KTI - Kapsch Tablet Infrastructure

Götze Sebastian, Hammer Samuel, Kaufmann Michael, Müller Konstanze, Steinhäuser Philip

DIPLOMARBEIT

eingereicht in der Abteilung

Höhere Informatik

der HTBLVA Spengergasse Wien 5

im Mai 2015

Eidesstattliche Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

HTBLVA Spengergasse Wien 5, am 15. Mai 2015

Sebastian Götze	Samuel Hammer
Michael Kaufmann	Konstanze Müller
Philip Steinhäuser	

Kurzfassung

Aufgabenstellung

Die Aufgabe von Kapsch an das Projektteam ist es, verschiedene Softwarelösungen zum Systemschutz von Android-Tablets zu testen. Dazu soll ein Untersuchungsbericht angefertigt werden. In diesem werden die Ergebnisse festgehalten und verglichen, um das beste System für die Kunden von Kapsch zu ermitteln. Ein Prototyp in Form eines Android-Tablets ist ebenfalls zu erstellen. Dieser soll mit dem besten, aus dem Untersuchungsbericht hervorgegangenen, Schutzsystem ausgestattet sein.

Realisierung

Die Firma Kapsch wünscht einen Untersuchungsbericht, aus dem hervorgeht, welcher Systemschutz für Android-Tablets am besten für spezifische Kunden geeignet ist. In diesem Bericht werden 2-3 verschiedene Systeme zur Absicherung eines Android-Tablets verglichen. Darin wird auf die Vor- und Nachteile des jeweiligen Systems eingegangen, sowie auf das Fehlen von Funktionen aufmerksam gemacht. Dies ermöglicht dem Auftraggeber das bestmögliche System für seine Kunden zu ermitteln.

Ergebnisse

Neben dem Untersuchungsbericht wird zusätzlich ein Tablet als Prototyp abgeliefert. Dieses Tablet ist mit dem Schutzsystem ausgestattet, welches aus dem Untersuchungsbericht als am empfehlenswertesten hervorgeht.
Am Ende des Projekts stehen 2 Endprodukte.

Der Untersuchungsbericht

Er vergleicht 2 bis 3 Softwarelösungen zur Absicherung von Tablets. Pro Lösung müssen ihre Vor- und Nachteile vorhanden sein. Sowie auch das Fehlen von Einstellungsmöglichkeiten.

Am Ende des Untersuchungsberichts hat ein Vergleich der Systeme zu stehen, aus dem herausgeht, welches, nach der Meinung des Projektteams, eingesetzt werden sollte. Dies muss mit Argumenten bekräftigt werden.

Der Prototyp

Ein Android-Tablet welches auf die Anforderungen eines Beispielunternehmens zugeschnitten ist. Dabei sollen nur bestimmte Aktionen möglich sein.

- Nutzung von 3 bestimmten Apps
- Kein Zugang zu den Einstellungen
- Keine Verbindung mit einem Computer möglich
- Kein Download von Apps, Fotos, Videos, etc.

- Kein Verlassen der gesicherten Umgebung

Abstract

Assignment of Tasks

Our project partner Kapsch provides enterprise grade solutions for major operating systems such as Microsoft Windows 8. Due to the Consumerization trend in industry platforms like Apple iOS and Android OS are on the rise in enterprise applications and leading system integrators like Kapsch are in need to understand those platforms, their applications and limitations. Based on experience and results from existing projects Kapsch has realized that for certain applications or scenarios the Apple iOS platform has its limitations and drawbacks. In particular industrial applications have a need for a rock-solid platform which enables a system integrator like Kapsch to operate a 24/7 application and service. So Kapsch is keen to expand their knowledge and experience towards the Android platform to be able to provide solution platform for industrial applications. Aim of this cooperation is to co-work on a set of different feasibility concepts for realizing such a platform based on Android.

Realisation

The company Kapsch wants an investigation report, which shows which system protection is suitable for Android tablets best for specific customers. In this report, 2-3 different systems to secure a Android tablets are compared. It takes into account the advantages and disadvantages of each system, as well as draw attention to the lack of features. This allows the customer the best possible system for its customers to determine.

Results

In addition to the investigation report a tablet is delivered as a prototype. This tablet is equipped with the protection system, which is apparent from the investigation report as most recommendable.
At the end of the project, there are 2 final results:

The Investigation Report

It compares 2 to 3 software solutions for securing tablets. For each Solution, advantages and disadvantages must be present. As well as the lack of configuration options.

At the end of the investigation report, a comparison of the systems has to stand, out of the proceeds, which should be in accordance with the opinion of the project team used. This must be confirmed with arguments.

The Prototype

An Android tablet that is tailored to the requirements of a model company. It should be possible only certain actions.

- Useof 3 specific apps

- No access to the settings
- No connection to a computer possible
- No downloading of apps, photos, videos, etc...
- Leaving the secure environment not possible

Inhaltsverzeichnis

Einleitung

1.1 Motivation

Im zunehmenden Maße finden sich mobile Endgeräte (z.B. Tablet) als Marketing, Informations- und Imageelement, aber auch als Teil von Prozessen im Retail- als auch im Industrieumfeld wieder. Diese ersetzen oder erweitern bestehende Lösungen (z.b. Kiosk, Digital-Signage, Laptops, Industrie PCs) um interaktive Elemente und unterstützen so Unternehmen in ihren Geschäftsprozessen. Bei der eingesetzten Lösung ist es aus Sicht des Retail und Industrie-Kunden wichtig die Handhabung und das Erlebnis des Gerätes für den Anwender zu erhalten. Die Einsatzgebiete reichen dabei von Tablets in Produktionsprozessen zu Zwecken der Qualitätssicherung (eigene Applikationen mit Dokumentationsfunktionen) bis hin zu Multimedia-Terminals im stationären Handel. Um diese verschiedenen Szenarien realisieren zu können ist es notwendig eine stabile, sichere und wartbare Plattform zu konstruieren, die zum einen die Anwendung in unterschiedlichen Einsatzgebieten ermöglicht und zum anderen für den Betriebsführer leicht bereitzustellen und zu warten ist.

Um den hier angeführten allgemeinen Anforderungen ihrer Kunden gerecht zu werden, benötigt Kapsch die passendste Softwarelösung im Bereich Security für die hierfür in Betracht gezogenen Tablets. Um diese Softwarelösung zu ermitteln, wurde ein Projektteam des fünften Jahrgangs der HTBLVA Spengergasse, Fachbereich Informatik mit der Aufgabe der Eruierung dieser Lösung und der Erstellung eines Konzepts betraut. Das Projekt KTI – Kapsch Tablet Infrastructure wird von dem Projektmanager Philip Steinhäuser geleitet und besteht aus den Teammitgliedern Sebastian Götze, Samuel Hammer, Kaufmann Michael und Konstanze Müller. Das Projektteam steht in engem Kontakt mit dem Project-owner welcher mit den Mitarbeitern Bernhard Bruckner und Jürgen Krammer vertreten ist.

Kapsch unterstützt das Projekt mit diversen Hilfeleistungen sowie Technischer Support, Intellektuelle Hilfe und finanzielle Unterstützung beim Kauf des Tablets, welches nach Beendigung des Projekts wieder in den Besitz von Kapsch übergehen wird.

1.2 Projektpartner

Abbildung 1.1: Kapsch Logo

Die Kapsch Group setzt sich aus 3 Hauptunternehmen zusammen:

- Kapsch TrafficCom
- Kapsch CarrierCom
- Kapsch BusinessCom

Die Kapsch TrafficCom ist internationaler Anbieter von Technologien, Lösungen und Dienstleistungen für den Intelligent Transportation Systems (ITS) Markt.

Die Kapsch CarrierCom ist globaler Lösungspartner für Telco-Carrier und Communication-Provider sowie Railway-Operator.

Unser konkreter Projektpartner ist jedoch die Kapsch BusinessCom. Mit 1.400 Mitarbeitern, einem Umsatz von knapp 300 Millionen Euro und Niederlassungen in Österreich, Tschechien, Slowakei, Ungarn, Rumänien und Polen, positioniert sich das Unternehmen als einer der führenden ICT-Servicepartner in Zentral- und Osteuropa. Kapsch setzt auf Partnerschaften mit führenden Branchengrößen wie Apple, Cisco, Google, HP oder Microsoft, um seinen rund 17.000 Kunden den bestmöglichen Service gewährleisten zu können.

1.3 Ursprungsproblem

Die Aufgabe von Kapsch an das Projektteam ist es, verschiedene Softwarelösungen zum Systemschutz von Android-Tablets zu testen. Die vom Projektowner zum Test gewünschten Softwarelösungen sind.

- MDM
- MDM + Container
- Samsung Knox

Zusätzlich bestünde noch die Möglichkeit einer **Linux Manipulation**, welche jedoch für Kapsch aus rechtlichen Gründen nicht in Frage kommt, da diverse Garantieverletzungen auftreten würden und zusätzlich enorme Kosten anfallen würden aufgrund von Hohen Entwicklungskosten und langen Entwicklungszeiten bis das System einwandfrei funktionieren würde.

Zu diesen Schutzsystemmöglichkeiten soll ein Untersuchungsbericht angefertigt werden. In diesem werden die Ergebnisse festgehalten und verglichen, um das beste System für die Kunden von Kapsch zu ermitteln.

Auf Basis dieses Untersuchungsberichts soll dann die Passendste Softwarelösung ausgewählt werden, welche in ein Konzept eingebaut wird. Ein Teil dieses Konzepts beinhaltet die Implementierung der ausgewählten Softwarelösung auf ein Tablet welches dann als Prototyp deklariert wird.

Der Untersuchungsbericht in Kombination mit dem Prototyp ist das Ergebnis was über den Ausgang dieses Projekts entscheidet.

1.4 Vorgehensweise

1.4.1 Vorbereitung

- Vorbereitungsmeeting mit Kapsch
- Einlesen in Technologie
- Erste Recherchen

1.4.2 Planung

- Projektantrag
- Vorstudie
- PSP (Projektstrukturplan)
- OSP (Objektstrukturplan)
- Timetable bzw. Gantt-Chart
- Lastenheft
- Pflichtenheft
- Projekthandbuch
- Checklist Template
- Research Template
- Untersuchungsbericht
- Diplomarbeit

1.4.3 Durchführung

- Recherche
- Erstellung des Untersuchungsberichtes
- Auswahl der passenden Lösung
- Konzept erstellen
- Ausgewählte Lösung in Konzept einarbeiten
- Konzept teilweise umsetzen
  - Konfiguration der Lösung auf Tablet (Prototyp

1.4.4 Ergebnis

1. Untersuchungsbericht
2. Prototyp

1.4.5 Organigramm

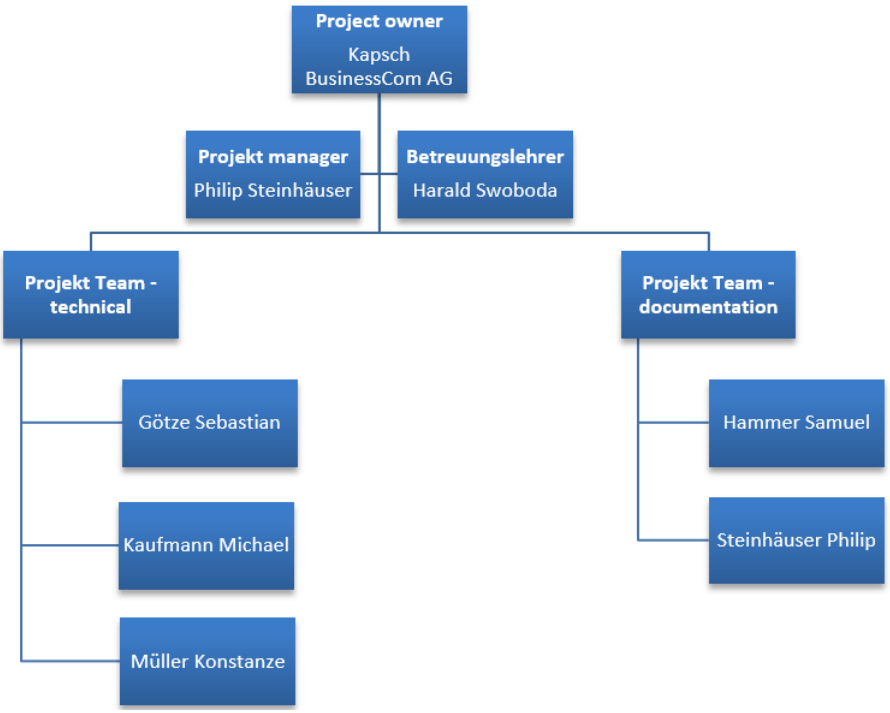

Abbildung 1.2: Projekt-Organigramm

Die Aufgaben des Projekts wurden ab diesem Zeitpunkt so verteilt das das Dokumentationsteam sämtliche Projektmanagementaufgaben und das Technik Team alle Research- und Testungsaufgaben übernimmt.

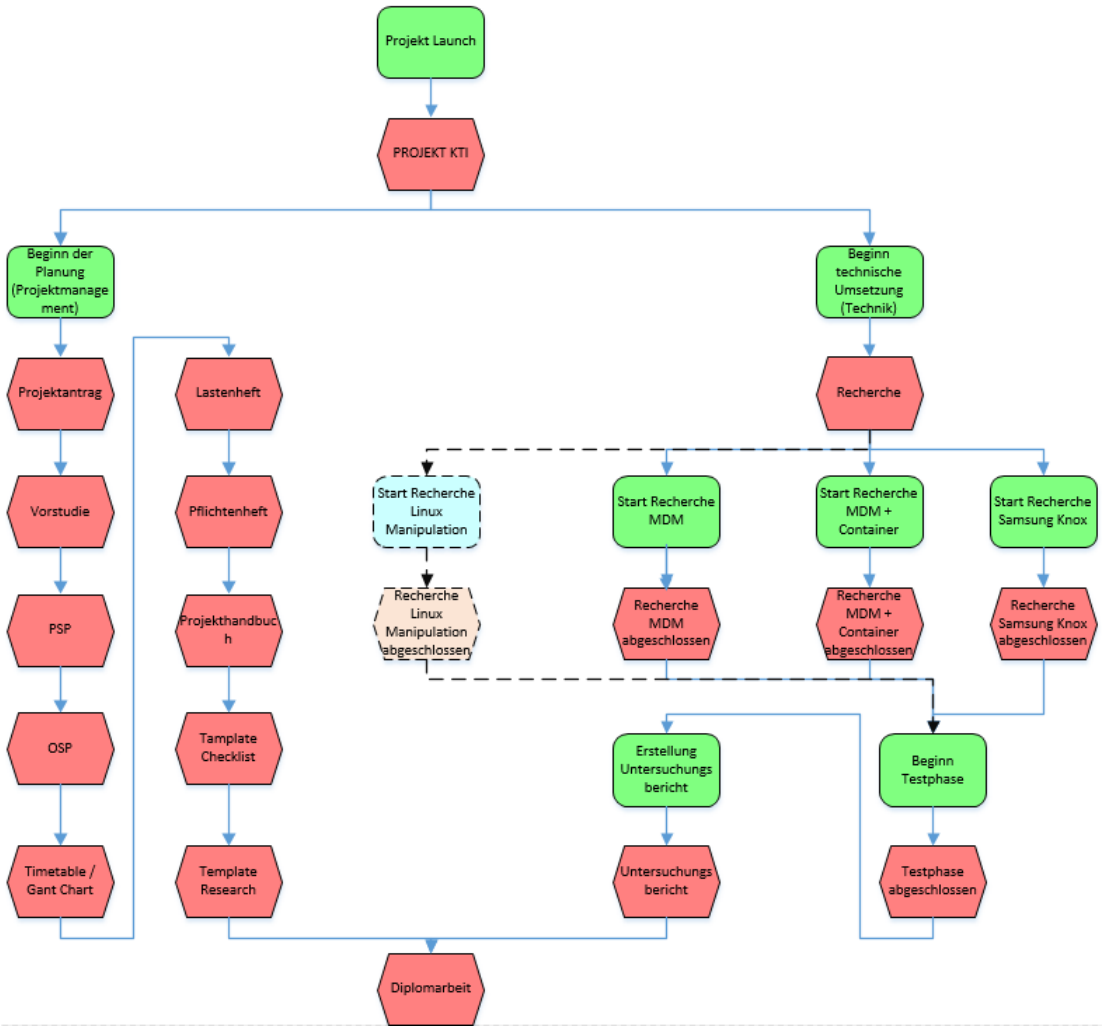

Abbildung 1.3: Aufgabenverteilung - Diagramm

Dieses Ablaufdiagramm zeigt auf der einen Seite die Reihenfolge der erstellten Dokumente, so wie beispielsweise, dass der OSP und der Timetable / Gant Chart auf dem PSP aufbaut, da dieser zuvor erstellt wurde und diese drei Dokumente konsistent sein müssen.

Auf der Technik Seite wiederum sieht man das zuerst eine umfangreiche Recherche nötig ist bzw. war, bevor die Wahl der passendsten Softwarelösung getroffen werden kann / konnte, welche dann schlussendlich für das Konzept und in weiterer Folge für die Konfiguration des Prototyps verwendet werden konnte / wurde.

Wenn man sich unsere Vorgehensweise nun Schritt für Schritt auf die beiden Sub Teams aufgeteilt anschauen würde, würde folgendes Bild entstehen:

#	Dokumentation	Technik
1	• Erstellung des Projektantrages • Erstellung der Vorstudie • Erstellung PSP • Erstellung OSP	-
2	• Erstellung des Gantt-Chart • Erstellung Lastenheft • Erstellung Pflichtenheft	• Beginn der Recherchephase – Recherche Linux Manipulation – Recherche MDM – Recherche MDM+Container – Recherche Samsung Knox
3	• Erstellung Projekthandbuch • Erstellung Research Template • Erstellung Checklist Template	• Festhalten der Rechercheergebnisse in Templates • Recherchephase abgeschlossen – Recherche Linux Manipulation abgeschlossen – Recherche MDM abgeschlossen – Recherche MDM+Container abgeschlossen – Recherche Samsung Knox abgeschlossen
4	• Erstellung Diplomarbeit	• Beginn der Testphase • Erstellung des Untersuchungsberichts
5	• Fertigstellung Diplomarbeit • Fertigstellung Projekthandbuch	• Abschluss der Testphase • Fertigstellung des Untersuchungsberichts

Tabelle 1.1: Projektablauf

Studie

2.1 Android

Android ist, im Gegensatz zu anderen mobilen Betriebssystemen auf dem Markt, eine offene Plattform. Das kommt im Wesentlichen daher, dass Android auf einem open source System, dem Linux Kernel, aufbaut. Durch die freie Verfügbarkeit des Codes, gibt es im Android-Bereich eine große Developer Szene die laufend eigens modifizierte Betriebssysteme (ROMs) und Apps hervorbringt. Ein so hoher Grad an Offenheit birgt jedoch auch gewisse Sicherheitsrisiken und öffnet Angriffsvektoren. Um diese Gefahren besser zu verstehen sollte man sich eingehend mit der Android Architektur und dem darin enthaltenen Sicherheitskonzept befassen.

2.1.1 Architektur

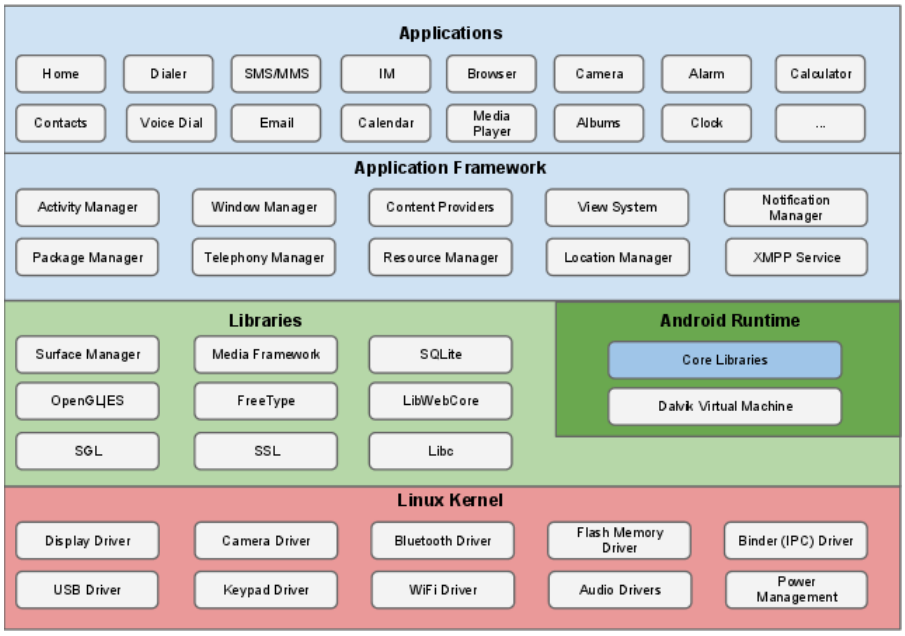

Abbildung 2.1: Android Systemarchitektur

Wie in Abbildung 1 zu sehen ist, bildet der Linux Kernel die unterste Schicht der Architektur. Auf ihm baut das gesamte Betriebssystem mitsamt aller Sicherheitskonzepte auf. Der Kernel stellt die Brücke zwischen Hard- und Software dar und enthält die Treiber für diverse andere Komponenten eines Smartphones wie beispielsweise Modem, GPS Empfänger, Kamera, etc. . .

In der darüber liegende Ebene sind die Android Libaries sowie die Runtime, also die „Dalvik Virtual Machine“ zu finden.
Android Apps sind meist in Java Programmiert und werden jeweils in einer eigenen Virtuellen Maschine ausgeführt, der DVM [1]. Diese Systematik ermöglicht das logische, parallele, unabhängige Ausführen von verschiedenen Apps mit verschiedenen Benutzerrechten. Die Libaries steuern und kontrollieren im Wesentlichen die Funktionen des Kernels und sind für die zentralen Funktionalitäten auf niedriger Ebene zuständig.

Die nächst-höhere Ebene bildet das Application Framework. Hier befinden die Grundfunktionen von Android wie zum Beispiel Telefonie, Location Services, Window Manager, Notification Manager, etc... Auf dieser Ebene werden Entwicklern äußerst umfangreiche APIs für das Entwickeln von Benutzeranwendungen (Apps) zur Verfügung gestellt.

Die oberste Schicht in der Android-Systemarchitektur sind die Applikationen welche der Benutzer selbst installiert und auf den darunterliegenden Schichten aufbaut.

Dieser gesamte „Stack“ (=Stapel) wird in einem ROM zusammengefasst und auf ein Smartphone installiert. Durch die zuvor erwähnte Offenheit des Systems können so durch Modifizieren eines ROM Paketes stark angepasste Versionen von Android entwickelt und verwendet werden.

2.1.2 Security

Android wird als Betriebssystem mittlerweile auf ca 84% aller Smartphones weltweit eingesetzt. [2] Ein Betriebssystem mit einem so großen Marktanteil erfordert ein durchdachtes und ausgereiftes Sicherheitskonzept. Die drei grundlegenden Sicherheitsobjekte sind:

- Schützen von Benutzerdaten
- Schützen von Systemressourcen (inklusive Netzwerk)
- Bereitstellen von Applikationsisolation

Um diese Objekte zu erreichen, stehen eine Reihe von Security-Features zur Verfügung

- Robuste Security auf der OS Ebene durch den Linux Kernel
- Erforderliche Sandbox für alle Apps
- Sichere interprozess Kommunikation
- Application signing
- Application-defined and user-grated permissons

2.1.2.1 System- und Kernel Security

Der Linux Kernel stellt das Fundament von Android dar. Durch dessen langjährige Wartung und Weiterentwicklung hat dieser sich zu einem äußerst sicheren und weit verbreiteten Kernel entwickelt, welcher auch in Sicherheitsempfindlichen Umgebungen eingesetzt wird.

Der Linux Kernel stellt einige der wichtigsten Sicherheitsfunktionen zur Verfügung.

- Ein Benutzer-basiertes Rechtemodell
- Prozessisolation
- Mechanismus für sichere IPC [3]
- Die Möglichkeit potentiel gefährliche Teile des Kernels zu entfernen

[1] Dalvik Virtual Machine
[2] Quelle: http://www.idc.com/prodserv/ smartphone-os-market-share.jsp Stand Q3 2014
[3] Interprozesskommunikation

Im Gegensatz zum normalen Linux, wo mehrere Anwendungen mit demselben Benutzer ausgeführt werden, weist Android jeder Applikation eine eigene User ID zu und führt diese mit ebendiesem User in einem separaten Prozess aus. Diese Vorgehensweise resultiert in einer „Kernel-level Application Sandbox“. Dadurch kann eine App von Haus aus mit keiner anderen App kommunizieren sofern dies nicht explizit erwünscht ist. Standardmäßig können Apps auch nicht auf Dinge wie Standort, die Telefonie-Funktion, etc. zugreifen sondern müssen erst um die Erteilung der entsprechenden Benutzerrechte ansuchen. Die Tatsache dass diese Sandbox-Systematik auf Jahrzehnte alter UNIX Technologie aufbaut, macht das System leicht überwachbar und transparent, jedoch gleichzeitig effizient.

Speicherfehler führen in vielen Betriebssystemen zu groben Sicherheitsrisiken und sind in der Lage die Sicherheit eines Gerätes komplett zu kompromittieren. Bei Android kann ein solcher Speicherfehler durch das Sandboxing aller Applikationen auf OS-Level keinen allzu großen Schaden anrichten, da eventueller Schadcode nur im Kontext einer bestimmten App, mit eingeschränkten Benutzerrechten ausgeführt werden kann.

Selbstverständlich ist auch die Applikations-Sandbox nicht zu 100% sicher, möchte man diese aber knacken, so muss man es schaffen den gesamten Linux Kernel zu kompromittieren.

2.1.3 Angriffsvektoren

Angriffsvektoren sind Pfade um ein Security Konzept zu durchdringen und so Zugriff auf gesicherte Ebenen eines Systems zu erlangen. Gerade bei einem Mobilen Betriebssystem wie Android, ist es von besonderer Wichtigkeit diese Gefahrenquellen zu kennen und soweit wie möglich zu eliminieren da Geräte welche auf Android laufen einen Großteil der Zeit angeschaltet und mit dem Internet verbunden sind.

Ein weiterer Grund warum besonders mobile Endgeräte Ziel solcher Attacken sind ist die Tatsache dass Daten auf einem Smartphone oder Tablet meist aktuell sind. So werden beispielsweise neue E-Mails immer sofort via push mail heruntergeladen und gespeichert. Auch sogenannte TAN Codes, welche sich in den letzten Jahren als Autorisierung-Methode für online Überweisungen etabliert haben und via SMS übertragen werden, machen Smartphones zu einem brauchbaren Ziel für Hacker.

2.1.3.1 Social Engineering

Social Engineering ist eine Methode unautorisiert an Daten zu gelangen, ohne dabei Technische Änderungen am System vorzunehmen. Hierbei wird versucht den User mit Hilfe von Falschen Informationen dazu zu bringen das System zu verändern oder eine schädliche Datei herunter zu laden und dem Angreifer so selbst den Weg zu ebnen. Ein Beispiel für Social Engineering ist der sogenannte „ILOVEYOU Virus“. Dieser wurde via E-Mail verschickt und gab vor ein Liebesbrief zu sein, sodass der Benutzer ihn öffnet und selbst ausführt. Gegen das Prinzip des Social Engineerings gibt es kein brauchbares Gegenmittel, da es einzig und allein auf den Entscheidungen des Users aufbaut.

2.1.3.2 Drive-by Exploitation

Diese Angriffsmethode macht sich Bugs in vorhandener Software zu Nutzen um schadhaften Code auf dem Device auszuführen. Mit Hilfe solcher Bugs gelingt es dem Angreifer Code über beispielsweise eine Website auf das Gerät des Opfers herunterzuladen und dann auf dem Zielsystem auszuführen. Drive-by Exploitation ist vor allem auf Desktop PC Systemen weit verbreitet. Durch diverse Sicherheitsmechanismen von Android wie das Sandboxing der Apps oder die DVM (Dalvik Virtual Machine), ist es auf solchen Geräten

wesentlich schwieriger schadhaften Code auszuführen und so das System nachhaltig zu beschädigen. Nichtsdestotrotz gibt aus auch auf Android eine große Anzahl an Apps welche native Software-Bibliotheken verwenden und so angreifbar gegen Drive-by Exploitation sind.

2.1.3.3 Phishing

Phishing Attacken werden benutzt um sensible Informationen wie Login Daten abzugreifen. Der Angreifer gibt sich dabei als offizielle Instanz aus und bewegt den Nutzer dazu seine Daten preiszugeben. Dies geschieht beispielsweise über eine gefälschte Website, welche gleich aussieht wie die echte Website, nur dass die eingegebenen Daten direkt an den Angreifer weitergeleitet werden. Phishing beinhaltet einige Aspekte des Social Engineerings, daher gibt es auch hier keine wirklich gute Möglichkeit solche Attacken zu verhindern. Auf Mobilen Geräten ist die Situation allerdings ein wenig anders, da es für die meisten online Services eine eigene App gibt, welche nur über den offiziellen App Store heruntergeladen werden kann. Daher verwenden nur wenige User die Website und laufen damit nicht Gefahr Ziel einer Phishing Attacke zu werden.

Varianten

3.1 Linux Manipulation

3.1.1 Allgemein

Dieser Teil befasst sich mit der Veränderung des Grundsystems eines jeden Android Geräts. Dieses Grundsystem basiert auf dem Open Source Betriebssystem Linux und existiert dabei in einer für Mobilgeräte angepassten Form. In seiner Standardimplementation bietet es zwar einige Funktion zur Erhöhung der Gerätesicherheit, jedoch nicht genügend, um es in einem betrieblichen Umfeld einzusetzen. Da Linux ein Open Source System ist, darf der Source Code von jedem angesehen und nach eigenen Wünschen verändert werden. Und genau hier setzt die Methode der Linuxmanipulation an. In dem man den Source Code so verändert, dass bestimmte Teile des Betriebssystems unzugänglich gemacht oder verschlüsselt werden, ist es möglich den späteren Benutzer vor unabsichtlichen Änderungen am System zu bewahren, welche den reibungslosen Betrieb stören könnten. Das bedeutet, dass dadurch ein vollkommen an die Bedürfnisse des Kunden angepasstes Betriebssystem möglich wäre. Um sich einen besseren Eindruck davon zu verschaffen, wie diese Manipulationen letztendlich aussehen, lohnt es sich auf die diversen frei am Markt erhältlichen Derivate zu werfen. Bekannte Beispiele dafür wären:

- CyanogenMod
- Android AOSP
- Paranoid Android
- Dirty Unicorns
- etc.

Zwar sind diese nicht mit securitytechnischen Absichten entwickelt worden, aber sie zeigen trotzdem auf was mit einer Menge an Entwicklungsaufwand möglich ist. Da aber durch die tiefgreifenden Eingriffe in das System eines Android Gerätes auch die Garantieansprüche verloren gehen, ist das Projektteam zu einem schnellen Fazit zu kommen.

3.1.2 Schlussfolgerung

Die Methode der Linuxmanipulation ist für die Zwecke der Kapsch leider absolut nicht einsetzbar dar sich bei ihr gewisse Konflikte bezüglich Garantieanspruch und Kosten ergeben. Zwar wären über diese Methode sämtliche Anforderungen an das Endprodukt erfüllbar, jedoch bedarf es dazu eines so großen Entwicklungsaufwands, dass dieser sich in einem dermaßen hohen Endkundenpreis niederschlagen würde, welcher von kaum einem Unternehmen zu bezahlen wäre. Denn nicht nur die Beschäftigung einer Vielzahl von Entwicklern über einen langen Zeitraum hinweg, sondern auch der anfallende Support für das Produkt würde sich selbst für ein großes Unternehmen wie Kapsch nicht rentieren. Ein weiteres großes Manko dieser Methode ist die verfallende Garantie für die Hardware. Egal mit

welchen Android Tablet diese Software ausgeliefert werden würde, sobald eine Veränderung der darauf vorinstallierten Software stattfindet, gehen sämtliche Garantieansprüche an den Hersteller verloren. Bei der geplanten Menge an ausgelieferten Geräten durch die Kapsch, wäre dies nicht vertretbar. Für einen industriellen Einsatzzweck ist die Variante daher absolut nicht geeignet und der damit einhergehende Aufwand würde sich nur durch einen extrem hohen Verkaufspreis ausgleichen lassen. Somit bleibt dem Projektteam nichts anderes zu sagen, als, dass diese Variante nicht passend für die Absichten der Kapsch ist.

3.2 Mobile Device Management (MDM)

3.2.1 Allgemein

Die folgenden Zeilen beschäftigen sich mit den Einsatz von Mobile Device Management Systemen als Betriebsplattform für potentielle Kunden der Firma Kapsch. Durchgeführt wurden alle Untersuchungen am bereits am Markt etablierten MDM-System MobileIron. Es wird beleuchtet was der MDM-Standard ist, welche Funktionen für den alltäglichen Gebrauch unumgänglich sind und wie diese im System realisiert sind. Ein besonders wichtiger Punkt hierbei ist auch das Aufzeigen von nicht vorhandenen Funktionen, die jedoch für das Unternehmen Kapsch von fundamentaler Wichtigkeit sind. Des Weiteren wird auf die Bedienbarkeit und die Komplexität der Installation eingegangen und inwiefern dies für den Projektauftraggeber und dessen Kunden relevant ist. Abschließend wird ein Statement abgegeben, ob bzw. wie es möglich ist diese Form von System für die von Kapsch gedachten Zwecke einzusetzen.

3.2.2 Mobile Device Management Standard

MDM bezeichnet einen von der Open Mobile Alliance (OMA) festgelegten industriellen Standard zur Verwaltung mobiler Endgeräte wie zum Beispiel Smartphones, Tablets oder Laptops. Es dient dazu die allgemeine Verwaltung einer Vielzahl von Geräten zu erleichtern und somit Zeit und Kosten zu sparen. Die mobile Hardware kann dabei vom Unternehmen zur Verfügung gestellt werden oder, sofern mit dem Mitarbeiter abgesprochen, von diesem selbst mitgebracht werden. „Bring Your Own Device“ (BYOD) nennt sich dieser Ansatz. Dieser Standard wird in der Software verschiedenster Hersteller implementiert, welche dann dieses Komplettsystem verschiedenen Unternehmen zur Verwaltung ihrer Geräte anbieten. Beispiele dafür sind.

- MobileIron
- Samsung EMM
- Cisco Meraki
- MaaS360
- AirWatch
- etc.

Bestandteile dieser Softwarelösung sind eine Serverkomponente und die verschiedenen mobilen Clients. Der Server dient dabei dazu die Konfigurationen und Statistiken für die Geräte zu halten und zu verwalten. Er sendet auch, auf dem MDM-Standard basierende, Management Kommandos an die Clients aus, wenn sich ein Parameter in deren Konfiguration verändert hat. War es am Anfang der MDM-Systeme noch notwendig das Gerät dazu physisch mit dem Server zu verbinden, geschieht dies heute vollautomatisch über Netzwerkverbindungen. Die implementierten Funktionen können zum Beispiel eine over-the-air (OTA) Verteilung von Applikationen, Daten oder Konfigurationen sein. So braucht ein Administrator nicht auf 100 Geräten das Wifi-Netzwerk einrichten, sondern kann per Knopfdruck diese Konfiguration auf alle im System registrierten Geräte verteilen. Auch in Punkto Sicherheit bieten MDM-Systeme einige Möglichkeiten und deshalb sind sie so interessant für die Zukunftspläne der Firma Kapsch. So bieten diese Systeme die Möglichkeit Passwortrichtlinien zu setzen oder sogar ganze Teile des Betriebssystems zu sperren, damit diese für den Benutzer nicht zugänglich sind. Dadurch soll die Anfälligkeit für Fehler im Berufsumfeld gesenkt und ein ordentlicher Arbeitsablauf genehmigt werden.

Name	MobileIron EMM
Hersteller	MobileIron, 415 East Middlefield Road, Mountain View, CA 94043
Version	7.5.0
Datum	27.01.2015
Preis	/
Website	http://www.mobileiron.com/
Dokumentation	https://support.mobileiron.com/eval/

Tabelle 3.1: MDM Übersicht

3.2.3 Informationen der getesteten Software

3.2.4 Installation

Die Installation von MobileIron gestaltet sich relativ einfach, wobei doch einige wichtige Dinge zu beachten sind. Nach dem Download einer Datei aus dem Online-Zugangsportal von MobileIron, kann über diese das Betriebssystem installiert werden. Diese gestaltet sich für erfahrene Nutzer sehr einfach, allerdings ist auf folgende Dinge acht zu geben:
Folgende Daten müssen bereit stehen bzw. eingerichtet werden:

- Lizensierungsinformationen (Firma, Kontaktperson, E-Mail)
- IP-Adresse
- Externer Hostname **(Sehr wichtig, weil die mobilen Geräte den Server von außerhalb erreichen müssen)**
- Command Line Interface Passwort
- Administratorname und -passwort
- Mind. 1 physikalisches Interface
- Subnetzmaske
- Default Gateway
- Mind. 1 zu erreichender DNS-Server
- Wahlweise
  - SSH-Zugriff
  - Telnet Zugriff
  - NTP

Ist die Einrichtung erfolgt, kann man das System nach einem Neustart bereits einsetzen. Während dem Evaluierungsprozess sind dem Projektteam allerdings einige wichtige Details aufgefallen. Ein funktionierender externer Hostname ist von höchster Wichtigkeit, weil ohne ihn zwar die Einrichtung der Software auf den mobilen Endgeräten funktioniert, leider jedoch die Verbreitung von Konfigurationen versagt. Da jedoch 99 Prozent aller modernen Unternehmen über solche Möglichkeiten verfügen sollten, dürfte dies im realen Betrieb weniger problematisch ausfallen. Hervorzuheben ist hierbei die hervorragende Dokumentation, die MobileIron für den Installationsprozess zur Verfügung stellt. Auf deren Website findet sich eine Sammlung an Dokumenten, welche den Administrator am Anfang zwar überwältigen könnten, aber sich als eine schnell zu durchforstende Sammlung an bebilderten Skripten zur Einrichtung sämtlicher Funktionen herausstellen. Generell lässt sich die Webplattform, welche MobileIron hier zur Verfügung stellt, gut bedienen und bietet Informationen zu den verschiedenen Implementierungsszenarien und Komponenten des Systems. So findet man sich nach kurzer Zeit bereits relativ gut zurecht und weiß wo man suchen muss, um die benötigte Information zu finden.

3.2.5 Features

3.2.5.1 Key-Features

In diesem Teil werden die wichtigsten von MobileIron gebotenen Features beleuchtet und erklärt. Die folgende Liste stellt die wichtigsten Funktionen dar, welche während des Evaluierungsprozesses festgestellt werden konnten:

- MDM System
- Management von verschieden vielen Geräten
  - Jeder Gerätetyp ist möglich, egal ob Smartphone, Tablet oder Laptop
  - Lokalisierung sämtlicher eingebundener Geräte
  - Leicht aufzusetzen und zu verwenden
    - Installation einer einzigen App ist notwendig, um das Gerät in die Plattform einzubinden
  - Verwendbar ab dem ersten Gerät
  - Mehrsprachig
  - Geringe Wartungskosten
- Konfiguration der eingebundenen Endgeräte
  - Vorkonfiguration von Email-Konten und sonstigem (Wifi, VPN, etc.)
    - Der Angestellte muss dies nicht selbst erledigen
    - Keine Chance einer Fehlkonfiguration
    - Verteilbar auf hunderte Geräte innerhalb von Sekunden
- Statistiken
  - Verfügbare Statistiken
    - Gerätestatus
    - Kompromitierungsstand
    - Betriebssystem
    - Betriebssystemversion
    - Zugehörigkeit (gehört dem Unternehmen oder dem Angestellten)
    - Netzbetreiber (3, A1, Telering, etc...)
    - Registrierungszustand
  - Logging von Events

Device Actions	**App -**	**Policy -**	**Space -**	**Status -**
Register	Install	Activate	Add Space	Not started
Wipe	Uninstall	Modify	Remove Space	In progress
Lock	Set setting	Deactivate	Change Space Priorization	Completed successfully
Retire	Unset setting		Assign Space Admin	Failed
			Delete Space Admin	

Tabelle 3.2: MDM Event Logs

- Policies
  - Dienen zur Erhöhung der Sicherheit von registrierten Mobilgeräten
  - Blockieren von Systemteilen oder Einstellungen

    - Zum Beispiel: Der Benutzer kann das WLAN oder GPS nicht mehr ausschalten.
    - Diese Funktion wird benötigt, wenn die Chance besteht, dass der Anwender durch gewollte oder ungewollte Aktionen das Gerät in einen unbenutzbaren Zustand bringt.
  - Passwortpflicht
    - Der User ist dazu gezwungen, ein Passwort nach Unternehmensrichtlinien zum Sperren und Entsperren seines Gerätes zu setzen.
  - Globaler Proxy
    - Sämtlicher Netzwerkverkehr wird durch einen Proxy-Server des Unternehmens geleitet, welche dazu dient ungewollten Websiteinhalte zu filtern oder um Unternehmensdaten vor dem Verlassen des Firmennetzwerks zu schützen.
  - Kiosk-Modus
    - Das Gerät wird in einen Zustand versetzt, in dem nur mehr das Benutzen einer einzelnen Applikation möglich ist.
  - Applikationen
    - Erlauben von spezifischen Apps
    - Verbieten –
    - Erfordern –

3.2.5.2 Zusatzfeatures

MobileIron bietet die Möglichkeit seine Vielfalt an Features zu erweitern, indem man es an eine sogenannte „Standalone Sentry“ anbindet. Diese ermöglicht es dem Administrator des MDM-Systems die von MobileIron entwickelten Applikationen auf allen Geräten zu installieren. Diese Applikationen dienen dazu den E-Mail-Verkehr, das Webbrowsing, die Installation von Applikationen und den Zugang zu Unternehmensdokumenten zu sichern. Sie heißen:

- Apps@Work
- Docs@Work
- Web@Work

Nachdem sich das Projektteam in diese vertieft hatte, hat es erkannt, dass diese Applikationen eine sogenannte Containertechnologie einsetzen und somit nicht Bestandteil eines standardmäßigen MDM-Systems sind. Daher werden diese in einem anderen Teil des Evaluierungsprozesses behandelt, welcher „MDM+Container“ heißt. Eine weitere Zusatzfunktion, die MobileIron bietet, sind „ActiveSync Policies“. Diese stellen dabei einfach nur durchgehend upgedatete Policies dar, welche bei einer Verletzung sofort Alarm schlagen.

3.3 MDM + Container

3.3.1 MobileIron Containertechnologie

MobileIron Content Mangement (MCM) ist die Containertechnologie der gleichnamigen Firma MobileIron. Die Containertechnologie basiert auf dem MobileIron MDM System, welches in diesem Projekt separat analysiert wird.
Der Hauptnutzen dieser bestimmten Containertechnolgie ist die Verwaltung von Benutzerressourcen. Ursprungsansatz der Entwicklung ist der fortschreitende Trend zur Arbeit auf mobilen Geräten wie Tablets. Das MCM System ist auf folgende Module aufgeteilt: Docs@Work, Apps@Work, Web@Work.

3.3.2 Informationen der getesteten Software

Name	MobileIron Mobile@Work
Hersteller	MobileIron, 415 East Middlefield Road, Mountain View, CA 94043
Version	
Datum	26.01.2015
Preis	/
Website	https://www.mobileiron.com/en/products/product-overview
Dokumentation	https://support.mobileiron.com/eval/

Tabelle 3.3: MDM + Container Übersicht

3.3.3 Installation

Die Konfiguration der Containertechnologie geschieht über ein Webinterface, welches man ebenso für das Standard MDM System von MobileIron verwendet. Dokumentation wie zum Beispiel ein Benutzerhandbuch über die Konfigurationsschritte von MobileIron direkt ist zwar vorhanden, wobei dieses erst größtenteils während unserer Projektzeit erschienen ist. Somit versuchte das Projektteam zunächst dies alleine zu bewältigen und scheiterten, da die gesamte Benutzeroberfläche nur gering selbsterklärend ist. Mittels dem Benutzerhandbuch gelang es uns nach mehreren Versuchen die Containertechnologien Web@Work und Apps@Work funktionstüchtig zu bekommen. Für mehr Aufwand sorgten Zertifikate und Konfigurationsdateien, welche man für die Konfiguration benötigt und nur sehr oberflächig in den bereitgestellten Dokumenten beschrieben worden sind. Weiters beschäftigte uns die nicht konstante und auffallend lange Synchronisationszeit zwischen dem Server und dem Client Device. Der Grund dafür war für uns nicht klar ersichtlich und hätte für eine genaue Analyse zu viel Zeit in Anspruch genommen. Jedoch deutet vieles darauf hin, dass die Ursache in unserem selbstaufgebauten Testnetzwerk entstanden ist, welches sehr klein und minimalistisch gehalten wurde.

3.3.4 End User Products

3.3.4.1 Docs@Work

Docs@Work stellt den Usern unternehmensinternen Content für deren tägliche Arbeit zur Verfügung. Dies basiert auf einem Cloud-Content-Management-System und wird am Gerät als eigene App dargestellt. In erster Linie gewährt die Containertechnologie einen sicheren Verbindungsaufbau zu „Content Repositories“. Unter „Content Repositories“ versteht man im Allgemeinen gewöhnliche unternehmensinterne Fileserver (SharePoint), welche

als Netzwerkressourcen zur Verfügung stehen. Besonders dabei ist, dass auch empfangende Daten via Email in dem gleichen Ausmaß wie normale „Content Repositories“ den Sicherheitsrichtlinien unterworfen sind. Dem User steht die Möglichkeit des Downloads dieser Dokumente offen. Diese Funktion, welche offiziell unter dem Namen „Secure Email Attachment“ geläufig ist, basiert auf MobileIron AppContent. Diese Technologie stellt eine konsistente, sichere Umgebung auf dem Android Gerät zur Verfügung. D.h. Unternehmensdaten welche auf dem Mitarbeiter-Gerät abrufbar sind, sind verschlüsselt.

Aus Administrator-Sicht gewährt Docs@Work neben der zentralen Verwaltung von Ressourcen für einzelne User ebenfalls die Möglichkeit bei nicht gewünschten Tätigkeiten eines Mitarbeiters seine Berechtigung einzuschränken. Administratoren können bestimmte Geräte von Mitarbeitern in Quarantäne verschieben bzw. diese ganz entfernen. Die betroffenen Geräte sind dann nicht mehr in der Lage sich mit dem Server zu verbinden.

Als gewissen Vorteil dieser Technologie kann man den Wegfall des üblichen zusätzlichen VPN auf den Geräten bezeichnen. Die Bedingung ist für den Anwender kompakter und schneller. Sofern der User Zugriff auf den „Content Repository“ hat, gilt die Regelung des Single-Sign-On. Dies gewährt einer nahezu einmaligen Anmeldung per einem User Account und garantiert Zugriff auf alle verknüpften Anwendungen ohne weitere Log-in Session.

Die oben genannte Secure Email Attachment Funktion ist nach Informationen von der offiziellen MobileIron Homepage bis zum heutigen Stand, nur mittels Email Applikationen namens Divide und Email+ funktionstüchtig.

Ebenso gibt es nur ausgewählte „Content Repositories“, welche Docs@Work unterstützen. Zu diesen zählen:

- Microsoft Sharepoint 2007/2010/2013
- CIFS Windows 2008 R2 SPI
- CIFS Samba CentOS 6.2
- Apache-based WebDAV content repositories
- IIS-based WebDAV content repositories

3.3.4.2 Web@Work

Web@Work garantiert Unternehmen einen sicheren, mitunter auch beschränkten Internetzugriff ihrer Mitarbeiter. Es basiert auf zwei Technologien namens AppTunnel und MobileIron Sentry, welche bei Nutzung von Web@Work auch konfiguriert werden müssen. Das Zusammenspiel dieser zwei Technologien gewährt eine Zugangsbeschränkung/Kontrolle sowie den verschlüsselten Datenaustausch. Die Administratoren sind in der Lage gewisse, in den Augen des Unternehmens wichtige Websiten für die Mitarbeiter frei zu schalten und somit den Besuch dieser zu erlauben. Unter diesen Websiten werden auch interne Unternehmens Websiten verstanden. Daten wie der Zwischenspeicher des Browsers, Cookies, die Web History als auch Daten von anderen Websiten werden alle verschlüsselt übertragen. Sofern ein Android-Gerät eines Mitarbeiters den Zugangsbestimmungen nicht mehr entspricht, werden all diese Daten aus Sicherheitszwecken gelöscht.

Laut der Dokumentation von MobileIron ist es möglich die User-/Geräte-Verwaltung dieses Systems mit dem Enterprise Directory des Unternehmens zu koppeln. Somit ist das Aktivieren und Zulassen von Websiten basierend auf bestimmten Gruppen von Usern möglich.

Die Gegenmaßnahme von MobileIron zur Prävention von DLP (Data Loss Prevention) ist die Deaktivierung vom Erstellen von Screenshots des Users.

Beim Einsatz dieser Technologie steht der Vorteil bezüglich des VPN im Vordergrund. Um Mitarbeitern einen sicheren, abgeschirmten Zugriff auf Webressourcen zu gewähren war bisher eine mögliche Variante, eine VPN Verbindung einzurichten. Web@Work ersetzt das VPN und gewährt weitere Möglichkeiten wie bereits oben beschrieben.

3.3.4.3 Apps@Work

Diese Technologie stellt dem Benutzer des Devices die benötigten Apps zur Verfügung. Dabei kann der User selbst nicht entscheiden, welche Apps er installiert, sondern der IT Administrator. Somit wird das Verwenden des Devices für nicht unternehmensinterne Angelegenheiten verhinder. Der IT Administrator deklariert alle Apps, die laut Unternehmensführung genehmigt sind. Alle anderen Apps welche nicht angeführt sind, sind somit nicht für die Installation zulässig. Diese Technologie basiert auf AppConnect und AppTunnel.

Die erstangeführte Technologie AppConnect ist zuständig für die Implementierung eines Containers um die eigentliche App. Das Resultat daraus ist ein Schutz gegen ‚data-at-rest' Daten des Devices bzw. Apps. Die vorhandenen Daten werden verschlüsselt und vor unberechtigten Zugriff geschützt. Auf dem jeweiligen Device sind alle App Container der Apps verbunden und kommunizieren miteinander. Dabei werden Informationen, wie zum Beispiel Richtlinien und Single sign-on Daten, ausgetauscht.
AppTunnel ist für die Sicherheit und den Schutz der data-in-motion Daten zuständig. Diese Technologie stellt sicher, dass die einzelnen Container um die Apps vom restlichen System abgeschirmt sind und keine Verbindung von außen über das Android Basissystem auf die Container stattfinden kann. Die Verbindung wird einzig und alleine zu autorisierten Apps, Usern und Devices aufgebaut. Die ‚certificate-based session authentication' verhindert man-in-the-middle Attacken.

Bei der Benutzung von Apps@Work unterscheidet man zwischen 3 verschiedenen Möglichkeiten Apps in das System einzubinden und dem User bereitzustellen.

Die wohl geläufigste Art Apps auf ein Android Device zu implementieren ist das Herunterladen mittels dem Google Play Store.

Der theoretische Ablauf ist folgender: Der IT Administrator tätigt einen Vorschlag für eine ausgewählt App des Google Play Store und hat somit eine Freigabe für den Download dieser App. In unserem Anwendungsfall trifft diese Lösung nicht zu, da der Google Play Store einzig und allein bei Verwendung eines Google Accounts auf dem Device funktioniert. Die Verwendung eines Google Accounts ist in unserem Fall jedoch nicht zielführend, da weitere Sicherheitsprobleme auftreten würden. Daraus ergibt sich die zweite Variante zum Management der auf den Devices installierten Apps anzuwenden. Die In-house Apps basieren hauptsächlich auf selbst entwickelten Applikationen, welche zum Bespiel das Unternehmen selbst in Auftrag gegeben hat. Für die Freischaltung zum Download dieser durch die User benötigt man die APK der App, welcher der IT Administrator zunächst selbst hochladen muss. Weiters kann man mittels Benutzung von In-house Apps ebenfalls zum Teil Apps installieren, welche im Google Play Store verfügbar sind. Voraussetzung dafür ist der rechtmäßige Besitz der APK Files der Apps.

3.4 Samsung Knox

Dieser Abschnitt beschäftigt sich mit dem Einsatz von Samsung Knox als System für potentielle Kunden von der Firma Kapsch. Hier findet sich sowohl Informationen zu Samsung Knox im Allgemeinen, als auch darüber, welche der von der Firma Kapsch erwünschten Features mit Samsung Knox realisierbar sind und welche nicht. Außerdem wird noch auf die Bedienbarkeit und die Installation eingegangen.

3.4.1 Informationen der getesteten Software

Name	Samsung Knox
Hersteller	Samsung
Version	2.1
Datum	08.07.2014
Preis	• Express(E): Free but limited to 250 seats • Premium(P): USD $1 MSRP per device/month • Workspace(W): USD $3.60 MSRP per device/month
Website	https://www.samsungknox.com/de
Dokumentation	

Tabelle 3.4: Samsung Knox Übersicht

Samsung Knox ist eine Sammlung Business-orientierter Sicherheitsfunktionen für Android-Geräte von Samsung. Das System, welches auf SE for Android basiert, ist speziell auf die Bedürfnisse von Unternehmen in Hinsicht auf die Sicherheit der Endgeräte ausgerichtet.

3.4.2 Samsung Knox Bestandteile

Um wirklich erklären zu können was Samsung Knox ist, muss man es zuerst einmal in seine 3 Hauptbestandteile unterteilen.

3.4.2.1 Platform Security

In jedem Samsung Gerät ist ein physischer Hardware Chip eingebaut, welcher automatisch einen höheren Schutz für Samsung Geräte bieten soll. Mittels diesem Chip ist es Samsung möglich, bis in die tiefste Software-Schicht eines auf Android basierenden Geräts einzugreifen.

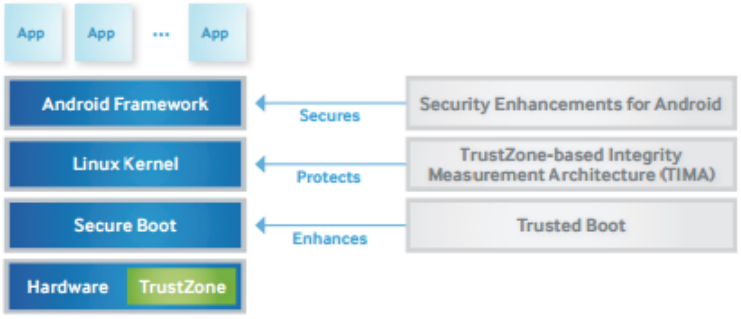

Abbildung 3.1: Samsung Knox Platform Security

In der Grafik links sieht man den groben Aufbau eines Android Systems.In der Grafik rechts sieht man die Technologien, mit deren Hilfe es Samsung möglich ist in die Schichten einzugreifen.

3.4.2.1.1 Kerntechnologien der Platform Security

- **SE for Android**
  - **Funktionsweise**
    SE for Android basiert auf SELinux-Technologie und definiert die Zugriffskontrolle auf Linux-Ebene. Mandatory Access Control (MAC) und Discretionary Access Control (DAC) überwachen und verwalten welche Dateien und Apps auf das System des Geräts zugreifen können.
  - **Sicherheit**
    SE for Android erzwingt MAC, wobei Apps nur genau die Rechte zugewiesen erhalten, die sie für den Zugriff auf das System des Geräts benötigen. Sollte ein böswilliger Benutzer oder eine bösartige App also Zugriff auf das Gerät erhalten, dann betrifft der Schaden immer nur einen bestimmten Bereich, während die restlichen Bereiche des Geräts geschützt bleiben.

    Wenn SE for Android ausgelöst wird, sendet es eine Präventionsinformationsmeldung an den Benutzer des Geräts, die in der Informationsleiste erscheint. Die Präventionsinformationsmeldung gibt an, welche Anwendung versucht hat, auf Daten des Geräts zuzugreifen, und sie wird die Deinstallation der betreffenden Anwendung empfehlen. Beachten Sie dabei jedoch, dass SE for Android auch von autorisierten Anwendungen ausgelöst werden kann. Dies kann vorkommen, wenn die App einen von dem in der SE for Android-Richtlinie vorgegebenen Pfad verwendet. Wenn dies der Fall ist, sollten Sie die Datei der Sicherheitsrichtlinie entsprechend aktualisieren.
- **ARM TrustZone-Based Integrity Measurement Architecture (TIMA)**
  - **Funktionsweise**
    TIMA schützt Ihr Gerät auf zwei verschiedene Weisen. Erstens prüft sie in regelmäßigen Abständen, ob der Kernel des Geräts geändert wurde, indem sie den gegenwärtigen Zustand mit dem Original-Kernel vergleicht. Zweitens authentifiziert TIMA Kernel-Module, wenn diese geladen werden, sodass Geräte nie ungeschützt sind.
  - **Sicherheit**
    Die TIMA TrustZone ist ein manipulationssicherer Sektor des ARM-Prozessors in Ihrem Gerät. Sie authentifiziert und verifiziert den Linux-Kernel über regelmäßige Messungen. Wenn TIMA for Android ausgelöst wird, sendet es eine Ermittlungsinformationsmeldung an den Benutzer, die in der Informationsleiste erscheint. In der Meldung werden Sie normalerweise zum Neustart des Geräts aufgefordert.
- **Secure Boot und Trusted Boot**
  - **Funktionsweise**
    Secure Boot verhindert, dass unbefugte Bootloader und Kernels in das Gerät geladen werden. Dies bedeutet, dass Ihr Gerät nicht manipuliert wurde und der KNOX-Container geladen werden kann.

    Trusted Boot vergleicht den Bootloader und den Kernel des Betriebssystems mit den originalen Werksversionen. Dies wird dadurch erreicht, dass die Originaldaten des Geräts aufgezeichnet werden und das Gerät permanent beim Systemstart mit diesen gegengeprüft wird, um sicherzustellen, dass sich diese Daten nicht geändert haben.

- **S**icherheit
  Es gibt vier Bootloader auf dem Gerät. Jeder Bootloader prüft die Gültigkeit des vorhergehenden Bootloaders oder Kernels. Trusted Boot sichert diese Bootloader. Die Funktion ist in der ARM TrustZone, einem manipulationssicheren Sektor des ARM-Prozessors, eingebettet. Trusted Boot verwendet kryptografische Schlüssel, um sicherzustellen, dass die Messungen am Gerät dem Original entsprechen. Diese Schlüssel werden erst von Trusted Boot freigegeben, wenn SE for Android bestätigt, dass genehmigte Firmware auf dem Gerät ausgeführt wird.

3.4.2.2 Application Security

3.4.2.2.1 Application Containers Container sind quasi ein Android im Android und bezeichnet einen gesicherten, separaten Bereich (Container) auf dem Gerät. Dieser Bereich hat einen eigenen Homescreen, eigene Anwendungen und eigene Daten. Diese Funktion ist damit vergleichbar mit einer Art Dual-Boot-Variante. Anwendungen außerhalb des Containers haben dabei keinerlei Zugriff auf die Daten oder Prozesse innerhalb des Containers. Anwendungen innerhalb des Containers haben grundsätzlich keinen Zugriff auf Daten außerhalb des Containers.

Mittels Richtlinienkonfiguration kann der IT-Verwalter des Gerätes einen read-only (nur lesen) Zugriff für bestimmte Anwendungen im Container auf Daten außerhalb des Containers einrichten. Umgekehrt besteht diese Möglichkeit allerdings nicht. Daten innerhalb des Containers werden dabei mit einem Verschlüsselungsalgorithmus und einem AES-256 Bit Schlüssel geschützt. Ein Zugriff ist erst nach Eingabe eines Passwortes möglich.

3.4.2.2.2 On-Device Data Encryption (ODE) ODE ist ein im Knox enthaltenes Feature zur Verschlüsselung von Daten. Dabei kann sowohl der sichere Container als auch der normale Bereich, sowie interner als auch externer Speicher verschlüsselt werden. Verwendet wird ein AES Algorithmus mit einem 256 Bit starken Schlüssel. Die Verschlüsselungsfunktion kann vom User selbst unter den Einstellungen oder vom IT Administrator durch eine Richtlinie aktiviert werden.

3.4.2.2.3 Virtual Private Network Support (VPN) VPN-Verbindungen verwendet man um sicherzustellen, dass Daten bei der Übertragung geschützt sind und um den Netzwerkverkehr nicht mit den Daten von persönlichen Apps zu belasten.

- **Funktionsweise**
  KNOX Workspace bietet 3 VPN-Optionen für den Schutz Ihrer Daten bei der Übertragung. Ein geräteweites VPN kann von Benutzern konfiguriert werden, sofern sie den entsprechenden Servernamen und die dazugehörigen Informationen zur Hand haben. VPN pro App oder ein containerweites VPN kann von IT-Administratoren über die MDM-Konsole eingerichtet werden. Auf der MDM-Konsole kann man bis zu 5 verschiedene VPN-Profile einrichten und diese einzelnen Apps zuweisen, um VPN pro App zu implementieren.
- **Sicherheit**
  VPN-Verbindungen sind die sicherste Methode, um die Daten bei der Übertragung zu schützen. Der KNOX VPN-Client kann ein bestehendes VPN-Gateway für den Schutz von Daten verwenden. KNOX VPN ist über Ihre MDM-Konsole im FIPS-Modus konfigurierbar. Zu den Sicherheitsfunktionen gehören NSA-Suite-B-Algorithmen, Unterstützung für X.509 mit Zertifikatsprüfung auf OCSP-Basis und 256-Bit-AES-Verschlüsselung. Wenn Ihr Unternehmen SmartCards verwendet, können diese mit VPN-Anmeldedaten konfiguriert werden.

3.4.2.3 Management

Mit MDM (Mobile Device Management) bietet Samsung Knox der IT Abteilung des Unternehmens die Möglichkeit eine integrierte Lösung zur Verwaltung und Administration von Samsung-Geräten vorzunehmen, ohne dabei auf Drittanbieter zurückgreifen zu müssen. Der Nachteil dabei ist die ausschließliche Verwendung mit Samsung-Geräten, welche über Samsung Knox verfügen.

Samsung Knox gibt es in 2 Varianten, wobei beide Varianten ein MDM als Basis benötigen um zu funktionieren.

3.4.2.3.1 Knox Lösungen

- **Knox Express**
  Diese Lösung ist für kleine bis mittelständische Unternehmen gedacht. Sie ist gratis, jedoch limitiert auf 250 Geräte.

  Falls die 250 Plätze nicht mehr reichen, kann man Knox Express problemlos auf Knox Premium updaten.

  Das ist die Variante, die die Projektgruppe empfehlen würde, da sie gratis ist und da laut Angaben von der Firma Kapsch es eher unwahrscheinlich ist, dass ein Kunde mehr als 250 Geräte hat und falls doch ist es ja problemlos auf Premium erweiterbar.
- **Knox Premium**
  Diese Komplettlösung eignet sich ideal für Unternehmen, in denen Sicherheit oberste Priorität hat.

  Knox Premium kostet pro Gerät pro Monat 1$ und kann um zusätzliche Add-ons erweitert werden.

  Weiters bietet Knox Premium einige zusätzliche Features für Android und eine bessere IOS Integration. In Hinblick auf die von Kapsch erwünschten Funktionalitäten bringen diese zusätzlichen Features jedoch keine Vorteile.
- **Add-Ons**
  - **Knox Workspace**
    Knox Workspace ist eine Erweiterung im Bereich Container.

    Dieses Add-on bietet eine einfachere Konfiguration, zwei Container pro Gerät, zusätzliche Apps und verbesserte Sicherheitsfunktionen.

    Die Hauptfeatures, die Workspace mit sich bringt, sind:

    * Erweiterte Containerverwaltung mit sicheren Richtlinien
    * Datenverschlüsselung bei jedem Entsperren des Containers
    * Pro-App-VPN für sichere und schnelle Verbindung
    * Support für zwei separate Container für maximale Produktivität und für Trennung von arbeitsbezogenen und privaten Daten

    All diese Einstellungen, die mit Knox Workspace dazu kommen, gelten nur innerhalb des Containers. Da es jedoch keine Einstellung gibt um zu verhindern, dass der Benutzer den Container verlässt und somit alle Workspace-Einstellungen umgeht, ist dieses Add-on in unserem Fall ungeeignet.

    Jedoch soll in Zukunft eine Art Contaier-only-mode implementiert werden. Sobald dieser vorhanden ist, wäre Knox Worspace auf jeden Fall ein Addon das in Betracht gezogen werden sollte.
  - **Knox IAM**
    Knox IAM ist eine SSO-Lösung(Single Sign-On-Lösung).

    Das heißt mit dem IAM Add-on hat man nur mehr einen einzigen Account, mit den man sich in allen Apps anmelden kann.

3.4.2.3.2 MDMs Knox bietet zur Verwaltung der Geräte ein eigenes MDM, das EMM, hat aber auch genügend Partner, deren MDMs Samsung Knox Funktionen implementiert haben.

Samsung Knox funktioniert also mit jedem MDM aus der nachfolgenden Liste:

- **Liste an Kompatiblen MDMs**
  - Samsung EMM
  - MobileIron
  - Absolute Software
  - AirWatch
  - CA Technologies
  - Centrify
  - Citrix
  - FancyIron
  - MaaS360
  - NQ Mobile
  - Samsung SDS
  - SAP
  - SOTI
- **Samsung EMM**
  Samsung KNOX EMM ist eine cloudbasierte Verwaltungslösung für Unternehmen. IT-Administratoren können damit Benutzer, Apps und plattformübergreifende Geräte über eine cloudbasierte Konsole verwalten. Außerdem bietet KNOX EMM Single Sign-On (SSO) und eine starke Authentifizierung für eine benutzerfreundliche und sichere Arbeitsumgebung für Mitarbeiter.

  Es ist einfach zu installieren und sehr übersichtlich gestaltet. Falls in der Firma noch keine andere MDM Lösung installiert sein sollte, ist EMM zu empfehlen.

Auswahl & Konzept

4.1 Endergebnis und Empfehlung

Nach Abschluss des Evaluierungsprozesses ist das Projektteam an diesem Punkt in der Lage eine Empfehlung an das Unternehmen Kapsch auszusprechen. In diesem Teil wird nun erläutert, welches evaluierte Konzept mit den Anforderungen des Auftraggebers übereinstimmt und wodurch dies zu Stande gekommen ist. Nachdem die Linuxmanipulation aus Garantietechnischen Gründen bereits als ausgeschieden gilt, befasst sich der folgende Abschnitt nur mehr mit den Konzepten MDM, MDM + Container und Samsung Knox.

4.2 Evaluierung

Um feststellen zu können welche der 3 Lösungen am besten die Anforderungen des Auftraggebers erfüllen kann, hat das Projektteam eine Nutzwertanalyse erstellt, mit derer Hilfe das Projektteam die Funktionen der Systeme direkt miteinander vergleichen konnten.

Nutzwertanalyse ERP Lösung		max. Punkte	Gewicht	MDM	Bewertung	MDM+Container	Bewertung	Samsung Knox	Bewertung
Allgemein			**8**						
	Nutzbar ohne Public-Cloud-Services		3	10	30	10	30	10	30
	Keiner speziellen Person zugewiesen		5	10	50	10	50	10	50
Nutzwert Allgemein		***80***			**80**		**80**		**80**
Inhaltseinstellung			***44***						
	Installieren von Apps kann verboten werden		6	7	42	7	42	10	60
	Deinstallieren von Apps kann verboten werden		6	3	18	3	18	0	0
	Kein benutzerseitiger Zugang zu Systemeinstellungen		8	10	80	10	80	10	80
	Standard Apps können ausgeblendet werden		2	2	4	2	4	2	4
	Browser Inhalt kann gefiltert werden		4	10	40	10	40	10	40
	Remote-Zugriff ist möglich		8	10	80	10	80	10	80
	Apps können per Remote verwaltet werden		5	6	30	8	40	10	50
	Websiten können per Remote verwaltet werden		2	0	0	0	0	0	0
	Inhalt (Dokumente, Videos, Bilder) können per Remote verwalten können		1	0	0	8	8	0	0
	Hintergrund und Speerbildschirm können per Remote verwalten können		2	5	10	5	10	5	10
Nutzwert Inhaltseinstellungen		***440***			**304**		**322**		**324**
Statistik			***7***						
	Up/Down-Time einsehbar		3	10	30	10	30	10	30
	Stastiken sind pro Device und Standort auslesbar		3	10	30	10	30	10	30
	Statisiken über Aktivitätszeit, App aufrufe und Websiten		1	4	4	4	4	10	10
Nutzwert Statistik		***70***			**64**		**64**		**70**
Benutzer und Geräte			***10***						
	Geräte können zu einem spezifischen Status zurückgesetzt werden		5	3	15	3	15	3	15
	Benutzerdaten (history, Spielstände,Lesezeichen) können gelöscht werden		3	3	9	4	12	3	9
	Automatischen zurücksetzen der Geräte nach einer Zeiteinheit		2	0	0	0	0	0	0
Nutzwert Benutzer und Geräte		***100***			**24**		**27**		**24**
Managementseitige Anforderungen			***20***						
	Geräteverwaltung von mehreren Standorten möglich		3	10	30	10	30	10	30
	Geräte (inkl. deren Statuse) können angezeigt werden		4	10	40	10	40	10	40
	Fehlerberichte sind pro Gerät und pro Standort verfügbar		4	7	28	7	28	9	36
	Geräte können zu einem spezifischen Status per Remote zurückgesetzt werden		5	3	15	3	15	3	15
	Statisiken können per Remote ausgelesen werden		3	10	30	10	30	10	30
	Inhalt (Dokumente, Videos, Bilder) können per Remote geändert können		1	0	0	8	8	0	0
Nutzwert Mangementseitige Anforderungen		***200***			**143**		**151**		**151**
Zusätzliche Anforderungen			***11***						
	Gerät ohne Kiosk-Mode verwendbar		5	10	50	10	50	10	50
	Hardwareunabhängig		1	10	10	10	10	10	10
	Geringer Wartungsaufwand		2	10	20	10	20	10	20
	mehrsprachen Unterstützung		2	10	20	10	20	10	20
	Einsetzbar ab dem ersten Gerät		1	0	0	0	0	10	10
Nutzwert Zusätzliche Anforderungen		***110***			**100**		**100**		**110**
Nutzwert Gesamt		*1000*			715		744		759
Rang Nutzwert					**3**		**2**		**1**
Kosten		*10*			6		4		10
Rang Kosten					**2**		**3**		**1**
Kosten/Leistungsverhältnis - Kosten je Pkt.					119,17		186,00		75,90
Rang Kosten/Leistungsverhältnis					**2**		**3**		**1**

4.3 Auswertung der Evaluierung

Um eine aussagekräftiges Ergebnis zu erhalten, wurden die Anforderungen des Gesamtsystems in verschiedene Unterpunkte unterteilt, welche auf ihre Funktionstüchtigkeit hin untersucht wurden.

Die Lösungen wurden dann je nach Erfüllungsgrad der einzelnen Punkte bewertet, was eine objektive Vergleichbarkeit schaffen sollte.

4.3.1 Allgemein

Da alle 3 Systeme sowohl ohne pulic-cloud-service nutzbar waren als auch keiner speziellen Person zugewiesen werden müssen, haben die alle 3 Lösungen volle Punkte erreich und sind somit was diesen Punkt angeht gleich auf.

4.3.2 Inhaltseinstellungen

In diesem Bereich hat sich herausgestellt ist Samsung Knox kapp aber doch noch vor der MDM+Container Lösung das am besten geeignete System. Denn auch wenn es mit MDM+Containern möglich ist, die Deinstallation von betriebsinternen Apps zu verbieten und man auch Inhalte der Geräte wie zum Beispiel Bilder, Dokumente, etc per remote verwalten kann, ist Samsung die hier die bessere Alternative, da es eine bei weitem bessere Verwaltung von Apps per Remote bietet.

Eine Anforderung die jedoch keines der drei Systeme erfüllen konnte war das Verwalten von Websiten per remote.

4.3.3 Statistik

Auch den Unterpunkt Statistiken kann Samsung Knox wieder für sich entscheiden. Diesmal jedoch mit einem Größeren Vorsprung. Zurückzuführen ist das darauf, dass Samsung Knox die Möglichkeit bietet, sich jede beliebige Statistik die man für sein Unternehmen haben will, einfach mittels SQL Code selbst zu erzeugen(siehe Punkt 8.3.1 Statistiken).

4.3.4 Benutzer und Geräte

In diesem Bereich kann die MDM+Container Lösung punkten. Denn im Gegensatz zu den anderen zwei Systemen, kann man mit dieser Lösung am Gerät gespeicherte Lesezeichen löschen.

Auch bei diesem Unterpunkt gibt es wieder eine Anforderung die mit keinem unserer getesteten Systeme umsetzbar war. Und zwar war es nicht möglich die Geräte so zu konfigurieren, dass sie sich nach einer gewissen Zeit von alleine auf einen definierten Stand zurücksetzen.

4.3.5 Managementseitige Anforderungen

Bei der Erfüllung der Managementseitigen Anforderungen sind Samsung Knox und die MDM+Container-Lösung gleich auf. Denn den Vorsprung den Knox gewinnt durch besseres Handling der Fehlerberichte, kann die MDM+Container-Lösung dadurch wegmachen, das sie im Gegensatz zu Samsung Knox eine Möglichkeit bietet, Geräteinhalt per remote zu ändern.

4.3.6 Zusätzliche Anforderungen

Da Samsung Knox das einzige System ist, welches bereits ab dem ersten Gerät einsetzbar ist und sonst alle Punkte von allen Systemen erfüllt werden ist auch hier Knox der klare Sieger.

4.4 Empfehlung

Nach eingehender Analyse ist das Projektteam zu dem Schluss gekommen, dass für die geplanten Projekte der Firma die Betriebsplattform Samsung Knox am ehesten geeignet ist. Es implementiert die meisten der benötigten Features, aber lässt dennoch einige fundamentale Punkte aus. Deshalb ist es hier für das Projektteam auch nicht möglich eine hundert prozentige Empfehlung zu geben. Den Informationen der Dokumentation von Samsung Knox zur Folge werden einige benötigte Features in kommenden Versionen eingebaut. Allerdings ist nicht absehbar ob und wann diese erscheinen. Für die beiden anderen Systeme kann deshalb keine Empfehlung ausgesprochen werden, weil sie weniger und besonders im Einsatz mit Nicht-Samsung-Geräten signifikant weniger Funktionen bieten.

Lessons Learned

5.1 Sebastian Götze

Durch das Diplomprojekt Kapsch Tablet Infrastructure habe ich gelernt, dass es besonders bei der Recherche wichtig ist, sich nicht immer auf den Inhalt von Dokumenten und Präsentationen zu verlassen, weil dieser oft nicht die Realität wiederspiegeln. Besonders beim System Samsung Knox hat sich dies deutlich herauskristallisiert, da in Samsungs Unterlagen zwar steht, dass es einen Modus gibt, in dem der Nutzer sich ausschließlich in einem abgesicherten Container befindet, aber dieser in der Praxis nur bedingt existiert. Der Modus war vorhanden, allerdings war es für den Nutzer jederzeit möglich aus diesem herauszuspringen. Ein weiterer wichtiger Lernaspekt für mich war es, dass man bei der Installation von unbekannten Systeme vor Beginn die Installationsanleitung lesen sollte und nicht sich danach auf Fehlersuche zu begeben. Besonders bei der Einrichtung der zusätzlichen Containerapplikationen zu MobileIron wäre dies von Vorteil gewesen, da es hier einige wichtige Details zu beachten gab, ohne die der Prozess nicht abgeschlossen werden konnte.

5.2 Samuel Hammer

Während das Projekts KTI habe ich in erster Linie vieles bezüglich Projektmanagement gelernt. Durch die oftmals eng gesetzten Zeitintervalle, war es äußerst wichtig die Koordination zwischen den zwei Subteams bzw. den einzelnen Teammitgliedern zu perfektionieren. Ein teilweise großes Problem war auch, dass sich Projektbezogene Abgaben und Termine immer wieder mit Tests, Schularbeiten und anderen unterrichtsbezogenen verpflichtungen überschnitten. Hier war es von großer wichtigkeit, sich mit den zuständigen Lehrern bzw. dem Projektpartner abzusprechen. Einiges gelernt habe ich natürlich auch in Hinblick auf die wichtigen Dokumente, welche es im Laufe des Projektes zu erstellen und aktualisieren gab. Alles in Allem war das Projekt "Kapsch Tablet Infrastructureëine wertvolle Erfahrung, aus der ich viel für meine zukünftige berufliche Laufbahn mitnehmen werden.

5.3 Michael Kaufmann

Einer der wichtigsten Punkte die ich durch die Arbeit am Diplomprojekt in Erfahrung bringen konnte ist, wie wichtig gut funktionierende Kommunikation innerhalb des Projekt Teams ist. Um den Arbeitsaufwand bestmöglich zu verteilen und um, so produktiv wie möglich, sein zu können haben wir Sub-Teams gebildet und unser Projekt in technischen und dokumentationsbezogenen Teil geteilt. Das Problem hierbei war nur das die Sub-Teams teilweise abhängig von einander waren und es hin und wieder vorkam das die Arbeit, auf die das eine Team warten musste um weiter machen zu können, bereits erledigt war, aber diese nichts davon wussten. Dies führte logischerweise zu Verzögerung des gesamten Projekts. Außerdem habe ich gelernt dass man der Dokumentation und den Whitepapers einer Fertiglösung nicht immer blind vertrauen kann. Denn wie zum Beispiel im Falle Samsung knox sich herausgestellt hat, werden oft Funktionen versprochen, welche tatsächlich nicht existieren. Dies führt nicht nur zu einer Verzögerung des Aufwandes was das testen der Lösung betrifft, sondern auch dazu das die Lösung vor dem Testen besser eingeschätzt wurde als sie tatsächlich war.

5.4 Konstanze Müller

Das diesjährige Diplomprojekt mit unserem Partner der Kapsch Business Com gewährte mir in erster Linie einen Einblick eines kompletten Projektdurchlaufes in der Praxis. Theoretische Inhalte für einen reibungslosen Projektverlauf, welche wir in den Jahren davor im Unterricht gelehrt bekommen haben, konnten wir als Projektteam in diesem Abschlussjahr erfolgreich umsetzen. Auch wenn unser Projektleiter Philip Steinhäuser in diesem Bereich die Oberhand hatte, unterstütze ich ihn so gut wie möglich und war somit auch mit formalen Abwicklungskriterien vertraut. Aus Projekttechnischer Sicht lernte ich, dass man den Aufwand für die Evaluierung von einem System schwer im Voraus abschätzen kann. Des Öfteren war es der Fall, dass Anforderungen von der Checkliste nicht auf Anhieb funktionierten. Selbstverständlich hört man nicht gleich auf zu testen, sondern probiert weiteres. Das Schwierige meiner Meinung war nun den richtigen Zeitpunkt zu finden, um mit einer hohen Wahrscheinlichkeit sagen zu können, dass diese Anforderung vom ausgewählten System nicht unterstützt wird. In Summe bedeutet dies, dass man in Projekten wie diesem sich selbst eine Grenze stecken muss um auch ein schlussendliches Ergebnis vorlegen zu können.

5.5 Philip Steinhäuser

Für mich als Projektleiter hat dieses Projekt ein Menge an Erfahrungen nicht nur im Managementbereich sondern auch im Zwischenmenschlichen Umgang bereitgehalten, da ich in einigen Situationen dazu gezwungen war ein Machtwort zu sprechen wenn es Streitigkeiten innerhalb des Projektteams gegeben hat. Zusätzlich hab ich einiges zum Thema Zeitplanung und Terminkoordination gelernt, was mir auf meinem weiteren Weg welcher mich hoffentlich in die Wirtschaft bzw. einen wirtschaftlichen Beruf führen wird helfen wird. Zu guter Letzt bleibt mir nicht mehr zu sagen als das ich dieses Projekt als ein für mich sehr Erfahrungsreiches Ereignis verbuche, welches mir in den Bereichen der Teamführung bzw. Projektleitung oder Projektcontrolling sowie Projektmanagement als auch Krisenmanagement eine bzw. mehrere Lehrstunden erteilt hat. Dennoch möchte nichts was wir an dem Projekt gemacht haben sowie die Zusammenarbeit mit diesem großartigen Projektteam missen.

Danksagung

Wir möchten uns in erster Linie bei unseren Betreungslehrern, Herrn DI Harald Swoboda und Herrn DI Hannes Färberböck für die großartige Unterstützung während des gesamten Projektverlaufs bedanken.

Großer Dank gilt auch Herrn DI (FH) Mag. Bernhard Bruckner und Herrn Jürgen Krammer, die uns im Namen der Kapsch BusinessCom bei Problemen immer mit Rat und Tat zur Seite standen.

Quellenverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis