

# Non-functional requirements

Performance	1
Security	1
Reliability and Availability	3
Usability	3
Interoperability	3
Data Management	4
Cost Efficiency	4
Operational	4

## Performance

**Latency: The time taken to confirm a transaction should be minimized and within acceptable limits.**

The system has been tested for a performance with 3 concurrent users and can be found in the performance.

We found that even with 3 concurrent users the blockchain execution is the bottleneck and the performance plateaus at around max. 16-18 seconds for a function call.

## Security

**Data Integrity: Ensure that data stored on the blockchain is tamper-proof and immutable.**

The BC24 contract once launched can not be changed.

MetaData can only be overridden by user with the same role.

**Authentication and Authorization: Implement robust mechanisms to verify the identity of users and control their access to the system.**

The only predefined and necessary role needed is the Admin role which is defined at contract creation. This wallet has all the privileges on the contract.

Role	Function	Authentication	Authorization
Admin	<ul style="list-style-type: none"> <li>- Setting permissions of users</li> <li>- Access to all functions in case of corrections/ auth deadlocks</li> </ul>	<i>AccessControl</i>	All

Communication with any function will follow the access control logic defined in the templates.

All signed transaction that invoke a function will be controlled for their respective permissions by using the ERC1155 Interface *AccessControl*.

**Smart Contract Security: Ensure that smart contracts are free from vulnerabilities.**

The contract is tested with unit tests and stress tests.

See documentation for more information.

## Reliability and Availability

**Uptime:** The system should guarantee high availability, with minimal downtime.

There are 4 nodes in our private network.

**Fault Tolerance:** The system should be resilient to failures and ensure continuous operation even in the event of node failures.

<https://blog.web3labs.com/web3development/comparing-byzantine-fault-tolerance-consensus-algorithms>

## Usability

**User Interface:**

There exists an API that allows easy and dynamic interaction with the contract.

All the functionality of the smart contract is to be integrated directly into the webapp Trace connect. In that sense, the aforementioned webapp is the only graphical interface.

**Documentation:** Comprehensive documentation should be available for users and developers.

The contract code itself is commented for easier understanding.

Additionally a brief description of functions and their arguments is given.

The contract is a generalization of a traceability project. As such it allows to digitally replicate a supply chain, given the production follows the described templates.

See the actual contract documentation for more information.

## Interoperability

**Compatibility:** Ensure the smart contract can interact with the webapplication Trace connect.

The smart contract is callable with its address from any context. As such it allows the connection via different means (python, php, js and others).

Smart Traceability REST API allows easy and intuitive access and interaction with the contract

**Standards Compliance:** Adhere to industry standards and protocols to facilitate interoperability.

The contract follows the industry standard of the ERC1155 implementation.

## Data Management

**No external datastorage: The contract should be usable without additional means such as external Databases or IPFS.**

The contract uses only on-chain data storage and utilises event- and transaction filtering to get the saved on-chain and transaction data.

## Cost Efficiency

**Transaction Fees: Minimize the cost of transaction fees for users.**

The network utilizes a gas price of 0 Gas.

## Operational

**Deployment: Ensure smooth and efficient deployment processes for smart contracts on the blockchain network.**

The deployment via the hardhat development kit is trivial.

See Readme of the repository.

**Monitoring: Implement monitoring tools to track the performance and health of the smart contract system.**

The system with validator, block explorer and further functionality is provided by the besu hyperledger private network infrastructure.