Systematic Heterogeneity: How to combine Smartphone related Apps with FIspace

Harald Sundmaeker, Peter Einramhof

ATB Institut für angewandte Systemtechnik Bremen GmbH, Germany
(ATB Institute for Applied Systems Technology Bremen GmbH)
Wiener Str. 1; D-28359 Bremen; Germany
Sundmaeker@atb-bremen.de, Einramhof@atb-bremen.de

Abstract

FIspace represents an Internet based B2B collaboration platform that can be used by actors along the supply chain, and specifically the agri-food chain, facilitating the design and usage of inter-organisational workflows. The FIspace platform was developed by using FIWARE, a European initiative to develop technologies for a Future Internet. At the same time, the requirements of actors from an agri-food related B2B environment were analysed. One of the key requirements was the demand for smartphone-based apps that can be easily used by any actor at any time in the scope of a B2B relationship of different organisations. However, realising inter-organisational workflows with a combination of smartphones and a platform for B2B collaboration raises business related requirements that are specifically in relation to interoperability and security. Therefore, the architectural principles of FIspace supported ecosystems are presented and the concept of developing FIspace software applications is detailed. The latter provides domain specific features and enables a systematic usage of heterogeneous components in business related ecosystems. Furthermore, different development strategies for smartphone apps are analysed to discuss the related implications with respect to effort, costs and interoperability when aiming at a combination with a FIspace supported ecosystem. This is compared with the general principles to develop features within a FIspace ecosystem to systematically identify the implications when integrating heterogeneous software and hardware solutions. The purpose of this paper is to present those design principles that shall help to systematically analyse end-user requirements, which need to be taken into account when developing or designing changes and extensions of a FIspace supported business ecosystem with the help of smartphone apps. Therefore, the paper addresses specifically software developers intending to use FIspace as well as business architects intending to change or design an ICT supported collaborative ecosystem.

Keywords: Business Collaboration; Heterogeneity; Food Chain; App Development; FIWARE; Future Internet; FIspace; Fruits & Vegetables

1 Introduction

Like other supply-demand chains, the agri-food sector is driven by the demand of “final customers”, the consumers. However, a key difference of the agri-food sector is the high frequency of buying decisions, especially when considering the segment of fresh produce or perishables with a fairly limited shelf-life. This represents a “weekly challenge” for retail as well as the overall chain to assure consumer loyalty. On the one hand, supplies need to satisfy basic quality requirements such as appearance, taste, freshness, availability, and compliance to expectations concerning ingredients and other substances. On the other, upcoming added value services represent a competitive advantage that is specifically provided by the usage of smartphones that are evolving into a key enabler to realise multi-channel shopping. As highlighted by IM+io (4/2013), already over 40% of smartphone owners are using their devices practically for shopping, and even 56.6% of younger users in the age range between 18 and 29. Furthermore, it is highlighted that there was even a yearly increase of 8% compared to 2012. At the same time, especially the upcoming EU regulation No 1169/2011 is asking for the provision of food information to achieve a high level of health protection for consumers and to guarantee their right to information.

These competitive challenges and legal requirements can be considered rather as opportunities than limitations. Specifically the need for information cannot be satisfied by a single actor in the agri-food chain. There needs to be a joint effort along the chain that requires an interaction between individual actors in the chain for realising features that enable the information exchange between “suppliers of a supplier” with “customers of a customer” to finally satisfy the needs of the consumers. At the same time, actors along the chain can also benefit from this information to optimise their processes and coordination of the workflow. Therefore, it is assumed that a too late uptake of this opportunity will cause a direct competitive disadvantage.

As outlined in Verdouw (2013) and Sundmaeker (2014a) based on the characteristics of the business sector handling perishables, the current state of the art of information and communication technology (ICT) in the agri-food logistics is characterized by large amounts of available data but a poor level of integration, and the support

for intelligent use of this data is insufficient. At the same time, the complexity of current solutions is too high and jeopardizes the development and operation of affordable solutions. Therefore, the sector is facing a large heterogeneity of actors, systems and solutions that cannot be fundamentally harmonised or changed. However, by making a virtue out of necessity and systematically handle the heterogeneous elements, added-value services for consumers can be fuelled with the required information, and actors in the chain will be enabled to collaborate and coordinate workflows based on a fundamentally changed information basis.

ICT in general – and more specifically the so called Future Internet (FI) initiative – is currently working on a large set of enablers as outlined in Sundmaeker (2013). This initiative offers a new dimension of added value features for both sides, the business actors and as a consequence also for consumers. It addresses diverse sectors, while an international consortium of partners from agri-food business and ICT is currently working on the development of a B2B collaboration platform – the so called FIspace platform[1] can specifically close the communication gap within complex network structures of actors producing, processing and handling food. It can help actors in the chain to coordinate their activities with respect to the business processes and handled objects as well as to facilitate the exchange of information.

The initial work on the FIspace platform specifically addressed the development of the different core components of the platform that are rather generic than sector specific. Subsequently, business sector specific software applications are developed. The initial work was focused on the FIspace architecture but also aimed at an open design to allow for an easy integration with existing and upcoming systems and solutions. Specifically smartphone related apps are considered as one of such systems that can be employed for both business actors and consumers. This requires an integration of systems as well as an analysis of architectures and features to appropriately design future ICT support. This paper presents an approach to systematically analyse and combine smartphone apps with FIspace supported ecosystems.

Based on examples, it is described how to develop and combine FIspace related and business sector specific software applications. Basic architectural principles for application design are discussed as well as how different applications or different instances of an application can interact. This considers the perspective of the single user as well as the collaboration of actors in the agri-food chain. For discussing the combination of smartphone apps with a FIspace ecosystem, the basic development options for smartphone apps are analysed that are generally operating in native environments like Android, iPhone, Windows Phone or Blackberry. Since those realisation alternatives have a direct impact on the combination with a FIspace ecosystem, it is further analysed which characteristics the requirements analysis process needs to focus on for being able to make the most appropriate decision with respect to the end-user requirements.

2 Methodology

The core FIspace platform is designed as a generic SaaS platform, not specifically dedicated to any specific business sector. Main purpose of the platform is to support the collaboration of different business actors within a supply chain. Therefore, especially the interaction between different actors or organisations is addressed, and the platform core components offer basic features like user dashboard, workflow control, app store, event processing, and system interfacing. It is assumed that any sector incorporating collaboration between suppliers and customers (e.g. agri-food, transport, logistics, manufacturing, health care) can make use of the core platform and its components. As outlined by the FIspace Project (2013), the FIspace platform is a Future Internet based extensible Software as a Service (SaaS) platform that will enable the seamless, efficient, and effective business collaboration across organizational boundaries and will facilitate the establishment of ecosystems with business benefits for both stakeholders from industrial sectors as well as the ICT industry. Extensibility of the FIspace platform is achieved by (1) addition of functionality through software applications and (2) configuration of the platform through collaborative workflows based on the specific needs of industry users.

To test and validate this assumption from an agri-food perspective, different teams, located in several European regions have been realising trials addressing different types of perishables (e.g. fruits & vegetables, flowers, meat, fish) as well as different processes in the agri-food chain (e.g. smart spraying, greenhouse management, quality controlled logistics, retail). Especially business end-users like farmers, traders and retailers were involved in the requirements analysis, while on of the involved retailers was involving consumers to test their perception with respect to smartphone apps. At the same time, partners that are active in standardisation initiatives (e.g. KTBL, Floricode, GS1, The Open Group) elaborated requirements based on their involvement in end-user groups (FIspace (2013b)). Those requirements were used to test and validate the overall architecture of the FIspace platform. Subsequently, different trial teams focused on specific collaboration aspects of business actors as well as

[1] The FIspace B2B collaboration platform http://www.fispace.eu/

consumers as presented in FIspace (2014a). In this paper, specifically the software applications that were developed for end users in the fruits and vegetables chain will be further outlined.

The analysed trial developed two applications, while addressing the exchange of information that can be assigned to three main categories: (A) product characteristics, (B) workflow status and (C) status of objects handled in the process. The software solutions themselves were tested and validated, their usage in combination with the FIspace platform as well as the combination of applications via the FIspace platform. The latter specifically addresses the implementation of the inter-application workflow via a FIspace platform hosted workflow engine. In parallel, it was analysed which data shall be exchanged via a platform and which directly between different applications; this also addresses classical questions regarding ownership of data and whether the related data shall be stored centrally or in a decentralised fashion.

Based on those experiences, it was analysed how to combine and integrate such FIspace related software applications with smartphone apps. It was not necessarily expected to realise new features that cannot be realised by the initial software applications, but rather to offer a different kind of user interfaces and user experience (UX). More specifically, the intention was to exploit the potentials of hardware that is already available in business processes as well as in the hands of consumers, for not reinventing the wheel and to avoid the purchase of additional hardware.

In summary, those trials specifically served the analysis and discussion of aspects with respect to interoperability, end-user related customisation, and usability. With respect to smartphone apps, a set of app characteristics was elaborated that shall be used in the scope of requirements analysis and app design.

3 FIspace Software Applications

3.1 PIA – the Product Information Application

Software applications are the main enabler to assure an extensibility of the FIspace platform. Within the FIspace fruits and vegetables trial, the focus lies on the exchange of information along the supply chain. As outlined in Sundmaeker (2014b), the related software application – Product Information App (PIA) – uses a systematic data model that is structured in process and product related data. The main purpose is to enable the exchange of product quality related data in parallel to the flow of physical goods (i.e. shipments) between actors in the food supply chain such as farmers, traders, and retailers. As presented in FIspace (2015), the app development was structured in three main releases to test and validate the following:

1. Basic app functionality from a domain specific end-user perspective.
2. Communication between different app instances and coordination of the workflow via the FIspace platform B2B collaboration core.
3. Adopting the flexible FIspace platform capability model and validating the coordinated workflow support.

The first two releases focused on the PIA and its interaction with the FIspace platform, while the third release is based on an integrated usage of different software applications in the scope of the fresh fruits and vegetables trial. This enabled the validation of the workflow design with the FIspace platform. Several test instances of the PIA running in different stages of the chain (i.e. farmer & trader) provided messages within the workflow that were also used by the so called BOXMAN application. The main goal of BOXMAN is to enable significantly simplified management of RTIs (reusable boxes, crates and pallets) with different pool management organisations (PMOs) within the food supply chain for both PMOs and their customers (e.g. traders and retailers). In the case of the aforementioned test, this is achieved by BOXMAN making use of the RTI information that is part of PIA data model.

In more detail, the third release of PIA provides the following features in the form of a W3C-widget compliant frontend (GUI), and a backend with data storage:

- Configuration of the app:
  - Adopting the contacts that are managed in the FIspace platform.
  - Setting up one or more delivery locations for shipments.
  - Define product favourites from a product master list.
  - Assign primary and secondary packaging favourites from a master list to each selected product.
  - Assign quality-related product information attributes from a master list.
  - Exchanging, updating and extending the data master lists.
- Displaying the history and current status of incoming and outgoing shipments.
- Compiling outgoing shipments and assigning product information attributes to the related products (manually and/or based on pre-configured attributes).
- Assign access rights for product information attributes.
- View and edit details of an outgoing shipment (products, packaging and product information).

- Announce deliveries to customers.
- Accept or reject incoming goods and provide product quality feedback to suppliers.
- Linking the outgoing with the incoming products to enable traceability.
- Backup and restoring of the data storage.

As presented in Figure 1 PIA consists of a backend with data storage and a frontend. The backend connects to the FIspace Platform via the System and Data Integration Module and to the app data storage via dedicated interfaces. It also serves as a connector between the data storage and the frontend. The backend exposes a part of its functionality to the FIspace Platform in the form of (business) capabilities. In addition to that, an API allows other instances of PIA – or other FIspace apps such as BOXMAN – to access shipment data and product information. This is due to the convention that payload data shall not be transferred directly via the FIspace platform, that is, the platform data traffic shall be limited to notifications about available resources and the URI thereof. The backend is implemented in Java and uses the Spring Framework to provide RESTful web services to the frontend to perform CRUD operations on the app data storage. To enable AJAX calls from frontends located in other domains, the backend allows cross-origin resource sharing (CORS). In addition, the communication between frontend and backend is secured via HTTPS. One PIA backend can serve multiple frontends, and to make the app scalable, there can be multiple instances of the PIA backend working together. Each backend decides whether payload data needs to be transferred depending on whether the frontends of two collaborating PIA users are connected to the same backend or not.

To store the data that is being exchanged between stakeholders in a supply chain a Mongo DB is used, enabled by the Spring Framework. The communication between frontend and backend is realised in the form of JSON data structures from which information items can be read and directly used by JavaScript (JS). Especially for test and validation purposes, the data storage is initialised and populated from master lists for products, packaging and product information attributes, generating random user data (configuration and shipments). The frontend provides the graphical user interface (GUI) also enabling, disabling and dynamically adding/removing GUI elements depending on the current state of the shipment and actions of the user. The frontend is a W3C widget using HTML, CSS and JavaScript. For communication with the backend, the frontend uses jQuery AJAX (Asynchronous JavaScript and XML). In addition, error handling has been implemented to maintain consistency between information displayed by the frontend and the corresponding data stored by the backend in case an API call to the backend might fail.

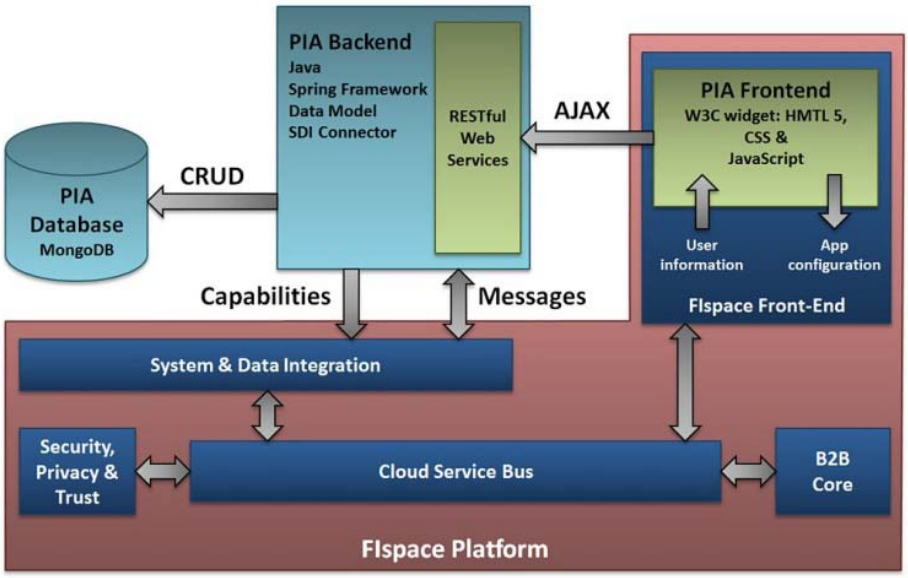

Figure 1: Interaction of the PIA with FIspace Platform components

3.2 Widget based Application Integration

As presented in Figure 1, the application is presented via a widget that is running within the FIspace frontend. For being able to do so, it queries the information about the logged-in user, enabled by the single sign-on mechanism authenticating the current FIspace user towards both the FIspace security component (SPT) and the FIspace frontend. The widget forwards this user information to the app backend for being able to access the PIA user data. Subsequently, the widget can use the FIspace frontend mechanism to persistently store user preferences in order to save the app configuration data, which comprises the PIA backend IP address and the port.

From a FIspace platform point of view, the PIA backend is considered as an external system. For being able to systematically integrate those "external systems" as well as to limit the amount of individual interfaces, the FIspace platform incorporates the System and Data Integration (SDI) component. The SDI uses the concept of "business capabilities" to enable each app to provide its functionality to FIspace capabilities. In the FIspace platform they are offered as general and thus reusable functionalities. Based on this, business architects can make use of app functionalities by invoking suitable capabilities in the workflow associated with a business process.

3.3 Workflow Design for Business to Business Collaboration

The domain specific applications provide the functionality that is obviously required by the business end-users, while the FIspace platform components offer a kind of background functionality[2] that enables the composition and support of the workflow. The main platform component supporting the workflow design is the so called B2B collaboration core. Using a workflow editor, which is support tool of FIspace's B2B component, a business architect can make use of the software application's capabilities. Each capability that an app offers has a capability type, which is associated with a set of messages that the app must be able to send or receive. These messages have defined message types, which are known to the B2B core and enable it to convert between message types. This allows the exchange of messages between different app types (via a business process) without the need for an app type to know the peculiarities of the other type.

Based on this principle, the workflow in a fruits and vegetables scenario was generated for test and validation, by using two instances of PIA (i.e. supplier and customer side) and an instance of the BOXMAN application. As soon as the platform related workflow is configured, the B2B collaboration core in the FIspace platform enables the exchange of messages. The following Figure 2 presents this workflow, describing how messages exchange product-related quality information in the context of shipments between two PIA instances and BOXMAN. Especially the latter receives information about type and number of reusable packaging, so-called returnable trade items (RTIs), from two PIA instances generated in the business process.

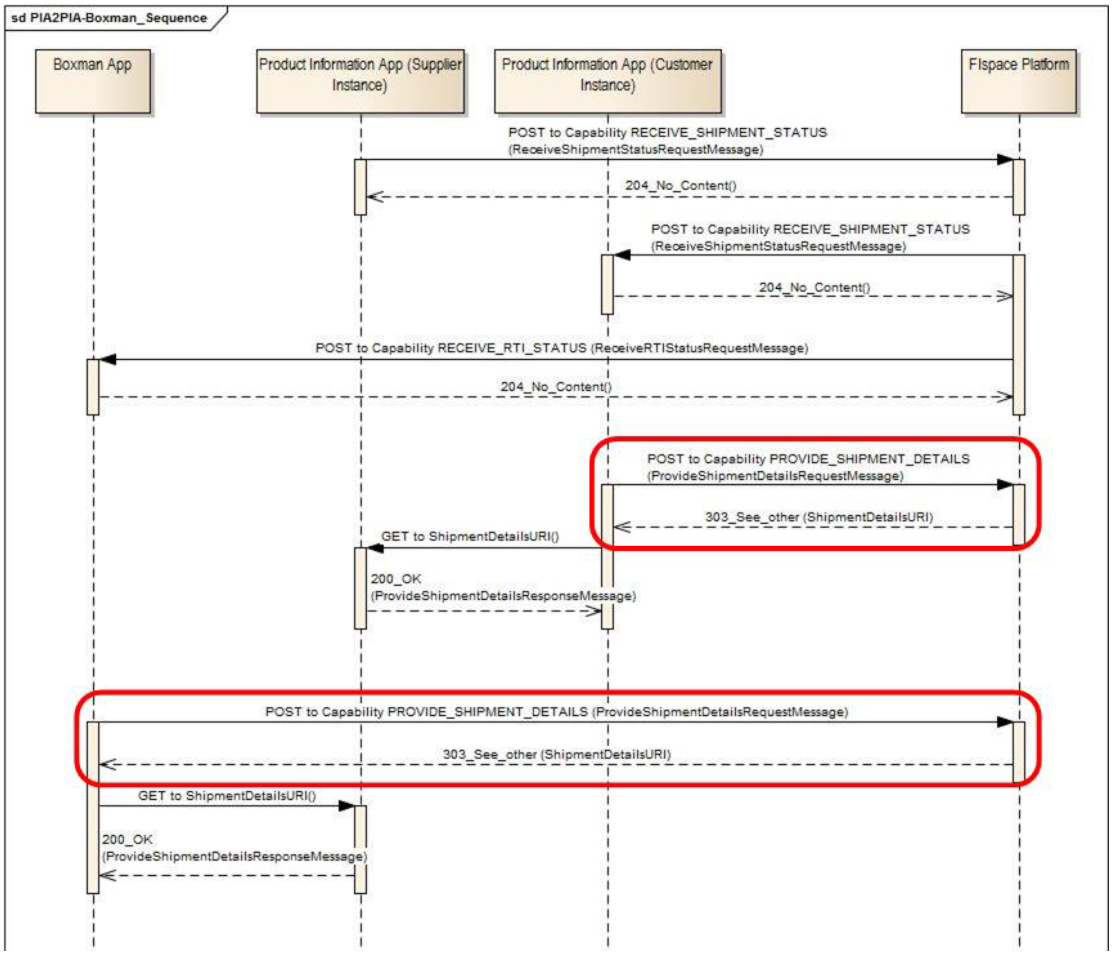

Figure 2: Workflow of two PIA instances and the BOXMAN app (with an abstract view on the FIspace platform)

[2] This background functionality represents a kind of non-functional features that are required for assuring a secure, scalable and coordinated communication within a network of business actors.

As initially outlined, the three software applications are connected to the FIspace platform via the SDI and have registered their capabilities so that their functionalities are accessible to business processes. The sequence is initiated by the PIA supplier instance sending a message to the SDI module. According to the behaviour defined in the associated business process, a POST operation on the capability RECEIVE_SHIPMENT_STATUS of the PIA customer instance is executed to notify about a status change of a shipment. However, this information is not directly exchanged between applications, but mediated by the B2B Core. Additionally, a "parallel" POST operation is executed on BOXMAN's capability RECEIVE_RTI_STATUS to notify the app about the availability of new RTI information. Once notified about the availability of data (e.g. shipment status, product quality information or RTI data), the PIA customer instance as well as the BOXMAN app can access the data directly – at the moment, the URI to the data is contained in the notification message itself. An alternative currently implemented in FIspace is to store the URI of the available data within a registry on the FIspace platform. Applications that receive an availability notification will subsequently request the URI of the data from the FIspace platform. Since this approach buffers the URI, it allows to deal with cases in which the target app cannot receive the URI immediately.

Therefore, the main communication channels that are supporting the business interaction in a FIspace based ecosystem can be summarised as follows:

- Internal communication of the application backend with the application front-end that is designed by the application developer.
- Persistent storage of workflow related data directly by the application backend.
- Advertising of application capabilities by the application to the FIspace platform.
- Provision/publishing of messages by applications to the FIspace platform
- Forwarding of messages received from applications to other applications by the platform B2B collaboration core. This represents a publish-subscribe mechanism for applications.
- Direct access of application related data by individual web-services that are hosted and maintained by the software application.

This approach applied in a FIspace ecosystem enables to combine heterogeneous systems that are following the systematic approach for advertising and using capabilities. Moreover, this approach does not constrain the specific technologies and architecture that are used for implementing the software application itself. It even enables combining existing (legacy) systems that are already operational within the workflow. At the same time, it is assumed that smartphone apps can represent an additional UI. The envisaged potentials of such apps are discussed in the following section, while also further analysing different app architectures to be taken into account.

4 Smartphone Apps – Potentials and Development Options

4.1 Cross-Platform Development

According to IM+io (4/2013), especially smartphones are the key enabler that accelerated the usage of Future Stores (i.e. shopping with a combined usage of the digital world). On the one hand, new shopping behaviours like "showrooming" are upcoming and the exploitation of users' data is becoming an opportunity. In that context Scheer (2014) highlights that this becomes problematic as soon as the end-user is not aware about the data usage or if their willingness to provide personal data is worth the service provided in return. On the other, as identified in the strategic research agenda of the cluster of European research projects on the Internet of Things in Sundmaeker (2010), there are diverse potentials for retail, logistics and supply chain management. Savings of 3-9% of sales worldwide are estimated just by avoiding that shelves go empty, facilitated by the related technology. At the same time, data from retail can be used to optimise logistics in the whole chain. Finally, end-users can be supported with shopping related guidance, fast payment as well as product related information for detection of relevant allergens or verification of an uninterrupted cold chain. However, prerequisite for such shopping experiences and synergies is the availability of appropriate hardware that is required:

- to run context related software,
- for identification of objects (either for classes of objects or unique items),
- storage of product, process, context and object related data,
- for communication between the different actors in the chain.

Smartphones are ideally suited to satisfy those requirements. They provide sufficient performance to run required apps and offer huge storage capacities. At the same time, cameras and NFC scanners can be used to identify things by their appearance or diverse kinds of identification technologies (e.g. barcode, QR-code, RFID). Furthermore, the offered communication interfaces are suitable for local and Internet connectivity (e.g. Bluetooth, WiFi, GPRS, UMTS, LTE) as well as to identify the user's context with respect to the location using e.g. GPS or WiFi.

When looking at the available smartphone hardware and more specifically the native operating systems, major players are Android and iOS as well as Windows Phone and Blackberry. Each of those different platforms is asking for different development approaches and technologies. An app developed for one native platform generally cannot run on another one. Therefore, just talking about "smartphone apps" is a kind of over-simplification, since even the development on one native platform can use different development strategies. As a consequence, one cannot just identify one type of smartphone app that should be combined with a FIspace ecosystem. There are different solution alternatives that might fit or even complicate the interoperability with FIspace.

Therefore, as soon as the development is being planned, it needs to be carefully decided on which native platform an app shall run. On top of that, as Francis (2013) is highlighting, an app cannot just rely on one native platform, if the app shall get in the hands of the masses. Therefore, one needs to evaluate the approach for enabling a cross-platform development. Sero (2014) identifies the following general options:

- Option A: Complete cross-platform development, also called a mobile website
- Option B: Cross-platform core, interacting with native views
- Option C: Native core, with native navigation and feature specific cross-platform views

Another Option D can be added as outlined by Francis (2013) and Viswanathan (2014), namely the usage of a cross-platform software development kit (SDK) like Corona, RhoMobile, PhoneGap, Appcelerator, MoSync or WidgetPad that abstract the application development from a specific mobile operating system. The following Table 1 lists those cross-platform development options in relation to key consequences, based on the assessments of Sero (2014) and Francis (2013).

Table 1: Consequences of smartphone app cross-platform development options based on Sero (2014) and Francis (2013)

Development Option	Consequence
Mobile Website	Large amount of code can be reused. Examples are the reuse of own Javascript (JS) libraries, of same HTML templates for the widget as well as reusing the model to organise and store data. As a consequence, it is possible to largely reduce efforts and related costs. It is hardly possible if at all to assure a platform related user experience (UX) and end-users won't be able to use native platform specific GUI elements. Based on constraints of running JS in a sandbox, the app's performance is limited and can cause issues with high performance requirements. At the same time, it remains questionable, if all device hardware features can be used what is generally constrained, due to security reasons.
Cross platform JS core with native views	Assuming that an app requires an amount of business logic, one can encapsulate the related JS and create a well-defined interface to reuse the code. At the same time, the GUI can be realised in the form of native UIs to optimise the UX. Sero (2014) also highlights that Android and iOS support running JS code without a web-view. However, it needs to be taken into account that the JS frameworks on Android and iOS are largely undocumented and interaction between native code and JS can cause issues. This could add a high risk to a project that is difficult to estimate at the beginning.
Native core & navigation with feature-specific cross-platform views	The native implementation offers excellent performance for complex and fast animations as well as assures the UX based on standard GUI elements for navigation. If also following the guideline to use native technology for hardware related features like gestures, multi-touch or geo-location, compatibility issues will be avoided. The implementation of cross-platform views for anything that is quite static while even complex with non-standard layout, HTML and responsive CSS could be helpful to reuse code on different platforms. However, this option represents the most expensive alternative and could still invoke problems with JS interoperability.
Cross Platform SDK	The SDK highly reduces effort for developing native operating system related code. Also responsive design of the UI is supported. But more importantly, the developed app will run on all supported native platforms. However, also the SDKs have drawbacks, especially with respect to UX. UIs are hardly native, while there might be a limited support of native platform features. At the same time, troubleshooting could require additional efforts, due to automatic code generation. On top of that, also those SDKs do not necessarily support all native platforms.

4.2 Prioritisation of Development Characteristics

After discussing the consequences of cross-platform development options for smartphone apps, one can identify the following main characteristics that need to be prioritised when aiming at the realisation of an app:

- Costs and effort for development, bug fixing, maintenance, and extension
- Usability of the app and envisaged user experience (UX)
- Performance of the app
- Possibility and consequence of reusing code
- Software development expertise required

It can be summarised that there is not a "silver bullet" that will enable the realisation of the most appropriate UX at lowest costs, but that there are different decision criteria in relation to the app purpose that will help to make the most appropriate decision. In relation to a FIspace ecosystem for the combination of smartphone apps with the FIspace platform as well as with its apps, at first one has to identify the related end-users. As long as the envisaged app shall run in a business environment with quite homogeneous hardware, one could even neglect the requirements for cross-platform compatibility. However, as soon as consumers shall be involved, at least Android and iPhone, possibly even Windows Phone shall be supported due to their wide spread.

5 Using Smartphone Apps within a FIspace Ecosystem

After assessing and prioritising the characteristics for the development of apps, one needs to carefully plan which functionality has to be implemented in an app as well as to evaluate, whether certain functionality need to be assigned to the FIspace software application or the FIspace platform. This evaluation is based on the following principles for solution design within a FIspace ecosystem:

- FIspace platform:
  - Generic features for workflow control, event processing, pub/sub mechanisms for coupling applications, widget based UIs as well as security, privacy and trust are provided by the FIspace platform itself.
  - The platform is hosted, operated and maintained on a cloud-based infrastructure.
  - The platform is generally not storing business related data, unless it is required for non-repudiation purposes in a commercial environment. In this context such data can serve as proof for the occurrence of a business transaction. Platform user data will be kept as far as it is required for the authentication and authorisation of transactions.
  - User role based mechanism in combination with the grouping of users in relation to their organisations allows user authorisation for interactions with users of other organisations (e.g. an employee of the shipment department at supplier X is authorised to access a shipment app, while the related data is forwarded to a quality assurance (QA) application that can be accessed by the QA employee at the customer organisation Y).
- FIspace software application:
  - Domain specific features are implemented with the help of such software applications. There is generally no specific rule on limiting the amount of functionality, however, it is assumed that such a FIspace software application is dedicated to one specific step or type of activity within the workflow – this fosters their reusability.
  - The FIspace software application generally contains an application back-end to interface with the FIspace platform as well as with other applications, and to provide the business logic and data storage.
  - The application front-end is deployed in the FIspace dashboard as W3C widget, and the application is offered via the FIspace app store.
  - The exchange of data within the business process is generally split in transactional data that is used to control the workflow and content data that is required by other applications to display specific information to end-users. The latter shall not be directly transferred via the FIspace platform to reduce the amount of central data traffic. Therefore, FIspace software applications shall be capable to offer web-services other apps allowing access to the content data.
- Smartphone app:
  - Features that require the availability of a mobile UI, due to the specifics of a specific working step or due to the dynamic or temporary access by related users.
  - Apps can be used as a kind of stateless data input device, to directly provide required data to the B2B collaboration core or more specifically the event processing engine in the FIspace platform. This is considered as a unidirectional communication, while the communication would require a specific configuration to allow a proper usage of the communication channels and authenticate the source of data. The smartphone app developer would need to coordinate the specific interfacing with the business architect that is implementing the workflow and event handling in the FIspace platform.

- Apps can also be directly coupled (via RESTful web services) with FIspace software applications that are possibly developed by the same developer. This would allow extending the functionality on smartphones, while the key responsibility for designing and coordinating the related workflow would remain with the FIspace app developer. This approach would also decouple the smartphone app and its development from the FIspace software platform. End-users could access context related information or even be notified about relevant events that require an activity of the end-user.
- SPT mechanisms need to be assured by the smartphone and FIspace application developers. Encryption of data exchange, authentication as well as authorisation could be based on the principles of the FIspace ecosystem, while user related data should be handled within the FIspace application. This might also ask for specific sign on mechanisms that need to be provided by the FIspace application to the smartphone app, if a static configuration of the communication channel is not feasible. However, this also requires a careful analysis and review with the FIspace business architect to avoid unintended communication towards the FIspace platform. Therefore, the FIspace application is considered as mediator to decouple the smartphone app from the FIspace platform itself.
- In general, a smartphone app would be advertised in the related native store, especially when offering the app to a large target audience, like consumers. This also enables end-users to access features of a FIspace ecosystem without being directly registered within the FIspace platform. Access credentials or related configurations could be offered in the course of the general workflow taking place within the agri-food chain. However, it would also be possible that app developers individually provide the smartphone app for installation without using the native store (i.e. obviously considering the related constraints of the native platform). This might be of specific relevance for business related environments, not requiring additional mechanisms to assure trust in the smartphone apps.

From a technical perspective, the combination of a FIspace ecosystem with a smartphone app is generally based on the usage of RESTful web services. However, as outlined above, there are two basic approaches: either the smartphone app interacts with the FIspace related software application, or, it directly provides messages to the FIspace platform via the SDI component as presented in the following Figure 3.

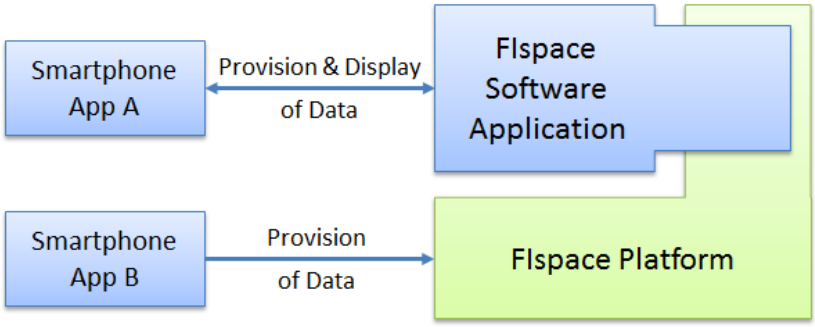

Figure 3: Basic approaches for combining smartphone apps with a FIspace ecosystem

Based on the smartphone development options as presented in Table 1, it is assumed that option B and C will enable both the display and provision of data. A native smartphone app could also automate the compilation of data as well as provide a context based notification of end-users. Option A, which is based on a so called mobile website, might constrain the functionality to some kind of explicit provision of or access to information via the smartphone app. However, this loose coupling via a web service could represent an easy to use tool at low costs, while enabling end-users of being informed about current questions in a workflow or shopping experience. Option D will highly rely on the features offered by the related SDK. However, this option offers quick solutions to realise added value functionality as well as to exploit available hardware at minimal costs. Of course, this might experience incompatibilities and a limited UX.

6 Conclusions

The FIspace platform is an innovative Internet based platform for B2B collaboration that specifically enables the coupling of different software applications that can be used by different actors in supply chains and more specifically in the agri-food chain. It enables the design of workflow control, while the systematic decoupling of the heterogeneous elements in a FIspace ecosystem facilitates the evolution of features and limits the risks in case of changes of individual software components.

Smartphone apps are an excellent opportunity to extent FIspace ecosystems with UIs and to reach a large number of end-users without the need to introduce new hardware in the existing collaboration. However, application developers need to be aware of the basic architecture of the FIspace platform and how to distribute/ allocate

specific functionality. This must also be based on the specific smartphone app implementation option, which determines the capability of the app and could directly constrain the UX. In parallel, the product owner needs to analyse the end-user requirements with their implications on the app development characteristics, keeping in mind that one generally cannot achieve a global optimum. Specifically costs and time for app development will be increased when aiming at an optimal UX and performance. Therefore, it is assumed that the product owner needs explicitly to prioritise certain characteristics, based on the envisaged types of end-users.

Nevertheless, the FIspace platform offers a high degree of freedom regarding the choice of development environment, how to design an app/application, and where to locate specific functionality in the domain specific part of a FIspace ecosystem. At the same time, it is possible to evolve the functionality of applications and apps, since related workflows and event processing patterns can be updated in the FIspace platform accordingly.

A limited first release of the FIspace platform was most recently published via www.fispace.eu for demonstration purposes. There will be updates in the next months, while interested developers can already access tutorials and documentation. Finally, it is planned to provide a so called FIspace lab as an experimentation environment for app developers. The main developers of FIspace are currently finalising their plans for the provision of commercial platform instances that will be provided for an operational use. Therefore, interested application and app developers can already start initial developments from a business domain perspective, while the integration with an operational FIspace instance can be realised in accordance to the availability of the commercial offerings.

7 Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 604 123.

8 References

FIspace Project (2013a); FIspace Technical Architecture and Specification. Public Deliverable D200.2 of the FIspace project, 30.10.2013.

FIspace Project (2013b); Guidelines for the use of Standards in FIspace. Public Deliverable D500.4.1 of the FIspace project, 19.09.2013.

FIspace Project (2014a); Progress report on trial experimentation and App development and updated plan for Phase 3 rollout. Public Deliverable D400.4 of the FIspace project, 22.10.2014.

FIspace Project (2015); Initial Applications Third Release. Public Deliverable D400.9 of the FIspace project, 09.01.2014.

Francis, W.J. (2013); Cross-platform vs. native development: Corona SDK. In Software Engineer, 03.07.2013; Available at: http://www.techrepublic.com/blog/software-engineer/cross-platform-vs-native-development-corona-sdk/

IM+io Fachzeitschrift für Innovation, Organisation und Management, 04/2013. Avatare schwitzen nicht: Das Konzept der Future Stores kommt nach und nach in der Gegenwart an. 28. Jahrgang, Heft 4, 2013, pp 8-9.

Regulation (EU) No 1169/2011 of the European Parliament and of the Council of 25 October 2011 on the provision of food information to consumers, amending Regulations (EC) No 1924/2006 and (EC) No 1925/2006 of the European Parliament and of the Council, and repealing Commission Directive 87/250/EEC, Council Directive 90/496/EEC, Commission Directive 1999/10/EC, Directive 2000/13/EC of the European Parliament and of the Council, Commission Directives 2002/67/EC and 2008/5/EC and Commission Regulation (EC) No 608/2004 (Official Journal of the EU L 304, 25.10.2011, S.18).

Scheer, A.-W. (2014); Mit Daten Bezahlen. In IM+io Fachzeitschrift für Innovation, Organisation und Management, 04/2013. 28. Jahrgang, Heft 4, 2013, pp 6-7.

Sero, M. (2014); Native vs. Hybrid: developing cross-platform mobile apps. 30.07.2014; Available at: http://www.theappbusiness.com/native-vs-hybrid-developing-cross-platform-mobile-apps/

Sundmaeker, H. (2014b); Business Collaboration in Food Networks: Incremental Solution Development. 8th International European Forum on System Dynamics and Innovation in Food Networks, 17-21.02.2014 - Innsbruck-Igls, Austria.

Sundmaeker, H., Guillemin, P., Friess, P., Woelfflé, S. (Eds.), 2010. Vision and Challenges for Realising the Internet of Things. European Commission, Brussels, March 2010.

Sundmaeker, H.; Verdouw, C. (2014a). Dynamic Business Collaboration in Supply Chains with Future Internet technologies: Acceleration of SME-driven App Development. International Conference on IoT as a Service, 27.-28-10.2014, Rome, Italy.

Sundmaeker, Harald (2013). Realising Future Internet Potentials for Food Chain SMEs: A Hierarchy of Needs. Int. J. Food System Dynamics 4 (4), 2013, 273-282; available online at www.fooddynamics.org.

Verdouw, C., Vucic, N., Sundmaeker, H., & Beulens, A. (2014). Future Internet as a Driver for Virtualization, Connectivity and Intelligence of Agri-Food Supply Chain Networks. International Journal On Food System Dynamics, 4(4).

Viswanathan, P. (2014); Top 5 Tools for Multi-Platform Mobile App Development. Available at: http://mobiledevices.about.com/od/mobileappbasics/tp/Top-5-Tools-Multi-Platform-Mobile-App-Development.htm