

**Supporting Agri food communities and networks in SEE
through developing services on digital food quality and
traceability using blockchain**

IMPLEMENTATION SPECIFICATION

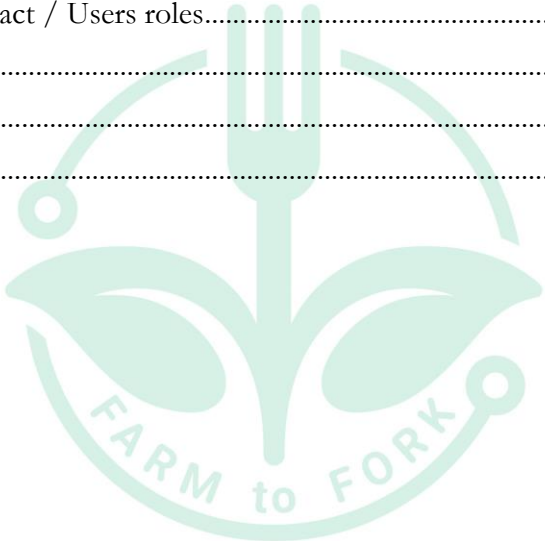
ROADMAP



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 818182

CONTENTS

- 1. INTRODUCTION.....3
 - 1.1. About Consortium4
 - 1.2. About Project.....4
 - 1.3. About SmartAgriHubs4
- 2. BLOCKCHAIN SOLUTION5
- 3. COMMUNICATION HUB.....6
 - 3.1. Accessing Smart Contract6
 - 3.1.1. Web implementation6
 - 3.1.2. Backend implementation7
 - 3.2. Users Smart Contract / Users roles.....8
- 4. CODE LISTS.....12
 - 4.1. Product code list.....12
 - 4.2. Location code list12



1. INTRODUCTION

This document Implementation specification describes in detail the implementation requirements of project “Supporting Agri food communities and networks in SEE through developing services on digital food quality and traceability using blockchain”.

This document represents the contractual delivery of:

- Deliverable 3 Technology roadmap
- Deliverable 4 Lab pilot case
- Deliverable 5 Prepared documents for education of DIHs

This document was created by Partner Cuspis d.o.o., as a result of Cuspis d.o.o. domain knowledge, meetings held with Consortium Partners and stakeholders and collected information from producers.

This document describes element of blockchain solutions – Communication hub (Smart Contract), user roles, data and accessing Smart Contract. Everything described is working on testing environment.



1.1. ABOUT CONSORTIUM

Consortium of four Digital Innovation Hubs in South East Europe partnering with one Competence Centre and one blockchain IT company, is geographically dispersed but reaches wider range of stakeholders throughout the Region.

Consortium Partners are:

[Green DIH](#), Digital Innovation Hub from Romania – Lead Partner

[DIH TERA](#), Digital Innovation Hub from Croatia

[DIH Hadas Bar](#), Digital Innovation Hub from Israel

[DIH Smart Village](#), Digital Innovation Hub from Bosnia and Herzegovina

[Centar Kompetencija](#), Competence Centre from Croatia

[Cuspis](#), Blockchain IT company from Croatia

1.2. ABOUT PROJECT

The Project “Supporting Agri food communities and networks in SEE through developing services on digital food quality and traceability using blockchain – DFTB” increased participation of DIHs in South East Europe Region in digital transformation of Agri-food sector with focus on food traceability using blockchain. Project focuses on supporting services that can significantly improve entrepreneurial business with the help of digitalization using blockchain technology creating transparent supply chain from farm to fork as well as enable customer to know history of the specific package or piece of food they are buying. Project focus is on food processors, customers, regulators, local government officials, and consumer protection association representatives. As a result of the Project, participating DIHs are able to offer range of new digital services: Scouting and ecosystem analysis, Technology Road Map, Proof of concept, Workshops and seminars, Lab facilities, Support for innovative business models.

1.3. ABOUT SMARTAGRIHUBS

SmartAgriHubs is a project funded by the European Union through the Horizon 2020 funding instrument that brings together a consortium of over 164 partners in the European agri-food sector. SmartAgriHub aims to digitize European agriculture by promoting an agricultural innovation ecosystem dedicated to excellence, sustainability, and success. The SmartAgriHubs project serves as a catalyst for innovation in smart farming. By embracing and promoting European innovation in agriculture and IT, SmartAgriHubs is expanding digital solutions in the agri-food sector and helping farmers and their businesses achieve actual and achievable results.

2. BLOCKCHAIN SOLUTION

Blockchain solution consist of following elements in testing environment:

- Self-assessment tool for use by stakeholders to determine their current stage of digitalisation. Output that will give directions how to use digital food quality and digitalisation tools
- Communication hub for collecting data with automatic integration to web site and blockchain
- Web site tool for stakeholders to directly input data about their production lifecycle or supply chain that will be automatically written and sent to communication hub and written to blockchain and published on web
- API interface for stakeholders that have their own IT system in place where they'll send their production lifecycle or supply chain data through API to communication hub which will be automatically published on web and written in blockchain
- Web site for consumers where they'll be able to search by the lot number and find all relevant information about production of this specific food they are buying, search by products and by producers



3. COMMUNICATION HUB

Communication hub is used for collecting data and has automatic integration to web site and blockchain. In this case communication hub is represented by Smart Contract.

Communication hub has 2 Smart Contracts – one is Admin Smart Contract and one is Users Smart Contract.

In general, Admin Smart Contract has methods for:

- adding/changing admin addresses that can use this Contract
- adding new products to blockchain with defined parameters and values for that product
- reading data stored on blockchain

In general, Users Smart Contract has methods for:

- adding new lot numbers and registering address to specific lot number
- adding new parameters for existing products to blockchain
- reading data stored on blockchain

Each Smart contract has its own roles – Admin Smart Contract has Admin role, and Users Smart Contract has Producer, Processor, Distributor and POS.

For connecting existing system to this system, user needs only Users Smart Contract.

3.1. ACCESSING SMART CONTRACT

Users Smart Contract deployed address is:

0xdE5335aFcB4858AF66C7ed84bB5941F41898A77A

There are two ways of accessing Users Smart Contract:

1. By web implementation
2. By backend implementation

3.1.1. Web implementation

This method is using prebuild library web3.min.js for accessing Users Smart Contract. web3.js is a collection of libraries that allows user to interact with a local or remote Ethereum node. The web3.js library is an open-source JavaScript library that includes functions to communicate with an Ethereum node via the JavaScript Object Notation - Remote Procedure Call (JSON-RPC) protocol. Using prebuild library makes implementation easier for users.

Example of sending information to blockchain

1. Setting the default address for specific transaction
2. Defining contract by putting ABI code and giving the contract address (both publicly accessible)
3. Calling the method setProduct and providing 3 parameters (lotNumber, params and measurementDate)
4. Result is in json format

```
web3.eth.defaultAccount = window.userWalletAddress;
    contract = new window.web3.eth.Contract(
        window.ABI,
        CONTRACT_ADDRESS
    );
    var result = await contract.methods
        .setProducts(lotNumber, params, measurementDate)
        .send({ from: window.userWalletAddress })
        .catch((err) => {
        });
```

3.1.2. Backend implementation

This method is using prebuild library Nethereum for .NET platform. Nethereum is an open-source .NET integration library for Ethereum that simplifies the access and smart contract interaction with Ethereum nodes both public or permissioned.

Example of using own Ethereum node

1. Defining location of the node
2. Defining the contract where we put ABI and contract address (both publicly accessible)
3. Defining the method that is called
4. Calling the method providing the necessary parameters

```
var web3 = new Web3Geth("NodeAddress");
    var contract = web3.Eth.GetContract(ABI, contractAddress);
    var result = "";
    var callMethod = contract.GetFunction(contractMethod);
    result = await callMethod.CallAsync<string>(param1, param2);

    return result;
```

3.2. USERS SMART CONTRACT / USERS ROLES

Users will have roles according to their role in product lifecycle and product stages – production, processing, distribution, sale.

- 1 – Producer
- 2 – Processor
- 3 – Distributor
- 4 – POS

USERS CONTRACT – methods with explanation and examples

setLotEndDate				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff
2	_endD	string	Y	DD.MM.YYYY 26.07.2022
3	_wAddress	address	Y	0x81cd3A3645FBA1Bf24Bf7F31187Ca15813dEa815

setLotNumberInfo				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff
2	_productId	uint256	Y	1
3	_station	string	Y	My Farm
4	_location	string	Y	1191917921
5	_year	string	Y	2022

setLotNumberAddresses				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff
2	_wAddress	address	Y	0x81cd3A3645FBA1Bf24Bf7F31187Ca15813dEa815

3	_sDate	string	Y	DD.MM.YYYY 26.07.2022
4	_eDate	string	Y	DD.MM.YYYY 26.07.2022
5	_role	uint256	Y	1

setProductParameters				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff
2	_attributes*	string[]	Y	["parameter1_value","parameter2_value","parameter3_value","parameter4_value","parameter5_value","parameter6_value","parameter7_value","parameter8_value","parameter9_value","parameter10_value"] ["18","50","11","0","1","","","","",""]
3	_measurementDate	string	Y	26.07.2022

* All parameter values that are measured at that date, needs to be entered here. For example, Eggs have following parameters value for role 1 - Producer:

- TEMPERATURE (°C) - 18
- HUMIDITY (%) - 50
- LIGHT (hours/day) - 11
- TEST FOR SALMONELLA BACTERIA - Absent
- TEST FOR PEST CONTROL – Present
- PROCESSING TEMPERATURE (°C)
- TRANSPORT TEMPERATURE (°C)
- TRANSPORT HUMIDITY (%)
- STORE TEMPERATURE (°C)
- STORE HUMIDITY (%)

That means that _attributes will be:

["18","50","11","0","1","","","","",""]

If parameter isn't measured by specific role, for this parameter is entered empty string.

getLotNumbersByAddress				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_wAddress	address	Y	0x81cd3A3645FBA1Bf24Bf7F

				31187Ca15813dEa815
--	--	--	--	--------------------

getProductAddressesById				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_productId	uint256	Y	1

getLotNumberInfoByLotNumber				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff

getLotNumberAddressesByLotNumber				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff

getProductsInfo				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
/	/	/	/	/

getProductInfoById				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_productId	uint256	Y	1

getDataByAddress				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_address	address	Y	0x81cd3A3645FBA1Bf24Bf7F31187Ca15813dEa815

getDataByAddressAndLotNumber				
ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_address	address	Y	0x81cd3A3645FBA1Bf24Bf7F31187Ca15813dEa815
2	_lotNumber	string	Y	

getTestsByAddressAndLotNumber				
--------------------------------------	--	--	--	--

ID	NAME	DATA TYPE	REQUIRED (Y/N)	EXAMPLE
1	_address	address	Y	0x81cd3A3645FBA1Bf24Bf7F31187Ca15813dEa815
2	_lotNumber	string	Y	personalized or predefined in format YYYY-ProductID-LocationCode-yyyyMMddHHmmssfff



4. CODE LISTS

List of Code lists used in this project

4.1. PRODUCT CODE LIST

This list includes list of products with product IDs, products parameters, optimal values, etc. in accordance with users' roles.

4.2. LOCATION CODE LIST

This list includes list of location code list.

