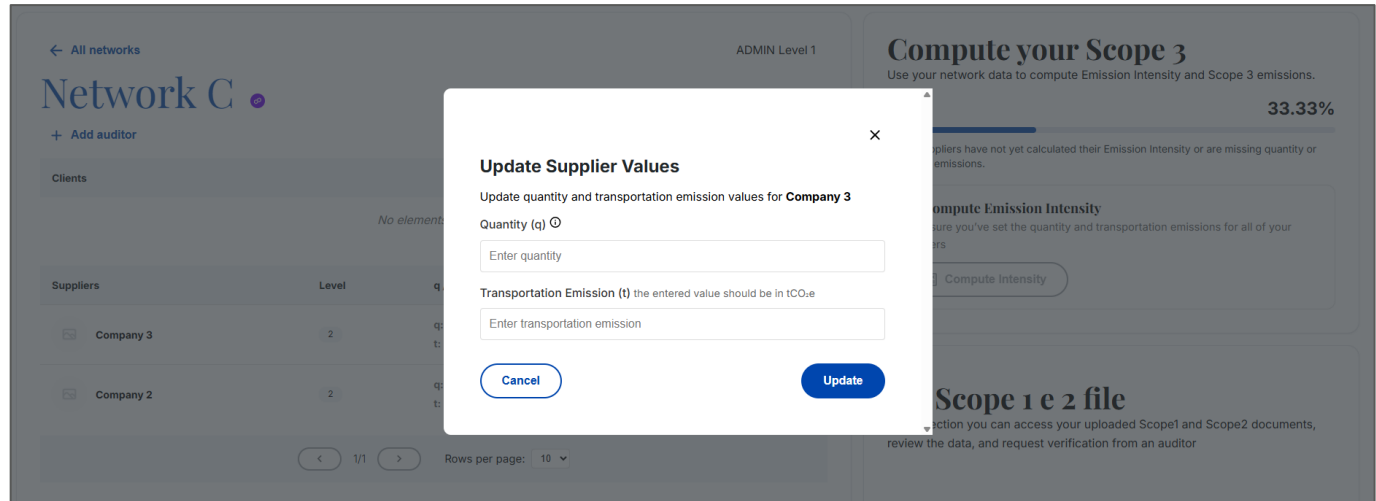
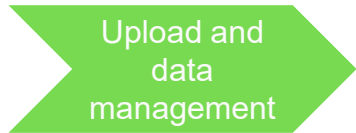
[Compute Intensity](#)



# BLEDGER PLATFORM

Data management – upload data – 4/4

- A network member must upload the **q** and **t** values of each of its suppliers to calculate its own *Emission Intensity*



The screenshot displays the Bledger Platform interface. On the left, a sidebar shows 'Network C' with options to 'Add auditor' and view 'Clients' and 'Suppliers'. The 'Suppliers' section contains a table with two entries:

Suppliers	Level	q	t
Company 3	2	q:	t:
Company 2	2	q:	t:

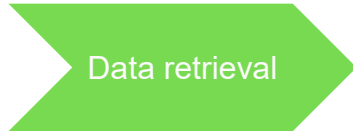
A modal window titled 'Update Supplier Values' is open, prompting the user to 'Update quantity and transportation emission values for Company 3'. It includes two input fields: 'Quantity (q)' and 'Transportation Emission (t)'. The modal also features 'Cancel' and 'Update' buttons. In the background, a 'Compute your Scope 3' section is visible, showing a progress bar at 33.33% and instructions to 'Compute Emission Intensity'.



# BLEDGER PLATFORM

Data retrieval – Scope 3 calculation

- Once the Emission Intensity is calculated, it is possible to calculate **Scope 3**



## Compute your Scope 3

Use your network data to compute Emission Intensity and Scope 3 emissions.

100%

Emission Intensity is ready to be computed.

**1 Compute Emission Intensity**  
Make sure you've set the quantity and transportation emissions for all of your suppliers

[Compute Intensity](#)

Emission Intensity  
**62.528.984,42 tCO<sub>2</sub>e**

**2 Enter Production Volume**  
Production Volume

e.g., 10

[Compute Scope 3](#)



## Compute your Scope 3

Use your network data to compute Emission Intensity and Scope 3 emissions.

100%

Emission Intensity is ready to be computed.

**1 Compute Emission Intensity**  
Make sure you've set the quantity and transportation emissions for all of your suppliers

[Compute Intensity](#)

Emission Intensity  
**62.528.984,42 tCO<sub>2</sub>e**

**2 Enter Production Volume**  
Production Volume

45

[Compute Scope 3](#)

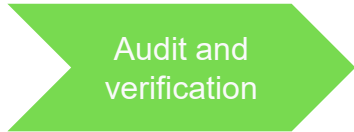
Scope 3 Emissions  
**2.813.804.298,9 tCO<sub>2</sub>e**



# BLEDGER PLATFORM

Audit and verification – 1/3


- A member network could **request the verification** of the data upload, and the network auditor will be able to view the files and verify them




## My Scope 1 e 2 file

In this section you can access your uploaded Scope1 and Scope2 documents, review the data, and request verification from an auditor


EMISSION DETAILS


 **scope\_1\_2\_data\_facts.csv**

Uploaded  23/10/2025


[Request verification](#)


EMISSION DETAILS

 **granular\_scope\_1\_data\_facts.csv**

Uploaded  23/10/2025

EMISSION DETAILS

 **granular\_scope\_2\_data\_facts.csv**


Uploaded  23/10/2025

[Download All](#)



### Request Audit Verification

Select an auditor to request verification for your Scope 1 & 2 data

Company 8 

*The selected auditor will verify your uploaded data.*

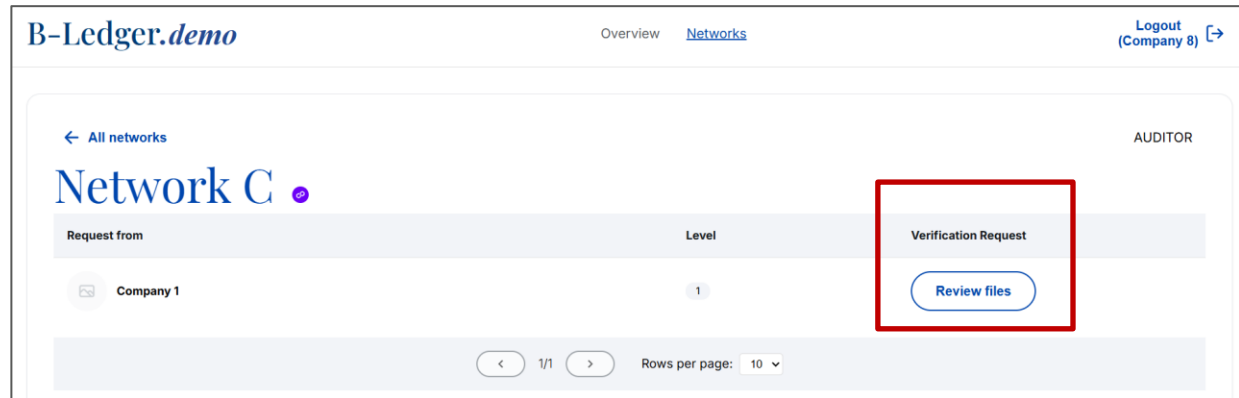
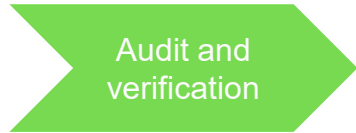
[Cancel](#) [Request Verification](#)



# BLEDGER PLATFORM

Audit and verification – 2/3

- The **auditor** will see the **verification request** within the network section



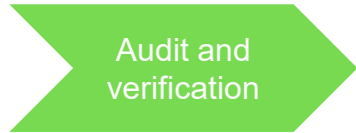
The screenshot shows the B-Ledger demo interface. At the top left is the logo 'B-Ledger.demo'. To its right are links for 'Overview' and 'Networks'. At the top right is a 'Logout (Company 8)' link with an external icon. Below the header, there is a navigation bar with a back arrow and 'All networks' on the left, and 'AUDITOR' on the right. The main content area is titled 'Network C' with a purple icon. Below this is a table with columns 'Request from', 'Level', and 'Verification Request'. The first row shows 'Company 1' under 'Request from', '1' under 'Level', and a 'Review files' button under 'Verification Request'. The button is highlighted with a red box. At the bottom of the table, there are navigation controls: '< 1/1 >' and 'Rows per page: 10' with a dropdown arrow.



# BLEDGER PLATFORM

Audit and verification – 3/3

- Once the data has been verified by the auditor, the network member will see that the **files** have been **verified** and can **check the transaction** on the block explorer



### Verification

SCOPE 1 & 2	Scope 1	47,700.94 tCO <sub>2</sub> e
<b>scope_1_2_data_facts.csv</b> Uploaded 23/10/2025	Stationary combustion	7,683.00 tCO <sub>2</sub> e
<a href="#">Download</a>	Mobile combustion	15.67 tCO <sub>2</sub> e
	Fugitive emission	10,256.00 tCO <sub>2</sub> e
	Business process emission	42.27 tCO <sub>2</sub> e
	Other direct emission	29,704.00 tCO <sub>2</sub> e
	<b>Scope 2</b>	<b>96,152.00 tCO<sub>2</sub>e</b>
	Scope 2 electricity emission	34,288.00 tCO <sub>2</sub> e
	Scope 2 other energy emission	61,864.00 tCO <sub>2</sub> e
	<b>Tot.</b>	<b>143,852.94 tCO<sub>2</sub>e</b>

EMMISSION DETAILS (SCOPE 1)  
 **granular\_scope\_1\_data\_facts.csv**  
Uploaded 23/10/2025  
[Download](#)

EMMISSION DETAILS (SCOPE 2)  
 **granular\_scope\_2\_data\_facts.csv**  
Uploaded 23/10/2025  
[Download](#)

[Close](#) [Verify data](#)

To verify the file, you have to download it first.



### 2025 My Scope 1 e 2 file

In this section you can access your uploaded Scope1 and Scope2 documents, review the data, and request verification from an auditor

EMMISSION DETAILS  
 **scope\_1\_2\_data\_facts.csv**  
Uploaded 23/10/2025 **Verified 23/10/2025**

EMMISSION DETAILS  
 **granular\_scope\_1\_data\_facts.csv**  
Uploaded 23/10/2025 **Verified 23/10/2025**

EMMISSION DETAILS  
 **granular\_scope\_2\_data\_facts.csv**  
Uploaded 23/10/2025 **Verified 23/10/2025**

[Download All](#)

