**DL-Tags**

**Grant agreement 688061**

# DL-Tags requirements and initial design

| | |
|---|---|
| Deliverable ID: | **D1.1** |
| Deliverable Title: | **DL-Tags requirements and initial design** |
| Revision #: | **2.0** |
| Dissemination Level: | **PU** |
| Responsible beneficiary: | **UniZG-FER** |
| Contributing beneficiaries: | |
| Contractual date of delivery: | **31.5.2018** |
| Actual submission date: | **20.6.2018** |

Start Date of the Project: 1 January 2016 Duration: 36 Months

# Contents

***Section 1 - Introduction ..... 4***

***Section 2 - Distributed Ledger Technology (DLT) ..... 5***

***Section 3 - DL-Tags ..... 7***

**3.1 System requirements ..... 9**

**3.2 Sequence diagram ..... 9**

***Section 4 - DL-Tags Architecture ..... 12***

**4.1 Components ..... 12**

**4.2 Functionalities ..... 12**

4.2.1 Create product ..... 13

4.2.2 Change owner ..... 14

4.2.3 Added stakeholder notification and reaching a consensus about product authenticity ..... 15

4.2.4 Subscribe/unsubscribe to notifications ..... 16

**4.3 Implementation plans ..... 16**

***Section 5 - Conclusions ..... 18***

***Section 6 - References ..... 19***

## Document history

| Revision | Date | Modification | Authors |
|---|---|---|---|
| v0.1 | 10.5.2018 | ToC | Pavle Skocir |
| v0.2 | 15.5.2018 | Initial version of the architecture | Federico Matteo Bencic, Pavle Skocir |
| v0.3 | 1.6.2018 | Elaborated version of the architecture | Federico Matteo Bencic, Pavle Skocir |
| v1.0 | 15.6.2018 | Introductory parts | Pavle Skocir |
| v2.0 | 17.7.2018. | Added Section 2 about DLT, updates based on comments by the reviewers | Federico Matteo Bencic, Pavle Skocir |
| | | | |

## Section 1 - Introduction

The main requirement of IoT-enabled circular economy is to share product-related information between involved stakeholders. European Product Passport (EPP), as one of the key ingredients facilitating the processes driving circular economy, depends on a sustainable and privacy-preserving information sharing model where involved stakeholders share/exchange product-related data, possibly encoded in Smart Tags, and have the possibility to check product's authenticity and modify its current state.

TagITSmart (TIS) system offers a solution for issuing Smart Tags and sharing product information between stakeholders involved in the product lifecycle. However, by using the existing TIS infrastructure, a product consumer has to implicitly trust the TIS platform as a Smart Tag creator and other stakeholders that they are providing authentic data about the product.

DL-Tags project steps into this environment as a solution for sharing product information in a privacy-preserving and decentralized way, under the assumption that a single centralized authority should be avoided to orchestrate the relevant processes. Data authenticity and integrity are vital to ensure sustainable management of circular economy processes. DL-Tags proposes the use Digital Ledger Technology (DLT) as a decentralized solution capable of ensuring data authenticity and integrity [1][2]. By enforcing a DL-Tags solution based on Blockchain as a DLT, stakeholders are required to reach a consensus about the information available through TIS Smart Tags, thus avoiding the possibility to blame one of the former stakeholders in the product lifecycle for the unacceptable facts. Furthermore, the proposed solution enables end consumers to verify if a product belongs to a certain brand as advertised, and to verify that the changes of product owners are acknowledged by all stakeholders.

The proposed decentralized solution provides a possibility for stakeholders to store and share product-related data in a peer-to-peer (P2P) fashion with the ability to control the flow of shared data. Blockchain and DLT serve as a trusted intermediary between the stakeholders, while the data exchange happens solely between the involved stakeholders by forwarding the physical product labelled with a Smart Tag. Therefore, stakeholders not willing to expose their business-related data to third parties who do not participate in product lifecycle might have the incentive to use the DL-Tags solution.

This deliverable describes the main benefits and initial design of DL-Tags system. Section 2 provides an insight into DLT and blockchain. Section 3 presents main features of the system and system requirements. Architecture of the system and implementation plans are introduced in Section 4. Concluding remarks are given in Section 5.

## Section 2 - Distributed Ledger Technology (DLT)

Distributed ledger technology (DLT) enables the maintenance of a global, append only, data structure by a set of mutually untrusted participants in a distributed environment [5]. The most notable features of distributed ledgers are immutability, resistance to censorship, decentralized maintenance, and elimination of the need for a centralized trusted third party. In other words, there is no need for an entity to be in charge of conflict resolution and upkeep of a global truth that is trusted by all stakeholders who do not trust each other. DLT is suitable for tracking the ownership of digital assets, and it's most prominent application is the Bitcoin network [6]. DLT holds promise beyond mere cryptocurrency transfer since an entry in the ledger may be generalized to hold arbitrary data.

DLT stores transactions in an open ledger. A ledger has its state. Transactions serve as inputs that cause the change to the state, hence DLT can generally be regarded as transaction-based state machines. Different data structures exist for maintaining the ledger. One of these specializations of DLT which is currently well-covered in literature and often used as synonym for DLT is blockchain [4]. The alternative is distributed acyclic graphs (DAG). While blockchain stores transactions in blocks, DAG stores transactions in nodes. Since blockchain implementations are more mature, we consider them as the DLT for the implementation of DL-Tags solution. Bitcoin [6] and Ethereum [7] are an example of stable reference blockchain implementations. New blockchain solutions occur such as EOS which is still in the development phase [10][11]. It is yet unstable and without adequate tooling.

Blockchain consists of ordered units called blocks. Blocks contain headers and transactions, as shown in Figure 1. Each block header, amongst other metadata, contains a reference to its predecessor in the form of the predecessor's hash. The initial state is hard-coded in the first block called the genesis block. Unlike other blocks, the genesis block has no predecessor.

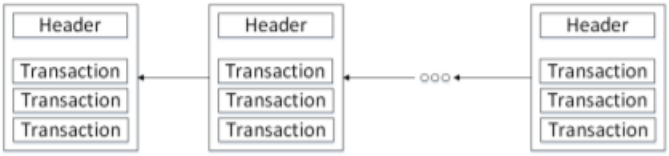


**Figure 1 Blockchain as a data structure**

Blockchain implementations can be public (permissionless) or private (permissioned). In public and permissionless blockchain each node can read from the ledger and append to the ledger. In private and permissioned blockchain direct access to blockchain data and transactions submission is limited to a predefined list of entities [9]. Ledger maintenance in private blockchain is performed by a group of entities. Therefore, a user needs to trust these entities to behave honestly, which makes private blockchain environment trusted. In this project we require all the stakeholders and consumers of a certain product to be able to read blockchain entries. Since the number of end consumers of a certain product type could be quite large, creating a predefined list of entities would be unattainable. Furthermore, we strive to develop the solution in a trustless environment eliminating the need for a trusted third party. Therefore, we consider only public blockchain for the needs of DL-Tags project. Bitcoin and Ethereum are both public blockchains.

Blocks in a public and permissionless blockchain can be malicious and cannot be implicitly trusted. Consequently, consensus about each entry needs to be reached in the network. This consensus is basically an agreement that is to be appended to the ledger by all nodes to confirm the validity of the transaction. The assumption is that a majority of nodes are honest and reliable. Both Bitcoin [6] and Ethereum [8] are based on a lottery function called the Proof of Work (PoW). The elected leader broadcasts the new entry to the rest of the participants who implicitly vote to accept the entry by adding it to their local copy of the ledger, and may propose subsequent transaction entries that build on the ledger.

The leader in the network becomes the first participant to successfully solve a cryptographic puzzle. For example, Bitcoin uses partial hash inversion as the cryptographic puzzle function. Partial hash inversion requires that the hash of a block of transactions together with a nonce (a free variable in the function) matches a certain pattern. The pattern starts with at least a predefined number of 0 bits. The function is intentionally difficult to solve since to manipulate the ledger, an attacker would need to have the supermajority of the computing power in the network, which makes an attack expensive to perform. Nodes that generate blocks in a PoW driven systems are called miners and the process is called mining. For the use of their resources, miners are granted tokens in the network as an economic incentive to mine (e.g. Ether in Ethereum, Bitcoin in Bitcoin). If there are no miners, no blocks can be mined and there is no transaction throughput.

Blockchain implementation Ethereum [7][8] enables holding arbitrary data by offering a possibility to implement *smart contracts*, distributed applications executed on the blockchain that support customized data storage and arbitrary business logic based on user needs. Smart contracts expand the blockchain potential to become a platform rather than only a cryptocurrency. Since customized data storage will be needed in the DL-Tags solution, Ethereum is chosen as the blockchain implementation.

One of the most relevant issues hindering global scale DLT adoption is its scalability. A block is created every time a PoW puzzle is solved, thus transaction rate is limited by the periodicity at which blocks are created and also by the block size. When increasing the number of nodes in the system, the frequency of block creation does not increase significantly due to the fact that the PoW puzzle difficulty is dynamic to enable converging of block generation time to a fixed value. In Ethereum, a block is mined roughly every 15 seconds with a dynamic block size not measured in bytes but in gas. Gas is the unit used to measure the fees required for a particular computation [8]. This value is dynamic and will adapt to network conditions. This feature enables Ethereum's transaction rate to be roughly between 7 and 15 transactions per seconds. Storing information to Ethereum blockchain can last for a couple of minutes, while reading from the blockchain is executed in real time [4]. End consumers in DL-Tags solution will only need to read information from the blockchain, which can be done in real time and free of charge, without deteriorating user experience.

## Section 3 - DL-Tags

DL-Tags project offers a decentralized, privacy-preserving and verifiable management of Smart Tags issued by TIS. It designs and develops a solution where Smart Tags information is shared directly between the involved stakeholders, while relying on the interoperable infrastructure provided by the TIS system. DLT is used as a trusted intermediary mediating the data exchange process between involved stakeholders, while ensuring data authenticity and integrity. Each interaction between stakeholders during the product exchange process needs to be stored on the blockchain. By reaching a consensus on the product authenticity, stakeholders cannot blame previous product owners for undesired information available through Smart Tags. Privacy is ensured by not storing human-readable information about the products on a public blockchain, but only product information digest created by applying cryptographic hash function. A blockchain entry is pseudo-anonymous, it discloses the identifier of the entity responsible for creation of the entry, but the entry itself cannot be deciphered.

The goal of DL-Tags solution is to dissolve situations where any of the involved stakeholders may attempt to attack the TIS system by propagating fraudulent or damaged products under false pretences. One particular situation that might arise is when a fraudulent stakeholder tries to modify a QR code and misdirect the future product users to a wrong URL. The incentive for such behaviour might be trying to sell a cheaper product as a more expensive, branded one. This attack has also been identified by the TIS consortium [3]. Figure 2 shows how DL-Tags can identify and dissolve this situation.

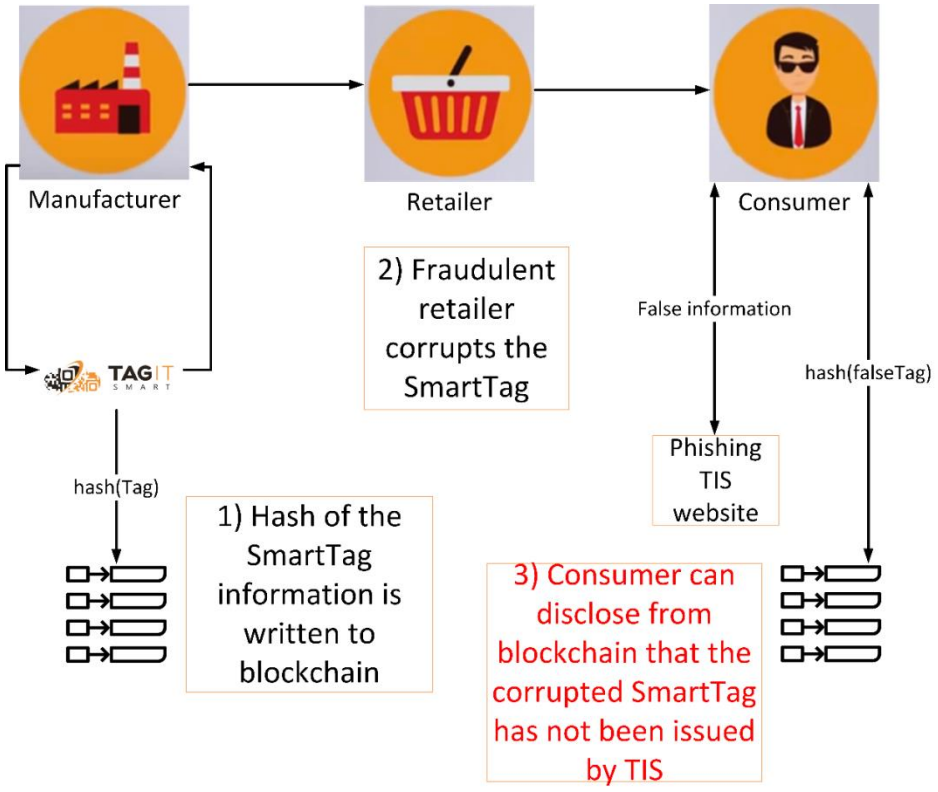


**Figure 2 Prevention of phishing site deployment**

In a regular scenario for using Smart Tags, manufacturer requires TIS system to issue a Smart Tag according to the provided product description. In the DL-Tags enhanced scenario, information about the Smart Tag creation is written to blockchain. A fraudulent retailer might try to modify the Smart Tag in order to sell the product as a branded and more expensive one and point to a phishing TIS site set up to provide deceitful information about the product. Without using the DL-Tags solution, consumer would scan the QR code and then be redirected to the phishing site which would provide false information. However, by using DL-Tags solution, consumer can perceive by analysing blockchain entries that TIS has not recorded the issuing of this particular Smart Tag. Consequently, the consumer can conclude that the Smart Tag (i.e. product description) has been manipulated during its lifecycle.

Another situation which might arise during the product lifecycle is the duplication of Smart Tags. Fraudulent retailer might want to duplicate Smart Tags of an expensive product and put it on cheaper, lower quality products, as shown in Figure 3. Without using the DL-Tags solution, this fraudulent retailer would be able to distribute these products to other stakeholders or end consumers. Succeeding stakeholders or product consumers might eventually realize the potential problem by detecting multiple checks for this particular product from TIS system. However, it would be difficult to verify which stakeholder committed the deceit. This kind of fraud would not be possible by using DL-Tags solution. In DL-Tags stakeholders are enforced to disown the product when selling it. Therefore, they would not be able to forward products with duplicated Smart Tags regularly, since they would not be their owners anymore.

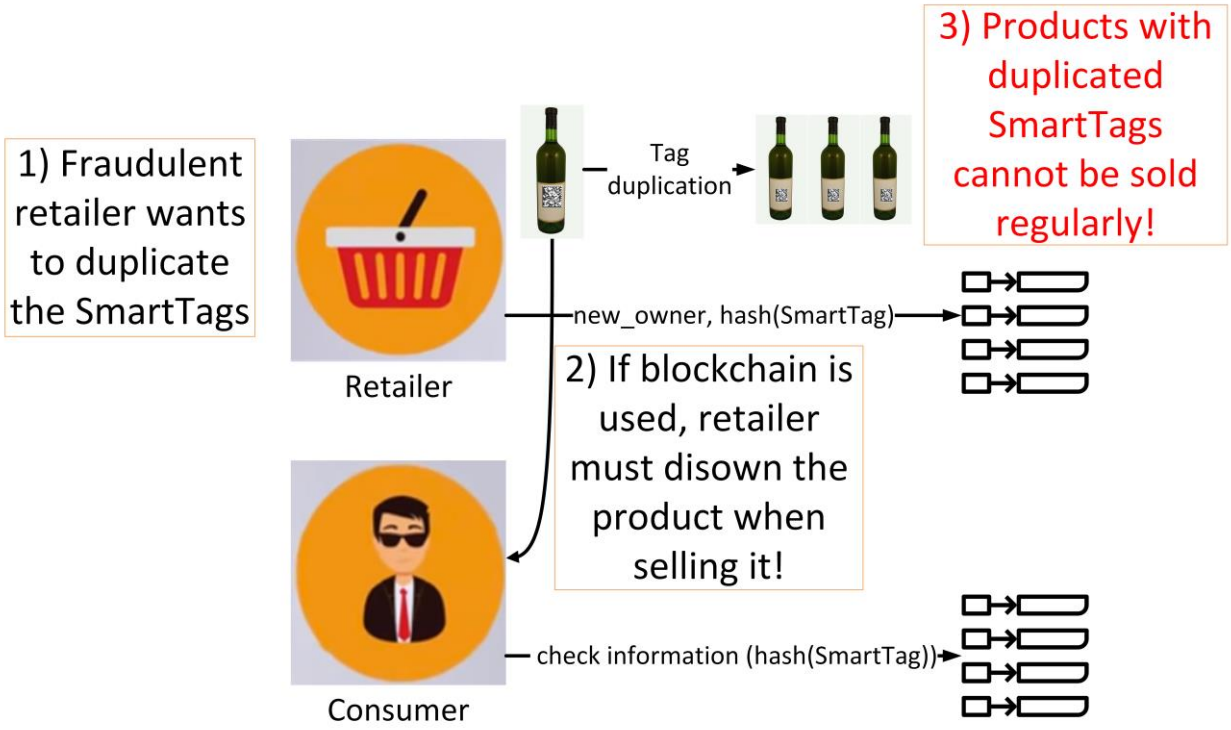


**Figure 3 Prevention of Smart Tag duplication**

Apart from phishing site deployment and Smart Tag duplication, a couple of other problems are identified for which DL-Tags offers a straightforward solution. A stakeholder might sell a

product to the next stakeholder, but without forwarding it. Afterwards, the buying stakeholder would end up without the bought product, while the selling stakeholder could assert having forwarded the product and accusing the buying stakeholder of deception. DL-Tags solution precludes such situations since the selling stakeholder needs to write on the blockchain that the product is sold, and receiving shareholder needs to confirm the receipt of the product. Additional problem might arise when a phishing TIS site is deployed and tries to issue Smart Tags with false information. This situation can also be resolved by using DL-Tags solution since blockchain entry upon product creation will not be made by using TIS *id*.

DL-Tags solution requires public and permissionless DLT which supports Smart Contracts. Smart Contracts are basically a customized code executed on the blockchain with a functionality needed for a specific application. In the proof-of-concept implementation Ethereum is used as the only stable solution which currently satisfies the listed requirements, but any other ledger technology that shares those properties can be used instead of Ethereum, which is known to exhibit scalability problems. Smart Contracts in Ethereum are usually referred to as Distributed Applications (DApp). Private chains are disregarded from consideration since they require authentication of all users, which might not be appropriate for TIS use cases with a large number of end consumers. Furthermore, private chains are implicitly trusted environments, while the focus of this project is on trustless environments.

The rest of this Section is organized as follows. System requirements are listed in Section 3.1, while sequence diagram showing the main functionalities of the system is depicted in Section 3.2.

## 3.1 System requirements

Requirements for the DL-Tags system are the following:

- The solution should be blockchain platform agnostic; various blockchain platforms should be able to be used with minor adjustments.
- Creation of Smart Tags by TIS platform must be stored on the blockchain.
- After stakeholders sell the product, they must disown it.
- Right after stakeholders receive the product, they must confirm its authenticity and record it on the blockchain.
- Each stakeholder should be able to unsubscribe to the information about the product after disowning it.
- Each stakeholder and consumer must be able to list all the blockchain entries for a specific product.

## 3.2 Sequence diagram

DL-Tags system needs to enable storing proofs of the transactions on the public blockchain. TIS must be able to store on the blockchain the proofs of the newly created Smart Tags. Each stakeholder needs to disown the product when delivering it to the following stakeholder in the product lifecycle, and each stakeholder needs to confirm the receipt of the product with the description kept in the Smart Tag. After disowning the product, stakeholders receive information about the new stakeholders and other events in this product's lifecycle. Stakeholders need an option to unsubscribe and to subscribe again to such information. Interactions with blockchain for the stakeholders in the product lifecycle are presented in Figure 4.

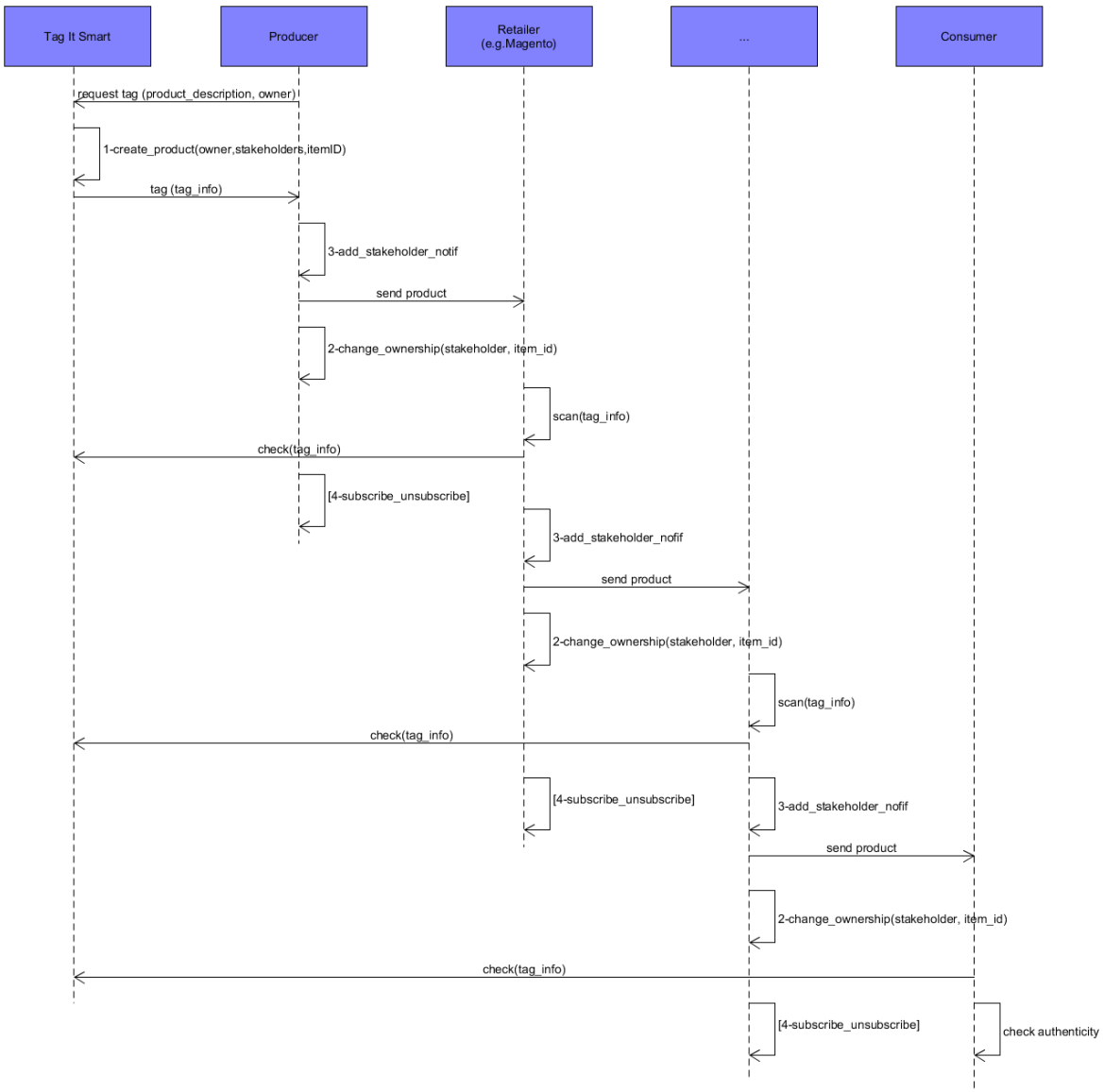


**Figure 4 Interactions between stakeholders**

**Description of the sequence diagram:**
In order to enable product tracking from the beginning of the product lifecycle, producer needs to issue a request to TIS for Smart Tag creation (message request tag). TIS needs to store this transaction on blockchain (message 1-create_product). This step practically handicaps the fraudulent stakeholder to issue false Smart Tags since it cannot store tag creation data with *id* and private key of the TIS system. Even if fraudulent stakeholder does issue a Smart Tag that might disclose false information, this would be easily observable on the blockchain.

When a producer wants to pass the product to another stakeholder (e.g. directly to retailer, to a shipping company, etc.), it is needed to write this information to blockchain along with the

*id* of the next stakeholder in the product lifecycle (message 2-change_ownership). Upon receiving the product, new owner gets information from the blockchain that he has been made the owner of the product (message 3-add_stakeholder). New owner can scan the Smart Tag and check if the provided information corresponds to the information stored on the blockchain. If it does not, new owner can check the previous records on the blockchain regarding this particular product and ascertain which stakeholder has released the product with a different Smart Tag.
All the stakeholders during product lifecycle need to store change ownership information to the blockchain when disowning the product, as well as confirmations that the Smart Tag information corresponds to blockchain entries upon receiving the product. Stakeholders also have the option to subscribe/unsubscribe to information about the products they disowned (message 4-subscribe_unsubscribe). By having this option, stakeholder can get notifications about new stakeholders of the products they managed.

Consumers buy the product and consume it. They have the option to check all the previous transactions from the blockchain to see the stakeholders in the product lifecycle. This process should be transparent to the consumer, and application should handle all the checks. Consumers do not need to confirm ownership of the product. Upon receiving and scanning the product, user application should check the blockchain entry for this product and return validation result, i.e., if the Smart Tag information has been stored on blockchain by previous stakeholders. If not, consumers become aware of the irregularities during the product lifecycle and refuse to take product into their possession.

## Section 4 - DL-Tags Architecture

Each stakeholder (referred to as *Platform* in the following diagrams) needs to communicate with the blockchain. In our solution we introduce a *DL-Tags Proxy* component responsible for handling all the interactions with the blockchain so that the stakeholders do not have to manage blockchain related messages on their own. Instead, stakeholders use proxy's interface implemented as JSON RPC to pass the data that needs to be stored on the blockchain.

### 4.1 Components

The components and their interfaces are shown in Figure 5. *DL-Tags Proxy* component developed within the project specifies an interface to receive platform data, while AMQP is used to forward information to Platforms. Interaction with the blockchain is technology specific, in our proof-of-concept *Blockchain Distributed Application*, shortly *Blockchain DApp*, also referred to as *Smart Contract*, is used to interact with Ethereum blockchain. However, the messages on *DL-Tags Proxy* interface will be formalized, and should easily be adapted for another public blockchain.

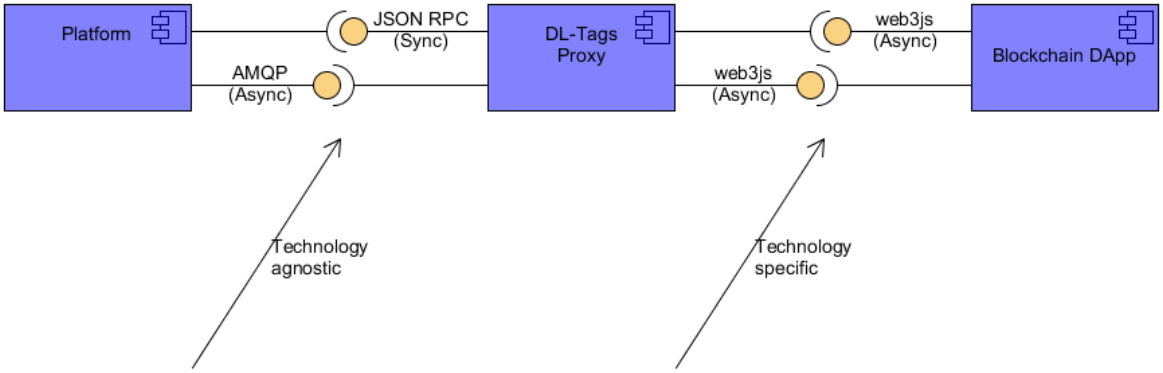


**Figure 5 DLT-Tags components**

### 4.2 Functionalities

Interactions with the blockchain shown in Figure 4 are presented in more detail in this Section by introducing interactions between all the components of the DL-Tags ecosystem. These functionalities include product creation (message 1-create_product from Figure 4), changing the owner (message 2-change_ownership from Figure 4), receiving added stakeholder notification (message 3-add_stakeholder from Figure 4), and subscribing or unsubscribing to product information (message 4-subscribe_unsubscribe from Figure 4).

*DL-Tags Proxy* implements the interfaces for the execution of the aforementioned functionalities. DApp implements the same functionalities as *DL-Tags Proxy* on the

blockchain and ensures that all the transactions related to product registration, ownership changes, and confirmations, are written on the blockchain. Furthermore, it initiates announcements to the stakeholders when necessary, e.g., when a stakeholder is added as a new product owner.

### *4.2.1 Create product*

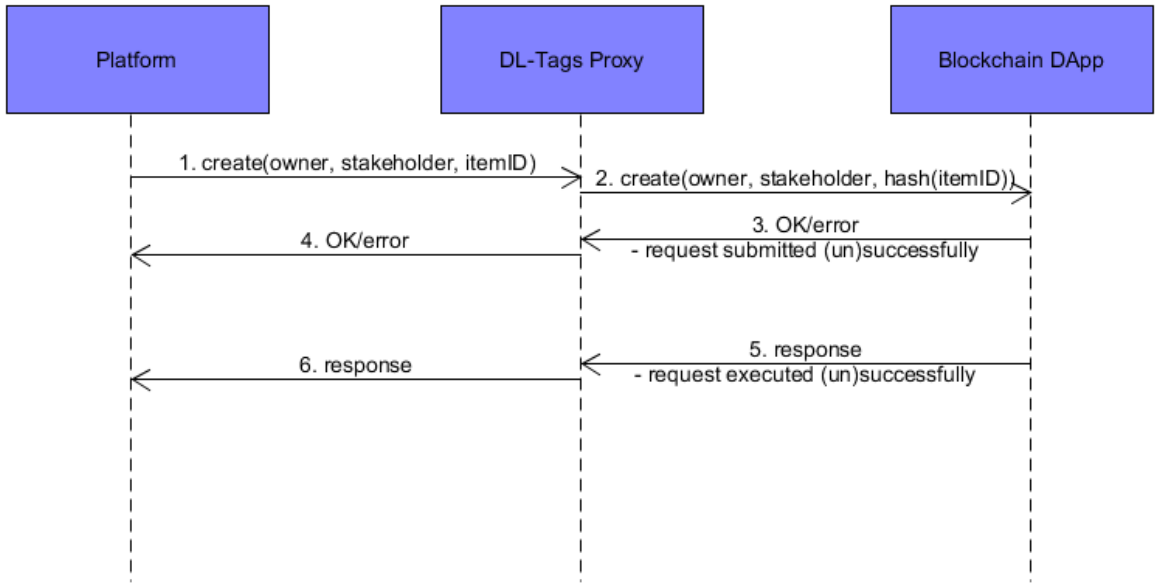


**Figure 6 Create product**

**Description:**
This procedure is in DL-Tags scenario executed by TIS platform, i.e., *Platform* in Figure 6 is TIS. The procedure is initiated after TIS generates a Smart Tag based on the request from product manufacturer.

1. *Platform* sends a create product request to the *DL-Tags Proxy*. Product producer *id* is inscribed in the *owner* field, while TIS *id* is inscribed in the *stakeholder* field.
2. Message generated by the *Platform* is forwarded by the *DL-Tags Proxy* to *Blockchain DApp*.
3. *Blockchain DApp* confirms the (un)successful receipt of the request.
4. Request status is forwarded back to the *Platform.*
5. *Blockchain DApp* provides status of the request execution.
6. Request execution status is forwarded back to the *Platform.*

### 4.2.2 Change owner

**Figure 7 Change owner**

**Description:**
This procedure is executed by a stakeholder who has possession of a product before forwarding it to the next stakeholder in the product lifecycle.

1. *Platform* sends add new stakeholder request to *DL-Tags Proxy*. The *id* of the new stakeholder is inscribed in the *stakeholder* field.
2. Message generated by the *Platform* is forwarded by the *DL-Tags Proxy* component to *Blockchain DApp*.
3. *Blockchain DApp* confirms the (un)successful receipt of the request.
4. Request status is forwarded back to the *Platform.*
5. *Blockchain DApp* provides status of the request execution.
6. Request execution status is forwarded back to the *Platform.*
7. After successfully adding a new stakeholder, former product owner disowns a product by sending the *changeOwner* message. *Owner* field in this message is the *id* of the new stakeholder added by exchanging messages 1-6.
8. Message generated by the *Platform* is forwarded by the *DL-Tags Proxy* component to *Blockchain DApp*.
9. *Blockchain DApp* confirms the (un)successful receipt of the request.
10. Request status is forwarded back to the *Platform.*
11. *Blockchain DApp* provides status of the request execution.
12. Request execution status is forwarded back to the *Platform.*

### 4.2.3 *Added stakeholder notification and reaching a consensus about product authenticity*

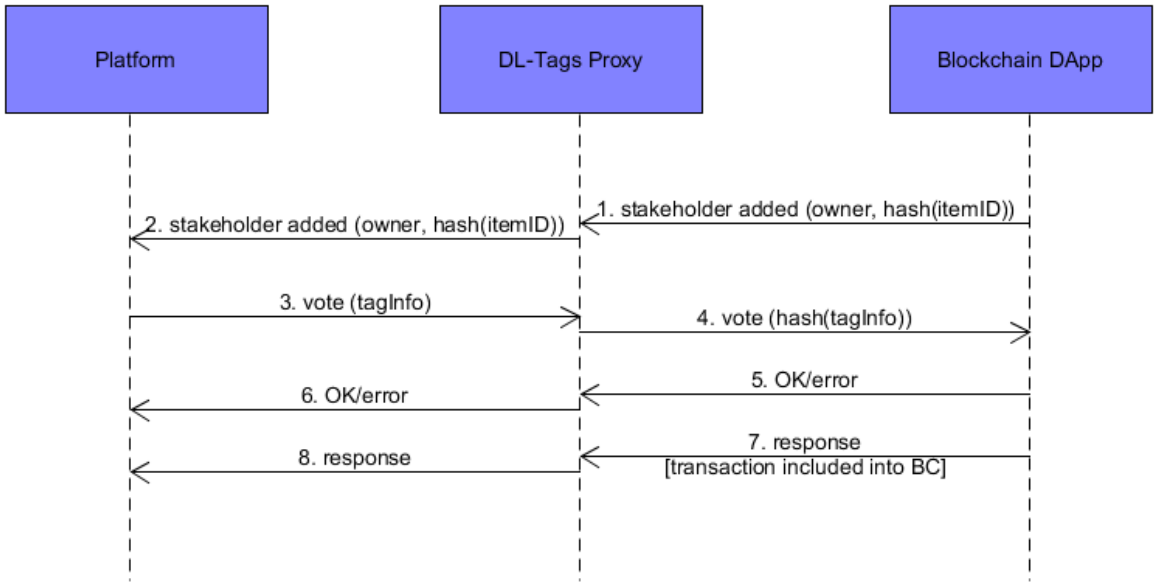


**Figure 8 Added stakeholder notification**

**Description:**
This procedure is initiated by the blockchain after a stakeholder is added to a product and after this stakeholder has been named as product owner. The stakeholder is notified about this event. Upon receiving the physical product, the stakeholder needs to confirm that information inscribed in the Smart Tag corresponds to the product information. By using this mechanism, stakeholders reach a consensus about product authenticity.

1. *Blockchain DApp* sends *stakeholder added* message to the *DL-Tags Proxy* of the corresponding stakeholder.
2. Message generated by the *Blockchain DApp* is forwarded by the *DL-Tags Proxy* component to *Platform*.
3. *Platform* votes on the validity of the Smart Tag, i.e., that the Smart Tag information written on the blockchain corresponds or not to the Smart Tag information attached to the physical product and provided by TIS. Voting option needs to be offered within a stakeholder application. Stakeholder can vote only once, and needs to vote before disowning the product.
4. Voting result is forwarded to *Blockchain DApp.*
5. *Blockchain DApp* confirms the (un)successful receipt of the voting result
6. Voting result status is forwarded back to the *Platform.*
7. *Blockchain DApp* provides status of the voting execution, i.e., if the voting information has been recorded on the blockchain.
8. Voting execution status is forwarded back to the *Platform.*

### 4.2.4 Subscribe/unsubscribe to notifications

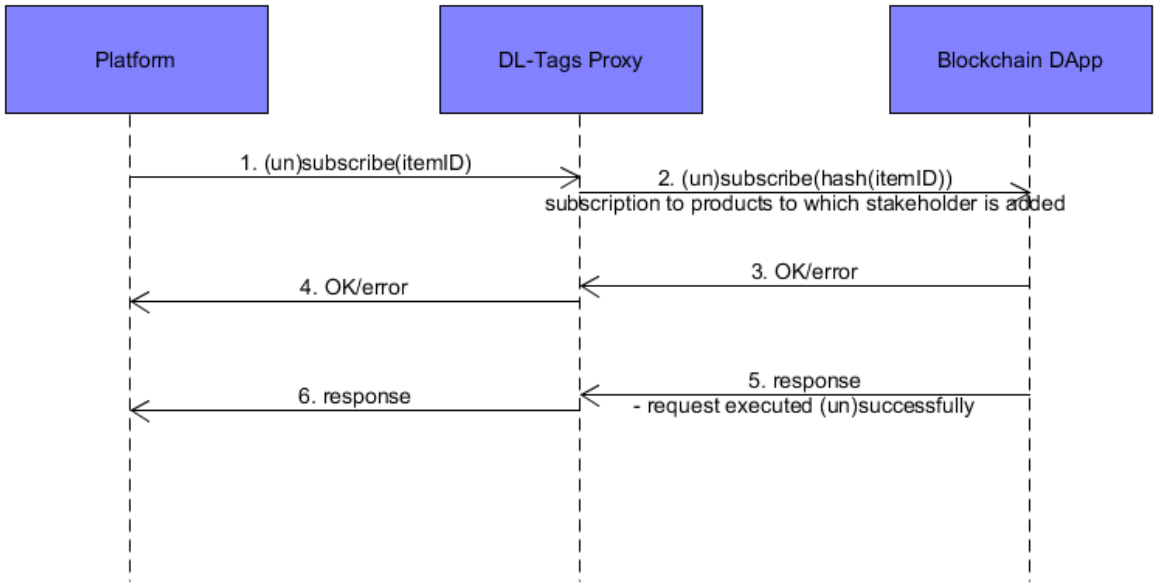


**Figure 9 Subscribe/unsubscribe to notifications**

**Description:**
Each product stakeholder can receive updates related to the product, e.g., when a new stakeholder is added, when product checks are initiated, etc. The stakeholder is subscribed by default to receiving product information. During the product lifecycle, each stakeholder can unsubscribe, or subscribe again to receive notifications about the product identified by the Smart Tag.

1. *Platform* sends (un)subscribe message with *itemID* to its *DL-Tags Proxy*.
2. Message generated by the *Platform* is forwarded by the *DL-Tags Proxy* component to *Blockchain DApp*.
3. *Blockchain* confirms the (un)successful receipt of the (un)subscribe request.
4. Request status is forwarded back to the *Platform.*
5. *Blockchain* provides status of the request execution, i.e., if the (un)subscription has been recorded on the blockchain.
6. Request execution status is forwarded back to the *Platform.*

## 4.3 Implementation plans

Planned proof-of-concept for the DL-Tags solution will consist of the following stakeholders, as shown Figure 10:

- Product Manufacturer,
- TIS,
- Magento e-commerce store, and
- Consumer.

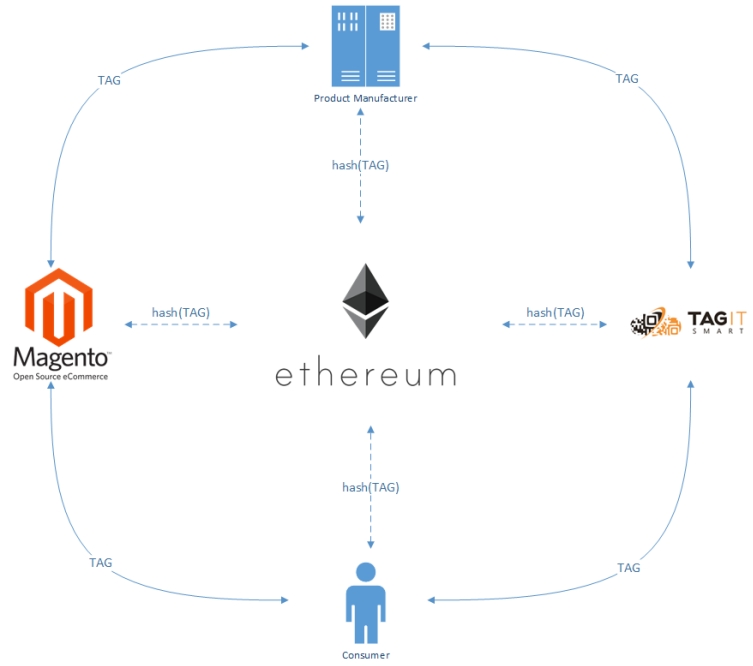


**Figure 10 DL-Tags solution**

The planned proof-of-concept is envisioned as follows. Product manufacturer produces a certain product and wants to distribute it through Magento e-commerce store that acts as a retailer. Magento e-commerce store in this proof-of-concept represents a stakeholder (it will be referred to as Magento stakeholder hereafter) responsible for collecting the physical product from the manufacturer, offering it through the on-line store, storing the product if necessary, and delivering it to the end consumer after the purchase. Consumer uses an application that can verify the originality of the product.

All the products need to be tagged by Smart Tags generated by TIS platform. Product manufacturer acquires the Smart Tag for each product by using TIS platform. Product is created on the blockchain by TIS platform according to the procedure described in Section 4.2.1. Product manufacturer then changes the owner of the product to the Magento stakeholder according to the procedure described in Section 4.2.2, and passes the physical product afterwards. After receiving the physical product, Magento stakeholder scans the Smart Tag, confirms the information written on the Smart Tag, and accepts taking over the ownership of the product according to the procedure described in Section 4.2.3. When a consumer buys the product, Magento stakeholder should disown it according to the procedure from Section 4.2.2. Consumer can verify the originality of the product by scanning the Smart Tag and contacting the blockchain to ascertain that both versions match. This verification procedure is done automatically, without user involvement.

Each stakeholder needs to run its own *DL-Tags Proxy* to communicate with the *Blockchain DApp*. To be able to communicate with the blockchain, each stakeholder needs to download the blockchain and synchronize with the network. Additionally, each stakeholder needs to implement a *Platform plugin* responsible for forwarding product-related events to the blockchain, and to receive notifications from the *DL-Tags Proxy* and the blockchain.

Product manufacturer application will be implemented as a process offering the possibility to contact TIS for a Smart Tag creation, and to forward the product to the next stakeholder, i.e., to the Magento e-commerce application.
TIS platform will be integrated with *DL-Tags Proxy* to create product on the blockchain after a Smart Tag creation.
Magento e-commerce store application will be deployed with a main functionality to offer products to costomers. It will need to be able to confirm product receipt and to disown the product after selling it.
Consumer application will be integrated with the existing TagItWine application already communicating with the TIS system to verify Smart Tag information. New functionality will be communication with the blockchain, i.e., with the *DL-Tags Proxy* that will contact the blockchain.

## Section 5 - Conclusions

DL-Tags solution enhances the existing TIS system by ensuring decentralized, privacy-preserving and verifiable management of Smart Tags directly between involved stakeholders, without the need of a third party storing transactions between stakeholders in the product lifecycle. Several frauds, such as deploying a phishing TIS site or Smart Tags duplication, can be avoided by using the DL-Tags solution. This deliverable presents the main features, requirements, and initial architecture of the DL-Tags system.

The architecture of the system consists of the *DL-Tags Proxy*, plugins for the *Platforms,* i.e. the stakeholders in the product lifecycle, and *Blockchain Distributed Application* running on Ethereum blockchain and responsible for storing and viewing all transaction related information.

Proof-of-concept will be implemented by integrating the *DL-Tags Proxy* with several stakeholders: web application for product manufacturer to issue a Smart Tag, TIS system, Magento e-commerce store, and mobile application for verifying Smart Tags developed within TagItWine project. This deployment enables the management of product descriptions through the TIS ecosystem enhanced by DL-Tags. It will demonstrate interactions between involved stakeholders facilitated by TIS information model and relevant components, as well as the applicability of DL-Tags solution in real-world scenario. Furthermore, it will give the opportunity for the DL-Tags solution to be tested and to investigate its potential to be applied in the realization of European Product Passport (EPP).

Implementation details related to *DL-Tags Proxy* and *Blockchain DApp* for Ethereum will be provided in Deliverable D2.1 DLT-Tags implementation. Integration with *Magento* will be described in Deliverable D2.2 Magento integration.

## Section 6 - References


[1] G. Zyskind, O. Nathan and A. Pentland, *Decentralizing Privacy: Using Blockchain to Protect Personal Data*, 2015 IEEE Security and Privacy Workshops, San Jose, CA, 2015, pp. 180-184.

[2] O. Svein, *Electronic Government*, vol. 9820, 2016, pp. 253-264, available online: http://link.springer.com/10.1007/978-3-319-44421-5

[3] TagItSmart consortium, *D2.3 - Final Enabler for FunCodes*, February 2018

[4] F.M. Bencic, I. Podnar Zarko, *Distributed Ledger Technology: Blockchain Compared to Directed Acyclic Graph*, 2018, available online: https://arxiv.org/pdf/1804.10013.pdf

[5] A. Narayanan, and J. Clark, *Bitcoin's academic pedigree,* Communications of the ACM, vol. 60, no. 12, 2017, pp. 36-45.

[6] P. Franco, *The Blockchain*, Understanding Bitcoin, 2014, pp. 95–122, available online: http://dx.doi.org/10.1002/9781119019138.ch7

[7] V. Buterin, *A next-generation smart contract and decentralized application platform,* Etherum, no. 1, 2014, pp. 1–36, available online: http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf

[8] G. Wood, *Ethereum: a secure decentralised generalised transaction ledger,* Ethereum Project Yellow Paper, 2014, pp. 1–32.

[9] BitFury Group and J. Garzik, *Public versus Private Blockchains. Part 1: Permissionless Blockchains,* 2015, pp. 1–23, available online: http://bitfury.com/content/5-white-papers-research/ public-vs-private-pt1-1.pdf

[10] *EOS.IO Technical White Paper v2,* March 2018, available online: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[11] EOS Mainnet Launch Group, *EOS Mainnet Launch: Security & Software Testing*, June 2018, available online: https://medium.com/@blockgenic/disclaimer-as-a-token-holder-it-is-important-that-you-do-nothing-during-this-period-as-block-42c27f9afa54