TagItSmart
Smart tags driven service platform for enabling ecosystems of connected objects

Grant agreement 688061

Open Call Deliverable Report for

TAGITWINE
PILOT FOR BRAND PROTECTION AND ANTI-COUNTERFEITING IN WINE INDUSTRY:

– System Architecture and Design –

Deliverable ID:	**D 1.1**
Deliverable Title:	**Pilot System Architecture and Design**
Revision #:	**1.0**
Dissemination Level:	TIS Consortium
Responsible beneficiary:	Durst
Contributing beneficiaries:	University of Donja Gorica
Contractual date of delivery:	31.10.2017
Actual submission date:	30.10.2017

Start Date of the Sub-Project: September 1st, 2017 Duration: 12 Months

Contents

Introduction *4*
1.1 Purpose of This Document 4
1.2 Contributors 4
1.3 Acknowledgment 4
Section 2 - Context *5*
2.1 Objectives 5
2.2 Approach to Implementation 5
Section 3 - Functional View *8*
3.1 Stakeholders 8
3.2 Main Actors 8
3.3 Use Cases 8
Section 4 - Logical View *16*
Section 5 - Process View *17*
Section 6 - Implementation View *18*
Section 7 - Deployment View *19*
Section 8 - Interface View *20*
8.1 TagItWine specific messages 20
8.2 TagItSmart HTTP API 21
8.3 Winemaker Web Application User Interface 22
8.4 Mobile Application GUI 24
Section 9 - Data View *26*
Section 10 - Non-functional view *31*
10.1 Technology selection 31
10.2 Security View 31
10.3 Network Requirements 31
Section 11 - Principles *32*
Section 12 - Conclusion *33*
List of Abbreviations *34*
Appendix *35*

	Name	Organization	Date	Visa
Project Management Team				

Document history

Revision	Date	Modification	Authors
v0.1	15.9.2017	Document outline proposd	Tomo
v0.2	10.10.2017	Content development	Tomo, Stevan
v0.3	10.10.2017	Content update	Marko, Biljana
v0.4	10.10.2017	Template of the deliverable	Ingmar
v0.5	20.10.2017	Content transferred into the template	Stevan, Tomo
v0.6	25.10.2017	Addl. input from Plantaze	Tomo, Stevan
v0.7	26.10.2017	Data view update	Stevan, Biljana
v0.8	27.10.2017	Internal review	All
v1.0	30.10.2017	Review and approval, submission	Tomo

Introduction

The objective of this pilot project is to use the TagItSmart (TIS) platform to implement and execute a fully functional pilot for brand protection and anti-counterfeiting in wine industry and to test the solution in collaboration with the company “13. jul Plantaze” (henceforth “Plantaze”). We foresee the use of various TIS architectural components supported by QR codes, NFCs and SmartTags used on wine bottles, boxes and pallets throughout the wine supply chain. It is expected that the brand protection pilot will include some functionality from digital product and lifecycle management use cases of the TIS project.

Plantaze is the biggest wine producer in Montenegro and owns the largest vineyard in one complex in Europe. The company is a market leader in the region of Southeast Europe in the segment of top quality wines and the biggest ex-Yugo exporter, which can be illustrated by annual sales of 17 million bottled products in 35 countries, out of which 90% is wine. Wines and brandies prices range from 1,37 to 119,85 €. Majority of the sales takes place in the region (Serbia, Bosnia), domestic market of Montenegro, and then Russia and China. Plantaze has been experiencing problems with counterfeits, especially in Albania, Kosovo, and recently in Russia. UDG partnered with Plantaze in order to validate TIS solutions for brand protection use scenario and evaluate business value potential of TIS for wine industry.

1.1 Purpose of This Document

The purpose of this document is to identify and specify pilot system architecture and design for TagItWine project. The documents is organized to provide various architectural views and perspectives in order to better communicate with different stakeholders. The architectural views presented in this document are used to facilitate discussions and capture architectural requirements for the pilot implementation.

This document serves as the deliverable D1.1 for the TagItWine project, work package WP1.

1.2 Contributors

The main contributors to this document are:

- Tomo Popović, PhD
- Stevan Šandi, MSc
- Marko Simeunović, PhD
- Biljana Stamatović, PhD

1.3 Acknowledgment

This work has been supported through Open Call #1 of the TagItSmart! project titled “Smart Tags driven service platform for enabling ecosystems of connected objects”, grant agreement number 688061, TGS SEC /01/2017, under under the “Horizon 2020 – the Framework Programme for Research and Innovation (2014-2020)” of the European Commission.

The authors would like to thank Srdjan Krčo, Nebojša Stojanović, Vesna Maraš, Sanja Radonjić, Anita Gazivoda, Ratka Kalezić for the support and suggestions during the development of this specification.

Section 2 - Context

Brand protection in wine industry is especially important as wine counterfeiting, besides its impact to producer's reputation, affects health of consumers, tradition and economy of a region, or even the whole country. The most common way of wine counterfeiting is printing a fake label with subtle changes in brand name and company logo in order to fool wine consumers. Sometimes, counterfeiters use the authentic labels removed from more expensive wines and place them on cheaper wine with similar bottles.

The project TagItWine targets expansion of the TIS ecosystem by further customizing the technology and implementing a pilot system for the Brand protection use case in a large industrial company environment (Fig. 2.1). This project is fully aligned with the existing TIS use cases as it proposes to utilize some of Digital product and Lifecycle management functionalities and assumes development of new Brand protection use case functionalities using TIS tools, solutions and their corresponding application programming interfaces (APIs). New applications shall be integrated into a fully functional wine anti-counterfeiting system. The pilot shall be installed and executed within the wine supply chain of one of the largest wine producers in Southeast Europe – Plantaze.

Figure 2.1. Context view of the TagItWine pilot project.

2.1 Objectives

The main objectives of the TagItWine pilot is to evaluate the use of TIS solutions and Smart Tags for brand protection and anti-counterfeiting in wine industry. The desired outcome of the pilot can be summarized into two categories:

- Technical analysis of the used TIS components, performance of the selected technology, and assessment of what worked well and what can be improved;
- Business evaluation of the potential of TIS solutions for brand protection in wine industry and industry in general; projections of economic benefits for Plantaze and similar companies.

2.2 Approach to Implementation

The main assumption is that components of TIS platform will functionally cover most of the implementation of brand protection pilot. Different approach is be considered for expensive and less expensive wine bottles. In both cases, we aim for a creation of "smart" bottles of

wine that can be traced and controlled through the wine supply chain and during their lifecycle. Smart bottles enable the creation of both producer and consumer oriented services based on the information that has been generated at different stages of the lifecycle in order to enhance anti-counterfeiting and brand protection.

The solution shall provide for wine authenticity control, as well as two-way communication and personalized experience for the consumer.

Figure 2.2. TagItWine: Concept and approach.

The use of QR codes supported by the use of functional inks and potentially combined with printed NFC tags and sensors will play the key role in “smartifying” wine bottles, boxes, and pallets. Smart tags and their features to change based on environmental (light, temperature) and time elapse conditions are especially important for protecting and controlling boxes and pallets during transport. The software part will consist of front-end mobile device application and a back-end cloud solution that integrates existing and/or customized TIS functionalities.

Figure 2.3. QR code in normal and active state.
Source: TagItSmart Open Call Announcement #1.

Each bottle will be digitized by assignment of a unique smart tag. The authenticity of the tag will rely on the capabilities of functional inks. For this pilot solution, the emphasis will be on photochromic (light sensitive) inks. Figure 2.3 illustrates the main concept of utilization of photochromic inks. QR codes printed with photochromic inks will have two states, normal

and active. Active state of the QR code will be achieved by illuminating the QR code in normal state with LED light (i.e. mobile phone flash light). That is how additional information can be carried within the same QR code based tag. After the source of the illumination is gone QR code will revert to normal state in short time (reversible ink).

Photochromic ink with non-reversible active state is considered for marking the bottle of wine as consumed to prevent refilling the authentic bottle. The non-reversible photochromatic ink is considered for box/pallets tagging, as well.

Figure 2.4. TagItWine: How does it work?

Figure 2.4 illustrates the usage of TagItWine solution. Conceptually, main functions of the system are:

- Consumer/customer scans the product and obtains information on that particular instance (i.e. red dry, vintage), as well as the info on its authenticity; depending on the outcome confirmation or warning is issued to the user and winemaker is notified.
- Winemaker gets the status of the particular bottle (i.e. in stores, on table, opened); if opened and consumed, user can utilize the option to rate the wine and provide feedback to winemaker.

Section 3 - Functional View

This section introduces architecture-significant use cases and their relationship. It discusses why these use cases are important for the architecture, what the system actually does, and what the key functional users are.

3.1 Stakeholders

The following stakeholders have been identified that may benefit from using TagItWine solution:

- Wine Producers – to protect their brand, reduce losses, improve quality of the product and services.
- Distributors – to validate way bills and improve transportation.
- Retailers – to provide an attractive service to their shoppers and increase sales.
- Customers – to get better information on the product and make sure it is original.

3.2 Main Actors

There are four actors/roles anticipated for describing the usage scenarios:

- Winemaker,
- Consumer/customer,
- Waybill clerk at the distribution center,
- Cashier in a retails store.

3.3 Use Cases

Use case view, or scenarios, provide an architectural view relevant for the functionalities of the TagItWine platform. Platform functional requirements are shown in Figure 3.1. The main scenarios include:

- *Wine Bottle Authentication*, which includes providing *Wine Information* to the customer. This function can be extended with *Rate Product* in order to provide customer feedback to wine producers. Optionally, an *Extended User Experience* can be offered to customers by using the platform to implement text and/or multimedia messages similar to greeting cards (Happy Birthday, Anniversary, etc).
- *Product Tagging* takes place at the wine maker's site. It includes *box/pallet tagging* and *individual bottle tagging* functions. Expensive bottle tagging may include additional options.
- *Distribution Control* or waybill check is an optional function borrowed from the Lifecycle Management use case. The idea is to make sure that the pallets or boxes arrived at the distribution centers as planned.
- *Retail Control* in a form of sales confirmation can be used as additional source of information needed to mark the status of the bottle (on shelves in store, purchased, opened and consumed).

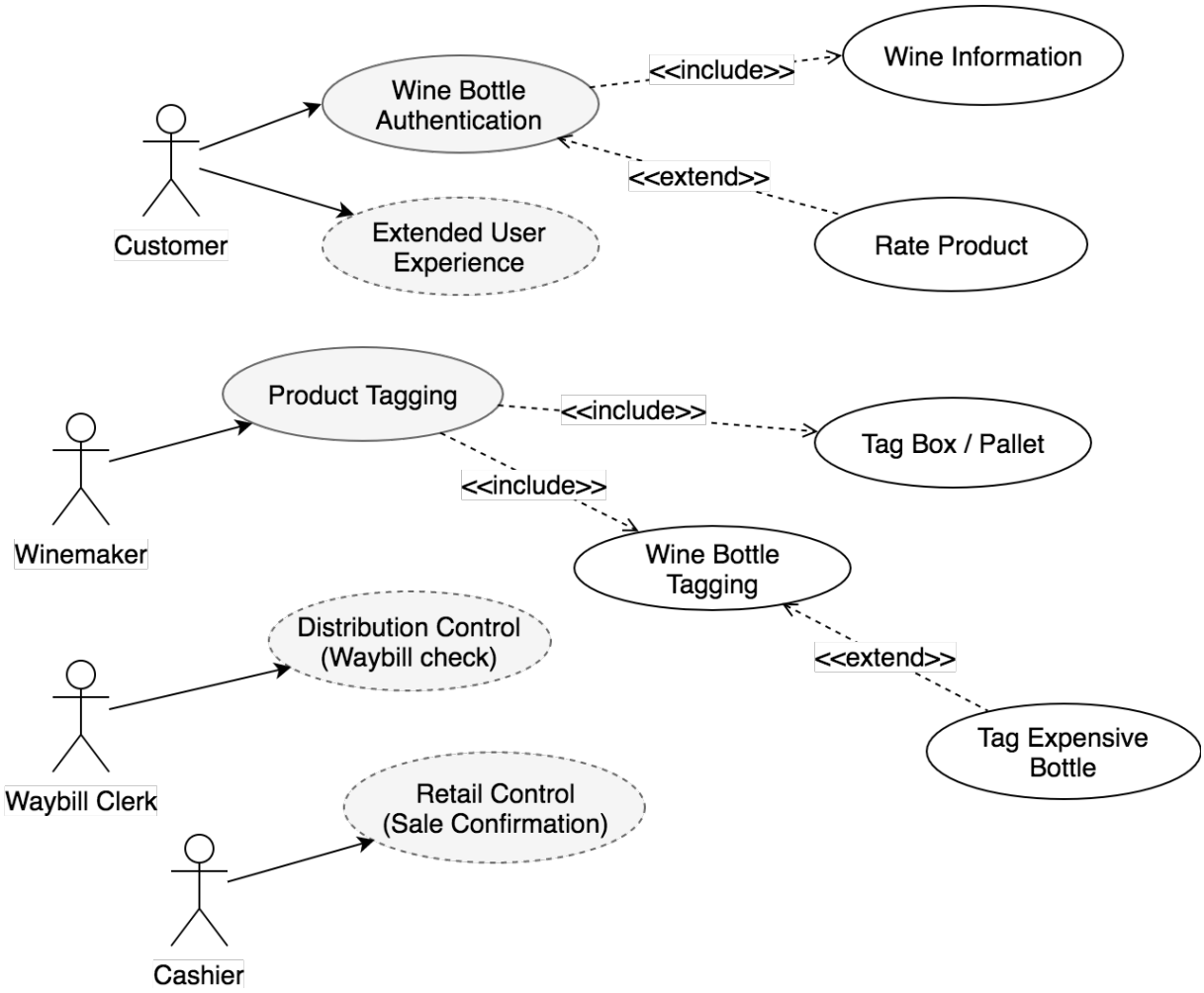

Figure 3.1. Functional view: TagItWine use cases.

Details of the each functionality identified in the use case diagram is elaborated in tables that follow.

Table 3.1. Wine Bottle Authentication use case.

UC1 – Wine Bottle Authentication (cont.)	
Description	Consumer must be able to check authenticity of each bottle of wine. By scanning the smart tag, consumer will, beside authenticity confirmation, retrieve info about the wine (vintage, region, etc.).
Priority	Essential
Used by	Consumer
Success end conditions	Scanned bottle of wine is authentic.
Failed end conditions	For specific bottle of wine cannot be said whether it is authentic or not.
Actors	Consumer, Winemaker
Trigger	Consumer decides to scan smart tag on bottle of wine.
Main Scenario	1. Consumer scans smart tag on the bottle of wine using customer mobile application. 2. Consumer selects whether the bottle of wine is on the shelf, purchased or already consumed. 3. Consumer gets information about the wine. 4. In case there is a problem with authenticity: 4a. Winemaker gets notified. 4b. Depending on the problem type consumer gets a warning.
Extensions	Customer must be able to give feedback about the consumed bottle of wine. 5. Consumer writes feedback using customer mobile app. 6. Consumer scans smart tag on the bottle of wine. 7. Consumer posts feedback for the scanned bottle of wine and marks it as consumed.

UC1 – Wine Bottle Authentication (cont.)	
Variations	*Smart tags in form of QR codes printed with photochromic ink.* 1a. Customer scans the QR code to retrieve information about the wine. 1b. Customer illuminates the QR code with LED light to make the photochromic ink to appear. 1c. Customer scans again the QR code now with functional ink visible. 1d. Customer receives info about the authenticity status. *Expensive bottle of wine* 1a. Customer scans the NFC tag with seal state status using customer mobile application. 1b. Customer receives additional info about the wine and whether bottle is opened or not.
Exceptions	1. Smart tag is nor present on bottle of wine. 2. Invisible ink not showing up after illumination. 3. Bottle of wine scanned in retail shop already marked as sold. (Possible case of refill fraud).
Info / Notes	
Open Issues	Possible reselling of bottle of wine by customer who purchased bottle of wine in retails shop.

Table 3.2. Extended user experience use case.

UC2 – Extended User Experience	
Description	Customers should be able to attach e-greeting card to smart tag. By pairing the e-greeting card (in form of text, photo or video message) with smart tag customer will be able to give a bottle of wine as a present with greeting card that can be previewed online by scanning the smart tag.
Priority	Optional
Used by	Customer
Success end conditions	E-greeting card has been attached and can be previewed by scanning the smart tag.
Failed end conditions	E-greeting card cannot be attached or cannot be previewed by scanning the smart tag.
Actors	Customer
Trigger	Customer decides to attach greeting message to a bottle of wine.
Main Scenario	1. Customer (sender) creates the e-greeting card. 2. Customer scans the smart tag to pair the e-greeting card with the bottle of wine. 3. Customer defines email of the end consumer (recipient) who will receive the bottle of wine as a gift. 4. Customer uploads e-greeting card to the TiW platform. 5. End consumer should create account with the email (defined in the step 3.) using customer mobile application. 6. End consumer scans smart tag on the received bottle of wine. 7. End consumer choses whether to preview the attached e-greeting card.
Extensions	
Variations	1a. Customer creates text e-greeting card using offered templates. 1b. Customer creates e-greeting card using photo from his own collection. 1c. Customer creates e-greeting card using video clip from his own collection.
Exceptions	1. Smart tag is not present on the bottle of wine. 2. E-greeting card cannot be uploaded to TiW platform.
Info / Notes	
Open Issues	

Table 3.3. Product Tagging use case.

UC3 – Product Tagging	
Description	Winemaker should be able to tag each bottle of wine with unique ID. Applied smart tags on bottles will provide authentication check for each bottle as well as additional information about wine.
Priority	Essential
Used by	Winemaker
Success end conditions	Every produced bottle has unique tag.
Failed end conditions	Every bottle does not have unique tag.
Actors	Winemaker
Trigger	Newly produced batch of bottles comes in.
Main Scenario	1. Winemaker creates info about the current batch (product type, production date, etc.) 2. Winemaker generates tags for current batch. 3. Winemaker applies generated tags on bottles.
Extensions	Expensive Wine Bottles with NFC tags 2a. Winemaker pairs NFC tag state values (opened/sealed) with each bottle from the batch. 3a. Winemaker applies NFC tag seals on each bottle.
Variations	
Exceptions	1. Smart tags cannot be generated. Use case failed. 2. NFC sealed state tags nonfunctional. Use case failed.
Info / Notes	
Open Issues	

Table 3.4. Distribution Control use case.

UC4 – Distribution Control	
Description	Distributors should be able to get package (box/pallet) info by scanning smart tags. If package content does not match with retrieved waybill specification alarm should be raised and Winemaker should further investigate this issue.
Priority	Optional
Used by	Distributor
Success end conditions	Retrieved waybill specification matches box/pallet content.
Failed end conditions	Waybill specification differs from box/pallet content.
Actors	Waybill clerk, Winemaker
Trigger	Ordered products arrive from winemaker.
Main Scenario	1. Waybill clerk scans smart tag on box/pallet with mobile app to retrieve waybill specification. 2. Waybill clerk checks whether box/pallet content matches with waybill specification. 3. If waybill specification matches with package content waybill clerk marks order as successfully received.
Extensions	
Variations	2a. Waybill specification might not match package content. Waybill clerk must inform winemaker about issue. Winemaker should further investigate this issue.
Exceptions	2a. Smart tag might not be present on box/pallet. Waybill clerk must inform winemaker about this issue. Winemaker should further investigate this issue. Use case failed.
Info / Notes	
Open Issues	

Table 3.5. Retail Control use case

UC5 – Retail Control	
Description	Retailer should be able to mark bottle as sold once it is purchased by customer. Winemakers will have real-time info about sold products. In case that already sold bottle should be flagged as sold once more an alarm should be raised and winemaker should further investigate this issue.
Priority	Optional
Used by	Retailer, Winemaker
Success end conditions	Purchased bottle of wine marked as sold.
Failed end conditions	Purchased bottle of wine not marked as sold.
Actors	Cashier, Customer, Winemaker
Trigger	Customer purchased bottle of wine.
Main Scenario	1. Cashier scans smart tag on bottle of wine with mobile app for retailers. 2. Cashier marks bottle as sold.
Extensions	
Variations	*Expensive bottle of wine with NFC seal-state tag.* 1a. Cashier scans NFC tag to confirm that bottle of wine is unopened. 1b. In case that NFC seal tag is broken (bottle opened) cashier should offer replacement and further investigate this issue. *Purchased bottle of wine already marked as sold.* 2a. Cashier should offer replacement for purchased bottle of wine and further investigate this issue.
Exceptions	1a. Smart tags not present on bottle of wine. Cashier should further investigate this issue. Use case failed.
Info / Notes	
Open Issues	

Section 4 - Logical View

Figure 4.1 illustrates the logical view of the TagItWine solution. The diagram depicts main components of the pilot and provides high-level connection between them. The TIS based cloud platform “glues” all of the end user applications together. The end user applications include:

- Web application at the wine maker’s site utilized to support tagging of bottled products and box/pallets. A nice to have feature would be to interface this component with the existing wine maker’s information system in order to extract information on the current product batch as well as to support automated way bill creation for boxes/pallets.
- Mobile app for distributors provides optional functionality to scan/control the content of the shipment received a the distributor’s site.
- Mobile app for retailer provides an optional function to provide sales confirmation to the TagItWine platform. Each sold bottle would be marked as sold. Optionally, this feature can be achieved with an integration/interface between retailer’s information system and TagItWine platform (nice to have).
- Mobile app for customer will provide main experience for the end users, wine consumers, and it will be used to perform wine bottle authentication. This app will be also used to obtain feedback from the users on the status of the bottle (unsold, sold, opened) and to obtain customer satisfaction (product rating).

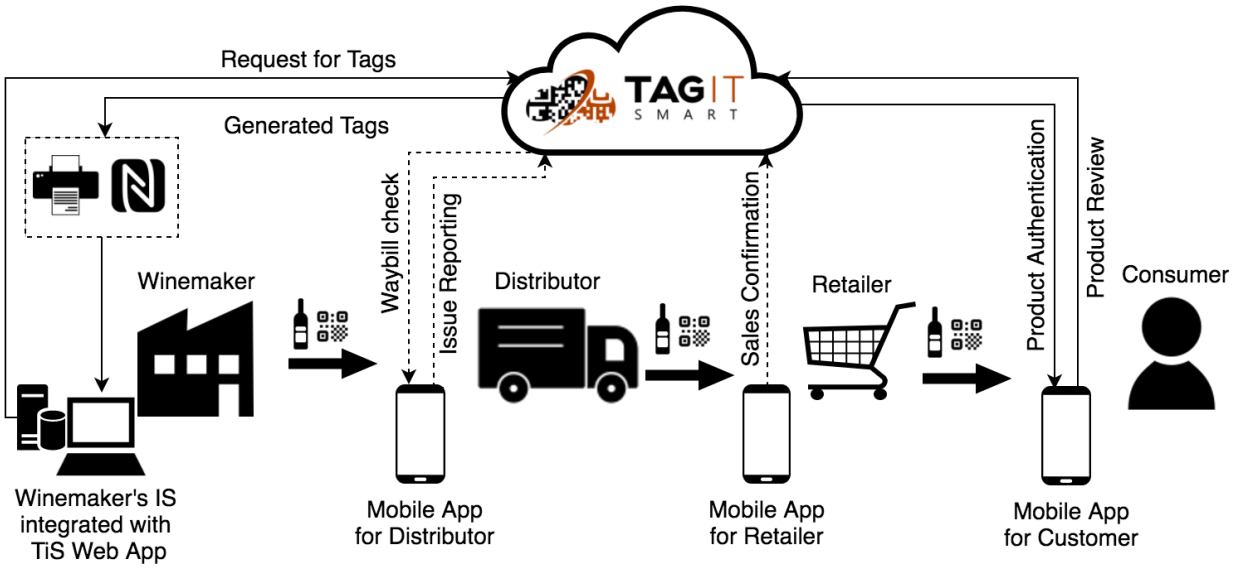

Figure 4.1. TagItWine logical view.

Section 5 - Process View

Process view for the TagItWine solution is illustrated in Figure 5.1 below:

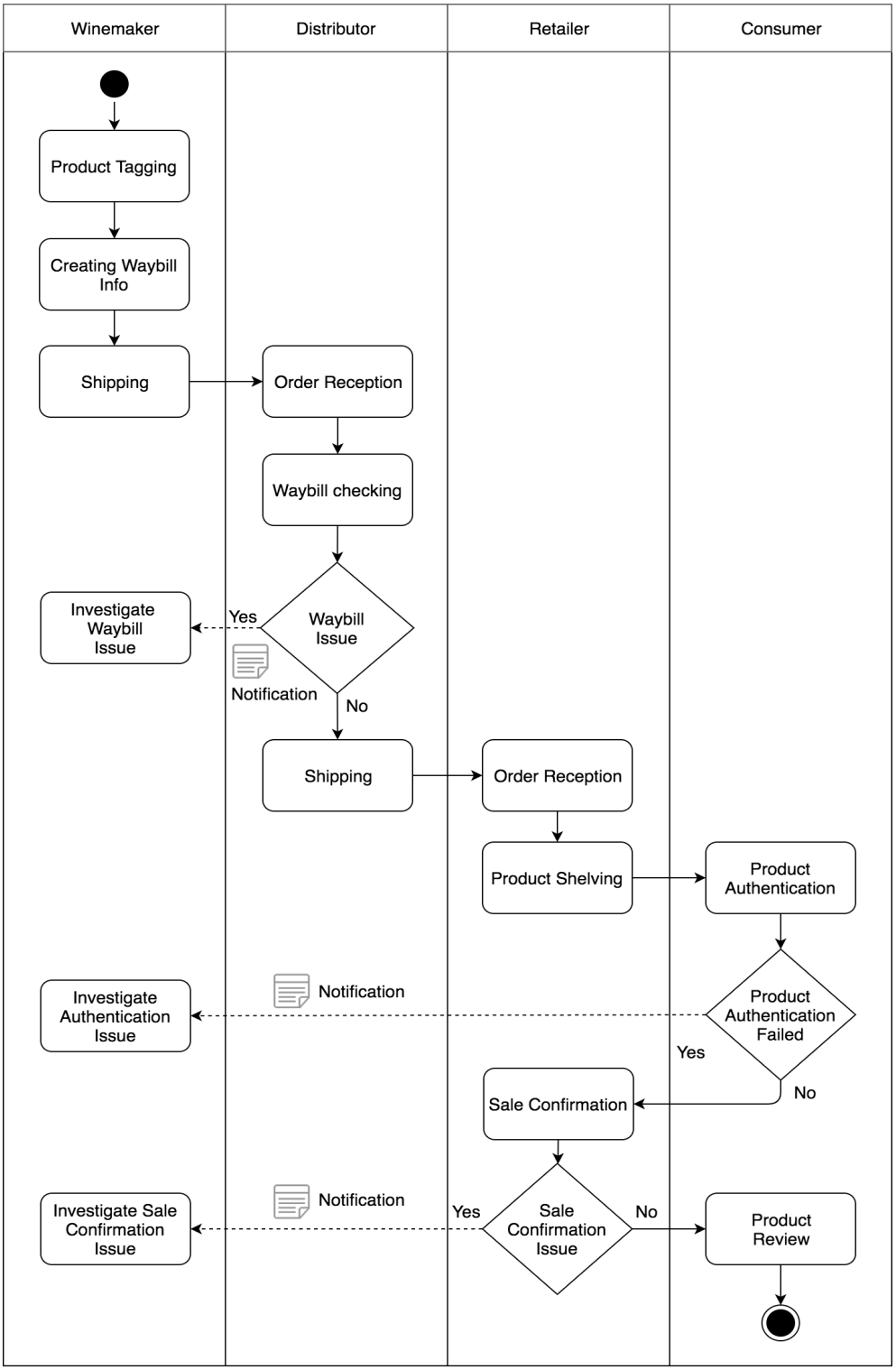

Figure 5.1. TagItWine process view.

Section 6 - Implementation View

The main components of the TagItWine solution and their organization is shown in Figure 6.1. Additionally, internal organization of mobile applications will follow the structure shown in Figure 6.2.

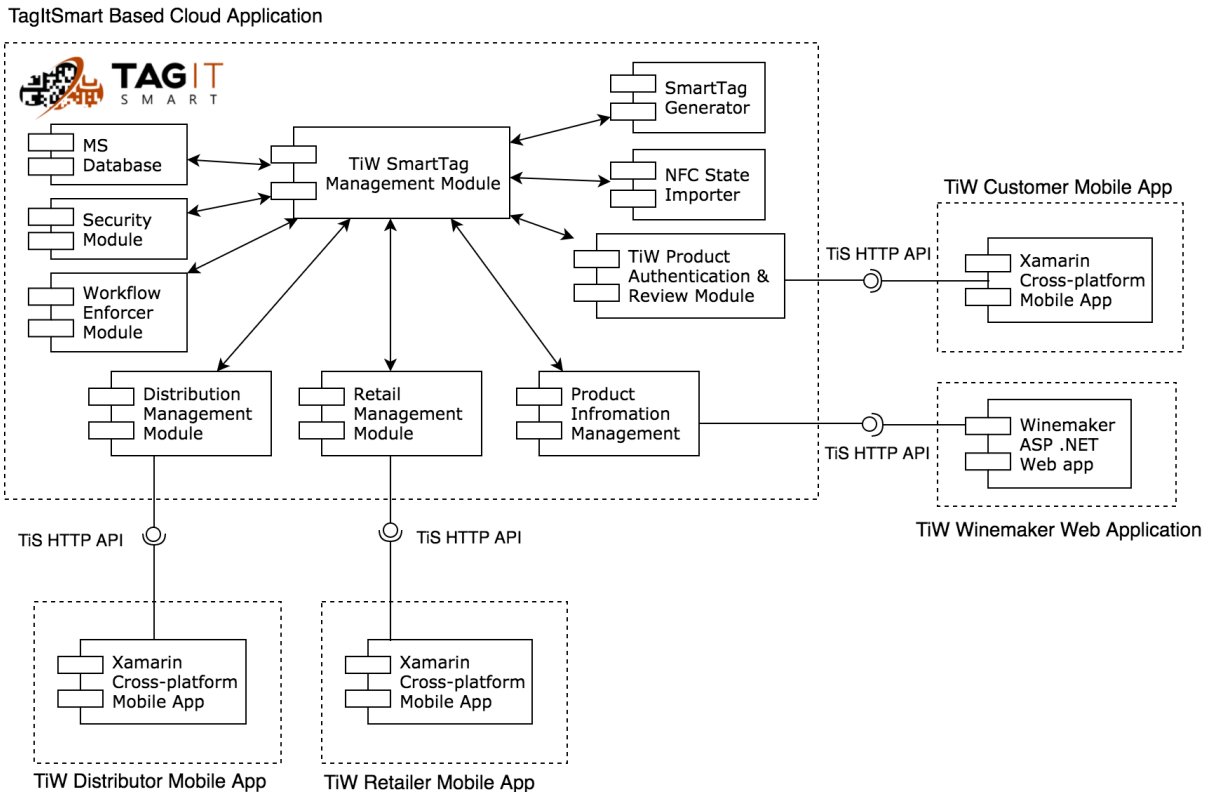

Figure 6.1. Implementation view: TagItWine component diagram.

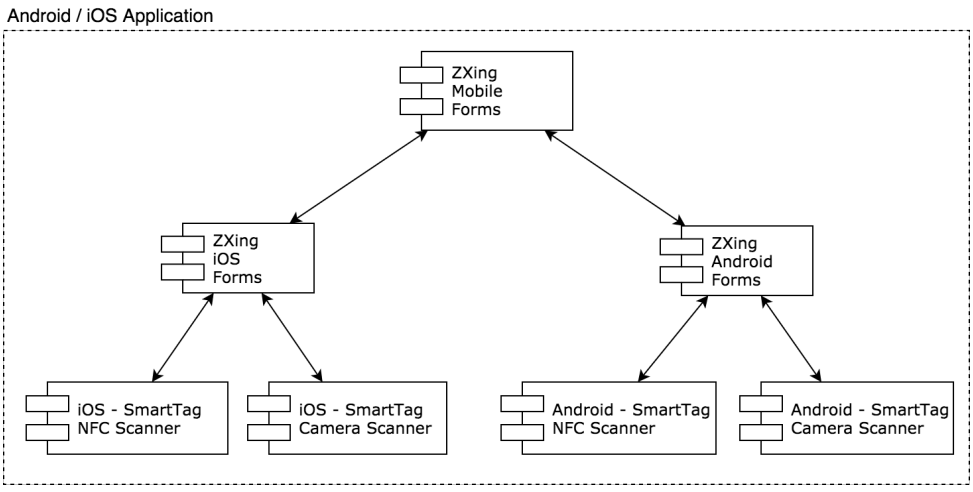

Figure 6.2. Implementation view: Mobile Application component diagram.

Section 7 - Deployment View

Physical organization of the TagItWine solution is illustrated with deployment diagram shown in Figure 7.1. The TagItWine cloud platform will be implemented using Azure Cloud server. It will host the key back-end components TIW management, product authentication and review modules, as well as the implementation of APIs needed to interact with other applications.

TIW Customer mobile app and TIW winemaker web app will play the key roles in the realization of main TagItWine use scenarios.

Optionally, the distribution and retail control functions will be implemented using mobile applications as well.

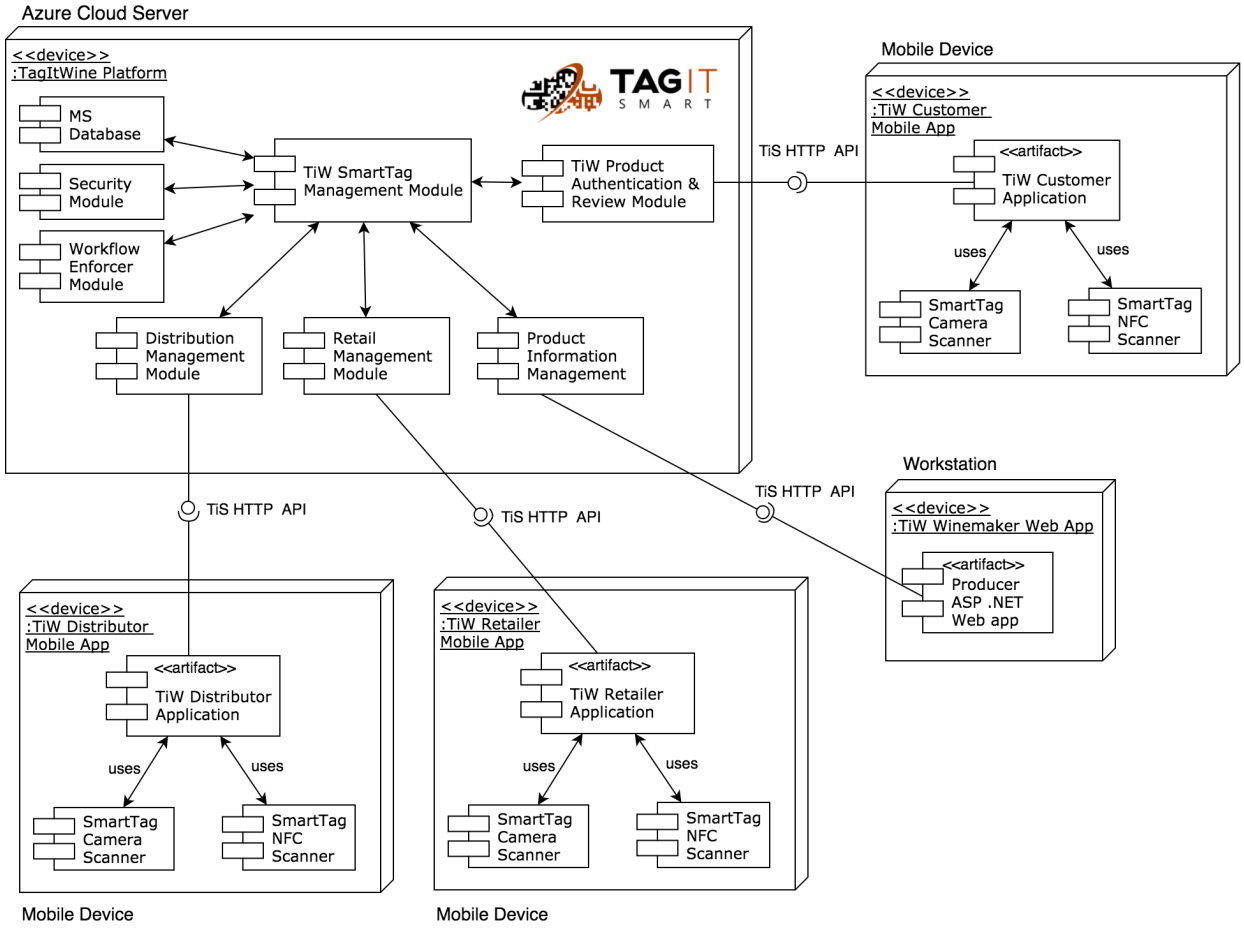

Figure 7.1 Deployment view.

Section 8 - Interface View

This section is a placeholder for all the information on APIs, key communication messages, access to data, etc. It is expected to be updated as the implementation of the pilot progresses.

8.1 TagItWine specific messages

The original TagItSmart API should be extended to support brand protection pilot specific messages.

- Request for dynamic QR code tag generation. Request message must contain unique batch number and number of required QR codes.
- Request for distribution waybill information. Request message must contain unique waybill identifier (retrieved by scanning the smart tag). In response waybill specification for the shipment should be received.
- Product sale confirmation update. Request for sale confirmation must contain unique bottle identifier retrieved by scanning the smart tag. In response, information about successful product status update should be found.
- Waybill issue report. Request message should contain waybill unique identifier (if it can be retrieved) and more detailed explanation of the problem. In response, information about successful issue recipient should be received.
- Sale issue report. Request message should contain product unique identifier (if it can be retrieved) and more detailed explanation of the problem. In response, information about the successful issue recipient should be received.
- Authenticity issue report. Request message should contain product unique identifier (if it can be retrieved). Request message might also carry additional information about the problem as well as the location where the problematic scan occurred. In response, information about the successful authenticity issue recipient should be received.
- Request for product authenticity status. Request must contain unique bottle identifier retrieved by scanning the smart tag. In response, information about the authenticity status should be received.
- Product review post. Request must contain unique bottle identifier retrieved by scanning the smart tag and the customer review. In response, information about the successful review post should be received.
- E-greeting card upload. Request message might contain multimedia content (text, image or video) with mandatory unique bottle identifier and end consumer e-mail. In response, information about successful upload should be received.

8.2 TagItSmart HTTP API

Table 8.1. TagItSmart HTTP API (DunavNET)

VERB	ENDPOINT	DESCRIPTION
User Account API		
POST	api/account/login	User login.
POST	api/account/registerconsumer	Crete new consumer account.
GET	api/account/userinfo	Get user info.
POST	api/account/logout	User logout.
POST	api/account/changepassword	Change user password.
POST	api/account/validatetoken	Validate authentication token on server-side.
POST	api/account/isauthenticated	Check whether user is authenticated.
POST	api/account/registerdevice	Register new user device.
POST	api/account/forgotpassword	Password retrieval procedure.
Product API		
POST	api/product/getproduct	Get product information.
GET	api/product/getlocations?name={name}	Get retail shop locations.
Distributor API		
POST	api/distibutor/getorders	Get orders.
POST	api/distibutor/startdeliverytoday	Mark that order delivery has started today.
Producer API		
POST	api/producer/getproducts	Get all products.
POST	api/producer/editproduct	Edit product info.
Order API		
POST	api/order/ordershistory	Get order history.
POST	api/order/statusorderchanged	Change order status.
POST	api/order/checkingstatus	Check order status.
SmartTag API		
GET	products/{id}/SmartTag/Info	Get Smart Tag context.
POST	products/{id}/SmartTag/Info	Get Smart Tag context.
POST	products/{id}/SmartTag/Activating	Activate Smart Tag sensor.

8.3 Winemaker Web Application User Interface

Winemaker web application will be used at Plantaze to request and create QR codes for production batch, to create digitized version of bottled products and link them with smart tags and labeled bottles. The web application for winemakers in this pilot will be based on the TIS component used in similar pilot demonstrations (Figures 8.1 and 8.2).

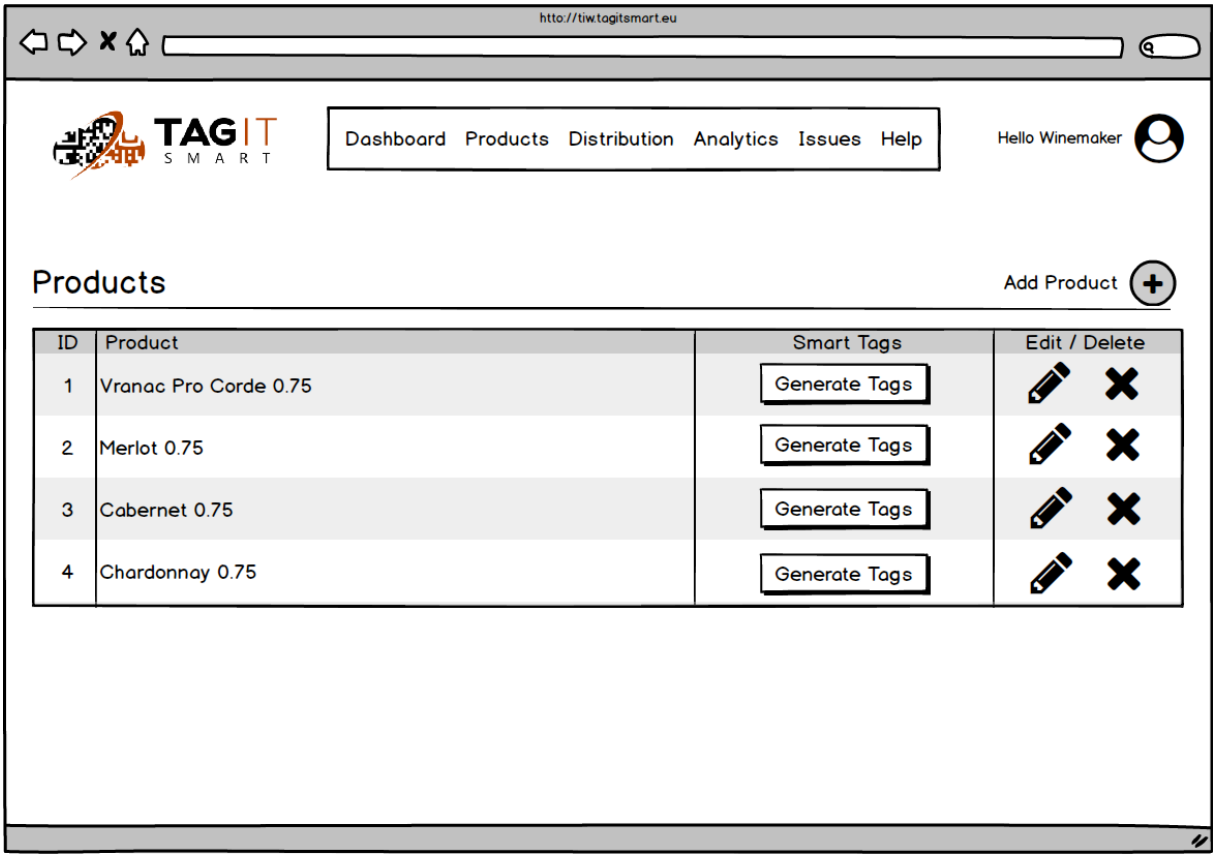

Figure 8.1. Winemaker Web Application Interface

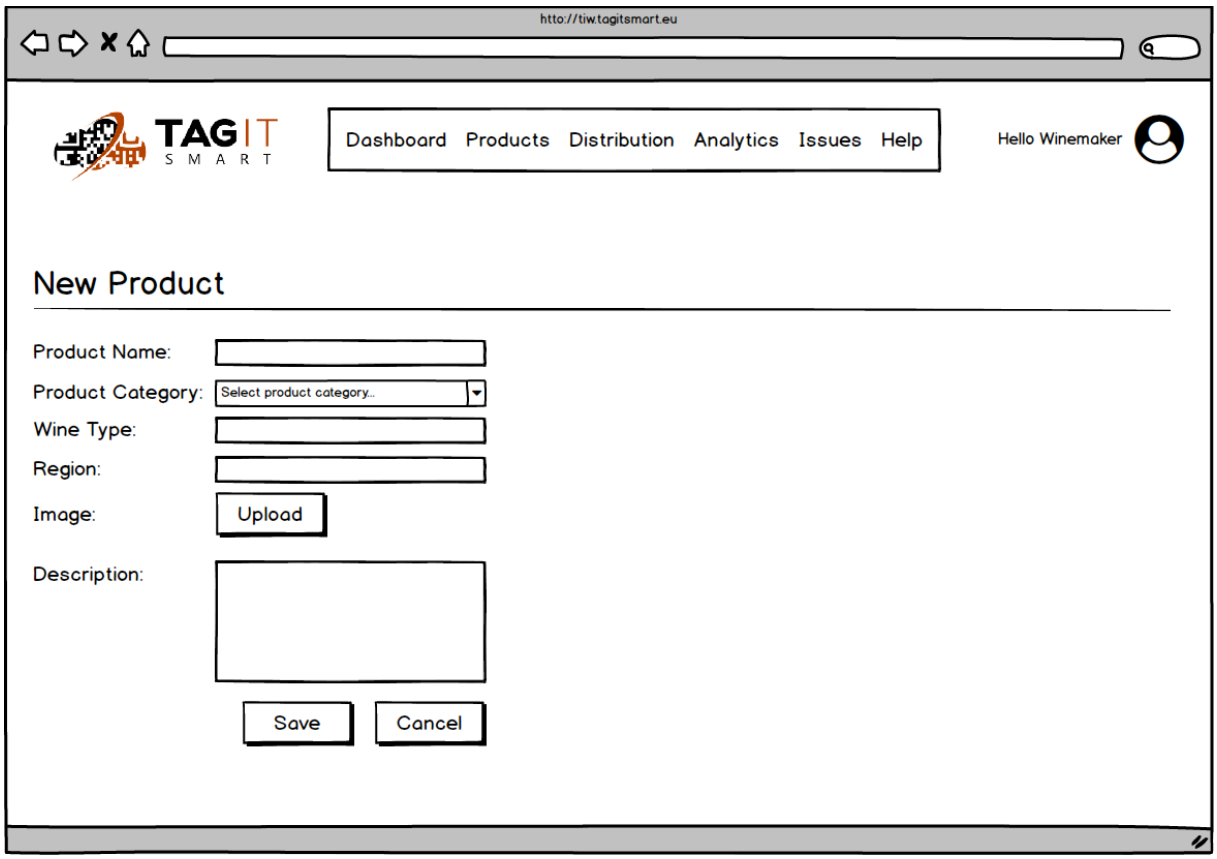

Figure 8.2. Web application: adding of a new product by winemaker

8.4 Mobile Application GUI

The GUI layout for the customer mobile app is depicted in Figure 8.3. First screen will offer a welcome message and options to scan a product or to search for stores. Selecting the scanning option will open the second screen (shown right) which will ask user to confirm:

- if the wine is still in store (unsold),
- if the wine is on the table (sold, not opened), or
- opened and consumed.

Figure 8.3. Mobile application welcome screen.

Scanning of a bottle and wine information screens are shown in Figure 8.4.

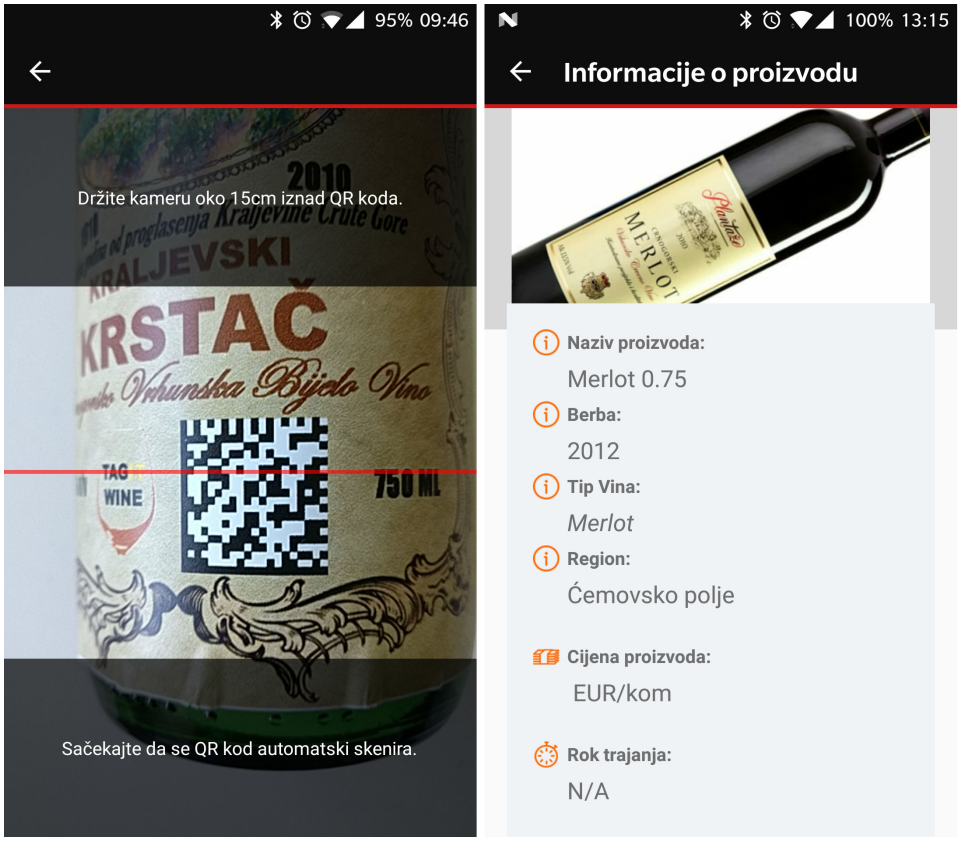

Figure 8.4. Mobile application scan and info screen.

Section 9 - Data View

The data flow diagram is shown in Figure 9.1.

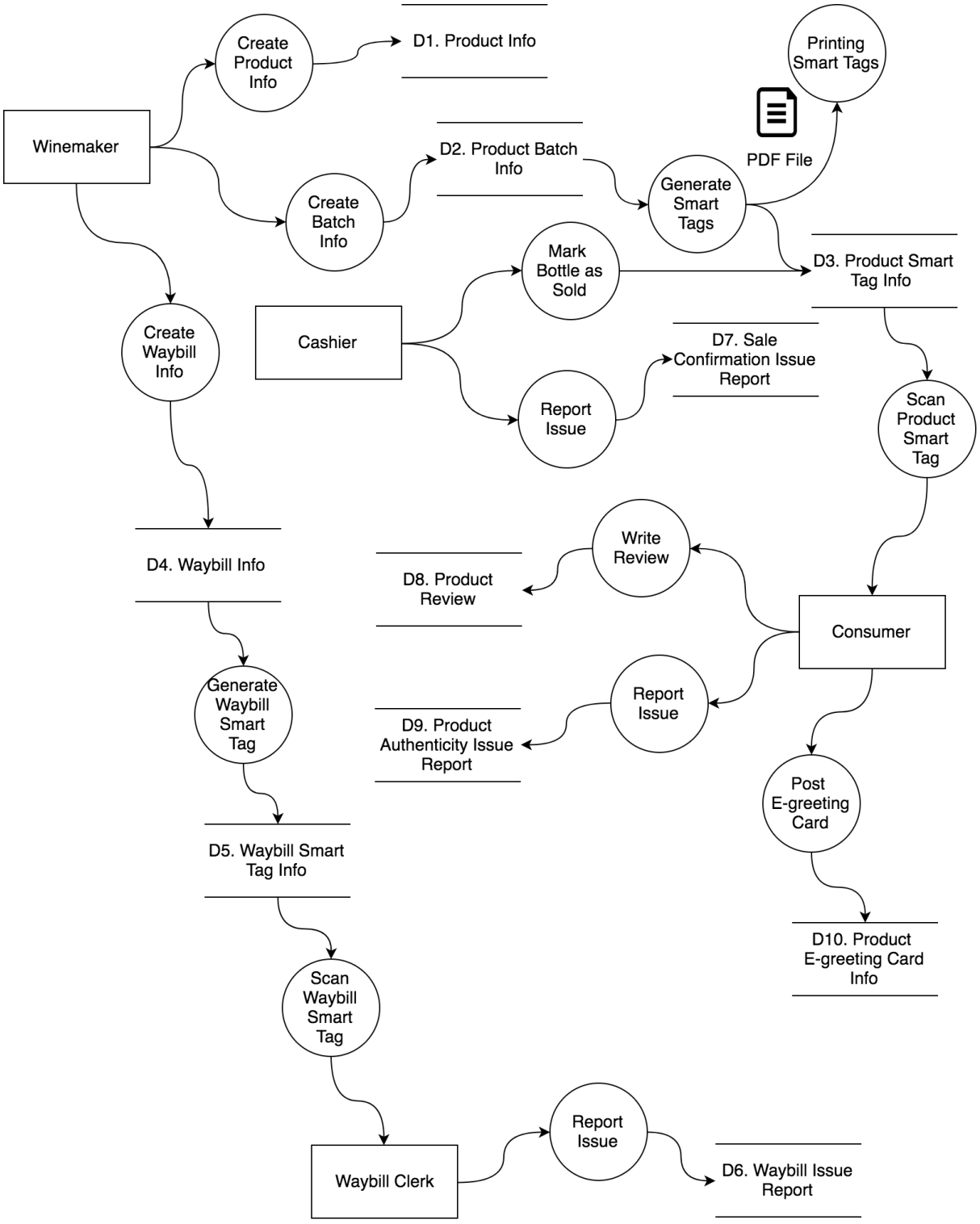

Figure 9.1. TagItWine Data Flow Diagram.

Table 9.1. Product Information Data Record.

D1 – Product Info	
Description	Winemakers must be able to store information about wine products. Product Info data record will hold general information about the product like product name, product category, ideal serving temperature and URL to product image.
Source	Winemaker
Destination	Database

Table 9.2. Product Batch Info Data Record.

D2 – Product Batch Info	
Description	Winemakers must be able to store information about the newly produced batch of products. Each record about the batch must contain the number of produced units in that specific batch. Beside this, information about the production date, vintage and region of grape origin must be stored as well. Relationship between Product Info data record and Product Batch Info should be established by using the foreign key field. If possible, product batch data record should contain the reference on the batch identification number from the winemaker's existing information system.
Source	Winemaker
Destination	Database

Table 9.3. Product Smart Tag Information Data Record.

D3 – Product Smart Tag Info	
Description	For each product unit (bottle) unique smart tag must be generated. Product Smart Tag Info record must contain unique identifier that will be paired with appropriate product unit. Beside this, Product Smart Tag data record will carry info about smart tag states (e.g. sealed/unsealed for NFC sensors) which will be utilized for product authentication and validation. Product Smart Tag Info data record will be related with all records that carry product unit specific data. Smart tag data record should have field that will indicate how many times smart tag has been scanned. If possible, product smart tag info data record should contain the reference on the product unit serial number from the winemaker's existing information system.
Source	TiW platform
Destination	Database

Table 9.4. Waybill Information Data Record.

D4 – Waybill Info	
Description	Winemakers should be able to create information about the shipments. Each order should have appropriate waybill that must match the shipment content. Alongside with details about each item that should be delivered, information about recipient should be stored.
Source	Winemaker
Destination	Database

Table 9.5. Waybill Smart Tag Info Data Record.

D5 – Waybill Smart Tag Info	
Description	For each waybill record, matching waybill smart tag record should be created. Beside smart tag unique identifier, general info about delivery (destination, start and end date of the delivery, etc.) should be stored. If possible, reference on the waybill's serial number from the winemaker's existing system should be stored as well.
Source	Winemaker
Destination	Database

Table 9.6. Waybill Issue Report Data Record.

D6 – Waybill Issue Report	
Description	In case of problems with delivery record about the issue should be created. Reference to Waybill Smart Tag Info should be stored that will easily identify Waybill Info that should be investigated. Beside this, there should be a text field where description of the problem should be stored.
Source	Distributor
Destination	Database

Table 9.7. Sale Confirmation Issue Report Data Record.

D7 – Sale Confirmation Issue Report	
Description	In case of sale confirmation issue (product already sold, smart tag not found, etc.) Sale Confirmation Issue Report data record should be made. If possible, product smart tag identifier should be stored alongside with issue description.
Source	Retailer
Destination	Database

Table 9.8. Product Review Data Record.

D8 – Product Review	
Description	Customer must be able to post a review for the consumed product. With mandatory reference to the product smart tag identifier actual comment about the consumed product should be stored. Additionally, device location at the moment of scanning should be logged, if possible. Also, field for star rating (1 – 5 stars) should be available.
Source	Customer
Destination	Database

Table 9.9. Product Authenticity Issue Report Data Record.

D9 – Product Authenticity Issue Report	
Description	Product Authenticity Issue Report data records should be created automatically once spotted by TagItWine platform. Beside collected smart tag product identifier and issue type, location of the device used for scanning should be logged (if possible).
Source	TagItWine Platform
Destination	Database

Table 9.10. Product E-greeting Card Info data record.

D10 – Product E-greeting Card Info	
Description	Consumer should be able to upload the e-greeting card details. E-greeting card data record should contain scanned product smart tag identifier, recipient e-mail (or phone number) as well as the e-greeting card message. Fields that will store URLs to media (photos or videos) should be also available. Additionally, e-greeting card expiration date might be stored.
Source	Customer
Destination	Database

Section 10 - Non-functional view

10.1 Technology selection

- **TagItSmart Cloud platform** – will be used to host database and keep the TiS HTTP API running.
- **SmartTag Encoding, Creation and Scanner SDK** – will be used to generate non-standard QR codes with dual states. Therefore, appropriate scanner libraries will be used in order to correctly detect both states.
- **Photochromic ink** – will be used to enable printed QR codes to have two states (normal and active).
- **NFC opensense** – will be used to easily identify seal status (opened or not-opened) on more expensive bottles of wines.
- **Xamarin Mobile based SDK** – will be used to write cross-platform mobile applications (for Android and iOS).
- **Workflow Enforcer** – will be used to properly manage workflow, events and notifications.
- **Semantic Parser / Android based user authentication** – will be used to authenticate users on mobile devices by using their location history or typing behavior.

10.2 Security View

Security aspects of the pilot solution will follow guidelines and recommendations provided by TiS. Security of the solution will rely on the security of the TiS components with respect to user authentication, user authorization, different types of users and user roles, restricted access, encryption, etc. The topics of considerations should include:

- Security procedures
- User authentication
- User authorization,
- Firewall use,
- VPN access,
- Types of users and roles.

10.3 Network Requirements

It is anticipated that the pilot implementation will require continuous Internet access to TagItWine cloud platform, as well as Internet access for the apps on mobile devices.

Section 11 - Principles

The main principles for the implementation of the pilot are summarized below:

- It is assumed that the pilot system implementation will rely on the use of existing components and technologies provided by TiS (see Appendix).
- Loose coupling: each component or software module that is implemented under the TagItWine project, should be designed and implemented as independent as possible. The components should be designed as stand-alone systems with clear specifications of the input and output data formats that match TiS API formats and easily integrates with TiS ecosystem.
- Minimize dependency: each component should be designed with minimal dependency on other TiS or TagItWine components, and whenever possible, avoid heavy dependency on third-party software libraries.
- Use of TiS APIs: TagItWine pilot components are expected to be integrated by the means of standardized and unified principles for the data exchange, as defined by TiS ecosystem of components.
- Data formats: all of the information exchanged between the pilot components should utilize a uniform and standardized formats and data structures that are in line with the rest of TiS ecosystem.

Section 12 - Conclusion

This report on pilot system architecture and design serves as the deliverable D1.1 for the TagItWine project, work package WP1.

The document discusses the objectives and approach to implementation and provides architectural views and perspectives that describe the system. Main functional requirements are described using use cases view, while the structure of the system is described using logical, process, component, and deployment views. Additionally, the document provides discussion on non-functional requirements including technology selection, interface view, and details on the data models.

This report on pilot system architecture and design provides starting point for the implementation and execution of the pilot, as planned for work package WP2.

List of Abbreviations

API – Application Programming Interface
HTTP – Hypertext Transfer Protocol
ID – Identification
NFC – Near-field communication
TiS – Tag It Smart
TiW – Tag It Wine
URL – Uniform Resource Locator
QR – Quick Response

Appendix

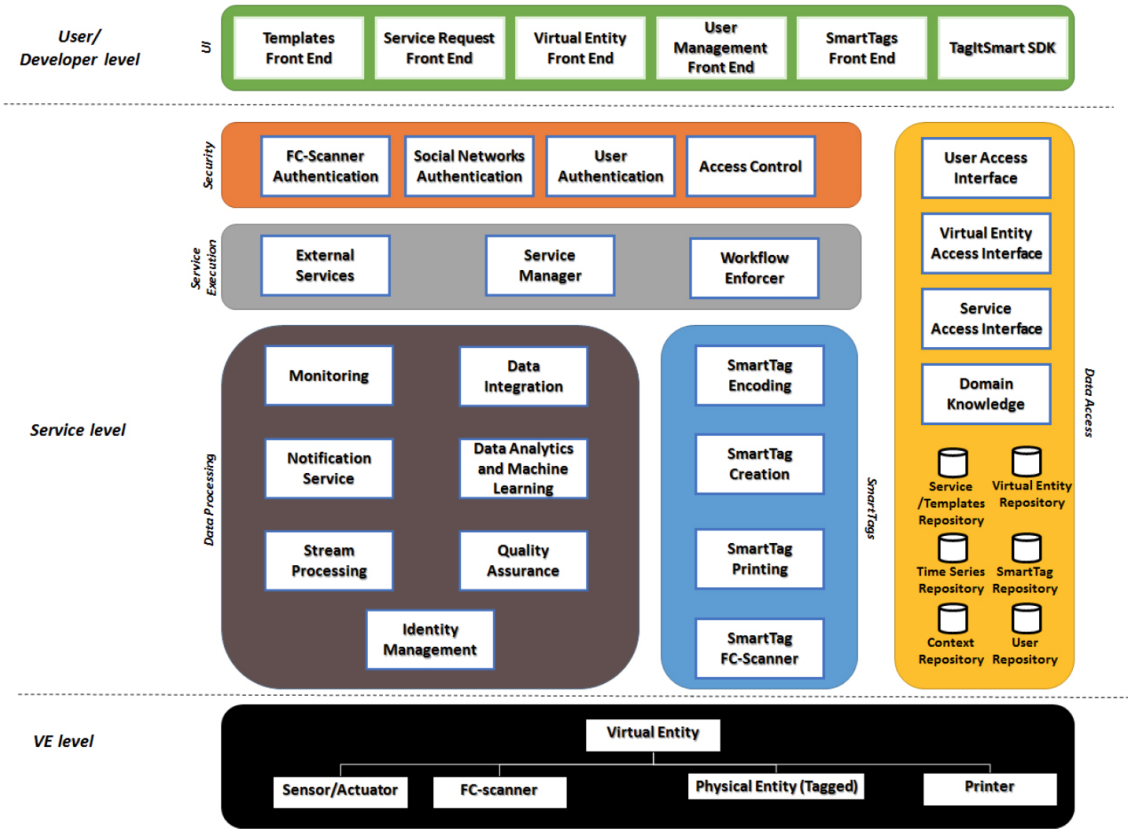

Figure A1. TagItSmart ecosystem.
Source: TagItSmart Open Call Announcement #1.