

IWI Diskussionsbeiträge

52 (20. August 2012)¹

ISSN 1612-3646



Plinc. Facilitates linking. – Ein Accenture Campus Challenge 2012 Projekt²

Helge Holzmann³, Kim Lana Köhler⁴, Sören C. Meyer⁵, Marvin Osterwold⁶,
Maria-Isabella Eickenjäger⁷ und Michael H. Breitner⁸



¹ Kopien oder eine PDF-Datei sind auf Anfrage erhältlich: Institut für Wirtschaftsinformatik, Leibniz Universität Hannover, Königsworther Platz 1, 30167 Hannover (www.iwi.uni-hannover.de).

² Das Plinc.-Projektteam hat sich in Vorentscheiden an der TU Braunschweig und bei ottobock in Berlin gegen zwei bzw. drei Konkurrenzteams durchgesetzt und am Finale der Länder Deutschland, Österreich und der Schweiz am 5./6.7.2012 in Kronberg bei Frankfurt teilgenommen.

³ Master Informatik Student, Leibniz Universität Hannover.

⁴ Bachelor Wirtschaftswissenschaften Studentin, Leibniz Universität Hannover.

⁵ Bachelor Wirtschaftswissenschaften Student, Leibniz Universität Hannover.

⁶ Bachelor Wirtschaftswissenschaften Student, Leibniz Universität Hannover.

⁷ M.Sc./B.Sc. Wirtschaftswissenschaften, wissenschaftliche Mitarbeiterin und Doktorandin, Betreuerin des Plinc.-Projektteams (eickenjaeger@iwi.uni-hannover.de).

⁸ Professor für Wirtschaftsinformatik und Betriebswirtschaftslehre und Direktor des Instituts für Wirtschaftsinformatik der Leibniz Universität Hannover, Betreuer des Plinc.-Projektteams (breitner@iwi.uni-hannover.de).

Inhalt

Abbildungsverzeichnis.....	4
Tabellenverzeichnis	5
1 Problemorientierte Einleitung.....	6
1.1 Ausgangspunkt und Relevanz der Arbeit.....	7
1.2 Zielsetzung und Vision der Arbeit.....	8
2 Plinc	10
2.1 Das Plinc Profil	10
2.2 Meetingchannel.....	12
2.2.1 Teilnehmerliste	13
2.2.2 Stream	13
2.3 Kontaktchannel.....	15
2.3.1 Verwaltetes Adressbuch	16
2.3.2 Stream	17
2.3.3 Verwaltete Dateiablage	17
3 Prototyp.....	18
3.1 Module.....	18
3.1.1 Server	19
3.1.2 Website	20
3.1.3 App	23
3.1.4 Externe soziale Netzwerke	28
3.2 Plincing.....	31
3.2.1 Explicit Plincing.....	31
3.2.2 Implicit Plincing	32
3.3 Codes	33
3.3.1 Model	33
3.3.2 Controller	42
3.3.3 App-Pages	48
4 Business Case	55
4.1 Kostenpositionen	55
4.2 Einnahmen.....	57
4.2.1 Lizenzmodell.....	57

4.2.2 Werbemodell	59
4.3 Erfolgsrechnung.....	61
4.4 Monetäre Vorteile aus Kundenperspektive	62
4.5 Nicht-Monetäre Vorteile aus Kundenperspektive	63
4.6 Distributionsstrategie	63
4.6.1 Werbe- und Vertriebsstrategie	64
4.6.2 Vertriebsmedium APP	65
4.6.3 Vermarktung Werbeflächen	65
4.7 Risiken.....	65
4.7.1 Verzögerung in der Produktentwicklung	65
4.7.2 Nutzerakzeptanz und Werbepartner	66
4.7.3 Datenschutz und Datensicherheit	66
4.7.4 Einführung neuer Betriebssysteme	66
4.7.5 Backuplösungen	67
4.7.6 Distributionsstrategie	67
4.7.7 Konkurrenz.....	67
5 Fazit	68
6 Anhang.....	69

Abbildungsverzeichnis

Abbildung 1: Unterschiedliche Netzwerke behindern Kontaktaufnahme.....	9
Abbildung 2 Plinc - Facilitates linking.....	9
Abbildung 3: Neue effizientere Ablaufgestaltung im Meeting.....	10
Abbildung 4: Integration bestehender Profile aus Sozialen Netzwerken.....	11
Abbildung 5: Profildarstellung in Abhängigkeit des Nutzers.....	12
Abbildung 6: Teilnehmerliste.....	13
Abbildung 7: Stream.....	13
Abbildung 8: Informationen zu einem Meeting.....	14
Abbildung 9: Meetingübergreifendes Adressbuch.....	15
Abbildung 10: Verwaltetes Adressbuch.....	16
Abbildung 11: Stream.....	17
Abbildung 12: Verwaltete Dateiablage.....	17
Abbildung 13: Schnittstellen von Plinc.....	18
Abbildung 14: Connect-Seite von Plinc. – facebook ist verbunden, die anderen Netzwerke nicht.....	20
Abbildung 15: Das Plinc.-Profil mit verschiedenen Sichtbarkeiten für die unterschiedlichen Felder.....	20
Abbildung 16: Die individuellen Benutzergruppen mit verschiedenen Profil-Sichtbarkeiten des angemeldeten Benutzers.....	21
Abbildung 17: Die Meetings des angemeldeten Benutzers.....	21
Abbildung 18: Die Meeting-Ansicht mit Teilnehmerliste und Stream.....	21
Abbildung 19: Das meeting-/projektübergreifende Adressbuch, gruppiert nach Benutzergruppen.....	22
Abbildung 20: Adressbuch in der hybriden App mit extrahiertem Navigationsmenü über dem Inhalt (links) und Meeting-Ansicht mit Teilnehmerliste (rechts).....	26
Abbildung 21: Startbildschirm der App mit Fotofunktion zum Beitreten eines Meetings (links) und Menü zur Verwaltung der Kontakte und Meetings (rechts).....	26
Abbildung 22: Sequenzdiagramm zum Ablauf eines Daten-Zugriffs mit OAuth2 (Quelle: facebook).....	28
Abbildung 23: Informationen zur Verknüpfung mit Plinc. im Autorisierungs-Dialog von facebook.....	29
Abbildung 24: Freigabe des dauerhaften Zugriffs von Plinc. auf die Daten des facebook-Profiles eines Benutzers.....	30
Abbildung 25: Plinc.-QR-Code (links) mit vereinfachten Plinc.-Code (rechts).....	32
Abbildung 26: Berechnung des WiFi-Fingerprints mit Signalstärken umliegender Access-Points.....	32
Abbildung 27: Entwicklung der Nutzer der werbefinanzierte Version.....	60
Abbildung 28: Ermittlung des Cash-Flows.....	61

Tabellenverzeichnis

Tabelle 1: Entwicklungskosten	55
Tabelle 2: Periodenkosten Monat 1-35.....	56
Tabelle 3: Periodenkosten Monat 1-35.....	56
Tabelle 4: Periodenkosten Monat 36-60.....	56
Tabelle 5: Periodenkosten Monat 36-60.....	57
Tabelle 6:Lizenzmodell	58
Tabelle 7: Linzenzenverkauf	59
Tabelle 8:Erlöse aus Lizenzen.....	59
Tabelle 9: Beispielrechnung: Monetäre Vorteile	62

1 Problemorientierte Einleitung

Als der Harvard Professor Adrew P. McAfee 2006 in seinem Artikel "The Dawn of Emergent Collaboration!" das Wort Enterprise 2.0 erstmals verwendete, hätte er sich wohl nicht träumen lassen, welchen Puls der Zeit er damit getroffen hatte, welcher seitdem die Unternehmenskultur verändert. Er beschrieb, wie die Konzepte des Web 2.0 und von Social Software auf die Zusammenarbeit in Unternehmen übertragen werden können. Dabei soll mithilfe webbasierter Software die Arbeit, insbesondere im Bereich der Informationsverteilung, effizienter gestaltet und folglich Kosten eingespart werden.¹ Die Auswirkungen der schlagartigen Ausbreitung des Web 2.0 sind für Unternehmen tiefgreifend. Dies gilt einerseits für die Entwicklung und Außendarstellung von Produkten, des Marketings, der Kommunikation und des Wissensaustausches unternehmensintern und -extern.² Der Begriff Web 2.0 wird sehr unspezifisch verwendet. Im Allgemeinen wird das Web 2.0 definiert als eine Menge von Technologien und Anwendungen, die eine effiziente Interaktion zwischen Menschen, Inhalten und Daten ermöglichen, um gemeinsam neue Geschäftsfelder, Technologieangebote und soziale Strukturen zu entwickeln. Der Begriff Social Media wird zumeist auf der Ebene der Anwendungen verwendet und betont besonders die Bedeutung der durch die Nutzer erstellten Beiträge und Inhalte (User Generated Content). Die Social Media umfassen z.B. Blogs, Wikis, Microblogs oder Soziale Netzwerke. Soziale Netzwerke spielen in der öffentlichen Diskussion eine zunehmende Rolle. Dabei handelt es sich um Netzgemeinschaften, die mit Hilfe von Profilen und Kontaktlisten der Vernetzung ihrer Mitglieder dienen.³ Unternehmen, die sich für das Web 2.0 interessieren, können sich auf bestehenden Plattformen im Netz engagieren und/oder eigene Investitionen in Web 2.0-Plattformen im Unternehmen zur internen oder externen Verwendung vornehmen. Letzteres wird üblicherweise als Enterprise 2.0 bezeichnet. Im Unternehmenskontext definiert Andrew McAfee Enterprise 2.0 als selbstorganisierende (emergente) Social-Software-Plattformen, die in Unternehmen oder zwischen Unternehmen und ihren Partnern oder Kunden eingesetzt werden. Das entscheidende innovative Element

¹ http://www.accenture.com/Microsites/campuschallenge/Documents/pdf/CampusChallenge/CaChFY12_Accenture_Enterprise_2.0-Plattformen.pdf

² <http://books.google.de/books?id=BNyKs3chlg0C>

³ http://www.dbresearch.de/PROD/DBR_INTERNET_DE-PROD/PROD0000000000260227.PDF

ist, dass durch die spontane, vernetzte Kommunikation der Nutzer über Web 2.0-Anwendungen mit der Zeit neue, individuelle Strukturen entstehen. Web 2.0-Plattformen können im Unternehmen zur Verbesserung von Prozessen, zur Förderung der Zusammenarbeit und allgemein für den Austausch von Wissen etabliert werden. Unternehmensextern sind z.B. die Funktionen Marketing, Reputations- und Issuemanagement, Imagebildung, Recruiting oder die Zusammenarbeit mit Experten oder Zulieferern, z.B. zur Produktentwicklung, relevant.⁴

1.1 Ausgangspunkt und Relevanz der Arbeit

Im Rahmen der Accenture Campus Challenge wird die Problematik der unzureichenden Integration der Social Media Möglichkeiten entlang der Wertschöpfungskette in der heutigen Unternehmenswelt aufgegriffen. Die Accenture Campus Challenge ist ein Wettbewerb für Studentinnen und Studenten technischer und wirtschaftswissenschaftlicher Studiengänge und findet in Kooperation mit vielen Hochschulen und Lehrstühlen in Deutschland, Österreich und der Schweiz statt.

Nun stellt sich die Frage, warum Unternehmen auf Konzepte des Web 2.0 setzen sollten. Braucht man im Unternehmen wirklich Blogs, Wikis und Chats? Unumstritten ist mittlerweile, dass Web 2.0-Komponenten vor allem in der Informationsverteilung und im Wissensmanagement Vorteile mit sich bringen. Relevantes Wissen lässt sich oftmals viel besser im Internet als im unternehmensinternen Intranet beschaffen. Bereits 77 Prozent der deutschen Unternehmen experimentieren daher mit Social Software und dem Wandel hin zu Enterprise 2.0. Eine Studie der Harvard Business School belegt das Potenzial dieser Konzepte und prognostiziert eine Verzehnfachung der Investitionen in Enterprise 2.0-Plattformen bis 2013 durch zahlreiche Konzerne auf insgesamt 4,6 Milliarden Dollar.⁵

Wie kann ein Unternehmen von den Vorteilen des Web 2.0 profitieren und seine Wettbewerbsposition ausbauen? Die Herausforderungen für Unternehmen liegen bei derartigen Fragestellungen längst nicht nur in der effizienten und professionellen Nutzung von Web 2.0 Tools. Auch die Unternehmenskultur erfährt Veränderungen in der Art und Weise der internen Zusammenarbeit, aber auch in Bezug auf die Einbindung des Kunden.

⁴http://www.dbresearch.de/PROD/DBR_INTERNET_DE-PROD/PROD000000000260227.PDF

⁵http://www.accenture.com/Microsites/campuschallenge/Documents/pdf/CampusChallenge/CaChFY12_Accenture_Enterprise_2.0-Plattformen.pdf

1.2 Zielsetzung und Vision der Arbeit

Als Teilnehmer der Campus Challenge waren wir gefordert, uns in die Lage eines Unternehmens zu versetzen, das seine Wertschöpfungsprozesse mit dem Wissen und den Mitteln der internen/externen Online-Community verbinden will. Das Ergebnis dieses Erfindergeistes sollte eine demonstrierbare IT-Lösung sein, die es dem Unternehmen ermöglicht, die Web 2.0-Prinzipien für die Generierung von neuen Ideen oder die Entwicklung neuer Produkte nutzbar zu machen.

Die Aufgabenstellung unterteilte sich in die drei folgenden Bereiche:

Ideenfindung:

Entwickeln Sie Ideen für eine Enterprise 2.0-Plattform, die unternehmensintern und/oder – extern die Möglichkeit bietet, nutzergenerierte Inhalte für die Ideen- und Produktentwicklung eines Unternehmens einzubinden. Beurteilen Sie Ihre Ideen und wählen Sie die tragfähigste Idee unter den Gesichtspunkten Innovationsgrad, technische Realisierbarkeit und potentieller Erfolg (wirtschaftliche und nicht-monetäre Aspekte sowohl für Sie als Mitarbeiter einer Beratungsfirma als auch für Ihre Kunden) aus.

Technologie und Gesamtlösung:

Arbeiten Sie die technologischen Details soweit aus, dass potentielle Interessenten vollumfänglich Kosten, Nutzen und Risiken beurteilen können (Business Case). Beurteilen Sie in diesem Zusammenhang alle technologischen und prozessualen Konsequenzen bei der Einführung Ihrer Enterprise 2.0-Lösung. Achten Sie auch auf mögliche soziale und unternehmenskulturelle Auswirkungen und die software-ergonomischen Aspekte Ihrer Idee.

Einführung und Anwendung Ihrer Lösung:

Ihr Projektteam sollte dem Bestreben folgen, die erarbeitete Lösung interessierten Unternehmen und Institutionen vorzustellen. Eine Lösungspräsentation soll daher dazu beitragen, diesen Interessenten darzustellen, welche Vorteile Ihre Lösung bietet, wie Sie technologische Möglichkeiten ausschöpfen und wie die tatsächliche Einführung Ihrer Lösung aussehen soll. Ihre Aufgabe ist es jetzt, ein Gesamtprojekt aufzusetzen, das alle diese beschriebenen Teilaspekte beinhaltet.



Abbildung 1: Plinc - Facilitates linking

hinsichtlich der gewonnenen Effizienz und Usability in der unternehmerischen Wertschöpfungskette. In der Vision ermöglicht Plinc. Mitarbeitern und Kunden eines Unternehmens unabhängig davon in welchem Sozialen Netzwerk sie aktiv sind in Kontakt zu treten, Daten auszutauschen und sich zu organisieren. Plinc ist dabei ein Kunstname aus

physical link und inc. vom englischen incorporated. Das Verbinden über das Plinc. Netzwerk wird als plincen bezeichnet. Versetzt man sich in die alltägliche Situation eines anfänglichen Treffens eines neu zusammengestellten Projektteams, in dem Kunden und Projektbeteiligte erstmals aufeinander treffen, so lassen sich schnell zeitraubende und ineffizient gestaltete Prozessabläufe definieren. So stellt sich jeder Teilnehmer am Anfang vor, es werden Visitenkarten ausgetauscht, E-Mail Verteiler eingerichtet und Protokolle versendet. Dies kostet Zeit, gerade bei größeren Meetings. Dazu ist der anfängliche Kennlernerfolg schnell verpufft und gleichzeitig ist ein umfassender Austausch von Kontaktinformationen nicht gewährleistet. Dieser unkomfortable Austausch personenbezogener Daten ist nicht mehr zeitgerecht im Zeitalter der sozialen Netzwerke, in denen bereits 74% der Bevölkerung aktiv sind⁶. Tendenz zunehmend. Beispielsweise in Xing, einer Karriereplattform. LinkedIn einer Austauschplattform für Fach- und Führungskräfte. Yammer, einem Unternehmensinternen Netzwerk. Jedoch pflegt nicht jeder der Projektbeteiligten aus dem obigen Beispiel einen Account in all diesen Netzwerken. Und so kann der Kontakt dadurch nicht zustande kommen, weil jeder ein anderes Netzwerk nutzt. Mit Plinc. ändert sich dies nun grundlegend. Plinc. bringt alle bekannten sozialen Netzwerke zusammen und aggregiert die vorliegenden Profile zu einem einheitlichen Profil und macht die Kommunikation Plattformunabhängig.

Ganz nach dem Motto „Der Erfolgreichste im Leben ist derjenige, der gut informiert wird“, wendeten wir uns nun ausgiebig der problematischen Verknüpfung von Enterprise 2.0, Social Media, Profils und der damit verbundenen Herausbildung von Synergieeffekten



Abbildung 2: Unterschiedliche Netzwerke behindern Kontaktaufnahme

⁶ http://www.bitkom.org/files/documents/BITKOM_Studie_Jugend_2.0.pdf

Mit Plinc. bekommt eine typische projektorientierte Sitzung einen völlig neuen Ablauf. Plinc. nutzt dabei die weitverbreiteten QR-Codes. Jeder Teilnehmer fotografiert zu Sitzungsbeginn einen in die Präsentation integrierten Code mit dem jeweiligen Smartphone. Plinc. erstellt nun vollautomatisch eine individuelle Teilnehmerliste auf den Smartphones über die der Nutzer Zugriff auf die Profildaten aller Teilnehmer erhält. Dazu gehören neben dem Namen und der Funktion z.B. die Telefonnummer, Anschrift und E-Mailadresse. Dabei ist das schöne, dass ein Meeting ganz bequem nach Personen, Orten, Zeit und Projekten geordnet werden kann.



Abbildung 1: Neue effizientere Ablaufgestaltung im Meeting

- ➔ Das heißt sie können ganz einfach einsehen mit wem sie letzte Woche im Meeting gesessen haben oder wann ein Meeting überhaupt war
- ➔ Dabei verfolgt Plinc das Konzept das sie Personen nach festen Benutzergruppen definieren können, so dass sie bestimmen welche Informationen sie von sich den anderen Meetingteilnehmern von ihrem Profil zeigen

Im Folgenden Kapitel folgt nun die detaillierte Darstellung der inhaltlichen und funktionalen Möglichkeiten, welche in Plinc. verwirklicht werden konnten.

2 Plinc

2.1 Das Plinc Profil

Plinc soll ein einfaches und schnelles Vernetzen von Nutzern ermöglichen. Unter dem Aspekt, dass ein Großteil der Menschen, die regelmäßig im Internet aktiv sind, bereits in mindestens einem sozialen Netzwerk wie Facebook oder Twitter ein Profil besitzen, schafft es Plinc diese Daten zu nutzen. Das bedeutet, die Daten der Nutzer liegen oftmals schon bereit und die Bereitschaft ein neues Profil zu erstellen sinkt. Die Daten besitzen häufig einen rein privaten Fokus, der mithilfe von Plinc nun in das Business übertragen werden kann. Der Vorteil ist, dass durch Plinc kein neues Profil erstellt werden muss. Die Daten

werden an einem Ort gesammelt und immer aktuell gehalten, da die Nutzer kein weiteres Profil pflegen müssen.

	twitter	Yammer The Enterprise Social Network	XING	facebook	Plinc
Bild					
Name	Sören M.	Sören Meyer	Sören	Sören Christian Meyer	Sören Christian Meyer
Geburtstag	19.05.1988				19.05.1988
Telefon Geschäftlich		0511-2643472		0511-2643472	0511-2643472
Mobil			0176-7538335		0176-7538335

Abbildung 2: Integration bestehender Profile aus Sozialen Netzwerken

Ein Profil bei Plinc ist folgendermaßen aufgebaut. Neben einem Profilbild werden Name und weitere Informationen zum Nutzer angezeigt. Die obige Abbildung zeigt, wie eine Integration bestehender Profile aus sozialen Netzwerken aussehen kann. In diesem Beispiel besitzt ein Nutzer vier Profile aus Facebook, Xing, Twitter und Yammer. Der Nutzer entscheidet selbst, welche Daten übernommen werden. Somit entsteht ein vollständiges Businessprofil, das zunächst sämtliche Daten bereitstellt.

Im zweiten Schritt werden die Berechtigungen von dem Nutzer vergeben. Je nachdem welche Person das Profil einsieht, wird entschieden welche Daten freigegeben werden dürfen. Die Abbildung nimmt das gleiche Beispiel wie im Vorhinein auf und zeigt exemplarisch, wie ein Profil aus Nutzersicht und in Abhängigkeit von der Rolle (Kunde, Partner, Vorgesetzter, Kollege und Freund) aussehen könnte.

		Kunden	Partner	Vorgesetzte	Kollegen	Freunde
Bild						
Name	Sören Christian Meyer	Meyer	Meyer	Sören Christian Meyer	Sören Meyer	Sören
Geburtsdatum	19.05.1988			19.05.	19.05.1988	19.05.1988
Telefon						
Geschäftlich	0511-2643472	0511-2643472	0511-2643472	0511-2643472	0511-2643472	0511-2643472
Mobil	0176-7538335					0176-7538335

Abbildung 3: Profildarstellung in Abhängigkeit des Nutzers

Die Idee dabei ist, dass im beruflichen Umfeld nicht jeder 1:1 alle Daten einsehen soll. So kann z.B. ein Freund in diesem Fall den kompletten Datensatz einsehen, wohingegen der Vorgesetzte nur einen Teil der Daten sieht. Weiterhin kann ein Freund ein anderes Profilbild sehen, das weniger in das berufliche Umfeld passt.

2.2 Meetingchannel

Grundsätzlich gibt es zwei Möglichkeiten das Plinc-Profil auszutauschen. Einmal über die Nutzung des Meetingchannels und über den Kontaktchannel. Der Meetingchannel bietet einen einfachen Austausch von Kontaktdaten aber auch den einfachen Austausch von Gedanken und Ideen während oder nach einem Meeting. Der Kontaktchannel hat einen anderen Fokus. Hier wird die Möglichkeit geschaffen, einen Kunden- und Unternehmenskontakt herzustellen. Beispielsweise, sollten potentielle Bewerber Informationen über ein Unternehmen suchen und mit dem Unternehmen in Kontakt treten wollen, so findet der Kontaktchannel mit unternehmensexterner Ausrichtung Anwendung. Auf den Kontaktchannel wird im Detail im nächsten Abschnitt eingegangen. Der Meetingchannel bietet grundsätzlich eine Teilnehmerliste und einen Stream.

2.2.1 Teilnehmerliste



Abbildung 4: Teilnehmerliste

Nachdem alle Meetingteilnehmer den QR-Code abfotografiert haben, werden sie automatisch zu der Teilnehmerliste hinzugefügt. Hier erscheinen sie mit dem Namen und einem Profilbild. Darüber wird das jeweilige Profil des Teilnehmers erreicht, das in Abhängigkeit von dem Nutzer mit unterschiedlich vielen Daten gefüllt ist.

2.2.2 Stream



Abbildung 5: Stream

Der Stream ist ein Ort für den Nachrichtenaustausch zwischen den Teilnehmern während oder nach einem Meeting. Dieser ist so aufgebaut, dass die Nachrichten chronologisch geordnet eingehen. Hier werden Gedanken, Ideen etc. an einem gemeinsamen Ort gesammelt. Im Nachhinein finden sich schnell und übersichtlich alle Kommentare wieder. Durch die Kommunikation in den Streams werden E-Mails gespart, die als unproduktiv gesehen werden. So können schon während eines Meetings Online-Protokolle erstellt werden und wichtige Zeit eingespart werden. Jeder Nutzer kann Dateien hinzufügen, die für ein Meeting von Relevanz sein können, sodass thematisch zusammengehörige Dateien an einem Ort schnell zu finden sind.

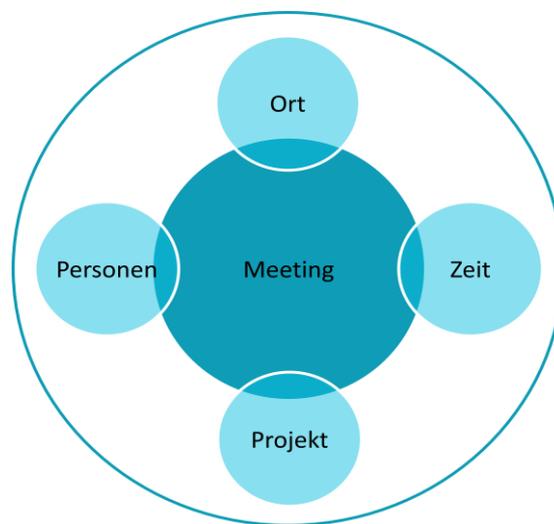


Abbildung 6: Informationen zu einem Meeting

Neben dem Stream und der Teilnehmerliste werden automatisch wichtige Informationen generiert. Zu jedem Meeting das erstellt wird, werden Personen hinzugefügt und es werden wichtige Informationen bereitgestellt. Diese Informationen gliedern sich in Orts- und Zeitangaben und es besteht die Möglichkeit, mehrere Meetings einem Projekt hinzuzufügen. Dies sind interessante Informationen, da beispielsweise Erinnerungen darüber bestehen wann ein Meeting war, aber die dazugehörigen Personen nicht einfach gefunden werden können.

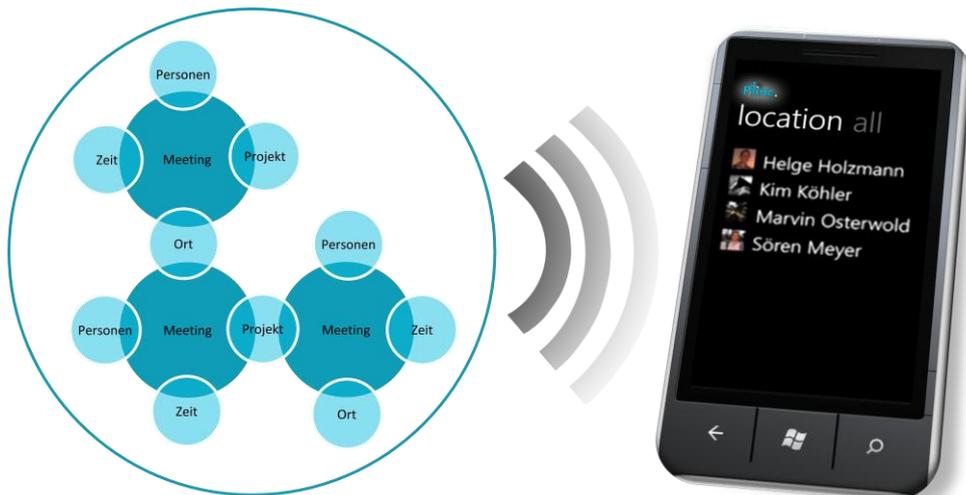


Abbildung 7: Meetingübergreifendes Adressbuch

Um nicht in den einzelnen Teilnehmerlisten nach den Kontaktdaten suchen zu müssen, entsteht ein meetingübergreifendes Adressbuch. Ein Nutzer von Plinc wird nach einiger Zeit an mehreren Meetings teilgenommen haben. Daher bietet das Adressbuch die optimale Möglichkeit nach den Informationen Ort, Zeit und Projekt zu filtern.

2.3 Kontaktchannel

Der Kontaktchannel hat die Möglichkeit unternehmensexterne Personen gezielt einzubeziehen. Er ist dafür da, den Kunden in bestimmte Prozesse zu integrieren und frühzeitig einen Mehrwert zu generieren. Der Kontaktchannel bietet drei grundlegende Funktionalitäten: das verwaltete Adressbuch, den Stream und die verwaltete Dateiablage. Dabei verfügt dieser über ein individuelles Branding des Unternehmens, um einen Wiedererkennungswert zu wahren. Das Branding besteht aus einem Logo und einer Unternehmensfarbe, eben dem Corporate Design des Unternehmens.

2.3.1 Verwaltetes Adressbuch



Abbildung 8: Verwaltetes Adressbuch

Das verwaltete Adressbuch beinhaltet die vom Unternehmen bereitgestellten Kontakte. Im Vergleich zu der Teilnehmerliste aus dem Meetingstream sind nicht alle Kontakte im Adressbuch, die sich über Plinc mit dem Unternehmen verbunden haben, sondern wichtige Kontakte des Unternehmens die für die Kunden interessant sein können. Unter Umständen können sehr viele Unternehmenskunden etc. auf Plinc zugreifen. Dann besitzt das Adressbuch wenig Sinn. Anders ist es, wenn gezielte Ansprechpartner seitens des Unternehmens im Adressbuch auffindbar sind. Zum Beispiel können potentielle Bewerber gleich Kontaktdaten für die Personalabteilung finden etc. Die obige Abbildung zeigt das verwaltete Adressbuch anhand des Beispiels von Accenture. Das Branding mit dem Logo von Accenture und der Farbe Rot lässt den Kunden intuitiv erkennen in welchem Kontaktchannel er sich befindet. Nun hat er die Möglichkeit unterschiedliche Kontaktdaten, zum Beispiel von Accenture Mitarbeitern, einzusehen.

2.3.2 Stream



Abbildung 9: Stream

Im Stream kann das Unternehmen wichtige Neuigkeiten posten, die dann von dem Kunden kommentiert werden können. Die Kunden bekommen also alle Neuigkeiten des Unternehmens schnell mit.

2.3.3 Verwaltete Dateiablage



Abbildung 10: Verwaltete Dateiablage

Die verwaltete Dateiablage beinhaltet vom Unternehmen bereitgestellte Dateien, die der Kunde einsehen kann. Im Vergleich zum Meetingchannel können Dateien nicht an bestimmte Kommentare im Stream angehängt werden, sondern werden zentral in der

Ablage verwaltet. Jederzeit können die Dateien vom Unternehmen gelöscht werden und neue hinzugefügt werden.

3 Prototyp

Neben dem Geschäftsmodell stellt der technische Prototyp den zweiten Hauptbestandteil des Projektes dar. Der Prototyp ist dabei als Proof-of-Concept zu verstehen. Er zeigt die technische Umsetzbarkeit der Konzepte von Plinc.

3.1 Module



Abbildung 11: Schnittstellen von Plinc.

Das Plinc.-System bietet mehrere Schnittstellen zum Zugriff und zur Erweiterung der bereitgestellten Dienste. Den Kern bildet ein Webserver. Dieser ist mit verschiedenen sozialen Netzwerken verbunden und stellt eine Benutzerschnittstelle zur Bedienung von Plinc. bereit. Dabei handelt es sich um eine Weboberfläche, welche sowohl direkt in einem Webbrowser als auch in der hybriden Smartphone-App verwendet werden kann. Eine weitere Schnittstelle ermöglicht die Anbindung zusätzlicher sozialer Netzwerke. So können beispielsweise auch die Daten firmeninterner Netzwerke in das Plinc.-System eingebunden werden.

3.1.1 Server

Der Webserver wurde mit dem Microsoft ASP.NET-Framework⁷ in der Programmiersprache C# implementiert. Zur logischen Strukturierung wurde dabei das MVC-Plugin⁸ eingesetzt. Des Weiteren kam die Razor-View-Engine zur Entwicklung der Benutzeroberflächen zum Einsatz.

Mithilfe des MVC-Plugins wurde der Servercode nach dem Model-View-Controller-Pattern strukturiert. Die Modelklassen beinhalten die Geschäftslogik sowie die Datenklassen, welche auch zur Persistenz der Serverzustände verwendet werden. Die Klassen in den View- und Controller-Namensräumen dienen der Steuerung des Servers. Razor ermöglicht dabei eine einfache Beschreibung der dynamischen HTML-Erzeugung zur Darstellung der Benutzeroberfläche, wie in Listing 1 dargestellt. Die Klassen der Model- und Controller-Namensräume können im Kapitel 3.3 Codes eingesehen werden.

```
<!DOCTYPE html>
<html>
  <head>
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
    <script src="@Url.Content("~/Scripts/jquery.js")" type="text/javascript"></script>
  </head>
  <body>
    <div>
      @{
        string userName = string.Empty;
        int id = int.Parse(ViewContext.RouteData.Values["id"].ToString());
        userName = GetUser(id).Profile.UserName;
        userName = userName == null ? string.Empty : " " + userName;
        <div>
          Willkommen@{@userName}!
        </div>
      }
    </div>
    <div style="margin: 10px;">
      @RenderBody()
    </div>
  </body>
</html>
```

Listing 1: Dynamische HTML-Beschreibung mit Razor (vereinfachter Auszug aus dem Plinc.-Prototyp)

Die Weboberfläche des Servers kann auf zwei Arten verwendet werden:

- als vollständige Website im Webbrowser
- als Bestandteil der hybriden Smartphone-App

In den folgenden Unterkapiteln werden diese beiden Zugriffsmöglichkeiten sowie die Schnittstelle zur Anbindung externer Netzwerke näher beschrieben.

⁷ <http://www.asp.net>

⁸ <http://www.asp.net/mvc>

3.1.2 Website

Über die Website von Plinc. können alle Funktionen, die keine Hardwareunterstützung benötigen, verwendet werden. Für Funktionen mit Hardwarezugriff auf Smartphones wird das hybride App-Konzept verwendet (siehe Unterkapitel 3.1.3 App). Zu den Funktionen, die eine Hardwareunterstützung benötigen, zählen alle standortbasierten Dienste sowie die Möglichkeit des Vernetzens per QR-Code.

Somit ermöglicht die Webseite im Prototyp die folgenden Anwendungsfälle:

- Verknüpfung mit sozialen Netzwerken



Abbildung 12: Connect-Seite von Plinc. – facebook ist verbunden, die anderen Netzwerke nicht.

- Einstellung der Profil-Sichtbarkeit

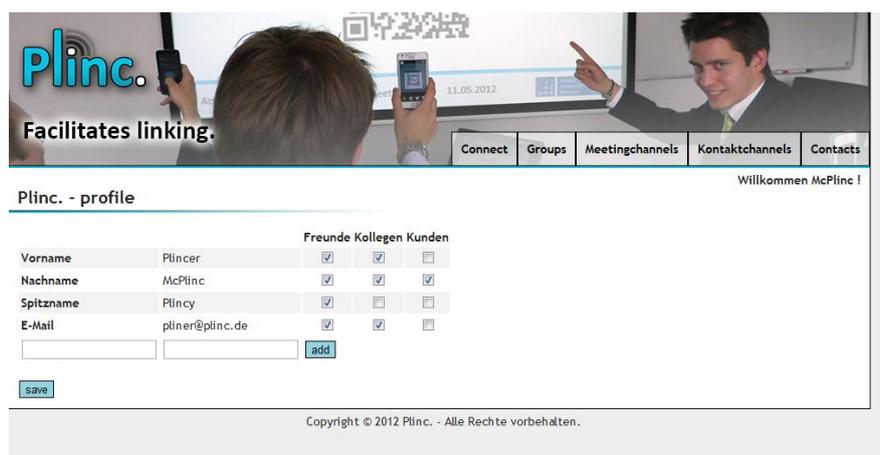


Abbildung 13: Das Plinc.-Profil mit verschiedenen Sichtbarkeiten für die unterschiedlichen Felder

- Verwaltung der Benutzergruppen



Abbildung 14: Die individuellen Benutzergruppen mit verschiedenen Profil-Sichtbarkeiten des angemeldeten Benutzers

- Verwaltung von Meeting-/Kontaktchannels



Abbildung 15: Die Meetings des angemeldeten Benutzers

- Anlegen neuer Meetings (mit QR-Code Generierung) / Ansicht der Teilnehmer u. des Streams



Abbildung 16: Die Meeting-Ansicht mit Teilnehmerliste und Stream

- Verwaltung und Ansicht des Adressbuches



Abbildung 17: Das meeting-/projektübergreifende Adressbuch, gruppiert nach Benutzergruppen

Wie in Abbildung 16 zu sehen, wird auf dem Server auch der QR-Code erzeugt, über den die Benutzer die Möglichkeit haben, sich mit einem Meeting zu verlinken. Zur Erzeugung des QR-Codes und zur Darstellung auf der Website wurde die `QrCode.Net`-Bibliothek⁹ verwendet. Mit ihrer Hilfe wird ein Code zur Identifizierung eines Meetings in dem QR-Code codiert. Daraufhin wird das Plinc.-Logo in der Mitte des Codes platziert und der QR-Code als Antwort der Webseite an den Browser des Benutzers übertragen. Ein Auszug des Meeting-Controllers, der diese Aufgabe übernimmt, ist Listing 2 zu entnehmen. Der vollständige Code dieses und der anderen Controller der Website kann dem Kapitel 3.3 Codes entnommen werden.

⁹ <http://qrcodenet.codeplex.com>

```

public byte[] GenerateQrCode(int meetingId)
{
    IPEndPoint ip = Dns.GetHostEntry(Dns.GetHostName());
    string ipAddress = ip.AddressList.Select(i => i.ToString())
        .FirstOrDefault(i => i.IndexOf('.', i.IndexOf('.', i.IndexOf('.') + 1) + 1) > -1);

    QrEncoder qrEncoder = new QrEncoder(ErrorCorrectionLevel.H);
    QrCode qrCode = qrEncoder.Encode("http://" + ipAddress + "/Meetings/Join/" +
        meetingId + " - meeting" + meetingId + "-" + ipAddress);

    Renderer renderer = new Renderer(5, Brushes.Black, Brushes.White);
    using (MemoryStream stream = new MemoryStream())
    {
        renderer.WriteToStream(qrCode.Matrix, stream, ImageFormat.Png);

        using (Image image = Image.FromStream(stream))
        using (Graphics graphics = Graphics.FromImage(image))
        using (Image logo = Image.FromFile(Server.MapPath("~/Content/images/Logo_small.png")))
        {
            graphics.DrawImage(logo, (image.Width / 2) -
                (float)Math.Ceiling(QrLogowidthHeight / 2d), (image.Height / 2) -
                (float)Math.Ceiling(QrLogowidthHeight / 2d), QrLogowidthHeight, QrLogowidthHeight);
            stream.Position = 0;
            image.Save(stream, ImageFormat.Png);
        }

        return stream.ToArray();
    }
}

public ActionResult Qr(int selection)
{
    return new FileContentResult(GenerateQrCode(selection), "image/png");
}

```

Listing 2: QR-Code Generierung

3.1.3 App

Die App basiert auf einem hybriden Aufbau. Dabei wird die oben beschriebene Website mithilfe spezieller Stylesheets für die mobile Darstellung angepasst und innerhalb einer Rahmenanwendung angezeigt. Dies ermöglicht den Zugriff auf die Hardware des Smartphones, wie GPS und Kamera. So werden auch standortbasierte Funktionen sowie das Vernetzen per QR-Code ermöglicht. Für den Prototypen wurde die App als Microsoft Windows Phone¹⁰ 7.5 App mit dem Microsoft Silverlight-Framework in C# entwickelt. Durch Verwendung der Website zum Darstellen der Benutzeroberfläche ist jedoch eine einfache Portierung auf andere mobile Plattformen möglich. Dazu muss lediglich der Zugriff auf die Hardware der Plattform implementiert werden. Zur Darstellung der Oberfläche kann der Webbrowser des jeweiligen Systems verwendet werden. Dabei können einzelne Elemente, wie die Navigationsmenüs der Unterseiten, problemlos direkt in die Anwendung integriert werden, um sie an die übliche Darstellung der Plattform anpassen zu können. Für die Inhalte der einzelnen Unterseiten sind dabei jeweils einzelne Webbrowser-Elemente zu verwenden.

Listing 4: Integration einer Website mit extrahiertem Navigationsmenü in hybrider App (aus Plinc.-

¹⁰<http://www.microsoft.com/windowsphone>

Prototyp, vereinfacht) zeigt vereinfacht, wie eine solche Integration in Windows Phone umgesetzt wurde. Für die anderen Systeme kann dies analog implementiert werden. Eine komplette Auflistung aller App-Pages ist im Kapitel 3.3 Codes zu finden.

Die Ausgabe des in Listing 4 dargestellten Codes ist in Abbildung 18(links) zu sehen. Das Beispiel zeigt das meeting-/projektübergreifende Adressbuch in der App. Dabei handelt es sich bei der Namensliste um Inhalte vom Webserver, die in die App eingebunden wurden. Das Navigationsmenü wurde dabei aus der Website extrahiert und oberhalb des Inhalts im Windows Phone üblichen Design dargestellt.

Außerdem ist in Abbildung 17 (rechts) die Teilnehmerliste einer Meetingseite zu sehen. Auf diese Seite gelangt ein Benutzer nachdem er sich mit einem Meeting verbunden hat oder eines seiner Meetings über das Verwaltungsmenü aufruft. Auch auf dieser Seite wurde das Navigationsmenü, welches aus Teilnehmerliste und dem Meetingstream besteht, aus der Website extrahiert und über dem Inhalt im typischen Metro-Design von Windows Phone eingebunden. Zwischen beiden Menüpunkten kann nun durch einen seitlichen Fingerwisch gewechselt werden. Zur Darstellung der Inhalte wurden dabei zwei getrennte Webbrowserelemente verwendet. So bekommt der Benutzer das Gefühl einer normalen App, mit den üblichen Steuerelementen, obwohl es sich bei den Inhalten um Websites handelt.

Die Hardware-Zugriffsfunktionen wurden noch nicht komplett umgesetzt. So sind die Standortbasierten-Dienste kein Bestandteil des Prototyps. Die Erkennung des QR-Codes wurde hingegen bereits komplett implementiert. Passend zu dem obigen Code in Listing 2, der die QR-Codes für die Website generiert, zeigt Listing 3, wie der QR-Code in der Plinc.-App gelesen und verarbeitet wird. Dazu wird die MessagingToolkit-Library¹¹ verwendet, welche das eine Foto nach QR-Codes analysiert und deren Inhalt ausgibt.

¹¹<http://platform.twit88.com/projects/mt-barcode>

```

private void HandleCameraCaptureImageAvailable(object sender, ContentReadyEventArgs e)
{
    Stream imageStream = (Stream)e.ImageStream;
    BarcodeDecoder barcodeDecoder = new BarcodeDecoder();

    Dictionary<DecodeOptions, object> decodingOptions = Dictionary<DecodeOptions, object>();
    List<BarcodeFormat> possibleFormats = newList<BarcodeFormat>(1);
    Result result;

    Dispatcher.BeginInvoke(() =>
    {
        WriteableBitmap qrImage = new WriteableBitmap(
            (int)camera.Resolution.Width, (int)camera.Resolution.Height);
        imageStream.Position = 0;
        qrImage.LoadJpeg(imageStream);

        possibleFormats.Add(BarcodeFormat.QRCode);
        decodingOptions.Add(DecodeOptions.PossibleFormats, possibleFormats);

        try
        {
            result = barcodeDecoder.Decode(qrImage, decodingOptions);

            string[] resultText = result.Text.Split(newchar[] { '-' });
            if (resultText.Length > 0 && resultText[1].Trim().StartsWith("meeting"))
            {
                string[] meetingText = resultText[1].Trim().Split(newstring[] { "meeting" },
                    StringSplitOptions.RemoveEmptyEntries);

                int meetingId;
                if (meetingText.Length == 1 && int.TryParse(meetingText[0], out meetingId))
                {
                    (App.Current asApp).MeetingId = meetingId;
                    NavigationService.Navigate(newUri("/AttendMeetingPage.xaml",
                        UriKind.Relative));
                }
            }

            if (resultText.Length > 1)
            {
                (App.Current asApp).ServerHostName = resultText[2].Trim();
            }
        }
        catch
        {
            return;
        }
        finally
        {
            capturing = false;
        }
    });
}

```

Listing 3: Code zum Einlesen eines QR-Codes von einem Foto mit dem MessagingToolkit



Abbildung 18: Adressbuch in der hybriden App mit extrahiertem Navigationsmenü über dem Inhalt (links) und Meeting-Ansicht mit Teilnehmerliste (rechts)

Bei der Gestaltung der App wurde sehr viel Wert auf Usability gelegt. So soll es möglichst schnell möglich sein, auf die Hauptfunktionen von Plinc. zugreifen zu können. Da die wichtigste Funktion das Plincing (siehe Kapitel 3.2 Plincing) ist, ist dies auch die erste Funktion, die man beim Starten der App erreicht. Der Startbildschirm der App ist in Abbildung 19 (links) zu sehen.



Abbildung 19: Startbildschirm der App mit Fotofunktion zum Beitreten eines Meetings (links) und Menü zur Verwaltung der Kontakte und Meetings (rechts)

```

using System;
using System.Text.RegularExpressions;
using System.Windows;
using Microsoft.Phone.Controls;

namespace Plinc
{
    public partial class ContactsPage : PhoneApplicationPage
    {
        private bool loaded = false;

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);

            loaded = false;
            AllContactsBrowser.Navigate(new Uri(@"http://" + ServerHostName + "/Contacts"));
        }

        private void HandleBrowserNavigated(object sender, NavigationEventArgs e)
        {
            (sender as WebBrowser).Navigating += HandleBrowserNavigating;
        }

        private void HandleAllContactsBrowserNavigated(object sender, NavigationEventArgs e)
        {
            if (loaded)
            {
                return;
            }

            string content = AllContactsBrowser.SaveToString();
            string menu = content.Substring(content.IndexOf("/div"));
            string[] links = menu.Split('|');
            foreach (string link in links)
            {
                Regex regex = new Regex("<a.*?href=\"(.*)\".*?>(.*?)</a>");
                Match match = regex.Match(link);
                if (match.Success)
                {
                    string uri = match.Groups[1].Value;
                    string title = match.Groups[2].Value;

                    if (title != "All")
                    {
                        WebBrowser browser = new WebBrowser();
                        browser.VerticalAlignment = VerticalAlignment.Stretch;
                        browser.HorizontalAlignment = HorizontalAlignment.Stretch;
                        browser.Navigated += HandleBrowserNavigated;
                        ContactsPivot.Items.Add(new PivotItem() { Header = title, Content = browser });
                        browser.Navigate(new Uri("http://" + ServerHostName + uri));
                    }
                }
            }

            loaded = true;
        }

        private void HandleBrowserNavigating(object sender, NavigatingEventArgs e)
        {
            if (loaded)
            {
                e.Cancel = true;
                string uri = e.Uri.ToString();
                (App.Current as App).ContactUri = uri;
                NavigationService.Navigate(new Uri("/ContactPage.xaml", UriKind.Relative));
            }
        }
    }
}

```

Listing 4: Integration einer Website mit extrahiertem Navigationsmenü in hybrider App (aus Plinc.-Prototyp, vereinfacht)

3.1.4 Externe soziale Netzwerke

Neben den Benutzerschnittstellen bietet Plinc. auch eine Schnittstelle zur Anbindung externer sozialer Netzwerke. Dazu wird das OAuth2-Protokoll¹² verwendet, welches von den großen sozialen Netzwerken, wie facebook¹³ oder twitter¹⁴ bereits unterstützt wird. Das Sequenzdiagramm in Abbildung 20 zeigt den Ablauf zum Abrufen der Daten von der öffentlichen API eines externen Netzwerks mit OAuth2 am Beispiel von facebook.

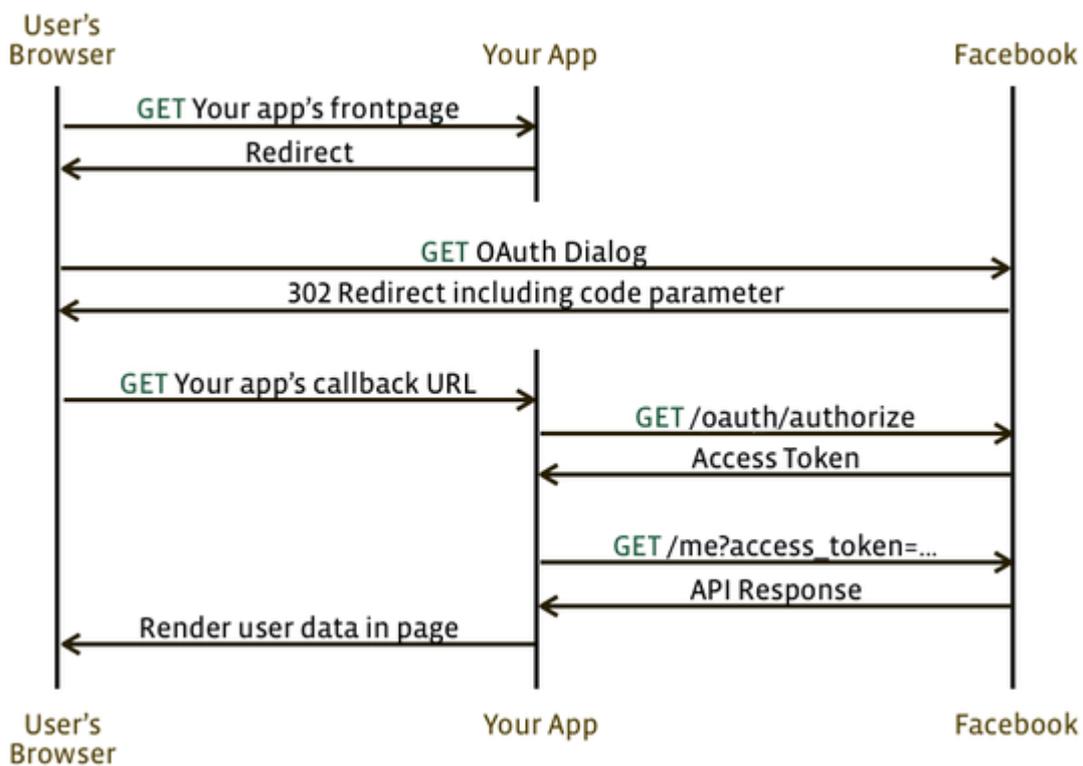


Abbildung 20: Sequenzdiagramm zum Ablauf eines Daten-Zugriffs mit OAuth2 (Quelle: facebook¹⁵)

Wählt ein Benutzer auf der Plinc.-Website nun aus, dass sein Profil beispielsweise mit facebook verbunden werden soll, gelangt er zu dem entsprechenden Autorisierungs-Dialogs des Netzwerks (Abbildungen Abbildung 21 und Abbildung 22). Listing 5 zeigt, wie ein solcher Ablauf auf dem Server verarbeitet wird.

¹² <http://oauth.net/2>

¹³ <http://www.facebook.de>

¹⁴ <http://www.twitter.de>

¹⁵ <http://developers.facebook.com/docs/authentication/server-side>

Durch diesen Mechanismus ist es ebenfalls möglich unternehmensinterne Netzwerke an Plinc. anzubinden und auf die Daten dieser Netze zuzugreifen. Dazu müssen auch diese eine OAuth2-Schnittstelle bereitstellen. Außerdem muss von diesen Netzwerken ein weiteres Protokoll zur Verfügung gestellt werden, welches beschreibt, wie Plinc. die Daten nach der Autorisierung mit OAuth2 für das Profil eines Benutzers verwenden kann. Durch ein solches Plinc.-Protokoll ist es möglich, die Anbindung komplett automatisiert ohne Erweiterung des Plinc.-Systems zu ermöglichen.

Damit die Benutzer immer aktuelle Profildaten ihrer Kontakte erhalten, werden die Daten aus den sozialen Netzwerken nicht bei Plinc. gespeichert, sondern bei jedem Profilaufruf direkt von den verknüpften Netzwerken abgerufen. Um die Datenlast zu verringern, wird dies jedoch pro Anfrage an den Webserver nur einmal durchgeführt. Dies wird durch den Cache-Mechanismus von Plinc. erreicht, der auch dafür sorgt, dass Profildaten angezeigt werden können, wenn eines der sozialen Netzwerke nicht erreichbar ist oder ein Abruf der aktuellen Daten zu lange dauert.

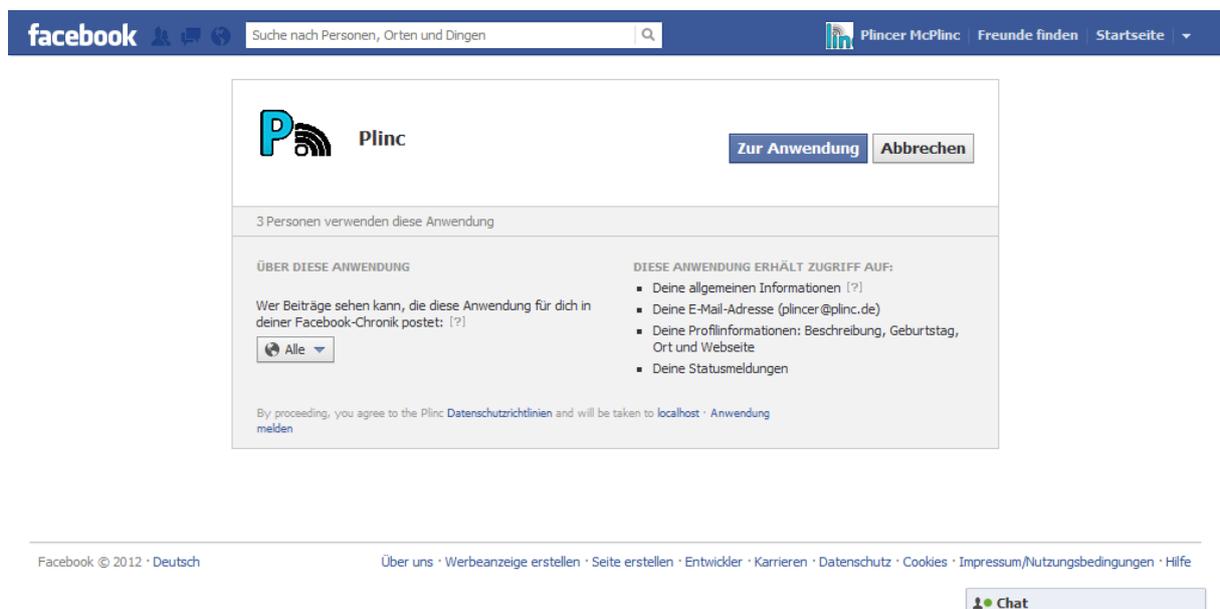


Abbildung 21: Informationen zur Verknüpfung mit Plinc. im Autorisierungs-Dialog von facebook



Abbildung 22: Freigabe des dauerhaften Zugriffs von Plinc. auf die Daten des facebook-Profiles eines Benutzers

```

publicconststring FacebookAppID = "1596586xxxxxxx";
publicconststring FacebookRedirectUrl = "http://../Manage/id/Connect/FacebookLink";
privateconststring FacebookClientSecret = "36735ffc796c7324f3471a9cxxxxxxx";

privatestring CreateUrl(string baseUri, Dictionary<string, string> uriParams)
{
    StringBuilder urlBuilder = newStringBuilder();
    foreach (var current in uriParams)
    {
        if (urlBuilder.Length > 0)
        {
            urlBuilder.Append("&");
        }
        var encoded = HttpUtility.UrlEncode(current.Value);
        urlBuilder.AppendFormat("{0}={1}", current.Key, encoded);
    }
    return baseUri + "?" + urlBuilder.ToString();
}

publicActionResult Index(int id)
{
    var uriParams = newDictionary<string, string>() {
        {"client_id", FacebookAppID}, {"response_type", "code"},
        {"scope", "user_about_me,user_birthday,user_location,
            user_status,user_website,email,offline_access"},
        {"redirect_uri", FacebookRedirectUrl.Replace("/id/", "/" + id + "/")}
    };
    ViewBag.FacebookLoginUrl = CreateUrl("https://www.facebook.com/dialog/oauth", uriParams);
    Plinc.Models.User user = Repository.Current.GetUser(id);
    return View(user);
}

publicActionResult FacebookLink(int id, string code)
{
    var uriParams = newDictionary<string, string>() {
        {"client_id", FacebookAppID},
        {"redirect_uri", FacebookRedirectUrl.Replace("/id/", "/" + id + "/")},
        {"client_secret", FacebookClientSecret}, {"code", code}
    };
    string url = CreateUrl("https://graph.facebook.com/oauth/access_token", uriParams);
    WebClient web = newWebClient();
    string response = web.DownloadString(url);
    Dictionary<string, string> responseParts = response.Split('&').Select(p => p.Split('='))
        .ToDictionary(p => p[0], p => p[1]);

    Plinc.Models.User user = Repository.Current.GetUser(id);
    user.SetAccessToken("facebook", responseParts["access_token"]);
    return RedirectToAction("Facebook");
}

```

Listing 5: Auszug des Connect-Controllers zur Verbindung mit dem facebook-Profil eines Benutzers per OAuth2

3.2 Plincing

Das Verlinken mit einem Meeting- oder Kontaktchannels wird Plincen genannt. Dieses Plincing ist auf verschiedene Arten möglich, sowohl explizit als auch implizit. Der Prototyp unterstützt davon lediglich das explizite Plincen per QR-Code, welches auch der typischste Anwendungsfall von Plinc. ist.

3.2.1 Explicit Plincing

Unter explizitem Plincen wird das ausdrückliche verlinken mit einem konkreten Meeting- bzw. Kontaktchannel verstanden. Der Benutzer wählt also selbst mit welchem Meeting er sich verplincen möchte. Dies kann auf unterschiedliche Arten geschehen. Die komfortabelste und einfachste Möglichkeit bietet der Plinc.-QR-Code (siehe Abbildung 23 links). Zu jedem Meeting, das in Plinc. angelegt wird, wird ein solcher QR-Code erzeugt. Andere Benutzer können sich nun mit diesem Meeting verplincen, indem sie den QR-Code mit der Plinc.-App fotografieren und so das Meeting explizit wählen.

Eine weitere Möglichkeit des expliziten Plincens ist die Eingabe eines Codes, der das gewünschte Meeting identifiziert. Ein solcher Code ist auch in dem QR-Code enthalten, welcher bedingt durch die benötigte Eindeutigkeit und erforderlichen Sicherheitsaspekte jedoch in einer Länge, die nicht für eine manuelle Eingabe geeignet ist. Für das Plincen per manueller Code-Eingabe wird deshalb ein sechsstelliger Code, bestehend aus Zahlen und Buchstaben, verwendet. Diese Form ist einfacher zu kommunizieren und einzugeben. Die Eindeutigkeit wird hierbei durch eine zeitliche Gültigkeit sichergestellt. Diese erhöht außerdem die Sicherheit und schützt vor unerlaubtem Plincen. Dies wird außerdem dadurch unterstützt, dass nur die Buchstaben zur Identifizierung eines Meetings genutzt werden und innerhalb einer Gültigkeitsperiode eindeutig einem Meeting zugeordnet sind. Es können somit also $26 \cdot 26 = 676$ gleichzeitig per Code unterschieden werden. Die Zahlen beschreiben die Buchstabenkombination eindeutig und werden zufällig generiert.

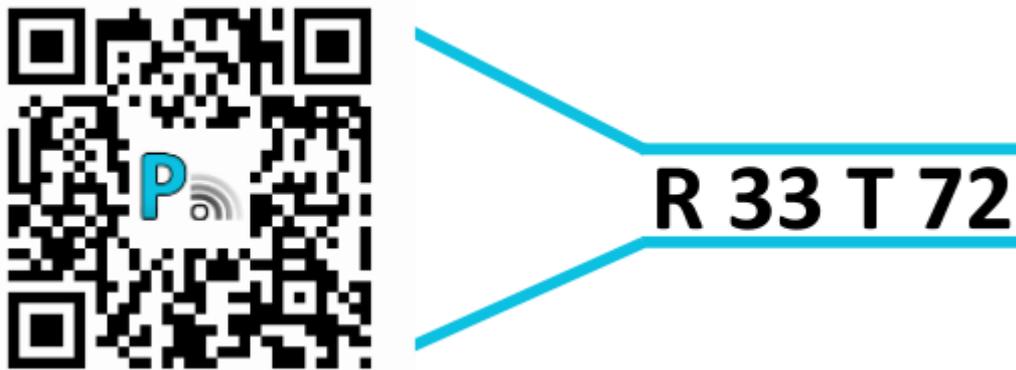


Abbildung 23: Plinc.-QR-Code (links) mit vereinfachten Plinc.-Code (rechts)

3.2.2 Implicit Plincing

Im Gegensatz zum expliziten Plincen muss der Nutzer beim impliziten Plincen das Meeting nicht ausdrücklich auswählen. Stattdessen wird das Meeting automatisch durch den aktuellen Kontext des Benutzers gewählt. Eine Möglichkeit dies zu realisieren stellt das Plincen per WiFi-Fingerprint dar. Dazu werden die Signalstärken der umliegenden w-LAN-Access-Points analysiert. So kann eine sehr exakte Position des Benutzers berechnet werden, wie in Abbildung 24 dargestellt.

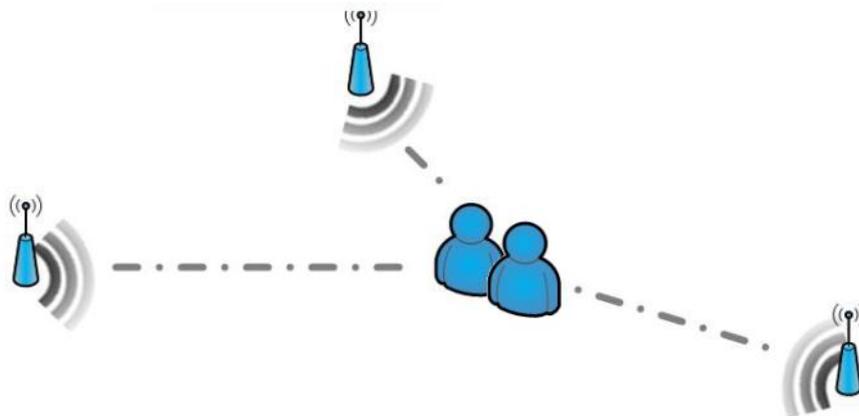


Abbildung 24: Berechnung des WiFi-Fingerprints mit Signalstärken umliegender Access-Points

Befindet sich nun die Person, die das Meeting erstellt hat, in der Nähe, besitzt sie einen ähnlichen WiFi-Fingerprint. Durch Berücksichtigung einer gewissen Toleranz kann sehr zuverlässig ermittelt werden, ob sich beide Personen in wenigen Metern Abstand voneinander entfernt, beispielsweise im gleichen Raum, entfernt befinden. Zu beachten ist, dass hierfür mindestens drei umliegende Access-Points benötigt werden. So können die Teilnehmer implizit mit dem Meeting verbunden werden, welches gerade um sie herum stattfindet, ohne dass sie es explizit auswählen müssen. Der Anwendungsfall würde dabei so

aussehen, dass die Benutzer die Plinc.-App öffnen und ohne den Zwischenschritt zur Meetingwahl direkt wie gewohnt die Benutzergruppe der anderen Beteiligten wählen können.

3.3 Codes

In diesem Kapitel sind die Model- und Controller-Klassen des Servers sowie die Page-Klassen der Windows Phone App aufgelistet. Bei allen Klassen handelt es sich um Implementierungen für den Prototyp von Plinc. Es wurde somit kein Wert auf Wartbarkeit und Effizienz gelegt, der folgende Code dient lediglich als Proof-of-Concept.

3.3.1 Model

Group.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Plinc.Models
{
    [Serializable]
    public class Group
    {
        private string name;
        private User user;
        private Dictionary<string, Dictionary<string, NetworkField>> visibilities =
            new Dictionary<string, Dictionary<string, NetworkField>>();
        private NetworkField userName = null;

        internal Group(string name, User user)
        {
            this.name = name;
            this.user = user;
        }

        public string Name
        {
            get
            {
                return name;
            }
        }

        public User User
        {
            get
            {
                return user;
            }
        }

        public string UserName
        {
            get
            {
                NetworkField userNameField = UserNameField;
                if (userNameField == null)
                {
                    return user.Id.ToString();
                }
            }
        }
    }
}
```

```

        return userNameField.Value;
    }
}

public NetworkField UserNameField
{
    get
    {
        if (userName == null)
        {
            if (visibilities.Count > 0)
            {
                return visibilities.First().Value.Values.First();
            }
            else
            {
                return null;
            }
        }

        return userName;
    }
}

public NetworkField[] VisibleFields
{
    get
    {
        return visibilities.SelectMany(p => p.Value.Values).ToArray();
    }
}

public void SetUserName(string network, string field)
{
    NetworkField networkField = GetField(network, field);
    if (networkField != null)
    {
        userName = networkField;
    }
}

private NetworkField GetField(string network, string field)
{
    if (IsVisible(network, field))
    {
        return visibilities[network][field];
    }

    return null;
}

public bool IsVisible(string network, string field)
{
    Dictionary<string, NetworkField> fields;
    if (visibilities.TryGetValue(network, out fields))
    {
        return fields.ContainsKey(field);
    }

    return false;
}

public void SetVisibility(string network, string field, bool value)
{
    if (value)
    {
        Dictionary<string, NetworkField> fields;
        if (visibilities.TryGetValue(network, out fields))
        {
            if (!fields.ContainsKey(field))
            {
                fields.Add(field, new NetworkField(user, network, field));
            }
        }
    }
}

```



```

    {
        get
        {
            return groups.Count;
        }
    }

    public NetworkField[] Fields
    {
        get
        {
            return groups.SelectMany(g => g.VisibleFields).Distinct().ToArray();
        }
    }

    public string UserName
    {
        get
        {
            if (userName == null)
            {
                string name = string.Empty;
                foreach (Group group in groups)
                {
                    string groupUserName = group.UserName;
                    if (groupUserName.Length > name.Length)
                    {
                        name = groupUserName;
                    }
                }

                if (name.Length > 0)
                {
                    userName = name.Trim();
                }
                else
                {
                    return null;
                }
            }

            return userName;
        }
    }

    public void AddGroup(Group group)
    {
        string groupUserName = group.UserName;
        if (userName != null && groupUserName.Length > userName.Length)
        {
            userName = groupUserName;
        }

        groups.Add(group);
    }
}

```

StreamEntry.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Plinc.Models
{
    [Serializable]
    public class StreamEntry
    {
        public StreamEntry(User user, string text)
        {

```

```

        User = user;
        Text = text;
    }

    public User User
    {
        get;
        private set;
    }

    public string Text
    {
        get;
        private set;
    }
}
}
}

```

User.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Plinc.Models
{
    [Serializable]
    public class User
    {
        private int id;
        private Dictionary<string, string> accessTokens = new Dictionary<string, string>();
        private Dictionary<string, Group> groups = new Dictionary<string, Group>();
        private Dictionary<Meeting, Group> meetings = new Dictionary<Meeting, Group>();
        private Dictionary<string, string> profileFields = new Dictionary<string, string>();

        [NonSerialized]
        private Dictionary<User, Profile> contacts = null;

        internal User(int id)
        {
            this.id = id;
        }

        public int Id
        {
            get
            {
                return id;
            }
        }

        public Group[] Groups
        {
            get
            {
                return groups.Values.OrderBy(g => g.Name).ToArray();
            }
        }

        public Meeting[] Meetings
        {
            get
            {
                return meetings.Keys.OrderBy(m => m.Name).ToArray();
            }
        }

        public Profile[] Contacts
        {
            get
            {

```

```

        CreateContacts();

        return contacts.Values.OrderBy(p => p.UserName).ToArray();
    }
}

public Profile Profile
{
    get
    {
        return new Profile(this, groups.Values);
    }
}

public string[] ProfileFields
{
    get
    {
        return profileFields.Keys.ToArray();
    }
}

internal IDictionary<string, string> PlincProfile
{
    get
    {
        return profileFields;
    }
}

public string GetProfileFieldValue(string key)
{
    string value;
    if (profileFields.TryGetValue(key, out value))
    {
        return value;
    }

    return string.Empty;
}

public void SetProfileField(string key, string value)
{
    key = (key ?? string.Empty).Trim();
    value = (value ?? string.Empty).Trim();

    if (string.IsNullOrEmpty(key))
    {
        return;
    }

    if (string.IsNullOrEmpty(value))
    {
        profileFields.Remove(key);
        return;
    }

    if (profileFields.ContainsKey(key))
    {
        profileFields[key] = value;
    }
    else
    {
        profileFields.Add(key, value);
    }
}

private void CreateContacts()
{
    contacts = new Dictionary<User, Profile>();
    foreach (Meeting meeting in meetings.Keys)
    {
        foreach (User user in meeting.Users)
        {

```

```

        if (user == this)
        {
            continue;
        }

        Profile profile;
        if (!contacts.TryGetValue(user, out profile))
        {
            profile = new Profile(user);
            contacts.Add(user, profile);
        }

        profile.AddGroup(user.meetings[meeting]);
    }
}

public Profile[] GetContacts(Group group)
{
    CreateContacts();
    var groupContacts = meetings.Where(m => m.Value == group).SelectMany(m => m.Key.Users);
    return contacts.Where(c => groupContacts.Contains(c.Key)).Select(c => c.Value)
        .OrderBy(p => p.UserName).ToArray();
}

public Profile GetProfile(User viewer)
{
    if (viewer == this)
    {
        return Profile;
    }

    if (viewer.contacts == null)
    {
        viewer.CreateContacts();
    }

    Profile profile;
    if (viewer.contacts.TryGetValue(this, out profile))
    {
        return profile;
    }

    return null;
}

public void JoinMeeting(Meeting meeting, Group counterpart)
{
    if (!meetings.ContainsKey(meeting))
    {
        meetings.Add(meeting, counterpart);
        meeting.AddUser(this, counterpart);

        foreach (User user in meeting.Users)
        {
            if (user.contacts != null)
            {
                user.CreateContacts();
            }
        }
    }
}

public void LeaveMeeting(Meeting meeting)
{
    if (meetings.ContainsKey(meeting))
    {
        meetings.Remove(meeting);
        meeting.RemoveUser(this);
    }
}

public bool CheckConnection(string key)
{

```

```

        key = key.ToLower();
        if (key == "plinc")
        {
            return true;
        }

        return accessTokens.ContainsKey(key);
    }

    public void SetAccessToken(string key, string token)
    {
        if (CheckConnection(key))
        {
            accessTokens[key] = token;
        }
        else
        {
            accessTokens.Add(key, token);
        }
    }

    public string GetAccessToken(string key)
    {
        if (CheckConnection(key))
        {
            return accessTokens[key];
        }

        return null;
    }

    public void CreateGroup(string name)
    {
        name = name.Trim();
        if (!string.IsNullOrEmpty(name) && !groups.ContainsKey(name))
        {
            groups.Add(name, new Group(name, this));
        }
    }

    public Group GetGroup(string name)
    {
        Group group;
        if (name != null && groups.TryGetValue(name, out group))
        {
            return group;
        }

        return null;
    }

    public void DeleteGroup(string name)
    {
        Group group;
        if (groups.TryGetValue(name, out group))
        {
            groups.Remove(name);

            foreach (Meeting meeting in meetings.Where(m => m.Value == group).Select(m => m.Key))
            {
                LeaveMeeting(meeting);
            }
        }
    }
}
}
}

```

3.3.2 Controller

ConnectController.cs

```
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Web;
using System.Web.Mvc;
using Plinc.Models;

namespace Plinc.Controllers
{
    public class ConnectController : Controller
    {
        public const string FacebookAppID = "1596586xxxxxxx";
        public const string FacebookRedirectUrl = "http://../Manage/id/Connect/FacebookLink";
        private const string FacebookClientSecret = "36735ffc796c7324f3471a9cxxxxxxx";

        private string CreateUrl(string baseUri, Dictionary<string, string> uriParams)
        {
            StringBuilder urlBuilder = new StringBuilder();
            foreach (var current in uriParams)
            {
                if (urlBuilder.Length > 0)
                {
                    urlBuilder.Append("&");
                }
                var encoded = HttpUtility.UrlEncode(current.Value);
                urlBuilder.AppendFormat("{0}={1}", current.Key, encoded);
            }
            return baseUri + "?" + urlBuilder.ToString();
        }

        public ActionResult Index(int id)
        {
            var uriParams = new Dictionary<string, string>() {
                {"client_id", FacebookAppID},
                {"response_type", "code"},
                {"scope", "user_about_me,user_birthday,user_location,
                    user_status,user_website,email,offline_access"},
                {"redirect_uri", FacebookRedirectUrl.Replace("/id/", "/" + id + "/")}
            };

            ViewBag.FacebookLoginUrl = CreateUrl("https://www.facebook.com/dialog/oauth", uriParams);

            Plinc.Models.User user = Repository.Current.GetUser(id);
            return View(user);
        }

        public ActionResult FacebookLink(int id, string code)
        {
            var uriParams = new Dictionary<string, string>() {
                {"client_id", FacebookAppID},
                {"redirect_uri", FacebookRedirectUrl.Replace("/id/", "/" + id + "/")},
                {"client_secret", FacebookClientSecret},
                {"code", code}
            };
            string url = CreateUrl("https://graph.facebook.com/oauth/access_token", uriParams);

            WebClient web = new WebClient();
            string response = web.DownloadString(url);
            Dictionary<string, string> responseParts = response.Split('&').Select(p => p.Split('='))
                .ToDictionary(p => p[0], p => p[1]);

            Plinc.Models.User user = Repository.Current.GetUser(id);
            user.SetAccessToken("facebook", responseParts["access_token"]);

            return RedirectToAction("Facebook");
        }

        public ActionResult Plinc(int id)
        {

```

```

        Plinc.Models.User user = Repository.Current.GetUser(id);

        ViewBag.Network = "plinc";

        return View(user);
    }

    [HttpPost]
    public ActionResult SavePlinc(int id, string button, string newField, string newFieldValue)
    {
        if (button == "save")
        {
            return ChangeVisibility(id);
        }

        Plinc.Models.User user = Repository.Current.GetUser(id);
        user.SetProfileField(newField, newFieldValue);

        return RedirectToAction("Plinc");
    }

    public ActionResult Facebook(int id)
    {
        Plinc.Models.User user = Repository.Current.GetUser(id);

        if (!user.CheckConnection("facebook"))
        {
            return RedirectToAction("Index");
        }
        else
        {
            ViewBag.UserProfile = NetworkCache.GetNetworkCache(user).GetNetworkProfile("facebook");
            ViewBag.Network = "facebook";

            return View(user);
        }
    }

    public ActionResult FacebookLink(int id, string code)
    {
        var uriParams = new Dictionary<string, string>() {
            {"client_id", FacebookAppID},
            {"redirect_uri", FacebookRedirectUrl.Replace("/id/", "/" + id + "/")},
            {"client_secret", FacebookClientSecret},
            {"code", code}
        };

        string url = CreateUrl("https://graph.facebook.com/oauth/access_token", uriParams);

        WebClient web = new WebClient();
        string response = web.DownloadString(url);
        Dictionary<string, string> responseParts = response.Split('&').Select(p => p.Split('='))
            .ToDictionary(p => p[0], p => p[1]);

        Plinc.Models.User user = Repository.Current.GetUser(id);
        user.SetAccessToken("facebook", responseParts["access_token"]);

        return RedirectToAction("Facebook");
    }

    [HttpPost]
    public ActionResult ChangeVisibility(int id)
    {
        var user = Repository.Current.GetUser(id);

        foreach (string key in Request.Form.AllKeys)
        {
            string[] keyParts = key.Split('-');
            if (keyParts.Length == 3)
            {
                string network = keyParts[0];
                string field = keyParts[1];
                string group = keyParts[2];
                bool selected = (Request.Form[key] != "false");
                user.GetGroup(group).SetVisibility(network, field, selected);
            }
        }
    }

```

```

        }
    }
    return RedirectToAction("Index");
}
}
}

```

ContactsController.cs

```

using System.Web.Mvc;
using Plinc.Models;

namespace Plinc.Controllers
{
    public class ContactsController : Controller
    {
        public ActionResult Index(int id, string selection)
        {
            User user = Repository.Current.GetUser(id);
            ViewBag.Group = user.GetGroup(selection);
            return View(user);
        }

        public ActionResult Groups(int id)
        {
            User user = Repository.Current.GetUser(id);
            return View(user);
        }

        public ActionResult Show(int id, int selection)
        {
            User viewer = Repository.Current.GetUser(id);
            User user = Repository.Current.GetUser(selection);
            Profile profile = user.GetProfile(viewer);
            if (viewer == null || user == null || profile == null)
            {
                return RedirectToAction("Index");
            }

            return View(profile);
        }
    }
}

```

GroupsController.cs

```

using System.Web.Mvc;
using Plinc.Models;

namespace Plinc.Controllers
{
    public class ContactsController : Controller
    {
        public ActionResult Index(int id, string selection)
        {
            User user = Repository.Current.GetUser(id);
            ViewBag.Group = user.GetGroup(selection);
            return View(user);
        }

        public ActionResult Groups(int id)
        {
            User user = Repository.Current.GetUser(id);
            return View(user);
        }

        public ActionResult Show(int id, int selection)
        {

```

```

        User viewer = Repository.Current.GetUser(id);
        User user = Repository.Current.GetUser(selection);
        Profile profile = user.GetProfile(viewer);
        if (viewer == null || user == null || profile == null)
        {
            return RedirectToAction("Index");
        }

        return View(profile);
    }
}

```

MeetingController.cs

```

using System;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Linq;
using System.Net;
using System.Web.Mvc;
using Gma.QrCodeNet.Encoding;
using Gma.QrCodeNet.Encoding.Windows.Controls;
using Plinc.Models;

namespace Plinc.Controllers
{
    public class MeetingsController : Controller
    {
        private const int QrLogoWidthHeight = 55;

        public ActionResult Index(int id)
        {
            return View(Repository.Current.GetUser(id));
        }

        public ActionResult Create(int id, string newMeeting, string counterpart)
        {
            User user = Repository.Current.GetUser(id);
            Group group = user.GetGroup(counterpart);
            if (group != null)
            {
                Meeting meeting = Repository.Current.CreateMeeting(newMeeting);
                Repository.Current.GetUser(id).JoinMeeting(meeting, group);
                return RedirectToAction("Show", new { selection = meeting.Id });
            }
            else
            {
                return RedirectToAction("Index");
            }
        }

        public ActionResult Join(int id, int selection, string counterpart)
        {
            User user = Repository.Current.GetUser(id);
            Meeting meeting = Repository.Current.GetMeeting(selection);
            if (meeting != null)
            {
                if (meeting.Users.Contains(user))
                {
                    return RedirectToAction("Show", new { selection = selection });
                }

                Group group = user.GetGroup(counterpart);
                if (group == null)
                {
                    ViewBag.Meeting = meeting;
                    return View(user);
                }
            }
            else
            {

```

```

        Repository.Current.GetUser(id).JoinMeeting(meeting, user.GetGroup(counterpart));
        return RedirectToAction("Show", new { selection = selection });
    }
}
else
{
    return RedirectToAction("Index");
}
}

public ActionResult Leave(int id, int selection)
{
    Meeting meeting = Repository.Current.GetMeeting(selection);
    if (meeting != null)
    {
        Repository.Current.GetUser(id).LeaveMeeting(meeting);
    }

    return RedirectToAction("Index");
}

public byte[] GenerateQrCode(int meetingId)
{
    IPEndPoint ip = Dns.GetHostEntry(Dns.GetHostName());
    string ipAddress = ip.AddressList.Select(i => i.ToString()
        .FirstOrDefault(i => i.IndexOf('.') < i.IndexOf('.') + 1) > -1);

    QrEncoder qrEncoder = new QrEncoder(ErrorCorrectionLevel.H);
    QrCode qrCode = qrEncoder.Encode("http://" + ipAddress + "/MobileManage/3/Meetings/Join/" +
        meetingId + " - meeting" + meetingId + "-" + ipAddress);

    Renderer renderer = new Renderer(5, Brushes.Black, Brushes.White);
    using (MemoryStream stream = new MemoryStream())
    {
        renderer.WriteToStream(qrCode.Matrix, stream, ImageFormat.Png);

        using (Image image = Image.FromStream(stream))
        using (Graphics graphics = Graphics.FromImage(image))
        using (Image logo = Image.FromFile(Server.MapPath("~/Content/images/Logo_small.png")))
        {
            graphics.DrawImage(logo, (image.Width / 2) -
                (float)Math.Ceiling(QrLogoWidthHeight / 2d), (image.Height / 2) -
                (float)Math.Ceiling(QrLogoWidthHeight / 2d), QrLogoWidthHeight,
                QrLogoWidthHeight);
            stream.Position = 0;
            image.Save(stream, ImageFormat.Png);
        }

        return stream.ToArray();
    }
}

public ActionResult Qr(int selection)
{
    return new FileContentResult(GenerateQrCode(selection), "image/png");
}

public ActionResult Show(int id, int selection)
{
    ViewBag.User = Repository.Current.GetUser(id);

    int meetingId = selection;
    Meeting meeting = Repository.Current.GetMeeting(meetingId);
    if (meeting == null)
    {
        return RedirectToAction("Index");
    }

    if (RouteData.Values["managetype"] as string == "MobileManage")
    {
        return View("Start", meeting);
    }

    return View(meeting);
}

```

```

}

public ActionResult Users(int id, int selection)
{
    ViewBag.User = Repository.Current.GetUser(id);

    int meetingId = selection;
    Meeting meeting = Repository.Current.GetMeeting(meetingId);
    if (meeting == null)
    {
        return RedirectToAction("Error", "Main");
    }

    return View(meeting);
}

public ActionResult InputStream(int id, int selection)
{
    ViewBag.User = Repository.Current.GetUser(id);

    int meetingId = selection;
    Meeting meeting = Repository.Current.GetMeeting(meetingId);
    if (meeting == null)
    {
        return RedirectToAction("Error", "Main");
    }

    return View(meeting);
}

public ActionResult Stream(int id, int selection)
{
    ViewBag.User = Repository.Current.GetUser(id);

    int meetingId = selection;
    Meeting meeting = Repository.Current.GetMeeting(meetingId);
    if (meeting == null)
    {
        return RedirectToAction("Error", "Main");
    }

    return View(meeting);
}

public ActionResult PostToStream(int id, int selection, string text, string origin)
{
    User user = Repository.Current.GetUser(id);
    ViewBag.User = user;

    int meetingId = selection;
    Meeting meeting = Repository.Current.GetMeeting(meetingId);
    if (meeting == null)
    {
        return RedirectToAction("Error", "Main");
    }

    meeting.PostToStream(user, text);

    if (origin == null)
    {
        // text parameter as random value
        return RedirectToAction("Stream",
            new { selection = selection, text = text, cache = "1" });
    }
    else
    {
        return RedirectToAction(origin, new { selection = selection });
    }
}
}
}
}

```

3.3.3 App-Pages

MainPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Threading;
using MessagingToolkit.Barcode;
using Microsoft.Devices;
using Microsoft.Phone.Controls;

namespace Plinc
{
    public partial class MainPage : PhoneApplicationPage
    {
        private PhotoCamera camera;
        private bool capturing = false;
        private string defaultMeetingDescription = string.Empty;
        private bool meetingDescriptionTextFocus = false;

        public MainPage()
        {
            InitializeComponent();

            defaultMeetingDescription = MeetingDescriptionText.Text;
        }

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            if (camera == null)
            {
                camera = new PhotoCamera();

                // fired when the camera is initialised
                camera.Initialized += HandleCameraInitialised;

                // fired when the button is fully pressed
                CameraButtons.ShutterKeyPressed += HandleCameraButtonFullPress;

                // fired when an image is available.
                camera.CaptureImageAvailable += HandleCameraCaptureImageAvailable;

                // set the VideoBrush source to the camera output
                VideoRotateTransform.CenterX = VideoRectangle.Width / 2;
                VideoRotateTransform.CenterY = VideoRectangle.Height / 2;
                VideoRotateTransform.Angle = 90;

                ViewfinderBrush.SetSource(camera);
            }

            base.OnNavigatedTo(e);
        }

        protected override void OnNavigatedFrom(System.Windows.Navigation.NavigationEventArgs e)
        {
            if (camera != null)
            {
                // unhook the event handlers
                CameraButtons.ShutterKeyPressed -= HandleCameraButtonFullPress;
                camera.CaptureImageAvailable -= HandleCameraCaptureImageAvailable;
                camera.Initialized -= HandleCameraInitialised;

                // dispose of the camera object
                camera.Dispose();

                camera = null;
            }
        }
    }
}
```

```

        base.OnNavigatedFrom(e);
    }

    private void HandleCameraInitialised(object sender, CameraOperationCompletedEventArgs e)
    {
        // set the camera resolution
        if (e.Succeeded)
        {
            var res = from resolution in camera.AvailableResolutions
                      where resolution.Width == 640 select resolution;
            camera.Resolution = res.First();
            camera.FlashMode = FlashMode.Off;
        }
    }

    private void HandleCameraButtonFullPress(object sender, EventArgs e)
    {
        if (capturing)
        {
            return;
        }

        capturing = true;
        camera.CaptureImage();
    }

    private void HandleCameraCaptureImageAvailable(object sender, ContentReadyEventArgs e)
    {
        Stream imageStream = (Stream)e.ImageStream;
        BarcodeDecoder barcodeDecoder = new BarcodeDecoder();

        Dictionary<DecodeOptions, object> decodingOptions =
            new Dictionary<DecodeOptions, object>();
        List<BarcodeFormat> possibleFormats = new List<BarcodeFormat>(1);
        Result result;

        Dispatcher.BeginInvoke(() =>
        {
            WriteableBitmap qrImage =
                new WriteableBitmap((int)camera.Resolution.Width,
                                    (int)camera.Resolution.Height);
            imageStream.Position = 0;
            qrImage.LoadJpeg(imageStream);

            possibleFormats.Add(BarcodeFormat.QRCode);
            decodingOptions.Add(DecodeOptions.PossibleFormats, possibleFormats);

            try
            {
                result = barcodeDecoder.Decode(qrImage, decodingOptions);

                string[] resultText = result.Text.Split(new char[] { '-' });
                if (resultText.Length > 0 && resultText[1].Trim().StartsWith("meeting"))
                {
                    string[] meetingText = resultText[1].Trim().Split(new string[] { "meeting" },
                                                                    StringSplitOptions.RemoveEmptyEntries);

                    int meetingId;
                    if (meetingText.Length == 1 && int.TryParse(meetingText[0], out meetingId))
                    {
                        (App.Current as App).MeetingId = meetingId;
                        NavigationService.Navigate(new Uri("/AttendMeetingPage.xaml",
                                                            UriKind.Relative));
                    }
                }

                if (resultText.Length > 1)
                {
                    (App.Current as App).ServerHostName = resultText[2].Trim();
                }
            }
            catch
            {
                return;
            }
        });
    }

```

```

        }
        finally
        {
            capturing = false;
        }
    });
}

private void HandleContactsClick(object sender, System.Windows.Input.GestureEventArgs e)
{
    NavigationService.Navigate(new Uri("/ContactsPage.xaml", UriKind.Relative));
}

private void HandleMeetingsClick(object sender, System.Windows.Input.GestureEventArgs e)
{
    NavigationService.Navigate(new Uri("/MeetingsPage.xaml", UriKind.Relative));
}

private void HandleCameraTap(object sender, System.Windows.Input.GestureEventArgs e)
{
    if (capturing)
    {
        return;
    }

    capturing = true;
    camera.CaptureImage();
}

private void HandleMeetingDescriptionTextGotFocus(object sender, RoutedEventArgs e)
{
    if (MeetingDescriptionText.Text == defaultMeetingDescription)
    {
        MeetingDescriptionText.Text = string.Empty;
    }

    MeetingDescriptionText.Foreground = new SolidColorBrush(Colors.Black);
    meetingDescriptionTextFocus = true;
}

private void HandleMeetingDescriptionTextChanged(object sender, TextChangedEventArgs e)
{
    if (meetingDescriptionTextFocus)
    {
        if (MeetingDescriptionText.Text.Trim().Length > 0)
        {
            CreateMeetingButton.IsEnabled = true;
        }
        else
        {
            CreateMeetingButton.IsEnabled = false;
        }
    }
}

private void HandleMeetingDescriptionTextLostFocus(object sender, RoutedEventArgs e)
{
    if (MeetingDescriptionText.Text.Trim().Length == 0)
    {
        MeetingDescriptionText.Text = defaultMeetingDescription;
        MeetingDescriptionText.Foreground = new SolidColorBrush(Colors.Gray);
        CreateMeetingButton.IsEnabled = false;
    }

    meetingDescriptionTextFocus = false;
}
}
}
}

```

ContactsPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using Microsoft.Phone.UserData;
using Microsoft.Phone.Tasks;
using System.Text.RegularExpressions;

namespace Plinc
{
    public partial class ContactsPage : PhoneApplicationPage
    {
        private bool loaded = false;

        public ContactsPage()
        {
            InitializeComponent();
        }

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);

            loaded = false;
            AllContactsBrowser.Navigate(new Uri(@"http://" +
                (App.Current as App).ServerHostName + "/Contacts"));
        }

        private void HandleBrowserNavigated(object sender, NavigationEventArgs e)
        {
            (sender as WebBrowser).Navigating += HandleBrowserNavigating;
        }

        private void HandleAllContactsBrowserNavigated(object sender, NavigationEventArgs e)
        {
            if (!loaded)
            {
                string content = AllContactsBrowser.SaveToString();
                int index = content.IndexOf("mobileHidden");
                if (index >= 0)
                {
                    content = content.Substring(index, content.IndexOf("/div", index) - index);
                    string[] links = content.Split('|');
                    foreach (string link in links)
                    {
                        Regex regex = new Regex("<a.*?href=\"(.*)\".*?>(.*?)</a>");
                        Match match = regex.Match(link);
                        if (match.Success)
                        {
                            string uri = match.Groups[1].Value;
                            string title = match.Groups[2].Value;

                            if (title != "All")
                            {
                                WebBrowser browser = new WebBrowser();
                                browser.VerticalAlignment = VerticalAlignment.Stretch;
                                browser.HorizontalAlignment = HorizontalAlignment.Stretch;
                                browser.Navigated += HandleBrowserNavigated;
                                ContactsPivot.Items.Add(new PivotItem() { Header = title,
                                    Content = browser });
                                browser.Navigate(new Uri("http://" +
                                    (App.Current as App).ServerHostName + uri));
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        }
    }
    loaded = true;
}
}

private void HandleBrowserNavigating(object sender, NavigatingEventArgs e)
{
    if (loaded)
    {
        e.Cancel = true;
        string uri = e.Uri.ToString();
        (App.Current as App).ContactUri = uri;
        NavigationService.Navigate(new Uri("/ContactPage.xaml", UriKind.Relative));
    }
}
}
}
}

```

MeetingsPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;

namespace Plinc
{
    public partial class MeetingsPage : PhoneApplicationPage
    {
        private bool loaded = false;

        public MeetingsPage()
        {
            InitializeComponent();
        }

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);

            loaded = false;
            MeetingsBrowser.Navigate(new Uri(@"http://" +
                (App.Current as App).ServerHostName + "/Meetings"));
        }

        private void HandleMeetingsBrowserNavigated(object sender, NavigationEventArgs e)
        {
            loaded = true;
        }

        private void HandleMeetingsBrowserNavigating(object sender, NavigatingEventArgs e)
        {
            if (loaded)
            {
                e.Cancel = true;
                string uri = e.Uri.ToString();
                (App.Current as App).MeetingUri = uri;
                NavigationService.Navigate(new Uri("/MeetingPage.xaml", UriKind.Relative));
            }
        }
    }
}

```

```
}
```

AttendMeetingPage.xaml.cs

```
using System;
using Microsoft.Phone.Controls;

namespace Plinc
{
    public partial class AttendMeetingPage : PhoneApplicationPage
    {
        public AttendMeetingPage()
        {
            InitializeComponent();
        }

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);

            int meetingId = (App.Current as App).MeetingId;
            PlincBrowser.Navigate(new Uri(@"http://" +
                (App.Current as App).ServerHostName + "/Meetings/Join/" + meetingId));
        }

        protected override void OnNavigatedFrom(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedFrom(e);

            NavigationService.RemoveBackEntry();
        }

        private void HandlePlincBrowserNavigating(object sender, NavigatingEventArgs e)
        {
            string uri = e.Uri.ToString();
            if (uri.Contains("/Meetings/Show/"))
            {
                (App.Current as App).MeetingUri = uri;
                NavigationService.Navigate(new Uri("/MeetingPage.xaml", UriKind.Relative));
            }
        }
    }
}
```

MeetingPage.xaml.cs

```
using System;
using System.Net;
using System.Windows;
using Microsoft.Phone.Controls;

namespace Plinc
{
    public partial class MeetingPage : PhoneApplicationPage
    {
        private string postToStreamUri;

        public MeetingPage()
        {
            InitializeComponent();
        }

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);

            string showUri = (App.Current as App).MeetingUri;
            string usersUri = showUri.Replace("/Show/", "/Users/");
            string streamUri = showUri.Replace("/Show/", "/Stream/");
        }
    }
}
```

```

        postToStreamUri = showUri.Replace("/Show/", "/PostToStream/") + "?text=";
        StreamPostButton.IsEnabled = true;

        UsersBrowser.IsScriptEnabled = true;
        UsersBrowser.Navigate(new Uri(usersUri));
        StreamBrowser.IsScriptEnabled = true;
        StreamBrowser.Navigate(new Uri(streamUri));
    }

    private void HandleStreamPostButtonClick(object sender, RoutedEventArgs e)
    {
        StreamBrowser.Navigate(new Uri(postToStreamUri + HttpUtility.UrlEncode(StreamText.Text)));
        StreamText.Text = string.Empty;
    }
}

```

ContactPage.xaml.cs

```

using System;
using Microsoft.Phone.Controls;

namespace Plinc
{
    public partial class ContactPage : PhoneApplicationPage
    {
        public ContactPage()
        {
            InitializeComponent();
        }

        protected override void OnNavigatedTo(System.Windows.Navigation.NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);

            ContactBrowser.Navigate(new Uri((App.Current as App).ContactUri));
        }
    }
}

```

4 Business Case

4.1 Kostenpositionen

Die Kosten für das Produkt Plinc. setzen sich zusammen aus den einmaligen Entwicklungskosten der APP und der Browserversion sowie aus regelmäßig auftretenden Periodenkosten. Da die Serverinfrastruktur preiswert „aus der Cloud“ bezogen werden kann entstehen hauptsächlich Personalkosten für z.B. den Vertrieb, den Support und die Weiterentwicklung der APP.

Position	Anzahl	mtl. Gehalt	mtl Kosten
Projektleiter IT Entw.	1	18.000 €	18.000 €
Wirtschaftsinformatiker	1	10.000 €	10.000 €
Programmierer	5	12.000 €	60.000 €
Spesen, Programme		7000 €	7.000 €
Patent, Puffer		5000 €	5000 €
Summe			100.000 €
Gesamt	6 Monate x 100.000 € = 600.000 €		

Mtl. Gehalt bezieht sich auf Arbeitgeberkosten bzw. Auftragsprogrammierung

Tabelle 1: Entwicklungskosten

Abteilung	Anzahl	mtl. Gehalt	mtl. Kosten
Projektleitung	1	11.000€	11000 €
Produktvertrieb	4	4500 €	18000 €
Gewinnung Werbepartner	4	4500 €	18000 €
Marketing	1	7000 €	7000 €
Support	2 - 4	4500 €	9000 € - 18000 €
Lizenzbearbeitung/ Verwaltung	1 - 2	3500 €	3500-7000 €
Weiterentwicklung	2	12000 €	24000 €

Summe			90.500 € - 103000€

Tabelle 2: Periodenkosten Monat 1-35

Abteilung	Anzahl	mtl. Nebenkosten	mtl. Kosten
Projektleitung	1	1000 € (Büro, Spesen)	1000 €
Produktvertrieb	4	500 € (Dienstwag.)	2000 €
Gewinnung Werbepartner	4	500 € (Dienstwag.)	2000 €
Marketing	1	8000 €(Budget)	8000 €
Support	2 – 4	500 € (Büro)	1000 € -2000 €
Lizenzbearbeitung/ Verwaltung	1-2	500 € (Büro)	500€ - 1000 €
Serverinfrastruktur		500+0,01€ x Nutzer	500 € -12.246 €
Weiterentwicklung	2	500€ (Büro)	1000 €
Summe			15.000 € - 29.246 €

Tabelle 3: Periodenkosten Monat 1-35

Der Anstieg des Aufwands für Support, Lizenzbearbeitung und Serverinfrastruktur liegt insbesondere in der ab Jahr zwei schnell ansteigenden Nutzerzahl begründet.

Abteilung	Anzahl	mtl. Gehalt	mtl. Kosten
Projektleitung	1	18.000€	11000 €
Produktvertrieb	8	4500 €	36000 €
Gewinnung Werbepartner	4	4500 €	18000 €
Marketing	1	7000 €	7000 €
Support	4-14	4500 €	18000 € - 63000 €
Lizenzbearbeitung/ Verwaltung	2-9	3500 €	7000-35000 €
Weiterentwicklung	4	12000 €	48000 €
Summe			145.000 € - 200000 €

Tabelle 4: Periodenkosten Monat 36-60

Abteilung	Anzahl	mtl. Nebenkosten	mtl. Kosten
Projektleitung	1	2000 € (Büro, Spesen)	1000 €
Produktvertrieb	8	500 € (Dienstwag.)	4000 €
Gewinnung Werbepartner	4	500 € (Dienstwag.)	2000 €
Marketing	1	13000 € (Budget)	8000 €
Support	4 – 14	500 € (Büro)	2000 € -7000 €
Lizenzbearbeitung/ Verwaltung	2-9	500 € (Büro)	1000€ - 4500 €
Serverinfrastruktur		500+0,01€ x Nutzer	12.869 € -29.283 €
Weiterentwicklung	4	500€ (Büro)	2000 €
Summe			32869 € - 57783 €

Tabelle 5: Periodenkosten Monat 36-60

Der Anstieg in der obigen Tabelle liegt in der ab Monat 36 angestrebten Expansion und intensivierten Weiterentwicklung der APP begründet.

4.2 Einnahmen

Das Produkt Plinc. nutzt eine Abwandlung der Freemium Strategie. Für den Endnutzer soll Plinc. vollumfänglich kostenlos sein, um Nutzungsbarrieren gering zu halten. Dieser Service wird werbefinanziert. Dazu dient das Hinzufügen lokaler Werbung, die sich nach dem Standort des Handynutzers richtet und in einen Meetingkontext passt. Unternehmen die eine Plinc.-Lizenz erwerben, erhalten hierfür zusätzliche Konfigurationsmöglichkeiten auf Administrationsebene.

4.2.1 Lizenzmodell

Der Erwerb einer Lizenz ermöglicht es dem Unternehmen weiter die Firmenbedürfnisse anzupassen und Plinc. professionell zu administrieren. Die Administratoren auf der Firmenseite können in der lizenzierten Plinc.-Variante z.B. die APP branden. Weiterhin können sämtliche Arbeitnehmerprofile aus firmeninternen Netzwerken mit Plinc. synchronisiert (insbesondere für erstmaliges Einrichten in Richtung Firma zu Plinc.) und die Einblendungen im Adressbuchs und die Kontaktchannels frei konfiguriert werden (Ausblenden von Werbung, hinzufügen von firmeninternen Supportnummern etc.). Dieses stellt insbesondere in der unternehmensübergreifenden Kommunikation einen echten Mehrwert dar (Stichwort: Wahrung der Corporate Identity). Treffen externe

Projektmitarbeiter auf Firmen, die eine Lizenzversion nutzen, ist eine Verbindung durch die werbefinanzierte Version problemlos möglich.

Lizenz	Max. Anzahl Mitarbeiter	Preis pro Monat zzgl. MwSt.
S	<10	25 €
M	<50	200 €
L	<250	1000 €
XL	≥250	5000 €

Tabella 6:Lizenzmodell

Die Lizenzgröße richtet sich nach der Anzahl der Unternehmensmitarbeiter, somit nicht nach der Anzahl an Plinc.-Nutzern. Die Einteilung der Mitarbeiter basiert auf der weit verbreiteten KMU-Definition. Die Preise sind nach dem zu erwarteten Nutzen für die lizenzwerbenden Unternehmen gestaffelt. Kleine Unternehmen werden Plinc. eher als Ersatz für einen selbstentwickelten Kundendialogstream auf ihrer Webseite oder einer Smartphone-APP nutzen. Hier stellt Plinc. mit 300 € pro Jahr eine deutlich kostengünstigere Alternative im Vergleich zu einem Upgrade der Webseite oder gar einer eigenen APP dar. Für mittlere und große Unternehmen bleiben Streams interessant, jedoch steigt bei ihnen die Meetinggröße und mehr firmeninterne Social Networks bzw. Adressdatenbanken sind vorhanden. Das Fehlen einer höheren Stufe ab 5000 Mitarbeiter wie es z.B. auf ein Unternehmen wie VW zutreffen würde, liegt darin begründet, dass Plinc. für diese einen besonders attraktiven Preis bieten möchte. Dies liegt in den von ihnen verursachten Netzeffekten begründet. Nutzen Großunternehmen wie VW die APP und setzen diese als Standard ein, tragen Sie aktiv zur Verbreitung der APP bei und bilden gleichzeitig ein Sprungbrett für spätere europaweite bzw. globale Expansion.

Auch bei den Lizenzkunden rechnen wir mit einem langsamen anfänglichen Geschäft. Nach etwa einem Jahr findet ein exponentieller Anstieg der Lizenzkunden statt, da einerseits die Netzeffekte einsetzen. Weiterhin gibt es eine Performance-Verbesserung, da durch den Einsatz von Plinc. die Unternehmenskooperation als „Best Practice“ schnell verbreitet wird. Die große Anzahl an S und M Lizenzen liegt allein in dem hohen Prozentsatz in der deutschen Wirtschaftsstruktur begründet. In Marktanteilen sind Unternehmen der L und XL Klasse die größten Kunden.

Lizenz	Max. Anzahl Mitarbeiter	Anzahl Lizenzen
S	<10	3000
M	<50	400
L	<250	100
XL	≥250	35

Tabelle 7: Lizenzenverkauf

Lizenz	Max. Anzahl Mitarbeiter	Preis pro Lizenz	Firmen	Summe pro Jahr
S	< 10	300 €	3000	900.000 €
M	< 50	2400 €	400	960.000 €
L	< 250	12000 €	100	1.200.000 €
XL	≥ 250	60000 €	35	2.100.000 €
Gesamt				5.160.000 €

Lizenzkunden nach 5 Jahren

Tabelle 8: Erlöse aus Lizenzen

4.2.2 Werbemodell

Die Werbung im Telefonbuch der APP soll vom Nutzer als Mehrwert angesehen werden. Dies erspart dem Nutzer das Suchen von Nummern wie von den örtlichen Taxis, Hotels, Lieferservices oder lokal verfügbarer Experten. Aus Sicht der Werbepartner soll die Werbung dazu dienen, ihren Service für Businesskunden, die oftmals Neukunden sind, auffindbar zu machen. Hierfür soll eine für Pay-per Lead marktübliche Umsatzbeteiligung von 10% des durchschnittlichen Transactions-Umsatzes erhoben werden. Angestrebt ist eine Win – Win – Win - Konstellation in der der Nutzer relevante Nummern im Telefonbuch findet, Werbepartner ihren Umsatz steigern und Plinc. Werbeeinnahmen generiert.

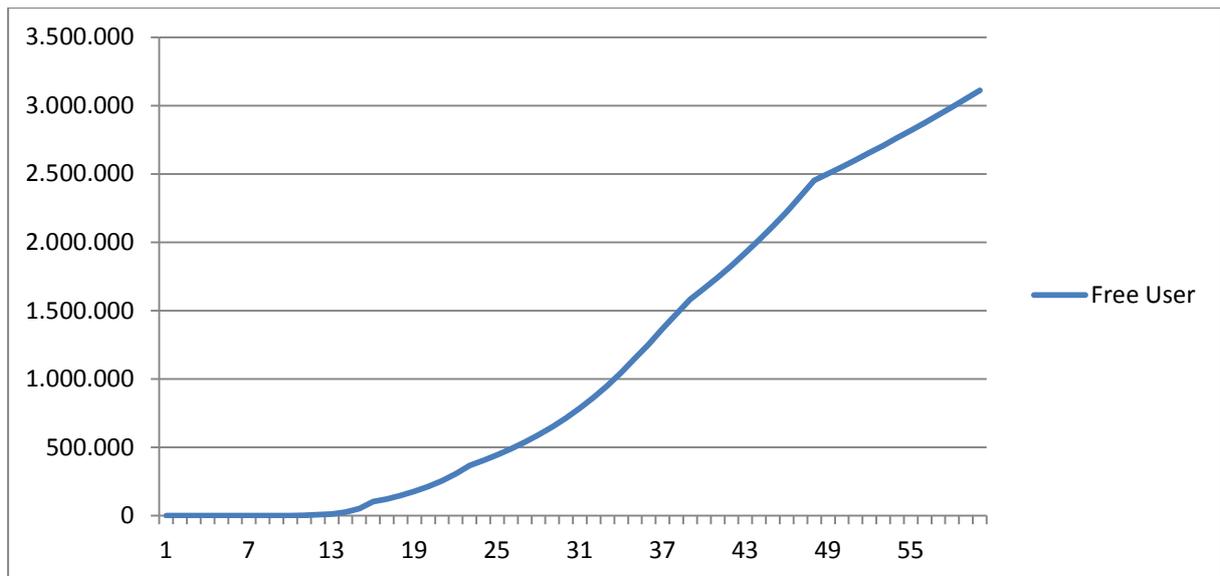


Abbildung 25: Entwicklung der Nutzer der werbefinanzierte Version

Der Graphik liegt die Annahme zu Grunde, dass in den ersten 6 Monaten (Entwicklung) keine Nutzer gewonnen werden. Anschließend wird das Produkt auf Tagungen und Konferenzen eingesetzt. Durch die Tagungen wird Plinc. in die Unternehmen integriert. Dort wird Plinc. als freie Version eingesetzt und jeder Nutzer der Plinc. nutzen möchte, wird automatisch zum Botschafter von Plinc.. Da ein Nutzer ein ganzes Meeting mit Plinc. in Verbindung bringt ist ein exponentielles Wachstum wahrscheinlich. Dies verlangsamt sich jedoch nach 3 Jahren, da die Verbreitung von Plinc. in der Zielgruppe nun teilweise dazu führt, dass nicht mehr jeder Meetingteilnehmer neu Plinc. beitrifft sondern viele bereits Nutzer sind. Ein weiterer Wachstumsschub wäre nun über eine europaweite Expansion möglich.

Die Werbeinnahmen hängen Einerseits von den Nutzerzahlen, andererseits von der Menge und Qualität der Werbepartner ab. Eingeplant ist eine kostenfreie Werbung im ersten Geschäftsjahr um Werbepartner zu gewinnen und „Successstories“ zu generieren. Im Laufe des zweiten Geschäftsjahres steigt der Werbeumsatz pro User linear von 0 Cent in Monat 12 bis 15 Cent im 26 Monat an. Dies liegt darin begründet, dass die inhaltliche und geographische Abdeckung der Werbekunden erst aufgebaut werden muss. Die erwarteten 15 Cent Werbeeinnahmen nach 26 Monaten, beruht auf der Annahme, dass entweder jeder 100ste Nutzer Plinc. intensiv mit Werbung nutzt und für 150 € Waren und Dienstleistungen über Plinc. nachfragt (z.B. zwei Übernachtungen über Plinc. bucht, Essen für alle Meetingteilnehmer in einem Restaurant reserviert) (1% x 10% Umsatzbeteiligung 150€ Umsatz = 0,15 €) oder jeder 5te Nutzer nur eine Kleinigkeit für 5,50 € in Anspruch nimmt

($20\% \times 5,50 \text{ €} \times 10\% \text{ Umsatzbeteiligung} = 0,15 \text{ €}$). Dieses Ziel liegt unter den Werbeeinahmen von Facebook, welches eine weniger kaufkraftstarke und inhomogenere Zielgruppe hat und zudem nur in der Browser Applikation wirbt. Plinc. ist lokaler, kontextorientierter und wirbt mit dem Adressbuch dort, wo der Kunde aktiv nach dem beworbenen Information sucht, anstatt diesen mit Bannerwerbung zu belästigen.

4.3 Erfolgsrechnung

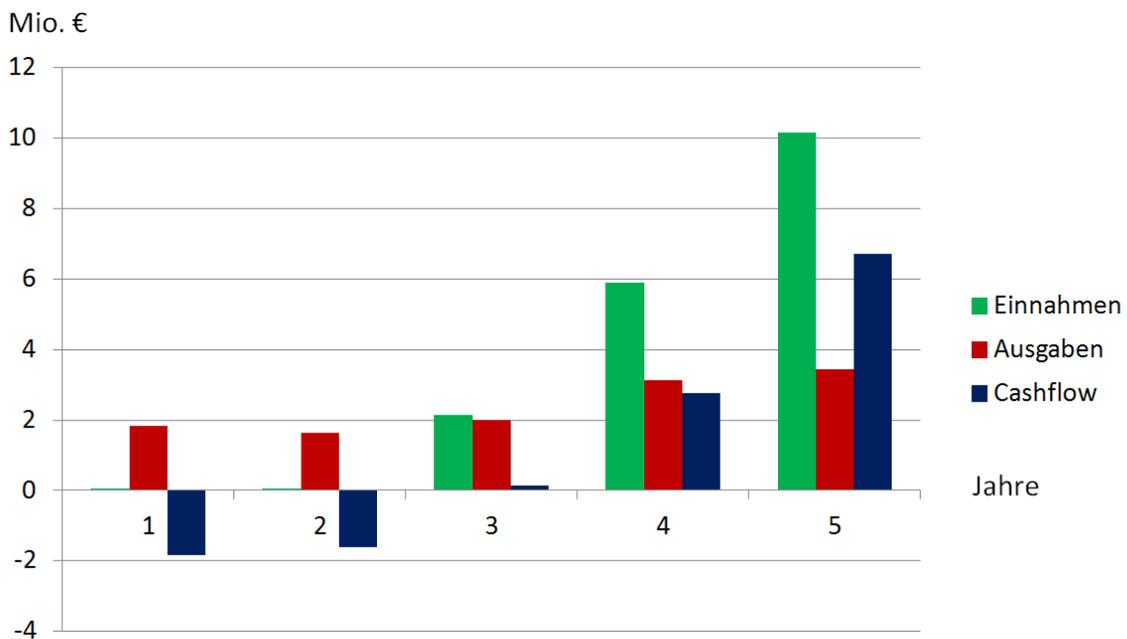


Abbildung 26: Ermittlung des Cash-Flows

Die GUV Rechnung beinhaltet die Einnahmen und Ausgaben der vorherigen Abschnitte sowie 10% Fremdkapitalzinsen. Hieraus ergibt sich ein Cashflow Break-Even nach 32 Monaten und ein Break Even nach 50 Monaten. Es werden 3,5 Mio. € benötigt um den Cashburn der Startphase zu finanzieren. Das Produkt Plinc .generiert nach 5 Jahren einen Gewinn von 0,5 Mio. € pro Monat, also 6 Mio. € pro Jahr. Dies wäre ein ROI von 170%. Die IRR beträgt bei 5 Jahren 40%, steigt jedoch bei einem längeren Betrachtungszeitraum jedoch aufgrund der 2-3 Jährigen Anlaufphase weiter an.

4.4 Monetäre Vorteile aus Kundenperspektive

Die monetären Vorteile des Einsatzes von Plinc. entstehen sowohl durch reduzierte Kosten für Eigenentwicklungen, als auch in erhöhter Effizienz in der Projektkommunikation.

Tätigkeit	Personen	pro Kontakt	Ges. Zeitaufwand	Kosten ¹
Aufnehmen der Kontaktdaten	10	1 Min,	10 Min.	8,30 €
Abtippen der Liste und verschicken	1 .	1 Min.	10 Min,	8,30 €
Einspeichern der Liste in Smartphone	10.	1 Min.	100 Min.	83 €
Summe				99,60 €

¹ Eine Person fügt 10 Kontakte je 1 min in das Adressbuch ein. Bei 10 Kontaktdaten ergeben sich 10 min pro Teilnehmer. Da jeder der 10 Teilnehmer dies vornehmen muss ergeben sich 10x10min ergo 100 min.

Die Kalkulation beruht auf Arbeitgeberkosten pro Meetingteilnehmer von 50 € / std..

Tabelle 9: Beispielrechnung: Monetäre Vorteile

Die Tabelle zeigt den Kontaktdatenaustausch in einem Meeting mit zehn Personen. Um eine reibungslose Kommunikation zu gewährleisten und damit späteres noch kostenintensiveres Suchen von Nummern zu vermeiden zeigt das Beispiel den klassischen papiergebundene Kontaktdatenaustausch. Hierbei wird zuerst manuell eine Kontaktliste in ein Protokoll aufgenommen bzw. separat erstellt dann digitalisiert und anschließend von jedem Meetingteilnehmer in sein Adressbuch überführt. Dies kostet rund 100 €. Im Vergleich bietet Plinc. die Möglichkeit dies wesentlich effizienter und gleichzeitig fehlerfrei durchzuführen. Wir rechnen für das Abfotografieren des QR Codes mit maximal zwei Minuten pro Teilnehmer was gleichzeitig erfolgen kann. Somit verzögert sich das Meeting nur um maximal zwei anstatt zehn Minuten. Hiernach sind die Adressbücher mit allen auf Wunsch auch ausführlicheren Kontakten (z.B. inkl. Bild) gefüllt. Somit reduzieren sich die Kosten für den Austausch von Kontaktdaten von 99,60 € auf 16,60 €.

Eine weitere Einsatzmöglichkeit von Plinc. sind Kontaktchannels zu Kunden, die seitens des Unternehmens mit der Corporate Identity gebrandet werden können. Dies gibt dem Unternehmen die Möglichkeit, einen auch für mobile Plattformen optimierten Kundenkontakt zu gewährleisten. Dies erspart dem Unternehmen kostenintensives Einbinden des Features in die eigene Website oder die Entwicklung einer eigenen Kontakt-APP. Da das Branding und Customizing aus dem Unternehmenslogo und –farben, der spezifischen Schriftart und der Eingabe aller relevanten Daten besteht, ist dies binnen weniger Stunden konfiguriert. Plinc. ist somit eine schnell verfügbare und kostengünstige Lösung, um Enterprise 2.0 Kundenkommunikation zu gewährleisten.

4.5 Nicht-Monetäre Vorteile aus Kundenperspektive

Plinc. ist eine innovative Lösung zur Verbesserung der Projektkommunikation. Es reduziert Stress beim Kampf mit fehlenden und fehlerhaften Kontaktdaten. Außerdem ermöglichen die Streams systematisches projekt- und meetingorientiertes Arbeiten. Hierdurch wird die unstrukturierte und ineffiziente Flut von Emails ersetzt. Anstelle des alltäglichen Abarbeitens von Emails, die lediglich nach dem Eingangszeitpunkt geordnet sind, bietet Plinc. die Möglichkeit Meeting für Meeting und Projekt für Projekt inhaltlich geordnet zu bearbeiten. Hierdurch kann fokussierter gearbeitet werden und Produktivität erhöht und gleichzeitig wahrgenommener Stress reduziert werden. Zugleich bietet die Struktur der Streams eine rudimentäre der Kommunikation während eines Meetings / Projekts. Dies erleichtert eine spätere Dokumentation des Meeting- oder Projektverlaufs. Auch der Einstieg neuer Mitarbeiter in bestehende Projekte wird durch die sofortige Verfügbarkeit der Projektkommunikation und –dokumente vereinfacht und beschleunigt. Auf Wunsch können selektiv Stakeholder mit in die Projektstreams einbezogen werden und den Projektverlauf betrachtet bzw. überwachen.

4.6 Distributionsstrategie

Plinc lebt vom „WoW“- Effekt. Die Einfachheit und das Potential lassen sich daher in einer Produktpräsentation sehr gut darstellen, wodurch Plinc. eine Direktmarketing Strategie verfolgt. Diese setzt einen persönlichen Kontakt mit dem potentiellen Kunden voraus. Auf

Tagungen und Kongressen herrschen optimale Präsentationsbedingungen, da hier viele Teilnehmer aufeinandertreffen, die in Kontakt bleiben wollen.

4.6.1 Werbe- und Vertriebsstrategie

Das Werbekonzept zu Beginn des Markteintritts, sieht wie folgt aus:

Zunächst wird die Einführung von Plinc. über Keykunden in den Markt eingebracht, die schon von sich aus ein innovatives Image verfolgen und so eine gewisse Symbolwirkung auf dem potentiellen Markt von Plinc. erzeugen werden. Der Einstieg erfolgt auf Messen, Tagungen und Kongressen.

In diesem Beispiel stellt Plinc. einem Tagungsausrichter gebrandete Kontaktstreams zu verschiedenen Themen kostenlos zur Verfügung, über die die Teilnehmer sich beispielsweise über die Tagung informieren, austauschen und in Kontakt treten können. Zur Nutzung von Plinc können sich die Teilnehmer über einen QR- Code in einer ausliegenden Broschüre, aufgehängten Plakat oder über eine Beamerpräsentation die APP herunterladen. Während der gesamten Tagung haben die Teilnehmer so die Möglichkeit die Funktionen von Plinc. zu entdecken und aktiv bei der Kontaktaufnahme mit anderen zu nutzen. Begleitend zu der Tagung wird ein Vertriebsmitarbeiter für Fragen und Interessenten die Veranstaltung begleiten.

Für den Veranstaltungsausrichter ergibt sich der Vorteil des innovativen Images, sowie der flexiblen und kostengünstigen Information seiner Gäste. Beispielsweise bei Ablaufänderungen oder ermöglicht im Anschluss der Tagung den weiteren fachlichen Austausch. Diese Form der Werbung ermöglicht es Plinc. schnell in die Unternehmen der Tagungsteilnehmer getragen zu werden und gleichzeitig die umfassenden Möglichkeiten des gebrandeten Kontaktchannels aufgezeigt zu bekommen. Aufgrund der einfachen Konfiguration könnten Interessenten Veranstaltungen direkt mit einem Entwurf der gebrandeten Version des individuellen Kontaktchannels verlassen und Firmenintern vorstellen, bei Wunsch auch durch einen Plinc. – Vertriebsmitarbeiter.

Das Vertriebsmodell baut darauf auf, dass die Tagungsteilnehmer Plinc auch aktiv in ihrem jeweiligen Unternehmen nutzen möchten. Selbst QR- Codes generieren und dadurch aktiv zur Verbreitung von Plinc. beitragen. Wird Plinc. erstmal in einem Unternehmen wie selbstverständlich genutzt, verbreiten diese über stattfindende Meetings Plinc. weiter an ihre Partnerunternehmen.

4.6.2 Vertriebsmedium APP

Die APP wird ausschließlich über den AppStore vertrieben, über den Plinc. in der kostenlosen und lizenzierten Variante gedownloadet werden kann. Sollte beim Abfotografieren Plinc. noch nicht auf dem jeweiligen Smartphone installiert sein, wird der User automatisch an den AppStore weitergeleitet, welches einen gewohnt schnellen Bezug von Plinc. ermöglicht. Entscheidet sich eine Firma für die Nutzung von Plinc., erfolgt unternehmensintern die Distribution der App an die Mitarbeiter über das W-LAN oder über aushängende QR- Codes. Risiken sind in dieser Vertriebsform nicht zu erwarten, da 99% aller Apps heute über den Appstore bezogen werden und die unternehmensinterne Verbreitung sehr unkompliziert verläuft. Die intuitive Ausgestaltung von Plinc. sind anschließende Schulungen für Mitarbeiter nicht geplant. Zur Gestaltung und aktiven Nutzung der Kontaktchannels wird eine ausführliche FAQ mitgeliefert.

4.6.3 Vermarktung Werbeflächen

Für die in der kostenlosen APP regional werbenden Unternehmen, wie dem Pizza Service, ergibt sich folgendes Modell:

Über die Anfertigung spezieller Werbematerialien, beispielsweise kompakten Broschüren, wird das Werbekonzept potentiellen Partnern näher gebracht. Während des ersten Geschäftsjahres wird einerseits die Werbeeinbindung kostenlos sein und zum anderen wird in dem ersten Geschäftsjahr gezielt in Plinc.- Verbreitungsclustern um Werbekunden geworben. Hierdurch ist es schnell möglich Erfolgsstorys zu generieren und die Werbepartner bekommen direkt Feedback, welche Wirkung ihre Werbung bei Plinc. erzeugt hat in Form von Weiterleitungsstatistiken zu ihrer Hotline.

4.7 Risiken

4.7.1 Verzögerung in der Produktentwicklung

In IT-Projekten besteht oftmals die Gefahr der Verzögerung bzw. der Überschreitung des Entwicklungsbudgets. Dies liegt insbesondere in dem schwer abzuschätzenden Entwicklungsaufwand neuer Module. Da die Plinc.-APP bereits als Proof of Concept existiert und die Hauptmodule wie z.B. der QR-Code-Reader oder API Schnittstelle zu verschiedensten Social Networks bereits existieren, halten wir das Risiko den Zeit- und Kostenrahmen zu überschreiten für gering. Hinzu kommt, dass in die Kalkulation ein Risikopuffer eingeflossen ist. Sollte dennoch der Entwicklungsaufwand höher ausfallen als

erwartet, erlaubt uns die Fremdentwicklung das hinzuziehen zusätzlicher Arbeitskräfte. Eine Überschreitung des Kostenrahmens würde zwar den Break-Even Point verzögern, aber das Projekt nicht verzögern.

4.7.2 Nutzerakzeptanz und Werbepartner

Produktneueinführungen sind immer mit einer Ungewissheit hinsichtlich der Nutzerakzeptanz verbunden. Da die APP generell in einer werbefinanzierten Version frei zugänglich ist, sind die Nutzungsbarrieren für Plinc. gering. Selbiges gilt für Werbepartner, da sie nur zahlen müssen, wenn sie tatsächlich einen Kunden über Plinc. akquirieren. Dennoch lässt sich ein Restrisiko nicht ausschließen, dass bei einer der beiden Gruppen Plinc. nicht die angestrebte Nutzerzahl erreicht. In diesem Fall müsste situativ entschieden werden, ob z.B. im Falle mangelnder Werbepartner Plinc. eine jährliche Lizenzgebühr pro Nutzer von z.B. 2 € p.a. zu erheben.

Angesichts der Einsparungen hinsichtlich Zeit und damit Personalkosten halten wir eine schleppende Gewinnung von Lizenzkunden für unwahrscheinlich. Sollte sich der Lizenzverkauf als nicht markttauglich erweisen so wäre ein Umsteigen auf eine Freeware-Lösung die allein durch kostenpflichtige Hilfe, Customizing und Support Umsatz generiert denkbar.

Generell gilt, dass sowohl alleine die Werbeeinnahmen als auch die Lizenzeinnahmen ausreichen um mehr als kostendeckend zu agieren.

4.7.3 Datenschutz und Datensicherheit

Plinc. speichert die Daten aus den importierten Social Networks nicht zwischen. Diese hierdurch besteht keine Gefahr des Datenverlustes und für die Datensicherheit. Des Weiteren werden zusätzliche Informationen des Plinc..-Profils und die Kommunikationen in den Streams verschlüsselt übermittelt. Denkbar ist auch ein Hinzufügen eines Passworts in den QR Code und eine Generierung des selbigen auf dem Gerät, sodass selbst die Administratoren von Plinc. keinen Zugang zu Nutzerdokumenten, Nachrichten und Daten haben.

4.7.4 Einführung neuer Betriebssysteme

Da Plinc. auf PC's browserbasiert ist, ist es Betriebssystem unabhängig. Dank der Nutzung des aktuellen Internetstandards HTML5 ist mit einer langfristigen Unterstützung durch

sämtliche Browser zu rechnen. Für die APP gilt, dass für jede Plattform lediglich die APP als Hülle um eine HTML5 Seite erstellt werden muss, somit nur ein Bruchteil des Programmieraufwands auf dem OS des Smartphones beruht. Für die Anpassung an aktuelle Versionen dieser Betriebssysteme stehen im Businesscase bereits zwei Programmierer zur Verfügung.

4.7.5 Backuplösungen

Für den Fall eines Serverausfalls steht in der von uns angestrebten Hosting Lösung bei Windows Azure jederzeit ausreichend Backupkapazität und ein sofort einsatzbereites Image zur Verfügung. Dies gilt auch für Lastspitzen.

4.7.6 Distributionsstrategie

Die Risiken des gesamten Marketingkonzeptes liegen in der Produktart von Plinc. selbst begründet. So entfaltet Plinc. seinen vollen funktionalen Umfang nur, wenn die meisten Meetingteilnehmer bzw. Tagungsteilnehmer die APP auch nutzen. So besteht die Gefahr, dass potentielle Kunden, also Firmen die Plinc für ihre Kommunikationsstruktur nutzen möchten, durch den elektronischen Eingriff in den Status Quo abgeschreckt werden. Des Weiteren ist die Wirkung auf den Kunden der mit einer Plinc. arbeitenden Firma in Kontakt tritt, noch nicht ausreichend vorhersagbar, ob die Skepsis oder Innovationsgedanke mehr überwiegen wird.

Um diesen möglichen Markteintrittsbarrieren entschlossen entgegen zu treten, ist vor dem Markteintritt eine universitäre Testphase geplant. Dabei wird Studenten eines neuen Jahrgangs die APP zur Verfügung gestellt und analysiert, wie sich diese auf Kontakterschließung mit neuen Kommilitonen auswirkt.

4.7.7 Konkurrenz

Plinc. stellt eine Ergänzung und keine Konkurrenz zu Social Networks da. So stellt z.B. Facebook bewusst die API bereit um solche Erweiterungen zu ermöglichen. Für noch nicht existente Wettbewerber in unserer Marktnische gilt, dass im Bereich der Social Networks durch Netzeffekte ein natürliches Monopol vorherrscht. Somit wird es ein neuer Wettbewerber schwer haben, sich gegen das dann bereits etablierte Plinc. zu behaupten. Um geographische Marktnischen als Nährboden für künftige globale Wettbewerber zu vermeiden, wird Plinc. zügig zunächst europaweit expandieren, sobald es bewiesen hat, sein Geschäftsmodell am Markt erfolgreich ist.

5 Fazit

Mit der Entwicklung des Plinc. Konzepts und des Prototyps ist es uns gelungen eine Lösung zu entwerfen um Informationen aus sozialen Netzwerken produktiv in Unternehmen einzusetzen. Während der Entwicklung des Konzeptes ist uns außerdem aufgefallen, dass beim größten sozialen Netzwerk Facebook die Möglichkeit eines selektiven Austauschs des Nutzerprofils mithilfe von beschränkten Leseberechtigungen sehr kompliziert ist. Dies ist möglicherweise nicht unbedingt ein Designfehler, sondern beruht auf dem Problem, dass jeder möglichst viel über jeden seiner „Freunde“ erfahren möchte. Würde es aber einfacher sein seine Daten privat zu halten, würde das Konzept nicht mehr funktionieren. Da sich dieses Problem insbesondere beim Verschmelzen von unternehmerischer und privater Nutzung von sozialen Netzwerken verstärkt, ist die Möglichkeit eines einfachen Berechtigungsmanagements eine essentielle Bedingung, um die Akzeptanz einer App zu erhöhen. Unser Lösungsansatz filtert Profile aus verschiedenen sozialen Netzwerken nach Freigabepreferenzen des Users und generiert als Ergebnis ein auf den Leser zugeschnittenes Business Profil.

Durch die intensive Beschäftigung mit den Zugriffsmöglichkeiten ist uns außerdem aufgefallen, wie einfach es ist vollständigen Lesezugriff auf Facebook Profile zu erlangen. Unser Prototyp zeigt, dass die API-Schnittstelle es mit einmaliger Erlaubnis ermöglicht, einen Großteil eines Profils auszulesen und in einer Datenbank zu speichern. Diese Funktion ist sehr nützlich und anwenderfreundlich für Programme wie Plinc., da vorhanden Daten nicht ein zweites Mal eingegeben werden müssen. Fraglich ist jedoch, ob allen Nutzern bewusst ist welchen umfangreichen Zugriff sie Anwendungen mit dieser Schnittstelle gewähren.

Plinc. zeigt, dass die Möglichkeiten personenbezogene Daten auszulesen mit Erlaubnis des Nutzers nahezu unbegrenzt sind. In klassischer personalisierter Webnutzung würde dies einen Wert für sich darstellen. Wir zeigen mit unserem Konzept jedoch auf, dass es möglich ist zu werben, ohne sensible Daten kommerziell zu verwerten bzw. auf Wunsch ganz auf Werbung zu verzichten, wenn dem Nutzer der richtige Mehrwert geboten wird. Dies gelingt uns mit dem Smartphone in einer Umgebung, wo klassische Werbung von Facebook bisher scheitert.

Ob unser Konzept tatsächlich dafür geeignet ist, um beide wichtigen Probleme der sozialen Netzwerke, Datenschutz aus Nutzersicht und Einnahmen (insbesondere auf dem Smartphone) auf Anbieterseite zu lösen, kann letztendlich nur die Praxis zeigen. Ein hierfür geeignetes Experiment wäre die Erstellung eines operativen Prototyps, der Kernkomponente von Plinc., dem Adressbuch inklusive Verbindungsmöglichkeit mit Xing und Facebook, sowie die Integration von einigen werbenden Firmen in das Adressbuch umsetzt. Dies könnte auf einer Tagung vorgestellt und eingesetzt werden.

Gelingt unter diesen idealen Umständen der Beweis, dass das Produkt vom Nutzer akzeptiert wird, sind wir zuversichtlich auch Geldgeber von unserer Idee begeistern zu können.

Insgesamt war das Projekt für uns alle eine lehrreiche Erfahrung, in der sicherlich auch die Mischung des Projektteams mit verschiedenem Wissen und unterschiedlichen Studienschwerpunkten für jeden die Möglichkeit bot, voneinander zu lernen und miteinander neue Lösungen zu entwickeln. Auch die Projektpräsentationen waren für uns sehr lehrreich, das sie uns gezeigt haben, wie schwer es ist eine Jury, die sich weniger intensiv mit den technischen Details von sozialen Netzwerken beschäftigt hat, von unserer Lösung zu überzeugen. Hierbei war es durchaus herausfordernd in nur 30 Minuten Plinc's wirtschaftliche und technische Machbarkeit zu präsentieren und zudem noch von seinem Nutzen zu begeistern. Wir waren immer wieder erstaunt, wie unterschiedlich die Zuhörer unsere Lösung verstanden haben. Somit haben wir während der verschiedenen Runden nicht nur durch Übung gelernt besser unter Druck zu präsentieren, sondern auch die Präsentation gezielt auf den Kenntnissstand der Zuhörer anzupassen. Dieses Wissen wird uns unabhängig vom Schwerpunkt zukünftiger Projekte dabei helfen unsere Zuhörer von unseren Lösungen zu überzeugen.

6 Anhang

1. Powerpoint Präsentation Kronberg
2. The Plinc. Song

Projects

Places

People

Plinc.

Profiles

Partner

Facilitates linking.

Enterprise 2.0

Incorporated

Projects
Places
Plinc.
People
Facilitates linking.

Marvin Osterwold
Vertrieb



Kim Köhler
Marketing



Helge Holzmann
Entwicklung



Sören Meyer
Finanzen





Plinc. ist einfaches Vernetzen

Problem → Lösung → Geschäftsmodell







Plinc. verbindet alle sozialen Netzwerke zu einem Profil



Problem → Lösung → Geschäftsmodell

	facebook.	XING	twitter	yammer The Enterprise Social Network	Plinc.
Bild					
Name	Sören M.	Sören Meyer	Sören	Sören Christian Meyer	Sören Christian Meyer
Geburtstag	19.05.1988				19.05.1988
Telefon Geschäftlich		0511-2643472		0511-2643472	0511-2643472
Mobil			0176-7538335		0176-7538335

Das Profil lässt sich benutzergruppen-spezifisch anpassen



Problem → Lösung → Geschäftsmodell

		Kunden	Partner	Vorgesetzte	Kollegen	Freunde
Bild						
Name	Sören Christian Meyer	Meyer	Meyer	Sören Christian Meyer	Sören Meyer	Sören
Geburtsdatum	19.05.1988			19.05.	19.05.1988	19.05.1988
Telefon Geschäftlich	0511-2643472	0511-2643472	0511-2643472	0511-2643472	0511-2643472	0511-2643472
Mobil	0176-7538335					0176-7538335

Das Profil kann über zwei verschiedene Channels ausgetauscht werden



Problem → Lösung → Geschäftsmodell

- **Meetingchannel**
 - Einfacher Austausch zwischen den Meetingteilnehmern

- **Kontaktchannel**
 - Einfacher Kontakt zwischen Kunden und Unternehmen

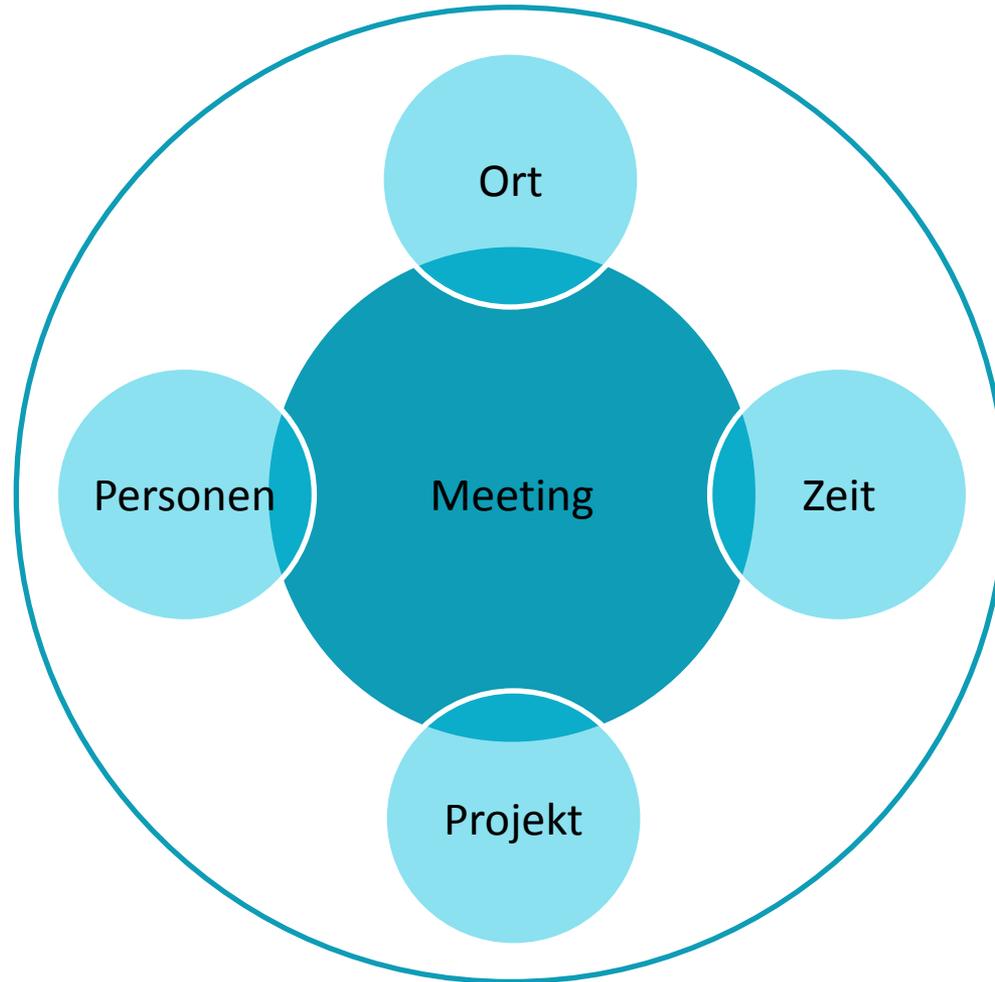


Der Meetingchannel bietet eine Teilnehmerliste und einen Stream

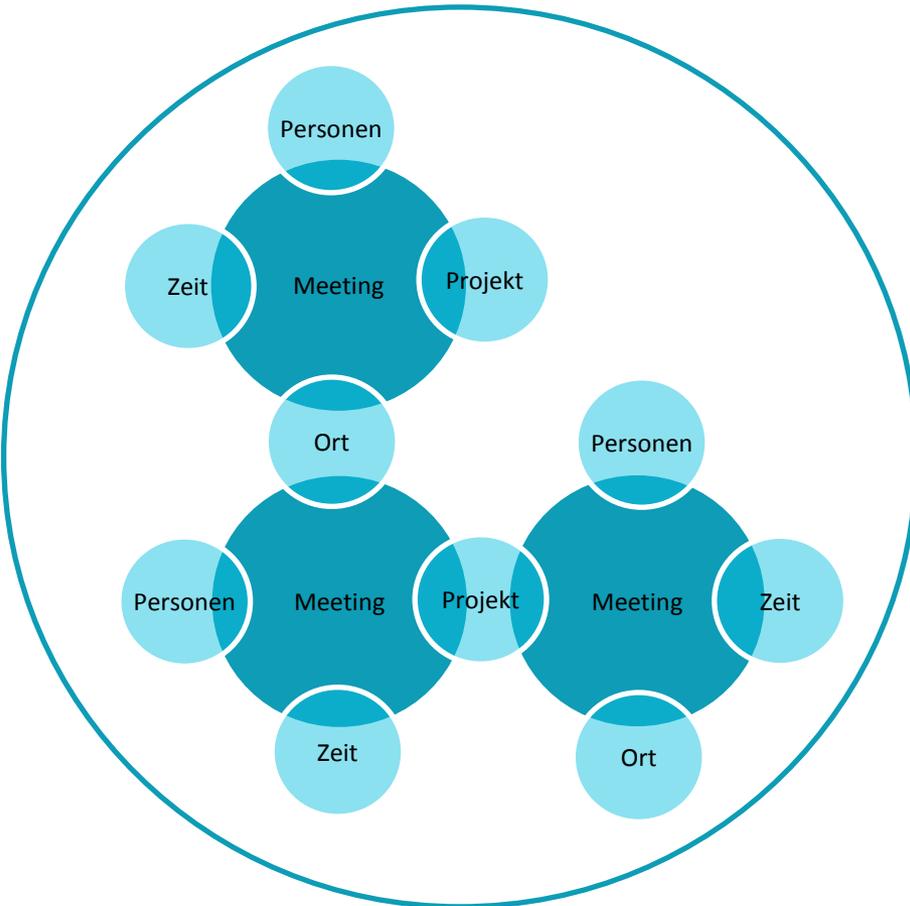
Problem → Lösung → Geschäftsmodell



Aus einem Meeting ergeben sich automatisch weitere Daten



Es werden automatisch Adressbücher erstellt



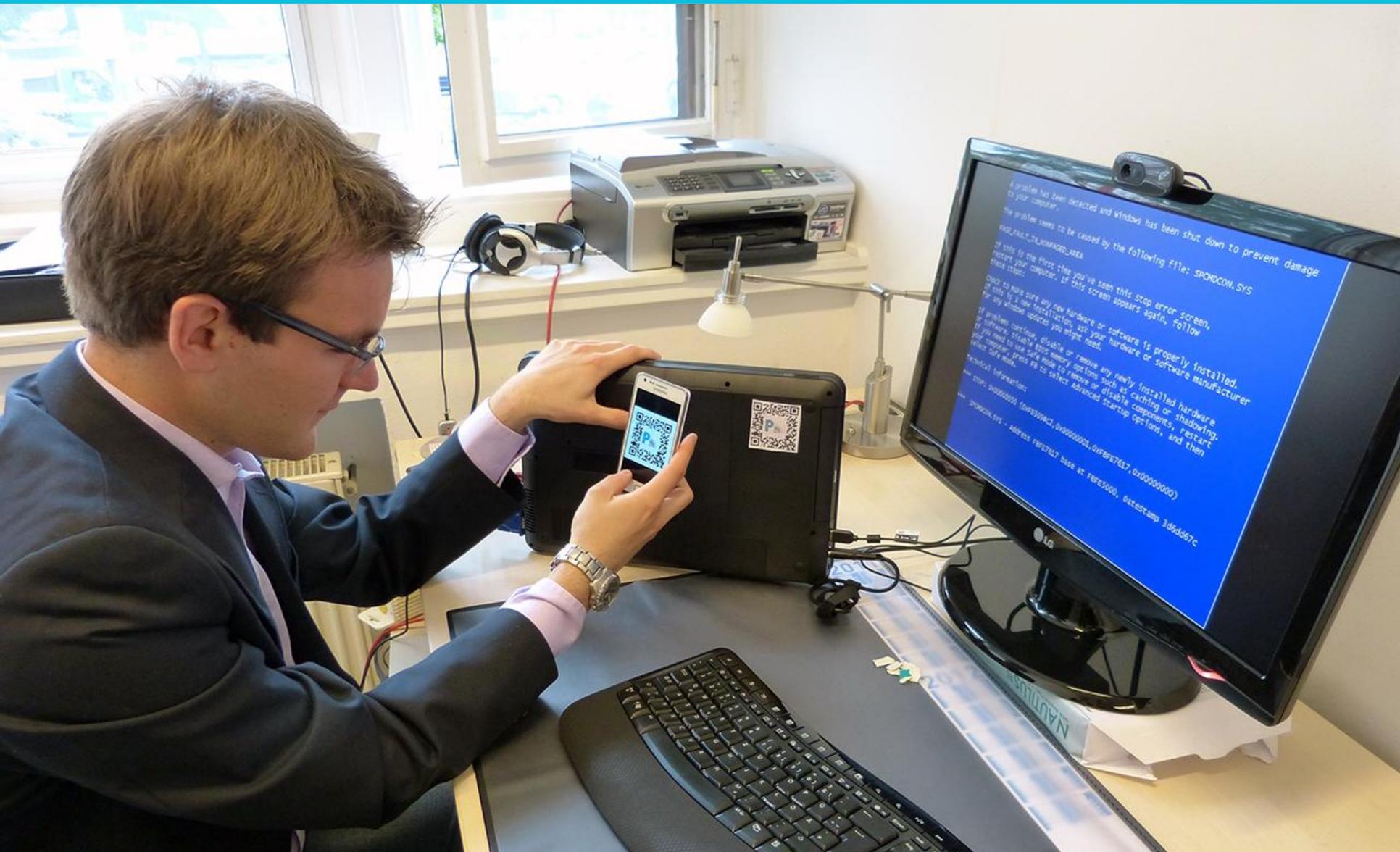
Der Kontaktchannel bietet Adressbücher, Streams und eine Dateiablage



Problem → Lösung → Geschäftsmodell





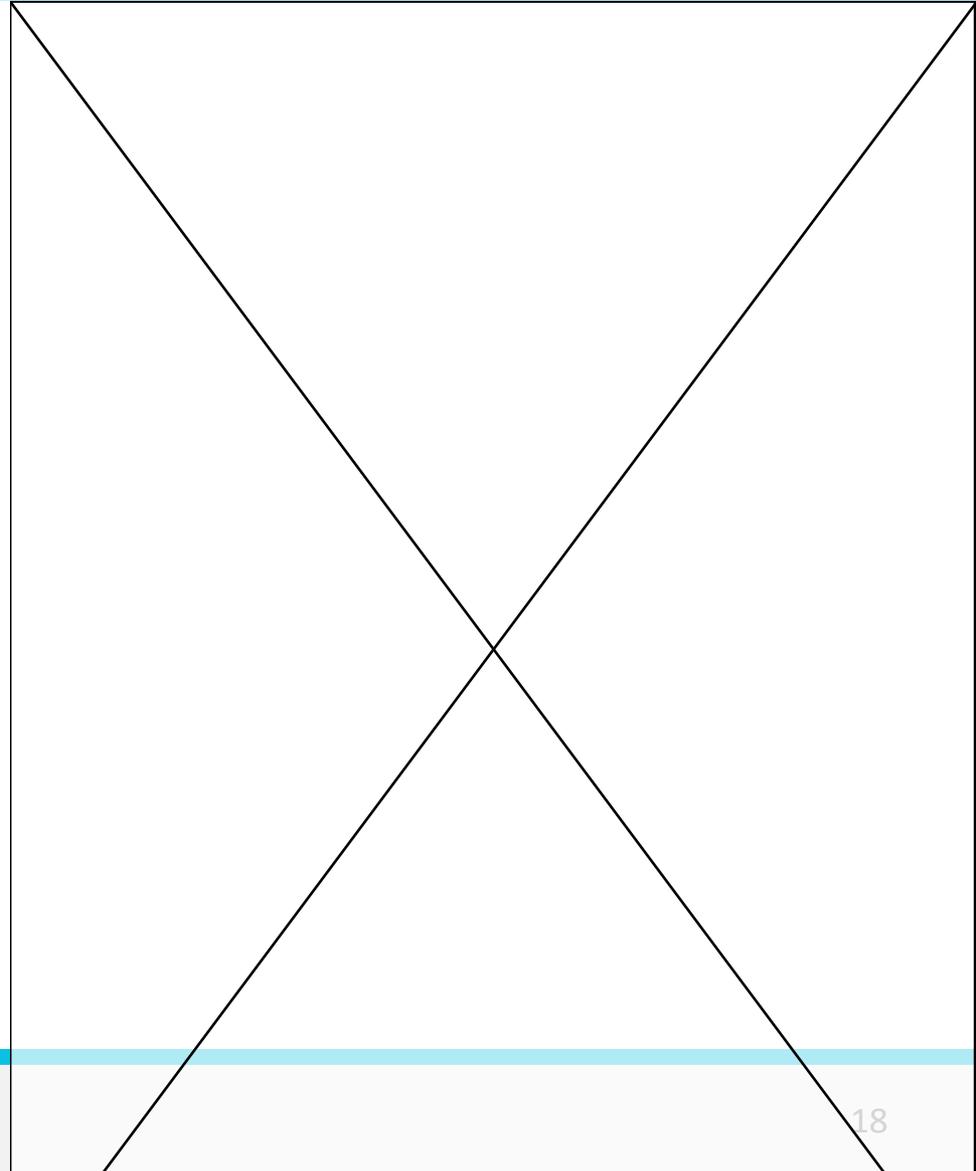


Wir bieten verschiedene Schnittstellen zu Plinc.



Problem → Lösung → Geschäftsmodell





Es gibt mehrere Möglichkeiten sich zu verplincen

- **Explicit Plincing**

- Plinc-by-QR

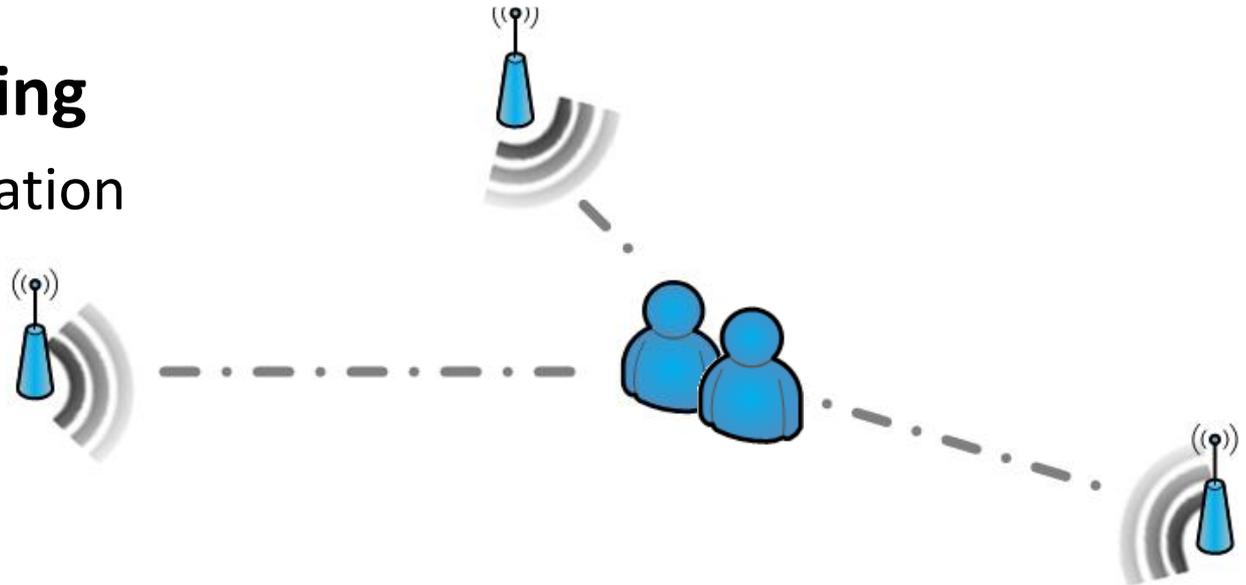
- Plinc-by-Code



R 33 T 72

- **Implicit Plincing**

- Plinc-by-Location



Plinc. finanziert sich durch Werbung und durch Lizenzen



Problem → Lösung → Geschäftsmodell

- **Kostenlose App**
 - Werbefinanziert
 - Echte Neukunden
 - Pay-per-Lead
 - 10 % des Umsatzes
 - 0,15 € monatliche Werbeinnahmen pro Nutzer



In 5 Jahren ist Plinc. das Synonym für schnelles Vernetzen



Problem → Lösung → Geschäftsmodell

Unsere Vision

- App auf allen mobilen Plattformen
- Große Bekanntheit im deutschsprachigen Raum
- Erste Wahl zum Austausch von Kontaktdaten
- Etabliertes Tool zur einfachen Projektkommunikation
- 2,8 Mio. Nutzer (werbefinanzierte Version)
- 130.000 Nutzer (lizenzierte Version)
- 6 Mio. Euro Jahresgewinn

Nach 5 Jahren ergeben sich 10,2 Mio. € Jahresumsatz



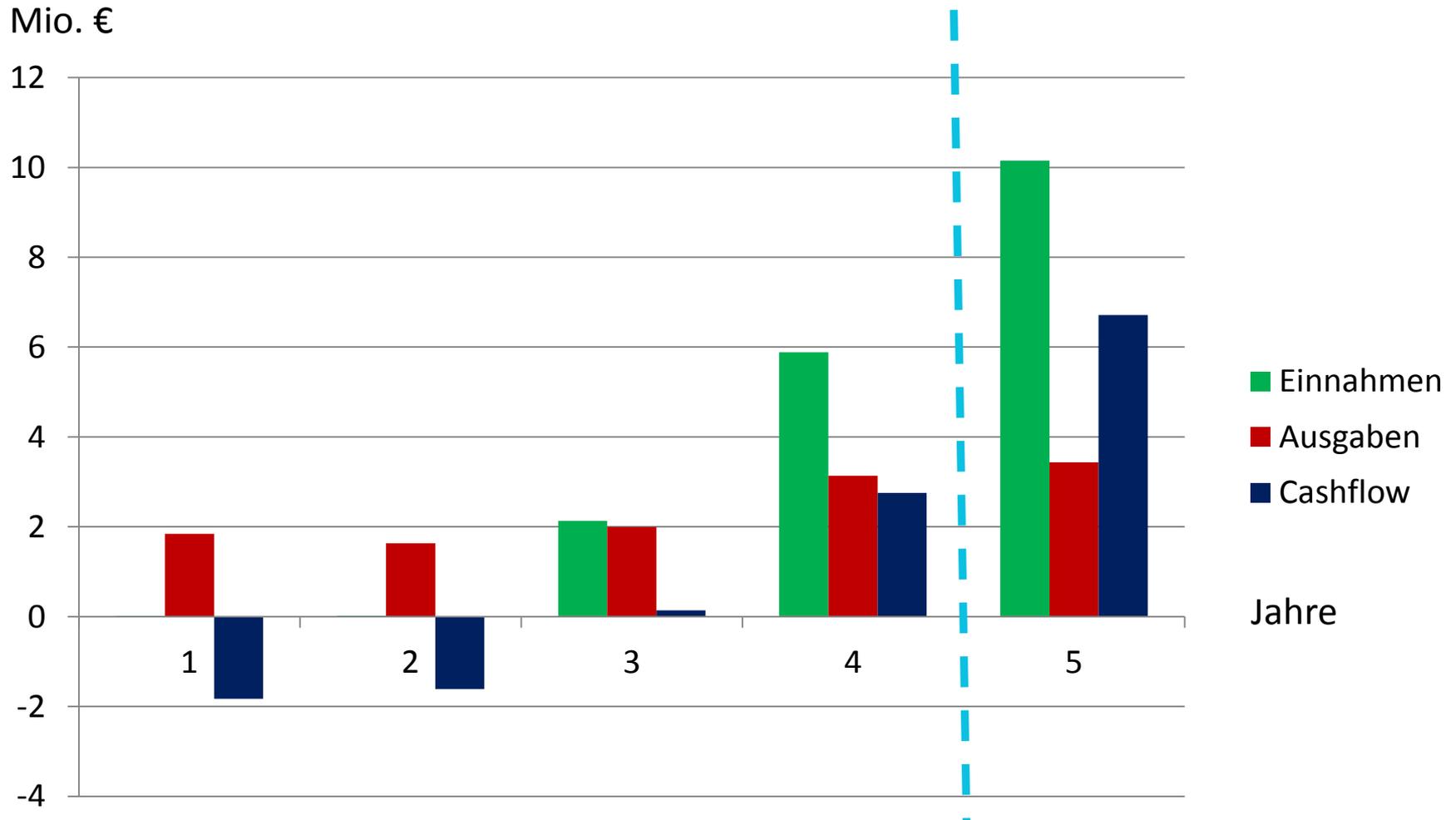
Problem → Lösung → Geschäftsmodell

Lizenz	Max. Anzahl Mitarbeiter	Preis pro Lizenz	Firmen	Summe pro Jahr
S	< 10	300 €	3000	900.000 €
M	< 50	2400 €	400	960.000 €
L	< 250	12000 €	100	1.200.000 €
XL	≥ 250	60000 €	35	2.100.000 €
Gesamt				5.160.000 €

Nutzer	Werbeeinnahmen pro Nutzer	Summe pro Jahr
2.800.000	1,80 €	5.040.000 €

Total	10.200.000 €
--------------	---------------------

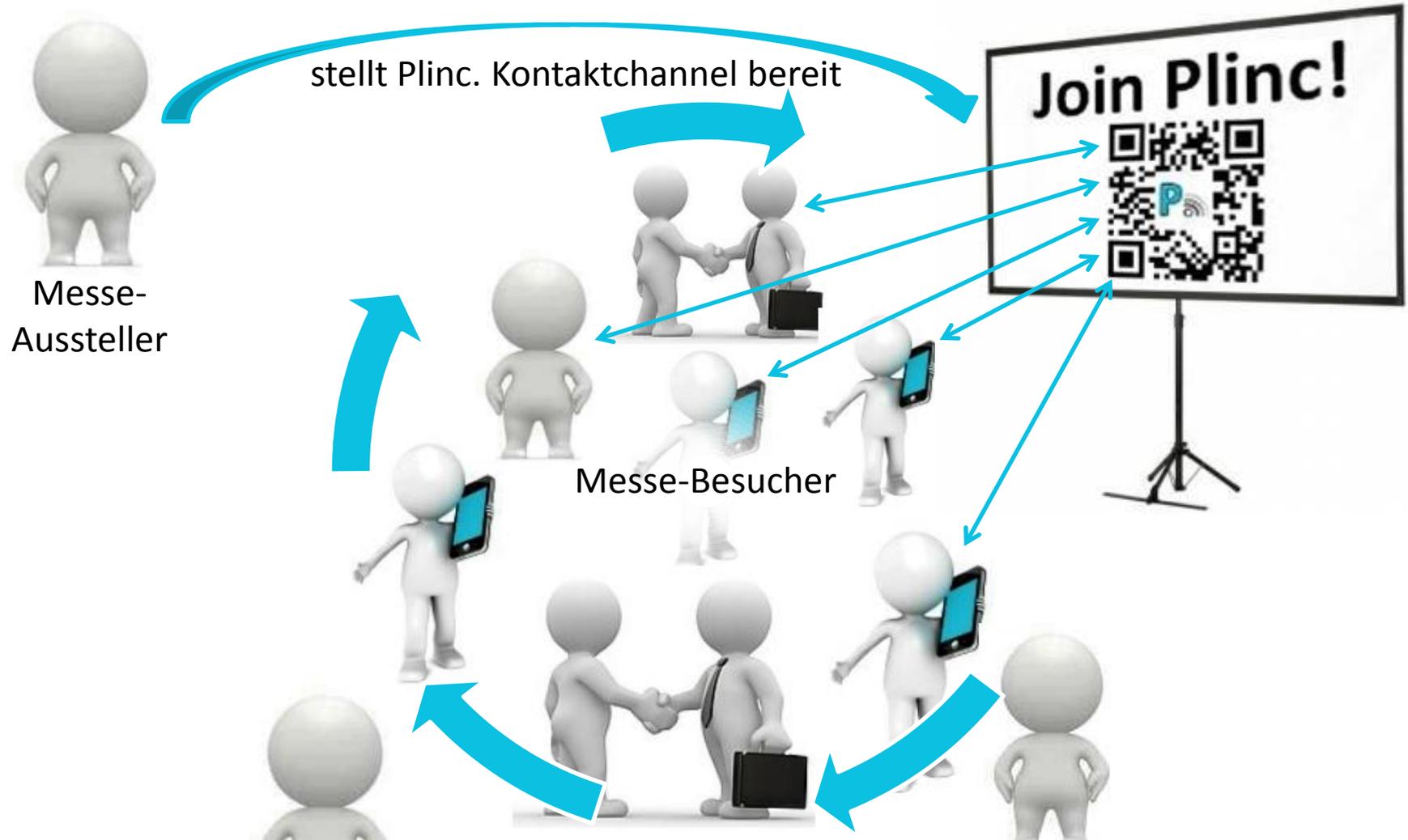
Bis zum Break-Even benötigen wir 3,5 Mio. € Investment



Wir starten mit Plinc. auf Messen, Tagungen und Kongressen



Problem → Lösung → Geschäftsmodell



Join Plinc!



www.Plinc.de • info@Plinc.de

IWI Discussion Paper Series/Diskussionsbeiträge

ISSN 1612-3646

- Michael H. Breitner, *Rufus Philip Isaacs and the Early Years of Differential Games*, 36 p., #1, January 22, 2003.
- Gabriela Hoppe and Michael H. Breitner, *Classification and Sustainability Analysis of e-Learning Applications*, 26 p., #2, February 13, 2003.
- Tobias Brüggemann und Michael H. Breitner, *Preisvergleichsdienste: Alternative Konzepte und Geschäftsmodelle*, 22 S., #3, 14. Februar, 2003.
- Patrick Bartels and Michael H. Breitner, *Automatic Extraction of Derivative Prices from Webpages using a Software Agent*, 32 p., #4, May 20, 2003.
- Michael H. Breitner and Oliver Kubertin, *WARRANT-PRO-2: A GUI-Software for Easy Evaluation, Design and Visualization of European Double-Barrier Options*, 35 p., #5, September 12, 2003.
- Dorothee Bott, Gabriela Hoppe und Michael H. Breitner, *Nutzenanalyse im Rahmen der Evaluation von E-Learning Szenarien*, 14 S., #6, 21. Oktober, 2003.
- Gabriela Hoppe and Michael H. Breitner, *Sustainable Business Models for E-Learning*, 20 p., #7, January 5, 2004.
- Heiko Genath, Tobias Brüggemann und Michael H. Breitner, *Preisvergleichsdienste im internationalen Vergleich*, 40 S., #8, 21. Juni, 2004.
- Dennis Bode und Michael H. Breitner, *Neues digitales BOS-Netz für Deutschland: Analyse der Probleme und mögliche Betriebskonzepte*, 21 S., #9, 5. Juli, 2004.
- Caroline Neufert und Michael H. Breitner, *Mit Zertifizierungen in eine sicherere Informationsgesellschaft*, 19 S., #10, 5. Juli, 2004.
- Marcel Heese, Günter Wohlers and Michael H. Breitner, *Privacy Protection against RFID Spying: Challenges and Countermeasures*, 22 p., #11, July 5, 2004.
- Liina Stotz, Gabriela Hoppe und Michael H. Breitner, *Interaktives Mobile(M)-Learning auf kleinen End-geräten wie PDAs und Smartphones*, 31 S., #12, 18. August, 2004.
- Frank Köller und Michael H. Breitner, *Optimierung von Warteschlangensystemen in Call Centern auf Basis von Kennzahlenapproximationen*, 24 S., #13, 10. Januar, 2005.
- Philipp Maske, Patrick Bartels and Michael H. Breitner, *Interactive M(obile)-Learning with UbiLearn 0.2*, 21 p., #14, April 20, 2005.
- Robert Pomes and Michael H. Breitner, *Strategic Management of Information Security in State-run Organizations*, 18 p., #15, May 5, 2005.
- Simon König, Frank Köller and Michael H. Breitner, *FAUN 1.1 User Manual*, 134 p., #16, August 4, 2005.
- Christian von Spreckelsen, Patrick Bartels und Michael H. Breitner, *Geschäftsprozessorientierte Analyse und Bewertung der Potentiale des Nomadic Computing*, 38 S., #17, 14. Dezember, 2006.
- Stefan Hoyer, Robert Pomes, Günter Wohlers und Michael H. Breitner, *Kritische Erfolgsfaktoren für ein Computer Emergency Response Team (CERT) am Beispiel CERT-Niedersachsen*, 56 S., #18, 14. Dezember, 2006.
- Christian Zietz, Karsten Sohns und Michael H. Breitner, *Konvergenz von Lern-, Wissens- und Personalmanagementssystemen: Anforderungen an Instrumente für integrierte Systeme*, 15 S., #19, 14. Dezember, 2006.
- Christian Zietz und Michael H. Breitner, *Expertenbefragung „Portalbasiertes Wissensmanagement“: Ausgewählte Ergebnisse*, 30 S., #20, 5. Februar, 2008.
- Harald Schömburg und Michael H. Breitner, *Elektronische Rechnungsstellung: Prozesse, Einsparpotentiale und kritische Erfolgsfaktoren*, 36 S., #21, 5. Februar, 2008.
- Halyna Zakhariya, Frank Köller und Michael H. Breitner, *Personaleinsatzplanung im Echtzeitbetrieb in Call Centern mit Künstlichen Neuronalen Netzen*, 35 S., #22, 5. Februar, 2008.
- Jörg Uffen, Robert Pomes, Claudia M. König und Michael H. Breitner, *Entwicklung von Security Awareness Konzepten unter Berücksichtigung ausgewählter Menschenbilder*, 14 S., #23, 5. Mai, 2008.
- Johanna Mählmann, Michael H. Breitner und Klaus-Werner Hartmann, *Konzept eines Centers der Informationslogistik im Kontext der Industrialisierung von Finanzdienstleistungen*, 23 S., #24, 5. Mai, 2008.
- Jon Sprenger, Christian Zietz und Michael H. Breitner, *Kritische Erfolgsfaktoren für die Einführung und Nutzung von Portalen zum Wissensmanagement*, 40 S., #25, 20. August, 2008.
- Finn Breuer und Michael H. Breitner, *"Aufzeichnung und Podcasting akademischer Veranstaltungen in der Region D-A-CH": Ausgewählte Ergebnisse und Benchmark einer Expertenbefragung*, 27 S., #26, 21. August, 2008.

IWI Discussion Paper Series/Diskussionsbeiträge

ISSN 1612-3646

- Harald Schömburg, Gerrit Hoppen und Michael H. Breitner, *Expertenbefragung zur Rechnungseingangsbearbeitung: Status quo und Akzeptanz der elektronischen Rechnung*, 40 S., #27, 15. Oktober, 2008.
- Matthias Paul, Hans-Jörg von Mettenheim und Michael H. Breitner, *Akzeptanz von Sicherheitsmaßnahmen: Modellierung, Numerische Simulation und Optimierung*, 30 S., #28, 16. Oktober, 2008.
- Markus Neumann, Bernd Hohler und Michael H. Breitner, *Bestimmung der IT-Effektivität und IT-Effizienz serviceorientierten IT-Managements*, 20 S., #29, 30. November, 2008.
- Matthias Kehlenbeck und Michael H. Breitner, *Strukturierte Literaturrecherche und -klassifizierung zu den Forschungsgebieten Business Intelligence und Data Warehousing*, 10 S., #30, 19. Dezember, 2008.
- Michael H. Breitner, Matthias Kehlenbeck, Marc Klages, Harald Schömburg, Jon Sprenger, Jos Töller und Halyna Zakhariya, *Aspekte der Wirtschaftsinformatikforschung 2008*, 128 S., #31, 12. Februar, 2009.
- Sebastian Schmidt, Hans-Jörg von Mettenheim und Michael H. Breitner, *Entwicklung des Hannoveraner Referenzmodells für Sicherheit und Evaluation an Fallbeispielen*, 30 S., #32, 18. Februar, 2009.
- Sissi Eklun-Natey, Karsten Sohns und Michael H. Breitner, *Building-up Human Capital in Senegal - E-Learning for School-drop outs - Possibilities of a Lifelong Learning Vision*, 40 p., #33, Juli 1, 2009.
- Horst-Oliver Hofmann, Hans-Jörg von Mettenheim und Michael H. Breitner, *Prognose und Handel von Derivaten auf Strom mit Künstlichen Neuronalen Netzen*, 34 S., #34, 11. September, 2009.
- Christoph Polus, Hans-Jörg von Mettenheim und Michael H. Breitner, *Prognose und Handel von Öl-Future-Spreads durch Multi-Layer-Perceptrons und High-Order-Neuronalnetze mit Faun 1.1*, 55 S., #35, 18. September, 2009.
- Jörg Uffen und Michael H. Breitner, *Stärkung des IT-Sicherheitsbewusstseins unter Berücksichtigung psychologischer und pädagogischer Merkmale*, 37 S., #36, 24. Oktober, 2009.
- Christian Fischer und Michael H. Breitner, *MaschinenMenschen - reine Science Fiction oder bald Realität?*, 36 S., #37, 13. Dezember, 2009.
- Tim Rickenberg, Hans-Jörg von Mettenheim und Michael H. Breitner, *Plattformabhängiges Softwareengineering eines Transportmodells zur ganzheitlichen Disposition von Strecken- und Flächenverkehren*, 38 S., #38, 11. Januar, 2010.
- Björn Semmelhaack, Jon Sprenger und Michael H. Breitner, *Ein ganzheitliches Konzept für Informationssicherheit unter besonderer Berücksichtigung des Schwachpunktes Mensch*, 56 S., #39, 3. Februar, 2009.
- Markus Neumann, Achim Plückebaum, Jörg Uffen und Michael H. Breitner, *Aspekte der Wirtschaftsinformatikforschung 2009*, 70 S., #40, 12. Februar, 2010.
- Markus Neumann, Bernd Hohler und Michael H. Breitner, *Wertbeitrag interner IT - Theoretische Einordnung und empirische Ergebnisse*, 38 S., #41, 31. Mai, 2010.
- Daniel Wenzel, Karsten Sohns und Michael H. Breitner, *Open Innovation 2.5: Trendforschung mit Social Network Analysis*, 46 S., #42, 1. Juni, 2010.
- Naum Neuhaus, Karsten Sohns und Michael H. Breitner, *Analyse der Potenziale betrieblicher Anwendungen des Web Content Mining*, 44 S., #43, 8. Juni, 2010.
- Ina Friedrich, Jon Sprenger and Michael H. Breitner, *Discussion of a CRM System Selection Approach with Experts: Selected Results from an Empirical Study*, 22 p., #44, November 15, 2010.
- Jan Bührig, Angelica Cuylen, Britta Ebeling, Christian Fischer, Nadine Guhr, Eva Hagenmeier, Stefan Hoyer, Cornelius Köpp, Lubov Lechtchinskaia, Johanna Mählmann und Michael H. Breitner, *Aspekte der Wirtschaftsinformatikforschung 2010*, 202 S., #45, 3. Januar, 2011.
- Philipp Maske und Michael H. Breitner, *Expertenbefragung: Integrierte, interdisziplinäre Entwicklung von M(obile)Learning Applikationen*, 42 S., #46, 28. Februar, 2011.
- Christian Zietz, Jon Sprenger and Michael H. Breitner, *Critical Success Factors of Portal-Based Knowledge Management*, 18 p., #47, May 4, 2011.
- Hans-Jörg von Mettenheim, Cornelius Köpp, Hannes Munzel und Michael H. Breitner, *Integrierte Projekt- und Risikomanagementunterstützung der Projektfinanzierung von Offshore-Windparks*, 18 S., #48, 22. September, 2011.

IWI Discussion Paper Series/Diskussionsbeiträge

ISSN 1612-3646

Christoph Meyer, Jörg Uffen and Michael H. Breitner, *Discussion of an IT-Governance Implementation Project Model Using COBIT and ValIT*, 18 p., #49, September 22, 2011.

Michael H. Breitner, Cornelius Köpp, Tim Rickenberg, Moritz Alsleben, Andreas Gebhardt, Alexandra Hauser, Marius Hilleke, Torsten Kühn, Mathias Maske, Hagen Meyer, Bastian Modler, Stephan Rode, Olga Rotärmel, Torsten Kühn, Farida Tazhmukhanova und Andreas Vieregge, *Beiträge zur Transformation des deutschen Energiesystems 2012*, 561 S., #50, 16. Februar, 2012.

Angelica Cuylen und Michael H. Breitner, *Anforderungen und Herausforderungen der elektronischen Rechnungsabwicklung: Expertenbefragung und Handlungsempfehlungen*, 49 S., #51, 5. Mai, 2012.

Helge Holzmann, Kim Lana Köhler, Sören C. Meyer, Marvin Osterwold, Maria-Isabella Eickenjäger und Michael H. Breitner, *Plinc. Facilitates linking. – Ein Accenture Campus Challenge 2012 Projekt*, 98 S., #52, 20. August, 2012.

