Deliverable D400.3

Report on validation activities and Detailed Specification Revision

WP 400

Project acronym & number:	SmartAgriFood – 285 326
Project title:	Smart Food and Agribusiness: Future Internet for Safe and Healthy Food from Farm to Fork
Funding scheme:	Collaborative Project - Large-scale Integrated Project (IP)
Date of latest version of Annex I:	18.08.2011
Start date of the project:	01.04.2011
Duration:	24
Status:	Final
Authors / Contributors :	Augusto Morales(UPM), Ignacio Gonzalez (Atos), Esther Mietzsch (KTBL), Marta Fontsere (Bonpreu), Huub Scholten(WU), Alfredo Pedromingo (Ariadna), Tim Sadowsky (GS1), Yann Cassing (SGS), Ramón Alcarria (UPM), Javier Garcia (Atos)
Due date of deliverable:	31.12.2012
Document identifier:	D400.3_v1.3.docx
Revision:	01

FUTURE INTERNET PPP

SEVENTH FRAMEWORK PROGRAMME

Smart Agri-Food

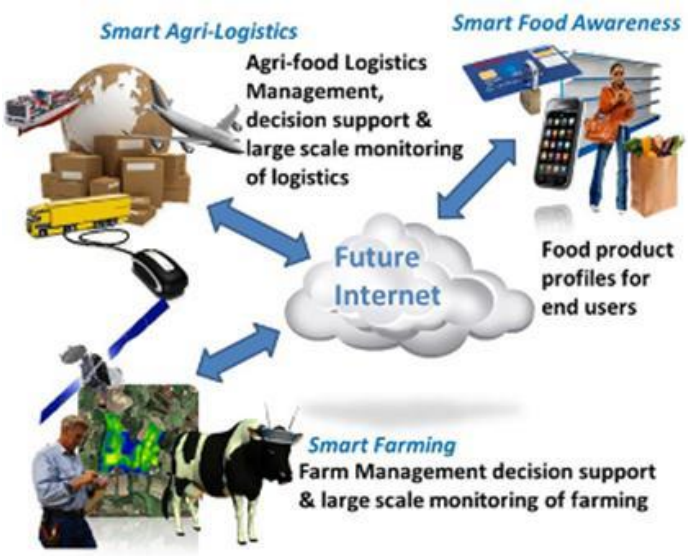
The SmartAgriFood Project

The SmartAgriFood project is funded in the scope of the Future Internet Public Private Partnership Programme (FI-PPP), as part of the 7th Framework Programme of the European Commission. The key objective is to elaborate requirements that shall be fulfilled by a "Future Internet" to drastically improve the production and delivery of safe & healthy food.

Project Summary

SmartAgriFood aims to boost application & use of Future Internet ICTs in agri-food sector by:

- Identifying and describing technical, functional and non-functional Future Internet specifications for experimentation in smart agri-food production as a whole system and in particular for smart farming, smart agri-logistics & smart food awareness,
- Identifying and developing smart agri-food-specific capabilities and conceptual prototypes, demonstrating critical technological solutions including the feasibility to further develop them in large scale experimentation and validation,
- Identifying and describing existing experimentation structures and start user community building, resulting in an implementation plan for the next phase in the framework of the FI PPP programme.



Project Consortium

LEI Wageningen UR; Netherlands	Campden BRI Magyarország, Hungary
ATB Bremen; Germany	Aston University; United Kingdom
TNO; Netherlands	VTT; Finland
CentMa GmbH; Germany	OPEKEPE; Greece
ATOS ORIGIN; Spain	John Deere; Germany
ASI S.L.; Spain	Wageningen University; Netherlands
Huawei; Germany	EHI Retail Institute GmbH; Germany
MTT Agrifood Research; Finland	GS1 Germany GmbH; Germany
KTBL e.V.; Germany	SGS S.A.; Spain
NKUA; Greece	Bonpreu S.A.U; Spain
UPM; Spain	

More Information

Dr. Sjaak Wolfert (coordinator)
LEI Wageningen UR
P.O. Box 35
6700 AA Wageningen

e-mail: sjaak.wolfert@wur.nl
phone: +31 317 485 939
mobile: +31 624 135 790
www.smartagrifood.eu

Dissemination Level

PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	X
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Change History

Version	Notes	Date
0.1	Table of content provided by UPM	1/07/12
0.2	Minor changes by UPM	15/12/12
0.3	Minor changes by UPM	20/12/12
0.4	Major changes. Integration of draft inputs from all the partners	31/12/12
0.5	Toc changes by UPM, inputs from ATOS	10/01/13
0.6	Inputs from SGS	20/01/13
0.7	Inputs from Ariadna, Bonpreu, and UPM	24/01/13
0.8	Final inputs from Ariadna and UPM	25/01/13
0.9	Some comments added, and conclusions by UPM	26/01/13
0.93	Modifications added (Chapter 4, S2.2.2) and comments resolved by WU	04/02/13
0.95	Final check on content and layout by UPM	05/02/13
1.0	Final version	06/02/13
1.1	Revised final version	
1.2	Editing changes	
1.3	Final recommendations resolved over the document summary	

Document Summary

This document can be understood as a continuation of previous deliverables related to this Work Package, D400.1 and D400.2. Therefore, as previous efforts were focused on detection of missing architectures, information models, and user awareness requirements, this document clarifies technical developments and pilots which were carried out to analyse and propose feasible solutions for these challenges.

Within the SmartAgriFood Project and especially under the WP400, user involvement and feedback are objectives which provide a broader scope and hands-on knowledge regarding the whole user awareness concept; so, this deliverable describes the two pilots which were deployed for this purpose.

The first pilot is the: Tailored Information for Consumers (TIC). This pilot is mainly focused on the data management and provision to consumers, in the form of interactions with smartphones. The second pilot is the: Tracking and Tracing and Awareness in Meat supply chains (TTAM), which targets the increase of consumers trust in meat by providing trustworthy and certified information. Both Pilots complement each other by tackling two different issues that involve users and the information they are willing to consume and trust. In addition, this deliverable also describes the integration with some FI-WARE's generic enablers that were possible to integrate within the developments by this date.

The structure of this document is divided in three parts. First, there is a review of the architecture proposed in deliverable D400.2. Then, pilot's developments are independently described, and finally chapter 5 explains the high level and low level integration efforts between the TIC and TTAM pilot. As this deliverable has been planned to be technical and descriptive, all the results of both pilots are explained in detail in D400.4.

Abbreviations

DSE	Domain Specific Enabler
FI-PPP	Future Internet Public Private Partnership
GE	Generic Enabler
GIS	Geographic Information System
ICT	Information and Communication Technology
KPI	Key Performance Indicator
RFID	Radio Frequency Identification
TIC	Tailored Information for Consumers
TTAM	Tracking Tracing and Awareness Meat
REST	Representational State Transfer
ESB	Enterprise Service Bus
OS	Operating System
HTML	Hyper Text Markup Language

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Objective

The purpose of this document is to report the validation activities that have been carried out in WP400. These activities include the Tailored Information for Consumers (TIC) pilot and the Tracking, Tracing and Awareness in Meat supply chains (TTAM) pilot. This document also includes a technological description of all the developments related to both pilots. In addition to this, this deliverable contains a detailed specification revision of the architecture and models proposed in previous deliverables of WP400 [3][4].

1.2 This Document

This document covers the technical description of the pilots as well as their integration with the Generic Enablers (GE) proposed by the FI-WARE project. Hence, this document can be taken as a technical specification of all the elements that have been implemented under the TIC and TTAM pilot. These elements include software developments, network deployment, physical/hardware deployment, integration efforts and architectural revisions.

Since the TTAM and TIC pilots aims to offer a real testing experience of SmartAgriFood's awareness systems, a description of the scenarios is provided. Hence, this document describes some of the real challenges of these scenarios taking into consideration that Future Internet's technologies are still in development.

The technical specification enclosed in this document includes:

- Software Developments
- Low level interface descriptions
- Data Models.
- Communication models

Another issue addressed in this deliverable concerns to the description of the integration of Generic Enablers within the pilots. By the time this deliverable was finished, there were some useful but not-implemented FI-WARE's GE [2]; so this document also describes the qualitative benefits of this integration.

As the TTAM and TIC pilots use of some novels developments and architectures, this deliverable also includes standardization sections for each pilot.

1.3 Target Audience

The dissemination of this document is of category PP, i.e. is restricted to other programme participants (including the Commission Services). The main audience consists of consortium members and especially those involved in WP400. It is also intended as technical description of the pilots.

2 Revised Architecture of the Smart-awareness Scenario

2.1 Revised Architecture

The revised architecture of the smart-awareness scenario describes modules, functionalities and protocols needed inside the TIC and TTAM pilots as a complement of the previously described architecture of D400.2. The revised architecture's implementation reflects the current and appropriate technologies that will be feasible to implement taking into consideration the current state of the art. Hence, it is explained in detail in each chapter of the pilots' specifications.

In order to implement the pilots we have defined, first, a set of domain specific enablers. These enablers act as specific characteristics that are not copped, at this moment, by any other FI-WARE generic enabler. Hence, these enablers are implemented using current technologies but could be easily replace, in the future, with the FIWARE's Generic Enablers (GE).

2.2 Validation of System Architecture

The designed architecture for SmartAgriFood Awareness validates the functional requirements identified in D400.2. The main objective of architecture is to provide all the functional components needed for providing information awareness to the user level. Due to the iterative process of the design of the architecture, in this document we describe the high level modification we have upgraded. As one of the goals of this work-package consists on the application of the FI-WARE Generic enablers, this document also states the integration efforts under the development of the pilots.

The specific validation of the SmartAgriFood Awareness architecture and the full set of requirements, developments and efforts conducted towards their fulfilments have been addressed in chapters 3 and 4.

2.3 Domain Specific Enablers

WP400 system development approach was based on several iterative cycles, where design, development and evaluation with real users were done. In the first steps of the design of the system no GEs were still available, hence, the integration of GEs was only theoretically explored by identifying which functional and technical components, that the system needed, could be provided by GEs based on their description in the HLD. The outcome of this study can be found in D400.2.

With the publication of FI-WARE's testbed, and taking into account the availability and completeness of the GEs and all the determining factors that involved their integration within the TIC solution, WP400 employs three specific GEs. A comprehensive description of these GE, and their implementation within the pilots can be found in SmartAgrifood Deliverable D500.2.

- IaaS Data Center Resource Management GE (DCRM)[1]. It provides the programming and also GUI framework to deploy and manage a Cloud infrastructure, offering provisioning and life-cycle management of virtualized resources associated with virtual machines. It is integrated in the TIC system in order to deploy the (server and mobile) services in a virtual cloud environment. Detailed description of these services can be found in Section 3.1.
- Identity Management GCP GE[2]. Allows the management of the user's registration and login lifecycle, based on the usage of the OpenID standard. Within the TIC system it is employed to manage users' information, preferences and the applied policies.
- Data Handling PPL GE[3]. Used to externalize the management of our user's data to a reliable entity using worldwide established security standards (XACLM-PPL). This enabler has been used, together with the GCP GE, in order to manage data from users. The detail specification can be found in D500.5.2

2.4 Link with FI-WARE's GE

Before the development of the Tailored Information for Consumers (TIC) application, many studies have been performed. The aplication was designed to fulfil the customers needs providing the customized information related to the selected product. One of the needs of the pilot was the definition of basic types such as computation resources and storage which could be allocated in the cloud. For a first approach of the application, some components where developed without using any Generic Enablers. One of this components was suppose to access the local databases in which all the different data related to the products and the users' preferences was storaged. Another requirements of the pilot was to reallocate information from the local supermarket's servers to the cloud. To reach this task and, based on the infrastructure developed, some computational resources were needed.

One of the TIC pilot's requirement was to maintain an easy resource handling (e.g. hardware, software or even information). Since one of the key characteristics was to permit users select products' ID through a QR code, it was also necessary to access some hardware capabilities (e.g. cameras), for reading the product code. As these sorts of accesses to capabilities is not standard, it was also necessary to encapsulate functionalities using a homogenous method. For the post shopping process was also necessary to design a system capable of notifying users whenever alimentary emergency occurred. If the supermarket detected that there were some product poisoning, the system could be capable of accurately detecting the right consumers (of

[1] http://catalogue.fi-ware.eu/enablers/iaas-data-center-resource-management-ge-fi-ware-implementation

[2] http://catalogue.fi-ware.eu/enablers/identity-management-gcp

[3] http://catalogue.fi-ware.eu/enablers/data-handling-ppl

those products) and later send them alerts. Therefore, shopping information should be stored in a common cloud (with the user agreement).

Concerning the TTAM pilot, since it requires high amount of data processing and storage, enablers that support these characteristics are suitable. The following table summarizes the implementation of the enablers in the first phase in the SmartAgrifood project.

Table 1. Generic Enablers Status within the WP400

Generic Enabler	Pilot	Status
IaaS Data Center Resource Management GE	TIC	Integrated and functional
Identity Management GCP GE	TIC	Integrated and functional
Data Handling PPL GE	TIC	Integrated and functional
Connected Devices Interface GE	TIC	Planned
Publish/Subscribe Context Broker GE	TTAM	Planned
Application Mashup - Wirecloud GE	TTAM	Planned

3 Tailored Information for Consumers Pilot Specification

3.1 High Level View of the TIC Pilot

The TIC pilot targets all the mechanisms (e.g. applications, infrastructure, data and communication models) that enable consumers to request information of a specific product using their Smartphone before/during and after their shopping process; so they only get the right product attributes of their interest according to their consumer shopping profile. This requires an infrastructure for managing consumer profile data (taking into account security and privacy issues) and for managing product attributes.

The TIC pilot is mainly focused on the data management and provision to consumers. As it already describe in the previous deliverable D 400.2 [4], there are two main information dissemination approaches:

1) The push approach, which enables consumers getting asynchronous information from products at any time, even if they are out of the scope of the supermarket.
2) The pull approach where consumers get tailored information on request during their visit in the supermarket's facilities.

The overall architecture [4], of the WP400, defines the architecture for supporting both approaches. The first phase of the pilot specification is focused on the second approach, which is less invasive to costumers and is more straightforward, in terms of application compatibility among different mobile devices. Nevertheless, in the pilot development process it has taken into consideration that the push communication approaches offers extended capabilities to the whole SmartAgrifood awareness scenario, so developments are prepared for it.

The data provision to the consumer is basically carried out by two ways:

- Providing tailored product information from selected products costumers will find in the supermarket.
- Showing hidden information from logos and signs which can be found in some products, usually processed products.

In order to match the consumer's interests, users can create a dynamic consumer profile in order to know what information they are interested in. Hence, the generation of tailored information depends on these profiles so this characteristic allows more accurate information matching in comparison with the generic and fixed information provision of products' labels.

Having tailored information after a matching process leverages privacy and security issues. As this information is supposed to be managed, in the future, by external entities in the form of GE; consumers, inside the TIC pilot, will be owners of all the tailored data they consume and produce. Consumers can also make use of anonymous profiles in the case they are not interested on permanently sharing their information with the supermarket, the service cloud and GEs behind.

As it is described in the previous deliverables the TIC pilot tackles tailored information generation and consumption through consumers' mobile devices. Thus, there are two main parts in the TIC pilot that are described in this section. The terminal side, which encloses the technological developments that allow consumers getting tailored information from the supermarket infrastructure using a personal mobile device (user domain). The infrastructure side encloses all the developments being executed in the supermarket domain. As the tailored information generation of this side, is not a stand-alone process, the supermarket domain makes use of a set of GE that provides some functionalities that will be describe in next sections.

3.1.1 Terminal Side

The design goal behind the terminal side of the smart food awareness scenario is to define a web service based architecture, and its corresponding implementation, able to tackle the traditional limitations exposed by typical client approaches where interfaces are tailored to fit specific deployment conditions (varieties of mobile and desktop environments). In this pilot, a web-style client provides service access with a solid and standard-compliant client framework that can be invoked from any user equipment with minimum requirements such as a web browser with HTML5 [5] support.

Since the terminal is the entry point of supermarket's users; the use of HTML5/Javascript based technologies (through web browsers) directly expands the reachability of users. This is because, there are several mobile operating systems, such as: Android, iOS, Blackberry and so on; so tackling the tailoring information process for every existing mobile Operating System (OS) could be a drawback for the whole process.

3.1.1.1 Mobile Architecture

The mobile architecture is mainly based on web technologies and the SmartWebProxy. The figure below shows the high-level client architecture implemented, where lower layers are specific to the device and the underlying OS. This solution stays on top of any particular OS kernel and any set of capabilities available in the terminal. The Web interface becomes the unique interaction tool for user as it is responsible from invoking local capabilities and clustering cloud-based data in order to render the composite service to terminal's screen. The Service Renderer and the Service Execution Engine is based on HTML5 and JavaScript.

This architecture fully supports the use of pure HTML web pages as the Service Description Language, instead of creating one APP coded on a native programming language such as Java/Android or iOS/Objective-C. Figure below illustrates how HTML5 is being used for defining functional elements of the service interface required by smart food awareness use case, and how this service is presented by a standard browser to the end user. The architecture allows defining one Service Description Language, based on standard HTML5, which is able to represent relevant user interactions for FI-PPP services. As mobile browsers are rapidly implementing HTML5 functionalities that are used in the pilot, we have evolved our original SmartWeb-

Proxy developments to a full web approach. Since most of the native code (e.g. invocations to cameras, sensors etc.) can be replaced by HTML5 code, the proxy side of the architecture has been implemented as an external Server layer, so it supports the mobile browser (that really perform as the execution layers) to fetch tasks in the form of HTML code. This approach in comparison with the original one, achieves better integration with several mobile OS, and allows updating and adapting "on the-fly" functionalities of the graphical interface and the underlying communication interfaces (e.g. camera, NFC or any other sensor). Hence, we consider this development more versatile for all the processes that involved consumers.

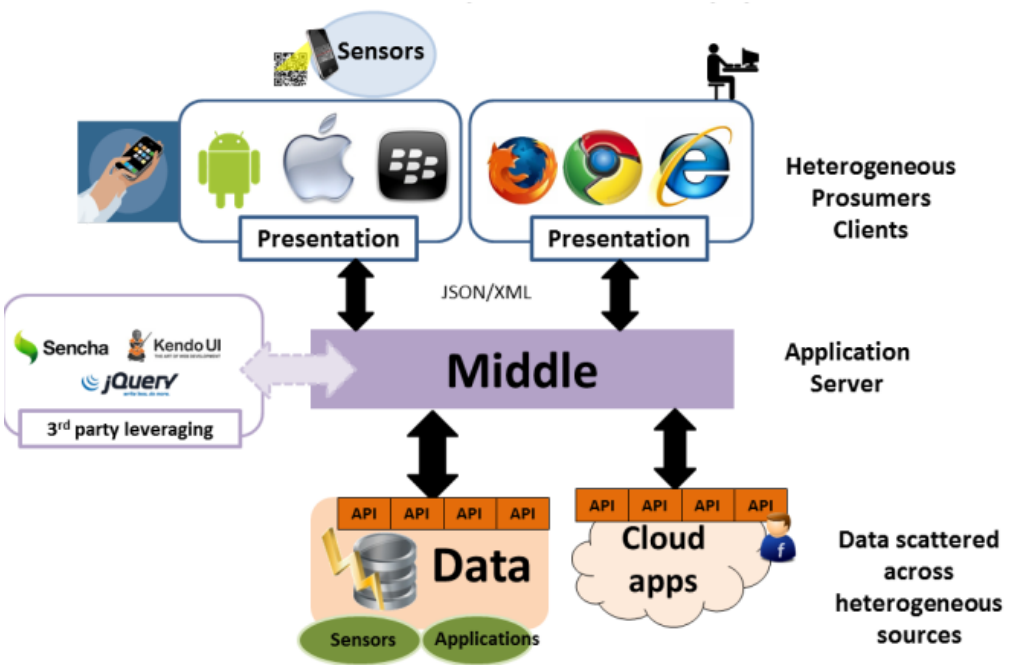


Figure 1. Mobile Interaction with different components

3.1.1.2 Detailed Specification

The figure shown below depicts the distinct technical layers used by the terminal side

- Client layer
- Server (middle) layer
- Cloud (data) layer

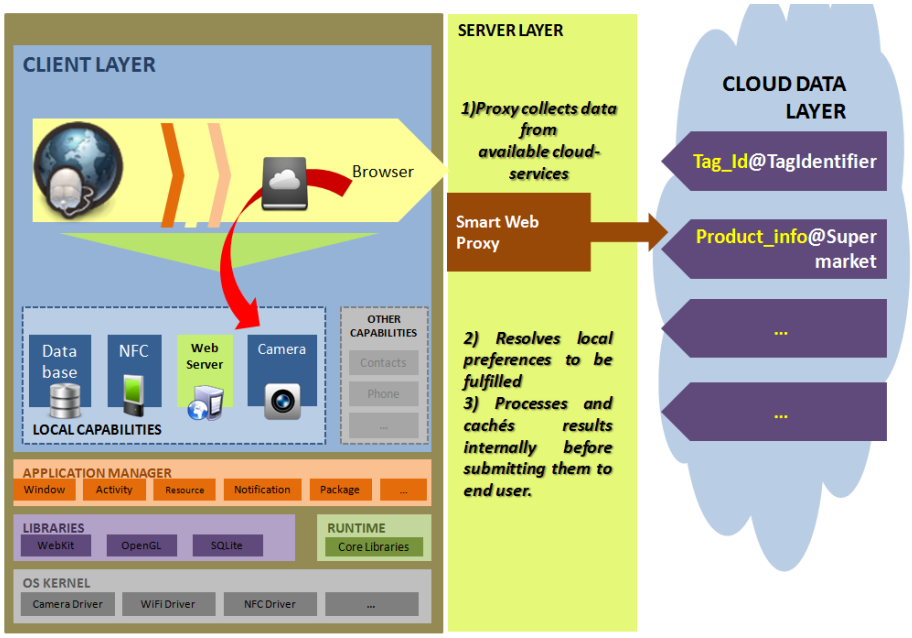


Figure 2. Implementation of the Mobile Architecture

Client layer is the unique point of interaction of the end user of the SAF pilot. This domain is represented by the handheld device itself and the software deployed within.

Server layer is composed by the elements located and consumed in web style and enables the collection and gathering of data from heterogeneous sources such as those from client side and cloud data layer.

Cloud data layer exposes content in the form of APIs of the Tailored Information System and is in charge of retrieving info from even more heterogeneous data sources such as: supermarket repositories, transport company logs, etcetera. This layer is detailed in next sections.

From the client point of view, access to tailored information is achieved using the web browser and the SmartWebProxy that has been defined at server layer in order to deal with both user preferences and complex access methods to scattered data.

3.1.1.2.1 Client Layer

Current support of HTML5 capabilities is variable for a number of layout engines. As HTML5 specification is still a working draft, not a W3C recommendation, it is not stable. Unless otherwise specified in footnotes, comparisons are based on the stable versions of the programs without any add-ons, extensions or external programs.

As the web browser is a key tool inside the pilot we summarize the state of the art of browsers engines. The Table below displays the compatibility of distict browser engines that can be used for the mobile device.

Table 2. Compatibility of Browser engines that can be used in for the graphical user interface

Layout engine	Release version	Preview version	Used by
Amaya	11.3.1	11.3-pre	Amaya
Gecko	18.0	19.0	All Mozilla software, including Firefox; SeaMonkey and Galeon; Camino; K-Meleon; Flock (up to 2.x); Debian IceWeasel; GNU IceCat (formerly GNU IceWeasel); Icedove, Iceape and Iceowl; Fennec. Also used by the Maple platform in 2010 and 2011 Samsung "SmartTV" devices which are based on the build used in Firefox 3.0.
KHTML		N/A	Konqueror
Presto	2.10.229	2.10.238	Opera; Opera Mobile, Nintendo DS & DSi Browser; Internet Channel
Tasman	(IE 5.2.3 for Mac)	none (aborted in 2003)	Internet Explorer 5+ for Mac OS X
Trident	5.0 (IE 9)	6.0 (IE 10)	Internet Explorer and other Internet Explorer shells like Maxthon (Microsoft Windows operating systems only), Windows Phone 7
WebKit	534.20	r83424	Tizen (both OS and browser), Safari (both desktop and mobile), Google Chrome, Maxthon 3, Shiira, iCab 4, OmniWeb 5.5+, Web, Adobe AIR, Midori, Adobe Dreamweaver CS4 and CS5, Android browser, Palm webOS browser, Symbian S60 browser, OWB, Steam, Rekonq, Arora, Flock (version 3+), RockMelt, Dolphin Browser, Sleipnir, Konqueror†, rekonq. Used by the LG Smart TV platform to enable web content on TVs.

User granted access: In order to make use of the TIC pilot, some actions are required to be taken from user side. Those actions include granting access to access to user-owned resources and devices like the camera or to user data residing in devices. In the development side, this action can be achieved by enabling cookies and Javascript on the web browser prior to use of web app and afterwards when solicited by the web app.

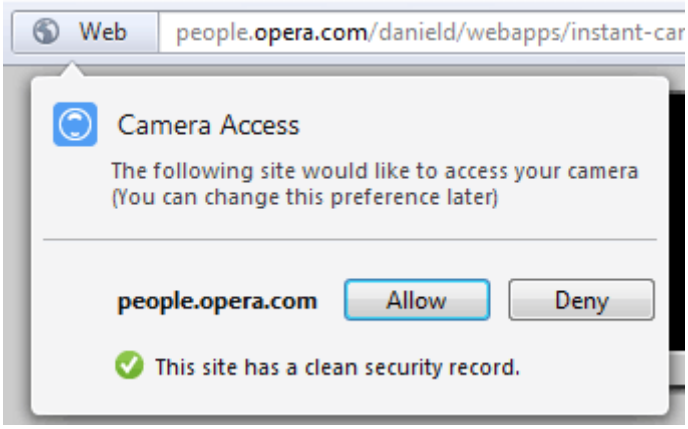


Figure 3. User Granted Access

3.1.1.2.2 Server (middle) layer

Concerning the implementation, an Apache [10] server was selected as the main web server due to its multiple service features, robustness and adaptability for specific needs as its source code is freely available (Apache Foundation license). Besides, there is a large public library of Apache add-ons.

It supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP.

Popular authentication modules include *mod_access, mod_auth, mod_digest, and mod_auth_digest,* the successor to *mod_digest.* A sample of other features include Secure Sockets Layer and Transport Layer Security support (*mod_ssl),* a proxy module (*mod_proxy*), a URL rewriter *(mod_rewrite),* custom log files *(mod_log_config),* and filtering support (*mod_include* and *mod_ext_filter).* Popular compression methods on Apache include the external extension module, *mod_gzip*, implemented to help with reduction of the size (weight) of web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for web applications. Apache logs can be analysed through a web browser using free scripts such as AWStats/W3Perl or Visitors. Hence, these modules can be easily plugged in TIC developments.

Virtual hosting allows one Apache installation to serve many different actual websites. For example, one machine with one Apache installation could simultaneously serve *www.example.com, www.example.org, test47.test-server.example.edu,* etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs). It supports password authentication and digital certificate authentication. Apache has a built in search engine and an HTML authorizing tool and supports FTP. These authentication mechanisms are especially useful in the TIC pilot.

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on Web sites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most Web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). Most web hosting providers support PHP for use by their clients. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use. PHP acts primarily as a filter, taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic Web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a Web server to a client, such as Microsoft's ASP.NET, Sun Microsystems' JavaServer Pages, and mod_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symfony, CodeIgniter, Yii Framework, and Zend Framework, offering features similar to other web application frameworks.

PHP Coding

As a server side technology, PHP leverages the access to different resources, particularly to those constrained by the Same Origin Policy (SOP). Cloud Data Layer APIs are natively constrained by this SOP policy.

The code displayed below shows an example of access to Cloud Data Layer resources.

```
<?php

include '../config/cfg.php';


// userPersonalData - user details
$url = $ip_server.':'.$port.'/services/product?idProd='.$idProd.'&batch='.$batch.'&'.http_build_query($_GET);
$url = $ip_server.':'.$port.'/services/product?idProd='.$idProd.'&batch='.$batch;
echo file_get_contents($url);

var_dump(json_decode(file_get_contents($url), true));


$thejson = file_get_contents($url);
$obj = json_decode($thejson);
```

Figure 4. Use of PHP in the mobile application

Javascript Coding

This client side technology serves as the basis for HTML5 related features of the mobile application side

The most common use of JavaScript is to write functions that are embedded in or included from HTML pages and that interact with the Document Object Model (DOM) of the page. Some simple examples of this usage are:

- Loading new page content or submitting data to the server via AJAX without reloading the page (for example, a social network might allow the user to post status updates without leaving the page)
- Animation of page elements, fading them in and out, resizing them, moving them, etc.
- Interactive content, for example games, and playing audio and video
- Validating input values of a web form to make sure that they are acceptable before being submitted to the server.
- Transmitting information about the user's reading habits and browsing activities to various websites. Web pages frequently do this for web analytics, ad tracking, personalization or other purposes.

Because JavaScript code can run locally in a user's browser (rather than on a remote server), the browser can respond to user actions quickly, making an application more responsive. Furthermore, JavaScript code can detect user actions which HTML alone cannot, such as individual keystrokes. Applications such as Gmail take advantage of this: much of the user-interface logic is written in JavaScript, and JavaScript dispatches requests for information (such as the content of an e-mail message) to the server. The wider trend of Ajax programming similarly exploits this strength.

Javascript has been employed in the mobile application to deal with input data validation (forms, navigation, etc.) and comprises a set of javascript files that provide the support for such operations.

Javascript Implementation

One of the simplest uses of Javascript is the DOM feature provided by the programming language. Javascript has been important in order to achieve access to specific DOM entities as shown below:

```
1.  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2.  "http://www.w3.org/TR/html4/strict.dtd">
3.  <html>
4.    <head><title>SAF-Bonpreu</title></head>
5.    <body>
6.      <h1 id="header">SAF pilot</h1>
7.      <script type="application/javascript">
8.        document.body.appendChild(document.createTextNode('Product info'));
9.        var h1 = document.getElementById("header"); // holds a reference to the <h1> tag
10.       h1 = document.getElementsByTagName("h1")[0]; // accessing the same <h1> element
11.     </script>
12.     <noscript>Your browser either does not support JavaScript, or has JavaScript turned off.</noscript>
13.   </body>
14. </html>
```

Figure 5. Operational Tasks

HTML5 Coding

Among the distinct HTML5 frameworks available nowadays in web technology, Jquery was selected as the main framework in the mobile application due to its capabilities of access to client resources and communication data. Presentation of information on screen is also a top feature of this framework. The code displayed below exemplifies the invocation of Jquery framework.

```
		echo "<script type='text/javascript'>window.location = '../index-menu.php'</script>";
}
?>
<!DOCTYPE html>
<html>

<head>
		<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
		<meta name="viewport" content="width=device-width, initial-scale=1">
		<title>Bonpreu-SAF</title>
		<link rel="stylesheet" href="../css/safstyle1.css" />
		<link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.css" />
		<script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
		<script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script>


 <script type="text/javascript" src="jquery.validate.js"></script>


		<script src="jquery.showpassword.js" type="text/javascript" charset="utf-8"></script>
```

Figure 6. Invocation of Jquery framework

Web app coding.

For the screen Formatting, Jquery was employed since it provides an efficient and smooth user interface.

```
<!DOCTYPE html>
<html>
<head>
		<title>Page Title</title>
		<meta name="viewport" content="width=device-width, initial-scale=1">
		<link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.css" />
		<script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
		<script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script>
</head>
<body>
<div data-role="page">
		<div data-role="header">
			<h1>Page Title</h1>
		</div><!-- /header -->
		<div data-role="content">
			<p>Page content goes here.</p>
		</div><!-- /content -->
		<div data-role="footer">
			<h4>Page Footer</h4>
		</div><!-- /footer -->
</div><!-- /page -->
</body>
</html>
```

Figure 7. Portion of code of Screen formatting

Operational tasks

Finally, the interaction of the distinct layers, depicted in the following image, allows end users to access tailored information through the mobile application where mode of use (anonymous, preferences) can be set and scan of different products located at supermarket premises can be achieved. More details about the codding part of the mobile application can be found in the Annex.

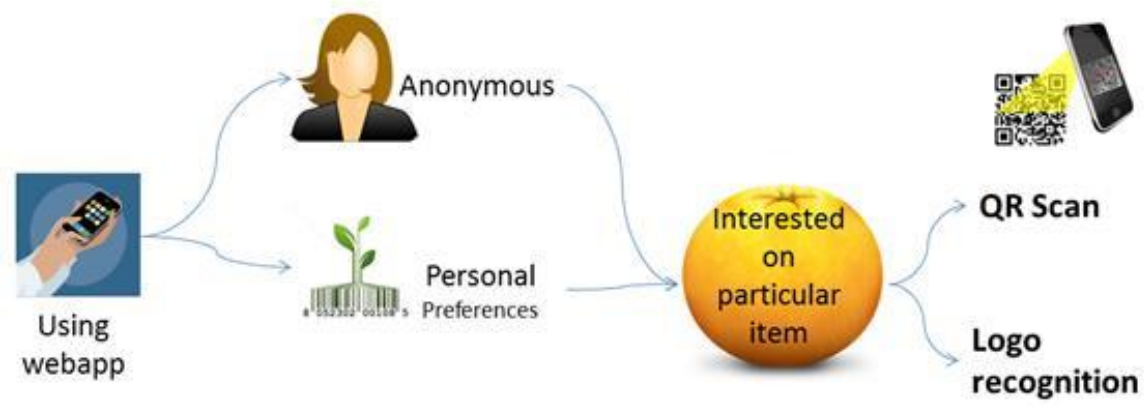


Figure 8. Operational Tasks

3.1.2 Infrastructure Side

The infrastructure part links existing and future product related information databases with consumer needs, and gives to them in the easiest way the tailored information requested. A key issue is to develop a sub-system that allows collecting and managing dynamic information and data generated at different points of the supply chain. This is carried out in the server domain.

3.1.2.1 Server Domain

In the server domain, an additional Mule module has been developed. It provides the means for the mobile application to retrieve Bonpreu's privacy policy, which describes what information, does Bonpreu need from the users of the application and the purpose of usage. As in our case, Bonpreu only requires the information for a single purpose that is, to provide tailored information, and always wants access to the same information; only one privacy policy was created and exposed via the Mule module.

The mule application is called from the GUI to obtain the needed information.

The figure below shows how the web services layer is developed. The ESB selected was Mule ESB[11]. It exposes Restful web services in order to be invoked by any application. The API of these services can be found in annexes. When a request arrives to the ESB, the message workflow (MF) analyses and determines from source of the URL that invokes this message. Next, The

URL filter determines if that URL is compatible with the existing patterns. Once the URL has passed the filter, it arrives to a router (choice component) that addresses the request to the right service. If there is an error, it is redirected to another message workflow that keeps a log of every request.

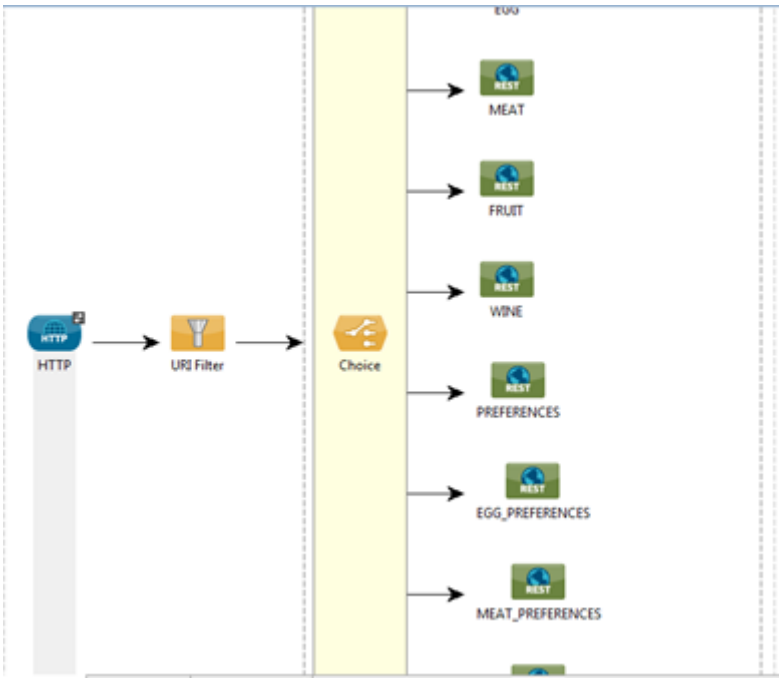


Figure 9. Part of the Message Workflow (MF) of the Web Services layer

The figure below shows the implementation of the Login Resource. The GUI sends to the back-end application the user login and password. After the ESB has routed these requests in the correct way, it arrives to the UserSystemResource where the class UserSystemResource.java determines that the request is a GET method (because it supposed to be the new session establishing). Once inside the correct method, the service invokes the correct module of the architecture (provided and explained in D400.2) and checks if the selected user exists in the database. All the information regarding databases can be found in the annex.

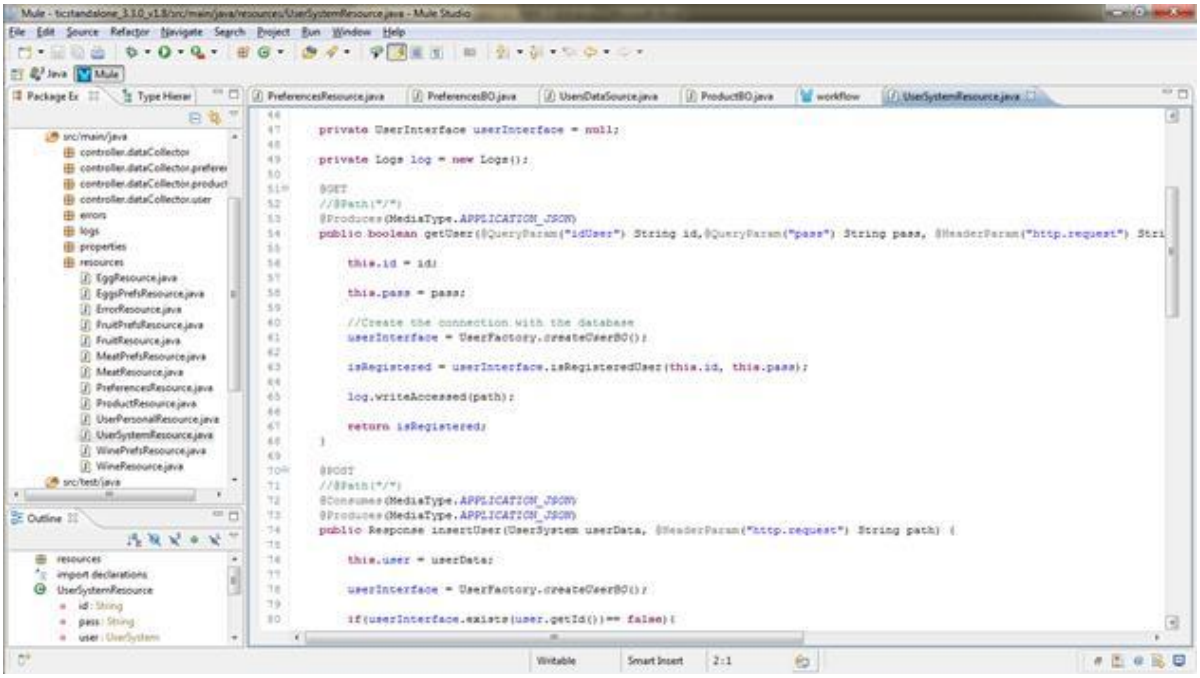


Figure 10. UserSystem Resource

3.1.3 Physical Deployment

The TIC pilot is deployed in Bonpreu facilities located in Calle Sicília 370-376, in the neighbourhood of l'Eixample in Barcelona (Spain).

The supermarket that is located there is a medium size store composed by a dry products section, a bakery and a big fresh food area with fresh fruit and vegetables, a butcher and fish/seafood services. The main client profile of this shop is middle-aged middle-high class consumers, according to the neighbourhood profile.

Above the supermarket, is the Consumers' Space. It is a place where part of Bonpreu communication and marketing department is situated, and is used for consumer-retailer interaction in order to have feedback from its regular consumers (the ones with Bonpreu fidelity card) about different subjects such as new products offered by the supermarket, cooking classes, master classes of nutrition, etc. It is a room with capacity for maximum 25 people with all the facilities for carrying out workshops, talks, cooking classes, and so on.

Facilities in Calle Sicília have been chosen as the best site for pilot deployment because of the following reasons:

- The medium size of the supermarkets represents the best conditions for a prototype test.
- The location of the supermarket and the Consumers' space is in a young and dynamic neighbourhood, so is the profile of its clients. Young and dynamic consumers are more likely to be used and interested in new technologies. Therefore, the panel of consumers consisting of clients of the supermarket will be easy to create and be involved in the project.

- The deployment of all equipment needed for TIC pilot can be set up in the Consumers' space, which provides enough room for it.
- The Consumers' space represents a perfect place for developing TIC pilot tests in a closed and controlled environment, using pilot products from the supermarket and totally equipped for the development of the workshops with consumers.

Figure 11. Entrance at the Consumers' space.

Figure 12. Fresh fruits and vegetables area.

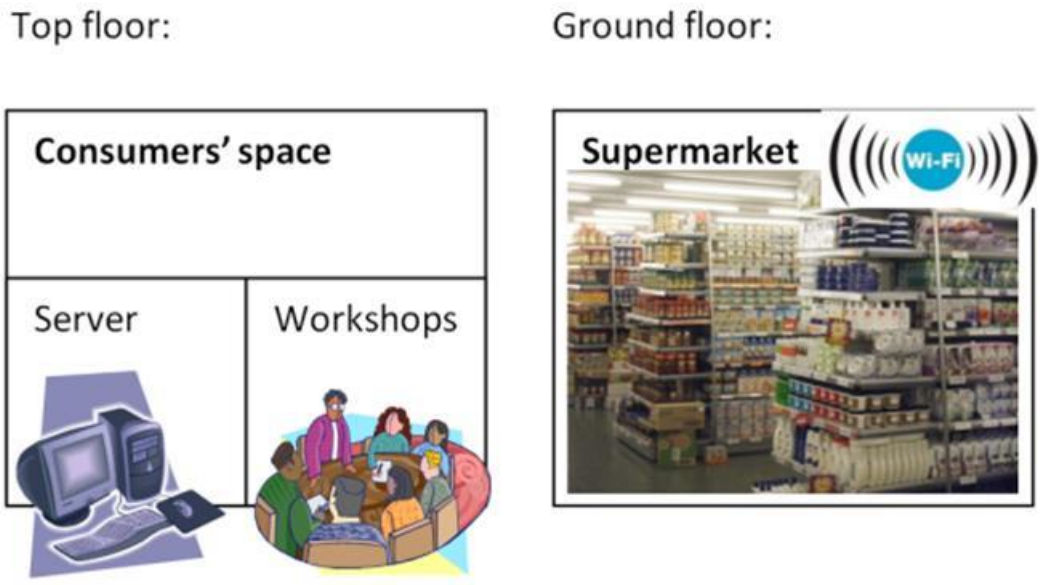


Figure 13. Top and Ground floors of facilities

The computer equipment is installed in a room located in the Consumers' Space, next to the room where workshops with consumers are carried out.

The Wi-Fi network is installed in the supermarket, in the fresh products area.

Both Consumers' space and supermarket receive the Wi-Fi signal, so consumers can access to it using their Smartphone from both sites. In the same way, the server is connected to the Wi-Fi signal in Bonpreu.

3.1.4 Overall System Deployment

About the infrastructure deployment of the TIC pilot, the complete development has been divided into two iterations, the last one taking over the results of the first one. The main difference between the two iterations is that the second one makes extensive use of the Generic Enablers specifications provided by FI-WARE, so the flow of the requests has been modified to use these new functionalities.

3.1.4.1 First iteration

In the first iteration, all the requests launched by the customers of the supermarkets via smartphone are gathered by an ESB server installed in Bonpreu. This ESB server redirects those requests to another ESB server located in Atos, which accesses the database and collects the information required.

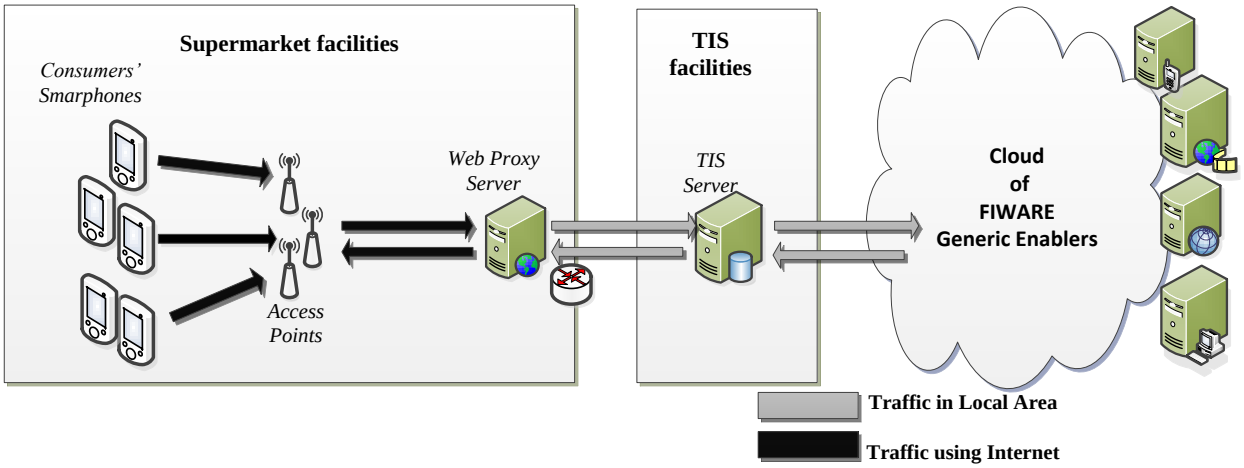


Figure 14. First version of system deployment

3.1.4.2 Second Deployment

In this second iteration, all the infrastructure side is contained in the Data Center Resources Management GE; so all the requests are gathered by this single node which is hosted in this GE.

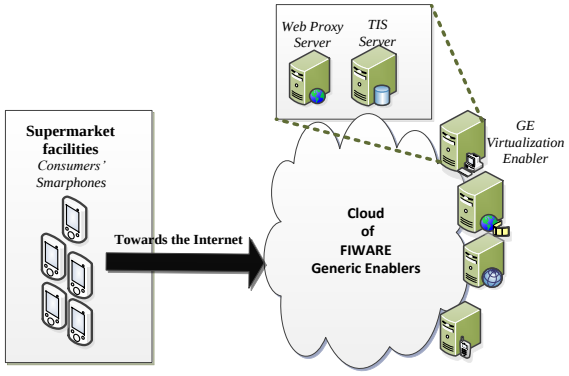


Figure 15. Second version of system deployment with the DCRM GE

3.1.4.3 System Test via Workshops

Here we explain the aim and mechanics of the workshops with consumers that are carried out in Bonpreu. A detailed explanation of the objective, content, and results of the developed workshops will be presented in next deliverable 400.4.

Feedback from end users (consumers) is a key point to assess viability and compliance with their expectations. For this, several sessions have been planned in order to involve consumers in the TIC pilot. These sessions are being carried out by Bonpreu in a special space dedicated for its clients called “Consumer’s space”.

The sessions are called *workshops with consumers*, because of their interactive and open to discussion nature between consumers and retailer.

The objective of these workshops is to involve a panel of 15-20 consumers in all the process of the TIC pilot, its conception, development and evaluation, as to identify their needs and requirements as consumers, then to test and to validate the TIC app mock-ups and the final TIC app.

The results of the workshops would help and assure the feasibility of an open deployment of the TIC pilot in a Bonpreu supermarket. The development of all workshops with consumers in a closed environment will allow detecting and improving the TIC app in order to make the decision for an open deployment in a real supermarket assuring the viability of it.

The mechanics for the development of each workshop follow the following procedure:

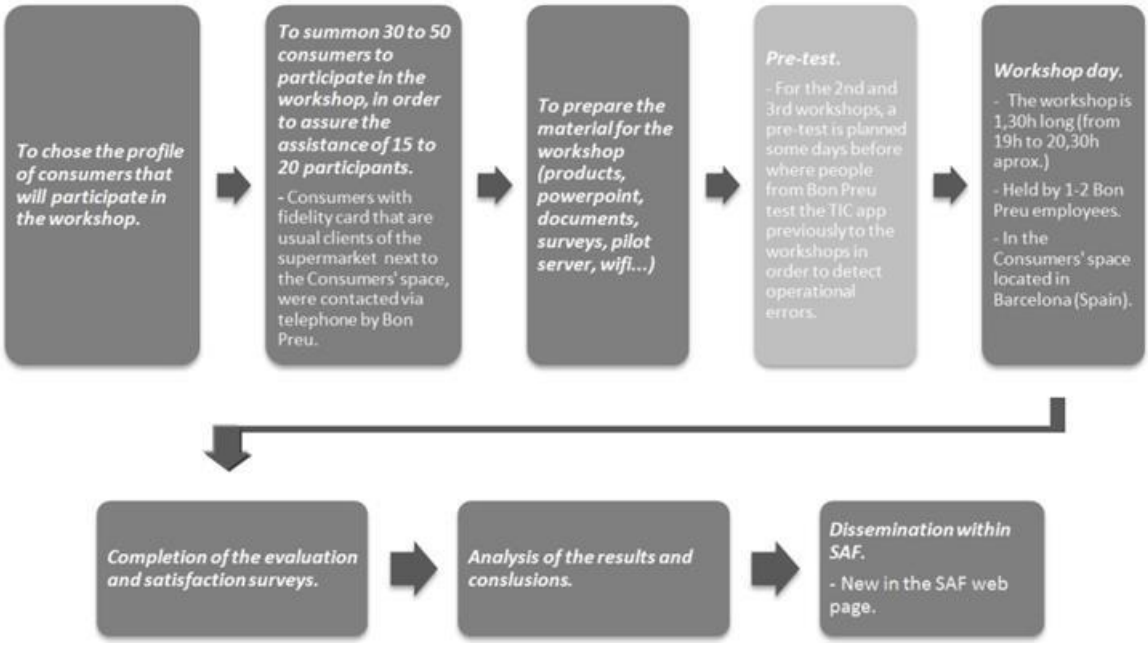


Figure 16. Workshop procedure

3.2 Domain Specific Enablers of TIC Pilot

In this section we include a description of the Domain Specific Enablers that have been employed under WP400.

3.2.1 SmartWebProxy

The SmartWebProxy was previously described in D 400.2 [4]; however we provide an overview for reason of completeness of this document. It is a high level model that allows a mobile device to carry out tasks that are necessary for the tailoring information process, in form of web technologies (HTML5, Javascript) instead of having native applications. Hence, the main objective of this model is to allow seamless information consumption among different mobile environments and overcome the limitations [7] of native applications, in terms of flexibility and operating system support. In the expected supermarket SAF scenarios there could be several consumers using different mobile operating systems (e.g. Android , iOS, Blackberry and so on), at the same time; so it imposes two main considerations (in the form of expected situations) which are reviewed in the following table.

Table 3. Benefits of the Domain Specific Enabler

N	Consideration	Actions of Consumers – mobile device	Actions of the Supermarket - infrastructure system
1	The supermarket is planning to improves the consumers' experience (updating the graphical user interface) and the functionalities (e.g. a new recommendation mechanism), it offers; so it is needed a flexible **front-end application**	Consumers should manually update the application, or trust in an external application delivery platform. Hence, each time the supermarket adds a new functionality the user must update the application itself or via a third party.	The infrastructure should generate new version of the application and upload it to the corresponding markets or application delivery platform (e.g Google play or Apple Store). It must ensure that all the new or even old functionalities still work, since an important part of their reputation and benefits leverage in these scenarios.
2	The supermarket has integrated a new set of functionalities (e.g. using a NFC tags, or augmented reality-based codes), so an update of the **Back-end application functionalities** is needed**.**	Consumers should manually reconfigure the application depending on the capabilities of their mobile devices (e.g. cameras, sensors and so on). They have to set application permits, reconfigure settings and check their operating system compatibility	The infrastructure should generate new version of the application and upload it to the corresponding markets or application delivery platform (e.g Google play or Apple Store). In addition, direct configuration of the applications will be slow, because consumers should wait until the application coordinates with the infrastructure systems (e.g. screen, proximity sensors and so on).

As we explained in table Table 3, using native application, as a tool for having Front-end and Back-end functionalities can slow their adoption; specially when successive changes occur in the supermarket scenarios such as the addition of new products, offers, and technologies (e.g. a smart shopping cart [6]). Hence, as the web proxy allows fetching functionalities and tasks in the form of HTML5 services, the reconfiguration and adaptation of these functionalities can be carried out on-the-fly.

As mobile web browsers improve their level of implementation of the HTML5 standard, the capabilities of the SmartWebProxy [7][4] will be enhanced. This is because all the development tools will be based on well-known web-standards and languages; so it is expected that the SmartWebProxy can offer a seamless integration with all the internal resources of mobile devices (e.g. cameras, sensors), peer resources of other mobile devices and finally infrastructure resources available in the supermarket.

The SmartWebProxy implementation can be supported in the mobile device side or in the infrastructure side. Hence, depending on the capabilities supported by the web browser in a given moment, it can run as an embedded application (in the form of HTML5 code) or as a middle-layer application. Despite this, the user experience and usability, which is one of the targets inside the WP400, are maintained.

The SmartWebProxy allows invoking local resources of the mobile device. Hence, it basically catches the web browser requests and then filters and redirects them to the infrastructure or directly to the Mobile operating system. Chapter 3 describes all the details of the implementation.

3.2.2 Logo Recognition

The different aspects of food quality, safety and integrity are often summarised under specific quality signs that are based on requirements of the quality sign owning organisation. By complying with these requirements agri-food enterprises gain the right to print the quality sign on their products. Consumers often look for these signs, because they are part of their quality expectations. However, the requirements and the dimensions covered by a specific sign are hidden in the descriptions and regulations of the signal owner. The scope of this DSE is to link this hidden information for the most common quality signs and provide it to consumers on demand.

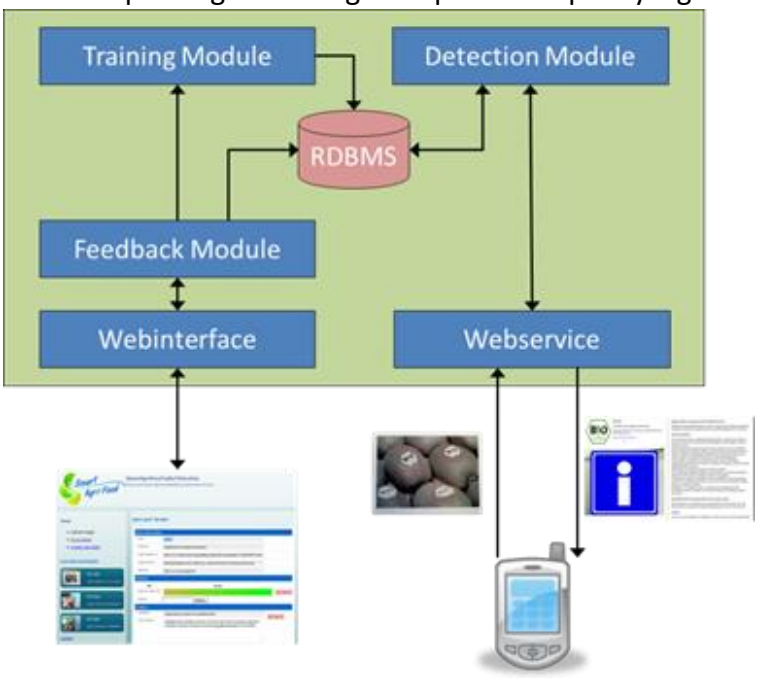


Figure 17. Architecture overview

The vision behind this DSE is that a consumer picks a product box from shelves and scans a specific part of the box where a product label or quality sign is situated with a smartphone. A picture/pattern recognition algorithm identifies the specific quality sign and compares it with available information on the Future Internet including the sign owner and the manufacture of the product. The consumer receives additional information about the sign and the requirements that have been fulfilled by the product.

3.2.2.1 Architecture

The DSE consists of five main modules (see

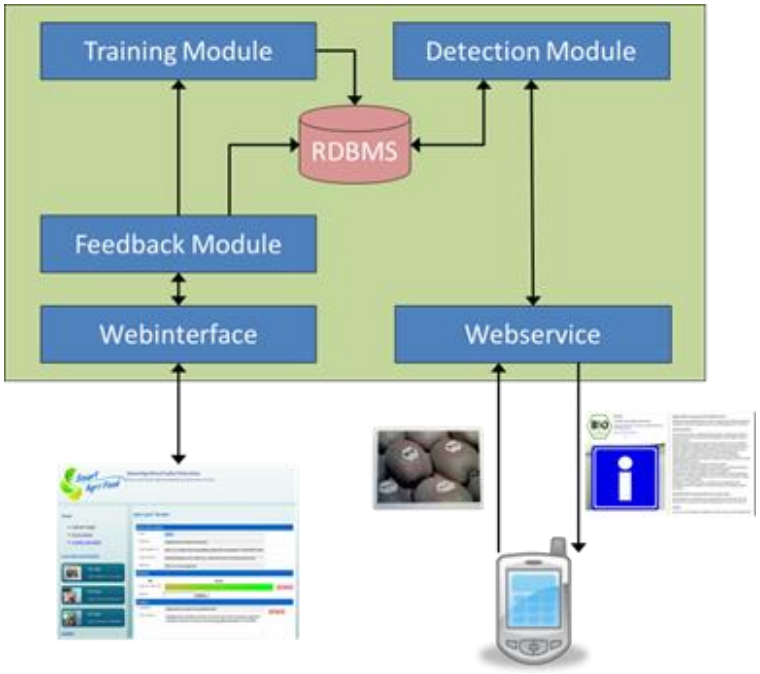


Figure 17):

- **Webinterface:** Provides basic functionalities to configure the DSE and allows the training of the algorithm. The latter procedure is described in Section 3.2.2.2.
- The **Feedback Module** allows the user to evaluate the results of the detection module.
- The **Trainings Module** takes the information gathered by the Feedback module and uses them to train the detection algorithm.
- **Detection module:** The module uses the trained algorithm to recognize labels on images and returns the position of the label and label related information (e.g. animal welfare).
- The DSE offers its detection service via a SOAP-based **Webservice** to the SmartAgriFood project and its pilots. This service accepts a picture (encoded as JPG, BMP or PNG) and returns an array of detected logos including their position, master data (e.g. name of the logo, logo owner, etc.) and a set of criteria (e.g. animal welfare), which the product fulfils.

3.2.2.2 Training

To ease the integration of new logos the DSE uses a supervised learning algorithm to include new logos and continually improve the detection quality. The acquisition and use of needed data is done in two modules: the feedback service and the training module.

The feedback service is a quite important module to improve the detection quality of the used algorithm. It allows the user to classify the detection result of the algorithm by dragging the detail into the green or red box (see

Figure 18). This classification is automatically used for the next training iteration. Furthermore it is also possible to mark and categorize labels which were not detected by the algorithm. This is achieved by selecting the label on the image and pressing the plus-button inside the yellow box.

Figure 18. Feedback interface

The training module creates an automatic set of positive and negative samples to train the detection algorithm from the user-rated and manually marked labels. This allows a community driven creation of a huge amount of test samples, which is needed to achieve a sufficient recognition quality by supporting an incremental learning approach (see Figure 19). Kuranov et. al for example used a set of about 5000 positive patterns and 3000 negatives for training their face recognition system [8]. Having the number of available labels in mind it becomes clear that a massive amount of test and trainings samples must be created to achieve and acceptable result.

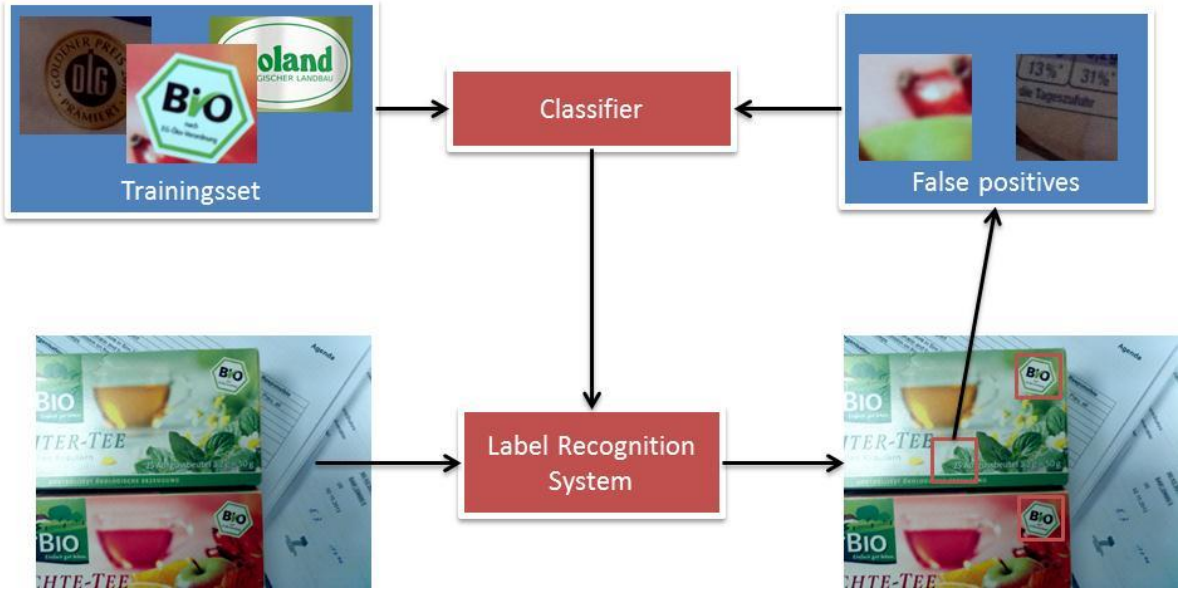


Figure 19. Incremental learning to improve performance

Another big challenge is to find the best settings for the training algorithm including the set of haar-like features, the sample size, number of training stages etc. It appears that the effect of these settings is often hard to predict, which leads to an empirical analysis for each label.

3.3 Related FI-WARE's GE to the TIC Pilot

For reason of completeness of this document, we summarize a table with the list of GEs that have been integrated in the TIC pilot. A detailed description of the mechanisms for communicating Domain Specific Enablers and GE can be found in D500.5.2. Regarding the next phases of the TIC pilots, some developments and functionalities within the TIC pilots are similar to those ones intentent to be provided by GE. However, at the time of the deployment phase they were not ready or even designed; thus, future efforts will be focused on integrating these GE in the TIC architecture.

Table 4. Related FI-WARE's to the TIC Pilot

Generic Enabler	Integration in TIC	Exploited by Module
Iaas Data Center Resource	Yes	Tailored Information System
Identity Management GE	Yes	Mobile Application
Data Handling GE	Yes	Mobile Application and its authentication system
Identity Management	Yes	Mobile Application and its authentication system

3.4 Validation of the TIC Pilot

This section will match previously defined functionalities and requirements within the scope of the pilot with the developed modules and classes. There are several functionalities and requirements that have been extracted from the scenarios that target WP400 and necessary

building blocks. As this information is already present in Deliverables 400.1 and 400.2, we limit this section to summarize which of those modular components, support those functionalities and how they are currently developed and supported under the TIC pilot, in both, architectural design and implementation.

Table 5. Comparison of Functional Requirements and Developed Modules.

Building Blocks	Integrated and Implemented Modules	Comment
User Profile Provider	Identity Management GCP GE	Identify management controls the way users define policies regarding preferences and privacy
Product Information Provider	Tailored information system	The information provider retrieves the information regarding products; so even if the matching occurs outside of the domain of the provider, as an example in the supermarket, this building block can work. As an example of this the front layer of the mobile application is decoupled from the database which stores product data.
Label Provider	Logo recognition tool	This functionality is fully implemented by the logo recognition tools.
Uniform mobile application development	SmartWebProxy	The SmartWebProxy architecture which is implemented using fully HTML5/Javascript codes, allows users to make use of the application using only web browsers.
Internal capability access	SmartWebProxy	SmartWebProxy allows to define which internal capabilities, such as cameras, will be used inside the TIC pilot.
Tailored information access	TIC and SmartWebProxy	The implementation of the architecture has proven the capabilities of these two developments to provide tailored information access.
Dynamic Service Adaptation	Front-end of mobile web application	Because all the developments are web-based it would be easy to modify the behaviour of the mobile application, rather than force users to download and install updates of the applications (e.g. through Google Play or Apple Store)

3.4.1 Towards future TIC scenarios and adaptation challengues

This section reviews how current TIC pilot architectures and developments could be adapted to support technologies that could enrich upcoming supermarket scenarios. These scenarios aim for a better shopping experience, information visualization, product identification, tailoring information and "prosumerization" [9].

Some of these scenarios will be tackled in the second phase of the SAF project.

Table 6. Adaptation of scenarios

Scenarios and Technologies	Core TIC Components Involved	Adaptation
Visualisation of information through Augmented reality	• Mobile Device – SmartWebProxy • Tailored Information System	The SmartWebProxy allows not only to support future HTML5-base code to overlap virtual objects with information (provided by the TIS), but also the architecture allow including calls to native code, which could be necessary to recognize new image patterns in low-capability devices and not-compatible browsers.
Improving Shopping experience using Gesture Identification	• Tailored Information System • Mobile Device - SmartWebProxy	The TIS could be adapted to interact with devices with multiple sensors that detect gestures (e.g. Kinect); so, a user can point at a product and the TIC can mix de information received by the device with the existing information. Finally, the TIC can push this information to consumers' mobile devices. Hence, this kind of scenario will only require slight modifications to the existing data format.
Extension of shopping trolley and integration with personal devices.	• Tailored Information System • Mobile Device - SmartWebProxy	An extended shopping trolley with screens or any other human interface can also be integrated with the TIC pilot. In the same way, as the Gesture identification scenario, the trolley can interact with the mobile device for fetching information regarding shopping lists, profiles or preferences. This exchange process could be easily plug-gin in throught NFC, Bluetooth or any other short range wireless technology, that are indeed supported by the HTML5 standard.
Integration of product identification technologies (RFID, barcodes, logo recognition etc.)	• Tailored Information System • Logo Recognition Tool • Mobile Device - SmartWebProxy	The product identification has been strongly tackled in the development; so any other kind of technology will only need an input system. A different camera software, or a new RFID sensor can be enough for this extension since the abstract information model. Even the Logo Recognition Tool has been designed to accept several photos of the same logo, learn and correctly detect them (even photos with different angles). In the pilot there is also a mobile applica-

		tion that helps supermarket staff to take photos of logos, mark and introduce information.
Prosumerization of supermarket scenarios.	• Tailored Information System • Mobile Device - SmartWebProxy	Mobile Device must integrate a composition environment which enables service personalization for end-users. Service publication, search and discovery and updating mechanisms must be implemented by the SmartwebProxy. Integration of TIC information system with current supermarket infrastructure (customer identification, accounting, billing, notification processes, etc). This is needed to develop self-contained components to be used in service creation and personalization processes by end-users following the prosumer approach. The composition and Mashup feature of FIWARE, extended with the functionalities of the second phase SAF platform will provide the graphical user interface for customers to customize and personalize provided services.
Enhanced tailored information through the integration of personal data.	• Tailored Information System • Mobile Device - SmartWebProxy	Future scenarios expect the integration of multiple information sources, especially from consumers. The TIS has been designed for accept “plug-and-play” information sources. Taking as example a mobile device that stores health information from the user (e.g. fetched from a portable glucose sensor) , it could be possible (implementing all the privacy mechanism an law) to take this information and build an adapted shopping list at the right moment. Hence, since the TIS and SmartWebProxy are designed and implemented keeping in mind ubiquitous web technologies, these sort of scenarios will be very straightforward atthe communication level

As the TIC pilot development is iterative, we have made public two versions of the mobile application one stable and another for testing.

1. Stable version: http://safprototype.no-ip.org/
2. Testing version: http://safprototype2.no-ip.org/saf3/

3.5 TIC Pilot and Standardizations

The term “standardization” can be interpreted in several ways:

- In test theory, standardization refers to measurements or assessments conducted under exact, specified, and repeatable conditions.
- In supply chain management, standardization refers to approaches for increasing homogeneity of part, process or product, reducing variability found in having many non-standard components.

In the case of this pilot, standardization could be interpreted as the guarantee that the steps and/or procedures used are well specified and don’t vary from one stakeholder to another, and allow to reach the pilot objectives in repeatable conditions. This could lead to a pilot certification according to some specifications (standard).

To do that, the main and key steps and/or procedures have to be identified and control specifications or (in site) “*checkings”* have to be implemented to ensure the compliance of the objectives resulting from those steps and procedures (results) in common and repeatable conditions.

3.5.1 Key steps and/or procedures and associated control for standardization

- Collection of product information

⇨ Control specification 1: ensure product information reliability

- Collection of criteria signs and logos

⇨ Control specification 2: ensure labels/logos information reliability

- Collection of user information

⇨ Control specification 3: ensure consumer anonymity

- Access to consumer information/preferences by filters (access profiles)

⇨ Control specification 4: ensure filtered information/preferences is not accessible (in site checkings)

- Matching user data with product data:
  - Comparative research: scanning of various products and obtainment of the product(s) that better match(es) the consumer’s preferences
  - List research: obtainment of a list of products that match the consumer’s preferences (without scanning)

⇨ Control specification 5: ensure reliability of suggested information (in site checkings)

- Virtual shopping basket

⇨ Control specification 6: ensure correspondence virtual/real basket contents (by matching weight)

Figure 20 shows the procedures

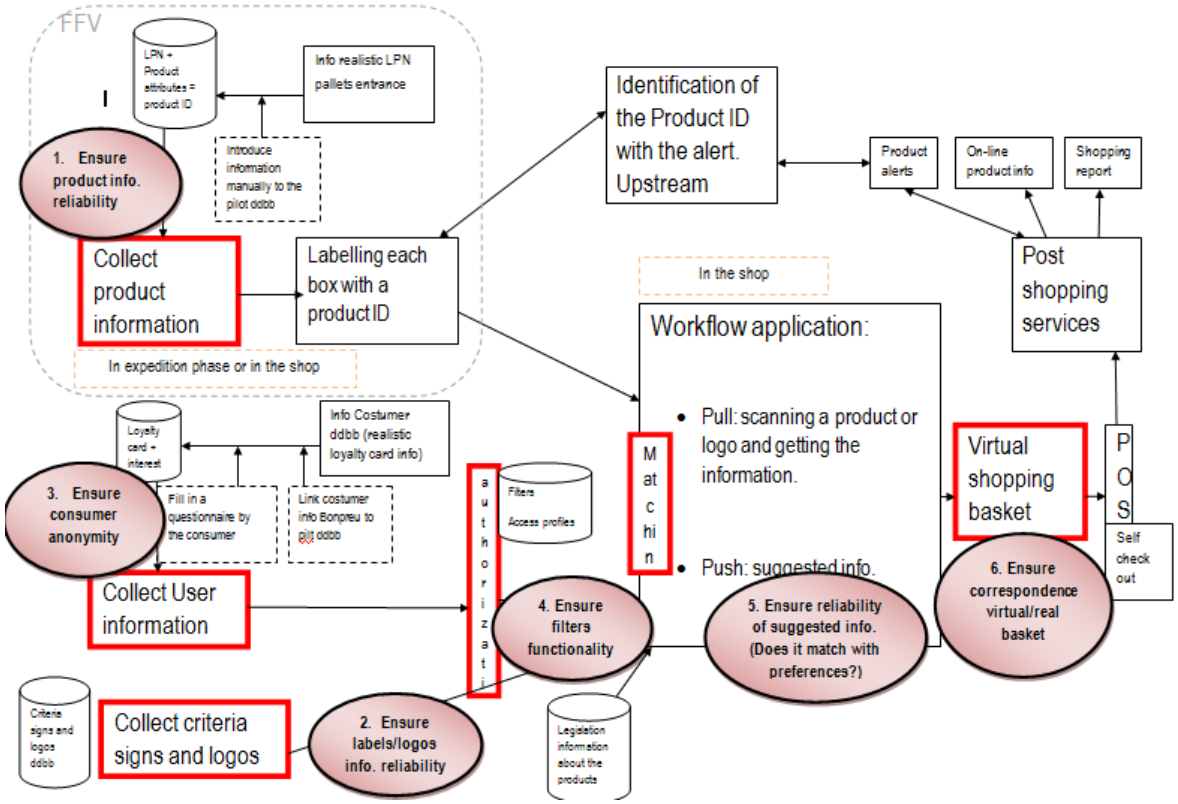


Figure 20. Key steps and/or procedures and associated control for standardization in the TIC pilot.

3.5.2 Control specifications description

3.5.2.1 Control specification 1 & 2: Ensure product & labels/logos/certifications' information reliability

This aspect is not considered for this pilot since the product information is entered manually, but the control infrastructure of the "Certification Services" described in section 3.3.1 of the deliverable D500.3 "Specification on network elements and functions of Core Platform" could be implemented to cover this aspect.

3.5.2.2 Control specification 3: Ensure consumer anonymity

Anonymous commercial transactions can protect the privacy of consumers. Some consumers prefer not to reveal their consumer profile, to prevent sellers from aggregating information or soliciting them in the future. Loyalty cards are linked to a person's name, and can be used to discover other information, such as postal address, phone number, preferences and so on. So, in order to preserve the anonymity of the consumer's profiles, some control specifications have to be implemented.

Specifications/procedure:

- Assign a numeric code (Id code) to each consumer identity (consumer Id) and elaborate an electronic file (consumer Id data base) with the corresponding numeric code/consumer Id.

- Protect this file (access rights) and assign a personal login to each authorized personal to access the data base.
- Change the personal login each month.
- Assign responsibilities and access authorizations to make available this file to a restricted number of persons (authorized persons).
- Only use the numeric code to identify/differentiate the consumer preferences and in all the processes of the pilot.
- Use the electronic data base with the correspondences only in the case of a safety risk or quality problem and only by the authorized personal in order to send an "alert message" to the consumer(s) concerned and identified.

Document/file:

- Create an electronic data base with the correspondence numeric code/consumer Id (data base).
- Keep actualized this data base (actualize the date if any changes).
- Create a document with the personal authorized to access this file (authorized personal' document):
- Keep actualized this document (actualize the date if any changes).

Records/Backups:

- Maintain a record of data base and document.

Responsibilities/training:

- Data base is elaborated by a management member part of the retail company (Bonpreu).
- Authorized personal document is elaborated by a management member part of the retail company (Bonpreu).
- The personal authorized to access the electronic data base is part of the retail company (Bonpreu's personal).

3.5.2.3 Control specification 4: Ensure filters functionality

In relation with the assurance of a good functionality of the filters, **in site checkings** must be implemented into the pilot to verify that only the information allowed by the consumer is accessible by the application.

3.5.2.4 Control specification 5: Ensure reliability of suggested information

In relation with the assurance of the reliability of the suggested information provided by the application based on the consumer preferences, **in site "checkings"** must be implemented dur-

ing the pilot to verify that the suggested information is reliable according to the consumer preferences.

3.5.2.5 Control specification 6: Ensure correspondence virtual/real basket contents

In order to check that the virtual basket contains the same products that the real basket, information about each product weight has to be managed to allow a final check by matching weight between the virtual and the real basket.

3.5.3 Data model and standardization

The TIC pilot requires data to be collected, managed and displayed along the whole food chain so that the information is available for the consumer according to his needs. This includes data for the identification of the product (unique identifier for the product, the supplier, the batch identifier etc.) and additional attributes which describe a specific product in more detail to meet the customer's need for information. All these data fields have to be filled as the product advances along the food supply chain. It has to be investigated how the information available at a specific step can be transformed into an attribute stored in the database. The data model proposed in the TIC pilot (see Table 7) lists the product data.

Table 7: Product data in the TIC pilot

Product	Idenfier	Comments
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	type	Indicates the type of the product (E=Egg, M=Meat, F=Fruit)
	variety	Variety of the product.
	supplier	Supplier of the product.
	batch	Batch number of the set of product.
	origin	Origin of the product.
	ecological	Indicates if the product is ecologic.
	local	Indicates if the product from the current location.
	ecoPacking	Indicates if the pack of the product is ecological.
	footprint	Indicates the carbon footprint of the product.
	dop	Indicates if the product has a designation of origin.
	igp	Indicates if the product has a protected geographical indication.
	additives	Indicates if the product contains additives.
	pesticides	Indicates if the product contains pesticides.
	preservatives	Indicates if the product contains preservatives.
	quality	Indicates if the product has a quality certificate.
	caducity	Indicates the date of expiry of the product.

Product	Idenfier	Comments
	allergens	Indicates if the product contains allergens.
	transgenic	Indicates if the product contains transgenic.
	halal	Indicates if the product conforms to Muslim dietary laws.
	kosher	Indicates if the product conforms to Jewish dietary laws.
	productionDate	Indicates the product's production date.
	price	Indicates the price of the product.
	inDate	Date on which the product arrive to the supermarket.
	outDate	Date on which the product was out of stock.
EggsFeatures		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	raising	Indicates the raising number for eggs (0-3).
MeatFeatures		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	welfareCert	Indicates if the product has an animal welfare certificate.
FruitsFeatures		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	season	Indicates if the product is seasonal.
WineFeatures		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	variety	Indicates the variety of the product.
	tastingNote	Indicates the note of taste of the wine.
	marriage	Indicates the kind of food that can combine with the wine.
Allergens		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	eggs	Indicates if the product contains eggs.
	nuts	Indicates if the product contains nuts.
	lactose	Indicates if the product contains lactose.
	gluten	Indicates if the product contains gluten.
NutritionalInfo		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.

Product	Idenfier	Comments
	kcal	Indicates the number of kilocalories of the product.
	greasy	Indicates the fat content of the product.
	carbohydrates	Indicates the carbohydrates contents of the product.
	sugar	Indicates the sugar content of the product.
	proteins	Indicates the proteins content of the product.
LogisticInfo		
	Products_idProd	Unique identifier for the product.
	Products_Batch	Indicates the product's batch.
	idPoint	Auto incremental Id.
	Latitude	Indicates the geographic north-south point.
	Longitude	Indicates the geographic east-west point.

The TIC pilot uses generic standards for the data exchange, namely HTTP for the protocol and JSON for the syntax. Concerning the semantics of the proposed data structure, the challenge is to identify existing labels for each attribute. For example, a number of labels exist in the EU and the member states which certify ecological production according to specific certification standards. The label used for a specific product could be stored in the field "ecological". However, a standard vocabulary has to ensure a common understanding on the meaning of each label. A challenge will also be to integrate existing standards used for the identification of the product. This data format proposal forms a sound basis for the further development of the pilot and can be envisioned to advance towards a standard for product description. This will require a more detailed analysis of each attribute and a common understanding by all stakeholders on its meaning.

4 TTAM Pilot Specification

4.1 High Level View of the TTAM Pilot

From the technical perspective the TTAM pilot resembles the TIC pilot in that it enables consumers to request about product information using their smart devices (smartphones and tablets) during and after shopping process. It differs from the TIC pilot in a number of aspects. First, the TTAM pilot is about tracking, tracing and awareness of a more specific product - meat. In TTAM we aim to address the requirements of recent regulatory requirements (e.g. EU Reg. No. 1169/2011) and increase consumers trust in meat by providing trustworthy and certified information. Second the TTAM pilot covers the whole meat supply chain from farm to retail – in contrast the TIC pilot focuses on consumers' interactions with retailers. Particularly, we aims to demonstrate how to provide more information on the provenance of the meat (place breeding, slaughter, deboning, *etc.*) and other attributes, such as recipes to improve consumers awareness of the diverse attributes of meat. Third, the TTAM pilot is based on existing and proven technology already in use in a German meat supply chain.

For the TTAM experimentation, the focus was mainly on beef. This meant that we excluded sausages, minced and diced meats, as well as pig, chicken and other types of meat from the TTAM pilot. Also we restrict ourselves to packed beef. We focus on five group of information, which are: general information, origin, quality, production and recipes. This is achieved by gathering traceability and transparency information from all partners of the supply chain in a centralized transparency database maintained by a third party. Instead of building a completely new system, the TTAM conceptual prototype builds on an existing application called fTRACE.

Table 8. Comparison of the initial fTRACE transparency system with the proposed final version of TTAM.

	The initial fTRACE system	The final fTRACE system
1	Managed by a supply chain actor	Managed by a trusted third party (GS1)
2	Specific supply chain	Any meat supply chain
3	Data gathered at the end by one of the supply chain actor	Data gathered from each supply chain actor by the trusted third party
4	Trusted data	Certified data
5	Tracking and Tracing of large batches	Tracking and tracing of small batches
6	Uses proprietary standards	Uses global (GS1 standards) standards
7	Uses proprietary components	Uses generic enablers

The initial and the final scenarios of fTRACE are described in Table 8. The path from the initial to the final version takes place in two stages. The current state of the fTRACE system, depicted in Figure 21, is an intermediate stage. The first two goals of the pilot are realized. The transparency system initially developed by a meat processor for own supply chain is now managed by GS1

(a trusted third party). Though the pilot focuses only on beef, the transparency system now provides information about fish and chicken meat products. The third, fourth and fifth objectives are being implemented and some aspects of these objectives are already functional. The functional design for the final version is developed in D400.2 is revised based and the final version will be fully implemented in the follow up project.

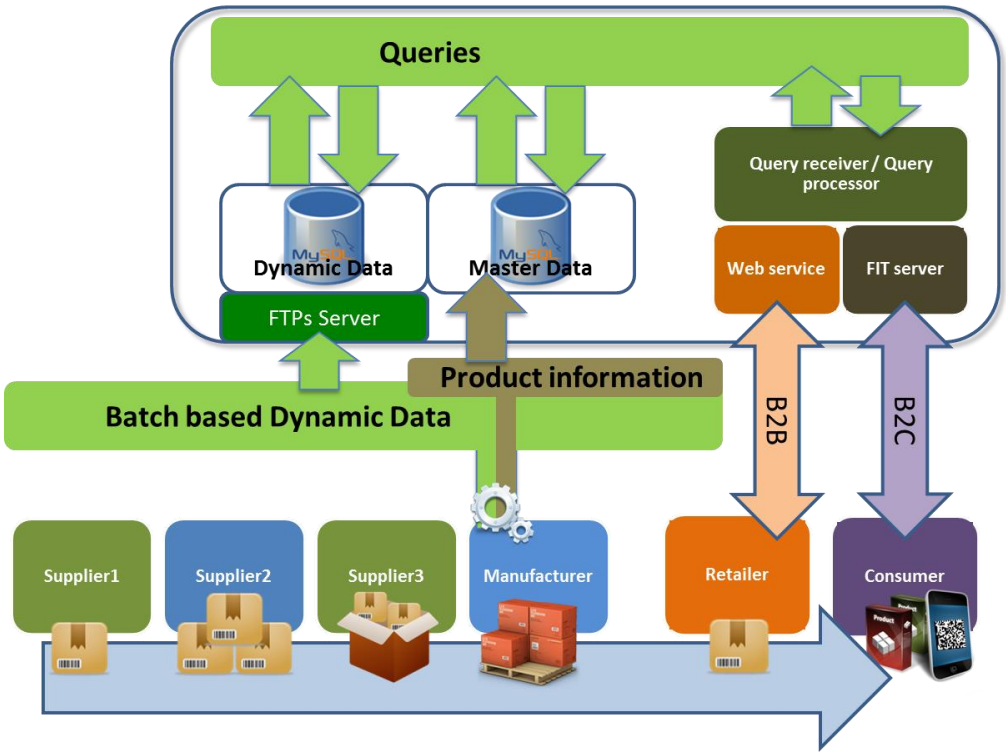


Figure 21. The architecture of fTRACE of the TTAM pilot

From the technical perspectives the various attributes can be divided into dynamic and static (called master in the figure) data. Static data is data that doesn't change from one particular piece of meat to the other, for instance, recipe and production. Dynamic data on the hand is product attribute that is specific to specific piece of meat, for instance, origin and expiry date. Queries mean the transparency data query requests from users. Queries are mainly done by scanning a 2D Barcode using a smart device. The 2D Barcode contains the web address of the fTRACE application and two data items. The first data item is the ID of the producer (depicted in the figure as *Manufacturer*) of the product and the second is the unique product item code. In our pilot the manufacturer is exclusively the meat processor which labels and delivers packed meat products. The majority of users are consumers, however, in some cases retailers are also users. Retailers use the fTRACE not from a mobile device through B2B connection using web services. The main use scenario of B2B is for the purpose of integrating fTRACE data with own mobile application. This last is used to integrate the TIC pilot with TTAM so that customers of Bonpreu can use a single mobile application for both functionalities.

4.2 Domain Specific Enablers of TTAM Pilot

In TTAM the following domain specific enablers are used/implemented:

1. 2D Barcode *reader*: Widely available 2D Barcode readers are the primary means of reading the 2D Barcode-encoded query from meat packages. The 2D Barcode contains also the address of the fTRACE server and thus the 2D Barcode reader and the default browser of the device together the inbuilt camera are used to send a query the fTRACE server.
2. *Database*: Relational data bases are used to store static and dynamic data. The large volume of dynamic data means the database needs to saleable.
3. *Server query cache*: The fTRACE server handles large volumes of querying. It is vital to cache query requests to serve identical queries from the cache in the web server instead of making unnecessary database query.
4. *B2C query module*: The B2C query module is a web service that generates HTML5 document from consumer query response to be sent to the user's mobile device.
5. *B2B query module*: The B2B query module is web service that generates XML document from business partner query response to be sent to the web server of the partner.

4.3 Related FI-WARE's GE to the TTAM Pilot

A detailed and deep deliberation has taken place which GE's could be appropriate for a mature product as fTRACE. Table 6 provides the list of generic enablers that could be appropriate for TTAM. However, none of the generic enablers were found necessary for the TTAM case. In addition integrating prototype implementations of generic enablers into a mature product that is actively being used by many users was considered a risky undertaking.

In the final release of fTRACE, however, an EPCIS (a global standard) enabler will be implemented. The application area of an implementation of EPCIS can however be used in many domains (logistics, faming, transparency system, warehouse management, to mention the few). We, therefore, would like to present the EPCIS implantation that is being undertaken as a generic enabler.

Table 9. GE's considered for the TTAM pilot

GE category	GE
Application/services frameworks	Application Mashup - Wirecloud
Data/Context management	Publish/Subscribe Context Broker
	Complex Event Processing (CEP)
Security	Identity Management - GCP
Cloud hosting	IaaS: Data Center Resource Management

4.4 Validation of the TTAM Pilot

A dummy meat package complete with label and 2D Barcode is used for testing instead of real meat for obvious purposes. Dummy data sets were prepared and made available through fTRACE server. The screen shots in this section are based on the dummy product and data. We would like to stress though that the use of a dummy product and data sets represents a fully realistic scenario since the system makes doesn't distinguish the dummy product from a real one.

4.4.1 The Smart device interface

Basically, it is negligible which Smartphone vendor or which operating system it has installed. It needs to have an internet connection (Wifi or GSM) and an application which can encode 2D-Barocdes. The only hardware requirement is that the Smartphone has a resolution minimum of 2 MP.

Tested and widespread free barcode reader apps are: ZXing, Scanlife, Barcoo Reader or i-nigma. Those barcode reader apps can read the fTRACE URL, which is encoded in the barcode, and forward it the standard web browser.

The fTRACE system has also its own app. It has its own Barcode Reader Encoding Engine and displays the result of the URL, which is encoded in the barcode, in an own web object. In addition to it, the app has the option to enter fTRACE Ids manually:

Figure 22. Barcode Reader in fTRACE App

Figure 23. Manual input in fTRACE App

4.4.2 The Data

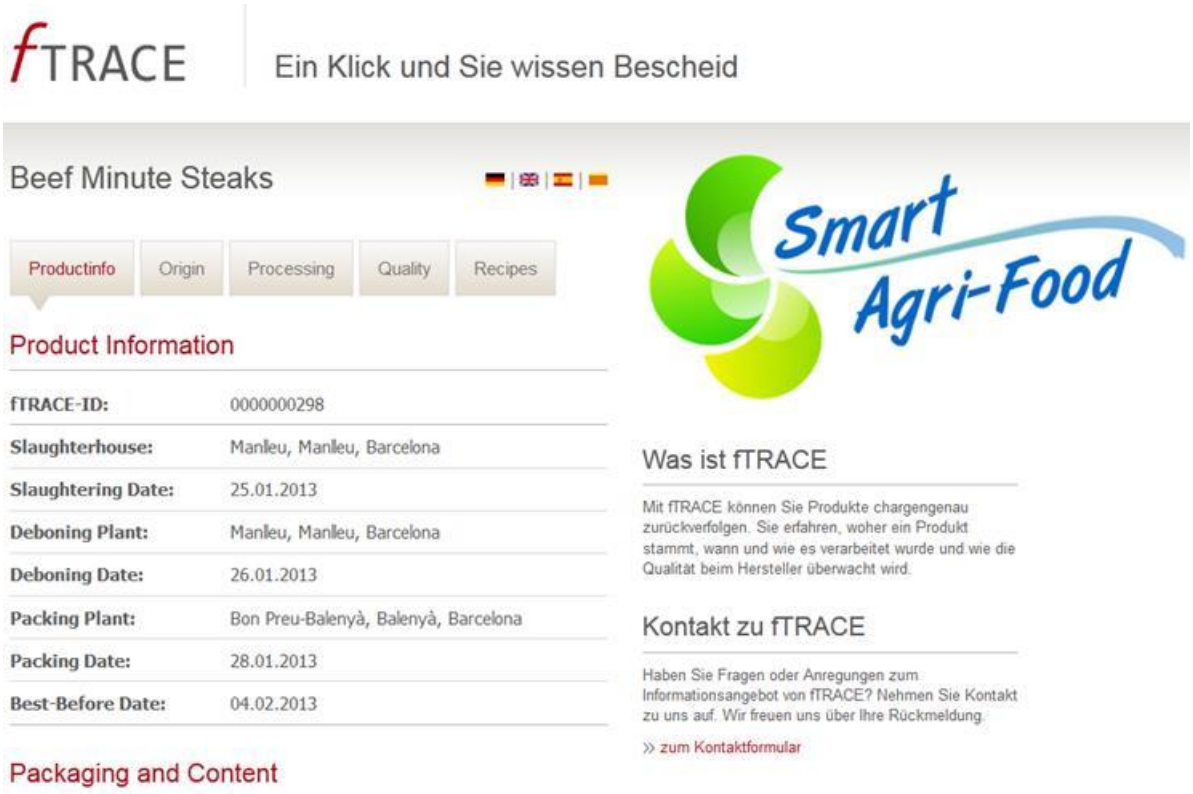


Figure 24. Dynamic product information e.g. slaughter, date of slaughtering, etc. on beef minute steaks collected and provided by fTRACE (also available from http://www.ftrace.de/?h=GS&i=0000000298).

Figure 25. Dynamic product information on origin of the minute steaks collected and provided by fTRACE.

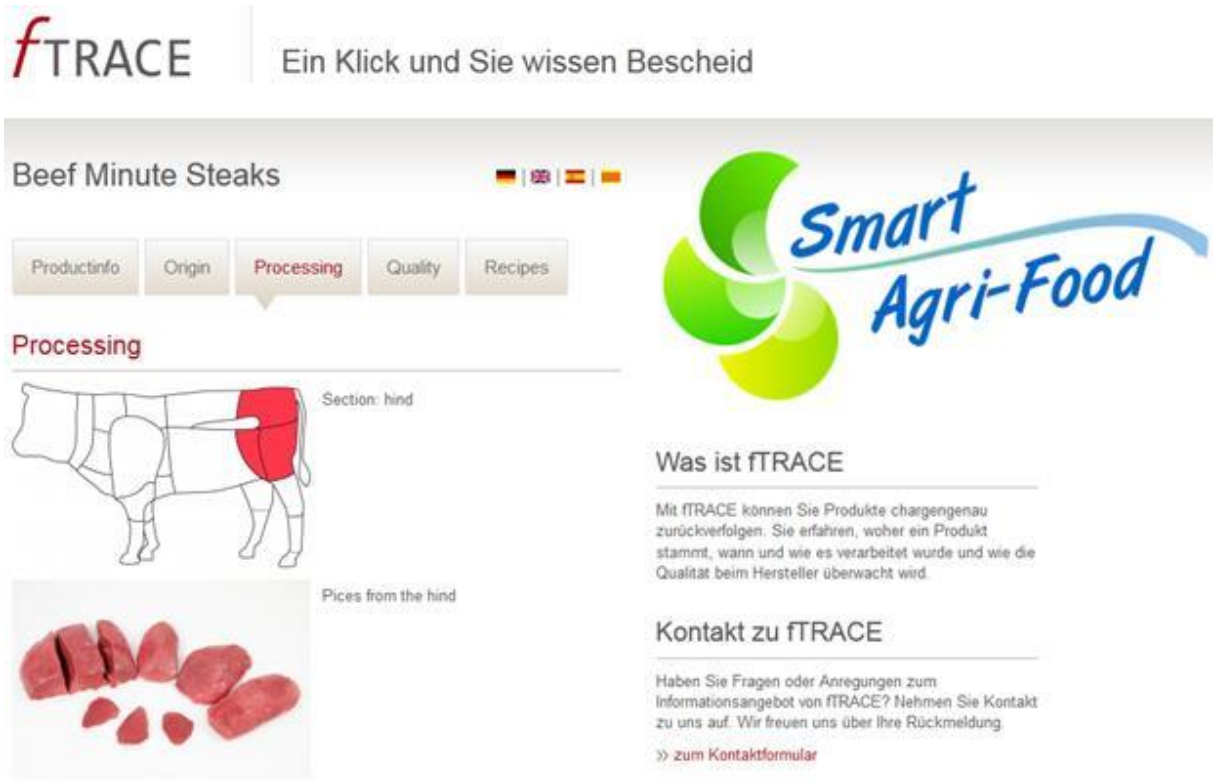


Figure 26. Dynamic product information primal cut of the minute steaks collected and provided by fTRACE.

Figure 27. Dynamic product information primal cut of the minute steaks collected and provided by fTRACE.

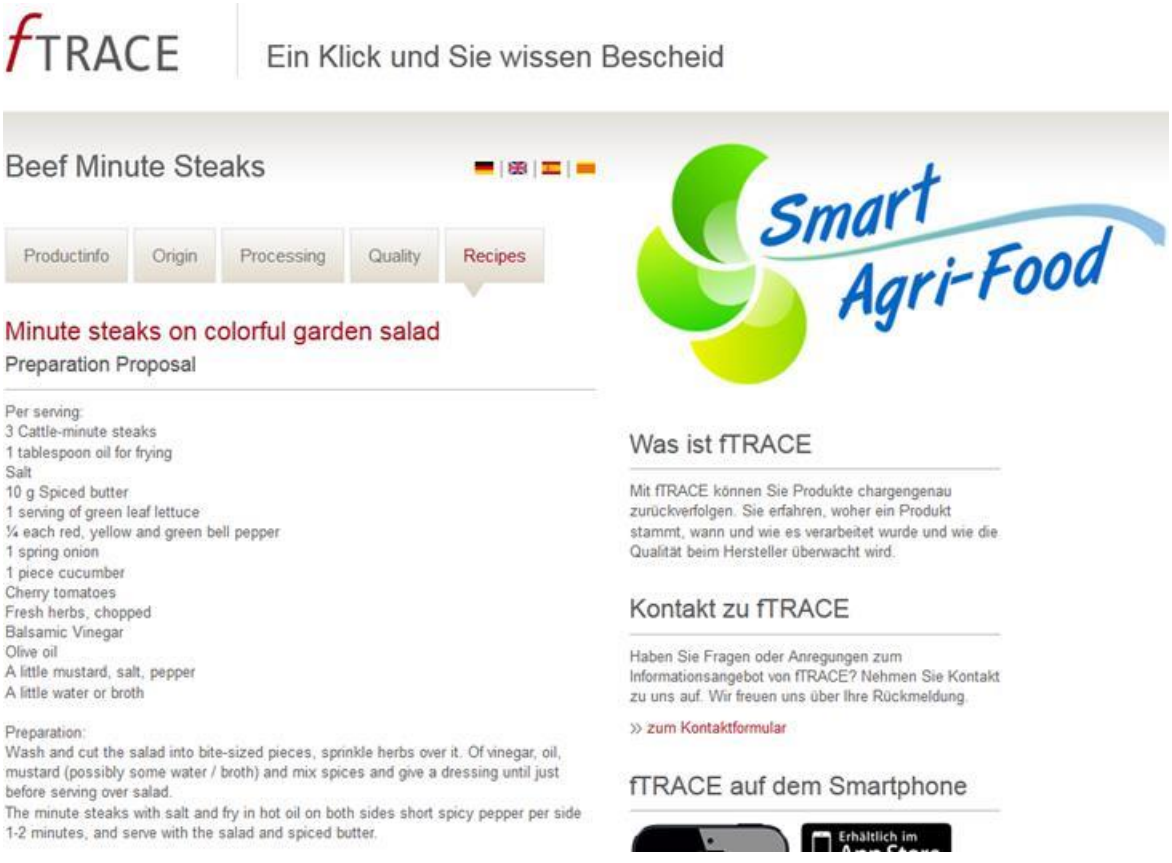


Figure 28. Static product information on recipes for minute steaks, provided by fTRACE.

4.4.3 The web service

The fTRACE web service is for the use of B2B scenarios. Companies and certification providers don't need websites. Those kinds of participants just need the raw data. Therefore data can be retrieved in XML format.

```
<?xml version="1.0" encoding="UTF-8"?>
- <foodtrace:foodTraceMessage xsi:schemaLocation="urn:gs1:ecom:foodtrace:xsd:3 FoodTrace.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:foodtrace="urn:gs1:ecom:foodtrace:xsd:3"
xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
xmlns:ecom_common="urn:gs1:ecom:ecom_common:xsd:3" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  - <sh:StandardBusinessDocumentHeader>
      <sh:HeaderVersion>1.0</sh:HeaderVersion>
    - <sh:Sender>
        <sh:Identifier Authority="GS1">4000001000005</sh:Identifier>
      </sh:Sender>
    - <sh:Receiver>
        <sh:Identifier Authority="GS1">4000001000005</sh:Identifier>
      </sh:Receiver>
    - <sh:DocumentIdentification>
        <sh:Standard>GS1</sh:Standard>
        <sh:TypeVersion>3.0</sh:TypeVersion>
        <sh:InstanceIdentifier>00000002</sh:InstanceIdentifier>
        <sh:Type>FoodTrace</sh:Type>
        <sh:CreationDateAndTime>2013-01-28T12:30:00.000</sh:CreationDateAndTime>
      </sh:DocumentIdentification>
  </sh:StandardBusinessDocumentHeader>
  + <foodTrace>
</foodtrace:foodTraceMessage>
```

Figure 29. *StandardBusinessDocumentHeader*, identifies the author and sender of this XML document.

```
- <foodTrace>
    <creationDateTime>2013-01-28T12:30:00.000</creationDateTime>
    <documentStatusCode>ORIGINAL</documentStatusCode>
    <documentActionCode>ADD</documentActionCode>
  + <foodTraceIdentification>
    <foodTraceTypeCode>MEAT</foodTraceTypeCode>
  + <productCertification>
  - <foodTraceLineItem>
      <foodProcessingActivityTypeCode>MIXED</foodProcessingActivityTypeCode>
    - <foodIdentification>
        <gtin>04000000000005</gtin>
        <foodIdentificationNumber foodIdentificationType="FTRACE">0000000298</foodIdentificationNumber>
        <additionalfoodIdentificationNumber
          foodIdentificationType="SUPPLIER_ASSIGNED">100200</additionalfoodIdentificationNumber>
        <foodTypeCode>BEEF</foodTypeCode>
      + <contentOwner>
      </foodIdentification>
    + <breedingDetails>
    + <breedingDetails>
    + <slaughteringDetails>
    + <cuttingDetails>
    + <preservationDetails>
    + <packingDetails>
    </foodTraceLineItem>
  </foodTrace>
</foodtrace:foodTraceMessage>
```

Figure 30. *foodTrace* section identifies a single batch dataset.

```
- <breedingDetails>
  - <foodProcessingParty>
      <gln>4000001000005</gln>
    - <address>
        <countryCode>ES</countryCode>
        <countyCode>Osona</countyCode>
        <state>Barcelona</state>
      </address>
    </foodProcessingParty>
  </breedingDetails>
```

Figure 31. *breedingDetails* section identifies a farmer who has produced for this batch

```
- <slaughteringDetails>
  - <foodProcessingParty>
      <gln>4000001000005</gln>
    - <address>
        <city>Manlleu</city>
        <countryCode>ES</countryCode>
        <name>Manlleu</name>
        <state>Barcelona</state>
      </address>
    </foodProcessingParty>
    <dateOfSlaughtering>2013-01-25</dateOfSlaughtering>
  </slaughteringDetails>
```

Figure 32. *slaughteringDetails* section identifies the slaughterhouse of this batch

```
- <cuttingDetails>
  - <foodProcessingParty>
      <gln>4000001000005</gln>
    - <address>
        <city>Manlleu</city>
        <countryCode>ES</countryCode>
        <name>Manlleu</name>
        <state>Barcelona</state>
      </address>
    </foodProcessingParty>
    <dateOfCutting>2013-01-26T00:00:00.000</dateOfCutting>
  </cuttingDetails>
```

Figure 33. *cuttingDetails* section identifies the deboning plant of this batch

```
- <packingDetails>
  - <foodProcessingParty>
      <gln>4000001000005</gln>
      <additionalLocationIdentification>D-EZ-4711</additionalLocationIdentification>
    - <address>
        <city>Balenyà</city>
        <countryCode>ES</countryCode>
        <name>Bon Preu-Balenyà</name>
        <state>Barcelona</state>
      </address>
    </foodProcessingParty>
    <packingDate>2013-01-28T12:12:53.000</packingDate>
  </packingDetails>
```

Figure 34. *packingDetails* section identifies the Deboning Plant of this batch

```
- <preservationDetails>
      <foodProcessingParty/>
      <bestBeforeDate>2013-02-04T00:00:00.000</bestBeforeDate>
  </preservationDetails>
```

Figure 35. *preservationDetails* section contains the best-before-date

4.4.4 fTRACE server

The fTRACE server is hosted on a virtual Server platform and separated in five “sub” servers.

1. Test server:

- Server with own database

- for further development

2. Database Server One:

- MySQL Server with test datasets

- can only be accessed by the test server

3. Staging server:

- Server with access to live data

- for pretesting any innovations with live data

4. Live server:

- Server with access to live data

- after approval in test and staging system, innovations will be staged live

5. Database Server Two:

- MySQL Server with live datasets

- can be accessed by staging and live server

In addition, to those servers, an extra service is installed on the live and staging server. The FIT Server; this server receives all queries from Smartphone’s or other devices. It recognizes the resolution, vendor and operating system of the enquiring device.

The main task of the FIT server is to translate and reformat any html, picture and video file, so that they fit to the enquiring devices.

4.4.5 Data acquisition

There are two kinds of data that are used for fTRACE.

1. Static Data or Master Data:

Those data contain the product information such as: pictures, article number, videos and additional texts.

That information will be delivered by the marketing divisions of the fTRACE system partners and entered into the fTRACE CMS by an operator.

2. Dynamic data:

Those data are transferred for each batch of a specific product. The data are saved in a GS1-XML based format and transferred via FTP and FTPs.

4.5 TTAM Pilot and Standardizations

In the TTAM pilot, and also in other related pilots, the unique identification of items is an essential requirement. The GS1 identification key is integrated in the current intermediate version of fTRACE. In the full version of fTRACE the GS1's EPCIS standard for the capture and exchange of dynamic transparency data identified with an electronic product code EPC will be implemented. As a consequence the current dynamic data repository will be replaced by the EPCIS repository.

Often EPCIS is associated with RFID. However, the use of EPCIS doesn't require the use of RFID technology. it is important to comprehend that EPCIS is data carrier agnostic. Thus, EPCIS does not necessarily require RFID technology.

In the final scenario data will be gathered by the a party transparency provider from each supply chain actor (farmer, slaughter house and meat processors). That means each time a production process is completed by an actor an event is generated containing transparency data and sent to the third part EPCIS repository. Besides data coming from the supply chain actors certification information from standardization bodies such as GlobalG.A.P. and information from authorities, such as the HIT database are implemented or are in the process of being implemented in the fTRACE system. Figure 36 depicts the final step of the implementation of the TTAM pilot.

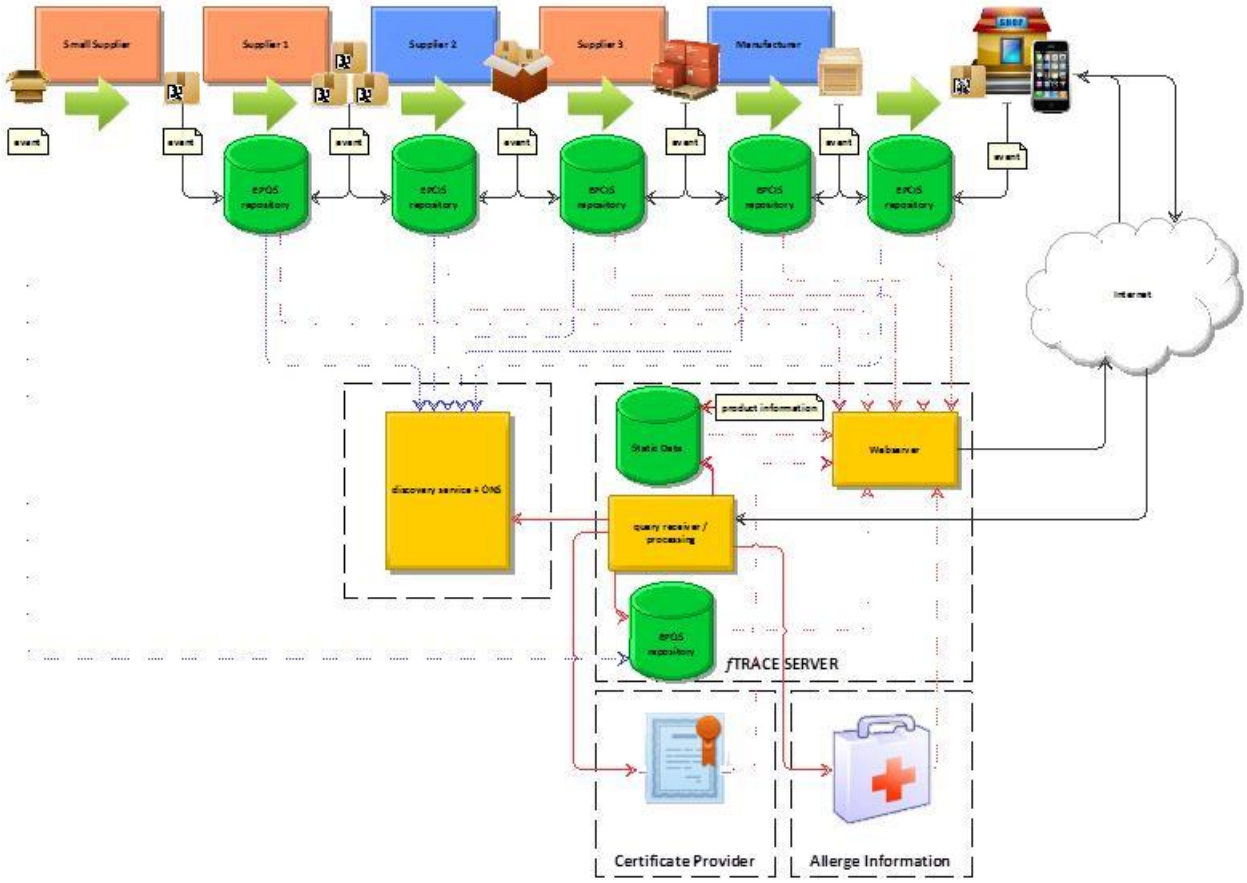


Figure 36: Final extension, in which all meat supply chain organisation may manage their own EPCIS server (local or in the cloud) or use a central server.

5 Integration of TTAM and TIC Pilot

This chapter explores the benefits and technological challenges resulting of the integration of the TTAM and TIC pilot. The first section describes a high level scenario and the architectural requirements for the integration. The following section explores our proposal for message exchange and information generation between the two pilots.

5.1 High Level Scenario

One of the main features of the TTAM and TIC pilot is that they can perform as single and independent systems; so consumers (depending on the situation) can use an fTrace-based application or a SmartWebProxy-based one in order to get two different sort of information on demand. However, there are needed common interfaces, languages and protocol for allowing a seamless information exchange between TTAM and TIC pilot. At first sight this integration can be cumbersome because the two pilots have addressed different goals concerning food awareness. However, in a high level joint scenario both pilot must offer a common information entry point which should work at least in consumers' mobile devices.

The TTAM pilot is focused on tracking, tracing and awareness of meat so D400.2 describes a basic TTAM service which allows consumers to retrieve information from the TTAM provider. Information can be fetched or pulled, so it offers the possibility of aggregating information on-the-fly, and customizing the system response. A high level integration scenario starts in this process. Since the TIC pilot defines a flexible architecture for the mobile device, it can incorporate the TTAM system as an additional information source which could be enabled, modified, or disabled on demand. Hence, the integration the tailoring information process can be enriched with information tracking, profiles, supermarket data, and so on, which will offer benefits to all the participants of the information chain.

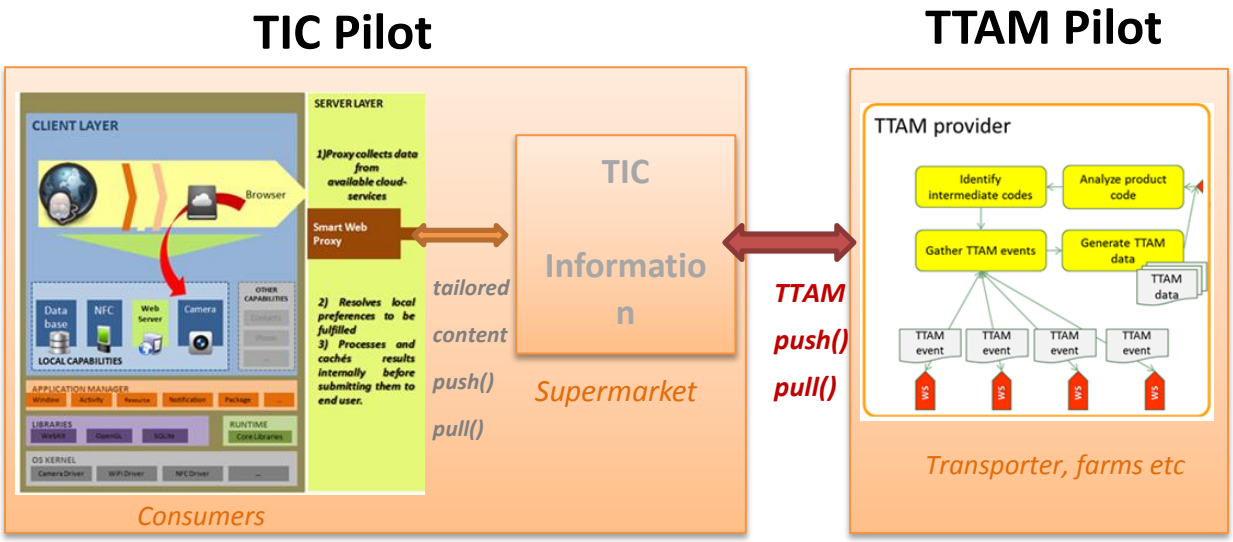


Figure 37. TIC Pilot and TTAM Pilot high level architectures

5.1.1 Architectural requirements for integration

By the end of this deliverable a fully integration of the TTAM and TIC pilots has not been planned; however, both pilots have made use of web technologies which allows a more straightforward integration. As it has been previously described, both architectures and their corresponding pilots make use of asynchronous events (pushing) and on demand (pulling) for information delivery, which makes ensures some sort of basic compatibility; however other facts should be addressed.

1. Data exchange model:
2. Communication protocol
3. Information unit
4. Security model and trust
5. Workflow coordination

These compatibility "requirements" are not trivial in any integration; however the own web-based implementation of the TTAM and TIC architectures makes more suitable to fulfil requirements 1, 2 and 3 in this phase of the project. Thus, in next sections describe an abstracted but compatible message exchange scenarios.

5.2 Message Exchange

Following the pulling and pushing communication approach, there are three available situations which can involve consumers, the TIC and the TTAM systems.

In the first case consumers request information on demand (pull), so they automatically trigger an information request to the supermarket (that could also redirect to a GE or process itself the information), and later the supermarket requests information to the TTAM system. In the *content adaptation process* the TIC application servers should make use of the existing profile information of the user for customizing the content, even if the request has just being issued.

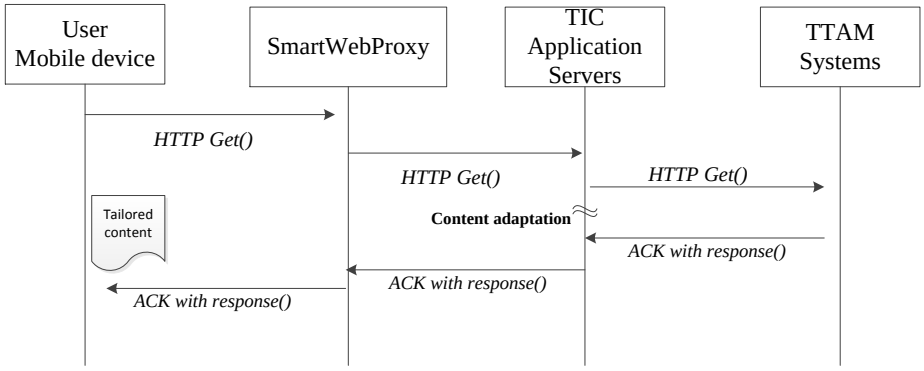


Figure 38. On demand exchange

In the second case the tailored information is pushed to the mobile device of the consumer; so here TIC systems should accepts event subscriptions from the clients and manage if this subscriptions involves asynchronous events from the TTAM systems. Hence, the TIC system should not only customizing the information but also *correlate the correct events* arriving from the TTAM systems.

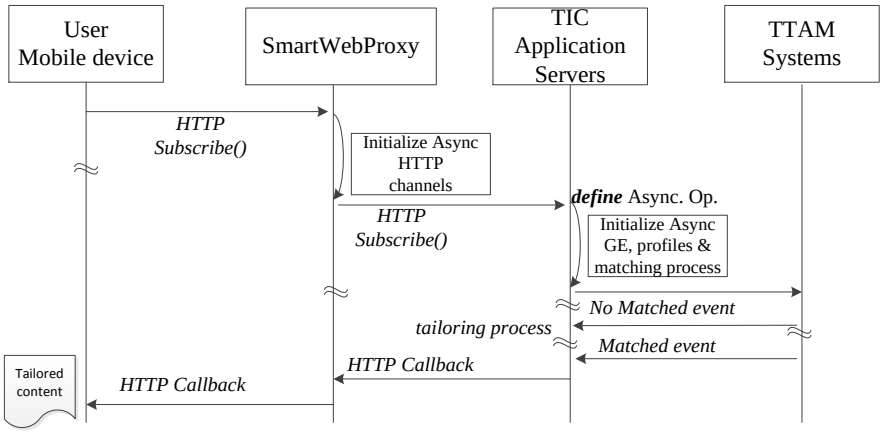


Figure 39. Push exchange

The third case is a mixed case where on demand and asynchronous events are delivered from/to consumers' mobile devices, TIC systems and TTAM systems. This case allows consumers to take advantage of TTAM proactive process that offer information (e.g. products' timestamp).

Concerning the data format, the TIC Pilot employs a document for all the exchanging process of products info and consumer profiles. In the TTAM pilot the EPICS standard identifies four types information for such an event: WHAT (identifying what product is), WHERE (the location where the event is generated), WHEN (the time the event took place), and WHY (the reason for the event, e.g. receiving shipping etc.). As it is shown in the next figure, the information combining attributes fetched from both pilots can be included in a single XML file (Annex C).

In the current stage the TIC and TTAM pilots are integrated in the mobile application, as it is shown in the following figure. This integration will be enhanced in future version and the goal is to allow users to get information using the same steps. Hence, this requires that TIC and TTAM should be coordinated in the way, the mobile device recognize a logo, and the resulting request is redirected to the right servers.

Figure 40. First version of the integration TIC and TTAM pilots

5.3 Management of Alerts

A fundamental part in the post-shopping activities, along with the management of consumer feedback about supermarket services or products is also the alert notification mechanism.

A key issue is to develop a sub-system that allows collecting and managing dynamic information and data generated at different points of the supply chain to have full transparency data available to the consumer.

In a food alert scenario, a health government organization could detect a problem in a particular delivery of food and trace the supermarkets that sell these products. The supermarket needs to locate the customer which purchased the problematic product and send him/her an alert. As the customer contact information is confidential and restricted to users with Fidelity card we need to change the customer registration business process to ask the customer if they want to be notified when a food alert occurs.

The alert delivery to customers is handled by different media, email, push notifications and also phone calls. In the case of push notifications the specification of the alert delivery is described in the workflow of the following figure.

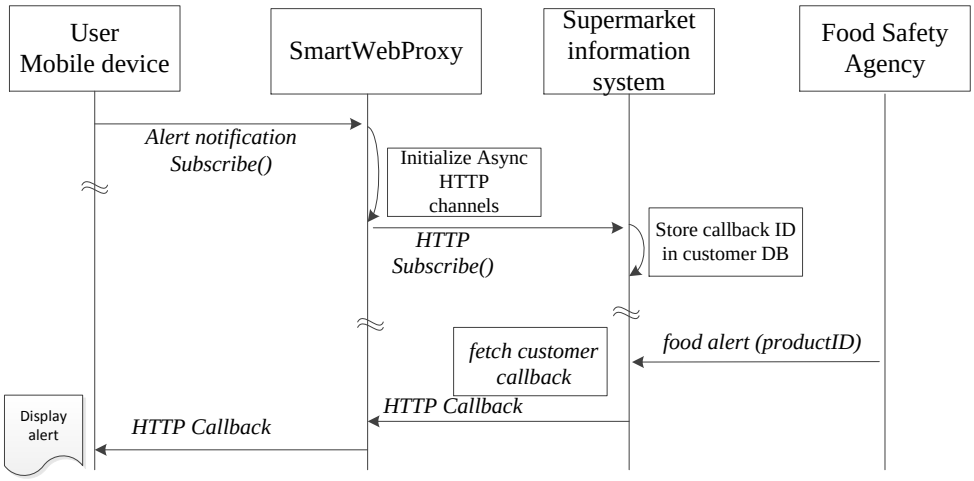


Figure 41. Alert notification by *push* approach

As can be seen, the customer's mobile device subscribes to the supermarket information system through the Web proxy and this system stores the *callback* information of the customer in a private database. When the food safety agency notifies the supermarket a food alert, the customer callbacks are fetched and the notification is delivered to their mobile phone.

6 Conclusions

The main achievements of this document are as follows:

- The technical description of the TIC and TTAM pilot has been provided. This document also details the different implementation steps which have been applied for a successful validation. It also illustrates the implementation efforts, inside WP400, for integrating some Generic Enablers provided by FI-WARE and the existing constrains and benefits for their adoption in the pilots.
- Some subsystems mentioned in D400.2 have been revised as well as a technological forecast of how food awareness could be improved from the consumers' perspective is given. In other words, how consumers can really adopt and employ this new approach using tools they are already familiar with.
- This deliverable reviews the different procedures, tasks and data models, inside the pilots, that can be candidates for standardization efforts, and finally, the further implications in the design, implementation and deployment phases.
- The Architectural integration of the TIC and TTAM pilot. As both pilots target different scenarios, this document describes paths for integrating different information sources, and how these new data can reach users depending on alerts situations.

The quantitative and qualitative evaluation results of both pilots will be covered in the D400.4 "Smart Food Awareness: Final assessment Report".

7 References

1. FI-WARE High-level Description 2011.
2. FI-WARE project. [online]. Retrieved from http://www.fi-ware.eu/
3. project, SmartAgriFood D400.1 Smart Food Awareness Specification for Experimentation
4. project, SmartAgriFood D400.2 Smart Food Awareness Generic Enablers and Architectural Requirements.
5. HTML5 Reference page [online]: Retrieved from: http://www.w3.org/html/wg/drafts/html/master/single-page.html
6. SmartShopping cart. http://www.smartplanet.com/blog/thinking-tech/shopping-cart-helps-consumers-make-smarter-food-choices/11403
7. Tomás Robles, Sergio Gonzalez-Miranda, Ramón Alcarria, Augusto Morales Dominguez: Web Browser HTML5 Enabled for FI Services. UCAmI 2012. Springer Lecture Notes in Computer Science, pp 181-184.
8. A. Kuranov, R. Lienhart and V. Pisarevsky, “Empirical Analysis of Detection Cascades of Boosted,” Microprocessor Research Lab, Intel Labs, 2002.
9. Ramon Alcarria, Tomás Robles,Augusto Morales, Sergio González-Miranda. New Service Development Method for Prosumer Environments, The Sixth International Conference on Digital Society, ICDS'12
10. HTT Apache Server Project [online]. Retrieved from http://httpd.apache.org/
11. Mule Software. [online]. Retrive from http://www.mulesoft.org

8 Annex A.

This annex shows some portion of code related for each key process of the mobile application. All the source code will be available upon request.

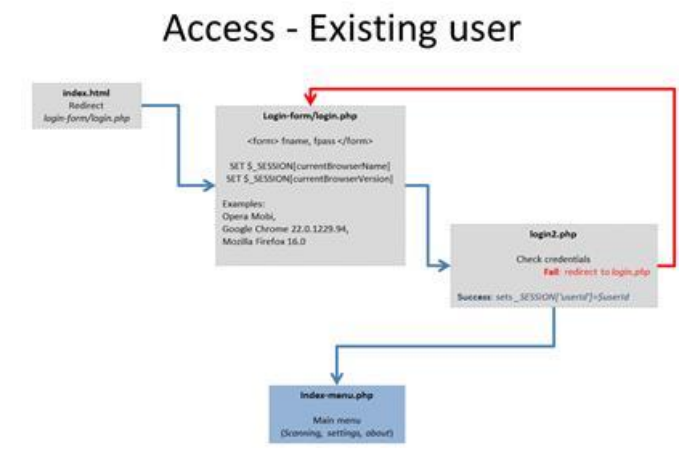
Login Process

PHP support in the login process is displayed below.

```php
<?php

include '../config/cfg.php';

$userId = $_POST['fname'];
$pass = $_POST['fpass'];

if (isset($userId) && isset($pass)) {
    if (!empty($userId) && !empty($pass)) {
        //echo 'OK';
    } else {
        echo 'Nothing entered.';
    }
}


include '../proxy/proxy.php';
if ((file_get_contents($url)) == 'true') {
    //echo "ok";
    $_SESSION['userId']=$userId;
```

Figure 42. Login Process

Javascript also has been used in order to redirect to specific resources through access to DOM elements as shown below:




```php
include '../proxy/proxy.php';
if ((file_get_contents($url)) == 'true') {
    //echo "ok";
    $_SESSION['userId']=$userId;

    echo "<script type='text/javascript'>window.location = '../index-menu.php'</script>";
    }
    else {
        //echo "error";
        //header("Location: login.php");
        echo "<script type='text/javascript'>window.location = 'login.php'</script>";
    }

?>
```

Figure 43. User Access

Process of Setting preferences

Figure 44. Setting Process

In this task, CURL command line tool for getting or sending files using URL syntax was employed as it supports a range of common Internet protocols, currently including HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, LDAP, LDAPS, DICT, TELNET, FILE, IMAP, POP3, SMTP and RTSP.

```
$url = 'http://'.$ip_server.':'.$port;


$ch = curl_init($url.'/services/preferences');

curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string))
);

$result = curl_exec($ch);
curl_close($ch);
echo $result;
```

Figure 45. Setting Process 2

Camera access process

Among the distinct another important use of Jquery is regarding camera access which is achieved by the invocation of native tags.

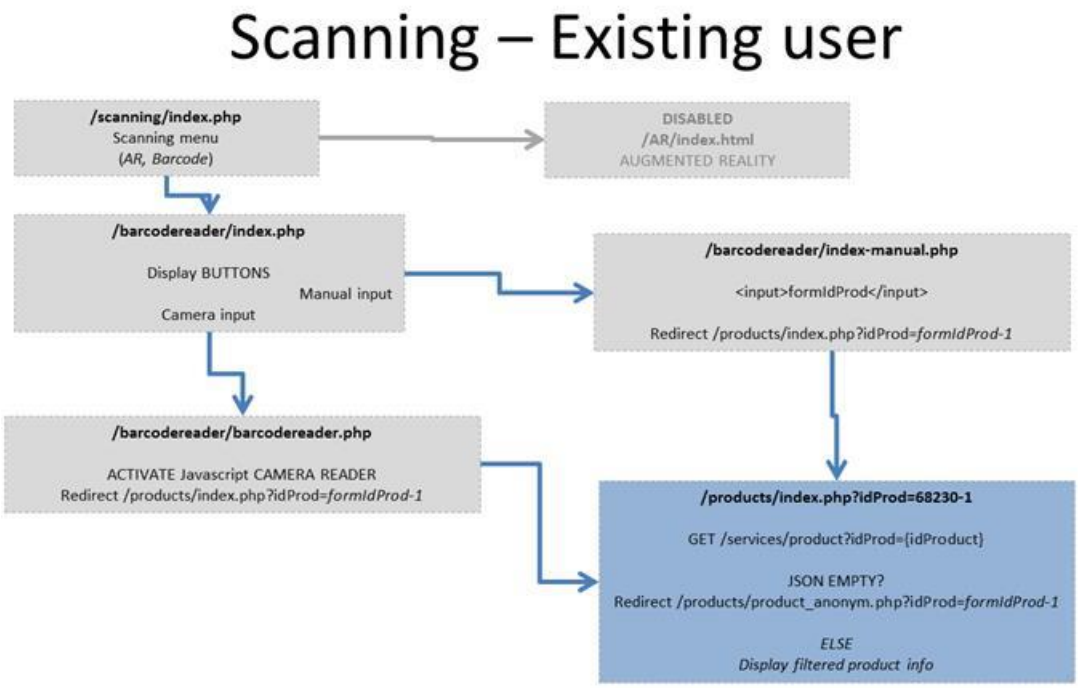


Figure 46. Camera Access Model

```
  var mybase = "../products/product_anonym.php?idProd=";
  curlink.href = (mybase + newlink.innerHTML);
  curlink.innerHTML = mybase + newlink.innerHTML;
  window.location = curlink.innerHTML;
  //alert(newlink.innerHTML);
  //$("#botonir1").show("slow");
}


$(document).ready(function(){
 $("#selector_manual").click(function(){
  $("#input_manual").show('slow');
  $("#input_camera").hide('slow');
 });


 $("#selector_camera").click(function(){
  $("#input_camera").show('slow');
  $("#input_manual").hide('slow');
 });
```

Figure 47. Camera Access portion of code

QR code scanning process

An opensource javascript library named WebQR has been employed to read such codes.

```
function passLine(stringPixels) {

    var coll = stringPixels.split("-");

    for(var i=0;i<320;i++) {
        var intVal = parseInt(coll[i]);
        r = (intVal >> 16) & 0xff;
        g = (intVal >> 8) & 0xff;
        b = (intVal ) & 0xff;
        imageData.data[c+0]=r;
        imageData.data[c+1]=g;
        imageData.data[c+2]=b;
        imageData.data[c+3]=255;
        c+=4;
    }

    if(c>=320*240*4) {
        c=0;
        gCtx.putImageData(imageData, 0,0);
        try{
            qrcode.decode();
        }
        catch(e){
            console.log(e);
            setTimeout(captureToCanvas, 500);
        };
    }
}

function captureToCanvas() {
    if(stype!=1)
        return;
    if(gUM)
    {
        gCtx.drawImage(v,0,0);
        try{
            qrcode.decode();
        }
        catch(e){
            console.log(e);
            setTimeout(captureToCanvas, 500);
        };
    }
```

Figure 48. QR Scanning Process

Multilanguage Support

Separate language files have been used to provide a multilingual interface.

```php
<?php
/*
-----------------
Language: English
-----------------
*/

$lang = array();

/*setting language*/
$lang['LANGUAGE_TITLE'] = 'Language';
$lang['LANGUAGE_INSTRUCTIONS'] = 'Please select your language';
$lang['LANGUAGE_CURRENT'] = 'Currently selected';

/*login.php*/
$lang['LOGIN_INSTRUCTIONS'] = 'Enter login details';
$lang['LOGIN_USERID'] = 'User Id';
$lang['LOGIN_PASSWORD'] = 'Password';
$lang['LOGIN_SIGNIN'] = 'Sign in';
$lang['LOGIN_CREATENEW'] = 'Create new account';
$lang['LOGIN_ANONYMOUS'] = 'Anonymous sign in';
$lang['LOGIN_SHOWPASSWORD'] = 'Show';
$lang['LOGIN_HIDEPASSWORD'] = 'Hide';

/*newUser.php*/
$lang['NEWUSER_TITLE'] = 'New account';
$lang['NEWUSER_TXT1'] = 'Re-type';
$lang['NEWUSER_TXT2'] = 'Step 1 of 2';
$lang['NEWUSER_TXT3'] = 'Go to step 2';
$lang['NEWUSER_TXT4'] = 'Step 2 of 2';
$lang['NEWUSER_TXT5'] = 'The selected entry already exists';
$lang['NEWUSER_TXT6'] = 'Sorry. The selected User Id is already taken';
$lang['NEWUSER_TXT7'] = 'We are now transferring you to start over';

/*newUser2.php*/
$lang['NEWUSER2_TXT1'] = 'Set your ';
$lang['NEWUSER2_SAVE'] = 'Save';

/*index-menu.php*/
$lang['MENU_WELCOME'] = 'Welcome';
$lang['MENU_LOGOUT'] = 'Log out';
$lang['MENU_TITLE'] = 'Main menu';
$lang['MENU_SCANNING'] = 'SCANNING';
$lang['MENU_SETTINGS'] = 'SETTINGS';
```

Figure 49. Multilanguage Support

Logistic information process

Maps have been represented through Google Maps API.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title>saf</title>
    <link href="default.css" rel="stylesheet">
    <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>
    <script>
      var directionDisplay;
      var directionsService = new google.maps.DirectionsService();
      var map;

      function initialize() {
        directionsDisplay = new google.maps.DirectionsRenderer();

        var barcelona = new google.maps.LatLng(41.383333, 2.183333);
        var mapOptions = {
          zoom: 6,
          mapTypeId: google.maps.MapTypeId.ROADMAP,
          center: barcelona,
          disableDefaultUI: true,
        }
        map = new google.maps.Map(document.getElementById('map_canvas'), mapOptions);
        directionsDisplay.setMap(map);
      }

      function calcRoute() {

        var start = "<?php echo $_GET['p0lat']."", ".$_GET['p0long']; ?>";

        var end = "<?php echo $_GET['p2lat']."", ".$_GET['p2long']; ?>";
        var waypts = [];


            waypts.push({
                location:"<?php echo $_GET['p1lat']."", ".$_GET['p1long']; ?>",
                stopover:true});

        var request = {
            origin: start,
            destination: end,
            waypoints: waypts,
            optimizeWaypoints: true,
```

Figure 50. Logistic Information process

9 Annex B.

This annex presents The API to Access the Tailored Information System (TIS). Below there is a description of each method accessible to obtain the required information.

Each service will access to a database placed in the cloud. The API is available for testing using the information below.

{ip_server}: 212.170.156.54

{port}: 8080

UserSystemData: Method related user system information. Includes login and registry methods.

GET (login): Returns true if the user exists in the system, otherwise returns false.
POST (registry): Registers a new user in the system.

```
{
  "id": "4",
  "pass": "admin",
}
```

UserPreferences(ATOS):

GET: Returns the user preferences.

```
{
  "id": "7",
  "origin": true,
  "chemical": false,
  "quality": false,
  "caducity": false,
  "allergens": false,
  "price": true,
  "sustainability": false,
  "productionDate": false,
  "nutritionalInfo": false,
  "dop": false,
  "kosher": false,
  "halal": false,
  "meatPrefs": {"welfareCert": true},
  "eggPrefs": {"raising": false},
  "winePrefs":   {
    "marriage": false,
    "variety": false,
    "tastingNote": true
  },
  "transgenics": false,
  "fruitPrefs": {"season": false}
```

}

POST: Submit the user preferences. There are three types of preferences: for eggs, meat and fresh fruits & vegetables. The input must be what follows:


```
{
  "id": "7",
  "origin": true,
  "chemical": false,
  "quality": false,
  "caducity": false,
  "allergens": false,
  "price": true,
  "dop": false,
  "kosher": false,
  "halal": false,
  "eggPrefs": {"raising": false},
  "meatPrefs": {"welfareCert": true},
  "fruitPrefs": {"season": false},
  "winePrefs":   {
    "variety": false,
    "marriage": false,
    "tastingNote": true
  },
  "transgenics": false,
  "sustainability": false,
  "productionDate": false,
  "nutritionalInfo": false
}
```


PUT : Updates the user preferences. The input is what follows:


```
{
  "id": "1",
  "origin": true,
  "chemical": false,
  "quality": false,
  "caducity": false,
  "allergens": false,
  "price": true,
  "dop": false,
  "kosher": false,
  "halal": false,
  "eggPrefs": {"raising": false},
  "meatPrefs": {"welfareCert": true},
  "fruitPrefs": {"season": false},
  "winePrefs":   {
    "variety": false,
    "marriage": false,
```

```
    "tastingNote": true
  },
  "transgenics": false,
  "sustainability": false,
  "productionDate": false,
  "nutritionalInfo": false
}
```


Products: Methods related to the product information.

GET: Returns the information related to the three different kinds of products.

- Information for Eggs:


```
{
  "raising": 1,
  "variety": "Ecological",
  "supplier": "Bonpreu",
  "origin": "BCN",
  "quality": "Selección BP",
  "kosher": false,
  "halal": true,
  "ecological": true,
  "dop": true,
  "transgenics": true,
  "ecoPacking": false,
  "batch": "1021",
  "footprint": "0.24",
  "igp": false,
  "pesticides": true,
  "additives": true,
  "preservatives": false,
  "caducity": "01-10-2012 04:28:26",
  "allergens": {
    "eggs": true,
    "gluten": false,
    "lactose": false,
    "nuts": false
  },
  "productionDate": "01-10-2012 12:00:00",
  "price": "1,34",
  "nutritionalInfo": {
    "kcal": "234",
    "greasy": "0,34",
```

```
      "carbohydrates": "63",
      "sugar": "28",
      "proteins": "89"
    },
   "logisticsInfo": [
      {
        "latitude": 41.9999,
        "longitude": 2.28416
      },
      {
        "latitude": 41.405,
        "longitude": 2.16911
      },
      {
        "latitude": 41.9303,
        "longitude": 2.25435
      }
    ]
}
```


- Information for meat:


```
{
  "welfareCert": true,
  "variety": "Ternera",
  "supplier": "Bonpreu",
  "origin": "Asturias",
  "quality": "Selección Q",
  "kosher": false,
  "halal": true,
  "ecological": false,
  "dop": false,
  "transgenics": true,
  "ecoPacking": true,
  "batch": "1021",
  "footprint": "0.35",
  "igp": true,
  "pesticides": false,
  "additives": true,
  "preservatives": true,
```

```
"caducity": "01-10-2012 04:28:26",
"allergens": {
  "eggs": true,
  "gluten": false,
  "lactose": false,
  "nuts": true
},
"productionDate": "01-10-2012 12:00:00",
"price": "4,65",
"nutritionalInfo": {
  "kcal": "432",
  "greasy": "12",
  "carbohydrates": "67",
  "sugar": "17",
  "proteins": "92"
},
"logisticsInfo": [
  {
    "latitude": 41.9999,
    "longitude": 2.28416
  },
  {
    "latitude": 41.405,
    "longitude": 2.16911
  },
  {
    "latitude": 41.9303,
    "longitude": 2.25435
  }
],
}
```


- Information for Fresh fruits & vegetables:


```
{
  "season": false,
  "variety": "Platano",
  "supplier": "Bonpreu",
  "origin": "Canarias",
  "quality": "Sin Selección",
```

```
"kosher": false,
"halal": true,
"ecological": true,
"dop": true,
"transgenics": false,
"ecoPacking": false,
"batch": "1021",
"footprint": "0.14",
"igp": true,
"pesticides": true,
"additives": false,
"preservatives": true,
"caducity": "01-10-2012 04:28:26",
"allergens": {
   "eggs": true,
   "gluten": false,
   "lactose": true,
   "nuts": true
},
"productionDate": "01-10-2012 12:00:00",
"price": "0,85",
"nutritionalInfo": {
   "kcal": "45",
   "greasy": "7",
   "carbohydrates": "15",
   "sugar": "86",
   "proteins": "24"
},
"logisticsInfo": [
   {
      "latitude": 41.9999,
      "longitude": 2.28416
   },
   {
      "latitude": 41.405,
      "longitude": 2.16911
   },
   {
      "latitude": 41.9303,
```

```
            "longitude": 2.25435
        }
    ]
}
```


- Information for wine:


```
{
        "variety": "Rioja",
        "tastingNote": "7",
        "marriage": "Carnes rojas",
        "supplier": "Don Simon",
        "origin": "La rioja",
        "quality": "Selección Q",
        "kosher": true,
        "halal": false,
        "ecological": false,
        "dop": true,
        "transgenics": false,
        "ecoPacking": true,
        "batch": "1021",
        "footprint": "0.39",
        "igp": true,
        "pesticides": false,
        "additives": false,
        "preservatives": true,
        "caducity": "01-10-2012 04:28:26",
        "allergens": {
            "eggs": false,
            "gluten": false,
            "lactose": true,
            "nuts": false
        },
        "productionDate": "01-10-2012 12:00:00",
        "price": "4,67",
        "nutritionalInfo": {
            "kcal": "98",
            "greasy": "62.2",
            "carbohydrates": "41.3",
            "sugar": "56",
```

```
            "proteins": "32"
        },
        "logisticsInfo": [
            {
                "latitude": 41.9999,
                "longitude": 2.28416
            },
            {
                "latitude": 41.405,
                "longitude": 2.16911
            },
            {
                "latitude": 41.9303,
                "longitude": 2.25435
            }
        ]
}
```


- Information for dry products:

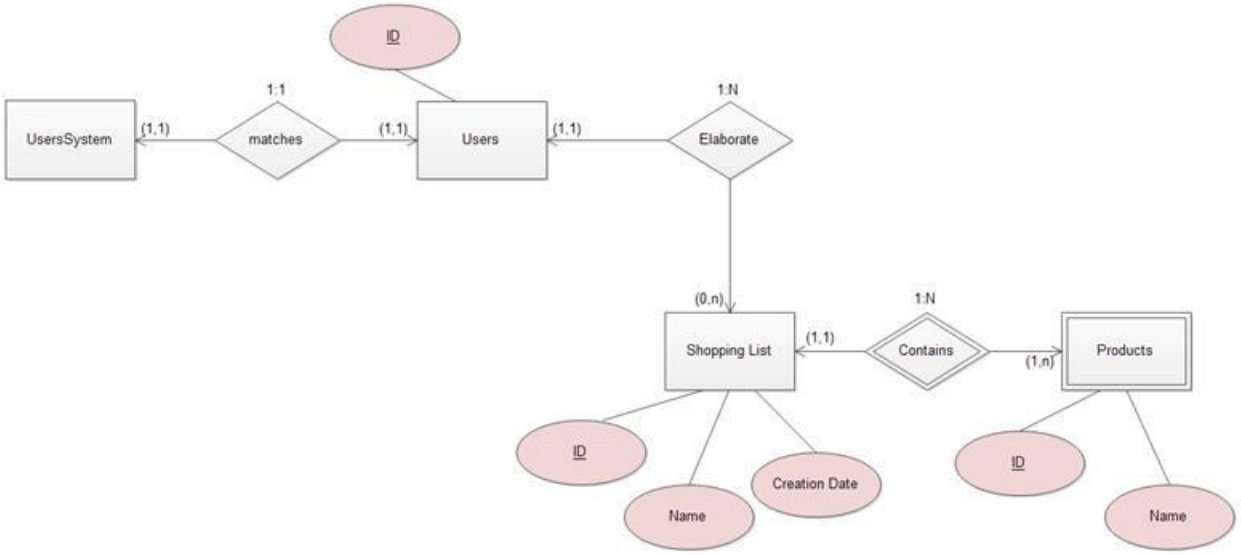

```
{
    "variety": "Cafe",
    "supplier": "Bonpreu",
    "origin": "Extremadura",
    "quality": "Sin Selección",
    "kosher": true,
    "halal": false,
    "ecological": false,
    "dop": true,
    "transgenics": true,
    "ecoPacking": true,
    "batch": "1021",
    "footprint": "0.22",
    "igp": false,
    "pesticides": true,
    "additives": false,
    "preservatives": true,
    "caducity": "01-10-2012 04:28:26",
    "allergens": {
        "eggs": true,
```

```
      "gluten": true,
      "lactose": true,
      "nuts": true
    },
    "productionDate": "01-10-2012 12:00:00",
    "price": "2,36",
    "nutritionalInfo": {
      "kcal": "45",
      "greasy": "0,34",
      "carbohydrates": "63",
      "sugar": "28",
      "proteins": "89"
    },
    "logisticsInfo": [
      {
        "latitude": 41.9999,
        "longitude": 2.28416
      },
      {
        "latitude": 41.405,
        "longitude": 2.16911
      },
      {
        "latitude": 41.9303,
        "longitude": 2.25435
      }
    ]
}
```


Databases

For the pilot, there will be three databases: one for the users, one for the products and another one for the logos. This document describes the information contained in the users' and products' databases.

User Database



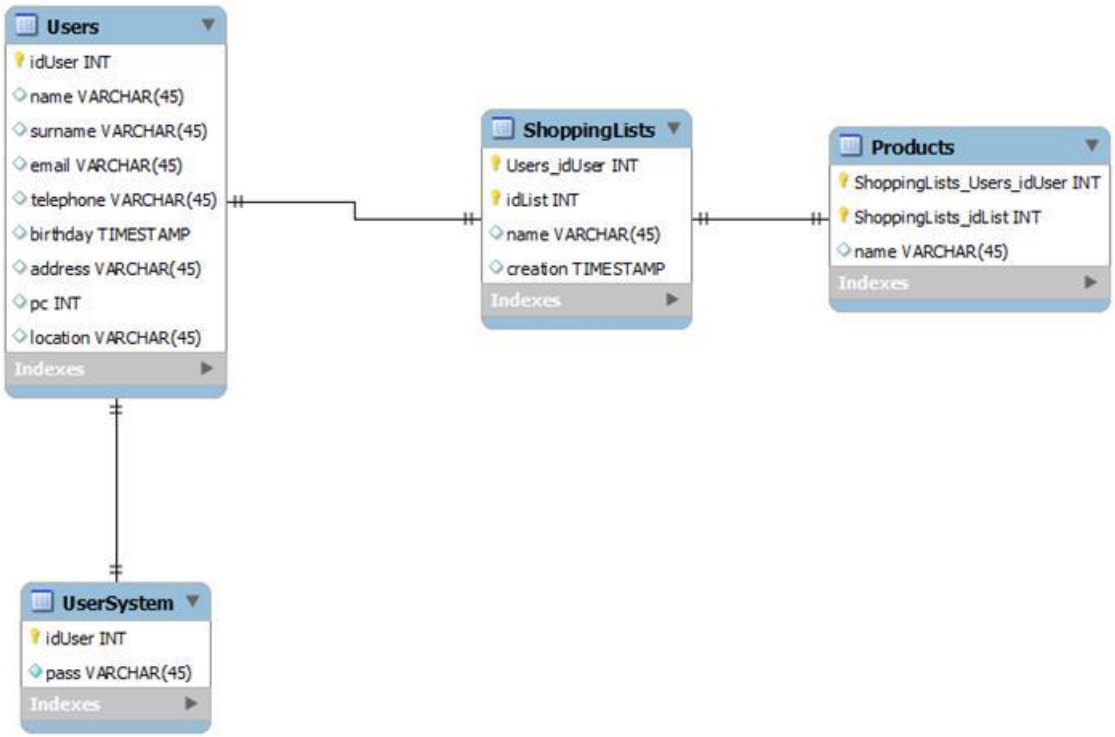


Figure 51. User Database

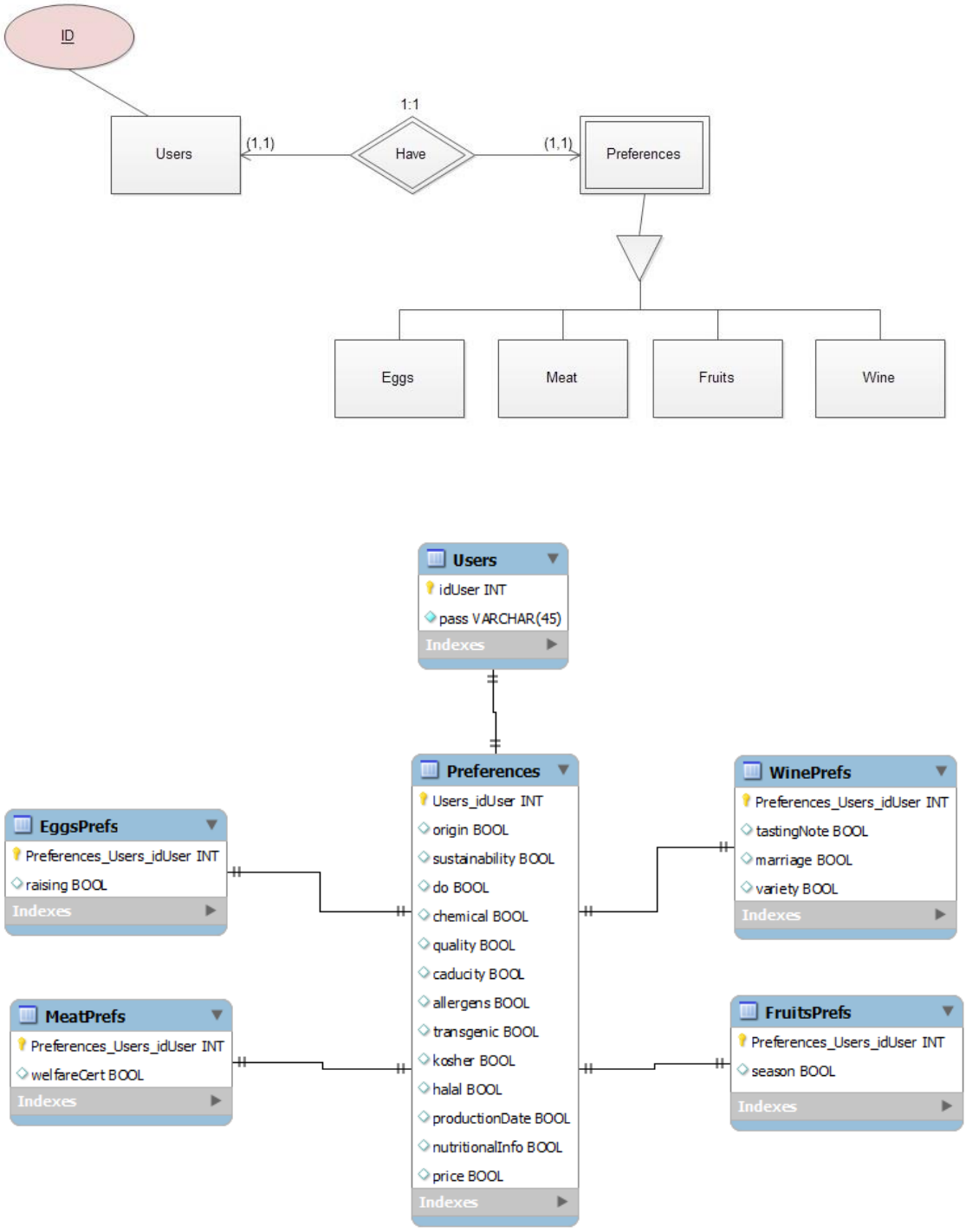


Figure 52. Preferences Database

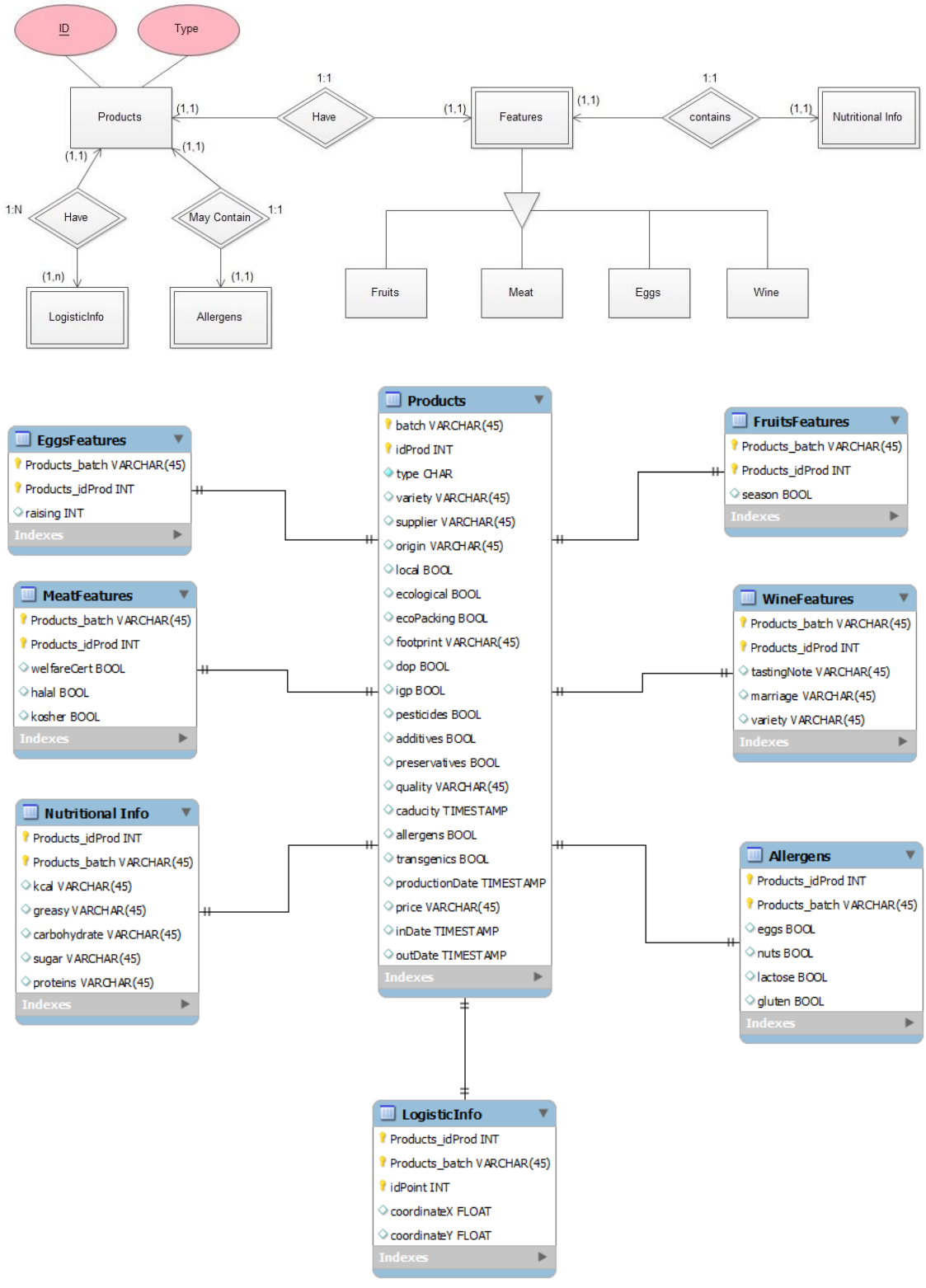


Figure 53. Product Database

10 Annex C.

The following XML File is the result of the integration of both pilots' data models.

```
<?xml version="1.0" encoding="UTF-8"?>
<foodtrace:foodTraceMessage xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ecom_common="urn:gs1:ecom:ecom_common:xsd:3"
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader" xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
xmlns:foodtrace="urn:gs1:ecom:foodtrace:xsd:3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:gs1:ecom:foodtrace:xsd:3
FoodTrace.xsd">
	<sh:StandardBusinessDocumentHeader>
		<sh:HeaderVersion>1.0</sh:HeaderVersion>
		<sh:Sender>
			<sh:Identifier Authority="GS1">4000001000005</sh:Identifier>
		</sh:Sender>
		<sh:Receiver>
			<sh:Identifier Authority="GS1">4000001000005</sh:Identifier>
		</sh:Receiver>
		<sh:DocumentIdentification>
			<sh:Standard>GS1</sh:Standard>
			<sh:TypeVersion>3.0</sh:TypeVersion>
			<sh:InstanceIdentifier>00000002</sh:InstanceIdentifier>
			<sh:Type>FoodTrace</sh:Type>
			<sh:CreationDateAndTime>2013-01-28T12:30:00.000</sh:CreationDateAndTime>
		</sh:DocumentIdentification>
	</sh:StandardBusinessDocumentHeader>
	<foodTrace>
		<creationDateTime>2013-01-28T12:30:00.000</creationDateTime>
		<documentStatusCode>ORIGINAL</documentStatusCode>
		<documentActionCode>ADD</documentActionCode>
		<foodTraceIdentification>
			<entityIdentification>00000001</entityIdentification>
		</foodTraceIdentification>
		<foodTraceTypeCode>MEAT</foodTraceTypeCode>
		<productCertification>
			<entityIdentification>00000001001</entityIdentification>
			<contentOwner>
				<gln>4000001000005</gln>
			</contentOwner>
			<lineItemNumber>1</lineItemNumber>
		</productCertification>
		<foodTraceLineItem>
			<foodProcessingActivityTypeCode>MIXED</foodProcessingActivityTypeCode>
			<foodIdentification>
				<gtin>04000000000005</gtin>
				<foodIdentificationNumber foodIdentificationType="FTRACE">0000000298</foodIdentificationNumber>
				<additionalfoodIdentificationNumber foodIdentification-
Type="SUPPLIER_ASSIGNED">100200</additionalfoodIdentificationNumber>
				<foodTypeCode>BEEF</foodTypeCode>
				<contentOwner>
					<gln>4000001000005</gln>
					<additionalPartyIdentification additionalPartyIdentificationType-
Code="BUYER_ASSIGNED_IDENTIFIER_FOR_A_PARTY">GS</additionalPartyIdentification>
				</contentOwner>
			</foodIdentification>
			<breedingDetails>
				<foodProcessingParty>
					<gln>4000001000005</gln>
					<address>
						<countryCode>ES</countryCode>
						<countyCode>Osona</countyCode>
						<state>Barcelona</state>
					</address>
				</foodProcessingParty>
			</breedingDetails>
			<breedingDetails>
				<foodProcessingParty>
					<gln>4000001000005</gln>
					<address>
						<countryCode>ES</countryCode>
						<countyCode>Selva</countyCode>
						<state>Girona</state>
					</address>
				</foodProcessingParty>
				<animalIdentification>
					<animalIdentificationNumber>DE094400000</animalIdentificationNumber>
				</animalIdentification>
			</breedingDetails>
			<slaughteringDetails>
				<foodProcessingParty>
					<gln>4000001000005</gln>
					<address>
```

```
                                        <city>Manlleu</city>
                                        <countryCode>ES</countryCode>
                                        <name>Manlleu</name>
                                        <state>Barcelona</state>
                                </address>
                        </foodProcessingParty>
                        <dateOfSlaughtering>2013-01-25</dateOfSlaughtering>
                </slaughteringDetails>
                <cuttingDetails>
                        <foodProcessingParty>
                                <gln>4000001000005</gln>
                                <address>
                                        <city>Manlleu</city>
                                        <countryCode>ES</countryCode>
                                        <name>Manlleu</name>
                                        <state>Barcelona</state>
                                </address>
                        </foodProcessingParty>
                        <dateOfCutting>2013-01-26T00:00:00.000</dateOfCutting>
                </cuttingDetails>
                <preservationDetails>
                        <foodProcessingParty/>
                        <bestBeforeDate>2013-02-04T00:00:00.000</bestBeforeDate>
                </preservationDetails>
                <packingDetails>
                        <foodProcessingParty>
                                <gln>4000001000005</gln>
                                <additionalLocationIdentification>D-EZ-4711</additionalLocationIdentification>
                                <address>
                                        <city>Baleny&amp;#224;</city>
                                        <countryCode>ES</countryCode>
                                        <name>Bon Preu-Baleny&amp;#224;</name>
                                        <state>Barcelona</state>
                                </address>
                        </foodProcessingParty>
                        <packingDate>2013-01-28T12:12:53.000</packingDate>
                </packingDetails>
        </foodTraceLineItem>
    </foodTrace>
</foodtrace:foodTraceMessage>
```

Figure 54. XML file for TIC and TTAM pilots