Published in: International Journal of Service Science, Management, Engineering, and Technology (IJSSMET), Volume 2, Number 4, 2011, pp.79-97

uRun – A Framework for User-Generated Mobile Services in the Health and Fitness Domain

Alexandra Chapko[4], Andreas Emrich[4], Stephan Flake[1], Frank Golatowski[2], Marc Gräßle[4], Andreas Kohlos[3], Nico Laum[2], Christian Lerche[2], Carsten Rust[3], Jürgen Tacken[1], Dirk Werth[4], Carsten Zoth[1]

1 Orga Systems GmbH, Am Hoppenhof 33, 33104 Paderborn, Germany
2 University of Rostock, Richard-Wagner-Str. 31, 18119 Rostock-Warnemünde, Germany
3 Morpho Cards GmbH, Riemekestr. 160, 33106 Paderborn, Germany
4 German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany

ABSTRACT

This article presents a framework which enables end users to create small, sharply focused mobile services directly on a mobile device. By this, end users are no longer only consumers of mobile services – they also become producers and providers of mobile services. The domain of mobile health and fitness applications has been chosen to demonstrate the feasibility of the approach. The article presents the underlying platform for easy creation of mobile services and describes our implementation of a Web-based editor for easy mobile service creation as well as our solution to access device capabilities out of Web applications.

Keywords: Information Systems, Healthcare, Service design and modeling, User-generated mobile services, Super prosumer, Mobile health applications, Web services-based sensor integration

1. INTRODUCTION

The number of people with chronic diseases like diabetes and arthritis is growing worldwide. Actions to encounter this trend take an ever-increasing proportion of national health care budgets. In view of the demographic change, effective prevention of diseases becomes an inevitable demand for our society. In this context mobile technology, intelligent assistance systems and user-drive may play an important role. According to a study from eMarketer, there were nearly 116 million US user-generated content consumers, along with 82.5 million content creators in 2008. Both numbers are expected to increase significantly by 2013 (Verna, 2009). Furthermore, usage of mobile devices, such as smart phones or tablets, has overtaken application of 'classical' devices like PCs or laptops. According to another study (Flurry Analytics, 2011), daily time spent in mobile apps has surpassed Web consumption. At the time of writing, the average user spends 9% more time using mobile apps than the Internet.

As a result, we expect that in the near future consumption and creation of user-generated content in the domain of health and fitness will move to the mobile environment. This will be much more suited to the way we carry out social interactions – anywhere and anytime. Soon, this will further evolve and the mobile world will provide not only contents, but also ad-hoc user-created services from the 'long tail', i.e., a large number of unique services with a comparatively small number of users. The mobile terminal will evolve to become also a server. When put into exploitation directly by using the mobile device, a mobile user will provide constantly updated information, relevant to other users' instantaneous interests.

We assume that technologies enabling the creation, deployment, discovery and sharing of user-generated mobile content and services are crucial for the further evolution of the mobile world. Individuals will become 'super prosumers', i.e., producers, providers, and consumers of mobile content and services. The super prosumer concept responds to the most relevant driver of mobile usage, which is the instantaneous and personalized response to a need for communication, information, and entertainment using powerful ubiquitous user devices and mobile network infrastructures.

This article is an extended version of two articles, in which first results of the project have been presented (Chapko et al., 2010a, 2010b). In this article, we present a more detailed view of the uService architecture, which is the underlying generic platform for mobile service management. As a concrete application area of a super prosumer scenario, the uService project investigates, among others, the health and fitness domain. This is due to the fact that people are increasingly looking for products and services that allow them to track, socially interact, and share data related to health and fitness while being on the move (Buttussi, Chittaro, & Nadalutti, 2006; Kurdyukova, 2009; Miluzzo et al., 2008). Furthermore, social interaction, challenge and reward systems are significant trends for health and mobile applications (Ashbrook, 2010). The uService project extends these ideas even further by enabling users to create mobile services on mobile devices on their own and provide them to others. The uRun framework can therefore be seen as an instantiation of the uService platform for the domain of user-generated health and fitness mobile services.

The remainder of this article is structured as follows. In Section 2, an overview about related work concerning the creation of mobile services and the current technologies for mobile running applications are reviewed. In Section 3, example scenarios for the use and creation of mobile applications in the health and fitness domain are outlined. The uService platform and the uRun framework are presented in Section 4. Finally, this article is closed in Section 5 with a conclusion and outlook on future work.

2. RELATED WORK

In this section, firstly we discuss related projects that are investigating user-generated services. Sections 2.2 and 2.3 outline important issues that are of particular importance in the field of user-generated services: While we explain how user-generated services can be secured and trusted in Subsection 2.2, we show how the revenue is shared between the participants in the eco-system in Subsection 2.3. In Subsection 2.4, we describe the state-of-the-art in intelligent mobile health and fitness systems, before we present the state-of-the-art in mobile running applications in Subsection 2.5, which is the focused application domain of the uRun framework.

2.1 Related Projects

Given that social networks have brought new kinds of social interactions between people and services in the Internet, Web Services have become pervasive for users around the globe. As a result, there are several projects that focus on novel Web Services-based applications. uService, with its goal to turn users into mobile service super prosumers by enabling them to easily create new services, shares some characteristics with recent pan-European projects.

The m:Ciudad project (http://www.mciudad-fp7.eu/) aimed to provide '*a service architecture, a set of mobile tools and a platform to allow users to create focused, knowledge-based mobile micro-services, which are called m:Ciudad U+ Services*' (Davies et al., 2008). Similar to uService, m:Ciudad followed the mobile service prosumer paradigm and can be seen as a predecessor of uService. However, m:Ciudad misses important features which are required in most applications: integration of sensors and mobile terminal capabilities and an infrastructure that provides common platform features, such as identity management, search and recommendation, charging and billing, and advertising.

The Do-it-Yourself Smart Experiences (DiYSE) project (http://www.dyse.org/) allows the user to easily create, setup, and control applications in smart living environments (Roelands et al., 2010). The project aims to make the Internet-of-Things usable and to enable a Web of Things by using Web 2.0 technologies. In contrast to uService, DiYSE rather tries to connect physical things to the Internet and is less focused on human interactions.

The Open Source Ambient Intelligence (OSAmI) project (http://www.osami-commons.org/) focuses on creating an open-source service-oriented platform for ambient intelligence applications (Lipprandt et al., 2009). The project's goal is to enable all types of co-operating devices and software to seamlessly work together. Whereas OSAmI has its focus on device and service interoperability, the creation of new services has not been investigated.

The Mobile User Generated Geo Services (Mugges, http://www.mugges-fp7.eu/) project again follows the mobile prosumer concept and aims at allowing users to provide services (Guggenmos, Pérez, López-de-Ipiña, & Gil, 2010). Services are called 'Mugglets' and are downloaded to the prosumers mobile device. Furthermore, Mugges focuses on location-based services and therefore each Mugglet is connected to a location. For the creation of Mugglets, Mugges uses a template approach, which means that users adapt ready-to-run services for their needs. The uService project considers service templates as one possible way for users to easily create new mobile services.

2.2 Security, Privacy, Trust, and Identity Management

Important issues in the context of mobile user-generated services are security, privacy, and trust as well as the closely related issue of identity management. For Web Services, WS-Security (Organization for the Advancement of Structured Information Standards, 2006) is a standard for applying security mechanisms. It defines a set of SOAP extensions providing message authentication, message integrity, and message confidentiality. WS-Security supports for multiple security tokens, trust domains, signature formats, and encryption technologies. It is a message-level security and working in the application layer, no special transport layer technique is required.

Concerning the management of identities, which is needed to perform secure authentication, user-centric identity management standards have to be considered. Well-known examples are

OpenID (http://openid.net), CardSpace and Information Cards (Bertocci, Serack, & Baker, 2007), and Liberty Alliance (http://www.projectliberty.org/).

In addition, Subscriber Identity Module (SIM) cards are also providing security mechanisms. The SIM card is well-established as a tamper-proof security device in mobile telecommunication scenarios. Its primary purpose is to identify a mobile phone user to the operator's network in a secure and consistent way. The majority of today's SIM cards is Java enabled and conforms to the Java Card 2.2 specification (Oracle Corporation, USA, 2011). There are several additional standards for SIM cards that support value-added applications (e.g., Short Message Information Service) and executable applications.

In general, SIM-based applications ensure authenticity, integrity, confidentiality, and privacy. For deploying these security enhancing applications within mobile services, concepts for communicating to the SIM on application level are needed. One frequently used ability is the SIM card Application Toolkit (ETSI TS 102 223). It defines a set of commands and procedures enabling the card to contain applications specific to the issuer (i.e., the network operator), allowing the operator to deploy services like, e.g., information and location based services, banking and Internet access.

New approaches from bodies like Java Card Forum, ETSI or Open Mobile Alliance (OMA) aim at utilizing Web technologies in the smart card world. This could extend the mashup idea to the usage of SIM cards in mobile services. A Smart Card Web Server (SCWS) is an HTTP server that is implemented in a smart card, i.e., in the mobile context in a SIM card. The SCWS can provide static content like HTML pages or media files as well as dynamic content. The content can easily be accessed, for instance from mobile phone Internet browsers.

For today's Java Card 2.2 platforms, the OMA has specified a SCWS including an interface for remote administration (Open Mobile Alliance, 2011). It will be adapted for upcoming techniques like high speed interface or TCP-enabled smart cards. The SCWS provides an extension programming interface similar to CGI (http://hoohoo.ncsa.uiuc.edu/cgi/) to deploy dynamic content. A real Java servlet container is not realized due to the limitation of the Java Card platform, e.g., the lack of multi-threading.

2.3 Revenue Sharing

Another important issue with user-generated content and services is how to make business and gain revenue. We believe that user-generated content and services should create value for their owners and everybody else in the ecosystem. Current business models in the fixed-line Internet domain offer content and services free of charge to the users, but require them to register themselves to a content and/or service platform. With this registration they provide some information about themselves and while using content and services they continuously update and extend their profile. This profile information is used by the providers to place advertisements on the Web pages, allowing them to earn money from the advertisers with their platform. But with such business models the user, or better prosumer, will not be able to get revenues for providing his content and services. Traditionally, the accounting and billing systems of the operators and Internet service providers have been geared towards billing the users, not to providing billing services to individual users. What is needed are fair user rewarding models where users receive a compensation for successful user-created content and services. This would support an innovative, fair and secure approach for mobile accounting and billing, adapted to the super

prosumer ecosystem, allowing not only the users, but also service providers to directly gain revenues with user-generated content and services.

On a technical level, we only know of one standard in the Web Services domain that provides an open interface for accounting and billing purposes, i.e., the Parlay X APIs for account management and payment (3rd Generation Partnership Project, 2009a, 2009b), now being standardized by the OneAPI initiative (http://www.gsmworld.com/oneapi/). The operations offered by the Payment API can build a basis to integrate charging mechanisms into user-generated mobile services.

2.4 Intelligent Mobile Health and Fitness Systems

Intelligent mobile health systems transform from mainly been focused on supporting medical professionals to self-care applications which are focused on the patient. Real-time feedback and improved communications to the network of care are amongst others the main features of intelligent mobile self-care applications. However, a general approach for smart and cost-efficient analysis of data for mobile health care applications is still an open issue (Haghighi, Zaslavsky, Krishnaswamy, & Gaber, 2009). In this context, you can differentiate between analysis of mobile data on the device and analysis in the backend. A combination of both is expected to provide the required quality and efficiency for health care applications. The main issue of analysis on the device is limitation of resources (e.g. battery or memory). In this context ubiquitous data stream mining (UDM) can perform real-time analysis on a mobile device while considering available resources (Gaber, Krishnaswamy, & Zaslavsky, 2004). In terms of analysis in the backend you can differentiate between emergency systems and assistance systems. Initially emergency systems were investigated and developed, i.e. the systems were designed to detect lethal threats to patients with chronic diseases (Kyriacou, Pattichis, Pattichis, Panayides, & Pitsillides, 2007). In terms of assistance systems, research is in an early phase and there are only few examples. Approaches which apply recommender system methods are either used for data mining purposes (Davis, Chawla, Blumm, Christakis, & Barabási, 2008; Davis, Chawla, Christakis, & Barabási, 2010; Steinhaeuser & Chawla, 2009) to analyze collected data for medical research purposes. Learning mechanism in combination with recommender system approaches are focused the generation of personalized, real-time recommendations for a patient. In this context, not only assistance systems for disease management (Folino & Pizzuti, 2010; Komninos & Stamou, 2006; Villarreal et al., 2009) but also assistance systems for disease prevention or healthy living are developed (Ho & Xiang, 2009; Ahamed, Haque, Stamm, & Khan, 2007).

2.5 State-of-the-Art Mobile Running Applications

Providers of mobile running applications can be differentiated according to software and hardware providers. The former develop mobile applications for mobile devices, whereas the scope of the application is limited by the technical features of the mobile device it is running on, e.g. whether a camera or GPS sensor is available. Usually, these applications are not platform-independent, and instead bound to certain mobile operating systems. Hardware providers offer mobile running applications connected to a certain device with different sensors. As a result the software application is fitted to the hardware device. In general, the most common providers are manufacturers of sports goods such as Adidas, Nike or Polar. As a last category there are hybrid running applications, i.e. hardware and software providers team up to provide applications that

run on a mobile device which in turn is connected to a another device of a sports good manufacturer.

Apples iTunes Store offers different running apps. One of the best known applications is ‘Smartrunner’(http://www.smartrunner.com/). In addition to running, Smartrunner also covers cycling, hiking and inline skating activities. The application provides an overview of average and maximum speed, distance travelled and calories burned. Furthermore, it enables a map view of the planned route. The travelled route can then be submitted to social networks like Facebook and Twitter, making it accessible for friends. Other partly fee-based apps like ‘Runtastic’ (http://runtastic.com/) offer the similar basic functions. Besides functions, such as, recording time, speed, distance, altitude and the route, Runtastic has additional functions like a compass, real-time statistics and voice feedback. The voice feedback is used to motivate the runners to keep their target in mind.

Collaboration and mutual motivation is the main objective of the project ‘Jogging over a Distance’(Mueller, Thorogood, & O’Brien, 2007). Information about speed and time of the runner is sent to a computer via a Smartphone. Since each runner is wearing a headset, runners can communicate and motivate each other, even if they are at different locations.

A product by a hardware provider is the sports mobile ‘athlosoft RUNNER’ (http://www.athlosoft.com). Using a heartbeat belt, the runner can monitor the distance, speed, step frequency and heart rate. The logged training data can be submitted to a Google Earth server, making the training track visible to others in real-time. Furthermore, the route can be transferred to Internet-enabled mobile phones, so that other users can track the path of the runners and join them if they want, even if they are already on their run. ‘fitbit’ (http://www.fitbit.com/) provides a permanent training monitoring. It is a small device that can be attached to the wrist or clothing. During the day, fitbit measures the number of calories burned. It furthermore provides an overview of rest and activity phases. The fitbit website offers data exchange, which makes a comparison of activities with friends possible.

A hybrid application is the ‘Nike + iPod Sport Set’ (http://www.apple.com/ipod/nike/run.html). Here, a sensor in the Nike running shoe records the running data and sends it to an iPod. The user can define goals, such as duration, speed and running track via the iPod. In addition, the user can select a motivational song on his iPod, which is automatically played when the speed drops below a certain level. All information can be sent to friends via Facebook and other social networks.

In summary, the majority of applications monitors the individual training by collecting different running data and applying statistics. The community aspect, however, is limited to publishing route information to social networks. The Runtastic application allows an additional data exchange on its website, where routes can be made available to others. Only Joggers Free and athlosoft RUNNER allow locating other runners using GPS. It can be seen, that all applications cover different training and social aspects. The uRun approach is different. It tries to enable the user to easily create his own customized Web-based application with all the features he requires.

3. URUN – APPLICATION EXAMPLES

In the following, a couple of uRun scenarios are described in which people are using the uService Player as a mobile running assistant and the uService Editor for mobile service

generation. Note that the term ‘uService’ will be used to denote a small, sharply focused mobile application with its own graphical user interface (GUI). First, a usage scenario is illustrated that demonstrates the variety of supporting mechanisms provided by the uService platform. Then, in Section 3.2, a service creation scenario is outlined in which a non-technical user employs the graphical uService Editor to create an example uService for the fitness domain. To better distinguish the actual scenarios from additional technical explanations in the following two subsections, all text in italic font refers to the scenarios themselves, whereas normal font denotes technical background explanations.

3.1 uRun Usage Scenario

Robert is a runner, who has recently registered for the uRun network. He is living in Paderborn. On his way to work, he remembers that he wanted to look for a running partner. In the uService Player on his mobile, Robert starts a search (cf. Figure 1(a)). *The results do not match his needs, but an interesting advertisement appears on the screen, promising that Robert can easily build his own uService as a ‘running partner search’ for a small fee* (cf. Figure 1(b)). *The service creation offered here is the creation based on a template. Thus, Robert creates a new uService by*

- *(a) providing the necessary data for the service template configuration* (cf. Figure 1(c)),
- *(b) confirming the payment of a fee* (cf. Figure 1(d)), *and*
- *(c) acknowledging the deployment of the service* (cf. Figure 1(e)).

Meanwhile Henry, another registered member of the uRun network, is on a business trip in Paderborn. He does not know the city very well. With the search functionality of the uService Player on his mobile, he wants to find out whether there are any interesting running events he could join during his stay. When looking for running events, he finds Robert’s uService (cf. Figure 1(f)). *Henry examines Robert’s proposals with their scheduled times and routes* (cf. Figure 1(g)) *and makes an appointment with Robert* (cf. Figure 1(h)).

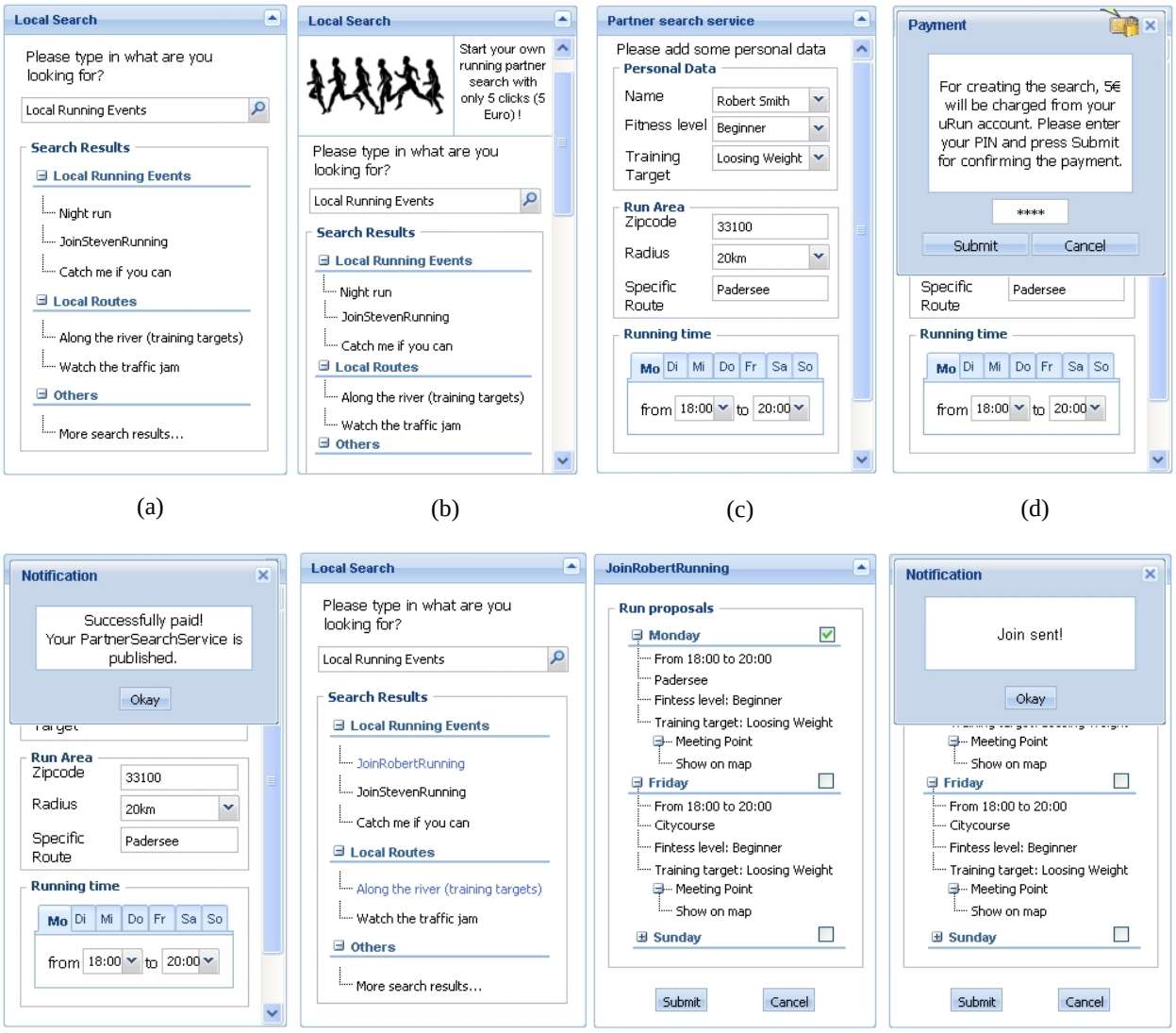


Figure 1: Robert creates a uService for Running Partner Search

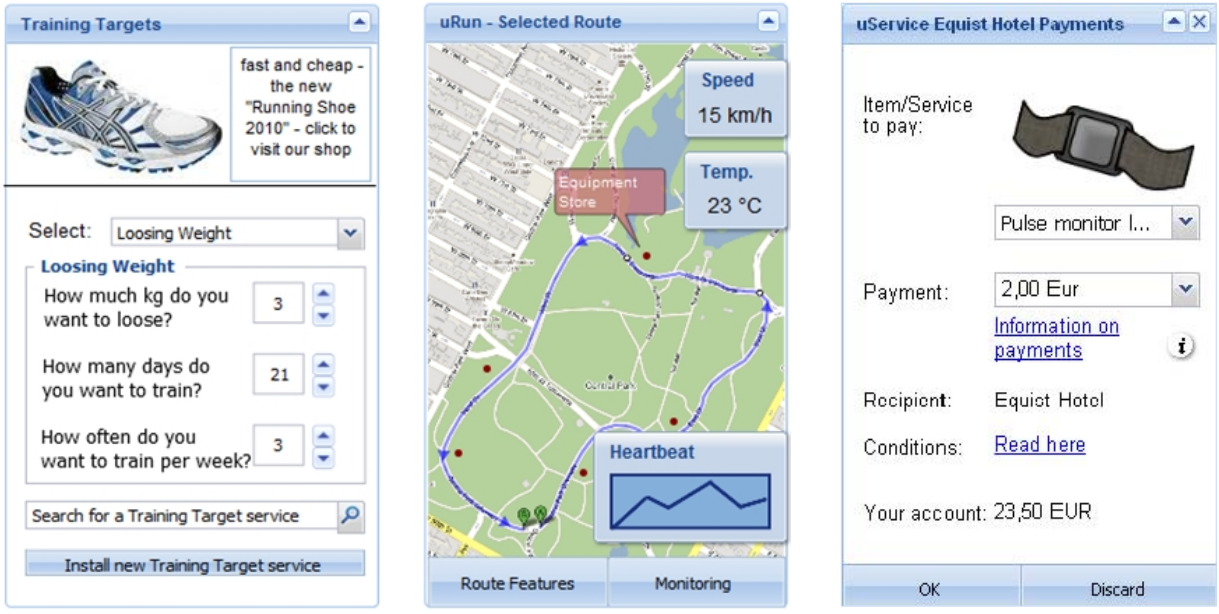
The scenario described above covers an important use case for a user-generated mobile service for runners. However, a number of additional features are necessary to support this core use case. On the one hand, the uService platform must have the capabilities to recommend services based on user profiles, to come up with targeted advertisements, and to handle payments for using uServices.

As an example, Figure 2(a) shows a targeted advertisement (here: a running shoe which is of the user's favorite brand) while the user is looking for a service to support his training targets.

On the other hand, there is the need for pre-defined graphical elements (so-called gadgets) in order to offer GUIs for uServices. A typical example is shown in Figure 2(b): a uRun monitoring

service, like ‘Carla’s favorite running track’ (see Subsection 3.1) shows a map of the chosen route and data gathered from different sensors. To deploy such services, users can make use of service templates like it is known from the Mugges project.

Figure 2(c) shows a uService for equipment rental offered by a hotel. This demonstrates that the uService platform does not only have the capability to charge for the use of uServices, but also to generate uServices that allow direct payments.



(a) (b) (c)

Figure 2: Further uService Examples

3.2 uRun Service Creation with the uService Editor

Whereas so far only potential uServices with mock-up graphical interfaces were presented, this subsection describes a service creation scenario that employs the Web-based uService Editor implementation. Further details about the technological realization are given in Section 4.

> *Carla Creative is a graphic designer living in Paderborn, Germany. She is a passionate runner and has used running applications for some time. Now she wants to share her favorite running track with other uRun users. She can do this by easily creating a new uService with the graphical uService Editor. The editor is a Web-based tool and in order to use it she has to enter her credentials to identify herself.*

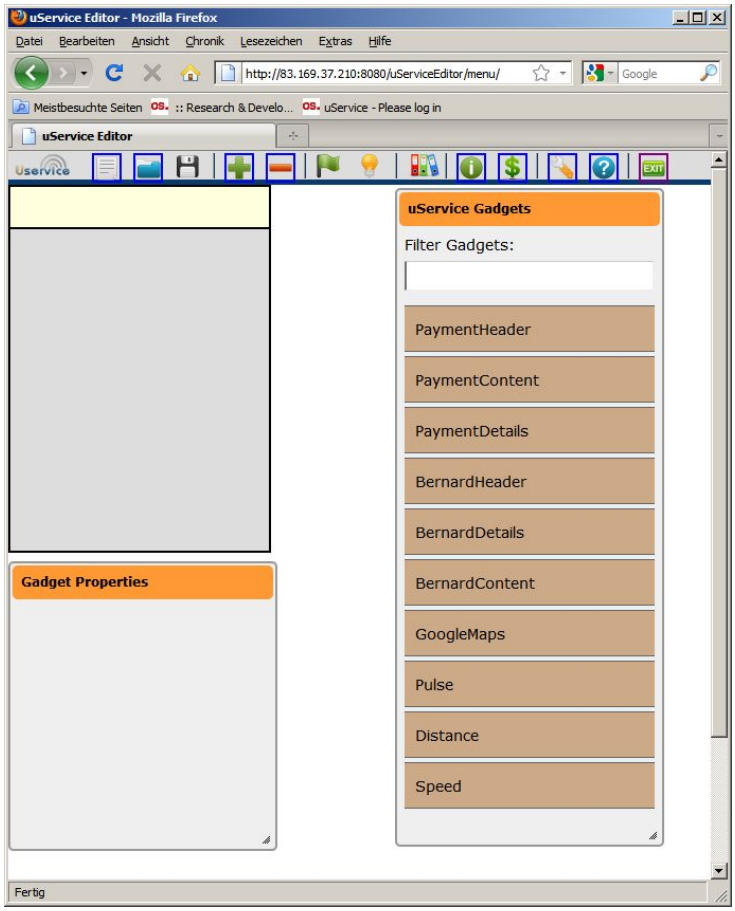


Figure 3: uService Editor – Initial View

Figure 3 shows the initial screen of the uService Editor as seen by Carla after she has provided her credentials. On the left hand side at the top, the 'viewport' is shown. This is the drawing area which represents a screen with the dimensions of the targeted display of a typical smartphone. These dimensions are configurable. On the right hand side, there is a list of available gadgets to select from. A gadget can be seen as a graphical element with embedded functionality that can become part of a uService GUI. With a mouse, gadgets can easily be dragged and dropped from the gadget list into the viewport. When a gadget is selected in the viewport, its properties are shown in the 'Gadget Properties' box at the lower left hand side of the editor screen. There, the service creator can review all gadget properties like e.g. the description of the gadget and can re-configure property values (e.g. the name) if necessary.

> *Carla enables the 'design mode' in the uService Editor by clicking on the corresponding flag symbol in the top menu. This allows her to interact with the gadget frame instead of the gadget itself. First of all, she moves the gadget 'GoogleMaps' to the viewport by drag and drop. She chooses a map of her favorite running track in Paderborn.*

Furthermore, she decides to put the gadgets for the sensors 'Distance', 'Speed', and 'Pulse' onto the screen, moves the corresponding gadgets to the viewport and drags their display fields to suitable smaller size. In order to allow the service consumers to replace the sensor displays according to their preferences, Carla sets the 'draggable' property value to 'true' in the Gadget Properties.

The result of Carla's actions can be seen in Figure 4 and Figure 5. Figure 4 shows the 'Design Mode' view. This view shows the dropped and configured gadgets in an editable mode – the service creator can move the gadgets around, resize and remove them. In the 'Design Mode' the creator cannot interact with the gadget itself, like e.g. selecting the location that should be initially shown in the Google Maps gadget.

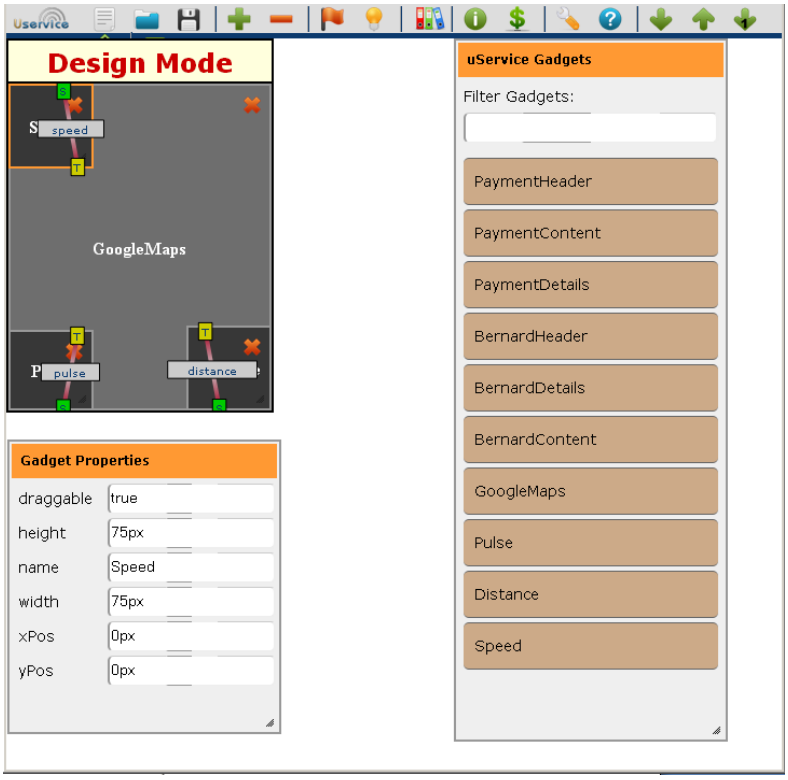


Figure 4: uService Editor – Design Mode

Figure 5 shows the 'Preview Mode'. In this mode, a preview of the display is shown with default values. Moving or changing the frame of the gadgets is not possible in this mode to prevent side-effects. But the creator can interact with the gadgets itself. Carla uses this mode to point the Google Maps gadget to the location of her favorite running track in Paderborn.

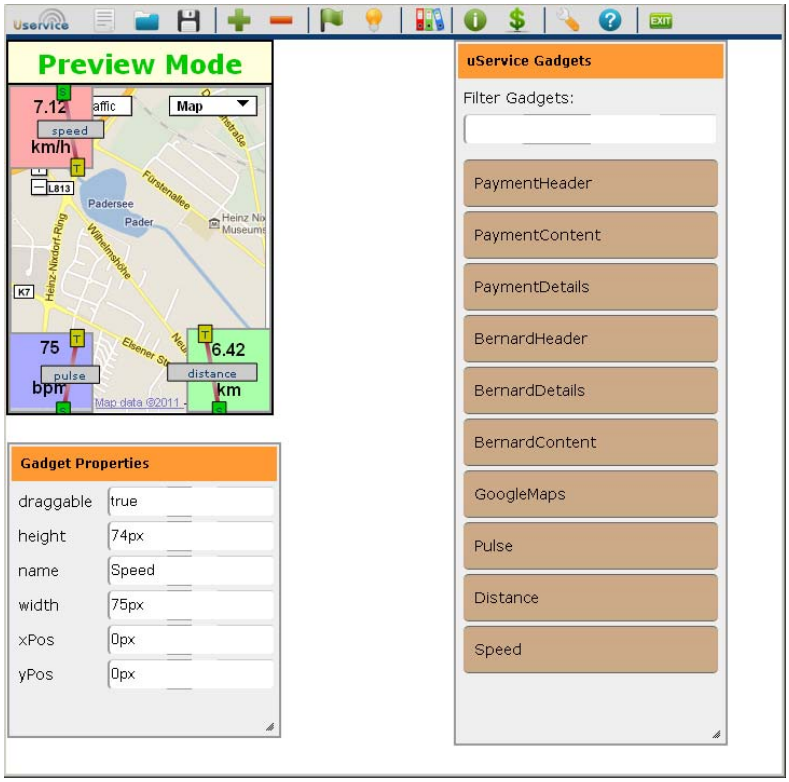


Figure 5: uService Editor – Preview Mode

> *As a next step, Carla defines the payment conditions for her new uService. She clicks on the dollar symbol in the top menu and fills in the form that appears as a pop-up (see Figure 6). She determines that a service user should pay in advance, thus she selects the chargingMode 'Pre'. As consumers should pay per use she selects the chargingModel 'Use'. Consumers shall pay 1,00 Euro for using Carla's service. She sets a validity period in order to restrict her service for being offered for one year.*

With the dialog in Figure 6 a creator is able to specify a set of payment definitions for different periods. All currently defined payment definitions with their validity periods are listed in a table in the dialog. This would e.g. allow to specify a free trial period by defining a payment definition with the charging model 'Free' for one week and then afterwards a payment definition with the charging model 'Use' valid for the rest of the lifetime of the service.

Another dialog that can be opened by clicking on the 'i'-Button allows configuring the properties of the service. These properties are, e.g., the name of the service, a description of the service, or the validity period.

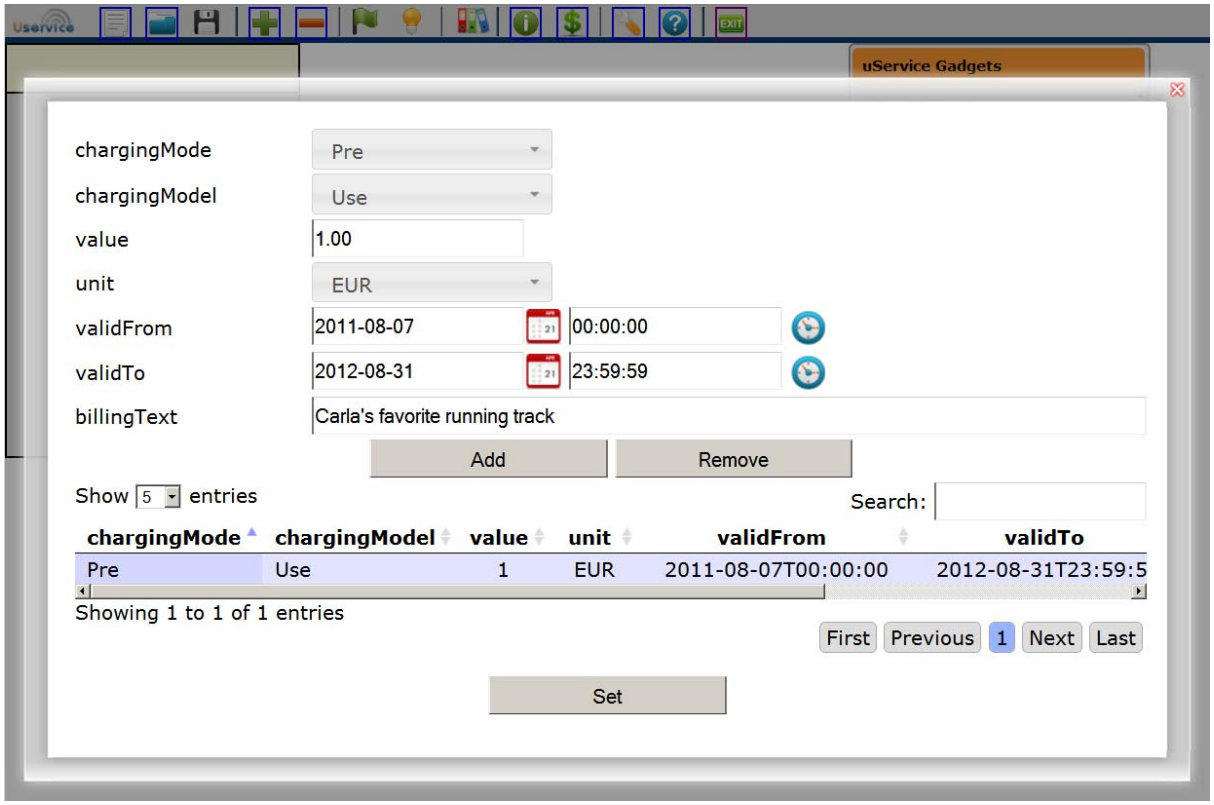


Figure 6: uService Editor – Charging Conditions

> *After specifying the payment conditions Carla uses the service properties dialog to set the name of her new service and to provide a description for it. She names her new service 'Carla's favorite running track'.*
> *With a click on the 'Save' button and a confirmation dialogue, she completes the setup and deployment of the new uService. Now other uRun users can use Carla's favorite running track with their uService Player.*

Beyond the creation of ready to use uServices, the uService Editor also supports the creation of service templates. A service template contains a set of variables that need to be filled by the final service creator to create a ready-to-use uService. The creation of a ready-to-use service from a template is even easier than the creation with the uService Editor and can be done with a mobile device while on the move.

4. THE USERVICE PLATFORM AND THE URUN FRAMEWORK

The uService platform supports sensing the users' context in both the physical and virtual world in order to provide them with appropriate recommendations for their individual situation. Figure 7 provides an overview of the uService platform architecture.

On the mobile device, users can easily create services with the Web-based uService Editor, publish them via a Publisher component, and use them with the uService Player. On the server side, the uService platform provides a Service Repository for storing and accessing service

templates and gadgets. A Publishing Warehouse component is responsible for the management of deployed services. A Search & Recommendation Engine is employed for semantic service discovery and recommendation. Security is handled by a Security and Identity Management component, whereas charging and billing is performed by an Accounting & Billing component. Last but not least, an Advertisement component is used to send targeted advertisements to the users based on their interests and other information that is allowed to be considered from their user profiles.

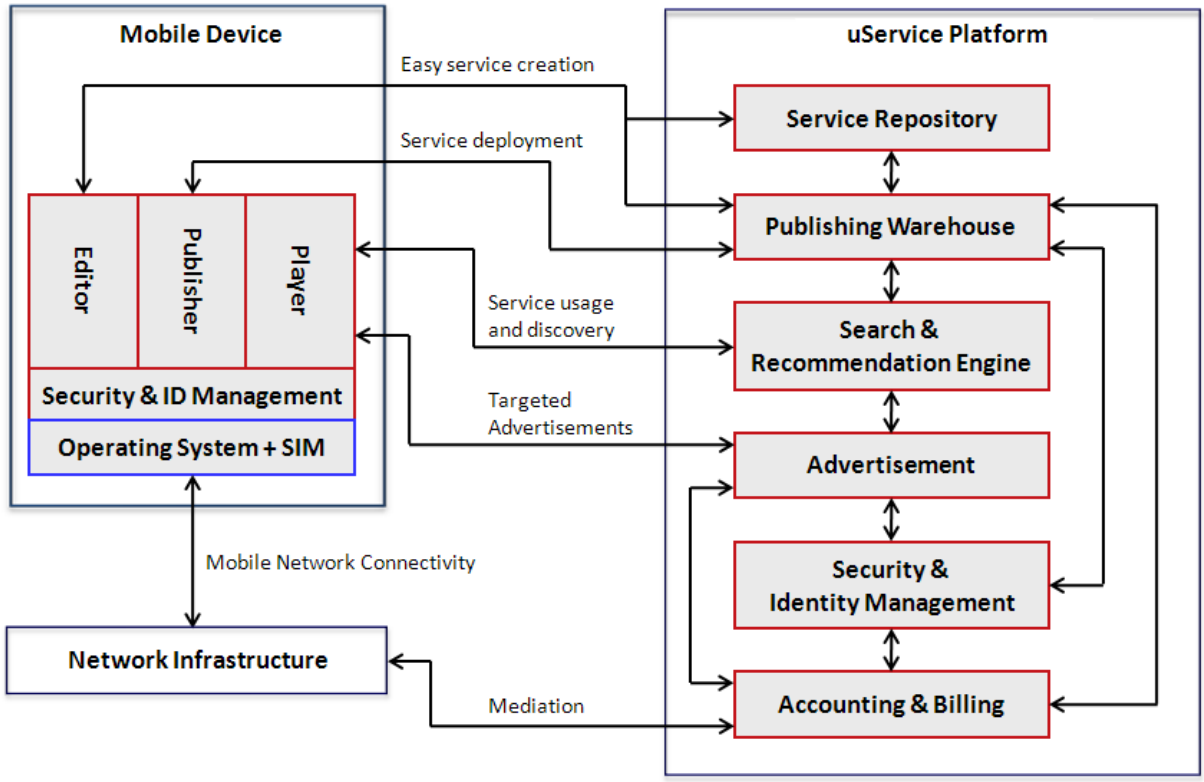


Figure 7: uService Platform Architecture
(showing only the most important connections between the components)

To come up with a uRun framework as a testbed for the considered uRun services, we have set up the server-side components of the uService platform on different servers. The components offer interfaces providing RESTful Web Services (Richardson & Ruby, 2007) to access their content and functionality. To restrict and secure the access to the Web Services, functionality of the Security & Identity Management component is used.

The uService Player and Editor components have actually server- and client-side parts and are implemented using several technologies. For the graphical framework on the server side, Grails is used (Brown & Rocher, 2009). For the client side, we use JavaScript (Haverbeke, 2011) with libraries like jQuery (http://jquery.com/) and jQuery Mobile (http://jquerymobile.com/) and an own Android App that is able to access data from a device's own sensors or other sensors connected to the device (see Subsection 4.2).

The example uRun services are implemented using so-called gadgets which are supported by the uService platform. A gadget is a dynamic Web content that can be embedded into a Web page. Gadgets are written in XML and can have HTML and JavaScript components. They can interact with each other via channels. Well known gadgets are Google Gadgets (http://www.google.com/webmasters/gadgets/) which can, e.g., be executed within Google's iGoogle personalized home page.

On the uService platform, the gadget container of the Apache Shindig Project (http://shindig.apache.org/) is used for the execution of the gadgets. This container is able to execute Google Gadgets as well as other gadgets compatible to OpenSocial (http://www.opensocial.org/). The uService Player as well as the uService Editor are based on this container. The uService Player is optimized for usage on mobile devices.

Figure 8 shows a screenshot of the uService Player with the implemented and deployed rental service that was shown as a mock-up in Figure 2(c). The service is a composition of several interacting gadgets, i.e., (a) a gadget for displaying the name of the service, (b) a gadget for listing the items to rent, and (c) a gadget to show the rated price of the currently selected item.

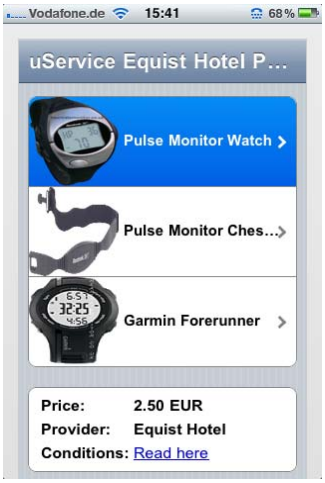


Figure 8: Screenshot of the uService Player with an Example uRun Service

In the following, more details about the functionality and the implementation of selected uService platform components are described. In particular, a method to remotely access third party devices (e.g., sensors) from mobile Web-based applications is presented in Section 4.1. The subsequent sections 4.2 to 4.4 then provide information about semantic search and recommendations for uServices, security and identity management issues, and accounting and billing functionality.

4.1 uService Player and Access to Device Capabilities

The uService platform inherently integrates OpenSocial. As a result, uServices are Web-based applications, which are in general composed of one or more gadgets. Each gadget can form a part of the user interface and is able to interact with other gadgets via a defined data exchange

mechanism. Given this innovative approach, users are able to easily compose their own individual applications on-the-fly as illustrated in Subsection 3.2.

However, by moving away from vendors' operating system APIs and programming languages towards an application development based on Web standards such as JavaScript and HTML, one main issue arises: These standards lack almost any support for accessing device capabilities. Speaking of smart phones, such capabilities are, for example, persistent storage, sound output and sensors such as integrated cameras, accelerometer-based step counters, GPS, or Bluetooth heart rate monitors. Recent efforts, which came together with HTML 5, such as the Geolocation API (http://www.w3.org/TR/geolocation-API/), try to fill this gap. It remains unclear, if and when this gap will be completely closed. Therefore, we propose a JavaScript-based solution to offer a sufficient API for Web-based applications. In contrast to the new extensions of HTML, the uService JavaScript library further allows to access both the own device's capabilities and the capabilities of a remote device. As a result, a uService may access a friend's GPS sensor the same way it would access its own local GPS. This way, an application to compete in running speeds could very easily be set up.

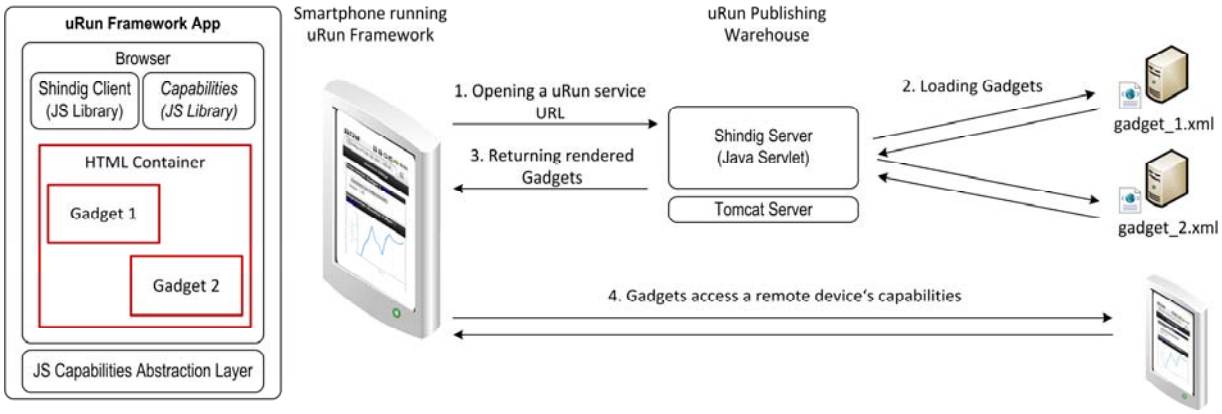


Figure 9: Data Flow in a uRun Service with Device Capability Access

Figure 9 illustrates which software components are required to run a uService with access to device capabilities. In general, only a browser is required to run a uService. If a uService needs access to device capabilities, the native uRun Framework App (uService Player) has to be installed. The uRun Framework App uses the OS's native API to provide a JavaScript capabilities abstraction layer. Thus, gadgets that load the uService capabilities library can request access to local and remote capabilities.

The basic data flow of a simple uRun service from loading the service until is executed during a run, is also depicted in Figure 9. After a user has created a uService, he can upload it to the uRun publishing warehouse. From there, it can be loaded via a common URL (1), which makes it easy to distribute it to friends. The OpenSocial server then resolves all gadgets the uService is referencing and loads the XML gadget description files from the individual hosting servers (2). After the Shindig server has received, parsed and validated all gadgets, the resulting Web page is send back to the client (3). The browser, which is integrated in the uRun Framework App, then displays the HTML user interface and loads required JavaScript libraries. Finally, the Web App runs in the browser and is able to access local and remote capabilities by calling corresponding

methods from the JavaScript capabilities library. Access to remote capabilities is accomplished by using REST-based Web Service interfaces. For slow mobile networks the uRun Framework App, does not only support HTTP, but also CoAP (https://datatracker.ietf.org/doc/draft-ietf-core-coap/), an upcoming Web transfer protocol for constrained networks.

4.2 Search & Recommendation Engine

The user-generated mobile services used in the uService platform can be very short-lived. Thus, there is a demand for a novel discovery approach. For a personalized discovery of uServices the uService platform uses different search and recommendation mechanisms.

Within the uService search the concept of “query extension” is used to expand the search query by adding user-related and contextual information (Pretschner and Gauch 1999). By this means, personalized and context-aware search results with a higher quality are achieved. uServices are also enhanced with semantic meta information which result in a semantic search engine. The semantic search engine utilizes ontologies and semantic annotations to associated uServices to the annotations and domain knowledge. The ontology defines vocabulary and logical relationships between concepts in a discourse domain which are also used for reasoning purposes. It is independent of the actual data but provides a common understanding of the semantics in the given domain. It is used for the explication of implicit and hidden knowledge in order to overcome the problem of semantic heterogeneity (von Wache, & Vögele, 2001). Our search mechanism for semantic annotations differs from conventional keyword-based engines. Different pre-processing mechanisms are used, e.g. based on the ontology structure or linguistic methods, and different information discovery mechanisms such as discovering new ontologies and annotations for indexing by crawling.

Recommendations are another type of information retrieval used in the uService platform. The recommendation problem is formulated as the problem of rating items that have not been seen by a user (Resnick et al., 1994). Whereas search queries are actively defined by the user, there are no explicit search terms for recommendations. However, there are user profile information and contextual data sent by the mobile devices. The uService platform follows the approach to compare new, unrated uServices with uServices that have already been judged and are similar to the new uService. This approach is also known as content-based recommendations (Salton, 1989). Another approach known as collaborative filtering is to ground recommendations on persons with similar preferences instead of considering similar uServices. The uService platform combines both approaches to overcome the cold-start problem, i.e. the fact that a critical mass of ratings has to be available for providing good recommendations.

In this context, personalization of recommendation becomes a crucial issue in our uService search and retrieval processes. Personalization means the incorporation of additional user-specific information that can be achieved for instance via incorporating context data. Context is defined as any information that can be used to characterize the current situation (e.g. time, place or users nearby) that are considered relevant for the interaction between a user and a uService (Dey, Abowd, & Salber, 2001). To further improve the recommendations by adding semantics to the context the uService platform combines context-aware recommender systems with ontologies.

4.3 Security & Identity Management

Security is ensured by a number of built-in functions, like authentication of all actors in the system, secure connections for communication, creation and verification of signatures, and secure management of profile data. These functions are provided by a dedicated uService Security and Identity Management (SIDM). As can be seen in Figure 7, the SIDM encompasses terminal-side and server-side functionality. On the terminal-side, the SIDM manages, e.g., user authentication based on credentials stored in the mobile's Universal Integrated Circuit Card (UICC). On the server-side, the SIDM manages the account data of the users. The other components of the uService platform, e.g., Accounting & Billing, gain access to the SIDM services through a secured connection. Communication is based on the REST protocol supported by the JSON data exchange format. An example of a RESTful request-response command for the retrieval of user profile data is given below.

```
Request:
HTTP/1.1 POST /sidm/profileaccess
Host: www.uservice.de
Accept: application/json
Content-Type: application/json
JSON content: {"eventType":"getUserProfileData", "userID":"max"}

Response:
Return code: HTTP/1.1 200
Content-Type: application/json
JSON content: {"userID":"max", "firstname":"Max", "lastname":"Mustermann",
"userPasswd":"*****", "street":"Musterstr. 123", "city":"Musterstadt",
"postalcode":"12345", "country":"Musterland",
"email":"muster@mail.com","phone":"123456789", "company":"Musterfirma",
"account":"654321"}
```

The server-side SIDM component provides persistent data handling for uServices as well as for basic identity management features like the creation, modification and deletion of user accounts, user groups and uServices such as uRun. On the terminal-side, all runtime data needed to run the uService platform is managed by the SIDM layer (see Figure 7). Moreover, existing social networks and community services are integrated, e.g. via the Facebook Connect API (http://developers.facebook.com/docs/guides/web/) or the OpenSocial API that was already mentioned above. These services enable uRun users for instance to manage the contacts to their running partners or to manage status messages. The secure and at the same time comfortable usage of different social networking platforms is supported by a terminal-side ID manager administrating the user's credentials.

Security features for authentication, verification and authorization include challenge-response protocols, secured connections and certificate handling, all realized based on well-known cryptographic primitives. On terminal-side, these primitives are supported by UICC functions. Moreover, the UICC is used for management of sensitive profile data.

4.4 Accounting & Billing

The Accounting & Charging component is an enabler for making business with the uService platform. It allows end users, third parties as well as operators to generate income with user-generated content and services. Different revenue sharing models can be established to form the basis for the business relationships and money flows (Bormann, Flake, Tacken, 2010).

Inside the Accounting & Charging component, an online charging and billing system (OCS) is located. This OCS is connected to the other platform components via two gateways. One gateway is responsible for real-time rating and charging operations and the other one for user and account management functions:

- Payment: This gateway offers rating and charging operations for rating and charging – either in a direct or in a session-based manner.
- User and Account Management: This gateway supports the registration of users to the OCS and the management of accounts, e.g. retrieving and updating current balances and retrieval of transaction histories.

Each gateway provides a Web Service interface that supports access via HTTP/REST based on the GSMA OneAPI standard (http://www.gsmworld.com/oneapi/). The OneAPI initiative defines a commonly supported set of lightweight and Web friendly APIs to allow mobile and other network operators to expose useful network information and capabilities to Web application developers. It aims at reducing the effort and time needed to create applications and content that is portable across mobile operators.

The gateways act as a server in a request/response manner. They do not perform any communication actions on their own. The requests are forwarded to the OCS. The responses from the OCS are then sent back via the gateways to the requesting application following the OneAPI standard.

The OCS is responsible for all rating and charging events originating from external charging entities. It holds all monetary and non-monetary information about uServices and their users. The OCS can also play an active role for certain situations, e.g. blacklisting potential harmful services or blocking advertisements if an advertiser has not deposited enough money.

5. CONCLUSION AND OUTLOOK

The approach described in this article is based on the vision that the next generation of mobile services is going to be created by end users. In the near future, users will become mobile ‘super prosumers’. That is, producers, providers, and consumers of mobile content and services. We summarized which key features a relevant platform needs to take into account and presented a prototype platform that addresses these features.

The proposed platform encompasses mechanisms for Web-based sensor integration, UICC-based security and ID management, semantic search and recommendation mechanisms, targeted advertisements, and a charging and billing system that allows for enhanced revenue sharing business models.

As an application area for the uService platform, we addressed the health and fitness domain, as such mobile applications provide an increased value for users; the ability to upload data and receive recommendations anywhere in combination with an activity like running offers the necessary flexibility to users to stay healthy and well. Gathering additional information about the current environment (e.g., a running track in the vicinity) supports them to pursue their individual fitness plans. We explained that accessing device capabilities from Web-based applications, such as in a mobile health application, is a major issue. Current Web standards lack sufficient support in this area. Therefore, we introduced a JavaScript-based solution that offers a common API to access device capabilities. For applications with social interactions, it allows to access both the own device’s capabilities and the capabilities of friend devices.

With a stable prototype system, we are going to perform a field operational trial in late 2011. The trial will take place in a realistic environment and with experienced runners to receive detailed feedback from end users.

ACKNOWLEDGMENTS

This work has been funded in part by the Federal Ministry of Education and Research (BMBF) under contract no. 01IS09020A-F (project uService) and is supported by the EUREKA cluster programme ITEA2 (http://www.itea2.org).

REFERENCES

3rd Generation Partnership Project. (2009a). Open Service Access (OSA), Parlay X Web Services, Part 7: Account Management (Release 8). Technical Specification TS 29.199-7 V8.1.0.

3rd Generation Partnership Project. (2009b). Open Service Access (OSA), Parlay X Web Services, Part 6: Payment (Release 8). Technical Specification TS 29.199-6 V8.1.0.

Ahamed, S.I., Haque, M.M., Stamm, K., & Khan, A.J. (2007). Wellness assistant: A virtual wellness assistant using pervasive computing. In *Proceedings of the 2007 ACM symposium on Applied computing (SAC'07), Seoul, Korea* (pp. 782–787). New York: ACM.

Ashbrook, D. (2010). *Mobile Health and Fitness*. Retrieved from http://rndbackyard.vodafone.com/2010/02/mobile-health-and-fitness/ (last visited August 14th, 2011).

Bertocci, V., Serack, G., & Baker, C. (2007). *Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities*. Addison-Wesley.

Bormann, F., Flake, S., Tacken, J. (2010). Obtaining Revenues from User Generated Mobile Services for Sport, Fitness and Health. In *Proceedings of Informatik 2010, Service Science – Neue Perspektiven für die Informatik, Workshop FitByIT – IT Dienste für Sport, Fitness und Gesundheit, Leipzig, Germany* (Lecture Notes in Informatics, Vol. P-175, pp 21-26).

Buttussi, F., Chittaro, L., & Nadalutti, D. (2006). Bringing mobile guides and fitness activities together: a solution based on an embodied virtual trainer. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile HCI 2006)* (pp. 29–36). New York: ACM.

Brown, J. & Rocher, G. (2009). *The Definitive Guide to Grails.* Second edition. Apress, ISBN 1590599950

Chapko, A., Emrich, A., Flake, S., Golatowski, F., Gräßle, M., Kösters, P., Prüter, S., Rust, C., Tacken, J., & Werth, D. (2010a). uRun – Der mobile Gesundheitscoach. In *Proceedings of Informatik 2010, Service Science – Neue Perspektiven für die Informatik, Workshop FitByIT – IT Dienste für Sport, Fitness und Gesundheit, Leipzig, Germany* (Lecture Notes in Informatics, Vol. P-175, pp. 9–14).

Chapko, A., Emrich, A., Flake, S., Golatowski, F., Prüter, S., Rust, C., & Tacken, J. (2010b). Towards a Platform for User-Generated Mobile Services. In *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), 5th International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 2010), Perth, Australia*. Los Alamitos: IEEE Computer Society Press.

Davies, M., Gil, G., Maknavicius, M., Narganes, M., Urdiales, D., & Zhdanova, A.V. (2008). m:Ciudad: An Infrastructure for Creation and Sharing of End User Generated Microservices. In *Proceedings of the Poster and Demonstration Track at the 1st Future Internet Symposium, Vienna, Austria*. CEUR Workshop Proceedings Vol. 399, ISSN 1613-0073.

Davis, D.A., Chawla, N.V., Blumm, N., Christakis, N., & Barabási, A.-L. (2008). Predicting Individual Disease Risk Based on Medical History. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08), Napa Valley, CA, USA* (pp. 69–778). New York: ACM. doi: 10.1145/1458082.1458185

Davis, D.A., Chawla, N.V., Christakis, N., & Barabási, A.-L. (2010). Time to CARE: a collaborative engine for practical disease prediction. *Data Mining and Knowledge Discovery*, 20(3), 388–415. doi:10.1007/s10618-009-0156-z

Dey, A. K., Abowd, G. D., Salber, D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications. In *Human Computer Interaction Journal*, 16, 97-166.

Flurry Analytics (2011). *Time Spent On Mobile Apps Has Surpassed Web Browsing*. Retrieved from http://techcrunch.com/2011/06/20/flurry-time-spent-on-mobile-apps-has-surpassed-web-browsing/ (last visited August 14th, 2011).

Folino, F. & Pizzuti, C. (2010). A Comorbidity-based Recommendation Engine for Disease Prediction. In *Proceedings of 23rd IEEE International Symposium on Computer-Based Medical Systems (CBMS 2010), Perth, Australia.*

Gaber, M., Krishnaswamy, S., & Zaslavsky, A. (2004) Ubiquitous data stream mining. In *Workshop Proceedings of Current Research and Future Directions, held in conjunction with the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Sydney, Australia.*

Guggenmos, C., Pérez, J., López-de-Ipiña, D., & Gil, G. (2010). A Hybrid and Semantic Location Management System for Mobile User Generated Geo Services. In *Toulouse Space Show - International Week on Space Applications, Toulouse, France.*

Haghighi, P.D., Zaslavsky, A., Krishnaswamy, S., & Gaber, M. (2009). Mobile Data Mining for Intelligent Healthcare Support. *In Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS '09), Big Island, HI, USA* (pp. 1–10). doi:10.1109/HICSS.2009.309

Haverbeke, M. (2011). *Eloquent JavaScript – A Modern Introduction to Programming*. First Edition. No Starch Press, ISBN 1593272820. Retrieved from http://eloquentjavascript.net/ (last visited August 14th, 2011)

Ho, T.C.T. & Xiang, C. (2009). ExerTrek: A portable handheld exercise monitoring, tracking and recommendation system, In *Proceedings of the 11th International Conference on e-Health Networking, Applications and Services (Healthcom 2009), Sydney, Australia* (pp. 84–88). doi: 10.1109/HEALTH.2009.5406194

Komninos, A. & Stamou, S. (2006). HealthPal: An Intelligent Personal Medical Assistant for Supporting the Self-Monitoring of Healthcare in the Ageing Society. In *Proceedings of UbiHealth 2006: The 4th International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications, held in conjunction with the 8th International Conference on Ubiquitous Computing (UbiComp 2006), Orange County, CA, USA.*

Kurdyukova, E. (2009). Interactive Community Notes to Support Mobile Runners. In *Mobile HCI 2009 Workshop on Context-Aware Mobile Media and Mobile Social Networks, Bonn, Germany*.

Kyriacou, E.C., Pattichis, M.S., Pattichis, C.S., Panayides, A., & Pitsillides, A. (2007). m-Health e-Emergency Systems: Current Status and Future Directions. *IEEE Antennas and Propagation Magazine*, 49(1), 216–231.

Lipprandt, M., Eichelberg, M., Thronicke, W., Krüger, J., Drüke, I., & Willemsen, D. (2009). OSAMI-D: A Domain-Specific Software Platform for Health Care Services. In *Proceedings of the 2nd International Conference on Human System Interaction (HSI 09), Catania, Italy* (pp. 139–145).

Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, L., Musolesi, M., Eisenman, S.B., Zheng, X., & Campbell, A.T. (2008). Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proceedings of the 6th International Conference on Embedded Networked Sensor Systems, (SenSys 2008), Raleigh, NC, USA* (pp. 337–350). New York: ACM.

Mueller, F., Thorogood, A., & O'Brien, S. (2007). Jogging over a Distance – Supporting a "Jogging Together" Experience Although Being Apart. In *Proceedings of the 2007 Conference on Human Factors in Computing Systems (CHI 2007), San Jose, CA, USA.* New York: ACM.

Open Mobile Alliance. (2011). *OMA Smartcard Web Server V1.0*. Retrieved from http://www.openmobilealliance.com/technical/release_program/scws_v1_0.aspx (last visited August 14th, 2011).

Oracle Corporation, USA. (2011). *Java Card Technology*. Retrieved from http://www.oracle.com/technetwork/java/javacard/overview/index.html (last visited August 14th, 2011).

Organization for the Advancement of Structured Information Standards. (2006). *Web Services Security: SOAP Message Security 1.1 (WS-Security)*, OASIS Standard Specification.

Pretschner, A. & Gauch, S. (1999). Ontology based personalized search. In *Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 391-398.

Richardson, L. & Ruby, S. (2007). *RESTful Web Services*. O'Reilly Media, ISBN 978-0-596-52926-0

Resnick, P., Iakovou, N., Sushak, M., Bergstrom, P., Riedl, J. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of the ACM 1994 Computer Supported Cooperative Work Conference, Chapel Hill, NC, USA, 175-186.

Roelands, M., Godon, M., Feki, M.A., Claeys, L., Zontrop, P., Criel, J., De Voegt, K., Geerts, M., & Trappeniers, L. (2010). Research Orientation towards Do-it-Yourself Internet-of-Things Mass Creativity Concepts. In *Workshop 'What can the Internet of Things do for the citizen (CIoT)?' at the 8th International Conference on Pervasive Computing, Helsinki, Finland*.

Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley.

Steinhaeuser, K. & Chawla, N.V. (2009). A Network-Based Approach to Understanding and Predicting Diseases. In *Proceedings of the 2nd International Workshop on Social Computing, Behavioral Modeling, and Prediction, Phoenix, AZ, USA* (pp. 209–216). New York: Springer.

Verna, P. (2009). *A Spotlight on UGC Participants*. Retrieved from http://www.emarketer.com/Article.aspx?R=1006914 (last visited August 14th, 2011).

Villarreal, V., Laguna, J., López, S., Fontecha, J., Fuentes, C., Hervás, R., López de Ipiña, D., & Bravo, J. (2009). A Proposal for Mobile Diabetes Self-control: Towards a Patient Monitoring Framework. In *Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN 2009), Salamanca, Spain* (LNCS 5518, pp. 870–877). New York: Springer.

Wache, V. U. H. & Vögele, T. (2001). Ontology-Based Integration of Information - A Survey of Existing Approaches. In *Proceedings of IJCAI 2001 Workshop Ontologies and Information Sharing*, Seattle, WA, USA.