Introduction to a mobile development methods investigation to Scania IT AB
– Case: Rundtursbuss application

Av: Juliana Moreira
Handledare: Petri Lankoski
Summary

The purpose of this work was both to explore the variety of methods to develop mobile applications and to create a requirement documentation and solution proposal to the case study Rundtursbuss. During the 10 weeks work-experience, I had the possibility to investigate a range of methods to develop mobile applications and apply it to a reference case. The questions regarding which method to choose when developing mobile application are not easy to answer, once there is not only one solution that can fit the needs of all the applications that will be developed at Scania. Each application needs to be evaluated not only from a technical perspective, but also from a user-centered way. One of the best ways to quickly evaluate a project is asking what that makes the project mobile The analysis of each criteria aided in the evaluation of the application requirements and in reaching the conclusion that the most promising methods to develop the application Rundtursbuss are either native or cross-compilation ones. In conclusion, it is crucial to emphasize that while Scania does not have a procedure on how to develop, maintain and coordinate the development of mobile applications, single initiatives will continue to emerge, which increase the cost and affect the quality of their IT products and services.

Keywords

Mobile development, mobile development methods, mobile development investigation, Scania, native, mobile web application, the effect map, PACT, mockup, IOAW, Rundtursbuss
Contents

Summary ......................................................................................................................... 1
    Keywords .................................................................................................................. 1
1. Introduction ........................................................................................................... 6
    1.1. Thesis Overview ........................................................................................... 7
2. Focus and result .................................................................................................... 8
    2.1. Project result ............................................................................................... 8
3. Definition of terms .............................................................................................. 9
4. Background .......................................................................................................... 11
    4.1. Scania CV AB ............................................................................................ 11
    4.2. Scania IT AB (InfoMate) ........................................................................... 12
    4.3. Mobile Development .................................................................................. 12
    4.4. Mobile Development at Scania .................................................................. 13
        4.4.1. Software ............................................................................................ 14
        4.4.2. Hardware ......................................................................................... 14
5. Part One – Evaluation Criteria ......................................................................... 15
    5.1. User Experience (UX) ................................................................................ 15
    5.2. Native Experience (NX) ............................................................................. 16
        5.2.1. Native integration ............................................................................. 16
        5.2.2. Native capabilities ........................................................................... 16
        5.2.3. Native controls ............................................................................... 17
    5.3. Time to Market ............................................................................................. 18
    5.4. Portability ....................................................................................................... 18
    5.5. Management and Maintenance .................................................................. 19
    5.6. Discoverability ............................................................................................. 19
    5.7. Distribution .................................................................................................... 19
    5.8. Required Competences ............................................................................... 19
6. Different methods to develop mobile applications ......................................... 20
    6.1. Web ................................................................................................................ 20
        6.1.1. Responsive Design ........................................................................... 20
    6.1.1. MWAs – Mobile Web Applications ..................................................... 25
7. Cross-platform frameworks .......................................................................... 34
    7.1 Cross-Compilation ....................................................................................... 34
        7.1.1. Behind the scenes ............................................................................. 35
    7.2 Hybrid ............................................................................................................. 39
        7.2.1. Toolkit ............................................................................................... 40
    7.3 Native ............................................................................................................... 44
        7.3.1. Native Application ............................................................................ 44
8 Conclusion Mobile Development Methods ........................................... 50
Developing applications at Scania ......................................................... 50
  8.1.1 Web .................................................................................................. 52
  8.1.2 Cross-Platform and Native ............................................................. 52
9 Part Two – Case: Rundtursbuss application ......................................... 54
  9.1 Internal buses at Scania ................................................................. 54
      9.1.1 Design ........................................................................................ 54
      9.1.2 Sketches vs. Prototypes ............................................................. 55
      9.1.3 Personas & Scenarios ................................................................. 56
  9.2 Methods ............................................................................................ 57
      9.2.1 Requirements ............................................................................. 57
      9.2.2 PACT .......................................................................................... 58
      9.2.3 The effect map ........................................................................... 59
10 Conclusion and Recommendations ...................................................... 63
11 Acknowledgment .................................................................................. 67
12 Bibliography ......................................................................................... 68
13 Attachments .......................................................................................... 71
    Attachment 1 – Mockup ....................................................................... 71
    Attachment 2 – Road Map Working Group Device API .......................... 73
    Attachment 3 – Requirements ............................................................... 74
    Attachment 4 – Personas ..................................................................... 75
      13.1.1 Tor Svanström, ........................................................................... 75
      13.1.2 Karl Jonsson .............................................................................. 75
      13.1.3 Helena Andersson ................................................................... 76
    Attachment 5 – Scenarios .................................................................... 77
      13.1.4 Tor Svanström, ........................................................................... 77
      13.1.5 Karl Jonsson .............................................................................. 77
      13.1.6 Helena Andersson ................................................................... 78
Figures

Figure 1 - Scania’s presence around the world ................................. 11
Figure 2 - Innovation online’s responsive design on three different screen sizes ........................................... 21
Figure 3 - Obama’s website tries to mimic the behaviour of a native application with a different way to present the context menu. This responsive design pattern is called “off canvas” (Wroblewski, 2012) .................. 22
Figure 4 - Web Application built with the HTML, CSS and JavaScript techniques ............................................ 25
Figure 5 - HTML5 features that most browsers support ........................................................... 26
Figure 6 - CSS rules and browser’s compatibilities ........................................................... 29
Figure 7 - The same application built to different platforms: android (left) and iOS (right) .............. 35
Figure 8 – Titanium’s architecture. Titanium works as a bridge between your code and the native operative system ............................................. 36
Figure 9 - Wikipedia’s hybrid applications to Android (left) and iOS (right) .............................................. 39
Figure 10 - Wikipedia’s hybrid applications to Android (left) and iOS (right) ......................................... 40
Figure 11 - The PhoneGap:Build process ........................................................... 40
Figure 12 - PhoneGap supported features on native platforms .............................................. 41
Figure 13 - Buxton explains that a prototype is a continuation phase of a sketch. Contrast in purpose and form is the main difference between the methods (2007, p. 140) .................................................. 55
Tables

Table 1 - Variations of hardware elements that organizations must take into consideration when developing mobile applications .......................................................... 17
Table 2 - Responsive design’s evaluation criteria ................................................................ 23
Table 3 - Mobile Web Application’s evaluation criteria .......................................................... 31
Table 4 – Cross-compilation’s evaluation criteria .................................................................. 37
Table 5 - Hybrid application’s evaluation criteria ................................................................. 42
Table 6 - Native application’s evaluation criteria ................................................................. 45
Table 7 - Native iOS registration and distribution of in-house applications ............................... 48
Table 8 - Google Play registration and distribution requirements .......................................... 49
Table 9 - Windows 8 registration and distribution requirements ........................................... 49
Table 10 - An overview of strengths and weakness of each method in relation to the evaluation criteria . 51
Table 11 - Considerations about the recommendations .......................................................... 64
1. Introduction

We live in times of rapid and drastic changes in which new forms of interaction emerge all the time and demand that organizations become more accessible to stakeholders. Nowadays, organizations are more aware of the necessity of making their content and mobile applications accessible to partners, shareholders and workers in order to increase productivity and customer satisfaction. While CIOs look at costs, risks, business goals and mobile user segmentation as strategy, IT managers direct their attention to, among other things, enterprise architecture and technology evolution (Redman & DeBiasi, 2011).

Given the complexity of this question, organizations need a detailed and substantial mobile strategy to tackle it. Simultaneously, designers and programmers have to rethink the methods they use to produce applications in order to meet the requirements of the amount of devices that has emerged. Not to mention the diversity of activities and contexts in which people use these interactive systems.

The will to explore the possibilities and challenges of mobile ecosystems led me to want to explore mobile situation within Scania IT AB.

The more the technology develops the more enterprises invest in new mobile devices to complement their existing web based systems. The solution I will present here can be useful not only for this specific project and users of the service but also as an input to develop a mobile development strategy.
1.1. Thesis Overview

• **Chapter two** describes the scope and the deliveries of the thesis.

• **Chapter three** provides a definition of the terms used throughout the investigation.

• **Chapter four** offers brief background information on mobile development, some facts about Scania CV AB, Scania InfoMate and the object of the case study.

• **Chapter five** describes the evaluation criteria the organization should take into consideration when choosing a mobile development method.

• **Chapter six** gives an overview of the different methods to develop mobile application. This chapter also suggests when it is appropriate to use different methods for new solutions.

• **Chapter seven** provides a summary about mobile development methods and its relation with Scania.

• **Chapter eight** offers a description of the design framework used during the conceptualization of the application.

• **Chapter eight** introduces the case study’s requirements and applies the methods described on the previous chapter.

• **Chapter nine** presents the technical and design recommendations based on analysis of the evaluation criteria and the application’s requirement.

• **Chapter ten** provides the conclusion and the recommendations of the thesis.
2. Focus and result

In this study, I will focus on mapping architecture, technology and standards for developing mobile applications to Scania CV AB. For instance, the project *Rundtursbuss* mobile solution, an application whose aim is to provide Scania’s workers with bus schedule, route and destination around Scania’s complex in Södertälje, will be used as a case. Furthermore, I will develop requirement specification document for this practical case.

In order to reach the objective of the project, it is necessary to briefly map out and investigate which existing methods fulfill the requirements of the project concerning items such as functionality, development, distribution, maintenance and required competences. Furthermore, it is necessary to have a clear image of user’s needs, why the application will be developed and its desired effects.

My individual objective is to acquire and widen the existing knowledge about mobile development at the same time as to learn how a large company deals with such a question and the challenges it brings about. In addition, I want to increase my comprehension of interaction design process and IT management.

2.1. Project result

The result of this study is divided in two parts:

- An investigation about mobile development methods.
- A mockup of an application in a documented requirement specification where design, and functions merge together.
### 3. Definition of terms

**Android** – Operative System from Google Inc. that runs in mobile devices from different manufacturers. The manufacturer who wishes to use Android into its devices must make sure that the device is compatible to the operative system requirements.

**Android SDK** – Android Software Development Kit. Android SDK is the environment where developers can compile, test and simulate the application.

**API** – “Application Programming Interfaces is a software interface that exposes access to some internal functionality of a piece of software for use by programmers, to get access to specific information, to trigger special behavior, or to perform some other action” (World Wide Web Consortium (W3C), 2010a)

**HSL(A)** – Combination of hue, saturation and lightness (alpha)

**IDE** – Integrated development environment is a software application that often provides a compiler and an interpreter and facilitates software development.

**iOS** – Operative System from Apple Inc. that runs into iPhone, iPad, iPod devices.

**iOS SDK** – iOS Software Development Kit. iOS SDK is the environment where developers can compile, test and simulate the application.

**Markup** - A detailed instruction that indicates the format and structure for an electronic document

**Markup Language** - A set of standards used to create a structure for an electronic document

**Native applications** - The email client, SMS program, calendar, maps, browser, contacts.

**Native code** - a code that is compiled, which is faster.

**NX – Native Experience** - Knoernschild (2011) explains that NX is the ensemble of characteristics that an application needs to have in order to provide the unique properties of the platform that the application is running on. He points out that NX is only one of many aspects of the total UX.
**PACT** – The acronym stands for People, Activity, Context and Technology. PACT is a design framework for thinking about a design situation when creating interactive systems.

**Patterns** – well-defined, well-researched best practices where the principles of the design and its correct application contribute and integrate a solution in a whole system. (Hoober & Berkman, 2012)

**Provisioning Profile** – Provisioning profile associates a developer certificates and a device with an application identifier. It means that the provisioning profile provides basic authorization for an application to run.

**RGB(A)** – Combination of red, green, blue, (alpha)

**RGB(W)** – Red, green, blue, (white)

**RMA** – Resident Mobile application

**SDK** – Software Development Kit

**User Interface** – the interface where the interaction between humans and machines occurs.

**UX** – User experience can be defined as a non-utilitarian aspect of interaction in which the focus alternates between the users affections, sensations, meaning and value of interactions on their routine, even though such a definition still lacks an overall consensus (Law, Roto, Hassenzahl, Vermeeren, & Kort, 2009)

**WMA** – Web mobile application

**W3C** – World Wide Web Consortium
4. Background

In this chapter, I will give a short overview about the organization I am writing this work to. As I mentioned before, this thesis will include a practical part where I will create and present a mockup of an application called *Rundtursbuss* (see attachment 1). In order to contextualize where the practical part of this work will be applied, it is necessary to provide some information about the bus system, Scania CV and its subsidiary Scania IT, where I have worked with this thesis during 10 weeks.

4.1. Scania CV AB

Scania was founded in 1891 and today is one of the world’s leading manufacturers of heavy trucks and buses. Scania operates in about 100 countries, owns 11 factories in five countries and has more than 34000 employees (Scania CV AB, 2011). The figure below gives a panorama over the presence of the Scania around the world.

![Scania's presence around the world](image)

*Figure 1 - Scania's presence around the world*
4.2. Scania IT AB (InfoMate)

Created in November 2001, Scania InfoMate became Scania’s wholly owned IT subsidiary. According to Scania InfoMate’s history, long before the actual company has been formed, Scania had transformed their IT department, Administration Technique, in Scania Data AB¹.

Mainly conducted in Sweden, France, the Netherlands and Latin America, InfoMate has as mission to provide IT products and services in order to support Scania’s business needs worldwide. With almost 700 employees and 250 consultants, InfoMate provides among other things development, maintenance and management of IT systems, different levels of support throughout the system’s lifetime and installation of IT equipment.

4.3. Mobile Development

Mobile applications have always existed in mobile telephones. Native applications pre-installed on phones during the manufacturing were the first ones we heard about. However, app, a reduction of the word application, has become more and more common in today’s vocabulary. It is due mainly to the popularity of Apple’s smart phone and its operative system, launched in 2007, as well as the opening of the first digital application distribution platform, App Store, back in 2008.

“iPhone is a revolutionary and magical product that is literally five years ahead of any other mobile phone. We are all born with the ultimate pointing device—our fingers—and iPhone uses them to create the most revolutionary user interface since the mouse.” (Apple Inc., 2007)

This was only the beginning of the revolution that would happen in the mobile market.

Nokia presented Symbian and Blackberry introduced its operative system RIM, which has a large adoption in USA. Google has also launched its operative system, Android, which has been

¹ Scania’s IT department was created 1986 and became Scania Data AB 1990.
adopted by a significant number of mobile manufactures, not to mention Microsoft which came with its Windows Mobile, and recently with its Windows 8\(^2\).

In a short period of time, the status of mobile phones changed from a device you can use to make telephone calls with to portable computers, which enables a large array of possibilities such as Internet access. This transformation carries not only a significant change in hardware but also in software and web development.

Today, we are not only living in a battle of device manufacturers but also in a war of giants between Google and Apple. The companies are behind the most spread operative systems on mobiles devices, Android and iOS. Not to mention the World Wide Web Consortium which works to deliver capable web standards that support the increasing demand for mobility.

### 4.4. Mobile Development at Scania

Scania has always followed the newest technologies available on the market. In the 1960, the organization bought its first computer and 13 years later installed the first data terminals. Concerning mobile technologies, the company had already in 2005 adopted Windows Mobile and introduced wireless synchronization of data with Microsoft Exchange Server 2003, in which could be seen as a step forward to a mobile solution within the organization.

Although Scania has for a long time had shy and occasional initiatives in this area, there is no specific group or unity in charge of the mobile issue. According to stakeholders I have interviewed during my work with this thesis, there is a large demand for mobile applications that is not being supplied because the organization is not prepared to respond to it.

The department of Brands & Marketing (KM) took a step forward on this question and is developing a Mobile Application guideline, which follows the internal identity manual and the Brand Communication Platform (BCP). The ambition of this guideline is to assure a consistent brand presentation throughout all Scania’s communication channels.

---

\(^2\) Windows 8 release is expected to between Q3 2012 and early 2013 (Rama, 2012)
4.4.1. Software

Until present time, there are occasional mobile application initiatives within Scania beginning to take form. Scania Dealer Locator, a native application that finds the Scania dealers and workshop closest to your current location, is already available on App Store since December 2011. This application was not developed in house, but by an external provider who also is taking care of the management.

As a result of the increasing interest for mobile applications, various groups within the organization have already been planning several other applications.

4.4.2. Hardware

Today, the organization has divided the large number of devices in the market in three main segments, basic, standard and premium, where price and properties such as synchronization and usability are determinants. The three most common mobile phones to have access to Scania’s network are Samsung Galaxy Ace, HTC Desire S and Apple iPhone, of which the first one is the one the company offers largest support for. At the time of writing, there are 5000 devices around the world that use Scania’s synchronization service.

Another recurrent question within the company is how to introduce tablets to the employees’ every-day routine. Tablet is a second category of mobile devices that is gaining strong popularity within Scania. The lack of a solid wireless infrastructure that facilitates the handling of this kind of device in large scale can be considered as one of the big obstacles to an early adoption.

An important factor towards a mobile strategy within the organization is the selection of a tool that will facilitate the management of mobile devices. Scania is still studying the adoption of a Mobile Device Management (MDM) tool, which can be useful to securely enroll devices, configure and update settings via wireless, lock and wipe devices and manage large-scale deployments.
5. Part One – Evaluation Criteria

To develop mobile applications has become a high priority task to organizations and programmers. It implies a considerable amount of decisions companies have to make. Therefore, the decision to develop any type of mobile applications should be based on business requirements and objectives. Business requirements describe the business outcome for the application and how the application can support these objectives. (Knoernschild, 2012b).

On this chapter, I describe eight aspects Scania should consider when developing mobile application. These aspects bring to light important characteristics that, if examined properly, can be used as an advantageous tool before choice of a mobile solution.

5.1. User Experience (UX)

Goldkuhl and Röstliger (1988) call attention to a common problem within IT projects: flawed user focus due to the fact that enterprises often lay their attention to the choice of the technology and therefore forget the users. In turn, users have developed a strong relation with their device allowing users to create high expectations about UX. (Olson et al., 2012; Knoernschild, 2011a).

The context – The environment the application is used —which can be multiple ones—hardware affordance, platform capabilities and UI conventions are elements that cannot be changed or controlled. They must be understood. Environment is nevertheless considered as an unpredictable factor because expectations that surge within contexts are not naturally part of the platforms (Charland & Leroux, 2011).

Since users are more committed with the content consumed through their tablets than with their smart phones, it is necessary to ensure the compatibility of designs among different devices (David, 2012). In order to enhance UX, the application ensemble (design and code) must provide ways to fulfill these expectations that users have within certain contexts. To do that, it is necessary to look beyond UI and address the attention to the complete UX. The complete UX can thus be defined as a non-utilitarian aspect of interaction in which the focus alternates between the users affections, sensations, meaning and value of interactions on their routine,
even though such a definition still lacks of a overall consensus (Law, Roto, Hassenzahl, Vermeeren, & Kort, 2009).

5.2. Native Experience (NX)

Knoernschild (2011a) explains that NX is the ensemble of characteristics that an application has to have in order to provide the unique properties of the platform the application is running on. He points that NX is only one of many aspects of the total UX.

In addition, Knoernschild considers that it is not all application requirements that demand a native experience in order to achieve usability requirements. However, it’s important to have in mind that there are certain NX requirements to take into account when developing mobile applications.

5.2.1. Native integration

The possibility the application has to integrate with — to have access to — other applications, data and system information. These are elements that can be controlled in an application.

5.2.2. Native capabilities

The possibility the application has to have access to the native hardware capabilities.

Context variation in terms of hardware can be retrieved through the amount devices existing in the market today. Maybe the most apparent one is the case of Android devices ecosystem. In table 1, it is possible to see some elements that can be taken into consideration:
Table 1 - Variations of hardware elements that organizations must take into consideration when developing mobile applications

<table>
<thead>
<tr>
<th>Capability</th>
<th>Processing Power, antenna, storage, memory, compass, GPS, gyroscope, accelerometer, Bluetooth, graphic processing unit (GPU), device information, local notifications, calendar, contact Database, SMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>Physical size, pixel density, screen resolution, color depth, aspect ratio</td>
</tr>
<tr>
<td>Input</td>
<td>Touch screen, physical keyboard, camera, microphone, trackball³</td>
</tr>
</tbody>
</table>

Charland & Leroux (2011) affirm that the combination of these attributes with the different ways the user can interact with them can affect how the application will materialize.

**5.2.3. Native controls**

In order to provide the best of user experience, the different mobile platforms created their unique user interface guidelines (GUI). This collection of design patterns, styles and principles helps developers to maximize human interactions with the application, which can be seen as a better way to create good applications.

Charland & Leroux refer to some common user expectations, like “go back” or “context menus”, that are presented on different ways depending on which device you have in your hands. While Android and BlackBerry devices designed a physical back button and webOS⁴ uses a back button and a back gesture, iOS chose to achieve this task with a virtual button.

Another issue that both native and web applications have to deal with is how these devices determined the placement of the context menus. Charland & Leroux suggest that devices that use physical tab bars provide a poor user experience because there is a risk that the user will close the application accidentally.

³ Trackball is a pointing device that detects a rotation of a ball between two axes, like an upside-down mouse. Some mobile devices, such as BlackBerry and most HTC, have trackballs.

⁴ Mobile operating system based on Linux and used by Hewlett Packard.
Performance

Performance is the term used to define how much resource the application consumes during the runtime. Performance impact can be evaluated through both latency and execution time. While the latency is the data the application consumes through the network, the execution time is the time the application takes to interpret the code. In most of cases, the more code to interpret, the longer the execution time.

Offline capability

Offline capability means how the application can deliver content to the user without being connected. According to Knoernschild (2012a), an application that adapts to network access variation and limitation delivers a good UX.

5.3. Time to Market

Time to market is how long it takes to develop and deliver an application for the desired platform. Organizations must be aware of business objectives when using this evaluation criterion as it is risky to assume that faster means cheaper. Smith (2005) states that even though a professional team is the most valuable tool to accelerate a project, most organizations have difficulties to gather people that can achieve a good performance together. Essentially, the teams within organizations are constituted of people with basic skills, which are called an effective group.

5.4. Portability

Portability is the possibility to develop applications that toggle a wide range of devices. Some other characteristics are the ease to use a single toolset during the development of a mobile application and the code reuse. Portability is one of the two elements that influence the cost of development thanks to aspects as code reuse and platform support (Knoernschild, 2011a)
5.5. **Management and Maintenance**

These criteria focus on the ease of managing deployment and upgrades in the production environment.

The maintenance aspect handles the ease of testing, compiling, packaging and maintaining the source code of the application.

5.6. **Discoverability**

Discoverability describes how easy it is to search and find a mobile application.

5.7. **Distribution**

Distribution will handle the requirements organizations need to fulfill in order to deliver in-house applications. This requirement can also cover the ease of distributing the applications.

5.8. **Required Competences**

The ease to learn, understand and transfer skills. Organizations usually have a large number of developers, with knowledge of a wide variety of programming languages.
6. Different methods to develop mobile applications

In this chapter, I will focus on the framework types in order to develop mobile application. I will discuss the strengths and weaknesses of three develop methods and give Scania a panorama over the different ways to "go mobile". Furthermore, I will write about some criteria to consider when choosing these frameworks to a mobile development project.

6.1. Web

This part of the investigation focuses on methods to develop mobile application using web technologies, such as responsive design approach and mobile web applications that use HTML, CSS and JavaScript to deliver a mobile experience. After a short explanation of each method, a table with the evaluation criteria mentioned on the previous chapter and suggestions of appropriate methods to build new solutions will be presented.

6.1.1. Responsive Design

Responsive web design\(^5\), a term coined by Ethan Marcotte (2010), is not a way to create different facets of the same experience but an approach that permit developers to create the web solutions that can adapt to the media that renders them. It means that you can create flexible websites founded on a system of proportions and percentages and render the content to several resolutions and sizes.

To use a responsive approach when designing web sites, Marcotte shows us a couple of specific tools that help the designers to adapt the solution to the constraints of the browser window or the device that renders it:

- **A flexible, grid-based layout**: A grid that can resize itself as the viewpoint without compromising the original proportions. Being context-aware is the same as

\(^5\) Also known as adaptive design
understanding the ratio-based relationships between element and container. The fluid grid is just a foundation of a responsive design.

- **Flexible images and media**: CSS rules that prevent images from exceeding the width of their container. Some old browsers, like IE 6, don’t support those rules, which can make the images adaptation more difficult. In addition, IE 7 and lower and Firefox 2 and lower on Windows do not scale images adequately.

- **Media queries**: A mechanism for identifying types of media and inspecting the physical characteristics of the devices and browsers that renders the content. Media queries adapt the design to several different resolution and viewpoint dimensions.

![Figure 2 - Innovation online’s responsive design on three different screen sizes](image)

According to Marcotte (2010, p. 9), if you have in mind one of the web’s central characteristics, the flexibility, you can almost respond to the user’s needs by creating a responsive design. For example, not all information of a desktop application needs to be presented to a mobile user.
A website that is truly responsive starts with a flexible layout that is prepared for different kinds of devices, such as tablets, or even those that do not yet exist. Building a responsive website, developers should consider the “mobile first” or “progressive enhancement” approach. It initially involves the definition of an appropriate layout to the least capable devices that progressively enhance the design as the resolution increases (Marcotte, 2010, pp. 124-129).

Even though Marcotte (2010, p. 107) argues that responsive design serves one HTML document using flexible layouts and media queries, there are some authors, which suggest that different designs deserve different markups. They explain that because mobile users’ goals often differ from one context to another, only one HTML document does not fulfill the users’ needs. Not to mention the bandwidth and the time to load the entire document which has markups that will not be used. While this issue is a relevant question, due to restricted time and resources it will not be possible to analyze it in this thesis.

![Figure 3 - Obama's web site tries to mimic the behaviour of a native application with a different way to present the context menu. This responsive design pattern is called "off canvas" (Wroblewski, 2012)](image)

---

6 “Mobile first” is a way to start thinking about why web sites should be designed for mobile first, thus forcing web developers to focus, for instance, on the implied constraints

7 Progressive enhancement is a web design methodology in which the content is the focus and can be accessed from anywhere.
### Table 2 - Responsive design’s evaluation criteria

<table>
<thead>
<tr>
<th>Responsive Design</th>
<th>User Experience (UX)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Web sites using a responsive approach adapt the design to different screen-sizes/browser windows. By means of a strong markup(^\circ) and CSS rules, the design can be friendlier to mobile devices. With the knowledge of responsive design principles and progressive enhancement, it’s possible to reach a higher UX than ordinaries web.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Native Experience (NX)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Native Integration</strong></td>
<td>For instance, it is not possible for sites built with the responsive approach to get access to devices APIs, which means that they cannot use local data or integrate with local application.</td>
</tr>
<tr>
<td><strong>Native capabilities</strong></td>
<td>This approach does not have access to device capabilities, such as local contact database, camera, render interactive graphics, and upload documents and media.</td>
</tr>
<tr>
<td></td>
<td>However, Android and iOS allow the access to the GPS capability through the browser.</td>
</tr>
<tr>
<td><strong>Native controls</strong></td>
<td>It is possible to imitate some aspects of native controls on responsive web sites, such as context menus. However, the technique is still young and is not reliable.</td>
</tr>
</tbody>
</table>

| Time to market                   | Because responsive design is platform-independent, the time to develop and deliver a web application is almost the same as the one to develop and deliver a web site that not follow the responsive think.                                                                                     |
|                                  | However, to choose such approach means that the need to plan which—                                                                                                                                                                        |

\(^\circ\) Markup is a detailed instruction that indicates the format and structure for an electronic document.
<table>
<thead>
<tr>
<th>Portability</th>
<th>or if— information will be left out on smaller versions is crucial. Layout is not like content. It is always necessary to pay carefully attention in which context the application will be used.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>As long as web application development consider the different sizes of existing devices, it is possible to cater for a wide range of them.</td>
</tr>
<tr>
<td></td>
<td>In addition, to develop responsive web sites does not require any special tool.</td>
</tr>
<tr>
<td></td>
<td>The difference in this case lies in how the information will be presented on different devices and how the web application is optimized.</td>
</tr>
<tr>
<td>Management and</td>
<td>The deployment routine of a responsive approach is the same as the one of a common web solution.</td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Also, there is no need to compile and package the source code of the application.</td>
</tr>
<tr>
<td>Discoverability</td>
<td>The discoverability aspect of a responsive web application is the same to the one applied in common web sites. It means that a user that search and find the website via her/his mobile device will not differ to the user that search and find on the stationary computer. Consequently, the developer group has to work on search optimization in order to achieve discoverability.</td>
</tr>
<tr>
<td>Distribution</td>
<td>There are no special issues to handle in this question. Any Mobile Device Management is usable to distribute the application, once the web site built with the responsive approach is on the Internet.</td>
</tr>
<tr>
<td></td>
<td>Because security is beyond the scope of this study, it will not be analyzed.</td>
</tr>
<tr>
<td>Required</td>
<td>Web developers with good knowledge in HTML, CSS and JavaScript can fulfill the organization needs on this question.</td>
</tr>
<tr>
<td>competences</td>
<td></td>
</tr>
<tr>
<td>When do I use it?</td>
<td>When the organization can live without optimization for specific devices</td>
</tr>
</tbody>
</table>
When the access to server-side solutions is limited.

When users expect the same or similar content

When the site requires continuous updates (news, blogs and similar)

When the purpose is to serve images and text to a large audience

6.1.1 MWAs – Mobile Web Applications

As stated by W3C Recommendation (2010b), Web Applications\(^9\) are an ensemble of web pages that, by way of HTTP, provide “application-like” experience through a web browser. By means of powerful frameworks based on JavaScript, MWAs can bring a similar native look and feel to users. To understand how MWA work, it is necessary to look at the technologies that serve as a foundation for these frameworks.

In spite of that, it is important to understand that MWA development requires an attentively evaluation because the limitations of the web browser set the constraints to development.

\(^9\) Also known as web app
HTML5

HTML5 is a powerful tool that allows developers to create web solutions in web browsers with performance equal to desktop applications. With help of JavaScript, it’s possible for HTML5 to support APIs that bring complex system integration into the web browser, such as local data storage and geolocation.

HTML5 gives the developers the possibility to use SVG (Scalable Vector Graphics) and CANVAS to describe how a 2D image should be displayed. The difference remains in the way it is accomplished. While SVG is an XML-based language, CANVAS uses JavaScript to do the same thing.

HTML5 has already started providing access to several native features (see figure 4), whereof some are described below.

![Figure 5 - HTML5 features that most browsers support](image-url)
Geolocation

A functionality that comes to empower the web mobile development capabilities is the Geolocation API Specification, released in September 2010 by W3C. At the time of writing, it is possible to find a proposed documentation with some important non-normative\(^\text{10}\) sections about the API, which is good to consider.

One of them treats the question that the Geolocation API is still an unsure method in returning the actual location of the device even though it provides a high-level interface to location information associated with the position of the device. Despite the imprecision of the method, the API provides ways to deal with localization, such as find points of interest in the user’s area, show the user position on a map and update location-tagged status in social networking application. (World Wide Web Consortium (W3C), 2012c)

Offline mode

Even though the first steps towards the ability to run an HTML application in an offline mode are dated back to 2008, there are not any recommended methods from W3C. (2008; 2012a). However, it is possible to store data temporarily thanks to the HTML5 client-side storage.

Despite this possibility, the rapport from the W3C Workshop on Future of Offline Web Applications (2011b; 2012c) show us some obstacles developers are still facing today when developing offline applications.

> Instead of enabling web developers to build experiences, which seamlessly transition from connected to offline and back, AppCache forces them to write offline applications, which can access the web when connected. In effect, web pages, which reference a cache manifest, get served from AppCache even when online. (Langel, 2011)

Physical Device Orientation, device motion & compass calibration

Based on the way the device is being held, the user can control how the content can be presented. Physical device orientation, known as landscape and portrait views, device motion and compass calibration are features that W3C Geolocation Working Group are working on.

\(^{10}\) A non-normative section contains additional information that is not part of the standard.
Despite the broad use and compatibility with most browsers, HTML5 is still not ready to substitute HTML4, the web standard recommended in December 1999 by the World Wide Web Consortium. One of the possible reasons is that APIs are not sufficiently stable, which imply a longer process of tests and discussions in order to deliver a more than satisfactory web standard. In addition, the change in the standards and browser implementation might require changes in applications to keep them working (see on figure 5).

**CSS3 and CSS4**

Despite the fact that CSS3 as a whole is still not considered a web standard by the W3C, the adoption of this stylesheet-language takes the styling of web applications to a higher level. In order to allow a more flexible use and incremental improvement to CSS, the CSS Working Group\(^\text{11}\) has decided to introduce a modular approach, where individual approval is independent from each other.

Cederholm (2010, p. 2) explains that thanks to the modular approach, releases are developed faster, which allows web designers to use many of CSS3 properties before the publication of one single document that defines CSS Level 3. While some of these modules, such as media queries or CSS3 selectors, are already official recommendations, many others are only candidate releases or work-in-progress drafts (see on figure 6).

In order to experiment some properties in a real-world environment, browser vendors can add support to modules that are not yet finished introducing prefixes that enable the use of these properties. Cederholm (2010, p. 10) points out that these prefixes indicate “work-in-progress”, being subject of browser’s interpretation and implementation. The rule looses the prefix, once the module has become a recommendation.

\(^{11}\) W3C Working Groups enables W3C to create web standards, produce guidelines, test suites, reviews of deliverables and support materials.
Since 2011, the World Wide Web Consortium is working on, among other things, the CSS4 Selectors Level 4 (World Wide Web Consortium (W3C), 2011a), whose main function is to replace both the selectors Level 2 and Level 3.

*JavaScript*

It might be necessary to explain why JavaScript is an important tool when discussing about web development and mobile web development in particular.
JavaScript is not a new technology. According to David (2012), Netscape Communication released LiveScript in its web browser in order to enable interactivity in the web experience. Later enhanced versions of JavaScript, allowed developers to manipulate content, structure and style of documents through the platform and language-neutral interface called DOM. (World Wide Consortium (W3C), 2005).

JavaScript has become a main tool not only to build web applications’ server elements, but also frameworks to develop mobile applications. Most frameworks available on the market are using JavaScript libraries, such as JQuery, to deliver a closer native experience. As Severance (2012) declares, with the emergence of HTML5, JavaScripts level among programming languages tends to increase and dominate both desktop and mobile applications.

Frameworks

Mobile Web Application’s framework is still a young technology. Despite being a new phenomenon, frameworks a native look and feel so that the subtle difference between web applications and native’s UI could even be erased.

Apart from tools as Adobe Dreamweaver, it is possible to develop MWAs with IDEs and plugins such as Sencha Touch2, JQTouch, JQuery Mobile, Yahoo User Interface Library (UYUI 3), Wink toolkit12, iUI13, DHTMLX Touch14.

12 http://www.winktoolkit.org/
13 http://www.iui-js.org/about
14 http://www.dhtmlx.com/touch/ (free open source JavaScript library)
### Table 3 - Mobile Web Application’s evaluation criteria

<table>
<thead>
<tr>
<th>MWA – Mobile Web Application</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Experience (UX)</strong></td>
</tr>
<tr>
<td>Mobile web Application</td>
</tr>
<tr>
<td>frameworks allow developers to deliver a native look and feel, providing response to events such as swipe, tap and pinch.</td>
</tr>
<tr>
<td>Even though MWAs can deliver the richest Web UX, they cannot provide richer UX than native applications.</td>
</tr>
<tr>
<td><strong>Native Experience (NX)</strong></td>
</tr>
<tr>
<td><strong>Native Integration</strong></td>
</tr>
<tr>
<td>For instance, it is not possible for MWAs to get access to devices APIs, which means that they cannot use local data or integrate with local application. (see attachment 2)</td>
</tr>
<tr>
<td><strong>Native capabilities</strong></td>
</tr>
<tr>
<td>MWAs have no access to certain device capabilities, such as local contact database, camera, render interactive graphics, upload documents and media.</td>
</tr>
<tr>
<td>However, Android and iOS allow the access to the GPS capability through the browser.</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
</tr>
<tr>
<td>Performance can be affected due to the access to some native capability through the browser. In addition, complex JavaScript execution can affect the web experience.</td>
</tr>
<tr>
<td>Bandwidth-limited network can also be a performance-key factor</td>
</tr>
<tr>
<td><strong>Offline capability</strong></td>
</tr>
<tr>
<td>As stated before, the aim of providing offline capability to an application is not only to develop it to go work without connection, but instead to program an offline application that can go online when</td>
</tr>
</tbody>
</table>
In addition, it is worth to mention that APIs, such as Geolocation and Offline mode, are not yet reliable and are still in the experimental phase, considering the state of the W3C recommendation.

**Time to market**

Because MWAs are easier to produce than native applications and do not require approval, the time to release the application is considerably faster.

However, frameworks and tools to create MWAs can be a threshold to developers who are not familiar with them, increasing time to market.

**Portability**

Mobile web applications provide a large range of devices access to the content. However, it is worth to evaluate that some functions can fail from one web browser to another, because some browsers are still limited and not providing good support to new technologies.

To deal with this problem, it is possible to get help of progressive enhancement, CSS media queries, frameworks and feature detection on browsers that do not support JavaScript.

**Management and Maintenance**

MWAs are often hosted centrally, which facilitate the deployment and management, and offer the same benefits as desktop applications.

A single codebase to maintain make this option more attractive from a maintenance perspective.

**Discoverability**

Application storefronts facilitate the discoverability, which is not the case of MWAs. It is more difficult to find a mobile web application because they are limited to search on the web.

**Distribution**

Users have no need to update the application, once the applications are updated through a new version on the server.

Because security is beyond the scope of this study, it will not be analyzed.

**Required**

Web developers with good knowledge in HTML, CSS and JavaScript
<table>
<thead>
<tr>
<th>competences</th>
<th>can fulfill the organization needs on this question. Knowledge in using frameworks is also good to have. Even though developers can be familiar with JavaScript, it does not necessarily mean that they can manage the frameworks easily.</th>
</tr>
</thead>
<tbody>
<tr>
<td>When do I use it?</td>
<td>When the access to device capabilities is not a requirement When integration to native applications, such as calendar or the contacts database, or information stored locally is not a necessity When portability is imperative. Consider do not compromise the overall UX to the detriment of portability When the need to reach a wider range of users is a priority. Consider the item above when evaluating the needs of the application. When developing less strategic applications. It is also possible to optimize design and content as alternative When time to market is a priority. Consider also the items above when evaluating the application needs</td>
</tr>
</tbody>
</table>
Cross-platform frameworks, popularly known by its slogan “write once, run everywhere”, provide three similar ways to develop mobile applications: cross-compilation, custom container and hybrid. Given the complexity and ambiguity of the above mentioned concepts, the focus on this chapter is to address cross-compilation and hybrid methods in order to give Scania an overall view on this question.

Before developing this point, it is necessary to make a brief distinction between these two methods. The main difference lies on that the hybrid method wraps the code in a native web view, which allows the developer to gain access to native APIs, and compiles as native apps.

### 7.1 Cross-Compilation

Cross-compilation frameworks create native mobile applications (see figure 7) that target multiple platforms by converting a single code base to native applications (Knoernschild, 2011a). Knoernschild affirms that even though cross-compilation frameworks allow you to leverage more of the native experience, the platforms that it supports are limited and the code reuse is lower. It is because developers need to adapt the code to each target platform before deployment and distribution.

For instance, Appcelerator’s Titanium platform uses JavaScript, Kony Studio uses Java and Xamarin Mono uses C# and .NET as its source language.
7.1.1 Behind the scenes

Using a pre-compiler, front-end compile and a collection of python scripts, Titanium, for example, analyzes the code and builds a dependency hierarchy of its APIs in your application in order communicate and interact with the native SDK tools.

Figure 7- The same application built to different platforms: android (left) and iOS (right)
Titanium’s architecture. Titanium works as a bridge between your code and the native operative system

For instance, the JavaScript are not converted to Objective-C or Java. Instead, the JavaScript is interpreted at runtime on iOS and precompiled to bytecode on Android. Figure 8 shows the Titanium’s architecture where developers program the application in JavaScript that calls Titanium’s API through its SDK, which translates those calls on native equivalent.
<table>
<thead>
<tr>
<th>Cross-compilation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Experience (UX)</strong></td>
</tr>
</tbody>
</table>
| **Native Experience (NX)** | **Native Integration**<br>Applications build with a cross-compilation framework have a good integration with other applications data.  
**Native capabilities**<br>It is possible to get a richer NX by using APIs that allow the developer to take advantage of platform capabilities.  
**Native controls**<br>Cross compilation frameworks tend to deliver natives UI controls, called abstraction approach. Abstraction approach increases the UX. |
| **Time to market** | Some frameworks support a limited range of platforms, which requires additional tools to deliver application to other platforms.  
However, the fact that organizations develop applications only a single time helps to increase time to market, even though some adjustments must be made. |
<p>| <strong>Portability</strong> | Even though a greater number of frameworks provide a good toolset reuse, it is necessary to use native SDK simulator and emulators in order to test the application. |</p>
<table>
<thead>
<tr>
<th>Management and Maintenance</th>
<th>Separate maintenance will be needed once applications will be delivered to different platforms.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoverability</td>
<td>Given the fact that cross-compiled application is a native one, the characteristics for discoverability will be the same as for native applications.</td>
</tr>
<tr>
<td>Distribution</td>
<td>Given the fact that cross-compiled application is a native one, the rules for distribution will be the same as for native applications.</td>
</tr>
<tr>
<td>Required competences</td>
<td>Cross-compilation frameworks usually use programming languages that are familiar to enterprise developers, such as Java, C#, JavaScript and .NET, which contributes to lower cost and allow reusing of internal resources. However, cross-platform frameworks do not provide any tools for testing. It is only possible through the simulators provided by natives SDK.</td>
</tr>
<tr>
<td>When do I use it?</td>
<td>When the application requires access to device capabilities</td>
</tr>
<tr>
<td></td>
<td>When the application requires integration to native applications, such as calendar or the contacts database, or information stored locally</td>
</tr>
<tr>
<td></td>
<td>When portability or UX are a requirement</td>
</tr>
<tr>
<td></td>
<td>When time to market is a medium priority.</td>
</tr>
<tr>
<td></td>
<td>Consider the balance between usability and portability.</td>
</tr>
</tbody>
</table>
7.2 Hybrid

In this approach you use a Web View control and create an outer layer of interface between the user and the internal operative system, i.e. shell application, which in its turn is used to render HTML. A thin layer that serves as a bridge between the webview control and native devices capabilities wraps the web content. In other words, hybrid applications are able to operate without other software and run like a native application without requiring files on a server on Internet.

![Today's featured article](image)

**Compsognathus** is a monotypic genus of small, bipedal, carnivorous theropod dinosaur. The species **Compsognathus longipes** was the size of a turkey and lived around 150 million years ago, the early Tithonian stage of the late Jurassic Period, in what is now Europe. Paleontologists have found two well-preserved fossils, one in Germany in the 1850s and the second in France more than a century later. Many presentations still describe **Compsognathus** as a

![Today's featured article](image)

The **Court of Chancery** was a court of equity in England and Wales that followed a set of loose rules to avoid the slow pace of change and possible harshness (or “inequity”) of the **common law**. The Chancery had jurisdiction over all matters of equity, including **trusts**, **land law**, the administration of the estates of lunatics

![Figure 9 - Wikipedia's hybrid applications to Android (left) and iOS (right)](image)

Knoernschild (2011a) explains that developers must create a special layer to each desired target platform. This special layer can increase the time to market and even the necessity to recruit external resources if the knowledge on the programming language to specific targets is not fulfilled.
7.2.1 Toolkit

At the time of writing, there is a wide range of framework/toolkits that promise to wrap the source code developed with web technologies into a layer that communicates with the device. PhoneGap, also known as Apache Cordova, is one of the most singular tools that convert web code (HTML, CSS and JavaScript) to native applications and gives the developer access to APIs and applications storefronts.

Organizations have two ways to develop hybrid applications with PhoneGap:

- PhoneGap open source
- PhoneGap:Build\textsuperscript{15}

Both methods require the developer to download the PhoneGap files in order to get the necessary connection between the web code and mobile native capabilities. The difference between the methods is that with PhoneGap:Build the developer does not have to install any SDKs, IDE, plugins or Virtual Machine for testing. The only thing needed is a zip file containing

\textsuperscript{15} PhoneGap:Build is a cloud based service that take your application build with web technologies and compile into a native application ready for distribution on multiple platforms. PhoneGap:Build is still in BETA version and offers paid plans to those who needs to transform web code into native applications.
the HTML, CSS and JavaScript files that will be uploaded to the service. The next step is choosing the target platform for the application. It is worth to mention that the process to sign up an account to each particular platform is not included on PhoneGap.

One advantage is that the developer does not have to start from scratch to build a hybrid application. Frameworks as JQTouch, Sencha Touch2 or JQuery mobile can be used as a foundation, which can facilitate the development of applications.

![PhoneGap supported features on native platforms](image.png)

**Figure 12 - PhoneGap supported features on native platforms**
### Table 5 - Hybrid application's evaluation criteria

<table>
<thead>
<tr>
<th>Hybrid Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Experience (UX)</strong></td>
</tr>
<tr>
<td><strong>Native Experience (NX)</strong></td>
</tr>
<tr>
<td><strong>Time to market</strong></td>
</tr>
<tr>
<td>Portability</td>
</tr>
<tr>
<td>Management and Maintenance</td>
</tr>
<tr>
<td>Discoverability</td>
</tr>
<tr>
<td>Distribution</td>
</tr>
<tr>
<td>Required competences</td>
</tr>
<tr>
<td>When do I use it?</td>
</tr>
</tbody>
</table>
7.3 Native

This section of the thesis comprehends a short description of the native application and its strengths and weaknesses.

7.3.1 Native Application

Native applications\(^{16}\) are those designed to run in the environment it is being run in. It can be an application developed by the device manufacturer or by third-party developers. Manufacturers provide native software development kits (SDK) equipped with tools such as simulator, debugger and user interface builder, runtime support, which serves to improve UX and to build mobile applications

One of the strong advantages in comparison to the other techniques is the full access to the native user interface APIs and the unique functions available on the device.

However, native application development requires some form of application management in order to facilitate deployment, provisioning of new devices, refreshing provisioning profiles and delivering updates to applications. These variables, alone or in combination, can increase the cost throughout the development and management of native applications. On the other hand, native applications can in the long run reduce costs if the applications need less traffic over the network.

\(^{16}\) Also known as RMA, Resident Mobile Application
### Table 6 - Native application's evaluation criteria

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Experience (UX)</strong></td>
<td>Native Applications deliver the richest mobile UX.</td>
</tr>
<tr>
<td></td>
<td>However, possibilities for portability and code reuse decrease</td>
</tr>
<tr>
<td><strong>Native Experience (NX)</strong></td>
<td>Native Integration</td>
</tr>
<tr>
<td></td>
<td>Native Applications provide the highest potential for a good NX.</td>
</tr>
<tr>
<td><strong>Native capabilities</strong></td>
<td>SDK allows native applications to have access to all native user interface</td>
</tr>
<tr>
<td></td>
<td>APIs and unique functions available on the device.</td>
</tr>
<tr>
<td><strong>Native controls</strong></td>
<td>Native application deliver total control over the UI</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Native application support the best performance</td>
</tr>
<tr>
<td><strong>Offline capability</strong></td>
<td>Native applications provide the possibility to work in either a</td>
</tr>
<tr>
<td></td>
<td>disconnected or connected mode. It is possible for example to cache and</td>
</tr>
<tr>
<td></td>
<td>retrieve data and execute business logic.</td>
</tr>
<tr>
<td><strong>Time to market</strong></td>
<td>Native applications increase time to market significantly because they</td>
</tr>
<tr>
<td></td>
<td>need to be developed for different platforms.</td>
</tr>
<tr>
<td><strong>Portability</strong></td>
<td>Native applications are not portable. It means that a separate</td>
</tr>
<tr>
<td></td>
<td>application must be developed to each platform the organization wants</td>
</tr>
<tr>
<td></td>
<td>to target. In addition, a single toolset and language is used on native</td>
</tr>
<tr>
<td></td>
<td>SDKs.</td>
</tr>
<tr>
<td><strong>Management and</strong></td>
<td>Mobile Device Management-solution is a good way for organizations to</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **Maintenance** | gather, control, manage devices and refresh provisioning profiles.  
In-house applications are not automatically updated, so organizations need to notify employees and instruct them to install the update |
| **Discoverability** | In-house applications is easy to discover because organizations can host the application on an internal web server, set up an in-house application catalog or use a mobile device management solution.  
Besides, storefronts facilitate the discoverability of the applications developed to others than employees, i.e. consumers. |
| **Distribution** | For information about in-house distribution for organizations, see tables below  
To distribute applications to others than employees, i.e. consumers, the organization will depend to the approval of storefronts.  
Because security is beyond the scope of this study, it will not be analyzed. |
| **Required competences** | Native applications require developers with knowledge on the native language.  
iOS: Objective-C  
Android: Java  
The exception is Windows 8, which uses a different approach called Metro style.  
Windows 8: HTML5, CSS3, JavaScript; XALM med C++, C# or Visual Basic; |
| **When do I use it?** | When the application require access to device capabilities, such as gyroscope, video recording, compass and accelerometer  
When the application require integration to native applications, such as calendar or the contacts database, or information stored locally |
When the application requires sophisticated graphics or custom controls to deliver the UX

When portability is not a big issue

When the purpose of the application cannot be achieved through a browser

When there is a need to access the application in offline mode. An example could be the access during a flight.

Distribution

The focus here is on how Scania can distribute native “in-house” applications through three stores: App Store (Apple), Google Play (Android) and Windows Store (Windows 8).

For each platform, it is necessary to create an account and pay the fee.
Table 7 - Native iOS registration and distribution of in-house applications

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>App Store</em></td>
<td>Create a dedicated Apple ID for this program</td>
</tr>
<tr>
<td></td>
<td>Dun &amp; Bradstreet (D-U-N-S) number. This information is necessary to validate the status as a business entity.</td>
</tr>
<tr>
<td></td>
<td>Others documents, such as business license, may be required to complete the identity verification.</td>
</tr>
<tr>
<td></td>
<td>Agree to the Enterprise program License Agreement</td>
</tr>
<tr>
<td>Cost</td>
<td>$299/year iOS Developer enterprise program membership</td>
</tr>
<tr>
<td>Enterprise App Stores</td>
<td>Wireless App Distribution (Apple Inc., 2012): host the application on a web server that employees get access and notify the users the URL via email, SMS, notifications or other methods, such as QR code.</td>
</tr>
<tr>
<td></td>
<td>In-house App Catalog, a self-service model where the organization can provide an own designed portal for distribution of the applications</td>
</tr>
<tr>
<td></td>
<td>Mobile Device Management</td>
</tr>
<tr>
<td>Ad-Hoc distribution</td>
<td>Up to 100 registered devices</td>
</tr>
<tr>
<td>Volume Purchase Program (VPP)</td>
<td>Organizations that enroll in the program can create a volume purchase apps for the users (only available in the US)</td>
</tr>
</tbody>
</table>
### Table 8 - Google Play registration and distribution requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Google Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>It was not possible to find any information about Google Play and some Enterprise program.</td>
</tr>
<tr>
<td>Cost</td>
<td>$25 USD one-time registration fee</td>
</tr>
<tr>
<td>Enterprise App Stores</td>
<td>Applications are not required to be on Google Play to get distributed. A mobile device management should be a simple solution for this issue</td>
</tr>
</tbody>
</table>

### Table 9 - Windows 8 registration and distribution requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Windows Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>Windows Store is not open for general submissions or registrations yet. Only those who own a registration code can get help to set up an account (Microsoft, 2012)</td>
</tr>
<tr>
<td>Cost</td>
<td>$99 membership fee&lt;sup&gt;17&lt;/sup&gt;</td>
</tr>
<tr>
<td>Enterprise App Stores</td>
<td>Organizations can distribute the application through the Windows Store and adhere to the same certification policies. Another possibility is to deploy applications directly to end-users without passing through the Windows Store. Windows Store will also support line-of-business&lt;sup&gt;18&lt;/sup&gt; (LOB) application</td>
</tr>
<tr>
<td>Volume Purchase Program (VPP)</td>
<td>Organizations can use a signed receipt that enables to identify the user running the app.</td>
</tr>
</tbody>
</table>

---

<sup>17</sup> Enterprise registration fee, individual registration fee will be $49 USD. The article do not specify is the fee is annually or one-time registration fee (Lebland, 2011)

<sup>18</sup> Mobile Route Accounting is an example of LOB
8 Conclusion Mobile Development Methods

After having written about different methods to develop mobile applications, it is important to summarize how Scania can use these methods based on actual resources and competences. In addition, I present in a visual and summarized way (see on table 9) the strengths of each method.

Developing applications at Scania

The questions regarding which method to choose when developing mobile application are not easy to answer. There is not only one solution that can fit the needs of all the applications that will be developed at Scania. Each application needs to be evaluated not only from a technical perspective, but also from a user-centered way. One of the best ways to quickly evaluate a project is questioning why the project shall deliver a mobile solution, what is the advantage of the mobile solution compared with a standard web solution.

The question can be answered considering both factors allied with how the organization is prepared to carry the cost of an introduction of a mobile solution to the employee's routine.
Table 10 - An overview of strengthens and weakness of each method in relation to the evaluation criteria.

<table>
<thead>
<tr>
<th>Method</th>
<th>Responsive design</th>
<th>Mobile Web Application</th>
<th>Cross-Platform</th>
<th>Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>User experience (UX)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Native Experience (NX)</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Time to Market</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Portability</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Management and Maintenance</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Discoverability</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Distribution</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Required Competences</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

- **Strong**
- **Intermediate**
- **Weak**
8.1.1 Web

Working on the assumption that the three most important factors of the evaluation criteria are time to market, management and maintenance and required competences, the web solutions could be a short and satisfactory answer. However, the lack of reliability of the current web standards present makes these methods less favorable to mobile developments. Most of the new tags, modules and rules of HTML5 and CSS3 are still not a final recommendation and are being tested by the browsers that adopt their own way to use such elements. Not to mention the security question that web technologies have to deal with, which is beyond the scope of this thesis.

The portability is also a criterion that could be related with the assumption above. Scania has a considerable amount of devices with access to its network, which is an important aspect to highlight. As I have mentioned before, the choice of the technology to supply this demand shall not be confined to the portability criterion, once it affects the user experience. In other words, it means that native applications oppose to web applications, because while the former can deliver the best user experience, the latter can reach a wider range of devices. The fact that Scania has a wide knowledge and resources on web development, make the web methods desirable to adopt.

Nevertheless, it is worth to underline that as long as web technologies progress, the web standard-candidates assume a reliable form and the access to device capabilities become stronger, web technologies are certainly a favorable method to consider in order to develop mobile applications.

8.1.2 Cross-Platform and Native

Concerning cross-platform technology, given the fact that applications build with such frameworks are eventually native ones, the questions regarding discoverability and distribution are covered by the native applications. By means of an adoption of a Mobile Device Management (MDM) by Scania, discoverability and distribution of native application can be easy to deal with.

Another point to discuss is that cross-platform frameworks cannot be considered independent from the native environment, even though they can deliver satisfactory native applications. As mentioned before, in order to test the application, it is necessary to use native SDK, which demands skills that are not internally available at Scania. The lack of the required skills affects substantially the maintenance and management aspect.
In conclusion, the fact that the organization does not have a mobile strategy addressing device adoption or software approach makes the mobile issue even more difficult to address. Additionally, it is crucial to emphasize that while Scania does not have a procedure on how to develop, maintain and coordinate the development of mobile applications, single initiatives will continue to emerge, which increases the cost and affect the quality of their IT products and services.
9 Part Two – Case: Rundtursbuss application

This part of the thesis describes the creation process of the design and technology proposal to the Rundtursbuss application based on its requirements (see attachment 3). A description of the method used to compile these requirements and the method to sketch the mockup will also be described.

9.1 Internal buses at Scania

In order to provide an alternative transport solution to the employees between the buildings at the industry area, Scania has created two alternatives to supply this issue. The “tour-buses” introduced in the employee’s routine in the seventies and the private cabs called KomFort (come right way or comfort) in the nineties.

The two tour-buses lines, blue and yellow, circulate after a timetable around the internal and external industry area, respectively. There are 12 bus stops within the Scania’s complex, whereof both lines have three in common. Between 07:00 and 16:50, eight buses are available; there are special times when the number decreases in half. During these special times the interval between the buses increase with 10 minutes.

The two lines transport approximately five hundred employees every day. In April 2012, the number of passengers reached almost 10 000. To support this traffic, HSUR, the department in charge of the buses, disposes of 13 Volkswagen’s buses and two Skoda superb, driven in shifts by 23 drivers Monday to Friday.

9.1.1 Design

Designing interactive systems involves more than the technology itself. It involves a considerable amount of variables that the designer needs to deal with in order to create a useful system. Among these variables, we can consider the evaluation process as a key feature in a design process. (Benyon, 2010, pp. 50-55). Another key stages of interactive design addressed by Benyon are understanding and envisioning both conceptual and physical design.
9.1.2 *Sketches vs. Prototypes*

Buxton (2007, p. 139) argues that both sketches' and prototypes' purposes are located in different stages of the design process. Two of the main differences between sketches and prototypes are that the latter requires more investment and takes longer to build.

![Diagram showing the difference between sketches and prototypes](image)

**Figure 13** - Buxton explains that a prototype is a continuation phase of a sketch. Contrast in purpose and form is the main difference between the methods (2007, p. 140)

As mentioned before, this study will only focus on the sketch phase.

By means of the design framework PACT, shortly described in chapter 3, it will be possible to understand the design situation of the application (Benyon, 2010, p. 26). Furthermore, by using the PACT framework the designer has the possibility to take on the four elements that compose and involve interactive systems.
I will also make use of “The effect map” method. This method consists of describing the desired effects and providing a hypothesis about how those effects can be achieved. (Ottersten & Balic, 2012).

Both methods are described as bringing more benefits to the project if they are made during the concept phase.

9.1.3 Personas & Scenarios

In order to understand the need of the system that will be designed, I have created some personas (see attachment 4) that represent the stakeholders that the system will be designed for. As Benyon (2010, p. 56) propose, the personas were created with a name, some background and goals in order to gaining knowledge about the habits and aspirations of the final users.

I have also created some scenarios (see attachment 5) that promote an activity-oriented view of designed artifacts. Benyon (2010, p. 64) sustains that both scenarios and personas are fundamental techniques for interactive system design because they can be useful to understand, envisage, evaluate and conceptualize design. The author presents four different types of scenarios: stories, use cases, concrete and conceptual scenarios.
9.2 Methods

Below, there is an overview of the process used to create the requirement list of the case study, a general analysis with the help of the PACT framework and a description of the effect map of the application *Rundtursbuss*. The two latter methods have been mentioned in the previous chapter.

9.2.1 Requirements

Technology as a means to facilitate the employee’s life at work has always been encouraged by Scania as an organization. However, the will to develop a mobile application to the Scania buses that transport employees around the industry area was not explicitly demanded by the Head of Services and Management department. Consequently, there were no requirements to start working from.

To compile a requirement for this application, we decided to start with a brainstorm session in the form of a workshop. Because of time restraints and lack of resources, only members of the IOAW were involved in this process.

The one hour long session was structured as follow:

- During 10-15 minutes, the participants have written all kind of ideas regarding the application's functionality. Each idea was documented on a Post-it.

- Each participant presented their ideas and explained to the others how the idea could be implemented. In addition, they have related some experiences when using the service.

- Similar ideas were put together in order to find a way of categorizing them.

- Seven categories were found where all the ideas were listed.

Given the fact that all participants are used to using the internal bus service, most ideas were based on the employee’s experiences with the service.

Before sending the requirement list to IOC, it was necessary to refine the functionalities. The categories have accordingly been reevaluated and assembled in order to simplify the application.
The feedback from IOC was mainly to divide the functionalities into steps that could be developed in future releases. These were the categories that have been chosen:

- **Journey planner** – Into this category, it will be possible to get the time of the buses based on either the user localization or on the start and destination of the trip.

- **Timetable** – Overall information about the time schedule of the buses.

- **Favorites** and the possibility to use a reminder are also included.

- **Map** – Here, it is possible to see all the bus stops on a map and the route of the buses.

- **Info** – Possibility to book KomFort car/buss to individual or group purpose, service time, phone number and explanation of KomFort-service and information about disturbances.

### 9.2.2 PACT

To get an understanding of the design situation of the application the PACT framework was used. The acronym stands for four necessary elements to be used during the design process.

**People** – Stakeholders who will use the application.

Employees and consultants who need to be transported from one building to another are the target group of this application.

**Activities** – In which activities the application will be used.

**Contexts** – In which context those activities are included in.

According to Benyon (2010, p. 36), because each activity takes place in a certain context, it’s necessary to analyze both, activity and context, together.

- The activities will take place indoors and/or outdoors. Passengers want to plan their journey before leaving their house or the office. There is a possibility that they make use of the application outdoors or even on a bus/car/train towards work.

- Most activities are work-related, in which employees or consultants are travelling between buildings in order to attend meetings. There are occasions when the service is used in between the train station and the offices.
Technologies – Which technologies the application needs to work properly.

- The need of Internet connection is not necessarily a must have, because the time schedule of the buses is constant. The need of connection is only mandatory if real time functions are available.

- Buses time schedules and stops need to be stored in a database

- A RSS can feed the application with the most recent deviations such as interruptions of the travels.

- A map with the positioning of all bus stops and the route of the buses. On a later release it is desirable to introduce GPS positioning where the users can find the closest bus stop based on their localization. There is also a will to bring real time bus information in the application.

9.2.3 The effect map

In order to complement the PACT framework and bring to light the desired effects of this application, I will use the effect map method proposed by Ottersten & Balic. A few factors to take in consideration in building the effect map to the application on this study is summarized below.

Objectives with the application: Why will this application be developed? – Changes and improvements that you want to reach.

- To facilitate the round-trip of employees and eliminate employees downtime.

In order to estimate if the desired effect described above has been reached, there are three factors to take into account:

Parameters: While we should consider the increasing number of the passengers on buses as a measure factor, it could also be interesting to use the employee's expectations around the time consumed when waiting for the bus as a second variable.

Numerical value: – the number of the passengers during a month and the passengers’ expectations on the waiting time.
The measure approach: The department responsible to take care of the bus service has already a mechanism to measure the number of passengers traveling on buses. A way to detect an increase in use of the service is to compare the numbers with earlier years.

Given the fact that an IT tool will be introduced into employee’s routine, questionnaires could be distributed regularly to measure their expectations.

*Target Group: Who will create the benefits through the use? Who will be affected by the application?*

**Regular users:** users that have meetings in different buildings around Scania regularly.

- New and long-term employees that are in motion everyday and have meetings in other places than in their buildings.
- New and long-term consultants working in a full-time or part-time manner that have meetings in other places other than in their buildings.

**Casual users**

- Visitors that accompany employers and consultants to meetings in other buildings.
- Temporary Consultants.

**All users**

- Remaining users.

**Affected agents**

- Employees and consultants who use the system can decrease downtime and plan their journey easily.
- The department in which these employees and consultants works. Because the downtime can decrease, the production time can consequently increase.
- The department responsible in supplying the increasing demand of transportation is one of the stakeholders that will be affected.
The design and requirements proposal that this study will deliver attend the needs of the personas created to represent the main target groups, i.e. the regular users of the system. After the requirements workshop, where some experiences have been related, I have created three personas with different experiences and positions in the organization that represents the regular users.

Usage goals and Actions

Usage goals describe what the different target groups needs or expect and what should happen in order for the goals to be achieved. (Ottersten & Balic, 2012, p. 51). In this study, I will only focus on regular users needs. In addition, actions describe what must be done to enable users to create what they expect.

Usage goal 1: the user needs to plan the journey towards the work and during the day.

Actions:

• Choose departures and arrivals.

• Choose which and how the information should be presented based on either the actual timetable or the future ones.

• See and choose the convenient time.

Usage goal 2: the user wants to get informed about the bus schedule when going to a meeting.

Actions

• Choose the actual location.

• Choose the bus line.

• Find the convenient time.

Usage goal 3: the user needs to get informed about the time to the next bus.

Actions:

• Choose departures and arrivals location.
• Choose how the information will be presented based on the time to the next bus. A list of five destinations will be presented. Two of them before the actual time and two of them after. The “actual time” can be five to ten minutes from the moment the user request the information.

• Look and choose the convenient time.

**Usage goal 4:** the user wants to save certain routes for a future consultation.

**Actions:**

• Choose departure and arrival location.

• Choose which and how the information will be presented based on either the actual timetable or future ones.

• Look at the desired route.

• Touch the button with a star on the top of the application to save as favorite.

**Usage goal 5:** the user wants to be reminded about the time to leave for a meeting.

**Actions:**

• Choose departures and arrivals location.

• Choose which and how the information will be presented based on either the actual timetable or the future ones.

• Look and choose the desired route.

• Touch the button “Reminder”.

• Choose when to be reminded.
10 Conclusion and Recommendations

Based on the effect map along with contexts to and technologies used in the previous chapter, I could identify some issues that are worth considering when building the application Rundtursbuss. The analysis of each criteria aided in the evaluation of the application requirements and in reaching the conclusion that the most promising methods to develop the application Rundtursbuss are either native or cross-compiled ones. In table 10, it is possible to read a detailed explanation how the recommendations and how the criteria was evaluated.

Furthermore, as it has been mentioned before, the higher the portability, the poorer is the user experience. It means that native applications are in a contrast to web applications and while the first one can deliver the best user experience, the second one can reach a wider range of devices. However, it is worth to underline that as long as web technologies progress, the web standard-candidates assume a reliable form and the access to device capabilities become stronger, web technologies are certainly a favorable method to develop this application.

Although the considerations will be presented below, I would like to emphasize that the recommendations on this chapter are only related to this case study in particular. They must not be considered as general ones, even though some of them can be applied to several applications. In addition, it is also necessary to point out that the recommendations in general are always based on the application requirements, which means information and functions needed for the system to achieve its purpose.

In conclusion, I would like to point out that while Scania does not have a procedure on how to develop, maintain and coordinate the development of mobile applications, single initiatives will continue to emerge, which increase the cost and affect the quality of the IT products.
<table>
<thead>
<tr>
<th>Rundtursbuss</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Experience (UX)</strong></td>
<td></td>
</tr>
<tr>
<td>Regular users at Scania with different degrees of technical knowledge will use the application. Considering this aspect, the user experience seems to be an important factor in this case.</td>
<td>Foremost Native.</td>
</tr>
<tr>
<td></td>
<td>Native application deliver the richest mobile UX.</td>
</tr>
<tr>
<td><strong>Native Experience (NX)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Native Integration</strong></td>
<td></td>
</tr>
<tr>
<td>According to the requirements, there is no need to use integration with for example local data on the device. However, if there is a will to develop the application further, it is desirable to consider if the method used support future needs.</td>
<td>Foremost Native</td>
</tr>
<tr>
<td></td>
<td>Cross-compilation framework can also be a solution to consider.</td>
</tr>
<tr>
<td></td>
<td>Native applications provide the best NX.</td>
</tr>
<tr>
<td><strong>Native Capabilities</strong></td>
<td></td>
</tr>
<tr>
<td>According to the requirements, there is no need to use integration with devices. On the other hand, a will to develop the application further exist, in form of GPS positioning, and therefore it is desirable to consider if the method used supports future needs.</td>
<td>Foremost Native</td>
</tr>
<tr>
<td><strong>Native controls</strong></td>
<td></td>
</tr>
<tr>
<td>A consistent presentation of the look, feel and behavior of a native application is expected by the user.</td>
<td>Foremost Native.</td>
</tr>
<tr>
<td></td>
<td>Cross-compilation deliver higher UX than Hybrid applications but a</td>
</tr>
<tr>
<td><strong>Performance &amp; Offline</strong></td>
<td>lower portability.</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Some delay of information presentation is accepted, but it does not compromise the application functionality.</td>
<td>Because hybrid does not have the same access to native capabilities, controls and integration as cross-compilation framework, it is a method to disregard.</td>
</tr>
<tr>
<td>Because Internet connection is not a must have, the application needs to be coded on constant offline mode and occasionally get access to the network.</td>
<td>Cross-platform framework can deliver the offline and performance requirement.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Time to market</strong></th>
<th>Cross-platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considering the application has not a hard deadline to be delivered, the time to market is not a determinant factor. In addition, the development can takes place in-house with the organization's existing resources.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Portability</strong></th>
<th>Foremost cross-compiled.</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application is meant to be used by employees and consultants that use different sort of devices.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Management and Maintenance</strong></th>
<th>Organizations developing native application do not necessitate deployment in any storefronts. The code can be maintained on the organization server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less code to maintain is better to maintain.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Discoverability</strong></th>
<th>Native or cross-compilation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any issue on the requirement list matches with this criteria.</td>
<td>Employees and consultants using the organization devices does not need to find the application, once it can be delivered by MDM.</td>
</tr>
<tr>
<td>However, I assume the discoverability level shall be high, due the application aims, mentioned on the previous chapter.</td>
<td>Assuming Scania does not have a MDM and shall develop a native or</td>
</tr>
</tbody>
</table>
### Distribution

| Cross-compiled application, it will be necessary to server the application somewhere and communicate the issue to those who wants to use it. Otherwise, this issue will be difficult to solve. |

| Any issue on the requirement list matches with this criteria. I also assume that the requirements for distribution also have a high level. |

| Developing in native or cross-compiled environments requires more maintenance because the application needs to be delivered on different platforms. However, depending of the system that will manage the application, there can be several ways to distribute the application to the target group. |

### Required competences

| Scania can use its own developers to program this application in-house. Still, the lack of an application owner and a responsible group the production and development can be affected. |

| Foremost Cross-compilation. Native. |

### Result

| Native or cross-compilation frameworks are the methods that the application Rundtursbuss should be developed with. |
11 Acknowledgment

I would specially like to thank Lisa Petersson and Jan Laestander at Scania IT AB for having believed in my potential and competence to accomplish this investigation; Johan Berggren for his patience, support and good discussions during the work period; all my colleagues from IOAW for being there and Petri Lankoski at Södertörn University for guiding me through this thesis work. I would also like to thank Paola Sartoretto and Clarice Goulart for helping me with texts adjustments and for their unconditional support and friendship. Last but not least, I would like to thank my partner Joseph Ah-King for his patience, support and love.
12 Bibliography


13 Attachments

Attachment 1 – Mockup
# Attachment 2 – Road Map Working Group Device API

<table>
<thead>
<tr>
<th>Specification</th>
<th>Internal draft</th>
<th>Public Working draft</th>
<th>Stable draft (Last Call)</th>
<th>Implementors feedback (CR)</th>
<th>Standard (Rec)</th>
<th>Test Suite</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Priority APIs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery Status API</td>
<td>01 May 2012</td>
<td>15 Sep 2011</td>
<td>29 Nov 2011</td>
<td>08 May 2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contacts API (reading from addressbook)</td>
<td>09 Feb 2012</td>
<td>9 Dec 2010</td>
<td>16 Jun 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTML Media Capture (camera/microphone interactions through HTML forms)</td>
<td>15 May 2012</td>
<td>14 Apr 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Media Capture API (aka getUserMedia, programmatic access to camera/microphone)</td>
<td>31 Dec 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Information API</td>
<td>31 Jan 2012</td>
<td>2 June 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proximity Events</td>
<td>21 May 2012</td>
<td>2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vibration API</td>
<td>01 May 2012</td>
<td>17 Nov 2011</td>
<td>2 Feb 2012</td>
<td>08 May 2012</td>
<td>draft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Intents (service discovery and light-weight RPC mechanism for web apps)</td>
<td>14 Mar 2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Work happens in the <a href="#">Web Intents task force</a></td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calendar API</td>
<td>05 May 2011</td>
<td>19 Apr 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dependency on TZDate</td>
</tr>
<tr>
<td>Menu API</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permissions for Device API Access</td>
<td>30 Sep 2010</td>
<td>5 Oct 2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calendar API</td>
<td>05 May 2011</td>
<td>19 Apr 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dependency on TZDate</td>
</tr>
<tr>
<td>Menu API</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permissions for Device API Access</td>
<td>30 Sep 2010</td>
<td>5 Oct 2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>At risk</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Audio/Video (read only)</td>
<td>05 May 2011</td>
<td>19 Apr 2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dependency on TZDate</td>
</tr>
<tr>
<td>Web Audio/Video (write)</td>
<td>21 May 2012</td>
<td>2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dependency on TZDate</td>
</tr>
<tr>
<td><a href="#">Web Audio/Video (write)</a></td>
<td>21 May 2012</td>
<td>2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dependency on TZDate</td>
</tr>
<tr>
<td><a href="#">Web Audio/Video (write)</a></td>
<td>21 May 2012</td>
<td>2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dependency on TZDate</td>
</tr>
<tr>
<td>Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No editor — currently at risk, probably to be merged into calendar API</td>
</tr>
</tbody>
</table>

## Informative documents

| MediaStream Capture Scenarios | 6 Mar 2012 | 6 Mar 2012 | | | | | Joint deliverable with the [Web RTC Working Group](#) through the Media Capture Task Force |
| Device API Access Control Use Cases and Requirements | 16 Mar 2011 | 17 March 2011 | | | | | |
| Privacy Requirements | 23 Jun 2010 | 29 Jun 2010 | | | | | |

## Exploratory work

| Discovery | | | | | | | |
| Privacy Ruleset | 6 Oct 2010 | | | | | | |
Attachment 3 – Requirements

1. Journey planner
   a. Journey planner (from, to)
      i. Favorites
   b. Reminder

2. Map
   a. Map/GPS Location (bus-stop on a map)
   b. Bus route maps for around Scania
   c. Real time position of the buses on the map

3. Timetables

4. Favorites

5. Info
   a. Book KomFort\(^{19}\) car/bus
   b. Book KomFort (for people in the same department)
   c. Time, phone number and explanation of the service
   d. Disturbance & latest information

\(^{19}\) KomFort is a Scania car that can be booked any time under a work journey by the employees.
Attachment 4 – Personas

13.1.1 Tor Svanström,

- 27 years old
- Tor will be moving from a student room on Götgatan to an appartment nearby Stockholm’s Södra.
- He is single, but has a girlfriend who lives in Umeå
- He is interested in technology and when he is not building his own computer he likes to walk and cycle on his free time.
- Tor owns the most recent iPhone and has got an HTC Acer from Scania InfoMate.
- Developer at Research and Development
- Graduated in Mechanical Engineering from the Royal Institute of Technology
- New consultant at Scania

13.1.2 Karl Jonsson

- 45 years old.
- Karl is married; he lives in a house in Gnesta with wife and two children.
- He loves driving lorries. He took his truck drive-license at age of 18 and since then has dreamed about to buy his own Scania truck. He is sedentary and rather stays in front of the TV than takes a walk.
- Karl loves all new technologies on Scania trucks but he has no interest in learning new things.
- He has worked at Scania for ten years and since 2007 works as Production Leader.
- He owns an old Sony Ericsson model but has got a smart phone from Infomate.
Karl is graduated in Industrial Economy from the Royal Institute of Technology.

13.1.3 Helena Andersson

- 33 years old.
- She lives together with John Wineberg and their daughter in Södertälje. They have recently moved to Södertälje because of the work.
- Helena and John started work at Scania after having participated of Scania trainee program, for five years ago.
- Her first role at Scania was as project leader on an IT area in a Research & Development department. Now she works as a Section Manager at InfoMate, a role that please her very much.
- She lives for her work and when she is not at the office, she is working from home.
- Helena is a very busy woman and when she gets stressed, she goes swimming. She loves traveling and she does it in a regular way.
- She owns a HTC Desire and an iPad that she is used to work with. Helena has also a work phone that she takes home every day, even during long weekends or the vacation period.
- Helena is graduated in Media Technology from the Royal Institute of Technology and has a doctoral exam in the same area.
13.1.4 Tor Svanström,

1. Tor lives in Stockholm and uses SJ to come to work every day. Because he heard about some buses drive around Scania, he started to take either the blue bus or the yellow one to building 117.

2. Although he access InLine constantly to read the timesheets, he has never read something about that buses are only for taking passengers for business meetings, until a colleague told him. He does not care about it because he noticed that is a common behavior at Scania.

3. Tor does not like to come late to work and because he is not so good at remembering the buses schedules he is often coming after 9 o'clock. Tor is also coming late to meetings because he does not know the Scania area and often takes the wrong bus.

4. He thinks that the schedules are complicated to understand and he would appreciate that a technical solution would be implemented.

5. InfoMate chose Tor to be one of the pilot testers to the application "Rundtursbuss". The application is intended to help him to come in time to work and to his meetings.

13.1.5 Karl Jonsson

1. Karl is a quiet and harmonious man. He never worries. Because he does not like to exercise, he is always waiting for the blue bus to take him to a meeting, even though the building is in a range of 500 m.

2. He rarely signs on to InLine before a meeting and, when he does, he does not find the timetables. Therefore, Karl prefers to wait for the bus at Chassiporten, even if he has to wait for a long time. Because he does not know the timetable by heart, he can wait longer during the drivers’ breaks. Karl notices that he spend a lot of time waiting or travelling, which generates much downtime.

3. Karl has talked about the downtime question with his superior that told him about the new concept that InfoMate is testing for the "Rundtursbuss".
4. Since there is only three applications he uses the most on the work phone, SL-application, to see the timetable from Gnesta to Södertälje Hamn, Calendar and Email, "Rundtursbuss" must be easy to learn and use so Karl could use it.

5. Considering the absence of a wireless communication on the industry area, it is necessary that the application in question works in offline mode

13.1.6 Helena Andersson

1. Helena is always on the run. Her calendar is always full of activities and meetings around Scania. She works at the building 209, but she is rarely at the office.

2. She usually walks to the office, but during the autumn and winter she prefers to take the first Scania yellow bus that appears. Because she always takes different ways to come to work, she does not know the entire bus schedule by heart.

3. She loves planning and usually uses her calendar on the cell phone to remind her when she must go to the bus station. Helena thinks that finding the right line and the right time via the PDFs via InLine on the telephone is an annoying and irritating moment.

4. She has downloaded all the timetables for the yellow buss on the iPad in order to plan her time in advance. However, she experienced that it was not the best solution since she gets irritated frequently. She does not think it is practical.

5. Helena's work requires that she is constantly online and needs to have access to the network. An application that helps her to plan her journeys with the possibility to save the favorites routes and a reminder could fit her needs.

6. She got to know that a group at InfoMate was in an initial phase of the concept "Rundtursbuss" and asked to be a volunteer to test the application.