

SOA in Real Life

Mobile Solutions

Any place, any time: the old promise from the dotcom age has never been more relevant. With the release of the iPhone, Apple set off a huge amount of hype. Many people now have a laptop with broadband Internet connection and/or WLAN or UMTS Internet access. Yet these devices are still too large, too awkward, and take too long to boot up to be usable at any time. On the other hand, almost everyone has a smartphone these days, making them more mobile than ever in today's economy.

Smartphones are enormously practical and are becoming more and more powerful. They are generally very easy to operate, can be used almost anywhere, and the mobile web is becoming both faster and cheaper. App stores are shooting up everywhere and new functions can be installed with a single click. As the saying has it: "There's an app for everything."

The use of built-in sensors provides for entirely new possibilities such as Google Maps integration, location-based services, augmented reality, etc. Built-in cameras are becoming more and more powerful and are often used as a second compact camera. Video telephony is becoming more common—not just on Skype, now long-established, but also through Apple Facetime. The speed of innovation is tremendous.

Use Cases

A very high percentage of apps are games, followed by information systems that are mainly of interest to private users. These information systems are making increased use of the built-in functions on the mobile device. For example, the system identifies my location via GPS and can provide me with information via a personalized localization. Using the integrated camera I can scan a barcode and run a price comparison through the system. Previously unthinkable "Star Trek" technology is now (almost) a reality. Soon we will have combined tricorders/communicators/tablets in a single device. More and more companies are viewing mobile solutions as a means by which to accelerate processes, incorporate external partners more easily into processes, and lower their own process costs. We are already using numerous examples for B2C services such as booking flights by cell phone, tracking packages, and finding out delivery dates and opening hours. To optimize these functions for your own company, creative ideas are required: How can I increase customer loyalty? While B2C applications are already spreading very rapidly and probably represent the core business of all non-game apps, B2B is developing only slowly. One of the central questions here is how additional services can be offered to the business partner. According to Gartner, top growth areas include location-based services, social networks, mobile search, mobile commerce, mobile cash, context-aware services, object recognition, mobile instant messaging, mobile e-mail, and mobile video.

Approaches to Development

Possible architectures for the development of mobile apps vary greatly. There are very different approaches for bringing mobile solutions onto cell phones. One approach uses the option to run a browser on the mobile device. A mobile app is a web application optimized for the mobile device. For enterprise software developers, this is initially the easiest method. Control remains with the central application, there is no need to download or update software on the mobile device, and, perhaps most importantly, apps continue to be developed in the same way as before. We are dealing here with the same "good old web development," just for smaller screens. Alternatively, native

applications can be created directly on the relevant target platform. Since this requires individual components to be developed for multiple platforms, costs increase. For this reason a number of interim solutions are already beginning to arise such as interpretative runtime environments, cross compilers, or MDA approaches, which take on this problem and generate applications from a uniform code base for diverse target platforms (PhoneGap, Titanium etc.). We cannot say yet which trend will win out. The app concept is currently growing fast, e.g. by being anchored in the Mac operating system, in Windows 8, or via a wide range of new app marketplaces (Amazon, etc.). By contrast the benefits of web-based solutions are apparent on "normal" interfaces that can be developed rapidly. They do not need to be deployed, nor does a distribution platform or similar need to be selected. Whether or not a general decision is now made in favor of Web-based solutions or for native apps will depend, as always, on specific requirements. Should the solutions be sold or merely offered? Are subscription models necessary? In this regard Android is generally more open than Apple, which is also the reason why the growth rates for Android are so high. The drivers here are new use cases that today are still not quite conceivable. Imagine for example the following scenario after a car accident. The victim starts up the insurance app on their smartphone, through which the location of the accident can be immediately located by GPS, the insurer could be immediately sent photos of the damage and, using the speech memo, a description of the accident recorded. The insurance company would then have all of the documents required to deal with the claim. How can this be developed as generically as possible for different smartphone platforms? The different options must be evaluated and the best one implemented for the customer.

Programming Languages

Almost every platform has its own programming language or a development environment with proprietary rights. But what are the most common approaches?

With Objective-C on iOS, Apple has its own language, as does Microsoft with .NET on Windows Phone 8.x, though the future is not clear here either. If market analyses are to be believed, in purely numerical terms the Android open-source operating system is currently enjoying the greatest growth (how this will be reflected in the business field remains largely unclear). Symbian continues to enjoy a major market share with feature phones, though they are becoming less and less important in our field, and RIM (Blackberry) still has a stronghold among smartphones; nonetheless, both are declining strongly. Google, one of the key players, started the Java-based Android platform and continues to play a key role in its development.

Almost all business-relevant mobile applications require some type of back-end to provide the services offered. Here, Java is extremely widespread and can fully exploit its strengths, e.g. in the provision of new end points for existing components or the implementation of adapter layers. Java also offers extremely good interoperability which makes it very popular in this environment. In addition, Java is used as a client-side programming language for the Android platform. By contrast JavaME has failed to establish itself, even though it is very widespread on millions of feature phones. And Micro Edition will probably fail to make the decisive step forward in the future. Java application development is also possible on the Blackberry platform and is also supported by Oracle in the form of ADF Mobile for various mobile devices (completely new release in 2012). The JavaFX Mobile initiated by SUN has been declared dead, but is trying its luck again with Oracle as JavaFX 2. The new HTML5 is a quantum leap forward in mobile development. This is the latest version of HTML with numerous enhancements, e.g. access to specific device features such as camera, microphone, compass, motion sensors, GPS functionality, etc., including use of JavaScript

and CSS. Thanks to this technology, cross-platform solutions can already be implemented today.

History: The first iPhone apps were exclusively web apps, which is why mobile browsers continue to offer a high level of comfort today. The closed app concept only began to be applied shortly thereafter.

The Renaissance of JavaScript

JavaScript is currently enjoying a renaissance. JavaScript used to be looked down upon because it was used mainly to "pretty up" websites and was not seen as a "real" language for professional application development. However, this trend has changed due to the enormous speed at which HTML5 has expanded. JavaScript is well on its way to becoming the language for application development and interface programming. Indeed it can be said that JavaScript has the same significance for web development as Java for Enterprise Computing in the back-end. For a programming language that is now 15 years old, JavaScript has many modern characteristics. Provided it is implemented correctly, it has no trouble competing with modern languages such as Groovy or Ruby. As a dynamically typed, object-oriented scripting language with prototype-based inheritance, it has a number of unusual attributes for the developer used to Java. For this reason, the developer may need time to get used to it and to learn how to write idiomatically good JavaScript. The sooner a developer comes to grips with it, the better. On servers, too, JavaScript is becoming more common in runtime environments like Node.js and is positioning itself as a very valid solution for certain problem categories such as applications that must support a large number of competing accesses to "slow resources." There are more and more tools that hide the actual JavaScript code from the developer, such as e.g. Google Web Toolkit, CoffeeScript, ClojureScript, or language approaches such as Google Dart. However, sound knowledge of the basic principles is still essential.

Mobile Solutions and SOA

In an Enterprise context, mobile solutions, whether they are apps or mainly web-based, almost always have to communicate with a back-end. In simple terms, mobile front-ends only offer a further output channel for applications which already exist anyway. Although true in principle, in practice the picture is more complicated. Due to the modified framework conditions such as a small, medium, or large screen; touchscreen, stylus, or keyboard operation; and special restrictions such as server communication from JavaScript contexts, existing interfaces cannot be transferred 1:1 to mobile end devices. In general, changes to service end points can be required particularly for new functionalities. All SOA principles now apply here including loose coupling, compositions, facades, etc., so that service-oriented thinking in the back-end comes fully into its own. New requirements can be implemented flexibly and quickly without major changes to the back-end system.

What do mobile solutions change in our architectures? Not much, at least not directly. Technically, REST communication and the exchange of JSON structures appear to suit the mobile environment better than SOAP services and XSDs. The enormous increase in mobile use cases means that investments in a company's own service-oriented architecture will make its benefits felt far more quickly. This surprisingly innovative environment leads to the realization of a huge number of new, business-relevant application cases, at a speed that would have been unthinkable just a few years ago. The flexibility of an SOA landscape will be paid back in full here, because it will make it possible to implement new opportunities very quickly. For example, an ESB is very easy to use as a protocol converter if an existing back-end "only" speaks SOAP but the mobile client needs to be connected via REST.

Single-Page Web Apps

What will change radically is the actual architecture of the web apps. There is a general move away from the usual pattern of classic web applications (**Fig. 1**), which are typified by the way in which page navigation is generally implemented using an MVC pattern that locates the PageflowController on the server. Interaction with a website triggers a request that is sent to the web server. Here, the PageflowController determines what page to display next and the server returns this page as a response to the browser. By contrast, with web apps the client is more intelligent. Generally the location of the PageflowController is shifted to the client, usually realized in JavaScript. Here, it is usual to speak of single-page web apps (**Fig. 2**). The name originates from the fact that when an HTML page is loaded, the entire web app is loaded at the same time, i.e. a single HTML page, of which only a part is displayed. If a different page is to be displayed, the PageflowController only shows a different part of the page already loaded on the client side. This gets around the request response cycle to the server and back. This cycle only occurs when services are accessed that are contacted through SOAP or REST, in order to communicate with a back-end.

mobile_1.tif

Fig. 1: Architecture of a classic web application

mobile_2.tif

Fig. 2: Architecture of a single-page web app

Mobile Solutions and BPM

Mobile solutions are a major driving force behind process automation solutions. In this environment there are no new technical options for mobile use. But a wide variety of options opens up merely by enabling processes to receive input from people on the move, resulting in much faster throughput times. The advantages range from preventing the need to take notes on paper to simply being able to approve something while sitting through the coming attractions at a movie theater. (This article does not discuss the influence of this technology on the private lives of these persons.) The possibilities are very wide-ranging, especially since the mobile devices' new sensors can be used: geolocation, card-based information, cameras, barcode scanners, or movement profiles, all of which improve process support. The ideas list in the "Possible Uses of Mobile Solutions" box provides a long but far from exhaustive list of use cases. With regard to BPM, understood here as process automation, the following options arise concerning use of mobile end devices.

- Start processes: There is no need to postpone the start of a business process until you get somewhere with desktop Internet access. Instead it can take place on the spot. As the above example concerning the car accident showed, this saves time but also provides opportunities for better data quality and accurate, up-to-date information.
- Mobile task list: Modern BPM systems involve people via mail inboxes or task lists. These can of course be made available on a mobile basis.
- Mobile process images: If an employee is frequently incorporated into more complex processes, it can be helpful to provide them with a clearer picture of what point or what process is affected by the decision they must take. For this purpose, mobile end devices can display process images including visualization of current status.
- Alarms: In emergency situations, systems that normally run independently require the attention of an expert. Here, alarms can be sent to mobile devices that may already have preselected

decision-relevant information and which therefore permit rapid access to business, process or system sequences.

- Dashboards or mobile BI solutions: The ability to access ongoing business on mobile devices, in particular tablet computers, is becoming increasingly popular. In addition to the display of "normal" reports optimized for touch displays, use cases are in particular business-specific query options. For example, a store manager who scans an item on their shelf would immediately receive the price and brand development for this item in their own store, including a comparison with the prices throughout the entire store chain.

In all these use cases, the aim at the highest level is to increase business efficiency by integrating the user into process flows earlier, more rapidly, or simply better. "Mobile BPM" does not however mean using lightweight process engines on the mobile devices themselves.

The Hell of Maintaining a "Lots of Cheap Apps" Architecture

In his blog, Max J. Pucher, the mastermind behind the BPM discipline of adaptive processes, discusses a potentially problematic development in this connection due to the increased use of mobile applications. Pucher's train of thought was triggered by a lucky entrepreneur who managed to buy a great app developed for very little money. Nothing wrong with being a lucky entrepreneur. But what happens if, due to this one person's success, more and more "small" apps are developed, all of which are cheap and fast? In the medium term, joy might turn to sorrow once we start thinking about how to maintain this type of application landscape. Due to the UIs and the high number of small apps, there is a risk that our flexible SOA back-ends that were once loosely coupled would have to be tightly connected again in the absence of a suitable provision. Modifications would then be possible only very slowly or not at all, as it might necessitate ten different service providers or freelancers to make changes to their apps all at the same time in order to accommodate a process change. This state of affairs, which we will call the hell of maintaining a "Lots of Cheap Apps" architecture, should be avoided. As described above, sensibly applied SOA principles represent sensible countermeasures.

Summary

On the front-end side, the technological future certainly belongs to HTML5, where we are witnessing a lot of new developments and innovation. New architectural patterns such as the single-page web app must be added to the architectural toolbox. However, there would be no point in coining the new term "mobile BPM" since there are no technical differences compared to other output channels to justify doing so. There are however a huge number of use cases that would benefit from a mobile implementation such that consideration of mobile solutions appears to make particular sense in the context of SOA and BPM. For BPM in particular, mobile use cases could become real drivers for development since smaller projects can supply a faster ROI on presumably expensive and complex central process engines. The mobile solutions environment therefore represents a genuine opportunity for IT to move away from being a cost center and towards being a value-creating business enabler, creating new business opportunities via technological innovation.

Use Cases for Mobile Solutions

Practical use cases for mobile solutions which, taken in isolation, produce a ROI quickly, are extremely diverse. The objective is always to exploit the possible new benefits of being mobile in order to optimize existing processes or create new ones. Here we must search not just in the core

value creation processes: the point is to identify those "leverage points" in all company processes that promise disproportionate benefits due to mobile support.

- *Inventory*: elimination of paper and avoidance of media breaks, better data quality
- *Service Level Agreements*: mobile check whether goods are presented as agreed, compliance with features of relevant SLAs (cleanliness, punctuality, etc.), compliance with contracts of any kind that are linked to the SLA
- *Sales & Distribution*: access to sales material, catalogs, electronic forms, etc., often in visually attractive contexts
- *Warehouse*: immediate information on containers, pallets, boxes, packages, etc.
- *Building management*: self-service applications that switch lights on outside of official office hours, report defective lamps, toilet flushes, general damage to inventory; information systems for climate and room control
- *E-Government*: public authority finder, mobile data entry for regulatory agencies, fault reporters (e. g. dirt in public areas), mobile police stations, occupancy of parking spaces and parking garages, personalized vehicle license plates, garbage alarms for collection dates, suspended bus services or school closures due to weather; objectives: addressing the target group appropriately, more efficient administration, crowdsourcing (citizen as the provider of input), proximity to the citizen
- *Logistics*: support of returns using barcode scanner and GPS information for specialist service providers
- *Tourism*: access to onboard, cabin, and shopping systems on cruise ships; tour guides
- *Transport*: mobile train/bus schedules and tickets
- *Work Planning*: flexible update options for the avoidance of large volumes of quickly outdated documentation and rigid checklists
- *Marketing*: use of augmented reality with supplementary real-time information for complex processes, e.g. on advertising that increases sales
- *Smart Home*: control of your own home regarding energy consumption, switching on energy-intensive consumers, remote control of heating, oven, lighting, alarm system, etc.
- *Health*: ability to connect new sensors such as those for blood pressure, diabetes, etc. to mobile end devices to be able to offer tricorder-like functionalities, with back-end services

BOX End