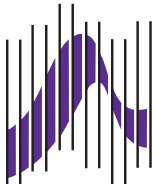
Entwicklung einer mobilen Anwendung zur Verwaltung von Cloud Infrastruktur

14. Juni 2012



Hochschule Ravensburg-Weingarten
Fakultät Elektrotechnik und Informatik
Studiengang Angewandte Informatik

Thomas Merkel
Matrikelnr. 19868
tm@core.io

Betreuer:
Prof. Klemens Ehret
Dipl.-Inf. Karl-Heinz Erdmann

Eidesstattliche Erklärung

Hiermit erkläre ich, Thomas Merkel, geboren am 10.10.1985 in Tettnang, dass ich die vorliegende Bachelorthesis mit dem Titel „Entwicklung einer mobilen Anwendung zur Verwaltung von Cloud Infrastruktur“ selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden.

Tettnang, den 14. Juni 2012

Abstract

Smartphones sind aus unserem Alltag nicht mehr wegzudenken. Ob beim Einkaufen, im Urlaub oder zu Hause auf dem Sofa, überall ist das Smartphone dabei. Durch Virtualisierung vereinfachte IT-Systeme, die heute als „Cloud“ bezeichnet werden, sollen unsere Daten oder Webinhalte ausfallsicher angeboten werden. Es gibt Administratoren die versuchen diesen Dienst für den Kunden immer erreichbar zu halten.

Daher soll mit dieser Bachelor-Thesis versucht werden, eine mobile Anwendung zur Verwaltung der Cloud Infrastruktur zu entwickelt. Im Vordergrund steht die Usability einer iPhone Anwendung. Während der Thesis wird ein Prototyp entwickelt und an den gängigen Usability-Regeln gemessen.

Das Kapitel der Grundlagen umfasst die Definition der verschiedenen „Cloud“-Arten bis zu den Usability-Regeln und die Relevanz für mobile Anwendungen. In der Analyse-Phase wird der IST-Zustand aufgenommen und alternative Anwendungen betrachtet. Darauf folgt die Konzeption anhand der iPhone Navigationsmodelle und Usability-Tests. Im letzten Kapitel, der Realisierung, wird die eingesetzte Technologie näher erläutert.

Vorwort

Die vorliegende Bachelorarbeit wurde in der Zeit vom 1. März bis zum 15. Juni erstellt. Sie ist Bestandteil des Bachelor-Studiengangs Angewandte Informatik an der Hochschule Ravensburg-Weingarten und entstand bei der Avira Operations GmbH & Co. KG in Tettnang.

> “Keep it simple and self-explanatory.”
>
> Original von Clarence "Kelly" Johnson

Mit der Arbeit wird versucht immer einen Kompromiss zwischen Design, Usability und Funktionalität zu finden.

Danksagung

An dieser Stelle möchte ich mich bei einigen Personen bedanken, die mich während dieser Arbeit unterstützt haben. Ich danke Karl-Heinz Erdmann, meinem fachlichem Betreuer bei der Avira, für sein Engagement und seine Geduld. Ebenso danke ich Prof. Klemens Ehret, der diese Arbeit auf Seiten der Hochschule Ravensburg-Weingarten betreut hat.

Besonderer Dank gilt den Mitarbeitern der IT Abteilung von Avira für ihre Hilfe, Ideen und guten Ratschläge.

Inhaltsverzeichnis

1 Einleitung

1.1 Projektträger

Der Auftraggeber und Träger dieses Projekts ist die IT Abteilung der Avira Operations GmbH & Co. KG[1]. (im Folgenden Avira genannt). Die IT Abteilung versteht sich als serviceorientierter Dienstleister für Avira, der Kunde und die Verfügbarkeit der Dienste liegen somit im Mittelpunkt.

Die Avira ist mit rund 100 Millionen Kunden und 500 Mitarbeitern ein weltweit führender Anbieter selbst entwickelter Sicherheitslösungen für den kommerziellen und privaten Einsatz. Das Unternehmen gehört mit mehr als 25-jähriger Erfahrung zu den Pionieren in diesem Bereich.

Als führender deutscher Sicherheitsspezialist verfügt Avira über fundierte Erfahrung im Entwickeln und Unterstützung ihrer Lösungen. Neben Programmen direkt für den Einzelplatzbetrieb bietet sie, hauptsächlich professionelle, Lösungen für systemübergreifenden Schutz von Netzwerken auf verschiedenen Ebenen an. Hierzu zählen unter anderem Produkte für Workstations, File-, Mail- und Web-Server. Auch Gateway-Rechner können, wie Arbeitsplatzrechner, über eine zentrale Verwaltungskonsole betriebssystemübergreifend verwaltet werden. Zu den Verwaltungsprodukten der einzelnen Lösungen kommen noch Sicherheitsprogramme für PDAs, Smartphones und Embedded Devices hinzu. Ein signifikanter Sicherheitsbeitrag ist Avira AntiVir Personal, das millionenfach bei Privatanwendern im Einsatz ist.

Avira ist in Tettnang am Bodensee einer der größten regionalen Arbeitgeber. Sie unterhält mehrere Unternehmensstandorte in Deutschland und pflegt Partnerschaften in Europa, Asien und Amerika. Mehrere Dutzend Virus-Researcher in verteilten Virenlabors kümmern sich, rund um die Uhr, um die lokalen und globalen Bedrohungen der Virenfront. Bestätigt wird diese Arbeit etwa durch mehrfache Testauszeichnungen mit dem VB 100% des Virus Bulletin[2] oder der wiederholten TÜV-Zertifizierung.

1. Avira Operations GmbH & Co. KG: http://www.avira.com
2. Virus Bulletin: http://www.virusbtn.com

Zu den nationalen und internationalen Kunden zählen namhafte, börsennotierte Unternehmen aber auch Bildungseinrichtungen und öffentliche Auftraggeber. Neben dem Schutz der virtuellen Umgebung kümmert sich Avira, durch Fördern der Auerbach Stiftung[3], um mehr Schutz und Sicherheit in der realen Welt.

1.2 Motivation

„Alles muss in die Cloud“, wie oft liest man dies in IT Magazinen oder hört die Aussage von Managern? Doch diese Infrastruktur muss auch gepflegt und administriert werden. Gerade im Notfall müssen IT Mitarbeiter, auch von unterwegs, schnell Probleme beheben. Eine Bereitschaft der IT Administratoren von 24 Stunden, 7 Tage die Woche können sich viele kleine Firmen nicht leisten und setzen auf das Engagement der Mitarbeiter. Einige der Probleme könnten bequem per Smartphone gelöst werden.

Der Marktanteil für Smartphones mit Google Android oder Apple iOS[4] steigt jedes Jahr um mehrere Prozent. So sind diese Betriebssysteme schon auf mehr als 50% der Geräte im Umlauf.[5]

Durch den immer weiter steigenden Verkauf von Smartphones wächst auch die Verwendung mobiler Anwendungen. Diese Anwendungen haben jedoch völlig andere Anforderungen als Programme für Desktop Computer oder Laptops. Herausstechendstes Merkmal ist der kleinere Bildschirm sowie die Bedienung mittels Touchscreen, hierauf muss beim Aufbau der Benutzeroberfläche besonders geachtet werden.

1.3 Problemstellung

> “Als Admin eines Online-Shops fahre ich mit meinem Sportwagen Richtung Süden. Dank meines iPhone bemerke ich, dass meine Internetseite auf Facebook angekündigt wird. Durch die vielen Zugriffe auf den Webserver wäre meine Seite nicht mehr erreichbar und ich würde Umsatz verlieren.”
>
> User-Story eines IT Mitarbeiters

3. Auerbach Stiftung: http://www.auerbach-stiftung.de
4. iOS Technology Overview: http://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechOverview.pdf
5. Marktanteile laut Gartner Inc. (April 2011): http://www.gartner.com/it/page.jsp?id=1622614

Jeder IT Mitarbeiter der Avira besitzt ein iPhone und soll, falls möglich, schnell auf Probleme reagieren können. Da viele Dienste auf virtuellen Servern in der Cloud Infrastruktur zur Verfügung stehen, müssen diese schnell und einfach verwaltet werden können.

Erfahrungen der Mitarbeiter zeigen, dass nicht die Dienste selbst das Problem sind, sondern oftmals eine Störung am virtuellen Server vorliegt. So muss, unter anderem, bei hoher Last des Servers mehr Arbeitsspeicher oder mehrere CPUs hinzugeschaltet werden. Da ein Smartphone leichter zu transportieren ist als ein Notebook, bietet sich eine mobile Anwendung für die Steuerung der Cloud Systeme an.

Zum aktuellen Zeitpunkt steht noch keine mobile Anwendung zur Verfügung, welche die Verwaltung der Server ermöglicht.

1.4 Zielsetzung

Durch dieses Projekt soll ein Konzept für eine iOS Anwendung erstellt werden, mit der die Möglichkeit besteht virtuelle Server innerhalb einer Cloud Infrastruktur zu steuern. Die Zielgruppe für die Anwendung sind IT Mitarbeiter und Administratoren. Die Anwendung kommt häufig im Problemfall zum Einsatz, somit ist auf eine einfache und intuitive Bedienung zu achten. Bei der Anwendung steht die Benutzerfreundlichkeit im Vordergrund.

Als Schnittstelle zwischen iOS Anwendung und den virtuellen Servern soll libvirt[6], eine API zur Verwaltung von verschiedenen Virtualisierungstechnologien, zum Einsatz kommen.

1.5 Abgrenzung

In der Bachelor-Thesis wird keine vollwertige iOS Anwendung entwickelt. Durch ein Konzept und die Erstellung eines Prototyps soll ein gutes Gleichgewicht zwischen Benutzerfreundlichkeit und Anwendungsumfang erreicht werden.

6. libvirt: http://www.libvirt.org

1.6 Zielgruppe der Thesis

Die Leserzielgruppe der Arbeit umfasst Entwickler von mobilen Anwendungen für iPhone Geräte sowie technisch versierte Systemadministratoren. Bei der Entwicklung der Anwendung werden Usability-Regeln und Design berücksichtigt, daher ist die Thesis auch für Usability-Forscher interessant.

1.7 Struktur und Aufbau der Arbeit

Die Bachelorarbeit ist in vier Teilbereiche gegliedert. Sie spiegelt dadurch den ungefähren Verlauf des Projekts wieder.

In der *Einleitung* wird der Projektträger vorgestellt und die Ziele des Projekts beschrieben. Die für die Arbeit relevanten Begriffe werden in den *Grundlagen* erklärt. Es erfolgt eine Vorstellung des verwendeten Mobilen Endgeräts und der Cloud Schnittstelle. Außerdem werden die Kriterien für die Usability-Tests beschrieben.

In den Kapiteln *Analyse* und *Konzeption* erfolgt eine Darstellung der aktuellen Situation. Ein erster Prototyp wird erstellt und beschrieben um Usability-Tests anwenden zu können. Anschließend wird die Zielgruppe zum Prototypen befragt. In der *Realisierung* werden die technischen Aspekte zur Entwicklung der Anwendung angesprochen. Es werden Quelltexte zur grafischen Benutzeroberfläche und Schnittstellen näher erläutert.

Den Abschluss der Thesis bildet das *Fazit* über das gesamte Projekt und die Arbeit. Es folgt ein Ausblick auf den Verlauf des Projekts außerhalb der Bachelor-Thesis.

1.8 Formale Hinweise

Unbekannte Begriffe oder weitere Hinweise zu einem Wort werden per Fußnote erklärt. Meist erfolgt ein weiterführender Link zu einer Internetseite. Zitate oder Quellenverweise sind per Eckiger-Klammer gekennzeichnet und im Quellenverzeichnis beschrieben. Verweise auf Kapitel oder wichtige Textstellen sind *kursiv* geschrieben.

2 Grundlagen

In diesem Kapitel werden einige Grundlagen und Grundbegriffe näher erläutert. Die Erklärung erfolgt auf Basis der benötigten Informationen für die Bachelorarbeit.

2.1 Mobile Endgeräte

Der Begriff mobile Endgeräte wird umgangssprachlich mit Mobiltelefone, Smartphones und PDAs gleichgesetzt. Unter diesen Begriff fallen aber alle Geräte, die ohne größere körperliche Anstrengung transportiert und mobil eingesetzt werden können. Somit auch MP3-Player, Notebooks oder Tablet-PCs. [wiki11a]

In der Bachelorarbeit wird der Fokus auf Smartphones und die mobilen Anwendungen gelegt.

2.1.1 Klassische Mobiltelefone

Die klassischen Mobiltelefone sind meist mit einem Nummernblock und integriertem T9 Ziffernblock[7] ausgestattet. Die Größe und Farbtiefe des Bildschirms ist deutlich geringer als bei einem Smartphone.

Bei neueren Modellen besteht zum Teil die Möglichkeiten Anwendungen in der Programmiersprache Java zu entwickeln. Die Schnittstellen zum Betriebssystem des Mobiltelefons erlauben aber nur einen Bruchteil der möglichen Funktionen.

2.1.2 PDAs

Die persönlich digitalen Assistenten, kurz PDAs, sind eine Kombination aus den klassischen Mobiltelefonen und Computern. Sie bieten zur Telefonfunktion auch persönliche Kalender-, Adress-, und Aufgabenverwaltung an. PDAs können zusätzlich Office-Dateien verarbeiten. Sie werden durch die Smartphones langsam vom Markt verdrängt.

7. T9 Ziffernblock Patent: http://worldwide.espacenet.com/publicationDetails/biblio?CC=EP&NR=1256871A2&KC=A2&FT=D&ND=3

2.1.3 Smartphones

Smartphones stellen die Computerfunktionalität auf einem Mobiltelefon zur Verfügung. Diese Geräte sind über die verschiedensten Telekommunikationsdienste, wie UMTS[8], HSDPA[9] oder Wireless-LAN, internetfähig.

Sie sind durch ihre Konstruktion und Bedienung nicht primär zum Telefonieren optimiert, stattdessen ermöglichen sie die Verwendung von einer großen Anzahl an Anwendungen. Daher sind typische Merkmale ein großer Touchscreen und eine alphanumerische Tastatur. Die Tastatur befindet sich bei den meisten Modellen im Touchscreen. Der Benutzer hat daher bei der Eingabe, keine oder nur eine akustische Rückmeldung, was die Eingabe erschweren kann.

2.1.4 mobile Anwendungen

Eine mobile Anwendung, im allgemeinen Sprachgebraucht „App“ genannt, ist eine Software die für ein Smartphone entwickelt wurde und meist über einen integrierten Onlineshop bezogen werden kann.

Die ersten Mobiltelefone enthielten schon Anwendungen wie Kalender, Taschenrechner oder Spiele. Durch die Entwicklung von Java ME[10] war es Entwicklern möglich weitere mobile Anwendungen zu programmieren und zur Verfügung zu stellen. Aber erst mit dem Erscheinen von Apples iPhone und Googles Android Smartphones, wurden die „App“ populär. Dies ist nicht allein dem Marketingkonzept zuzuschreiben, auch wenn dies einen großen Einfluss auf die Popularität hatte. Die vereinfachte Installation und Bereitstellung über integrierte Onlineshops konnte die breite Bevölkerung ansprechen.

Im Oktober 2011 befanden sich über 400.000 Anwendungen[11] im App Store von Apple. Dazu wurden im März 2012 über 20 Millionen Downloads[12] App Store im gemessen. Durch diese Zahlen ist der Trend für steigende Entwicklung von mobilen Anwendungen klar erkennbar.

8. Universal Mobile Telecommunications System
9. High Speed Downlink Packet Access
10. Java ME: http://www.oracle.com/technetwork/java/javame
11. Statista, Anzahl der Apps in den Top App-Stores im Oktober 2011: http://de.statista.com/statistik/daten/studie/208599/umfrage/anzahl-der-apps-in-den-top-app-stores/

2.2 Cloud-Computing

Die Cloud ist ein umschreibender Begriff für den Ansatz IT Infrastruktur, wie zum Beispiel Rechenkapazität, Speicherbedarf oder auch Software über ein Netzwerk zur Verfügung zu stellen. Durch die Aufteilung der einzelnen IT Ressourcen können diese dynamisch und an den Bedarf angepasst, angeboten werden.

Vereinfach kann man sich die Cloud wie folgt vorstellen: Man besitzt ein Online-Kleidergeschäft, mit zwei Mitarbeitern. Diese Mitarbeiter pflegen die Seite, verpacken und versenden die Ware. Während dem Jahr können die zwei Mitarbeiter alle Kunden jederzeit bedienen und es kommt zu keinen Engpässen. Über die Weihnachtszeit ist der Andrang aber so hoch, dass Kunden ihre Kleidung nicht rechtzeitig erhalten, da dass verpacken viel zu lange dauert.
Nun gibt es für den Besitzer drei Möglichkeiten. Er stellt einen weiteren Mitarbeiter ein, den er aber ein ganzes Jahr beschäftigen müsste und zum Teil nicht ausgelastet ist. Die zweite Möglichkeit wäre, dass der Besitzer auf den Mehrumsatz verzichtet. Oder der Besitzer wendet sich für diese Zeit an einen Dienstleister der das verpacken und versenden für Ihn übernimmt.

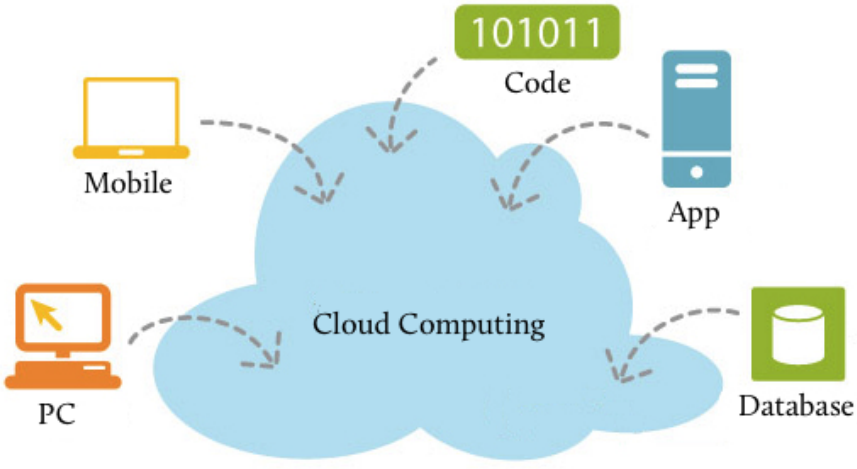


Abbildung 1: Elemente des Cloud-Computing

12. Statista, Anzahl der heruntergeladenen Anwendungen im Apple App Store: http://de.statista.com/statistik/daten/studie/20149/umfrage/anzahl-der-getaetigten-downloads-aus-dem-apple-app-store/

Aus der Sicht der Informatik, hat der Besitzer einen Online-Shop, dieser läuft auf einem Server. Für die üblichen Benutzerzahlen im Jahr ist dieser Server völlig ausreichend, jedoch steigt die Zugriffszahlen über die Weihnachtszeit. Statt einen weiteren Server zu kaufen, der jährlich Geld kostet, mietet sich der Besitzer für eine geringe Zeitspanne weitere Rechnerressourcen in der Cloud.

2.2.1 Technische Realisierungen

Cloud-Computing wird in drei technische Schichten, in einen so genannten Cloud-Stack, aufgeteilt. Jede Schicht stellt einen Grad der Abstraktion dar und wird in diesem Abschnitt näher erläutert. [bartonb08]

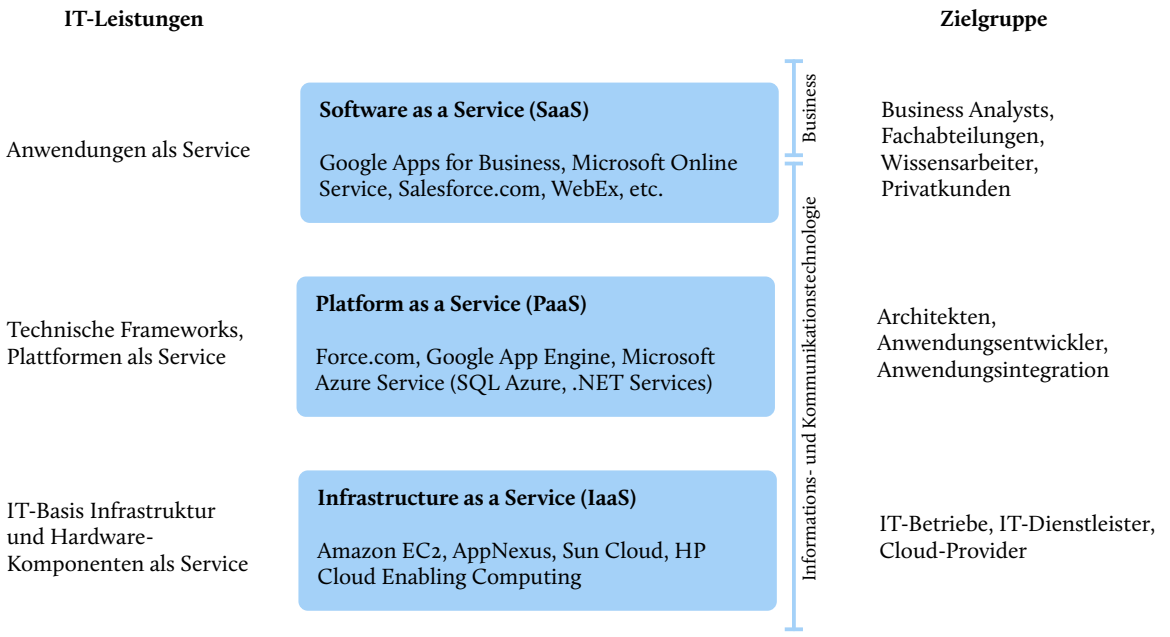


Abbildung 2: Cloud-Stack, die drei technischen Schichten

Infrastruktur (IaaS)

„Infrastructure as a Service“ befindet sich an der untersten Schicht im Cloud-Stack. Die Cloud-Provider bieten dem Benutzer Rechnerinstanzen, meist in Form von virtuellen Servern, an. Diese kann der Benutzer selbst verwalten. Der Cloud-Dienst ist skalierbar ausgelegt, jedoch nicht zwingend die Programme die der Benutzer auf den Rechnerinstanzen installiert.

Je nach Anforderung kann der Cloud-Dienst um Rechnerinstanzen erweitert oder verkleinert werden. Der Benutzer ist hierbei ab der Betriebssystemebene für seine Instanz verantwortlich.

Plattform (PaaS)

Bei „Platform as a Service“ steht die Anwendung des Entwicklers im Vordergrund. Der Entwickler stellt seine Anwendung über die Cloud zur Verfügung. Die Cloud selbst kümmert sich um die Aufteilung auf die Rechnungsinstanzen.

Da der Entwickler nur seine Anwendung liefert, kann die Cloud die Anzahl der tatsächlichen Instanzen jederzeit erhöhen oder reduzieren. In der Cloud werden vom Entwickler gelieferte Daten verarbeitet, das umliegende System ist für ihn nicht einsehbar.

Anwendung (SaaS)

Der Benutzer verwendet bei „Software as a Service“ eine bereits bestehende Anwendung in der Cloud. Für ihn sind die Platform und die Infrastruktur nicht sichtbar.

Beispiele für Cloud-Anwendung sind unter anderem Google Drive[13], DropBox[14] und Microsoft Office Communications Online[15].

2.2.2 Organisatorische Arten

Die Cloud wird meist, abhängig vom Anwendungsfall, in drei Organisationsformen eingeordnet. Eine Beschreibung der Organisationsformen erfolgt in diesem Abschnitt. [cloudadop10]

Private Cloud

Die Anbieter und Nutzer der „Private Cloud“ stammen aus der selben Organisation oder dem selben Unternehmen. Daten innerhalb dieser Cloud sind nur der Organisation zugänglich und nicht außerhalb erreichbar, dies bietet einen großen Sicherheitsaspekt.

13. Google Drive: http://drive.google.com
14. DropBox: http://www.dropbox.com
15. Microsoft Office Communications Online: http://www.microsoft.com/online/de-de/prodComm.aspx

Public Cloud

Die „Public Cloud“ ist nicht für eine Organisation beschränkt, sie ist öffentlich erreichbar und für jeden zugänglich. Eine wichtige Rolle spielt hierbei die Datensicherheit. Jeder Benutzer muss selbst Entscheiden welche und wie viele Daten er in der Cloud speichert. Ein bekanntes Beispiel für die „Public Cloud“ sind die Amazon Web Services[16].

Hybrid Cloud

Bei der „Hybrid Cloud“ handelt es sich um eine Mischung aus „Private und Public Cloud“. Eine Organisation verwendet eine „Private Cloud“ und wechselt im Fehlerfall oder bei hoher Belastung zur „Public Cloud“, dies wird auch „Cloud-Bursting“[17] genannt.

2.3 Virtualisierung

Für die Virtualisierung gibt es viele verschiedene Methoden, abhängig davon ob eine Anwendung oder ein gesamtes Betriebssystem virtualisiert werden soll. Die Virtualisierung spielt eine wichtige Rolle für die Cloud Infrastruktur. Für die Bachelorarbeit ist die Bereitstellung ganzer Rechnerinstanzen relevant. Die beste Leistung wird mit der Paravirtualisierung erreicht. [yaovt10] [yaocpu10] [yaodisk10]

2.3.1 Hardware Virtualisierung

Die virtuelle Maschine stellt dem Gastbetriebssystem Teilbereiche der Hardware in Form von virtueller Hardware zur Verfügung. Somit kann ein unverändertes Betriebssystem darauf in einer isolierten Umgebung laufen. Das Gastsystem muss hierbei für den gleichen CPU-Typ ausgelegt sein.

2.3.2 Emulation

Im Gegensatz zur Hardware Virtualisierung wird dem Gastbetriebssystem die komplette Hardware simuliert. Es ist damit möglich beinahe jedes Betriebssystem in virtuellen Umgebung zu starten. Der CPU-Typ muss nicht dem des Hostsystems entsprechen.

16. Amazon Web Services: http://aws.amazon.com
17. Cloud-Bursting: http://aws.typepad.com/aws/2008/08/cloudbursting-.html

2.3.3 Paravirtualisierung

Bei Paravirtualisierung wird zwar ein Betriebssystem virtuell gestartet, jedoch wird keine Hardware virtualisiert, sondern die Betriebssysteme verwenden eine abstrakte Verwaltungsschicht, um auf gemeinsame Ressourcen, wie Netzanbindung oder Festplattenspeicher, zuzugreifen. Damit das Betriebssystem auf der virtuellen Maschine ausgeführt werden kann muss es teilweise portiert werden. Durch diese Portierung kann sich die Leistung der virtuellen Maschine erhöhen. [degelas08]

2.3.4 Schnittstelle zur Virtualisierung

Durch die verschiedenen Virtualisierungstechnologien und Verfahren ist es schwierig für den Entwickler, eine Anwendung zu entwerfen, die mit allen Schnittstellen umgehen kann. Libvirt bietet eine einheitliche Schnittstelle (API) um verschiedene Technologien wie Linux KVM[18], Xen[19] und VMware ESX[20] zu verwalten.

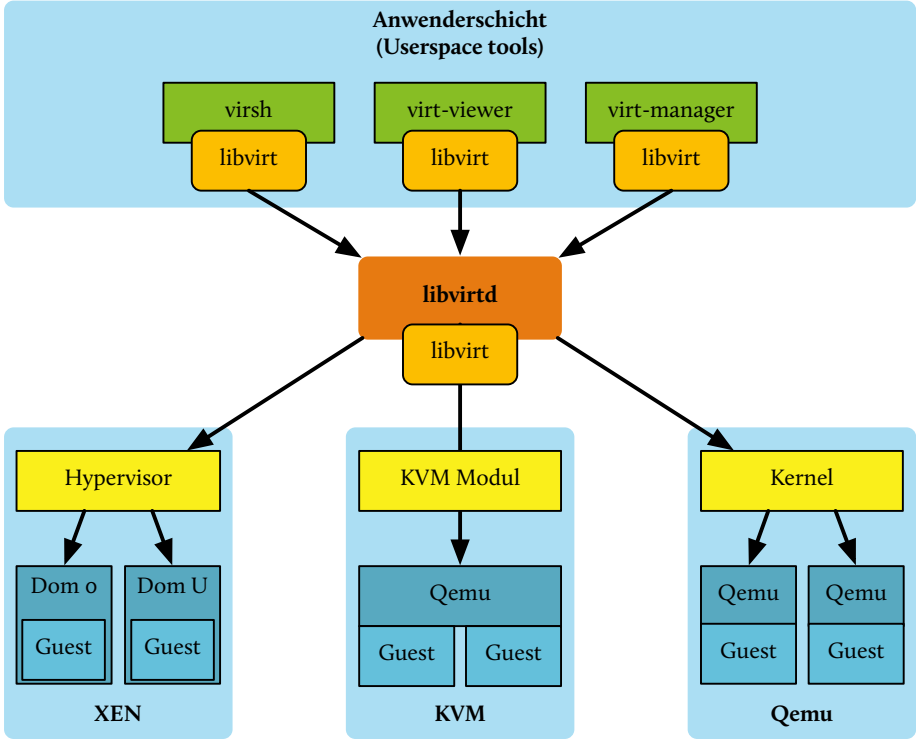


Abbildung 3: Einbindung libvirt Schnittstelle

18. Linux KVM: http://www.linux-kvm.org
19. Xen: http://www.xen.org
20. VMWare ESX: http://www.vmware.com

2.4 Usability und User friendlyness

> “People don't want dump from your app; they want simplicity and ease.”
>
> Josh Clark - iPhone Designer und Entwickler

Usability (Gebrauchstauglichkeit) und User friendlyness (Benutzerfreundlichkeit) sind eng verwandte Begriffe und sollen die Nutzbarkeit und Zufriedenheit für den Kunden erhöhen. So kann durch [ISO 9126] (Benutzbarkeit) und [ISO 9241] (Gebrauchstauglichkeit) folgende Formel abgeleitet werden:

```
Benutzbarkeit = Effektivität + Effizienz
Gebrauchstauglichkeit = Benutzbarkeit + Zufriedenstellung
Benutzerfreundlichkeit = Gebrauchstauglichkeit + Zufriedenheit + Nützlichkeit
```

Zudem lassen sich folgende fünf Attribute für den Begriff Usability definieren [nie93]:

- Effektivität
  Das System sollte leicht zu erlernen sein, damit der Benutzer möglichst schnell seine Arbeit erledigen kann.
- Effizienz
  Das System sollte effizient genutzt werden können, damit der Benutzer produktiv damit arbeiten kann, nachdem er es erlernt hat.
- Einprägsamkeit
  Das System sollte sich dem Benutzer leicht einprägen, damit er auch nach einer längeren Pause, in der der Benutzer das System nicht verwendet hat, es wieder verwenden kann ohne es neu lernen zu müssen.
- Fehler
  Das System sollte wenige Fehler vom Benutzer zulassen, Fehler sollten sich leicht beheben lassen und schwerwiegende Fehler sollten gar nicht erscheinen.
- Zufriedenheit
  Das System sollte angenehm bei der Verwendung sein, damit der Benutzer mit dem System zufrieden ist.

In dieser Arbeit bezieht sich der Begriff Usability nur auf die Gebrauchstauglichkeit von mobilen Anwendungen, nicht auf die Usability des Gerätes oder anderer Software.

In diesem Abschnitt werden die bekanntesten Usability Regeln aufgeführt und die Relevanz für mobile Anwendungen dargestellt. Die eigentliche Anwendung der Regeln und Usability Konzepte erfolgt in Kapitel *Analyse*.

2.4.1 Usability-Regeln

Die große Anzahl an verschiedensten Usability-Regeln erschwert es die relevanten Regeln zu erkennen. Einige der seit Jahren anerkannten Regeln werden kurz erklärt und bewertet.

Handlungsschritte nach Norman

Schon 1988 veröffentlichte Donald Norman sein Buch „The Design of Everyday Things“, [norman88] welches sieben Handlungsschritte zum Erreichen eines Ziels beschreibt.

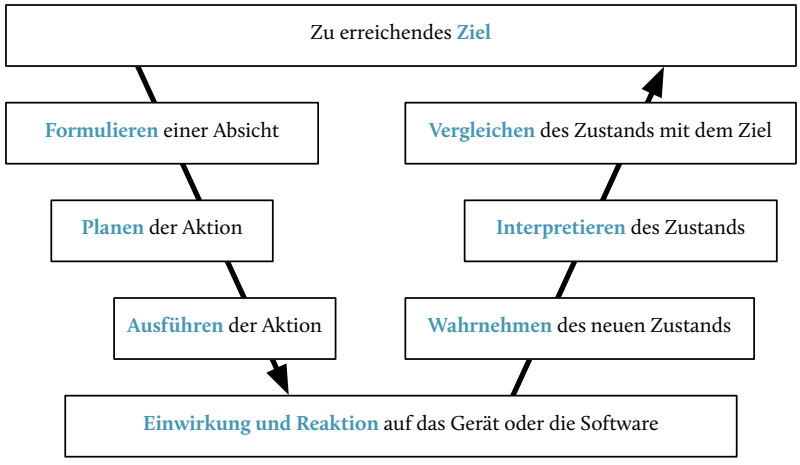


Abbildung 4: Die sieben Handlungsschritte von Norman [dahm06]

Aus den Handlungsschritten lassen sich zwei Handlungsphasen ableiten, die auftreten, wenn es für den Benutzer zu Problemen kommt das Ziel zu erreichen:

- Der Gulf of Execution beinhaltet Probleme, die während der Formulierung, Planung oder Ausführung auftreten. Diese treten auf, wenn der Benutzer nicht weiß, wie er zu der Lösung seines Problems kommt. Zu diesen Problemen gehört ein unverständliches Bedienkonzept oder eine mangelhafte Navigation.
- Der Gulf of Evaluation stellt die Probleme zwischen Anzeige und Interpretation dar. Diese entsteht durch Fehlinterpretation, mangelhafte Darstellung oder Rückmeldung.

Die sieben Handlungsschritte nach Norman sind durchaus wichtig für die Entwicklung einer mobilen Anwendung. So ist zu beachten das der Benutzer stets eine Rückmeldung auf seine Aktion erhält. Ebenso soll ein, in sich stimmiges, Bedienkonzept existieren, da sonst die Unzufriedenheit des Benutzers erhöht wird. Im schlimmsten Fall würde der Benutzer die Anwendung nicht mehr verwenden oder neue Benutzer vom Kauf der Anwendung abhalten. [norman88]

Usability-Heuristik von Nielsen

Der Software-Ergonom Jakob Nielsen verfasste 1990 unter dem Titel „Improving a Human-Computer Dialogue“ zehn Prinzipen, welche als Kategorien für die Usability-Probleme dienten. Heutzutage findet diese Heuristik Anwendung in der Gestaltung von neuen Benutzerschnittstellen.

- Einfache und natürliche Dialoge
  Dialoge sollen nur die nötigsten Informationen beinhalten. Alle weiteren Informationen, die nicht relevant sind, verringern die Wichtigkeit der nötigen Informationen. Die Darstellung soll natürlich und logisch sein.
- Ausdrucksweisen des Anwenders
  Die Dialoge sollen in einer, dem Benutzer vertrauten, Sprache geschrieben sein. Fachausdrücke sollten, falls möglich, vermieden werden.
- Minimale mentale Belastung des Benutzers
  Der Benutzer soll sich keine Informationen über mehrere Dialoge hinweg merken müssen.
- Konsistenz
  Die Dialoge sollen immer auf die gleiche Art und Weise dargestellt werden. Der Benutzer soll sich während der Verwendung nicht umgewöhnen müssen.
- Rückmeldung
  Die Anwendung sollte dem Benutzer immer, in einer vernünftigen Zeit, informatives Feedback liefern. Siehe auch Handlungsschritte nach Norman.
- Kontrolle über die Funktionen
  Sollte der Benutzer eine falsche Funktion ausgeführt haben, muss er diesen Vorgang jederzeit abbrechen oder einen Schritt zurückgehen können.
- Abkürzungen
  Das System soll dem erfahrenen Benutzer Abkürzungen, z.B. durch Tastenkürzel oder Funktionstasten, bieten können. Somit ist gewährleistet, dass erfahrene und unerfahrene Benutzer das System effizient nutzen können.
- Gute Fehlermeldungen
  Eine Fehlermeldung soll nie den Benutzer kritisieren, sie sollte defensiv, präzise und konstruktiv sein.

- Fehlervermeidung
  Fehler sollten durch das Design und die Benutzereingabe weitestgehend vermieden werden.
- Hilfe und Dokumentation
  Auch wenn das System ohne Hilfe und Dokumentation benutzbar ist, sollte eine Dokumentation und Hilfe immer zur Verfügung stehen.

Für die Gestaltung von neuen Benutzerschnittstellen bieten die Usability-Heuristiken von Nielsen eine gute Grundlage. Sie unterstützten die Entwicklung und das Design einer mobilen Anwendung. [nie93] [niemo90]

KISS - Keep it Simple and Straightforward

Bei dem KISS-Prinzip soll immer eine einfache Lösung für das Problem gewählt werden. Dabei ist KISS ein Akronym, welches auch unter „Keep It Simple and Stupid“ oder „Keep It Short and Simple“ bekannt ist.

Abbildung 5: KISS Prinzip anhand einer iPhone Anwendung [tapworthy11]

Abgeleitet auf die mobile Anwendung bedeutet dies, dass die Benutzerinteraktion und Oberfläche so einfach wie möglich gestaltet werden muss, um das Ziel zu erreichen. Ein Auswahlmenü sollte somit nicht 20 Funktionen anzeigen, sondern nur diese die er zur Lösung seiner Aufgabe benötigt.

2.4.2 Relevanz für mobile Anwendungen

Durch die Besonderheit des mobilen Geräts gegenüber herkömmlichen Computern müssen spezielle Usability-Regeln bei der Entwicklung beachtet werden. So bietet ein kleinerer Bildschirm und Touchscreen andere Eingabemöglichkeiten für den Benutzer. Die Umgebung in welcher die Anwendung eingesetzt wird spielt auch eine tragende Rolle für die Usability.

Umgebung

Es sollte immer bedacht werden, dass eine mobile Anwendung gerade Unterwegs zum Einsatz kommt. iPhones werden meist mit einer Hand verwendet und es wird nur mit einem kurzen Blick auf den Bildschirm geschaut. Wie die *Abbildung 6* zeigt, ist eine Anwendung überladen und es werden zu viele Informationen dargestellt. Dem Benutzer ist es nicht möglich all diese Informationen aufzunehmen.

Abbildung 6: iPhone Anwendung, Nutzung Unterwegs [tapworthy11]

Durch die unterschiedlichen Lichtverhältnisse in der Umgebung muss auf ein starkes Kontrastverhältnis geachtet werden. Die Anwendung kannst sonst bei starken Spiegelungen oder Lichteinfall nicht erkennbar sein.

Bildschirmgröße

Der Bildschirm bei einem iPhone oder einem anderen Smartphone ist deutlich kleiner gegenüber einem Computer. Es ist daher kein Platz für unnötige Informationen die den Benutzer auf dem aktuellen Bildschirm nur irritieren könnten.

Durch den kleineren Bildschirm können weniger Informationen auf einem Ausschnitt platziert werden, dadurch muss mit einer guten Navigation gearbeitet werden. Diese sollte nicht zu Tief „verwinkelt“ sondern für den Benutzer intuitiv nutzbar sein.

Touchscreen

Da ein Touchscreen meist mit den Fingern und nicht mit der Maus bedient wird, sind einige Designaspekte zu beachten.

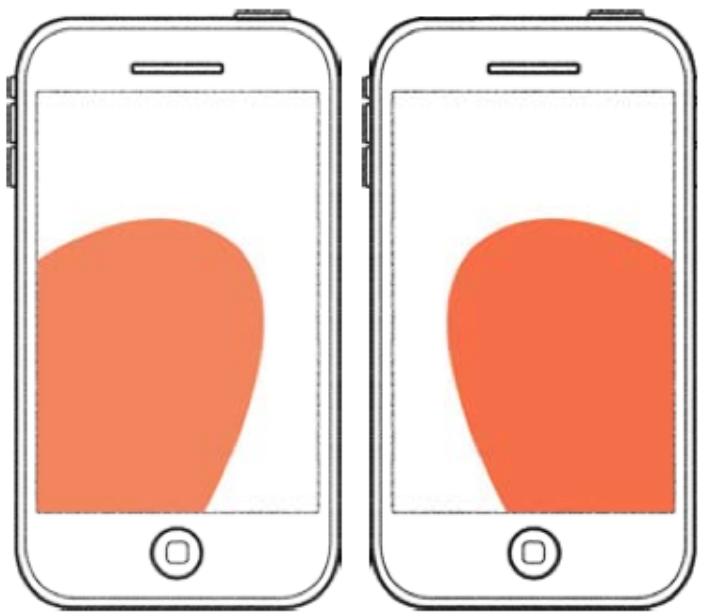


Abbildung 7: Der Daumen bedeckt einen Großteil des Bildschirms [tapworthy11]

Die Finger oder der Handrücken überdecken, wie *Abbildung 7* zeigt, bei der Navigation zum Teil die Anwendung. Die Hauptnavigation innerhalb einer Anwendung sollte daher am oberen Rand des Bildschirms angezeigt werden. Am unteren Rand sollten sich Funktionstasten für die Anwendung befinden.

Mit der vollwertigen QWERTZ-Tastatur, gegenüber der T9-Tastatur, ist es zwar einfacher Eingaben vorzunehmen aber es ist dennoch mühsam. Für den Benutzer sollte man daher die Text-Eingaben so gering wie möglich halten.

3 Analyse

In diesem Kapitel erfolgt die Analyse zur Entwicklung der mobilen Anwendung.

> “ Plan your work and work your plan. ”
>
> Vince Lombardi

Die ersten Abschnitte beziehen sich auf die Projektplanung, wie die Definition der Zielgruppe und Ist-Zustandsanalyse. In den darauffolgenden Abschnitten wird die Konzeption und Identität anhand von Navigationskonzepte, Design eines Logos und Apple Design verdeutlicht. Gefolgt werden diese von der Strukturierung der Anwendung durch ein Story-Board und Usability-Tests anhand einiger Vorlagen.

3.1 Zielgruppe

Die Definition der Zielgruppe und die Abfragen der Interessen der Anwender ist entscheidend für die Realisierung einer guten Anwendung.

3.1.1 Brainstorming

Um die Zielgruppe und deren Anforderungen besser spezifizieren zu können, wurde in einem Meeting allen Mitglieder der Abteilung, ein Brainstorming abgehalten. Durch dieses Verfahren war es einfacher möglich eine genaue Ziel-Definition zu erstellen.

Mit der in *Abbildung 8* erstellten Mindmap konnte eine Definition abgeleitet werden. Für eine bessere kreative Arbeit wurde das Brainstorming aber mit Papier und Stift erstellt. Das Original befindet sich in der Anlage.

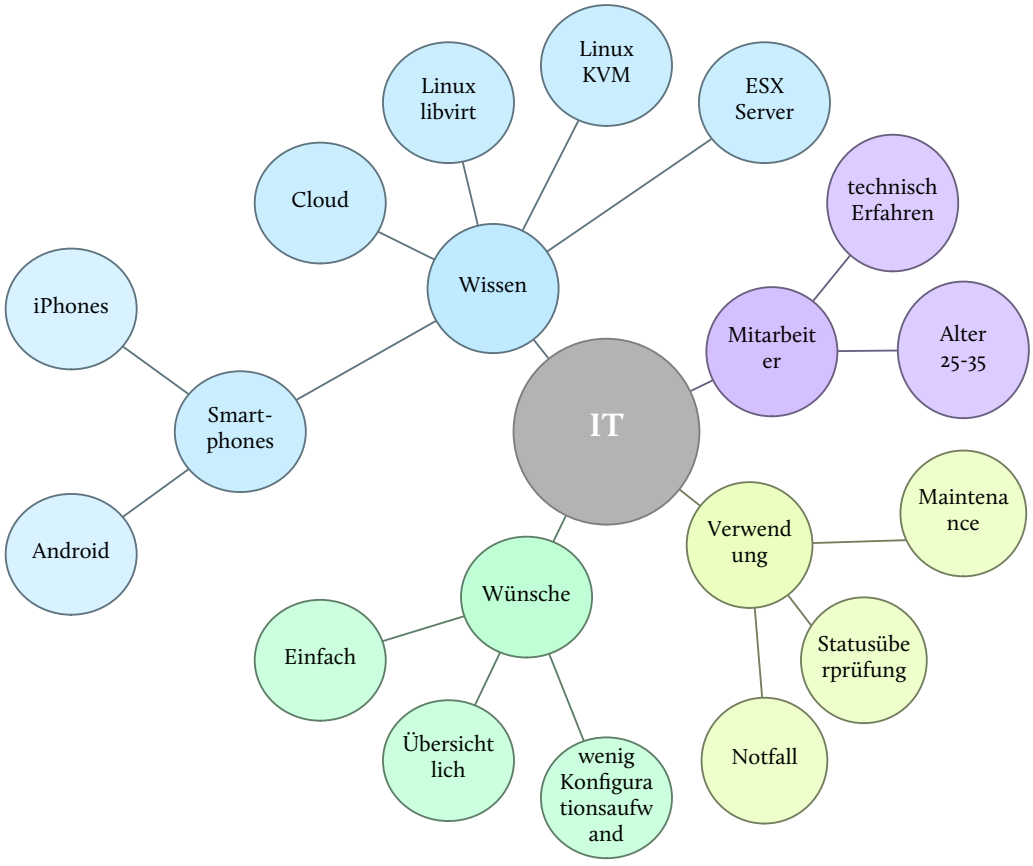


Abbildung 8: Brainstorming mit und über die Zielgruppe

3.1.2 Definition

Die Zielgruppe für das Projekt umfasst Mitarbeiter einer IT Abteilung. Diese Mitarbeiter sind erfahrene IT Administratoren für die Betriebssysteme Linux und Windows. Sie sind vertraut mit der Handhabung eines Smartphones, insbesondere iPhones.

Den Benutzern ist die Verwendung von virtuellen Servern in einer Cloud-Landschaft bekannt. Die Technologie libvirt und Linux KVM im Zusammenhang mit Hardware-Virtualisierung ist ein Begriff.

3.2 Ist-Zustandsanalyse

Zum jetzigen Zeitpunkt existiert noch keine mobile Anwendung (iPhone Anwendung) zur Verwaltung der Cloud Infrastruktur.

Zum Einsatz im Unternehmen kommt Citrix CloudStack [21]. Hierbei handelt es sich um eine OpenSource Webanwendung zur Verwaltung einer Infrastruktur Cloud. Zwischen der Weboberfläche und der Virtualisierungstechnologie befindet sich die libvirt-Schnittstelle. Nach *Abbildung 3 Einbindung libvirt Schnittstelle*, befindet sich CloudStack in der Anwenderschicht. Durch diese Schnittstelle ist es möglich verschiedene Virtualisierungstechnologien zu nutzen.

Eine zufriedenstellende Verwendung von CloudStack ist nur in einem Browser mit ausreichend großem Bildschirm möglich. Auf einem Smartphone ist die Vielzahl an Funktionen unübersichtlich und im Notfall nicht verwendbar.

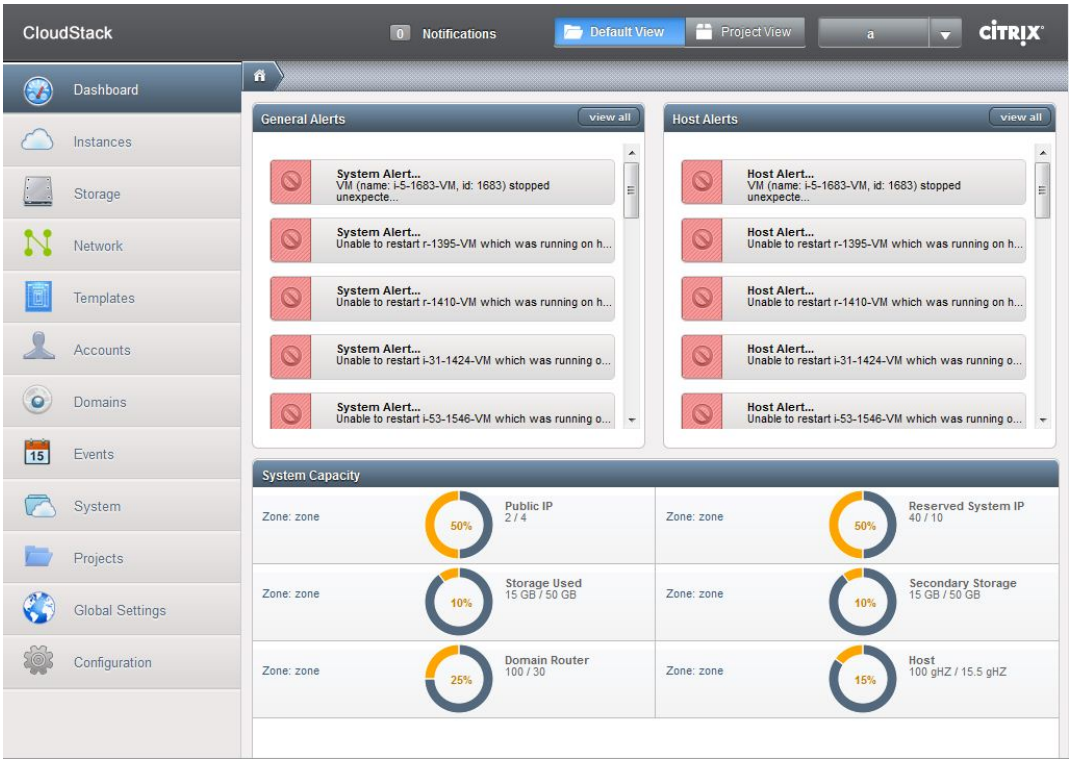


Abbildung 9: CloudStack Dashboard [cstack12]

21. Citrix CloudStack: http://www.citrix.de/produkte/cloudstack/

Da durch CloudStack auf die libvirt-Schnittstelle aufgesetzt wird, ist es sinnvoll diese auch für die Entwicklung der mobilen Anwendung zu verwenden.

3.3 Anwendungsfunktionen

Um die Aufgabe der Anwendung besser definieren zu können folgt ein Aufstellung der Funktionen die geboten werden sollen. Die genauen Schritte für den Benutzer werden in der Konzeptionsphase und im Story-Board beschrieben.

- Cloud Server
  Beim Cloud Server handelt es sich um das Host-System der virtuellen Server. Der Cloud Server stellt dem virtuellen Server Rechnerressourcen zur Verfügung.
  - Hinzufügen (Anzeigename, Hostname, Port, Benutzername, Kennwort)
  - Bearbeiten
  - Löschen
- Virtueller Server
  Virtuelle Server sind Gast-Systeme auf einem Cloud Server. Der virtuelle Server erhält Rechnerressourcen vom Cloud Server.
  - Hinzufügen (Anzeigename, Rechnerressourcen)
  - Bearbeiten
  - Löschen
  - Anzeige Rechnerressourcen
  - Bearbeiten der Rechnerressourcen
  - Starten
  - Stoppen
  - Neustarten

Die veränderbaren Rechnerressourcen werden wie folgt definiert:

- Anzahl der CPUs
- Verwendbarer Arbeitsspeicher
- Angebotener Festplattenspeicher
- Aktivierung oder Deaktivierung von Remote Verbindungen

Beispielsweise, befindet sich der Administrator gerade nicht an seinem Arbeitsplatz, hat aber sein iPhone dabei. Durch die weitere Anwendung „Nagios“, für das Überwachen der virtuellen Server, erhält er den Hinweis, dass kein Arbeitsspeicher auf einem Server verfügbar ist. Mit der zu entwickelnden Anwendung kann der Administrator zum Beispiel per Schieberegler den Arbeitsspeicher eines virtuellen Server erhöhen.

3.4 Alternative Anwendungen

Da aktuell noch keine Anwendungen für die Verwaltung von virtuellen Rechnerinstanzen existieren werden alternative IT-relevante Anwendungen betrachtet. Ein Teil der Erfahrungen mit diesen Anwendungen wird in der Realisierungsphase nützlich sein.

Die Anwendungen werden kurz Beschrieben und die grafischen Benutzeroberflächen bewertet.

3.4.1 iSSH

Die Anwendung „iSSH - SSH / VNC Console“ bietet IT Administratoren die Möglichkeit sich mit einer Remote Verbindung auf einen Server zu verbinden und diesen zu verwalten. Unterstützt werden viele verschiedenste Protokolle, wie SSH[22] und Windows Remote Desktop[23].

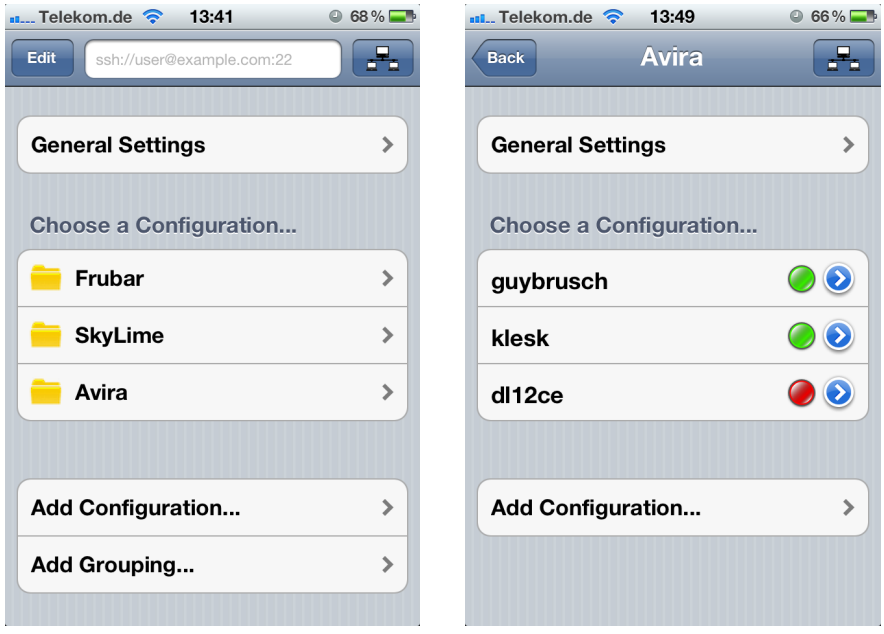


Abbildung 10: iSSH Kategorisierung und Übersicht der Server

22. SSH: http://openssh.org
23. Windows Remote Desktop: http://support.microsoft.com/?scid=kb%3Ben-us%3B186607&x=13&y=11

Das sortieren von Servern in verschiedene Kategorien bzw. Ordner bietet eine hohe Übersichtlichkeit. Durch grüne oder rote Punkte wird Angezeigt ob ein Server erreichbar ist. Somit hat der Benutzer eine direkte Rückmeldung, ob er sich zum Server verbinden kann.

Einige Nachteile bieten die vielen Einstellungsmöglichkeiten auf der Übersichtsseite (*Abbildung 10*, linkes Bild). Die „General Settings“, „Add Configuration“ und „Add Grouping“ wären an der Unterseite des Bildschirms oder in den iPhone Einstellungen besser aufgehoben.

3.4.2 IPMI Touch

Bei „IPMI Touch“ handelt es sich um ein weiteres IT Verwaltungswerkzeug von Servern. Es liest Hardwareinformationen eines Servers über das IPMI Protokoll[24] aus. Für diese Informationen werden Temperatursensoren des Gehäuses ausgelesen, die Geschwindigkeit der Lüfter und der aktuelle Stromverbrauch. Über die Anwendung ist es dem Administrator auch möglich den Server auszuschalten.

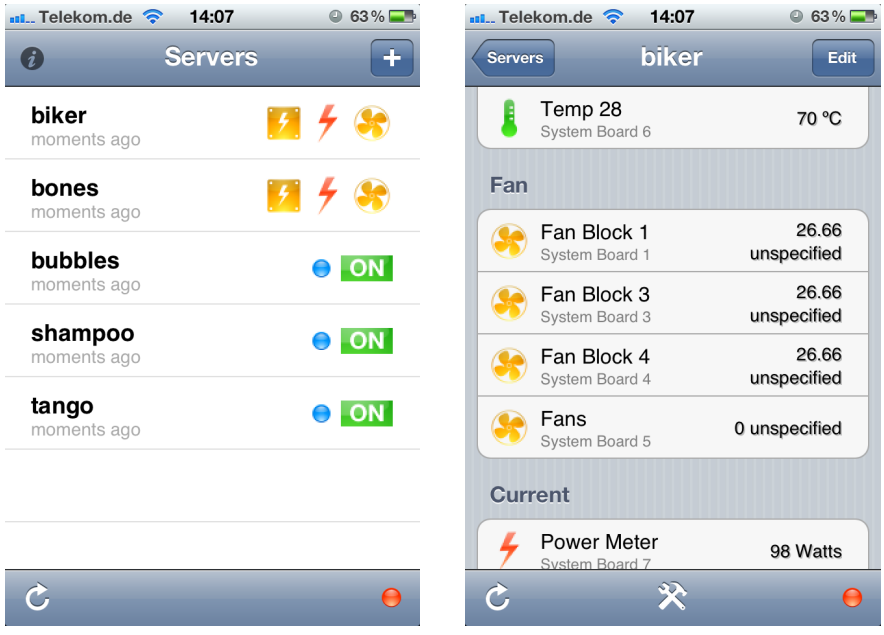


Abbildung 11: IPMI Touch Übersicht und Hardwareinformationen

24. IPMI: http://www.intel.com/design/servers/ipmi/

Auf der Übersichtsseite der Anwendung ist durch die farblichen Unterschiede und Symbole sofort erkennbar bei welchem Server Probleme vorhanden sind. Klickt der Benutzer auf einen Menüpunkt erhält er eine detaillierte Anzeige der Hardware, in der ebenso farblich Probleme gekennzeichnet sind. Hinzufügen neuer Server erfolgt über das Plus („+“) in der oberen rechten Ecke des Bildschirms und braucht somit keinen Platz in der Anzeige.

Server lassen sich bei dieser Anwendung nicht in Kategorien oder Ordner einsortieren, dies erschwert bei einer großen Anzahl an System die Übersichtlichkeit.

3.5 Situationen für mobile Anwendungen

Es gibt viele verschiedene Situationen in denen eine iPhone Anwendung Verwendung findet. Laut Josh Clark, iPhone Designer und Entwickler, können diese in drei Situationen zusammengefasst werden. [tapworthy11]

3.5.1 „Meine minimale Aufgabenverwaltung“

Ein Großteil der Anwendungen in dieser Kategorie sind zur Terminplanung, Aufgabenverwaltung oder Administration. Sie dienen der Produktivitätssteigerung und somit zum effektiven lösen von Problemen.

Abbildung 12: Screenshot des Kalender und Things „App“

Die Anwendungen „Kalender“ oder „Things“, die zur Verwaltung von Terminen und Aufgaben dienen, sind zum Anlegen von neuen Terminen oder Aufgaben optimiert. Hierfür befindet sich auf jedem Bildschirm ein Plus („+“), wie in der *Abbildung 12* zu sehen.

Die mobile Anwendung zur Verwaltung der Cloud Infrastruktur fällt in diese Kategorie.

3.5.2 „Was ist in meiner Umgebung?“

Hierbei handelt es sich hauptsächlich um Anwendungen die auf Kartenmaterial, beziehungsweise auf Sensoren des iPhones, zugreifen. Im Vordergrund der Anwendungen steht die Navigation und das Mitteilungsbedürfnis des Benutzers.

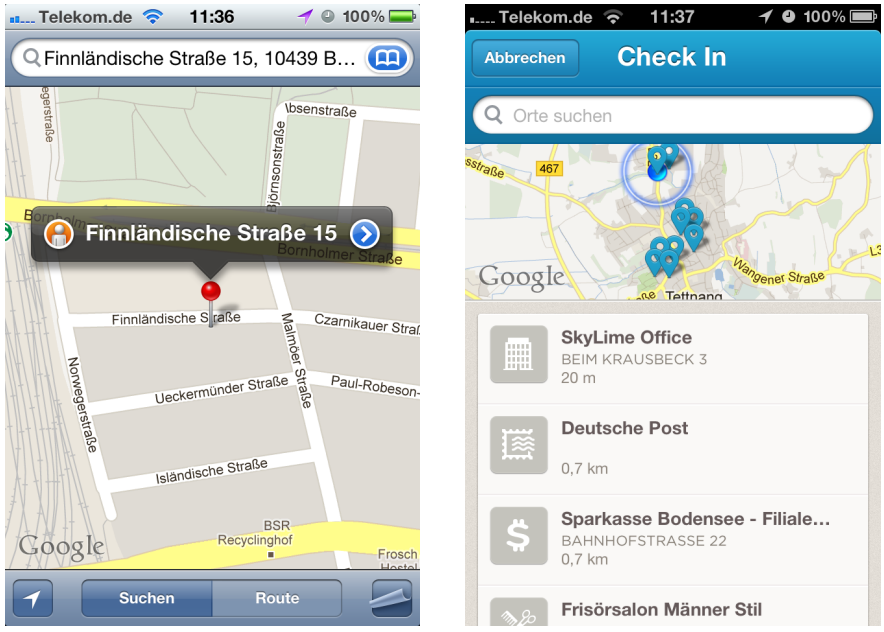


Abbildung 13: Screenshot der Karten und der Foursquare Anwendung

3.5.3 „Mir ist Langweilig“

In diese Kategorie fallen sowohl Spiele als auch Tools wie das „Musik App“. Diese sind auch die beliebteste Kategorie[25] im App Store. Die Beliebtheit wird nach dem Anteil aller verfügbaren „Apps“ im App Store gemessen.

25. Statista, Top 15 Kategorien im App Store (2012): http://de.statista.com/statistik/daten/studie/166976/umfrage/beliebteste-kategorien-im-app-store/

Wie auf dem Screenshot *Abbildung 14* zu erkennen ist, hat das Spiel „Angry Birds“ keine iPhone Navigationselemente. Spiele nehmen meist den gesamten Bildschirm ein und verwenden ihre eigenen Symbole für die Navigation. Sogar die Statusbar mit der Empfangsanzeige, Uhrzeit und der Batterieanzeige werden ausgeblendet.

Abbildung 14: Screenshot des Spiels Angry Birds

4 Konzeption

> “ Structuring your app the Apple way. ”
>
> Josh Clark

Für die Entwicklung einer iPhone Anwendung sollten grundlegende Punkte, wie die Struktur und die Benutzerinteraktion, beschrieben werden. Dies erfolgt in diesem Kapitel.

4.1 Navigationsmodelle

Der Aufbau der Navigation hängt von der Aufgabe der Anwendung ab, wie der vorherige Abschnitt *Situation für mobile Anwendungen* zeigt. Apple bietet hierzu drei verschiedene Navigationsmodelle an. Diese können sich innerhalb einer Anwendung nicht ändern, daher sollte man sich, vor der Entwicklung, für eines der folgenden Modelle entscheiden.

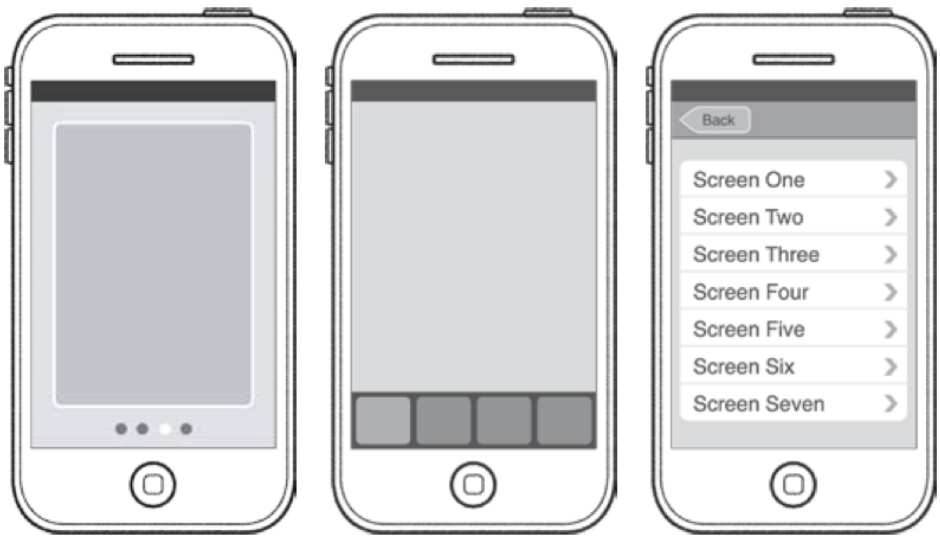


Abbildung 15: Navigationsmodelle (von links) - Ebenen, Tab-Navigation, Baum-Struktur

4.1.1 Ebenen

Die einzelnen Ebenen lassen sich am besten mit Spielkarten, die sich auf einem Stapel befinden, vergleichen. Auf allen Ebenen ist der Inhalt meist im selben Stil aufbereitet, somit findet dieses Navigationsmodel meist bei Tools und kleinen Anwendungen Verwendung.

Abbildung 16: Screenshot der Wetter „App“

Für die Navigation befindet sich ein „Punkt“ am unteren Bildschirmrand. Über diesen ist ersichtlich auf welcher Ebene man sich gerade befindet. Die Navigation erfolgt durch das verschieben der Ebenen von links nach rechts mit einem Finger.

Vorteile	Nachteile
• Ideal für zielgerichtet Inhalt; geeignet für gelegentliches verwenden der „App“ • Einfache Navigation; Gestengesteuert • Viel Platz für den Inhalt; Navigation nimmt wenig Platz im Bildschirm ein	• Man muss durch alle Ebenen durchblättern; der Sprung zu einer speziellen Ebene ist nicht möglich • Mehr als 20 Ebenen können nicht angezeigt bzw. erstellt werden • Nur geringes Scrollen möglich; somit nur teilweise langen Inhalte darstellbar

Tabelle 1: Vorteile und Nachteile für das Navigationsmodell Ebenen

4.1.2 Tab-Navigation

Am unteren Rand des Bildschirms befindet sich eine Tab-Navigation, ähnlich die eines Browsers. Im Gegensatz zu den „Tabs“ im Browser sind diese fest in der Anwendung verankert. Sie bietet auch maximal Platz für fünf Symbole.

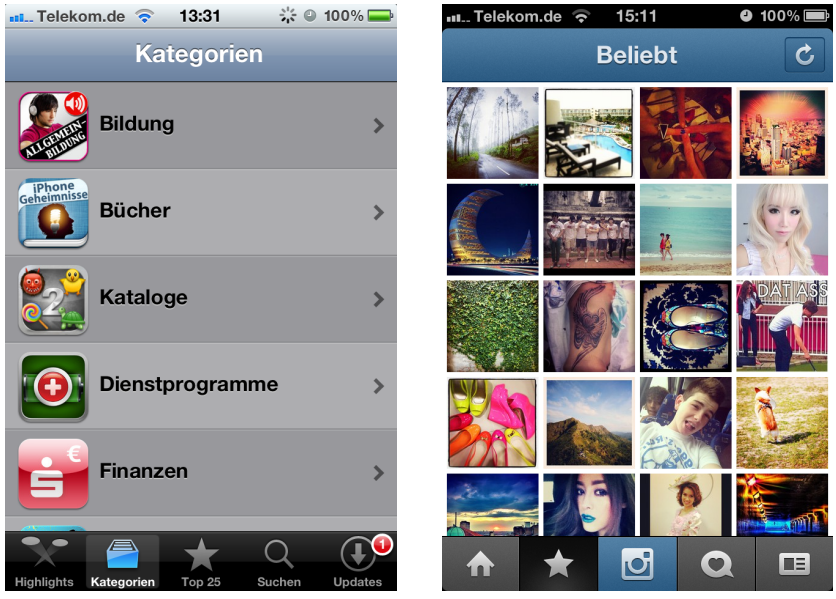


Abbildung 17: Screenshot des App Stores und der Instagram „App“

In der Tab-Navigation befinden sich meist Menüpunkte auf die der Anwender sehr schnell oder häufig zugreifen muss. So bietet sie bei der Anwendung „App Store“ die Suchfunktion, Kategorien oder verfügbare Updates an.

Vorteile	Nachteile
• Sofortigen Zugriff auf die Hauptfunktionen der Anwendung • Klar erkennbare Menüpunkte anhand Symbol und Beschreibung	• Nur fünf Menüpunkte können gleichzeitig angezeigt werden • Navigation verbraucht sehr viel Platz

Tabelle 2: Vorteile und Nachteile für die Tab-Navigation

4.1.3 Baum-Struktur

Die Baum-Struktur ist eines der häufig benutzten Navigationsmodelle. Gerade wenn viele Menüpunkte oder Ebenen benötigt werden kommt diese Struktur zum Einsatz. Eine sorgfältige Planung, welche Ebene nach welchem Menüpunkt kommt, ist zwingend erforderlich. Andernfalls kann es schnell sein, dass sich ein Benutzer nicht mehr in der Struktur zurecht findet.

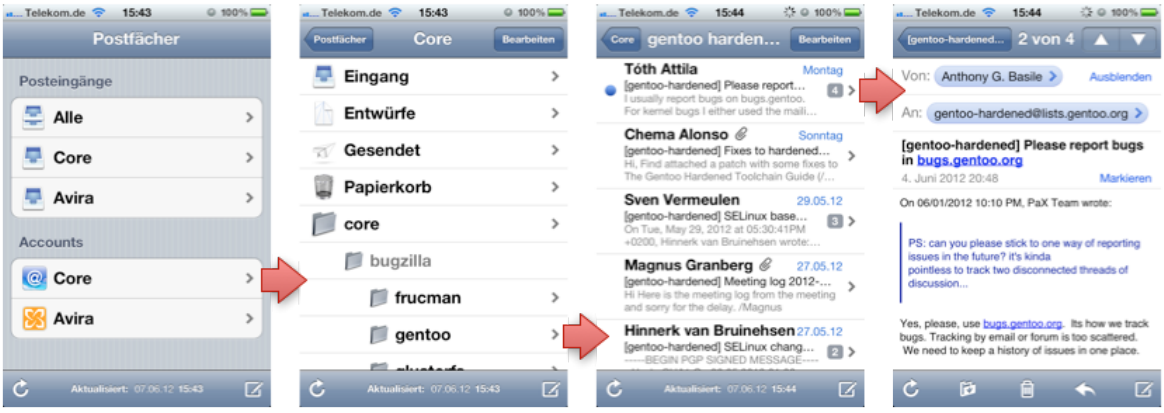


Abbildung 18: Screenshot Baum-Struktur von Apple Mail

Bei diesen Anwendungen ist meist die erste Ebene eine Übersichtsebene in Tabellenform, wodurch dann zu weiteren Ebenen navigiert werden kann. Im obersten Teil des Bildschirms befindet sich die Navigation um zur vorherigen Ebene zurückzukehren.

Vorteile	Nachteile
• Übersichtlich bei vielen Kategorien, Ebenen oder Menüpunkten • Organisationsstruktur ist gut verständlich • Direkte Interaktion mit dem Inhalt	• Übersichtsebene befindet sich nur am Anfang der Ebenen • Ineffiziente Navigation bei vielen Ebenen

Tabelle 3: Vorteile und Nachteile für das Baum-Struktur Navigationsmodell

4.1.4 Kombination der Navigationsmodelle

Meist findet eine Kombination aus zwei Navigationsmodellen statt, wie die Abbildung 18 von Apple Mail zeigt. Durch dies können mehrere Vorteile der unterschiedlichen Modelle genutzt werden.

Bei der Apple Mail Anwendung wird die „Tab-Navigation“ verkleinert, so das nur Symbole angezeigt werden. Hierbei handelt es sich um eine abgewandelte Form der Navigation und kann als Aktions-Buttons verstanden werden. Dem Benutzer ist es dadurch möglich schnell auf seine wichtigsten Funktionen zuzugreifen ohne einen Großteil seines Bildschirms einzuschränken.

4.2 Darstellung des Inhalts

Ein klar strukturierter Inhalt erlaubt dem Benutzer auf die benötigten Informationen zuzugreifen. Ist die Darstellung unübersichtlich oder überladen beendet der Anwender die „App“ im schlimmsten Fall. Es sollte daher immer ein Kompromiss zwischen der Masse an Informationen und der Bildschirmgröße gefunden werden.

In vielen Fällen findet eine Kombination aus der Tabellendarstellung und der Benutzereingabe durch Formulare statt.

4.2.1 Tabellen

Die Tabellen Darstellung beim iPhone kann nicht zu 100% mit klassischen Tabellen verglichen werden. Die Informationen werden meist zeilenweise gruppiert und es wird nur eine Spalte angezeigt.

Aus folgenden Gründen kommt die Darstellung in Tabellenform zum Einsatz:

- Navigation durch die Baum-Struktur
- Liste an Optionen oder Einstellungen
- Lange Listen durch die schnell navigiert werden soll
- Detaillierter oder gruppierter Inhalt

Abbildung 19: Screenshot verschiedener Anwendungen mit Tabellen Darstellung

Durch bestimmte Symbole an der rechten Seite ist erkennbar ob es sich um ein Navigationselement handelt. Die Schriftfarbe blau zeigt an, ob sich der Inhalt des Elements ändern lässt oder ggf. dynamisch ändert.

Drei der Standart Symbole sind wie folgt erklärt:

	Weitere Informationen Symbol; durch das Tippen auf das Element wird eine neuer Bildschirm mit weiteren Informationen für den Benutzer sichtbar.
	Details Symbol; ähnlich dem „Weitere Informationen Symbol“, beim direkten Tippen auf das Symbol sollen weitere Details zu diesem Element angezeigt werden.
	Check-Box Symbol; durch Tippen auf das Element wird die Check-Box aktiviert oder deaktiviert.

Tabelle 4: iPhone Standart Symbole

Die Gruppierung der Elemente geschieht entweder durch ein Register (wie bei *Abbildung 19*, links) oder durch farblich, zum Hintergrund abgehobene, Boxen (*Abbildung 19*, mitte und rechts).

4.2.2 Formulare

Die Formulare dienen dem Benutzer zur Eingabe von neuem Inhalt in die Anwendung. In dem meisten Fällen wir die Benutzereingabe durch die Auto-Korrektur unterstützt.

Es ist Abhängig vom Eingabefeld welches Tastaturlayout für den Benutzer angezeigt wird. So wird bei üblicher Texteingabe die QWERTY-Tastatur angezeigt, beim Wählvorgang in der „Telefon-App“ jedoch nur der Nummernblock.

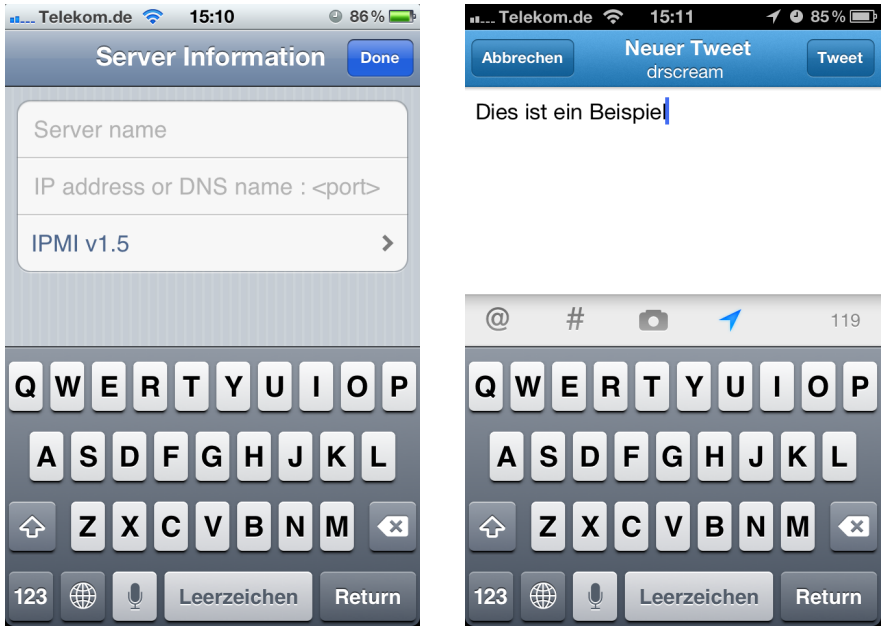


Abbildung 20: Screenshot von IPMI Touch und Twitter „App“ mit Eingabeformular

4.3 Identität

Die Identität einer Anwendung ist enorm wichtig für den Benutzer und dessen Wiedererkennungswert. Diese werden durch den Namen, eine klare Beschreibung, das Logo und Aussehen bewirkt.

Dieser Abschnitt bezieht sich explizit auf die Umsetzung der mobilen Anwendung zur Verwaltung der Cloud Infrastruktur.

4.3.1 Name

Bei der Wahl des Namens ist zu beachten, dass nur begrenzt Platz auf dem Bildschirm und unterhalb des Logos ist. Da Apple zwischen dem Namen auf dem iPhone und dem im App Store unterscheidet, können zwei unterschiedliche Namen gewählt werden. So sollte der Name für das iPhone kurz und prägnant sein, der Name für den App Store mehr Beschreiben.

Namen die auf dem iPhone zu lang sind, werden automatisch in der Mitte gekürzt. Dadurch wird aus „Cloud infrastructure control via libvirt“, „Cloud...bvirt“, was für den Benutzer nicht aussagekräftig ist. Von vielen Anwendungen wird daher ein allgemeiner Name auf dem iPhone verwendet, so wird aus „Dragon Dictation“ nur „Dictation“ oder aus „Al Jazeera English“, „AJE Live“.

Folgende Kurznamen standen während der Identitätsbildung zur Auswahl:

- Cloud Control
  Aussagekräftiger Namen für die Verwaltung der Cloud. Die verwendete Technologie ist aber nicht erkennbar.
- libvirt Control
  Verwaltung der verwendeten Technologie ist erkennbar.
- libvirt Cloud
  Verwaltung der verwendeten Technologie ist erkennbar. Aber nicht das es sich um eine Anwendung zur Verwaltung dieser handelt.
- InfraCloud
  Durch „Infra“ ist teilweise die Infrastruktur Cloud ersichtlich.
- virtCloud
  Die Worte „Cloud“ und „virt“ für Virtuell können fast gleich gesetzt werden. Daher ist die Beschreibung redundant.

Nach Absprache und Analyse mit den Mitarbeitern der IT Abteilung wurde der Name „Cloud Control“ als aussagekräftigster Name ausgewählt. Die verwendete Technologie kann in der Beschreibung oder im Namen für den App Store angezeigt werden. Als Name für den App Store wird daher „Cloud Control – manage your libvirt infrastructure“ verwendet.

4.3.2 Logo

Das Logo ist ein weiteres Identitätsmerkmal für die Anwendung, wenn nicht sogar das wichtigste. Der erste Blick des Benutzers fällt meist auf das Logo anstatt auf den Namen. Es sollte entweder die Funktion oder den Namen im Logo wiedererkennbar sein. Bekannte Anwendungen oder Unternehmen greifen meist auf ihr Logo in abgeänderter Form oder ihren Initialen zurück.

Abbildung 21: „App“-Logos (von links). Things, Instagram, Cellar Rat, Twitter, Facebook

Wie gut ein Logo erkennbar ist, hängt meist von der Auflösung und der Detailgenauigkeit ab. Ein „App“-Logo muss mindestens für drei verschiedene Auflösungen erstellt werden. So verwendet das iPhone für kleine Logos eine Auflösung von 29x29 Pixel und für mittlere Logos 57x57 Pixel. Ein großes Logo mit 512x512 Pixel wird für den App Store benötigt.

Für die eigene Anwendung soll das Logo die Funktion widerspiegeln und Begriffe wie „Cloud“, „Computer“, „Netzwerk“ und „libvirt“ sollen erkennbar sein. Nach Skizzen, die sich im Anhang befinden, wurde ein Logo entworfen. Im größten Logo wird sich der Begriff „libvirt“ als Zusatz befinden. In den kleineren ist dies nicht mehr lesbar, daher wurde er entfernt.

Abbildung 22: Erster Entwurf. Logos des Cloud Control „Apps“

Nach einigen Befragungen der Mitarbeiter wurde festgestellt, dass das Logo zwar die Begriffe darstellt aber gegenüber anderen Logos langweilig und zu technisch wirkt. Es wurde daher ein modernes Logo entworfen, dass die Begriffe „Cloud“ und „Control“ darstellt.

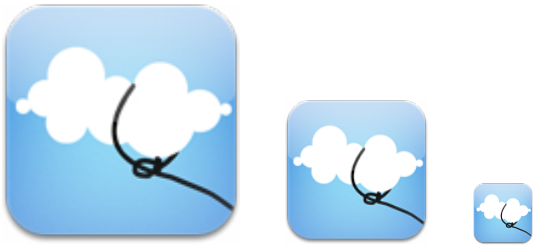


Abbildung 23: Zweiter Entwurf. Logos des Cloud Control „Apps“

4.3.3 Design

Design und äußerliche Erscheinung sind meistens Geschmacksfragen, doch spielt der Erkennungswert und die Benutzerfreundlichkeit auch eine Rolle. Die Navigation und die Anordnung bzw. Strukturierung von Inhalt kann nur teilweise bei iPhone „Apps“ verändert werden.

Einige Anbieter von „Apps“ setzen daher auf das Standart iPhone Design und passen lediglich Farbgebung und Symbole an. Um die Benutzer der „Cloud Control App“ nicht zu verunsichern, sowie Übersichtlichkeit und Einfachheit zu gewährleisten wird ebenfalls das Standart iPhone Design verwendet.

4.4 Strukturierung

Eine gute Struktur der Anwendung ist vergleichbar mit dem „roten Faden“ in einem Manuskript. Ohne diesen ist es dem Benutzer nicht möglich zu erkennen welches Ergebnis auf ein Navigationselement erfolgt.

Abhängig vom verwendeten Navigationsmodel muss die Anwendung anders strukturiert werden. So bietet die Baum-Struktur mehr Möglichkeiten der „Verzweigung“ im Gegensatz zu den flachen Ebenen. Eine Strukturierung erfolgt am besten durch die Skizzen auf Papier oder durch Mockups mit einer Anwendung am Computer.

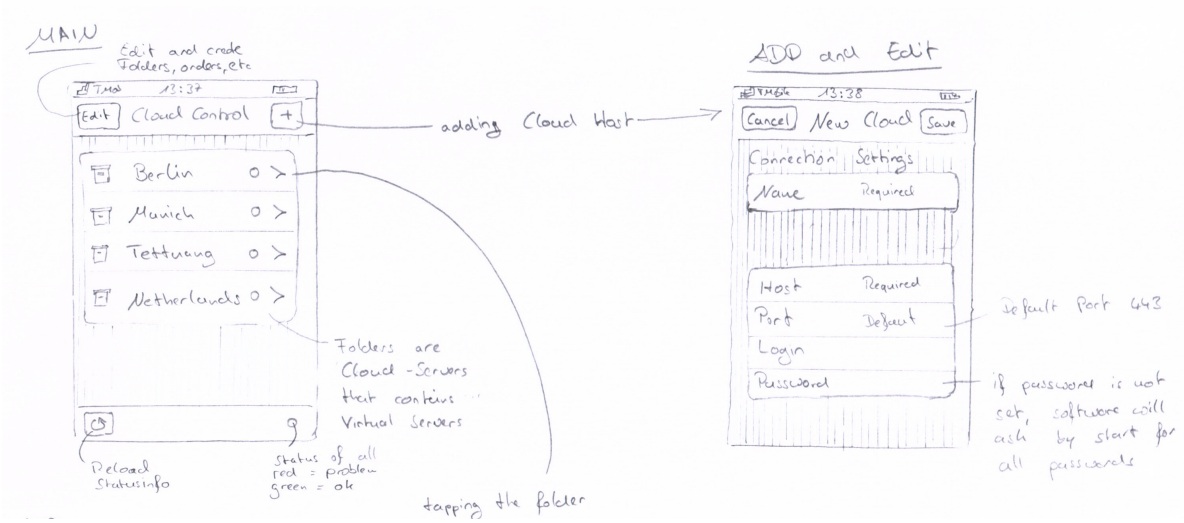


Abbildung 24: Ausschnitt Cloud Control Story-Board

Für diesen Zweck wurde ein „Story-Board“ auf Papier entworfen, dieses befindet sich im Anhang. Mit diesem war es möglich erste Prototypen zu entwerfen und in den Usability-Tests anzuwenden. Das Story-Board wurde während den Tests verändert und erweitert.

4.5 Usability-Tests

Die Usability-Tests dienen dem Entwickler seine Anwendung mehr aus der Sicht des Nutzers zu betrachten. Die Sichtweisen sind meist Unterschiedlich, da der Entwickler seine Anwendung entworfen hat und das Navigationskonzept kennt.

Ziel des Tests ist es Unstimmigkeiten in der Navigation und Handhabung der Anwendung aufzudecken um diese im Anschluss optimieren zu können.

4.5.1 Durchführung

Für den Zweck des Usability-Tests wurde mit drei Mitarbeitern (männlich, Alter 20-30) ein Thinking-Aloud Test[26] durchgeführt. Die Mitarbeiter haben Vorkenntnisse über Smartphones, mobile Anwendungen und kennen das Bedienkonzept des iPhones.

26. Thinking Aloud Test: http://www.useit.com/alertbox/thinking-aloud-tests.html

Jeder Mitarbeiter wird einzeln befragt und erhält während der Testphase Aufgaben die er an Prototypen durchführen soll. Es werden zwei Prototypen betrachtet:

- Prototyp 1: Mock-up auf Papier
- Prototyp 2: Anwendung auf dem iPhone

Die Ergebnisse werden schriftlich fixiert um daraus eine Optimierung ableiten zu können. Es wird expliziert gebeten Ideen und Veränderungsvorschläge einzubringen. Der Test dauert pro Person zwischen 20-40 Minuten.

4.5.2 Szenarien

In diesem Abschnitt werden die Aufgaben der Mitarbeiter beschrieben. Zwischen den Prototypen sind die Aufgaben ähnlich, beim Prototyp 1 können einige Aufgaben nicht vorgenommen werden.

- Start der Anwendung
  Betrachten Sie die Anwendung und die Navigationselemente. Was ist Ihr erster Eindruck? Ist Ihnen das Ziel der Anwendung klar?
- Cloud Server hinzufügen
  Fügen Sie einen neuen Cloud Server hinzu. Wie gehen Sie vor?
- Cloud Server bearbeiten
  Ändern Sie den Namen oder ein anderes Attribut des Cloud Servers. Wie gehen Sie vor?
- Status Abfrage
  Wie erkennen Sie den aktuellen Status Ihrer Server? Wo würden Sie nachsehen?
- Virtual Server hinzufügen
  Fügen Sie einen virtuellen Server zu einem bestehenden Cloud Server hinzu. Wie gehen Sie vor?
- Virtual Server Details
  Lassen Sie sich Details zu einem virtuellen Server anzeigen. Wie viel Speicherplatz wird verwendet?
- Virtual Server bearbeiten
  Ändern Sie ein Attribut eines virtuellen Servers. Fügen Sie z.B. mehr Arbeitsspeicher hinzu. Wo würden Sie dies tun?
- Virtual Server Neustarten
  Starten Sie einen virtuellen Server neu. Wie gehen Sie vor?
- Virtual Server löschen
  Löschen Sie einen virtuellen Server von einem Cloud Server. Wie würden Sie vorgehen?

4.5.3 Ergebnisse

Es wurde festgestellt, dass der Prototyp 1, das Mock-up auf Papier, für den Test ungeeignet war. Somit wurde lediglich Prototyp 2, die iPhone Anwendung, bewertet. Zur Konzeptionsphase und für Erweiterungen wird aber weiterhin das Mock-up verwendet.

Erster Start

Die Anwendung wirkt beim ersten Start leer, da noch keine Cloud Server vorhanden sind. Der Benutzer findet sich daher nur schwer zu recht und fühlt sich mit der neuen Anwendung „allein gelassen“.

Es wurden Tooltips hinzugefügt die beim erster Start und falls noch keine Server vorhanden sind, angezeigt werden.

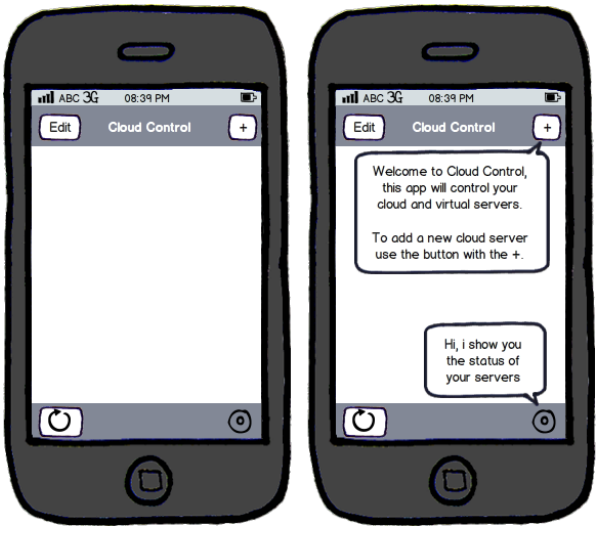


Abbildung 25: Erster Start der Anwendung, Erweiterung durch Tooltips

Navigation

Die Navigation wurde von den Benutzern gut angenommen, sie fanden sich in der Baum-Struktur gut zu recht.

Es stellte sich aber heraus, dass für das Löschen eines virtuellen Servers keine Funktion im Story-Board vorgesehen war. Somit war es dem Benutzer nicht möglich einen virtuellen Server zu löschen. Dies wurde mit einen Editieren-Button bei der Übersichtsseite der virtuellen Server gelöst.

Des Weiteren fiel auf, dass virtuelle Server die Ausgeschalten sind nicht auf der Übersichtsseite erkennbar waren. Nach einer Änderung werden diese nun grau in der Liste angezeigt und an die unterster Stelle sortiert.

Status Abfrage

In der ersten Version wurde der Cloud Status nur an der unteren rechten Seite angezeigt. Somit war es für den Benutzer nicht zu erkennen welcher Server ein Problem aufweist. Es wird daher bei jedem Cloud und virtuellen Server der Status an der rechten Seite angezeigt.

Abbildung 26: Cloud Status, rechts bei jedem Cloud Server

Sonstige Anmerkungen

Beim Neustart des virtuellen Servers erfolgte keine Abfrage ob der Benutzer wirklich den Server Neustarten möchte. Hierzu erscheint nun eine Dialogbox die vom Benutzer bestätigt oder abgebrochen werden kann.

5 Realisierung

> “Let us think the unthinkable, let us do the undoable, let us prepare to grapple with the ineffable itself, and see if we may not eff it after all.”
>
> Douglas Adams

In diesem Kapitel werden technische Details zur Realisierung erklärt und einige Quelltext und API Beispiele aufgeführt. Ein weiterer Abschnitt beschreibt die Entwicklungsarbeit an Prototypen.

5.1 libvirt Schnittstelle

Wie in *Kapitel 2.3* beschrieben handelt es sich bei libvirt um eine Schnittstelle zwischen Anwendungen und verschiedenen Virtualisierungstechnologien. Libvirt ist eine C-Bibliothek, auf diese kann aber durch eine weitere Abstraktionsschichten auch mit anderen Programmiersprachen[27] zugegriffen werden. Dadurch werden Programmiersprachen wie Python, Perl, Ruby, Java und PHP unterstützt.

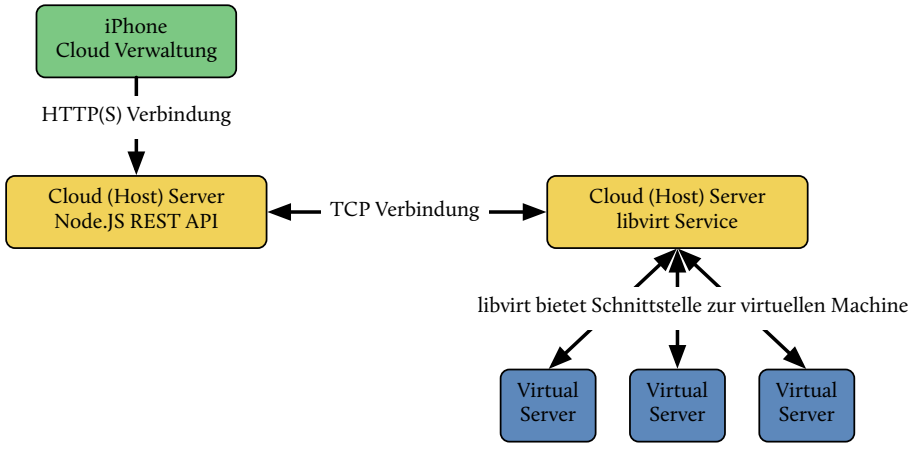


Abbildung 27: Schnittstellenaufbau mit Node.js und libvirt

27. libvirt Bibliothek und weitere Programmiersprachen: http://libvirt.org/bindings.html

Der Softwareentwickler Camilo Aguilar[28] hat eine weitere Bibliothek[29] für libvirt und Node.js entwickelt. Dieses bietet für Node.js viele Schnittstellenimplementationen von libvirt an und kommt für den Prototyp zum Einsatz.

5.1.1 Node.js

Bei Node.js handelt es sich um ein JavaScript Framework zur Erstellung von Server- und Konsolenanwendungen. Der JavaScript Quelltext wird mit der JavaScript-Engine V8[30] von Google in Maschinencode übersetzt und ausgeführt.

Durch die von Camilo Aguilar entwickelte Bibliothek ist es möglich direkt über die Programmiersprache JavaScript mit Node.js auf libvirt zuzugreifen.

```
var hypervisor = new Hypervisor(conf.uri);
var domain;

app.get('/domain/:uuid', function(req, res) {
    domain = hypervisor.lookupDomainByUUID(req.params.uuid);
    domainInfo = domain.getInfo();
    domainInfo.name = domain.getName();

    if (domainInfo != null)
        res.send(domainInfo);
});
```

Listing 1: Quelltext Auszug der REST Node.js Schnittstelle

Mit dem Node.js Web-Framework express[31] kann durch wenigen Codezeilen eine REST (Representational State Transfer) API[32] entwickelt werden. REST vereinfacht die Schnittstelle zwischen Systemen auf eine standardisierte Menge an Aktionen. Für die Umsetzung wird in diesem Fall das HTTP-Protokoll verwendet.

```
http://example.com/domain/b84f1d60-cbc4-4cff-704e-96fc768f0922
```

Listing 2: Beispiel URL

28. GitHub Profil von Camilo Aguilar: http://github.com/c4milo
29. libvirt-node.js Bibliothek: http://github.com/c4milo/node-libvirt
30. JavaScript-Engine V8: http://code.google.com/p/v8/
31. Node.js Web-Framework express: http://expressjs.com/
32. Dissertation von Roy Fielding, zu REST: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Wie im Quelltext Auszug zu erkennen ist wird die HTTP-Funktion GET aufgerufen, dieser wird die eindeutige ID (UUID) der virtuellen Maschine übergeben. Über die libvirt-Funktionen werden zusätzliche Informationen abgefragt und an den Benutzer zurückgeliefert. Die Rückgabe erfolgt in JSON[33], einer JavaScript-Objekt Notation.

5.1.2 JSON

Bei JSON handelt es sich um eine JavaScript-Objekt Notation, die zum kompakten Datenaustausch zwischen Anwendungen dient. Ein großer Vorteil ist die kompakte aber dennoch für den Menschen lesbare Struktur. Jedes JSON-Dokument ist auch ein gültiges JavaScript und kann daher von dieser Programmiersprache einfach interpretiert werden. Es existieren aber auch Parser für andere Programmiersprachen.

```
{
    "state":1,
    "max_memory":524288,
    "memory":524288,
    "vcpus_number":1,
    "cpu_time":3434280000000,
    "name":"example"
}
```

Listing 3: Beispiel Rückgabeinformationen

Alle Daten die über die REST Node.js API angefragt werden, sollen als JSON zurückgeliefert werden. Der Prototyp für die iPhone Anwendung muss somit JSON interpretieren und anzeigen.

5.2 Prototyp

Das Ziel eines Prototyps ist in erster Linie einen Teil der Funktionsfähigkeit der Anwendung dem Benutzer bereitzustellen. Da der Fokus auf der Benutzerfreundlichkeit liegt, wurde ein erster Prototyp ohne Verbindung zur API entworfen. Die angezeigten Daten sind daher fest im Programm verankert. Die erste Erstellung des Prototyps erfolgte nach Fertigstellung des Story-Boards.

In diesem Abschnitt erfolgt eine Beschreibung zur Realisierung der verschiedenen Prototypen.

33. JSON: http://json.org/

5.2.1 Mock-up

Nach der Konzeptionsphase durch das Story-Board wurde ein erstes Mock-up mit der Anwendung Balsamiq Mockups[34] erstellt. Dies soll dem Benutzer einen ersten Eindruck der Cloud Control „Apps“ vermitteln.

Abbildung 28: Erstes Mock-up von Cloud Control

Balsamiq Mockups bietet viele gestalterische Möglichkeiten eine iPhone Anwendung darzustellen. Für Usability-Tests ist eine native iPhone Anwendung für den Benutzer aber besser zugänglich.

5.2.2 XCode und PhoneGap

Die Entwicklung einer iPhone Anwendung erfolgt mit XCode und in der Programmiersprache Objective-C[35]. Bei XCode handelt es sich um die Entwicklungsumgebung von Apple und diese bietet auch Simulatoren für iPhone oder iPad an. Bei Objective-C handelt es sich um eine Erweiterung der Programmiersprache C um Sprachmittel zur objektorientierten Programmierung.

34. Balsamiq Mockups: http://www.balsamiq.com/products/mockups
35. Objective-C: http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html

Durch die komplexe Programmiersprache dauert es sehr lange einen Prototypen zu entwickeln. Der von XCode verwendete Designer, zur Gestaltung der Benutzeroberfläche, ist für einen Prototyp nicht ausreichend, da weiterhin Objective-C auch für kleine Anpassungen benötigt wird.

Als Alternative zur Realisierung wird PhoneGap verwendet, einem OpenSource Framework zur Entwicklung von mobilen Anwendungen. Durch PhoneGap erfolgt die Entwicklung mit den Programmiersprachen bzw. Beschreibungssprachen HTML5, JavaScript und CSS3. Dem Framework liegen CSS- und JavaScript-Dateien für das iPhone Design bei.

Bei PhoneGap handelt es sich um eine Erweiterung für XCode. Die entwickelte Anwendung bleibt somit eine native iPhone „App“.

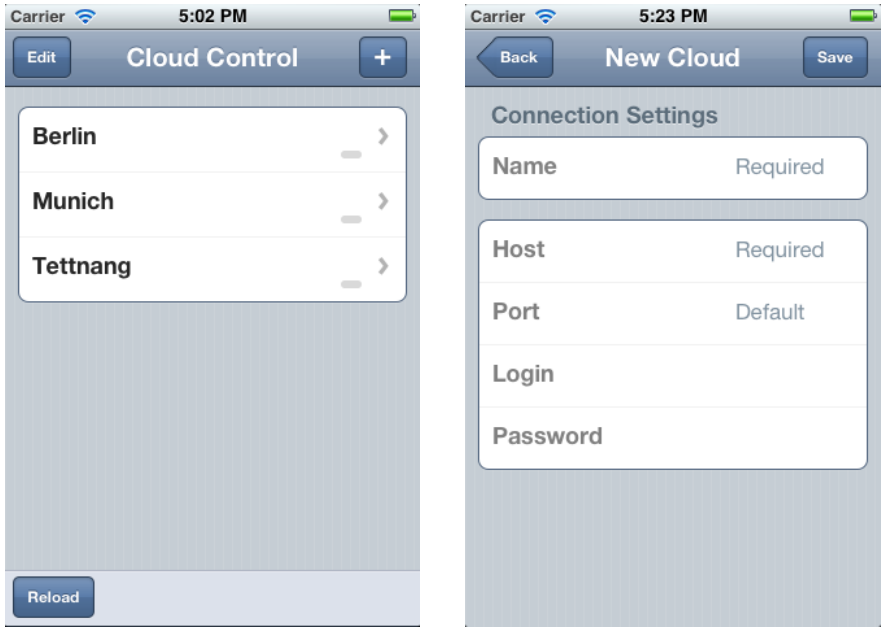


Abbildung 29: Erste Versionen entwickelt mit PhoneGap

Da PhoneGap JavaScript verwendet, ist es möglich direkt auf die Node.js API zuzugreifen. Die API liefert die Daten als JSON zurück, was direkt von JavaScript interpretiert werden kann.

```
<div id="home" class="current">
    <div class="toolbar">
        <a class="button back" id="backButton" href="#back">Back</a>
        <h1>New Cloud</h1>
        <a class="button save" id="saveButton" href="#save">Save</a>
    </div>
    <form class="scroll">
        <h2>Connection Settings</h2>
        <ul class="edit rounded">
            <li>
                Name
                <span class="right">
                    <input type="text" name="name" placeholder="Required" />
                </span>
            </li>
        </ul>
    </form>
</div>
```

Listing 4: Quelltext Ausschnitt für den Bildschirm zum hinzufügen einer neuen Cloud

6 Fazit und Ausblick

> “When you're finished changing, you're finished.”
>
> Benjamin Franklin

Das Ziel der Thesis, einen Prototypen einer mobilen Anwendung zur Verwaltung von Cloud Infrastruktur zu entwickeln, wurde erreicht. Zu diesem Zweck wurde in den ersten Kapitel Grundlagenforschung betrieben, welche in der Analyse und Konzeptionsphase Anwendung fanden. Die technische Realisierungsphase erfolge im gleichen Schritt mit der Konzeption.

Die Analyse der Zielgruppe und Anforderungen konnte im Team, mit den Benutzern, erstellt werden. Dies bildete eine gute Grundlage und stärkte das vertrauen zwischen Benutzer und der neuen Anwendung. Durch das Darstellen der Navigationswege über das Story-Board konnte ein „roter Faden“ in der Anwendung hergestellt werden. Mit den Usability-Tests wurden Fehler schon während der Entwicklungsphase aufgedeckt und verbessert.

Die Programmiersprache Objective-C bringt durch ihre Komplexität längere Entwicklungszyklen mit sich, welches für schnelles entwickeln eines Prototyps und experimentieren von Usability Konzepten ungeeignet ist.Es wurde daher entschieden für die Prototypen auf das alternative Framework PhoneGap zu wechseln, was sich als Vorteil herausstellte.

Als persönliche Anmerkung zum Ende des Fazits, kann gesagt werden, dass sich eine gute und intensive Planungsphase und das frühzeitige einbeziehen der Anwender stets positiv auf die Entwicklung auswirkt.

Das Konzept und der Prototyp kann als Grundlage für die Entwicklung einer marktreifen Anwendung verwendet werden. Durch den immer weiter steigenden Markt für Cloud-Computing, ist genau diese Anwendung für die Verwalter der Cloud hilfreich. Auch die libvirt-Schnittstelle findet sich, durch stetige Weiterentwicklung, in immer mehr Cloud-Anwendungen.

„Cloud Control“, soll genau diese zwei Marktanteile in einer iPhone Anwendung vereinen und eine gute „App“ für Systemadministratoren und Benutzer der Cloud werden.

7 Abbildungsverzeichnis

8 Listingverzeichnis

9 Tabellenverzeichnis

10 Literaturverzeichnis

[bartonb08] Barton George: The Sassy part of the Cloud.
Barton's Blog, 2008.
http://bartongeorge.net/2008/12/18/the-sassy-part-of-the-cloud/

[cloudadop10] IBM Corporation: Defining a framework for cloud adoption.
IBM Corporation, 2010.
ftp://ftp.software.ibm.com/common/ssi/pm/xb/n/cie03069usen/CIE03069USEN.PDF

[cstack12] CloudStack: CloudStack.
Citrix Systems, Inc., 2012.
http://cloudstack.org/

[dahm06] Markus Dahm: Grundlagen der Mensch-Computer-Interaktion.
Pearson Studium, 2006. - ISBN 978-3827371751

[degelas08] Johan De Gelas: Hardware Virtualization: the Nuts and Bolts.
AnandTech, 2008.
http://www.anandtech.com/show/2480

[iso9126] International Organization for Standardization: Norm ISO/IEC 9126.
International Organization for Standardization, 2004-2007.
http://www.iso.org/iso/search.htm?qt=9126&published=on&active_tab=standards

[iso9241] International Organization for Standardization: Teile der Norm ISO 9241 in Vorbereitung.
International Organization for Standardization, 2002-2008.
http://www.iso.org/iso/search.htm?qt=9241&published=on&active_tab=standards

[nie93] Jakob Nielsen: Usability Engineering.
Morgan Kaufmann, 1993. - ISBN 978-0125184069

[niemo90] Jakob Nielsen & Rolf Molich: Improving a human-computer dialogue.
Magazine Communications of the ACM, 1990.

[norman88] Don Norman: The Design of Everyday Things.
Perseus Books, 1988. - ISBN 978-0465067107

[tapworthy11] Josh Clark: Tapworthy - Designing Great iPhone Apps.
O'Reilly Media, 2011. - ISBN 978-1449381653

[wiki11a] Mobile device.
International Organization for Standardization, 2011.
http://en.wikipedia.org/wiki/Mobile_device

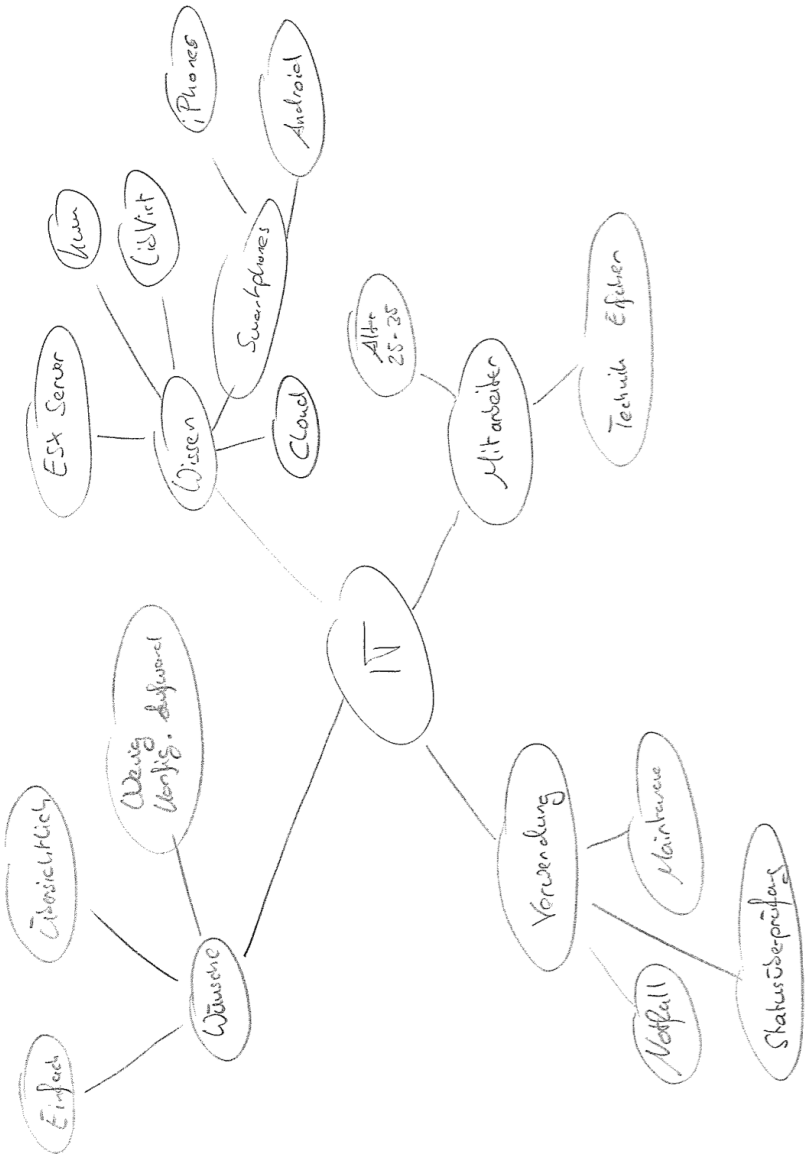
[yaocpu10] Yushu Yao: CPU Performance Xen/Kvm.
Virtualization Studies, 2010.
http://vmstudy.blogspot.de/2010/04/cpu-performance-xenkvm.html

[yaodisk10] Yushu Yao: Disk Performance Xen/Kvm with LVM and Para-virt drivers.
Virtualization Studies, 2010.
http://vmstudy.blogspot.de/2010/04/disk-performance-xenkvm-with-lvm-and.html

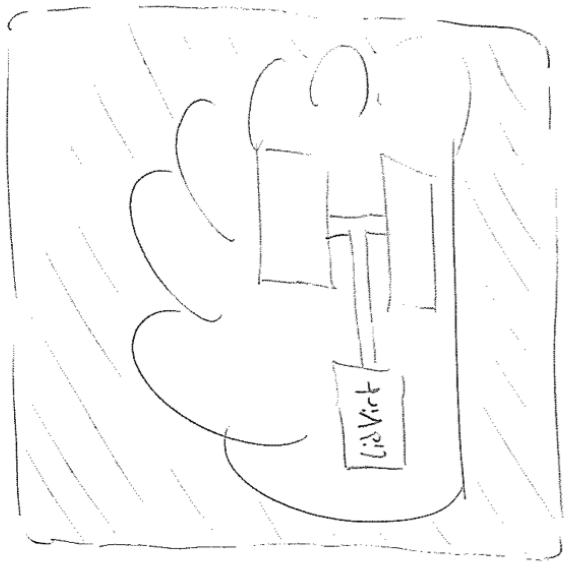
[yaovt10] Yushu Yao: Network Performance Test Xen/Kvm (VT-d and Para-virt drivers).
Virtualization Studies, 2010.
http://vmstudy.blogspot.de/2010/04/network-performance-test-xenkvm-vt-d.html

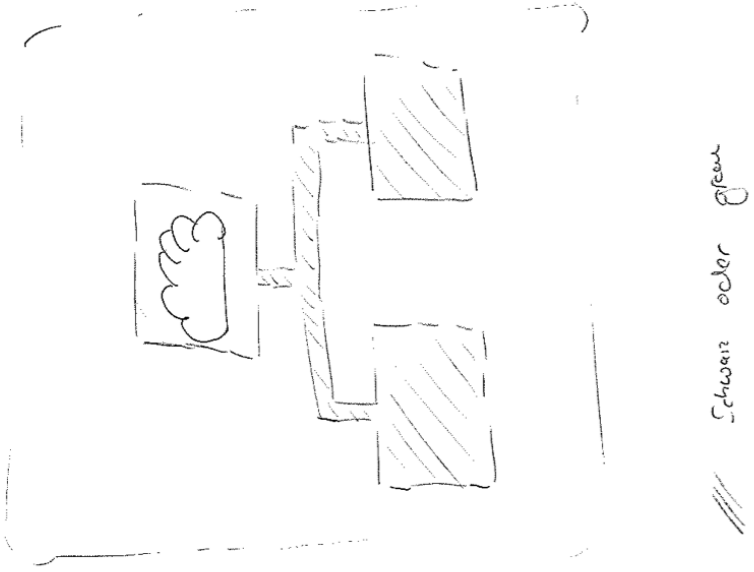
II Anhang

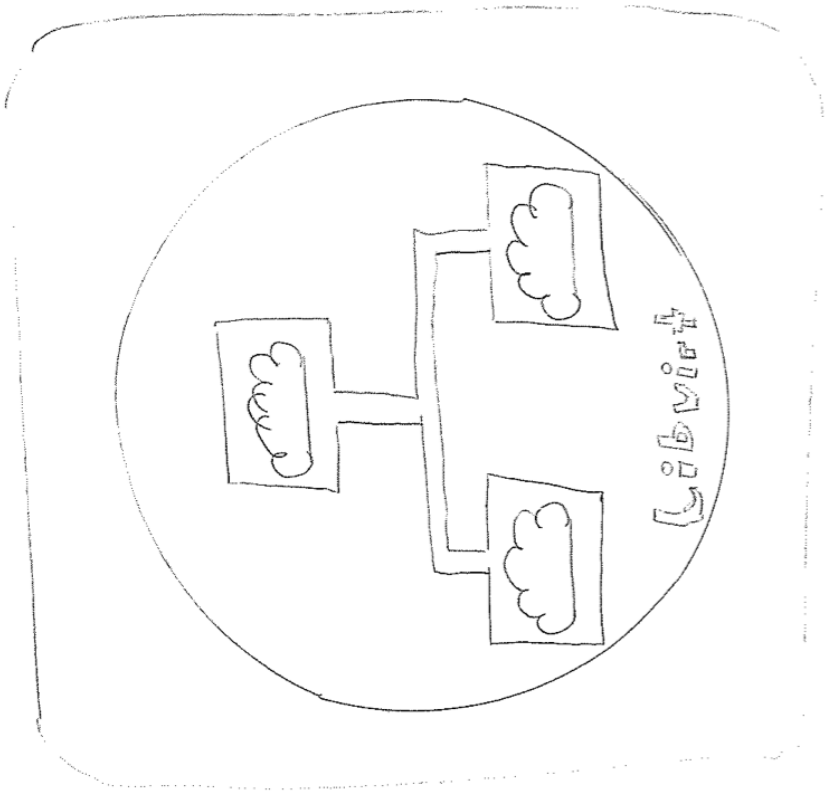
II.1 Zielgruppe Brainstorming

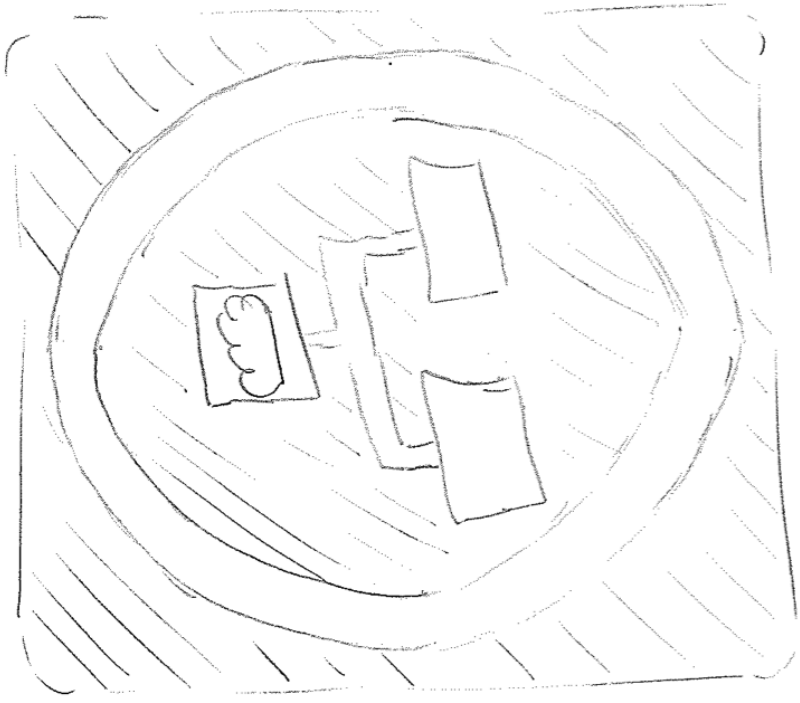


11.2 Logo Konzeption

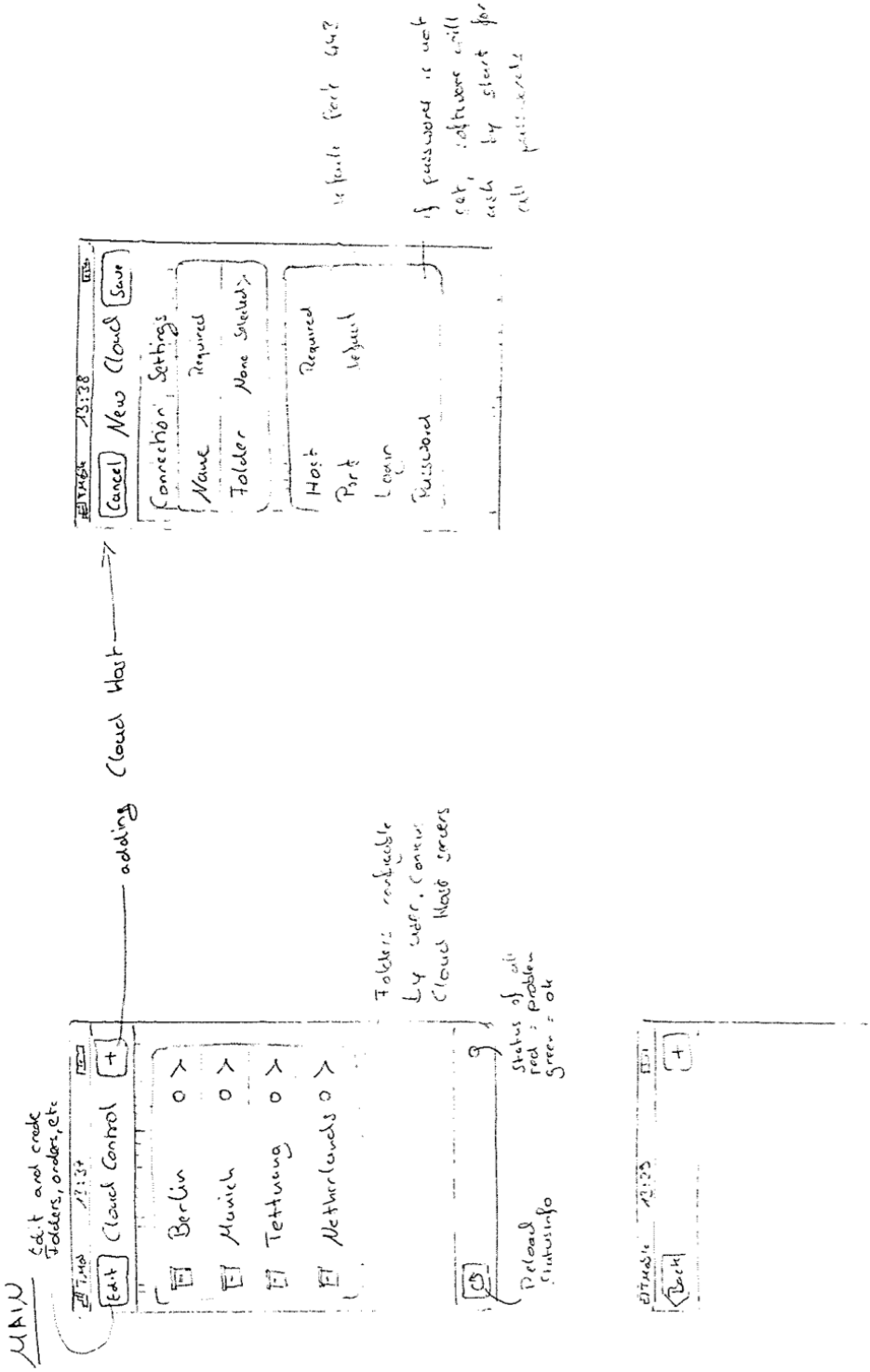


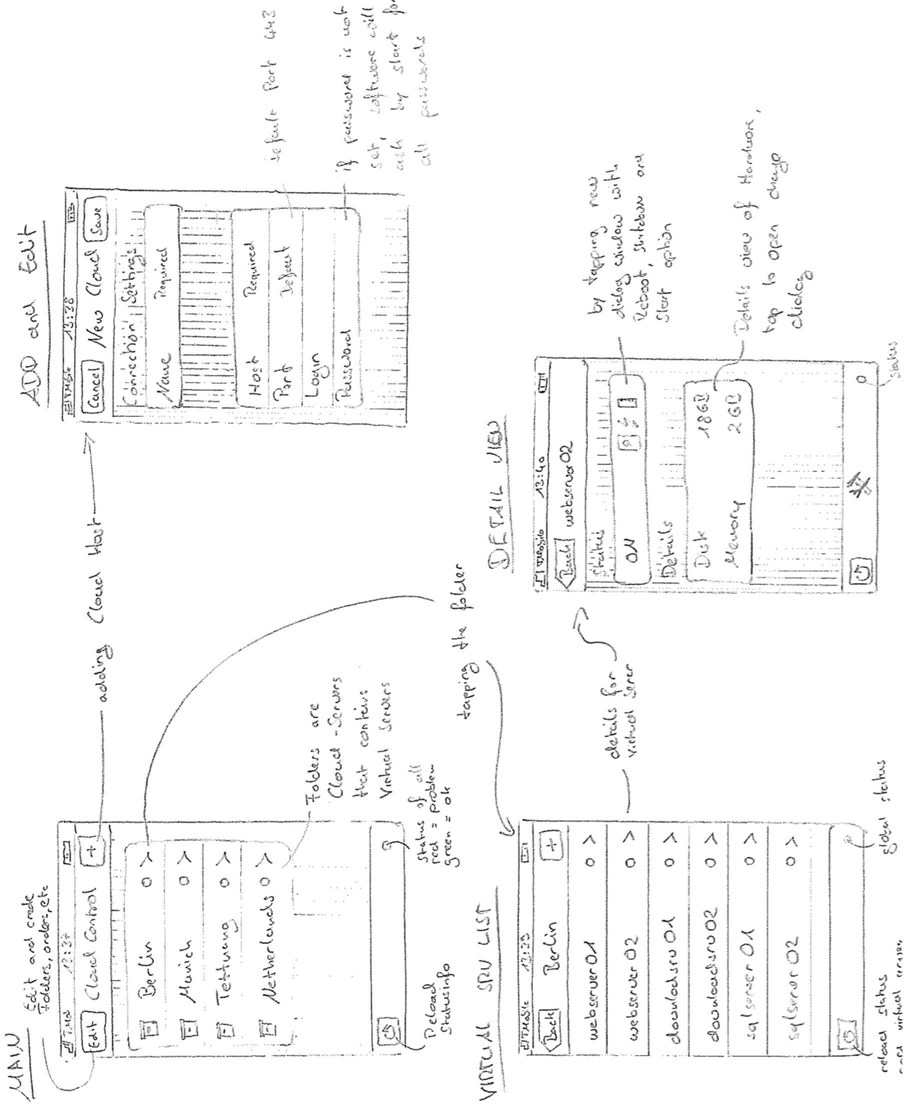






11.3 Story-Board auf Papier




MAIN
Cloud Control
Berlin
Munich
Netherlands
ADD and Edit
Cancel
New Cloud
Save
VIRTUAL SRV LIST
Back
DETAIL VIEW
webserver 02

11.4 Story-Board als Mock-up