

# Offline-Fortführung von Geschäftsprozessen auf mobilen Geräten

7. Januar 2010

**Bachelorarbeit** im Fachbereich Informationstechnik im Studiengang Softwaretechnik der Hochschule Esslingen

**Verfasser** Georg Schessler

**Prüfer** Prof. Dr. Väterlein

**Betreuer** Dipl. Inform. Peter Rudolph

**Bearbeitungszeitraum** 1. September 2009 bis 7. Januar 2010

## Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Esslingen, den 7. Januar 2010 \_\_\_\_\_

## Kurzfassung

Die Verarbeitung von Geschäftsprozessen<sup>1</sup> spielt in der heutigen Zeit eine immer größer werdende Rolle. Mobilgeräten kommt dabei noch eine untergeordnete Rolle zu. Das Verbesserungspotential durch Mobilgeräte, die den Außendienst mit dem Innendienst vernetzen können, wird nicht genutzt. Diese Arbeit untersucht die Möglichkeiten zur Unterstützung von Geschäftsprozessen mit Mobilgeräten.

Aufgrund der technischen Einschränkungen und einer nicht flächendeckenden Netzverbindung sind Mobilgeräte im Gegensatz zu stationären Clients gesondert zu betrachten. So ist beispielweise eine Interaktion der Anwender über einen Webbrowser nicht möglich, wenn Anwender in einem vom Mobilfunk nicht abgedeckten Gebiet, wie zum Beispiel einem Funkloch oder einem Keller arbeiten.

EDV-gestützte Geschäftsprozesse werden heutzutage anhand eines Client-Server Modells realisiert, das eine permanente Netzverbindung voraus setzt. Für eine hohe Akzeptanz bei Außendienstmitarbeitern, wird jedoch eine Offline-Fähigkeit benötigt. Als Basis dafür dient eine auf dem Mobilgerät lauffähige Client-Anwendung.

Im Rahmen der Bachelorarbeit wurde das Spektrum von Business Process Management-Systemen analysiert und drei Konzepte erarbeitet, die eine Teilnahme mit Client-Anwendungen in Geschäftsprozessen ermöglichen. Das Konzept mit den besten Eigenschaften wurde anschließend ausgewählt. Um eine Offline-Fähigkeit zu erreichen, wurden auf dem Mobilgerät Funktionen des Servers abgebildet. Dies ermöglicht ein Arbeiten im Kontext von Geschäftsprozessen ohne Verbindung zum Server. Die Änderungen im Geschäftsprozess werden automatisch auf dem Server aktualisiert, sobald eine Netzverbindung besteht.

Das Konzept wurde anhand von Fallbeispielen untersucht. Es wurde untersucht, wie sich Mobilgerät und Business Process Management-System synchronisieren. Fallbeispiele zur Offline-Fähigkeit zeigen den Ablauf mit unter-

---

<sup>1</sup>Erklärung von Fachbegriffen erfolgt im Glossar

brochener Netzverbindung auf und gehen auf parallel arbeitende Clients in einem Geschäftsprozess ein. Abschließend wurde auf eine Implementierung unter Verwendung des PI-Mobile Frameworks eingegangen.

## Danksagung

Ich möchte mich herzlichst bei den Personen bedanken, die mich beim Verfassen dieser Bachelorarbeit unterstützt haben.

Herr Prof. Dr. Peter Väterlein möchte ich für die Treffen danken, bei denen er mit mir Problemstellungen durchgegangen ist und mir mit seiner Erfahrung hilfreiche Tipps geben konnte.

Bedanken möchte ich mich auch der PI-Data AG, die mich in diesem sehr interessantem Thema unterstützt hat. Vor allem geht mein Dank an Herr Dipl. Inform. Peter Rudolph, der mich im Verlauf der Arbeit betreute. In den zahlreichen Diskussionen entstanden viele Ideen. In seinem Bestreben, mich vieles hinterfragen zu lassen, entstand eine ausgereifte Arbeit. Mein Dank geht auch an seine Frau Rebekka Rudolph, die zur Mittagszeit wunderbar gekocht hat.

Auch meiner Familie möchte ich danken, da sie hinter mir stand und mir die Zeit gegeben hat, die ich benötigt habe.

# Inhaltsverzeichnis

<b>Eidesstattliche Erklärung</b>	<b>2</b>
<b>Kurzfassung</b>	<b>3</b>
<b>Danksagung</b>	<b>4</b>
<b>Inhaltsverzeichnis</b>	<b>5</b>
<b>1 Einleitung</b>	<b>8</b>
<b>2 Mobillösungen</b>	<b>9</b>
2.1 Mobilgeräte . . . . .	9
2.1.1 Mischformen und Begriffsunschärfe . . . . .	12
2.1.2 Besonderheiten . . . . .	13
2.2 Software . . . . .	14
2.2.1 Betriebssysteme . . . . .	14
2.2.2 Ausgelieferte Betriebssysteme . . . . .	15
2.2.3 Java ME . . . . .	16
2.3 Netzanbindungen . . . . .	18
2.3.1 GSM . . . . .	18
2.3.2 UMTS . . . . .	18
2.3.3 Bluetooth . . . . .	19
2.3.4 WLAN . . . . .	19
<b>3 Geschäftsprozesse</b>	<b>20</b>
3.1 Business Process Management . . . . .	21
3.2 Business Process Management-Systeme . . . . .	22

3.2.1	Human Workflow Mangement . . . . .	23
3.2.2	Service-Orchestration-Workflows . . . . .	24
3.3	Business Process Modeling Notation . . . . .	24
3.4	Business Process Execution Language . . . . .	25
3.5	jBPM Process Definition Language . . . . .	26
3.6	Business Process Engines . . . . .	28
3.7	Geschäftsprozess-Modellierung . . . . .	29
3.8	Beispiel für einen Geschäftsprozess . . . . .	30
3.8.1	Nodes in jPDL . . . . .	31
3.8.2	Handwerksprozess . . . . .	32
3.8.3	Grafisch dargestellter Handwerksprozess . . . . .	33
3.9	Backends in Geschäftsprozessen . . . . .	34
<b>4</b>	<b>Konzeption</b>	<b>36</b>
4.1	Requirements . . . . .	36
4.1.1	Funktionale Requirements . . . . .	36
4.1.2	Nicht funktionale Requirements . . . . .	37
4.2	Aufbau des Servers . . . . .	38
4.3	Ohne Business Process Engine auf dem Mobilgerät . . . . .	39
4.4	Business Process Engine auf dem Mobilgerät . . . . .	42
4.5	Ergänzung durch einen Backend-Simulator . . . . .	44
4.6	Mobile Prozessschritte . . . . .	45
4.7	Backend-Simulator . . . . .	46
4.8	Synchronisierung des Mobilgerätes . . . . .	48
4.9	Untersuchung der Offline-Fähigkeit . . . . .	49
4.10	Parallel arbeitende Mobilgeräte . . . . .	55
4.11	Umgang mit Versionen von Geschäftsprozessen . . . . .	56

<b>5 Implementierung</b>	<b>57</b>
5.1 PI-Mobile Framework . . . . .	57
5.2 Business Process Management-System . . . . .	58
5.2.1 Aufbau von jBPM . . . . .	59
5.3 Server-Implementierung . . . . .	60
5.3.1 Prozessbeschreibung . . . . .	60
5.3.2 Prozessinstanzen . . . . .	62
5.3.3 Aktualisierung durch den Client . . . . .	65
5.4 Client-Implementierung . . . . .	67
5.5 Backend . . . . .	67
<b>6 Zusammenfassung</b>	<b>69</b>
<b>7 Ausblick</b>	<b>70</b>
<b>Literatur</b>	<b>71</b>
<b>Abkürzungsverzeichnis</b>	<b>75</b>
<b>Glossar</b>	<b>77</b>
<b>Anhang</b>	<b>78</b>
Geplanter Zeitplan . . . . .	78
Tatsächliche Zeitverteilung . . . . .	78
Handwerksprozess aus Kapitel 3.8 . . . . .	78
Schema der exportierten Prozessinstanz . . . . .	81

# 1 Einleitung

Geschäftsprozesse dienen der ergebnisorientierten Beschreibung von betrieblichen Abläufen. Oftmals werden Geschäftsprozesse in einem Unternehmen gelebt, jedoch nicht dokumentiert. Von einer informationstechnischen Unterstützung ihrer Geschäftsprozesse versprechen sich viele Unternehmen Vorteile, wie zum Beispiel eine Verbesserung der Qualität der Geschäftsprozesse, da diese eindeutig dokumentiert werden und somit eine Grundlage für Optimierungen bieten.

Mobilgeräte, wie zum Beispiel Smartphones und Notebooks können durch ihre stetige Weiterentwicklung, dem einhergehenden Ausbau der Kommunikationsnetze und der steigenden Geschwindigkeit der Netze zunehmend in die Geschäftsprozesse einbezogen werden. Dadurch kann ein Medienbruch verhindert werden.

Zum Beispiel kann von einer informationstechnischen Integration von Mobilgeräten ein Versicherungskaufmann profitieren. Im Prozess eines Vertragsabschlusses kann ein Versicherungskaufmann im Außendienst nach Abschluss eines Kundenvertrages die Daten über sein Mobilgerät an die Zentrale übermitteln. Die Zentrale kann den Vertrag bearbeiten und dem Kunden den Vertragsabschluss schneller bestätigen.

Eine Netzverbindung ist bei Mobilgeräten jedoch nicht immer garantiert. Ein in der Tiefgarage parkender Versicherungskaufmann kann beispielsweise nicht von einer funktionierenden GSM- oder UMTS-Verbindung ausgehen. Deshalb ist die volle Funktionsfähigkeit der Anwendungen auch ohne Online-Verbindung von entscheidender Bedeutung für die Akzeptanz der Nutzung mobiler Endgeräte.

Ziel dieser Bachelorarbeit ist es daher, eine Lösung zu erarbeiten, bei der Mobilgeräte in Geschäftsprozesse integriert werden können, ohne eine ständige Netzverbindung voraus zu setzen. Ein wichtiger Aspekt dabei ist die Kommunikation des Mobilgerätes mit dem Server. Eventuell muss der Server angepasst werden, um ein reibungsloses Zusammenspiel zu ermöglichen.



## 2 Mobillösungen

Im Geschäftsbereich bieten Mobillösungen durch ihre Mobilität eine Reihe von Möglichkeiten. Durch das Personal Information Management (PIM) wird eine Verwaltung von Adressen, Telefonnummern, Terminen, Aufgaben, etc. ermöglicht. Eine weitere Möglichkeit liegt im Speichern von Notizen und Sprachaufzeichnungen. Der für diese Arbeit interessante Aspekt ist das Mobile Business, das einen Zugriff auf Unternehmensdaten bietet und der Datenverwaltung dienen kann.

### 2.1 Mobilgeräte

Es gibt eine Vielzahl verschiedener Mobilgeräte, die sich in ihrer Größe, der Bedienung und ihrer Leistung unterscheiden. Diese werden im Folgenden anhand ihrer unterschiedlichen Eigenschaften vorgestellt.

#### Mobiltelefone

Mobiltelefone, auch Handys genannt, sind tragbare Telefone, die inzwischen von allen Bevölkerungsschichten verwendet werden. Durch eine gute Netzabdeckung können diese fast flächendeckend in Deutschland verwendet werden. Typische Ausstattungsmerkmale sind:

- ein kleines Display
- Tastatur mit Mehrfachbelegung der Tasten
- Schnittstellen wie Infrarot, USB-Client, Bluetooth
- oftmals eine Kamera



Abbildung 1: Mobiltelefon

## PDAs

Ein Personal Digital Assistant (PDA) dient vorwiegend als Personal Information Manager (PIM), der eine Kalender-, Adress- und Aufgabenverwaltung ermöglicht. PDAs wurden mittlerweile durch Smartphones vom Markt zurückgedrängt, da diese zusätzlich über Telefonie- und Internetfunktionen verfügen. Typische Ausstattungsmerkmale sind:

- Touchscreen
- Bildschirmtastatur
- Schnittstellen wie Infrarot, USB-Client, Bluetooth
- PIM-Funktionen



Abbildung 2: PDA

## Smartphones

Smartphones bilden eine Verschmelzung von PDA und Handy und erweitern ein Mobiltelefon um diese Eigenschaften:

- Touchscreen
- teilweise eine QWERTZ<sup>2</sup> Tastatur
- PIM-Funktionen zur Verwaltung von persönlichen Daten wie Kalender-, Adress- und Aufgabenverwaltung



Abbildung 3: Smartphone

## Notebook

Ein Notebook, auch Laptop genannt ist, ein mobiler PC, der durch eine kompakte Größe und einem Akku auf den mobilen Bereich ausgelegt ist.

---

<sup>2</sup>QWERTZ-Tastaturen bieten fast alle Tasten einer Computertastatur



Abbildung 4: Notebook

Typische Eigenschaften sind:

- Bildschirm mit 10,4 bis 20 Zoll
- vollwertige Tastatur, meist ohne Nummernblock
- optische Laufwerke (CD, DVD)
- Schnittstellen wie USB, WLAN, Bluetooth, Firewire, CardReader
- Vollwertiges Betriebssystem (Windows, MacOS, Linux)

### 2.1.1 Mischformen und Begriffsunschärfe

Bei der Recherche wurde eine sehr ungenaue Trennung der verschiedenen Mobilgeräte ersichtlich. So gibt es von Sony Ericsson das P1i, das über einen Touchscreen, PIM-Funktionen und das bei Smartphones verbreitete Betriebssystem Symbian OS verfügt. Dennoch wird es von Sony Ericsson als Handy vermarktet.

Es sind auch Geräteklassen anzutreffen, die in den meisten Eigenschaften einer der bereits genannten Klassen entsprechen und daher nicht aufgeführt wurden. Als Beispiel sind Netbooks zu nennen, die sich von einem Notebook in der geringeren Größe, Leistung und dem Fehlen eines optischen Laufwerks unterscheiden.

### 2.1.2 Besonderheiten

Es gibt bei der Benutzung von mobilen Endgeräten viele Eigenheiten, die bei der Softwareentwicklung beachtet werden müssen. Beim Display muss von verschiedenen Größen und Farbtiefen ausgegangen werden, da diese auch innerhalb der Geräteklassen stark variieren können. Auch beschneiden Display-Tastaturen, wenn diese eingeblendet werden, das Display um etwa 30 Prozent.



Abbildung 5: Eine Bildschirmtastatur blendet einen Teil des Inhaltes aus

Von vielen Geräten wird eine Drehung des Inhalts um 90 Grad angeboten. Durch die neuen Proportionen wird der Inhalt anders dargestellt und es muss ggf. gescrollt werden. Das Umstellen erfolgt durch die Software, über eine Taste, über Neigungssensoren oder durch das Ausklappen einer eventuell vorhandenen Tastatur.

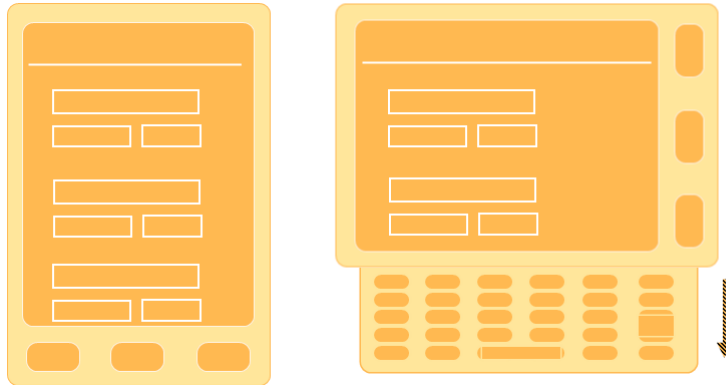


Abbildung 6: Im Querformat wird der Bildschirminhalt von unten beschnitten

## 2.2 Software

Sowohl auf der Ebene der Betriebssysteme, als auch auf der Softwareentwicklung bieten mobile Endgeräte eine große Vielfalt. Daher werden geläufige Betriebssysteme bei Mobilgeräten aufgezählt und auf Java ME eingegangen, das bei den meisten Mobilgeräten anzutreffen ist.

### 2.2.1 Betriebssysteme

Ein Mobilgerät besteht aus einem Prozessor, Arbeitsspeicher, Bildschirm, Eingabegeräten, Funkmodulen und vielem mehr. Um diese in der Softwareentwicklung einfach nutzen zu können ist eine Abstraktion der Geräte mit Hilfe von Schnittstellen nötig. Betriebssysteme stellen Schnittstellen zur Verfügung und verwalten Programme und Ressourcen. Die folgenden Betriebssysteme werden bei Mobilgeräten häufig verwendet.

**Android** Ein auf dem Linux-Kernel basierendes Betriebssystem der Open Handset Alliance, deren Vorreiter Google ist.

**iPhone OS** Ein auf Unix basierendes Betriebssystem für das iPhone von Apple.

**Palm OS** Ein von Palm (später PalmSource) entwickeltes Betriebssystem, dessen Zukunft laut einem Bericht der Networkworld [net09] ungewiss ist.

**BlackBerry OS** Das proprietäre Betriebssystem vom kanadischen Unternehmen RIM, das in allen BlackBerrys Verwendung findet.

**Symbian OS** Ein auf die ARM-Architektur aufbauendes Betriebssystem von Symbian.

**Windows Mobile** Das mobile Betriebssystem von Microsoft, das durch seine Gestaltung eine Ähnlichkeit zu Windows auf Desktops hat.

## 2.2.2 Ausgelieferte Betriebssysteme

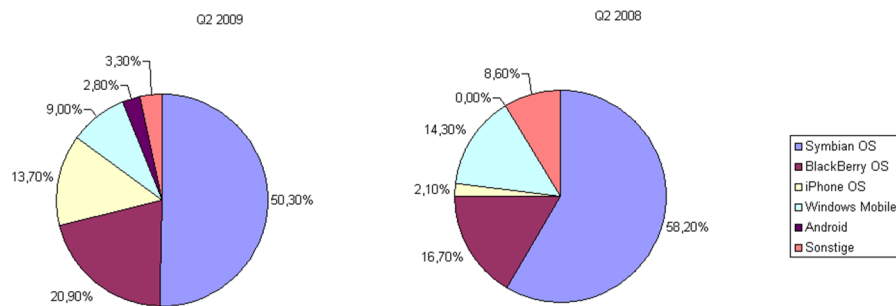


Abbildung 7: Ausgelieferte Betriebssysteme im zweiten Quartal 2009

In einer globalen Marktuntersuchung von canalsys [can09] wurde anhand der ausgelieferten Smartphones der Anteil der Betriebssysteme für Smartphones gemessen. Die Statistik bezieht sich auf das zweite Quartal 2009 und es ist eine eindeutige Dominanz durch Symbian zu sehen. Eine große Rolle spielen auch RIM mit dem Blackberry und Apple, wobei Apple ihren Anteil innerhalb eines Jahres von 2,1% auf 13,7% steigern konnten. Der Anstieg Apples ist dem iPhone zu verdanken, das im Juni 2007 seinen Verkaufsbeginn hatte. Der Markt ist ständig in Bewegung und es ist keine Stabilisierung der Aufteilung zu sehen.

### 2.2.3 Java ME

Es bestehen mehrere Möglichkeiten, Software für mobile Endgeräte zu entwickeln. Zum einen stehen oftmals C und C++ zur Auswahl, die durch ihren compilierten Code in Sachen Performance einen großen Vorteil haben. Zum anderen wird von SUN Microsystems für ressourcenbeschränkte Geräte Java ME angeboten, das sich durch folgende Eigenschaften auszeichnet:

- Plattformunabhängigkeit
- objektorientierte Programmierung
- bedarfsgerechte Bereitstellung ohne Installation
- Sicherheit der Daten durch ein Sandbox-Konzept<sup>3</sup>

Im Gegensatz zum beim Desktop verwendeten Java SE bietet Java ME ein optimiertes Speicherverhalten, da zum Beispiel der Bytecode-Verifikator ca. 100 Byte Arbeitsspeicher anstatt der sonst üblichen 100 kB Arbeitsspeicher für die Verifikation einer typischen class-Datei benötigt. Durch diese Vorteile konnte sich Java ME auf mobilen Endgeräten mit wenig Leistung bis hin zu performanteren Geräten wie Navigationsgeräten durchsetzen.

### Architektur

Die Architektur von Java ME basiert auf einem Kern, der Konfiguration genannt wird. Dieser muss vom Endgerät komplett unterstützt werden, um Java ME-fähig zu sein und beinhaltet die Java Virtual Machine (JVM) mit-samt Bibliotheken.

Darauf aufbauend sind die Profile, die optional eingebunden werden können und die Konfiguration um weitere Funktionen erweitern. Es handelt sich um eine Klassenbibliothek, in der die zusätzlichen Funktionen implementiert sind.

---

<sup>3</sup>Ein Verfahren, in dem ein Programm in einer virtuellen Umgebung ausgeführt wird, um Zugriff auf andere Programme zu verhindern.



Abschließend können optionale Pakete eingebunden werden, die in den Java Community Process(JCP)-Arbeitsgruppen zusammengestellt werden und die Portabilität einer Anwendung zu anderen Geräteherstellern eingrenzen, da diese nicht durch alle Hersteller implementiert werden.

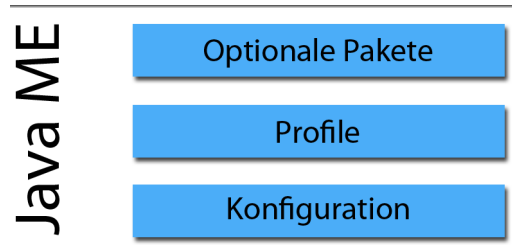


Abbildung 8: Architektur von Java ME

### **Connected, Limited Device Configuration**

Die bei mobilen Endgeräten übliche Konfiguration ist CLDC. Es handelt sich dabei um eine ressourcenschonende virtuelle Maschine, die mit geringen Voraussetzungen an die Hardware zurechtkommt. Dies wurde durch viele Einschränkungen realisiert, zu der eine minimale Klassenbibliothek zählt.

### **Mobile Information Device Profile**

Das mittlerweile in den meisten Mobiltelefonen verfügbare Profil MIDP ergänzt die Klassenbibliothek von CLDC. Zu den Features der aktuellen Version 2.1 zählen:

- User interface (Aus- und Eingabe, sowie Erweiterungen für Spiele)
- Soundausgabe
- Transaktionssicherheit durch HTTPS
- Over The Air-Provisioning zur drahtlosen Verteilung von Applikationen

## Hardware-Anforderungen

Merkmal	CLDC 1.0	CLDC 1.1	MIDP 1.0	MIDP 2.0
Prozessor	16 oder 32 Bit			
Speicher:				
-ROM	128 kB	160 kB	128 kB	256 kB
-RAM	32 kB	32 kB	32 kB	128 kB
-Persistent	-	-	8 kB	8 kB
Display				
-Auflösung	-	-	96 x 54 Pixel	96 x 54 Pixel
-Farbtiefe	-	-	1 Bit	1 Bit
Eigabemöglichkeit	-	-	Tastatur oder Touchscreen	
Netzwerkanbindung	Bidirektionale, drahtlose, unstabile, langsame Verbindung			
Multimedia	-	-	Möglichkeit einer Soundausgabe	

Quellen: [cld03, mid02, Sch07]

Tabelle 1: Hardware-Anforderungen von CLDC und MIDP

## 2.3 Netzanbindungen

### 2.3.1 GSM

Das Global System for Mobile Communications (GSM) ist das typische Netz für mobile Telefonie. Außerdem kann dieses für Datenübertragungen genutzt werden, wobei nur 9,6 kbit/s erreicht werden. Durch Einsatz von High Speed Circuit Switched Data (HSCSD) konnte die Datenrate auf 14,4 kbit/s gesteigert werden und durch General Packet Radio Service (GPRS) sogar auf 56 kbit/s. Mit Enhanced Data Rates for GSM Evolution (EDGE) wurden die Möglichkeiten von GSM ausgeschöpft und 384 kbit/s erreicht.

### 2.3.2 UMTS

Universal Mobile Telecommunications System (UMTS) ist als Nachfolger von GSM gedacht. Datenraten von 384 kbit/s und bis zu 7,2 Mbit/s in Verbindung mit High Speed Downlink Packet Access (HSDPA) sind derzeit möglich.

### **2.3.3 Bluetooth**

Bluetooth dient der drahtlosen Datenübertragung zwischen Mobiltelefonen, Smartphones, Computern und Zubehör wie zum Beispiel Headsets und bietet Geschwindigkeiten von rund 2,1 MBit/s. Durch eine geringe Reichweite von nur 10 Metern (Bluetooth-Klasse 3) bis zu 100 Metern (Bluetooth-Klasse 1) ist diese Technik nur zum Datenaustausch benachbarter Geräte und zum Anschluss von Zubehör interessant.

### **2.3.4 WLAN**

Wireless Local Area Network (WLAN) bietet durch seine geringe Reichweite von bis zu 300 Metern im Freien und 20-30 Metern in Gebäuden eine große Einschränkung für mobile Endgeräte und ist daher aus wirtschaftlicher Sicht nicht überall einsetzbar. Ein Vorteil ist dagegen die hohe Geschwindigkeit, denn WLAN erreicht mit dem Standard IEEE 802.11n bis zu 300 Mbit/s, bisher sind bei den meisten Endgeräten jedoch noch 54 Mbit/s (IEEE 802.11g) üblich.

Das „Ausleuchten“ eines Fabrikgeländes ist mit WLAN und Bluetooth durch den Einsatz vieler Access Points weitestgehend möglich. Eine Roaming-Funktionalität ermöglicht das automatische Wechseln zwischen Access Points.

### 3 Geschäftsprozesse

Geschäftsprozesse, auch Prozesse genannt, haben das Ziel einen betrieblichen Ablauf ergebnisorientiert zu beschreiben. Dazu wird der Ablauf eines Prozesses in Teilschritte unterteilt und der Ablauf festgelegt.

Ein Handwerksbetrieb hat bei Aufträgen einen Ablaufplan, bei dem Kunden eine Anfrage haben und auf diese ein Angebot erhalten. Bei einer Zusage werden die Arbeiten durchgeführt und eine Rechnung geschrieben, die der Kunde zu bezahlen hat.



Abbildung 9: Beispiel eines groben Geschäftsprozesses

Die einzelnen Schritte des Prozesses können verfeinert werden. Eine Kundenanfrage kann auf verschiedenen Wegen stattfinden, beispielsweise über ein Telefonat oder einen persönlichen Besuch des Kunden. Eine Verzweigung entsteht, wenn ein Angebot erstellt wird, der Kunde aber das Angebot nicht wahrnimmt. Gleichzeitig kann es zu einer Schleife kommen, in der mit dem Kunden über den Preis verhandelt wird. Parallelität kann entstehen, wenn mehrere Handwerker die Arbeiten zeitgleich ausführen. Meist werden betriebliche Abläufe in Unternehmen standardisiert, aber nicht dokumentiert. Auch werden die Abläufe nicht immer als Geschäftsprozess erkannt.

In der Literatur stößt man oft auf den Begriff Workflow, der lange Zeit für den Begriff Geschäftsprozess verwendet wurde. In aktueller Literatur ist zu beobachten, dass Workflow durch den Begriff Business Process ersetzt wird. Da Geschäftsprozesse koordiniert werden müssen, werden sie in [BL03, S. 53] über die folgenden Interdependenzen dargestellt.

**Organisationseinheiten** Eine Organisationseinheit kann ein Unternehmen darstellen und somit eine Struktur bilden. Wenn man einen Teil eines Un-

ternehmens als Organisationseinheit darstellt, so spricht man von einer Funktion. Sie dienen der Arbeitsteilung und der Koordination.

**Aktivitäten** Aktivitäten sind einem Prozess zugeordnete Schritte. Diese werden von Menschen oder informationstechnisch bearbeitet.

**Prozesse** Ein Prozess bildet einen Ablauf, aufbauend auf den Aktivitäten und den Organisationseinheiten.

**Informationen** Informationen spielen in Geschäftsprozessen eine wichtige Rolle, denn diese werden in einem Prozess gesammelt, verarbeitet und ausgegeben. Sie werden in Aktivitäten verarbeitet und können den Weg beeinflussen, den ein Prozess läuft.

### 3.1 Business Process Management

Für Business Process Management (BPM) gibt es laut Freund und Götzer [FG08, S. 3] keine allgemein gültige Definition, jedoch soll BPM laut dem Artikel [Fre08] „für eine Reihe von Methoden, die der strategischen Ausrichtung sowie der Entwicklung und Verbesserung von Geschäftsprozessen dienen“. Dies muss nicht mit Hilfe von EDV-Systemen realisiert werden, sondern stellt einen Weg dar mit den Informationen umzugehen, die in einem Prozess entstehen. Im BPM ist von der Planung und Modellierung bis zur Überwachung und Protokollierung alles inbegriffen.

Durch eine vollständige Modellierung von Geschäftsprozessen wird das Optimierungspotential erstmalig ersichtlich. Auch können durch die automatisierte Verwaltung offene Angebote überwacht werden (Reporting) und bei offenen Angeboten ohne Kundenrückruf Anfragen an den Kunden gerichtet werden (Monitoring).

### 3.2 Business Process Management-Systeme

Business Process Management-Systeme (BPMS) sind laut Strohmeier [Str08, S. 305] „Anwendungen zur Gestaltung, Ausführung und Steuerung von Geschäftsprozessen über verschiedene Anwendungssysteme und Anwender hinweg“. Somit stellen sie die informationstechnische Realisierung von BPM dar.

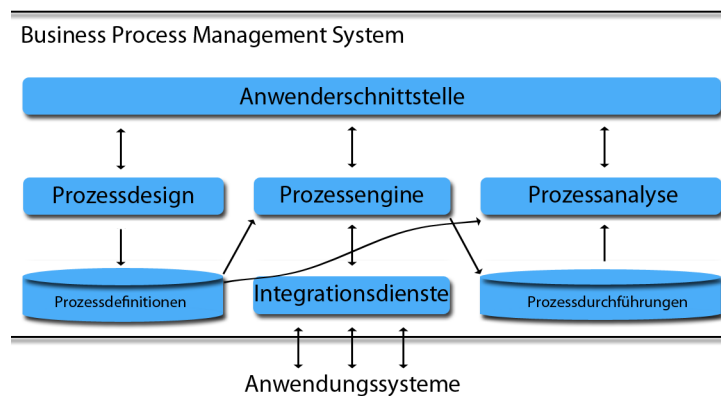


Abbildung 10: Aufbau eines Business Process Management-Systems [BL03]

Der Ablauf eines BPMS erfolgt schrittweise. Dazu wird erstmal eine Prozessdefinition erstellt. Die Prozessdefinition wird in einer Datenbank gesichert und der Prozessengine übergeben. Die Prozessengine stellt die Prozesse über Integrationsdienste Anwendungssystemen zur Verfügung. Die Sicherung der Prozessinstanzen in der Prozessengine erfolgt in der Datenbank zu Prozessdurchführungen. Prozessinstanzen repräsentieren den Zustand eines einzelnen Vorgangs, wozu die aktiven Aktivitäten, die dazugehörigen Akteure (im Falle von Human Workflow Management) und die Informationen gehören. Zum Reporting/Monitoring wird die Prozessanalyse verwendet, die auf die Datenbank zugreift, um eine Analyse zu erstellen und diese über die Anwenderschnittstelle zur Verfügung stellt.

Durch zwei mögliche Schnittstellen ist eine Anbindung von Anwendungen und von Anwendern realisierbar. Daher werden BPM-Systeme in zwei Bereiche aufgeteilt.

### 3.2.1 Human Workflow Management

Bei einer Beteiligung durch Menschen in einem Geschäftsprozess wird von Human Workflow Management gesprochen. Über die Anwenderschnittstelle wird eine Kommunikation zum Prozessdesign, zur Prozessengine und zur Prozessanalyse ermöglicht.

Laut Freund [FG08] kann die Interaktion der Anwender durch vier Arten erfolgen

**Eigenständige Desktop-Anwendung** Die Interaktion findet in Anwendungen statt, die den Geschäftsprozess steuern können.

**Integration in vorhandene Collaboration Clients** Dazu zählen in Unternehmen eingesetzte Anwendungen wie zum Beispiel E-Mail.

**Webanwendungen** Anwender nehmen über eine Webanwendung am Geschäftsprozess teil.

The screenshot displays the JBoss jBPM Administration Console interface. At the top right, it shows 'Logged in as: manager' with a 'Log Out' link. The main navigation bar includes 'Manage: Processes', 'Personal/Group Tasks', 'All Tasks', 'Jobs', and 'Identities'. The central content area is divided into two main sections: 'Task Summary' and 'Zeiterfassung' (Time Recording).

**Task Summary:**

- Task Link: [ID\\_102](#)
- Name: Erfassung freischalten
- Status: **Not Started**
- Assigned To: manager
- Token: [ID\\_1192 "Erfassung beginnen"](#)
- Process Instance: [ID\\_40](#)
- Process: [handwerk2 v1](#)
- Created Date: Dec 1, 2009 6:02:09 PM

**Views:**

- [Task Form](#)
- [Comments](#)
- [Variables](#)
- [Transitions](#)

**Actions:**

- [Suspend](#) this task
- [Start](#) this task
- Reassign this task to:
  - 
  -

**Zeiterfassung (Time Recording):**

- Auswahl der Datenerfassung: Material (dropdown)
- Auswahl des Users: user-mobil1 (dropdown)
- Ende: Nein (dropdown)
- weiter:

The footer of the console reads 'JBoss jBPM Administration Console'.

Abbildung 11: jBPM Webconsole

**Mobile Clients** Laut Freund finden mobile Clients „in der Praxis bislang keine breite Verwendung“.

### 3.2.2 Service-Orchestration-Workflows

Zur Integration von informationstechnischen Systemen in Geschäftsprozessen werden Service-Orchestration-Workflows verwendet. Über die Integrationsdienste wird eine Kommunikation zur Prozessengine ermöglicht.

## 3.3 Business Process Modeling Notation

Die Business Process Modeling Notation (BPMN) dient der visuellen Darstellung für Geschäftsprozesse und erfordert keine IT-Kenntnisse, wodurch Personal aus leitenden Ebenen in die Modellierung mit einbezogen werden kann. Eine automatische Umwandlung zu einer beschreibenden Sprache mit der eine Prozessengine umgehen kann, ist aus heutiger Sicht nicht möglich und erfordert eine manuelle Bearbeitung durch IT-Personal.

In dem folgenden Beispiel ist ein Investitionsantrag zu sehen, der in die drei Benutzer/Benutzergruppen Antragsteller, Prüfer, Geschäftsführung aufgeteilt ist. Durch ein Ereigniss, das durch einen grünen Kreis dargestellt wird, startet der Antragsteller einen Investitionsantrag. Dieser wird vom Prüfer bearbeitet und wird im gelben Entscheidungspunkt einer Prüfung unterzogen, aus der hervorgeht, welchen Weg der Antrag läuft. Denkbar wäre hier eine Prüfung der Investitionskosten, um entscheiden zu können, ob der Investitionsantrag durch die Geschäftsführung geprüft werden muss. Der rote Kreis symbolisiert das Ende dieses Prozesses.



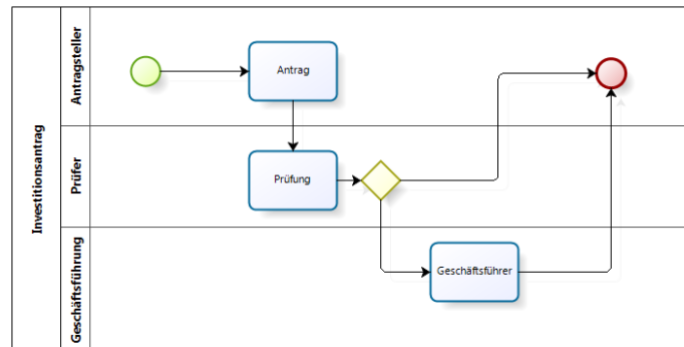


Abbildung 12: Darstellung der Business Process Modeling Notation

### 3.4 Business Process Execution Language

Business Process Execution Language (BPEL) ist eine ausführbare Beschreibungssprache für Geschäftsprozesse. Sie basiert auf XML und dient nicht der visuellen Darstellung des Modells, weshalb keine standardisierte Darstellungsform für BPEL existiert. Anhand eines vereinfachten Beispiels soll der Aufbau von BPEL erläutert werden.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <process name="Investitionsantrag">
4    <partnerLinks>
5      <partnerLink name="Bestellung"
6        partnerLinkType="qname" myrole="bestellService">
7      </partnerLink>
8    </partnerLinks>
9
10   <receive createInstance="yes" />
11
12   <sequence>
13     ...
  
```

```

14      <switch>
15          <case condition="...">
16              ...
17          </case>
18          <otherwise>
19              ...
20          </otherwise>
21      </switch>
22  </sequence>
23 </process>

```

In Zeile 3 beginnt der Process, der als optionale Attribute Namespaces zulässt und in Zeile 23 geschlossen wird. In den „partnerLinks“ (Zeile 4-8) werden Webservice-Schnittstellen definiert, die im weiteren Verlauf verwendet werden können. Der Anfang eines Prozesses wird durch das „receive“ (Zeile 10) dargestellt, der durch weitere Attribute erweitert werden kann und in diesem Fall den Start einer Prozessinstanz auslöst. In der „sequence“ (Zeile 12-22) werden Aktivitäten definiert, um diese der Reihenfolge nach ablaufen zu lassen. Durch ein „switch“ (Zeile 14-21) wird der Weg getrennt und definiert bei welchen Bedingungen ein Weg eingeschlagen werden kann. Die Entscheidung wird durch „case“ mit „condition“-Parameter (Zeile 15) definiert.

### 3.5 jBPM Process Definition Language

jBPM Process Definition Language (jPDL) ist eine ausführbare Beschreibungssprache für Geschäftsprozesse. Sie basiert auf XML und wird von der Business Process Engine jBPM interpretiert. Anhand eines einfachen Beispiels soll der Aufbau eines Prozesses verdeutlicht werden. Darin wird im ersten Schritt ein Investitionsantrag gestellt und dieser abhängig von Wert \$Betrag sofort genehmigt oder ab einem Wert von 1000 weitergeleitet an die Geschäftsführung. Nach der Bearbeitung durch den Geschäftsführer wird der Prozess beendet.

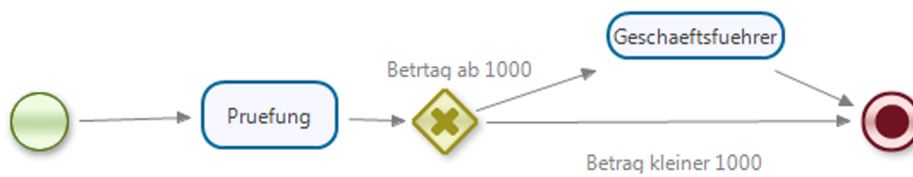


Abbildung 13: Darstellung von jPDL (Version 4.0)

Der zum Schaubild gehörende Code sieht folgendermaßen aus.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <process name="Investitionsantrag"
4   xmlns="http://jbpm.org/4.0/jpdl">
5   <start name="Antrag">
6     <transition to="Pruefung"/>
7   </start>
8   <task name="Pruefung">
9     <transition to="Betragspruefung"/>
10  </task>
11  <decision name="Betragspruefung">
12    <transition name="Betrtag ab 1000"
13      to="Geschaeftsfuehrer">
14      <condition expr=">1000"/>
15    </transition>
16    <transition name="Betrag kleiner 1000" to="Ende"/>
17  </decision>
18  <task assignee="gf_user" name="Geschaeftsfuehrer">
19    <transition to="Ende"/>
20  </task>
21  <end name="Ende"/>
22 </process>

```

In Zeile 3 beginnt der Process und wird in Zeile 20 geschlossen. Jeder Process muss mit einem „start“ (Zeile 4) beginnen und mit einem „end“ (Zeile 21)

beendet werden. Zeile 6 beinhaltet einen „task“, der eine Aktion durch einen Menschen vorsieht. Mit einem „decision“ (Zeile 11-17) wird der Weg getrennt und definiert bei welchen Bedingungen ein Weg eingeschlagen werden kann, indem mit einem „condition“ (Zeile 14) wie in diesem Fall die Variable „Betrag“ größer als 1000 sein muss, um den Weg über die Geschäftsführung zu gehen. Mit Hilfe eines „assignee“-Parameters (Zeile 18) kann ein „task“ einem Benutzer zugeordnet werden.

### 3.6 Business Process Engines

Eine Business Process Engine bildet den Kern eines BPMS. Sie nutzt modellierte Prozesse und bietet über Schnittstellen eine Anbindungsmöglichkeit für Human Workflow Management und Service-Orchestration-Workflows. Durch eine Auslegung auf eine Erweiterbarkeit auf beliebige Beschreibungssprachen kann eine höhere Flexibilität gewährleistet werden.

	jBPM	Sliver	Apache ODE
Beschreibungssprachen	jPDL	BPEL	BPEL
Auslegung auf Erweiterbarkeit	ja	nein	nein
Human Workflow Management	ja	nein	nein
Service-Orchestration-Workflows	ja	ja	ja
Mobilgerät-Kompatibel	nein	ja	nein
Programm-Plattform	JDK 5.0	J2ME MIDP 2.0	JDK 5.0
Open Source	ja	ja	ja
Lizenz	LGPL	LGPL	Apache Licence 2.0

Quellen: [jboss.org/jbossjbpn](http://jboss.org/jbossjbpn), [mobilab.cse.wustl.edu/projects/sliver/](http://mobilab.cse.wustl.edu/projects/sliver/) und [ode.apache.org](http://ode.apache.org)

Tabelle 2: Übersicht verschiedener Prozessengines

Bei Sliver ist anzumerken, dass seit 2006 keine Weiterentwicklung stattfand und eine Dokumentation zum Projekt nicht vorhanden ist.

### 3.7 Geschäftsprozess-Modellierung

Geschäftsprozesse werden bei der PI-Data bereits erarbeitet und grafisch dargestellt, um diese dem Kunden vorstellen zu können. Eine Transformation wird anhand der erarbeiteten Grafiken manuell vorgenommen und in einer eigenen GUI-Beschreibungssprache und Services beschrieben. Die Beschreibungen und Services werden auf dem Server für Mobilgeräte bereitgestellt. Das Ziel besteht daher darin, den Weg der manuellen Transformation zu automatisieren oder zu erleichtern.

Ein möglicher Weg ist, BPMN zu verwenden. Stand heute ist BPMN nicht ausführbar und daher eine Umwandlung in eine Beschreibungssprache notwendig. Geschäftsprozesse werden in BPMN grafisch modelliert und manuell oder durch Unterstützung von Software nach BPEL umgewandelt. Nach Schumm [Sch08] gibt es mehrere Werkzeuge, um Geschäftsprozesse mit BPMN zu modellieren und diese in BPEL zu übersetzen. Verfügbare Softwarelösungen zur Modellierung von BPEL wurden in der Diplomarbeit von Kujath [Kuj06] untersucht. Schumm [Sch08] hat mehrere Unterschiede von BPMN und BPEL aufgelistet, die eine Transformation erschweren. Durch den bereits bei der PI-Data verwendeten Enterprise Architect ist eine solche Softwarelösung bereits vorhanden.

Einen anderen Weg kann man gehen, wenn man auf jPDL setzt, da dieses eine grafische Notation hat und in XML beschrieben wird, wodurch es auf Business Process Engines ausgeführt werden kann. Eine Transformation entfällt somit und durch den Graphical Process Designer existiert ein Open Source Modellierungstool für jPDL. Der Nachteil gegenüber dem ersten Weg sind die grafischen Einschränkungen von jPDL, da dieses eine geringere Anzahl an grafischen Notationen verwendet und auf grafische Unterteilung von Benutzer und Benutzergruppen verzichtet. Benutzer und Benutzergruppen sind nur als Eigenschaften der Prozessschritte und in XML ersichtlich.

Über das sich in der Entstehung befindende BPMN 2.0 ergibt sich ein sehr interessanter Weg, da dieser die Vorteile beider Systeme koppelt. Laut der Beschreibung von Allweyer [All] wird die grafischen Notation aus BPMN

nur geringfügig erweitert und durch ein standardisiertes Austauschformat in XML ergänzt, das von BPEL unabhängig ist und eine Transformation unnötig macht. Aufgrund der noch bevorstehenden Entwicklungszeit, die BPMN 2.0 vor sich hat, ist eine Implementierung derzeit nur beschränkt sinnvoll.

PI-Data will sich in der Zukunft dadurch auszeichnen, dass Kunden durch ihre eigene IT-Abteilung Geschäftsprozesse ändern können. In diesem Fall stellt BPMN eine Hürde dar, wenn manuell nach BPEL transformiert werden muss. jPDL nimmt diese Hürde und schränkt im Gegenzug bei der grafischen Notation ein. Bei BPMN 2.0 fallen die Nachteile beider Lösungen weg.

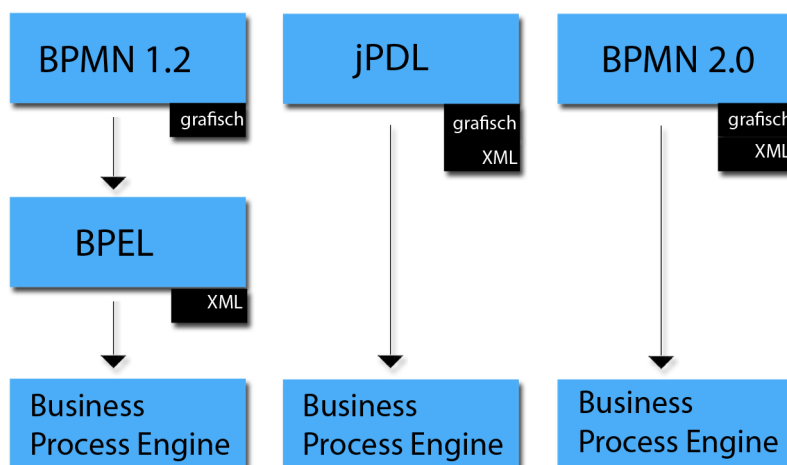


Abbildung 14: Möglichkeiten für die PI-Data

### 3.8 Beispiel für einen Geschäftsprozess

Zur einfachen Veranschaulichung wird in Kapitel 4 das erarbeitete Konzept mit einem Handwerksprozess geprüft.

### 3.8.1 Nodes in jPDL

Zum Verständnis der grafischen Notation von jPDL in der Version 3.2.6 werden einige der grundlegenden Nodes erklärt. Nodes sind einzelne Schritte, die im Prozess durchlaufen werden und Benutzereingaben erfordern können oder vordefinierte Aktionen selbständig durchlaufen.

Um in einem Geschäftsprozess vorzugeben, an welchem Punkt gestartet werden muss, wird ein „Start State“ verwendet. Durch die Möglichkeit ein Formular an diesem Node anzuhängen hat er die selben Eigenschaften wie ein „Task Form“.

Ein „Task Form“ ist ein Node, in dem durch ein Formular vom Benutzer Informationen gesammelt und ausgegeben werden können. Somit bildet dieser Node die Schnittstelle zu Menschen, die am Prozess beteiligt sind. Informationen werden in Variablen gespeichert, die im fortlaufenden Prozess weiter verwendet werden und ggf. überschrieben werden können.

Meist anhand von Variablenwerten oder anderweitigen Informationen wird im „Decision“ über vordefinierte Regeln entschieden, welchen Weg der Prozess läuft. Es wird nur ein Weg eingeschlagen. Optional kann ein Weg ohne Bedingung definiert werden, der durchlaufen wird, wenn alle anderen Bedingungen nicht zutreffen sollten.

Ein Node, der Parallelität ermöglicht, ist der „Fork“. Ein Aufruf des Forks löst im Prozess mehrere Wege aus, die zur selben Zeit abgearbeitet werden können. Die „Fork“s können beliebig komplex werden, da jBPM eine beliebige Anzahl an aktiven Nodes handhaben kann. Durch Parallelität kann die Durchlaufzeit eines Prozesses optimiert werden.

In einem „Join“ laufen „Fork“s zusammen. Dadurch wird sichergestellt, dass alle Wege beendet sind, bevor der Prozess weiterläuft.

Weitere Nodes sind vorhanden, werden hier jedoch nicht aufgeführt, da sie im Beispielprozess nicht verwendet werden.

### **3.8.2 Handwerksprozess**

Der Handwerksprozess bildet den Ablauf in einem Kundenauftrag wieder. Im ersten Schritt, dem „Auftrags-Start“ werden im Office Auftragsdaten gesammelt und festgelegt, ob eine Anzahlung vereinbart wurde. Wurde diese vereinbart, so wird der Prozess erst beim Zahlungseingang fortgesetzt. Anschließend erfolgt eine Terminvereinbarung. Nach der Terminvereinbarung findet eine Zuweisung der Mitarbeiter zum Auftrag statt, die vor Ort beim Kunden sein werden. Bei der Mitarbeitererfassung werden den Mitarbeitern die Zeit- und Materialerfassung freigeschaltet. Der weitere Verlauf findet auf den Mobilgeräten der Mitarbeiter statt. Mitarbeiter haben die Möglichkeit, ihre Zeiten und verwendete Materialien einzugeben. Alle Mitarbeiter müssen ihre Zeit- und Materialerfassung beendet haben, damit eine Rechnung gedruckt werden kann.



### 3.8.3 Grafisch dargestellter Handwerksprozess

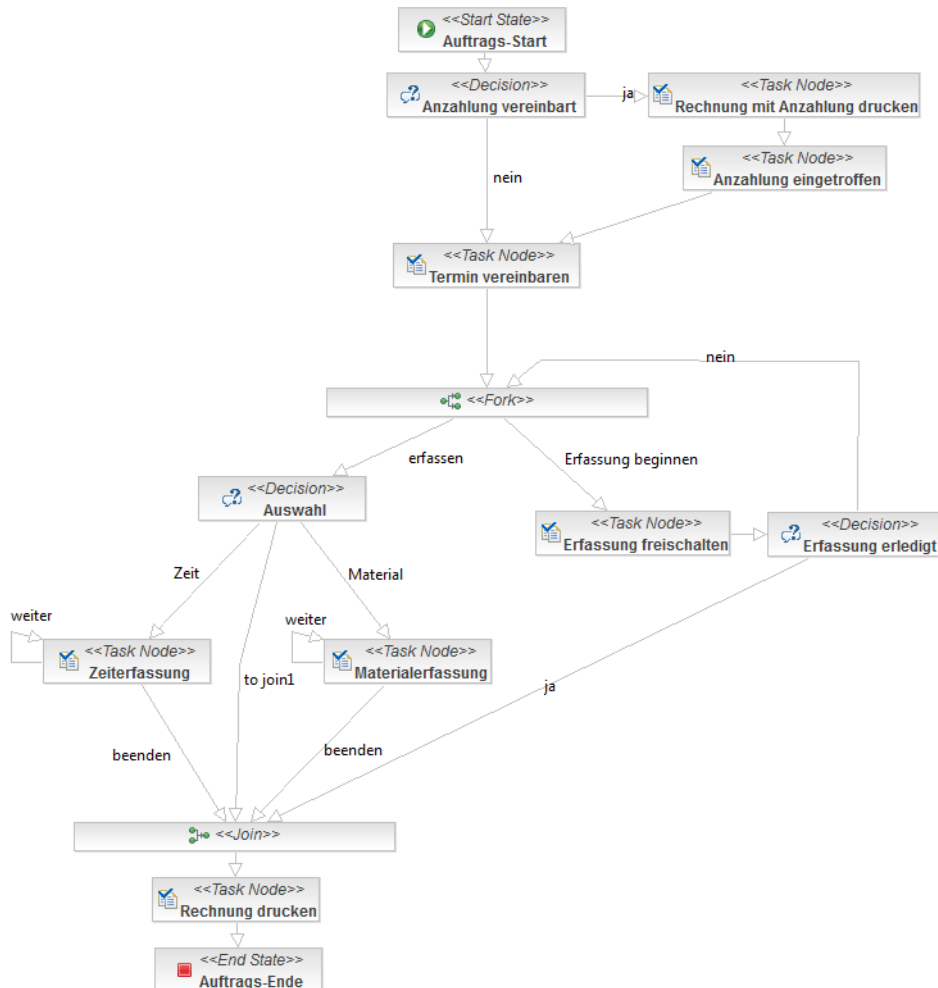


Abbildung 15: Handwerksprozess in jPDL

Im ersten Schritt, dem „Auftrags-Start“, wird die Prozessinstanz gestartet und es werden von einem Mitarbeiter anhand eines Formulars alle Auftragsdaten erfasst, wozu auch eine eventuell vereinbarte Anzahlung gehört. Im nächsten Schritt wird in einem „Decision“ anhand einer Variable für die Anzahlung bestimmt, ob diese vereinbart wurde. Bei der Auswahl einer Anzahlung müssen zwei weitere Formulare zum Abwickeln der Anzahlung durchlaufen werden. Wenn keine Anzahlung vereinbart wurde, werden diese

Schritte übersprungen und direkt eine Terminvereinbarung gestartet.

Nach einer Terminvereinbarung kommt es zu einem „Fork“, der den Ablauf in zwei Wege aufteilt. Der erste Weg führt zur Zeit- und Materialerfassung und führt in diesem Fall direkt ins „Join“, da noch keine Mitarbeitererfassung durchgeführt wurde. Der zweite Weg durchläuft die Mitarbeitererfassung, die im Office stattfindet. Es wird hier eine beliebige Anzahl an Mitarbeitern dem Auftrag hinzugefügt, indem nach jedem Eintrag eines Mitarbeiters mit- samt seiner Berechtigungen zur Zeit- und Materialerfassung das Join erneut aufgerufen wird, bis die Mitarbeitererfassung endgültig abgeschlossen wird.

Dadurch werden für die mobilen Mitarbeiter die entsprechenden Aktivitäten gestartet. Diese können mobil beliebig oft aufgerufen werden und enden erst mit der expliziten Bestätigung des Mitarbeiters auf seinem Mobilgerät. Dies ist wichtig, damit Mitarbeiter nicht erst zum Ende des Auftrags ihre Zeiten und verbrauchten Materialien eintragen und so ggf. etwas vergessen, wenn der Auftrag eine Woche gedauert haben sollte.

Die Rechnung kann erst gedruckt werden, sobald alle Aktivitäten im „Join“ angekommen sind, d.h. die Mitarbeitererfassung muss abgeschlossen sein und die Zeit- und Materialerfassung aller Mitarbeiter muss beendet worden sein. Sobald dies erfolgt ist, kann die Rechnung im Office gedruckt und der Auftrag abgeschlossen werden.

### **3.9 Backends in Geschäftsprozessen**

Backends spielen in Geschäftsprozessen eine wichtige Rolle. Im einfachsten Fall ist das Backend eine Datenbank auf einem Server. Es kann aber auch ein beliebiges System zur Datenverarbeitung oder Verwaltung sein. In der Realität greifen Geschäftsprozesse auf verteilte Backends zu. Um die unterschiedlichen Möglichkeiten aufzuzeigen, wird der Handwerksprozess aus dem Kapitel 3.8 verwendet.

Beim Erstellen einer Auftrages wird geprüft, ob der Kunde bereits im CRM-System angelegt worden ist. Bevor sich die Handwerker auf den Weg zum

Kunden machen, holen sie sich die Adresse des Kunden aus dem CRM-System. Beim Kunden angekommen können die Handwerker die verwendeten Materialien erfassen, indem sie Materialien aus dem Materialienbestand des SAP-Systems auswählen. Das SAP-System kann automatisch Materialien bei den Lieferanten nachbestellen, wenn diese zur Neige gehen. Der Aufbau der IT-Landschaft ist in Abbildung 16 beispielhaft dargestellt.

In der Realität sind mehr Backends in der IT-Landschaft häufig.

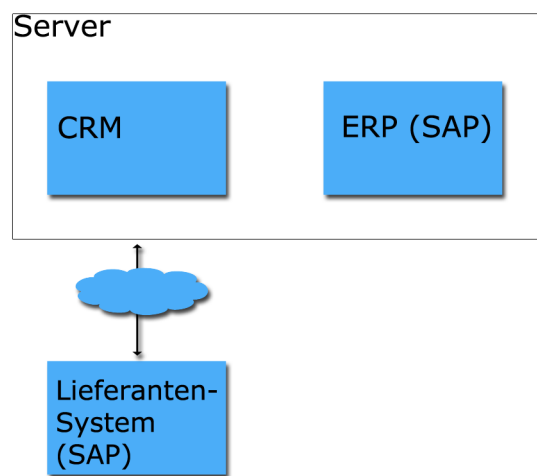


Abbildung 16: Beispielhafter Aufbau einer IT-Landschaft

Das Beispiel soll verdeutlichen, dass ein Geschäftsprozess mit einer beliebig komplexen IT-Landschaft aufgebaut werden kann und Backends eine zentrale Rolle in Geschäftsprozessen spielen.

## 4 Konzeption

### 4.1 Requirements

Vor dem Aufstellen eines Konzepts muss die Aufgabenstellung klar spezifiziert werden. Dazu werden Requirements aufgestellt, die eine durchgehende Kontrolle im weiteren Verlauf ermöglichen. Es erfolgt eine Trennung in funktionale und nicht funktionale Requirements. Funktionale Requirements beschreiben Funktionen, über die das System verfügen muss. Nicht funktionale Requirements sind technische Anforderungen, die in der Implementierung eine Rolle spielen.

#### 4.1.1 Funktionale Requirements

**Requirement 100 - Lauffähig auf Mobilgeräten** Geschäftsprozesse müssen auf Mobilgeräten lauffähig sein.

**Requirement 110 - Mobile Clients** sind kleine Mobilgeräte wie Mobiltelefone bis hin zu Laptops, auf die im Kapitel 2 eingegangen wurde. Alle Geräteklassen sind zu unterstützen.

**Requirement 200 - Offline-Fähigkeit** In den meisten Mobilszenarien kann nicht von einer einwandfrei funktionierenden Netzverbindung ausgegangen werden. Aus Sicht des Anwenders funktioniert die Software nicht korrekt, wenn sie aufgrund einer fehlenden Netzverbindung nicht reagieren kann. Dies kann die Akzeptanz bei Anwendern für Mobilgeräte negativ beeinflussen und ist daher durch eine Offline-Fähigkeit zu verhindern.

#### **Requirement 300 - Zentrale Ablage für Geschäftsprozesse**

Geschäftsprozesse liegen zentral im Business Process Management-System. Wenn eine Prozessinstanz gestartet wird, wird die aktuelle Version des Geschäftsprozesses vom Business Process Management-System verwendet. Temporäre Kopien der Geschäftsprozesse sind auf dem Mobilgerät zulässig.

### **Requirement 310 - Aufteilung der Geschäftsprozesse**

Geschäftsprozesse bestehen aus einer Prozess- und einer Oberflächenbeschreibung und werden getrennt betrachtet.

### **Requirement 400 - Zentrale Verwaltung von Prozessinstanzen**

Prozessinstanzen werden zentral im Business Process Management-System verwaltet und dienen als Anlaufstelle für Clients.

**Requirement 500 - Beschränkung auf Anwender** Geschäftsprozesse müssen nicht zusätzlich auf Anwender beschränkt werden. Es ist Aufgabe der Geschäftsprozessbeschreibung zu definieren, welche Anwender den Geschäftsprozess starten dürfen.

## **4.1.2 Nicht funktionale Requirements**

**Requirement 600 - Plattform** Es gibt auf dem Markt wie bereits im Kapitel 2 beschrieben eine große Anzahl an Mobilgeräten. Das Entwickeln einer Software für alle Geräte erweist sich durch unterschiedliche Plattformen als sehr schwierig. Daher ist eine gemeinsame Plattform notwendig, die von den meisten Gerätetypen unterstützt wird. Mit Java ist eine solche Plattform vorhanden. Es wird eine Kompatibilität mit MIDP, Personal Profile und Java SE 6 erwartet, um alle Java-fähigen Geräte zu unterstützen.

**Requirement 610 - Grafische Oberfläche** Die grafische Oberfläche muss auf allen Mobilgeräten dargestellt werden können. Dies ist anhand von Oberflächenbeschreibungen für verschiedene Mobilgeräte zu ermöglichen. Der Inhalt der Oberfläche darf nicht beschnitten werden.

**Requirement 700 - Parallelität** Geschäftsprozesse können durch parallel ablaufende Aktivitäten einen Zeitgewinn aufweisen. Dieser Vorteil darf durch die Einbindung offline-fähiger Mobilgeräte nicht verloren werden. Daher ist es mehreren Mitarbeitern zu ermöglichen, zeitgleich an einer Prozessinstanz arbeiten zu können.

### **Requirement 710 - Konkurrierender Zugriff auf Prozessinstanzen**

Ein Anwender kann einem Prozessschritt zugeordnet sein und erhält in dem Fall vom Server eine Kopie der Prozessinstanz. Offline getätigte Änderungen an der Prozessinstanz können auf dem Server nur dann garantiert fehlerfrei ausgeführt werden, wenn sich in der Zwischenzeit nichts an der Prozessinstanz geändert hat. Daher muss der Fehlerfall abgefangen werden.

**Requirement 720 - Mobile Prozessinstanz** Ein Geschäftsprozess darf im mobilen Teil keine Effekte hervorrufen, die durch einen komplett online ausgeführten Geschäftsprozess nicht aufgetreten wären.

**Requirement 800 - Ressourcenbedarf** Im mobilen Bereich sind Hardwareressourcen begrenzt. Daher ist mit CPU, Memory und Storage sparsam umzugehen.

**Requirement 900 - Backend-Zugriff** Geschäftsprozesse greifen auf das Backend zu. Diese Möglichkeit darf im mobilen Bereich nicht eingeschränkt werden.

Aus Sicht des Backends darf es keine Rolle spielen, ob ein Mobilgerät oder der Server auf das Backend zugreift.

## **4.2 Aufbau des Servers**

Der Aufbau des Servers ist in den folgenden Konzepten identisch und wird hier grundlegend erklärt.

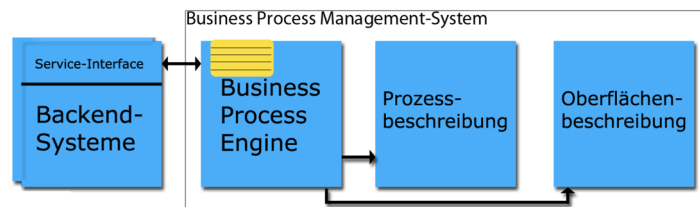


Abbildung 17: Aufbau des Servers

Der Server besteht aus vier Teilen und wird in Abbildung 17 aufgezeigt. Zum einem beinhaltet er eine Business Process Engine, die Teil eines Business Process Management-Systems ist und somit eine Prozessanalyse ermöglicht und der Prozessdurchführung dient. Geschäftsprozesse werden entwickelt, durch das Business Process Management-System entgegengenommen und von der Business Process Engine interpretiert. Durch die zentrale Sicherung der Geschäftsprozesse auf dem Server wird Requirement 300 erfüllt.

Für die Betrachtung der Konzepte werden die Prozess- und Oberflächenbeschreibung als eigenständig betrachtet (Requirement 310).

Geschäftsprozesse greifen über die Business Process Engine auf Backend-Systeme zu. Zwischen Backend und Business Process Engine liegt ein Service-Interface, dass Zugriffe auf das Backend vereinfacht.

### 4.3 Ohne Business Process Engine auf dem Mobilgerät

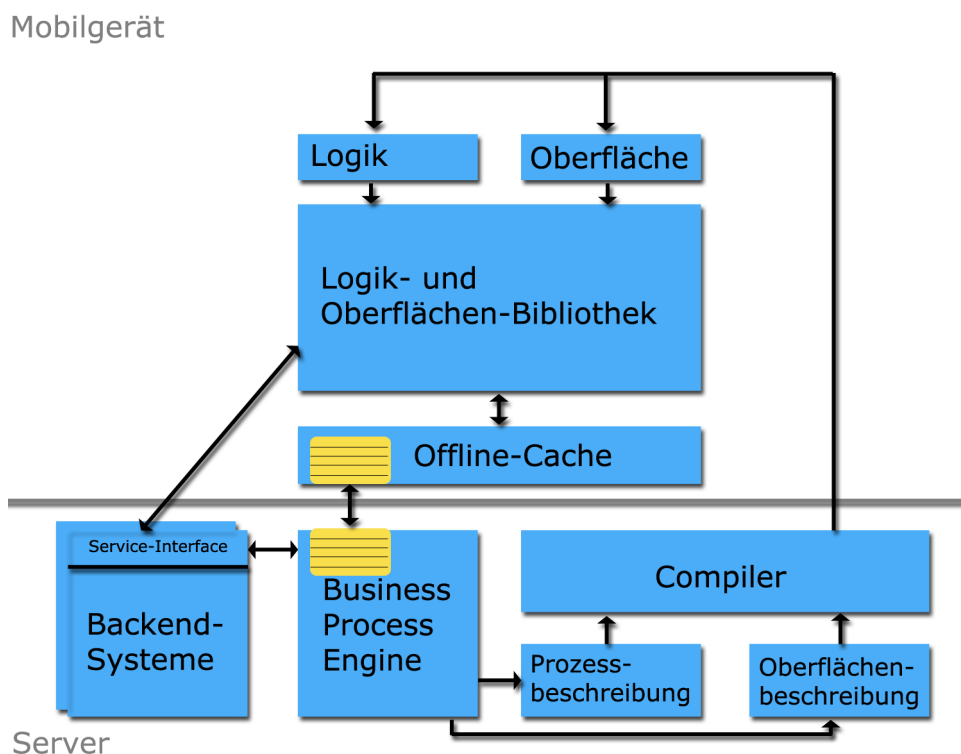


Abbildung 18: Konzept ohne Business Process Engine auf dem Mobilgerät

Die Geschäftsprozesse werden auf dem Server ausgeführt und verbleiben in ihrem Status, sobald mobile Prozessschritte ausgeführt werden müssen. Dem Mobilgerät werden Prozessinstanz und, wenn nicht bereits vorhanden, auch Prozess- und Oberflächenbeschreibung übertragen. Darauf wird auf eine Interaktion des Mobilgerätes in der Prozessinstanz gewartet.

Der Server wird um einen Compiler erweitert. Auf dem Compiler werden Prozess- und Oberflächenbeschreibung vom Server zusammengetragen und in eine für das Mobilgerät ausführbare Ausführungs- und Oberflächenbeschreibung kompiliert. Der Compiler kann zu einer einfachen Implementierung darauf ausgelegt werden, dass er nur einen Prozessschritt offline ausführbar macht und danach eine Netzverbindung verlangt wird. Für eine vollständige Implementierung und eine über mehrere Prozessschritte ermöglichte Offline-Fähigkeit muss es dem Compiler möglich werden einen kompletten Geschäftsprozess interpretieren zu können.

Auf dem Mobilgerät findet sich ein anderer Aufbau, als auf dem Server. Das Starten neuer Prozessinstanzen und Verwalten aller auf dem Mobilgerät vorhandenen Prozessinstanzen wird in einem Business Process Management System in einer Business Process Engine durchgeführt. Diese Aufgabe wird auf dem Mobilgerät vom Offline-Cache übernommen. Der Offline-Cache beinhaltet alle Prozessinstanzen, die dem Benutzer des Mobilgerätes zugeordnet sind.

Um mobile Interaktionen auf dem Server zu aktualisieren, protokolliert der Offline-Cache alle Aktivitäten des Anwenders im Geschäftsprozess und lässt bei einer Netzverbindung alle Aktivitäten auf dem Server wiederholen. Die Reihenfolge stellt er sicher, indem eine Transaktionsverbindung zwischen Offline-Cache und Business Process Engine aufgebaut wird.

In diesem Konzept liegt eine Model View Controller-Architektur auf dem Mobilgerät vor. Darin findet eine Trennung vom Modell, Steuerung und Präsentation statt.



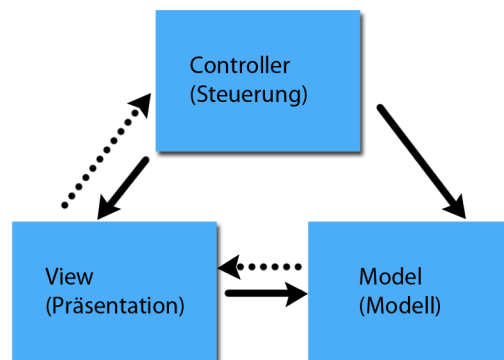


Abbildung 19: Schematischer Aufbau des Model-View-Controller-Konzept

Das Modell und somit die Datenhaltung liegt im Offline-Cache. Die Präsentation liegt in der Oberfläche vor und die Steuerung ist in der Logik. Die Geschäftslogik kann laut Definition in Wikipedia[MVC] „unterschiedlich realisiert werden“ und in der Steuerung oder dem Modell liegen. Bei diesem Konzept liegt sie in der Steuerung vor, da in der Logik das Verhalten gesteuert wird und der Offline-Cache nur der Datenhaltung dient.

## 4.4 Business Process Engine auf dem Mobilgerät

Mobilgerät

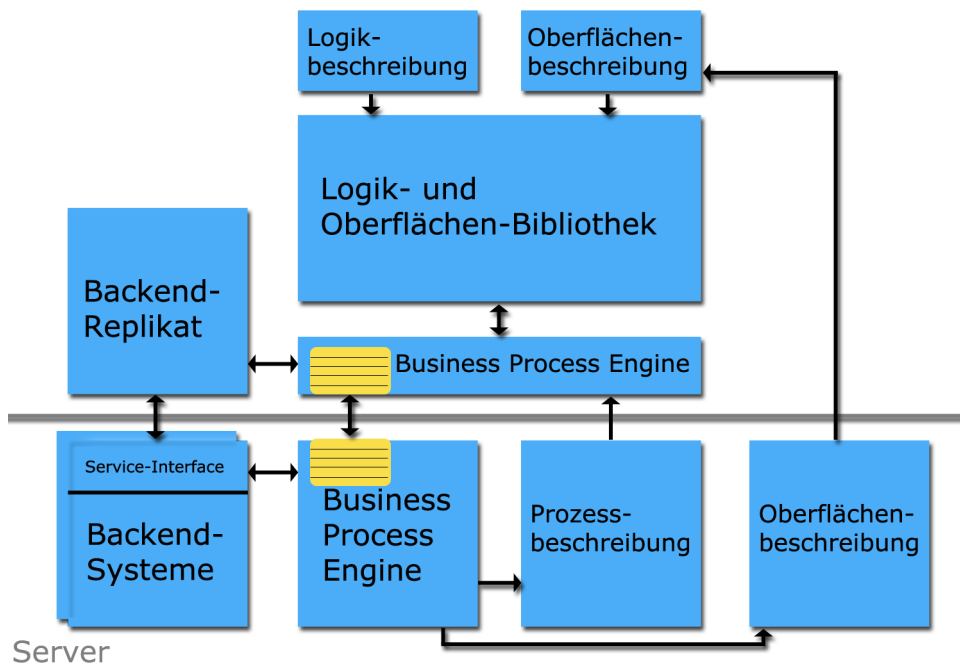


Abbildung 20: Konzept mit Business Process Engine und einem Backend-Replikat auf dem Mobilgerät

Auf dem Mobilgerät wird auf den Compiler und auf den Offline-Cache verzichtet. Durch eine mobile Business Process Engine wird eine vollständige Unterstützung eines Geschäftsprozesses ermöglicht. Durch eine Business Process Engine kann einfacher sichergestellt werden, dass der mobile Geschäftsprozess wie der Online-Geschäftsprozess abläuft.

Die Oberflächenbeschreibung für die Geschäftsprozesse wird auf dem Mobilgerät abgelegt und von der Logik- und Oberflächen-Bibliothek verwendet. Die Logikbeschreibung bleibt für alle Geschäftsprozesse identisch. In der Logik findet eine Beschränkung statt. Es wird nur das Login und eine Menüführung zu den Prozessinstanzen beschrieben. Die Logik der Geschäftsprozesse ist hier nicht enthalten. Für diese mobilen Interaktionen wird eine Oberflächenbeschreibung nötig, die auf dem Mobilgerät vorliegen muss.

Das Backend wird komplett auf dem Mobilgerät gehalten. Ein Geschäftsprozess wird durch einen Backend-Zugriff keine Netzverbindung benötigen und kann durch die Business Process Engine den kompletten Mobilteil eines Geschäftsprozesses ausführen. Das Backend wird die Offline-Fähigkeit unterbrechen, sobald ein Backend-Teil nicht Teil des eigenen Unternehmens ist, sondern über eine Schnittstelle auf ein externes Backend zugreift.

In diesem Konzept liegt eine Model View Controller-Architektur auf dem Mobilgerät vor. Das Modell liegt in der Business Process Engine vor. Die Präsentation liegt in der Oberflächenbeschreibung vor und die Steuerung ist in der Logikbeschreibung. Bei diesem Konzept liegt die Prozessbeschreibung in der Steuerung und somit auf der Business Process Engine vor.

Es müssen auf dem Mobilgerät viele Daten im Backend gehalten werden, die ggf. nur in Prozessschritten gebraucht werden, die nicht auf einem Mobilgerät ausgeführt werden. Dadurch wird Speicher verschwendet und durch die nötige Übertragung die geringe Netzgeschwindigkeit des Mobilgeräts unnötig beansprucht. Das Requirement 800 (Ressourcenbedarf) wird nicht erfüllt.

## 4.5 Ergänzung durch einen Backend-Simulator

Mobilgerät

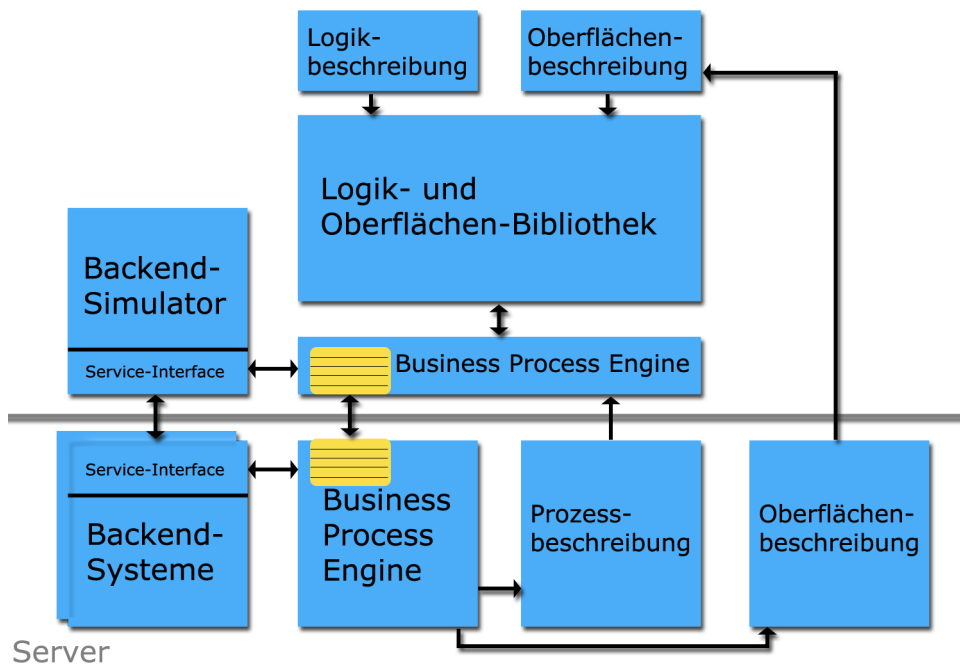


Abbildung 21: Konzept mit Business Process Engine und einem Backend-Simulator auf dem Mobilgerät

Das Konzept mit der mobilen Business Process Engine wird aufgegriffen. Um Ressourcen zu schonen wird das Backend optimiert.

Das Backend kann aufgrund seiner Größe den Speicher eines Mobilgerätes unnötig belasten. Daher ist es sinnvoll, das Backend soweit zu begrenzen, dass Geschäftsprozesse offline mit Backend-Zugriff durchgeführt werden können. Dazu greift die Logik- und Oberflächen-Bibliothek auf das mobile Service-Interface zu. Das Service-Interface greift auf das Backend im Mobilgerät zu, wenn die benötigten Daten darin vorliegen. Bei nicht vorhandenen Daten wird eine Netzverbindung notwendig und ein Offline-Fortführen des Geschäftsprozesses wird verhindert. Schreibende Zugriffe erfolgen nur auf dem mobilen Teil des Backend-Simulators und werden nicht auf dem Server durchgeführt. Schreibzugriffe werden immer durch die Business Process Engine auf dem

Server durchgeführt. Dazu kann die Business Process Engine auf dem Server über das Service-Interface auf das Backend lesend und schreibend zugreifen. Dieser Schritt ist notwendig um sicher zu stellen, dass Backend-Änderungen und Aktivitäten in Geschäftsprozessen auf dem Server in der selben Reihenfolge ausgeführt werden, wie es auf dem Mobilgerät der Fall war.

Dieses Konzept ist aufgrund seiner Eigenschaft, dass es eine bestmögliche Offline-Fähigkeit mit sparsamer Ausnutzung der verfügbaren Performance auf Mobilgeräten bietet eine geeignete Lösung und wird daher im weiteren Verlauf dieser Arbeit ausgearbeitet. Der Vorteil gegenüber dem ersten Konzept ohne Business Process Engine ist, dass eine auf dem Server bestehende Business Process Engine möglicherweise für das Mobilgerät portiert werden kann und somit der Implementierungsaufwand verringert wird.

## **4.6 Mobile Prozessschritte**

Die im Kapitel 3.7 aufgeführten Beschreibungssprachen für Geschäftsprozesse führen in ihren Spezifikationen keine Beschreibungsmöglichkeit für mobile Prozessschritte ein. Im Verlauf dieser Arbeit wird es für den Server notwendig, unterscheiden zu können, wann er einen Geschäftsprozess auf einem Mobilgerät ausführen lässt.

Eine Möglichkeit ist daher das Hinzufügen eines mobilen Prozessschrittes in ein Business Process Management-System. Durch eine grafische Abgrenzung wird bereits im Schaubild ersichtlich, welcher Bereich mobil durchlaufen wird und durch die beschreibende Abgrenzung kann die Business Process Engine die Übergänge zum mobilen Bereich erfassen. Eine solche Implementierung setzt ein offenes Business Process Management-System voraus, da in der grafischen Prozesserstellung, sowie an der Business Process Engine Anpassungen stattfinden müssen.

Durch die Modifikation der Spezifikation der verwendeten Beschreibungssprachen für Geschäftsprozesse wird man den Vorteil des beliebigen Einsatzes von Modellierungs-Tools verlieren.

Um die Spezifikation nicht zu umgehen, wird eine andere Möglichkeit in Betracht gezogen. Dazu wird eine Trennung der Benutzer in Office- und Mobil-Benutzer durchgeführt, durch die ersichtlich wird, welche Prozessschritte mobil durchgeführt werden.

Auf dem Mobilgerät verfolgt die Business Process Engine den selben Ansatz. Sobald ein Prozessschritt erreicht ist, der nicht dem mobilen Benutzer zugeordnet ist, wird die Prozessinstanz auf dem Mobilgerät nicht fortgeführt.

## 4.7 Backend-Simulator

Ein Backend muss nicht von einem Geschäftsprozess abhängen, sondern darf von mehreren Geschäftsprozessen und auch Versionen von Geschäftsprozessen verwendet werden. Andererseits dürfen verschiedene Versionen auf unterschiedliche Backends zugreifen, wenn dies gewünscht sein sollte. Ein Geschäftsprozess darf beliebig viele Backends verwenden und es dürfen unterschiedliche Backends verwendet werden, beispielsweise sind Datenbanken, SAP-Zugriffe und Web-Services möglich.

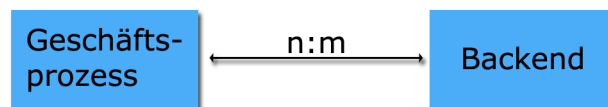


Abbildung 22: Backend zu Geschäftsprozessen

Backends ermöglichen in Geschäftsprozessen viele Möglichkeiten. So kann in der Materialerfassung von den Mitarbeitern aus einer Liste von vorhandenen Materialien ausgewählt werden. Die Liste der Materialien kann über das SAP oder in einer Datenbank verwaltet werden. Materialien können mit vielen Daten hinterlegt werden. Neben Materialeigenschaften und Lagerbeständen können sie mit für den mobilen Benutzer unwichtigen Daten wie Einkaufspreis, Lagerplatz oder einer Auflistung der Bestellungen hinterlegt werden.

Um mit dem knappen Speicher der Mobilgeräte zurecht zu kommen, müssen die Daten reduziert werden. An einem Beispiel im Handwerksprozess soll das Potential verdeutlicht werden.

Im Handwerksprozess ist im Office zu entscheiden, welche Datensätze der Mitarbeiter brauchen wird. Als generelle Vorauswahl kann der Teilabschnitt der Materialien dienen, da die weiteren Teile des Backends für die Materialerfassung nicht von Relevanz sind.

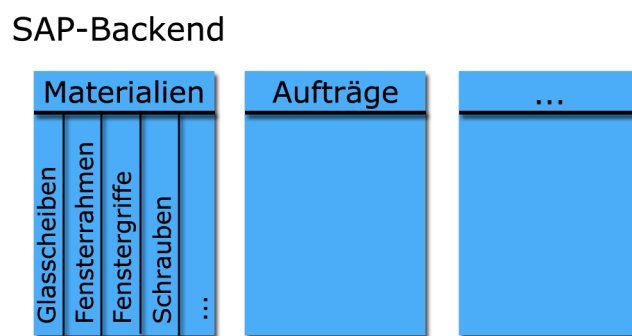


Abbildung 23: Beispielhaftes SAP-Backend

Diese Auswahl kann ggf. weiter begrenzt werden, wenn bereits mit dem Kunden nur ein Austausch eines Fenstergriffes vereinbart wurde. Andere Materialien wie Glasscheiben und Fensterrahmen werden vom Mitarbeiter nicht auf dem Mobilgerät benötigt und müssen daher nicht vorliegen. Die Auswahl wurde somit auf Fenstergriffe und Schrauben zum Befestigen begrenzt.

Die Auswahl kann weiter eingeschränkt werden. Schrauben zum Befestigen sind nur in einer begrenzten Länge notwendig. Auch die Fenstergriffe können eingeschränkt werden, wenn der Kunde dem Office das gewünschte Material nennen konnte und abschließbare Fenstergriffe ausschließen konnte.

Aus dem großen SAP ist auf dem Mobilgerät ein Backend mit einer kleinen Anzahl an Fenstergriffen und Schrauben entstanden, der einen geringen Speicherbedarf hat.

## 4.8 Synchronisierung des Mobilgerätes

Mobilgeräte sollen auf zwei Arten die Möglichkeit haben, Prozessinstanzen zu erhalten.

Solange ein Anwender nur mit einem Mobilgerät arbeitet, sollen Prozessinstanzen automatisch zum Mobilgerät übertragen werden.

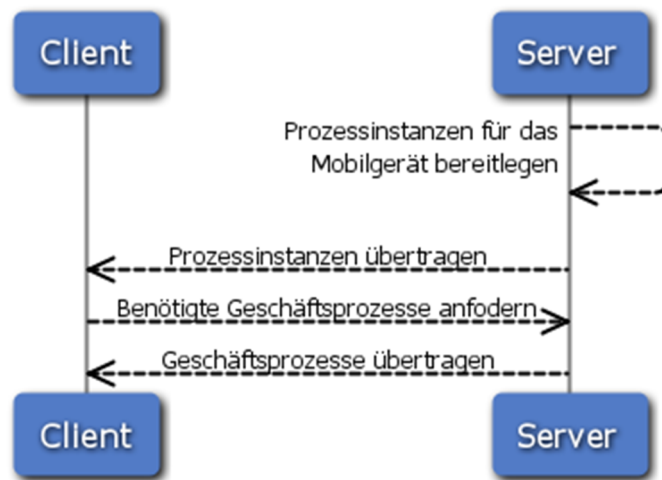


Abbildung 24: Sequenzdiagramm der Client-Server Kommunikation bei Übertragung der Prozessinstanz durch den Server

Der Server stellt anhand des Benutzers der aktuellen Prozessinstanz fest, an wen die Prozessinstanz übertragen werden muss. Daraufhin legt er die Prozessinstanz für das Mobilgerät bereit. Sobald eine Verbindung zwischen Mobilgerät des Benutzers und Server besteht, wird die Prozessinstanz übertragen. Nicht vorhandene Beschreibungen zu Geschäftsprozessen erkennt das Mobilgerät und fordert diese wie in Abbildung 24 zu sehen vom Server an.

Wenn ein Anwender mit verschiedenen Mobilgeräten arbeitet, birgt die automatische Übertragung eine Gefahr. Da eine Prozessinstanz auf mehreren Mobilgeräten vorliegen und offline bearbeitet werden kann, ist die Gefahr einer Inkonsistenz der Daten zu hoch. Daher wird dem Anwender eine Liste mit Prozessinstanzen übergeben, über die der Anwender die Übertragung einer Prozessinstanz auf sein Mobilgerät anfordern kann. Eine Prozessinstanz



kann dabei nur auf einem Mobilgerät des Benutzers liegen. Abbildung 25 verdeutlicht die manuelle Auswahl einer Prozessinstanz.

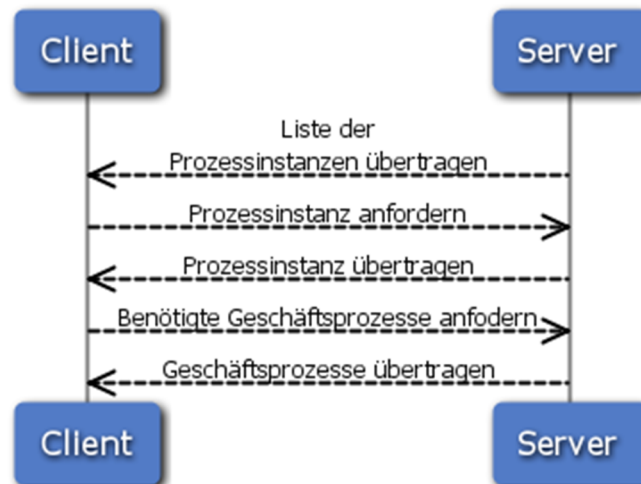


Abbildung 25: Sequenzdiagramm der Client-Server Kommunikation zur Auswahl und Übertragung einer Prozessinstanz

## 4.9 Untersuchung der Offline-Fähigkeit

### Durchgehende Netzverbindung

Bei einer störfreien Netzverbindung kann eine durchgehende Kommunikation zwischen Server und Client stattfinden. Der Server wird zeitgleich auf dem aktuellen Stand gehalten.

Die Offline-Fähigkeit kommt nicht zum Einsatz, solange eine Netzverbindung besteht.

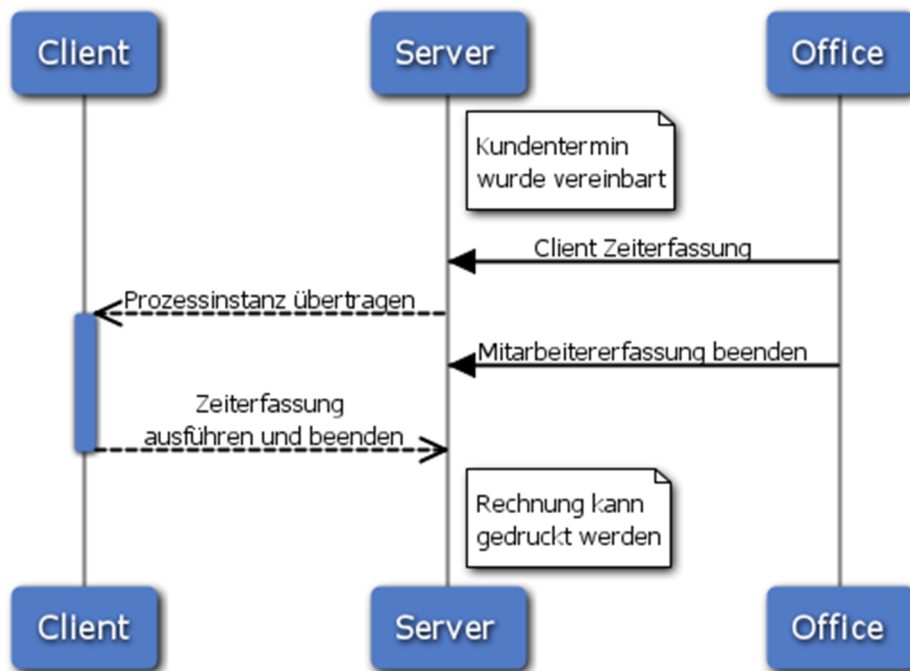


Abbildung 26: Sequenzdiagramm der Client-Server Kommunikation bei durchgehender Netzverbindung

In Abbildung 26 ist der Ablauf bei einer durchgehenden Netzverbindung zu sehen. Im Office wird ein Mitarbeiter die Zeiterfassung für einen Auftrag freigeschaltet und die Mitarbeitererfassung für diesen Auftrag beendet. Der Server überträgt die Prozessinstanz zum Mobilgerät des Clients. Der Anwender führt die Zeiterfassung ein mal durch und beendet die Zeiterfassung für diesen Auftrag. Das Mobilgerät erkennt anhand der Prozessbeschreibung, dass nach der Zeiterfassung ein Join und darauf ein nicht mobiler Schritt stattfindet und führt die Prozessinstanz nicht weiter aus.

Die Prozessinstanz wird auf dem Server nach jedem mobilen Schritt aktualisiert, indem die mobilen Aktivitäten auf dem Server wiederholt werden.

Eine abschließende Übertragung der Prozessinstanz vom Mobilgerät auf den Server ist nicht nötig. Denn sobald das Mobilgerät eine Prozessinstanz erhält, haben die Business Process Engines von Mobilgerät und Server eine identische Prozessinstanz. Änderungen durch Aktionen der Anwender auf dem

Mobilgerät werden auf dem Server wiederholt, wodurch die Prozessinstanzen weiterhin identisch bleiben. Somit haben nach der Durchführung Mobilgerät und Server eine identische Prozessinstanz.

### **Unregelmäßige Netzverbindung**

Eine durchgängige Netzverbindung kann nicht vorausgesetzt werden. Daher ist es notwendig, zu prüfen, wie sich das Konzept bei einer unregelmäßigen Netzverbindung bewährt.

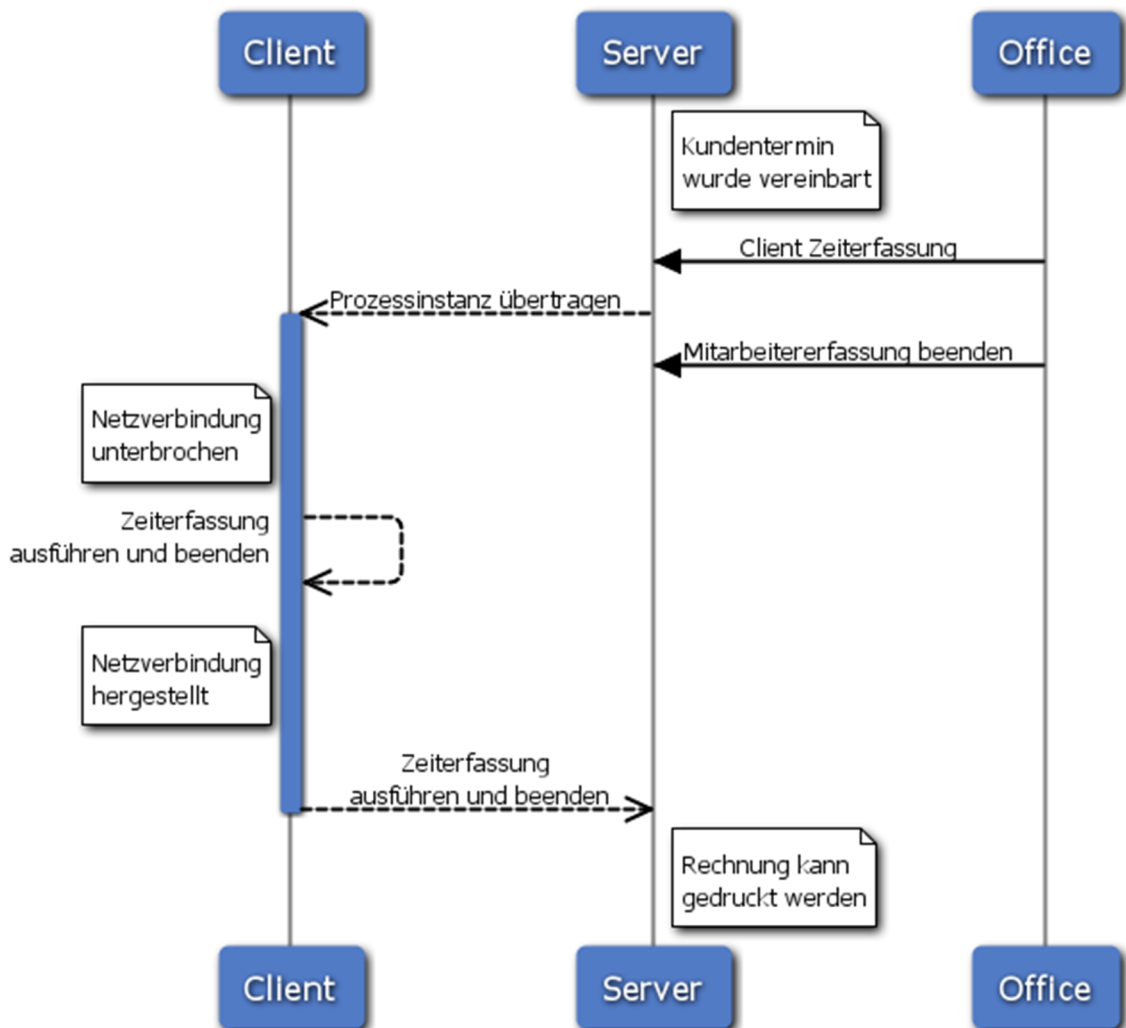


Abbildung 27: Sequenzdiagramm der Client-Server Kommunikation bei ausfallender Netzverbindung

In Abbildung 27 ist zu sehen, dass bis zur Übertragung der Prozessinstanz auf das Mobilgerät alles identisch zur Abbildung 26 ist. Nachdem das Mobilgerät die Prozessinstanz hat, wird die Netzverbindung getrennt.

## **Keine Netzverbindung**

Ein Auftrag kann in ländlichen Gebieten oder abgeschirmten Räumen stattfinden, in denen keine Netzverbindung besteht. In dem Fall kann ein für das Mobilgerät anstehender Prozessschritt erst bearbeitet werden, sobald eine einmalige Netzverbindung aufgebaut wurde. Dazu sollte das Mobilgerät beispielsweise morgens bei Arbeitsantritt in der Firmenzentrale manuell angeschaltet werden, wodurch sich das Mobilgerät aktualisieren lässt. Durch die Offline-Fähigkeit wird auf der Baustelle im ländlichen Gebiet keine Netzverbindung mehr benötigt. Materialien und die Zeiten können offline erfasst werden. Mitarbeiter können am selben Tag zum Feierabend oder am darauf folgenden Arbeitstag ihr Mobilgerät in der Firmenzentrale erneut anschalten, um die erfassten Daten online zu aktualisieren. Dabei werden alle Erfassungsschritte nacheinander auf dem Server abgearbeitet und online aktualisiert. In Abbildung 28 ist dieses Beispiel aufgezeigt.

Durch Office-Mitarbeiter kann mit Hilfe des Monitorings täglich der aktuelle Stand eingesehen werden.

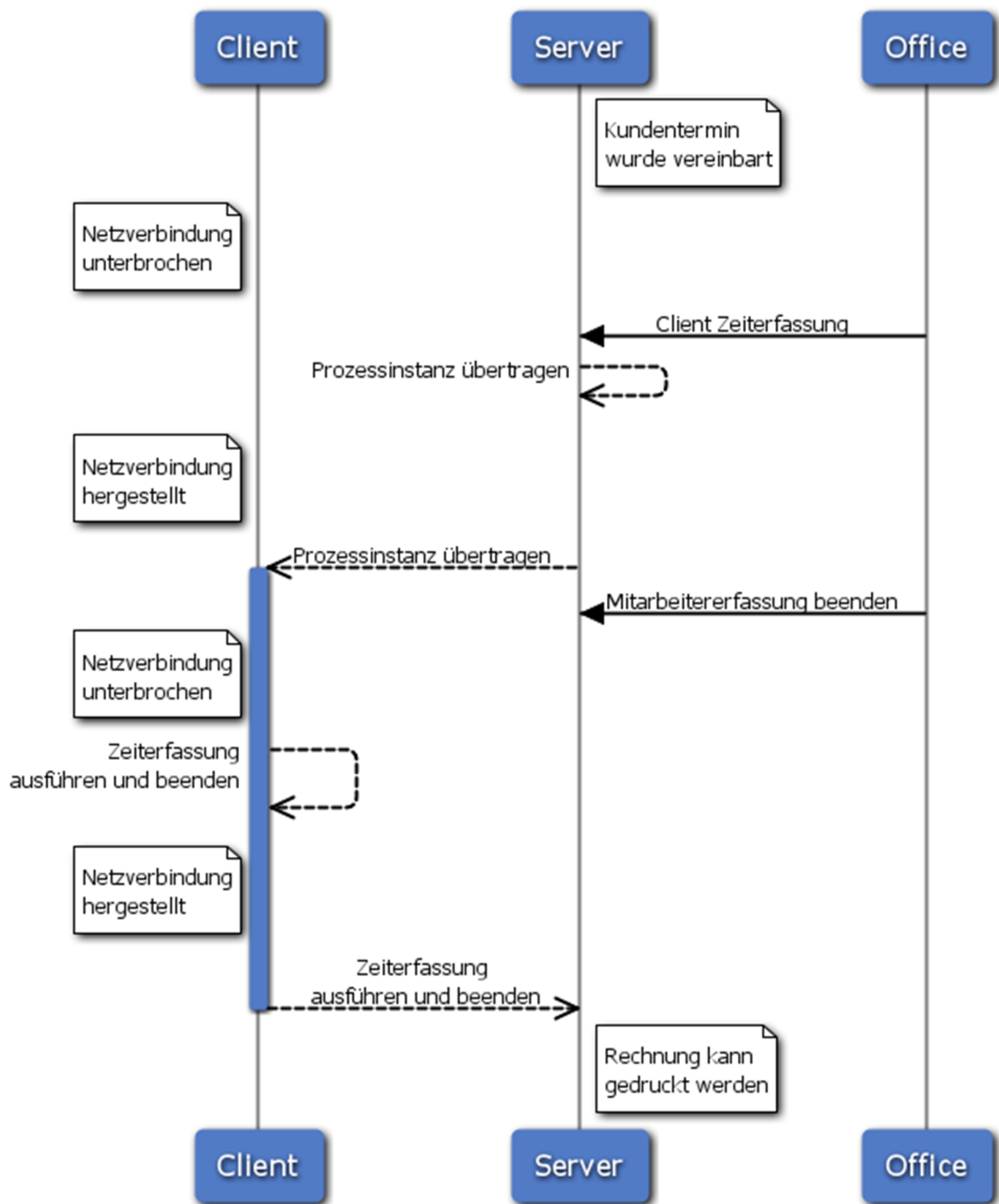


Abbildung 28: Sequenzdiagramm der Client-Server Kommunikation bei keiner Netzverbindung

## 4.10 Parallel arbeitende Mobilgeräte

Laut Requirement 700 muss es möglich sein, dass mehrere offline-fähige Mobilgeräte parallel am Handwerksprozess arbeiten. Bei dem Handwerksprozess sind die Aufgaben voneinander unabhängig und können daher parallel bearbeitet werden.

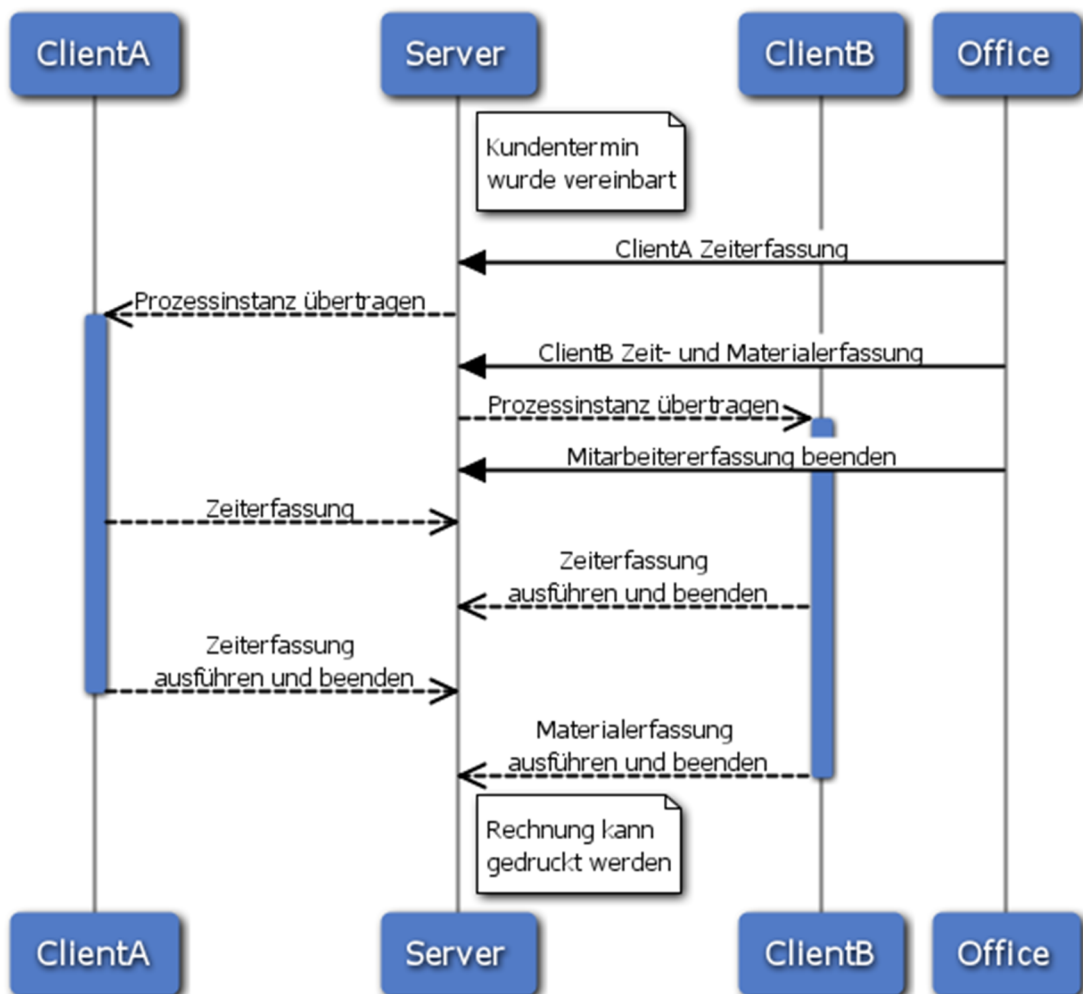


Abbildung 29: Sequenzdiagramm der Client-Server Kommunikation bei zwei Clients

In Abbildung 29 erhält ClientA eine Zeiterfassung und ClientB eine Zeit- und Materialerfassung für den selben Auftrag. ClientA führt eine Zeiterfassung

aus, beendet diese jedoch nicht. Anschließend führt ClientB die Zeiterfassung durch und beendet diese. ClientA kann darauf die Zeiterfassung ergänzen und beenden.

Aus Sicht des Servers laufen zwei unabhängige Prozessschritte parallel. Das Requirement 710 wird erfüllt, da es durch die Mobilmachung der parallelen Prozessschritte aus Sicht des Servers keinen Unterschied macht.

## **4.11 Umgang mit Versionen von Geschäftsprozessen**

In vielen Business Process Management-Systemen ist es möglich, von einem Geschäftsprozess verschiedene Versionen zu pflegen. Dies hat den Vorteil, dass Prozessinstanzen zu Ende geführt werden können, während neue Prozessinstanzen innerhalb der neuen Version gestartet werden. Der Unterschied zwischen den Versionen kann beliebig groß sein und als einzige gemeinsame Eigenschaft dient nur der Name des Geschäftsprozesses. Eine Konvertierung älterer Prozessinstanzen auf die aktuellste Version ist aufgrund der möglichen Unterschiede nicht immer möglich.

Um eine mögliche Fehlerquelle zu umgehen, ist das Konvertieren auszuschließen und auf dem Mobilgerät eine Berücksichtigung verschiedener Versionen zu realisieren.



## 5 Implementierung

Im Verlauf dieses Kapitels werden Erkenntnisse aufgezeigt, die im Zuge der Implementierung zusammen kamen.

### 5.1 PI-Mobile Framework

Die PI-Data AG setzt bei der Entwicklung für mobile Applikationen das eigenentwickelte Framework PI-Mobile ein. Es bietet ein Client-Server Modell und deckt die meisten Mobilgeräte ab, die auf Java ME (MIDP oder Personal Profile) oder Java SE basieren. Die Besonderheiten dieses Frameworks wurden bereits von Gallant[Gal06] erarbeitet und werden im Folgenden daraus zitiert:

- „Das Framework implementiert eine eigene GUI-Bibliothek, die Controls – in MIDP auch Fenster genannt – selbst zeichnet.“
- „Ein eigener, vor allem im Speicherbedarf optimierter, XML-Parser und Binder.“
- „Eine eigene Persistenz-Schicht, mit Implementierungen für XML in Dateien und RecordStores, sowie Datenbanken über JDBC.“
- „Möglichst vieles wird in XML beschrieben, was vom Server interpretiert wird. Der Code für Mobilgeräte wird aus XML generiert, wobei der eigene Parser extrem sparsam mit Speicherressourcen umgeht.“
- „Eigene Modellklassen – als Ersatz für fehlende Collections in J2ME – werden in einem XML-Schema definiert und über einen Codegenerator in Javaklassen geschrieben, um einen typisierten Zugriff zu gewähren.“<sup>4</sup>
- „Geräte werden nach APIs unterschieden und nach Plattform gruppiert.“
- „Es werden ANT-Tasks für verschiedene API-Gruppen eingesetzt.“

---

<sup>4</sup>J2ME wird mittlerweile als Java ME bezeichnet

- „Zum Testen wird kein Simulator benötigt, da das Framework erkennt für welches Gerät es die GUI bauen soll.“

Durch die Verwendung dieses Frameworks werden viele Implementierungsschritte aus dem Konzept vereinfacht. Die Logik- und Oberflächen-Bibliothek, der Backend-Simulator und die Log-Verbindung, die eine Übertragung in der korrekten Reihenfolge ermöglicht, sind im PI-Mobile vorhanden.

## 5.2 Business Process Management-System

Beim Einsatz eines Business Process Management-Systems fiel die Entscheidung zunächst auf BPMN und BPEL aufgrund der Tatsache, dass es bereits eine Implementierung einer Business Process Engine für Mobilgeräte gibt. Das Projekt der Washington University in St. Louis mit dem Namen Sliver [Hac06] eignet sich dank seiner Implementierung auf MIDP für Mobilgeräte und wird in einer Publikation [GHR] als sehr performant angepriesen. Aufgrund der fehlenden technischen Dokumentation stellte sich heraus, dass eine Implementierung mit viel Aufwand verbunden ist. Eine Korrektheit des Codes muss erst nachgewiesen werden. Seit dem Jahr 2006 fand keine Weiterentwicklung an diesem Projekt statt. Es fand ein Versuch statt, Sliver zu verwenden und beispielhafte Geschäftsprozesse aufzusetzen. Durch den Nachteil einer fehlenden technischen Dokumentation und den daraus folgenden Schwierigkeiten bei der Implementierung, wurde der Einsatz von Sliver verworfen.

Im weiteren Verlauf wurde das Business Process Management-System jBPM ausgewählt. Durch den Einsatz der Beschreibungssprache jPDL ergibt sich der im Kapitel 3.7 erwähnte Vorteil der nicht benötigten Transformation zwischen grafischer Notation und der Beschreibung von Geschäftsprozessen. Durch ein ausgereiftes Konzept und die gute technische Dokumentation von jBPM war eine einfachere Implementierung zu erwarten. Laut Freund [FG08] hat jBPM eine „weite Verbreitung gefunden“.

### 5.2.1 Aufbau von jBPM

jBPM ist ein Open Source Business Process Management System und bietet einen großen Umfang an informationstechnischer Unterstützung für Geschäftsprozesse. Über den groben Aufbau von jBPM wird in diesem Kapitel eingegangen.

Die grafische Modellierung von Geschäftsprozessen erfolgt mit einem Eclipse-Plugin, dem jPDL Prozessdesigner (Graphical Process Designer). Geschäftsprozesse werden grafisch in jPDL erstellt und können in XML bearbeitet werden. Geschäftsprozesse können als Projekte in Eclipse angelegt werden und mit der Oberflächenbeschreibung hinterlegt werden. Das Projekt wird mit Hilfe des Plugins aus Eclipse auf den Server übertragen und dort ausführbar gemacht.

Zur Unterstützung von “Human Workflow Management” wird eine Web-Konsole bereitgestellt, die Anwendern die Interaktion in Geschäftsprozessen ermöglicht. Administratoren können über die Web-Konsole Geschäftsprozesse hochladen, administrieren und löschen. Als Web-Server wird JBoss verwendet. Die Authentifizierungs-Logik wird durch JBoss bereitgestellt und kann beliebig erweitert werden, beispielsweise auf Basis von LDAP.

Die jPDL Bibliothek (jPDL Library) ist der Kern von jBPM. Diese ist in Java SE implementiert.

Die Bibliothek ermöglicht das Parsen von Geschäftsprozessen in XML und verwaltet sie. Die Datenbankbindung und Persistenz basiert auf Hibernate und ist daher sehr flexibel bei der Anbindung verschiedener Datenbanken. In der Datenbank werden Prozess-Definitionen, Prozessinstanzen und die Historie verwaltet. Die Bibliothek ist eingebettet und eigenständig lauffähig, auch die Web-Konsole lässt Prozesse über diese Bibliothek laufen.

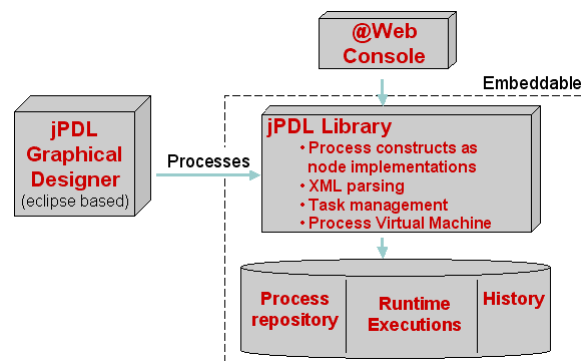


Abbildung 30: Übersicht über jBPM [www.jboss.com](http://www.jboss.com)

Wie in vielen Software-Projekten üblich, sollte man bei einem großen Versions-Update einer Software nicht zeitgleich auf die neueste Version für den produktiven Einsatz setzen. Oftmals sind Bugs in der Software vorhanden, die den produktiven Einsatz gefährden. Auch jBPM hatte mit dem Versions-Update auf die 4. Version mit vielen Bugs zu kämpfen, zu denen ein Bug gehörte, der das parallele Abarbeiten von Aktivitäten verhinderte [For09] und erst im Laufe der Bachelorarbeit behoben wurde. Daher wurde auf die bewährte Version 3.2.6 gesetzt, um in der Prozessmodellierung keine Einschränkungen zu haben.

## 5.3 Server-Implementierung

Auf dem Server liegt das Business Process Management System jBPM und das Backend vor. Ergänzt wird der Server durch Funktionen des PI-Mobile Frameworks.

### 5.3.1 Prozessbeschreibung

Die Prozessbeschreibung wird auch für die Mobilgeräte, die an einem Geschäftsprozess teilnehmen benötigt, da diese ebenfalls anhand der Prozessbeschreibung die mobilen Prozessschritte durchführen sollen.

Prozessbeschreibungen werden im Server nicht als Files gesichert, sondern werden geparsed, im Objekt „ProcessDefinition“ dargestellt und die Objekte wiederum werden über Hibernate in der Datenbank gesichert.

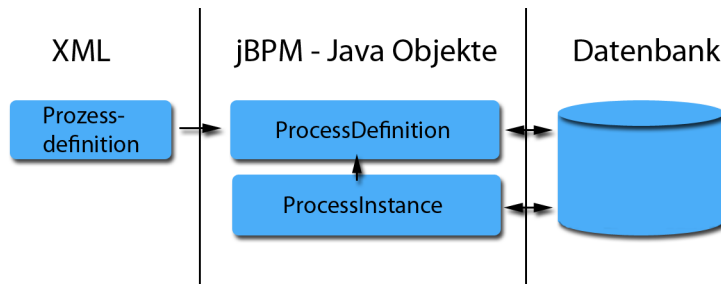


Abbildung 31: Eine Prozessdefinition zu Java-Objekten geparsed und über Hibernate in der Datenbank gesichert

Um an die Prozessdefinition zu gelangen und diese den Mobilgeräten zur Verfügung stellen zu können, sind folgende Wege möglich:

- Prozessbeschreibung vor dem Parsen abfangen und im Filesystem sichern
- Prozessbeschreibung aus dem Objekt „ProcessDefinition“ zusammensetzen
- Prozessbeschreibungen über die Datenbank wiederherstellen

Die erste Methode ist mit dem geringsten Aufwand zu realisieren und wird daher verwendet. Typischerweise werden Geschäftsprozesse für jBPM mit dem beigefügten GPD (Graphical Process Designer), einem Eclipse-Plugin erstellt und deployed. Die Web-Konsole bietet über das Servlet „ProcessUploadServlet“ die Möglichkeit externer Software, Geschäftsprozesse zu deployen. Der GPD greift auf das Servlet zu und überträgt eine komprimierte Zip-Datei, die alle zum Geschäftsprozess gehörenden Daten enthält. Diese Datei enthält:

- Beschreibung des Geschäftsprozesses

- Grafische Abbildung des modellierten Geschäftsprozesses
- Oberflächenbeschreibungen zu Prozessschritten mit menschlicher Interaktion
- XML-Datei zur Zuordnung der Oberflächenbeschreibungen zu den Prozessschritten

Durch eine Erweiterung des Servlets wurde die Zip-Datei abgefangen und auf das Filesystem kopiert, bevor die einzelnen Dateien verwertet und gelöscht werden. Die Zip-Datei dient als Grundlage, um die Clients im späteren Verlauf mit der aktuellen Prozessdefinition versorgen zu können. Als Dateiname wird die vom jBPM vergebenen eindeutige Id des Prozesses verwendet und kann vom Mobilgerät übernommen werden.

Geschäftsprozesse werden von jBPM mit Versionsnummern belegt, wenn diese nicht bereits in der Prozessdefinition definiert werden. Dadurch können Prozesse weiterhin bestehen und komplett beendet werden, während die neueste Version für neue Instanzen verwendet wird.

### **5.3.2 Prozessinstanzen**

Um vom Client aus an einem Geschäftsprozess teilnehmen zu können, wird das Mobilgerät mit der Prozessinstanz versorgt. Anhand der in Abbildung 32 aufgeführten Prozessstruktur wird erklärt, aus welchen Daten sich eine Prozessinstanz für das Mobilgerät zusammensetzen wird.

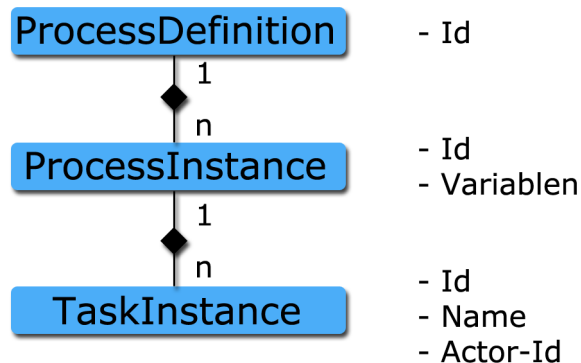


Abbildung 32: In jBPM vorliegende Prozessstruktur

Das Mobilgerät hat die Prozessbeschreibung bereits vorliegen und mit derselben Id versehen, wie sie auch auf dem Server verwendet wird. Ids sind eindeutige Schlüssel und können nicht mehrfach vorkommen. Das Mobilgerät benötigt nur noch die Prozessinstanz und die dazugehörige ProzessId.

Die Prozessinstanz hat wiederum eine eindeutige Id und kann Variablen enthalten, die im Geschäftsprozess erarbeitet wurden. Außerdem hat eine Prozessinstanz eine beliebige Anzahl an Aufgaben (TaskInstances). Die Aufgaben werden über eine Id identifiziert, haben einen Namen, der dem Prozessschritt entspricht und eine Benutzerzuordnung, der Actor-Id. Aufgaben bilden ein Node (Prozessschritt), der aus der Prozessdefinition ersichtlich ist und daher nicht übertragen werden muss.

Aufgaben bilden für Mobilgeräte den benötigten Teil zum Fortführen eines Prozesses. Aus der übergeordneten Prozessinstanz sind die Variablen und eine Id zur Zuordnung nötig. Zur Zuordnung zu einem Geschäftsprozess ist die Id der Prozessdefinition nötig.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<taskInstance id="102">
  <variables>
    <entry>
      <key>anzahlung</key>
      <value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xsi:type="xs:string">nein</value>
    </entry>
  </variables>
  <node>Erfassung freischalten</node>
  <actorId>manager</actorId>
  <processInstance id="40">
    <processDefinition id="79"/>
  </processInstance>
</taskInstance>

```

Abbildung 33: Exportierte Aufgabe aus jBPM

Aufgaben (Nodes) werden in jBPM durch einen Status versehen

- Not Started
- Not Started (Suspended)
- Running
- Suspended
- Ended

Der Status einer Aufgabe muss auf „Not Started“ oder „Running“ stehen, um ihn auf das Mobilgerät übertragen zu können. Bei „Not Started (Suspended)“ und „Suspended“ wird in aller Regel ein Administrator oder Office-Mitarbeiter den Status manuell suspendiert haben. Der Status „Ended“ steht für einen beendeten Prozessschritt, der bereits durchgeführt wurde.

Prozessinstanzen werden in jBPM durch einen Status versehen

- Running
- Suspended



- Ended

Prozessinstanzen werden nur im Status „Running“ an Mobilgeräte übergeben.

Prozessinstanzen beinhalten weitere Informationen, wie zum Beispiel das Startdatum einer Prozessinstanz. Diese Informationen spielen für den Mobilteil keine Rolle und werden daher nicht aufgeführt. Die für den mobilen Teil erforderlichen Informationen werden aus den in Abbildung 33 dargestellten Klassen entnommen.

### **5.3.3 Aktualisierung durch den Client**

Eine zum Mobilgerät übertragene Prozessinstanz wird vom mobilen Anwender bearbeitet. Die API von jBPM bietet sich für Änderungen an Prozessinstanzen an. Dazu stellt der Server einen Service für die Mobilgeräte bereit. Mobilgeräte greifen über ServiceCalls auf die API zu.

Anhand der Ids von Prozessinstanz und Aufgabeninstanz im Kontext von jBPM, kann auf die korrekte Prozessinstanz zugegriffen werden. Im weiteren Verlauf werden Variablen aus den Benutzereingaben in den Kontext der Prozessinstanz gesetzt. Abschließend wird die Aufgabe abgeschlossen.

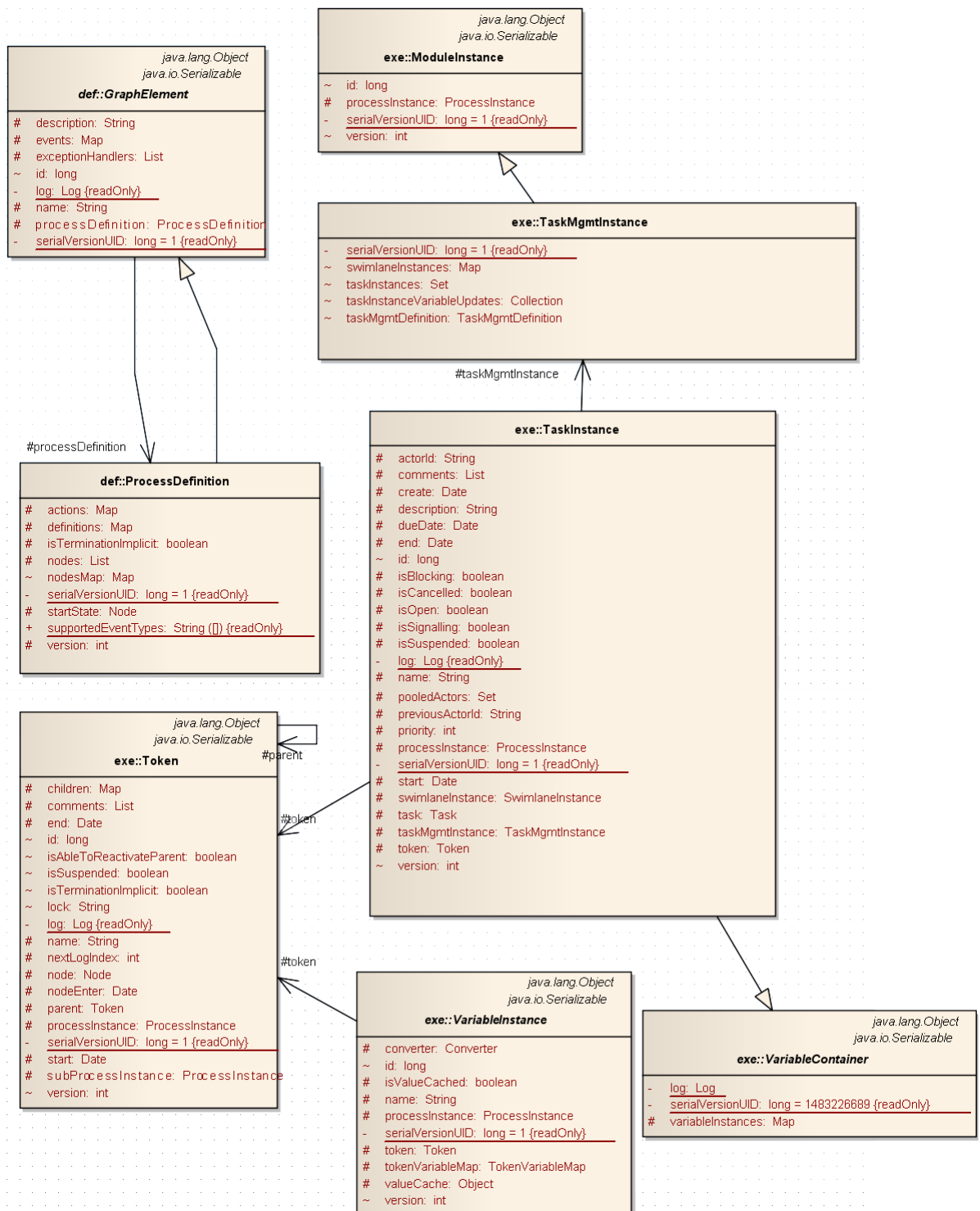


Abbildung 34: In jBPM vorhanden Klassen, die Informationen zur Prozessinstanz beinhalten

## 5.4 Client-Implementierung

Das PI-Mobile Framework deckt die geforderte Funktionalität der Logik- und Oberflächen-Bibliothek ab. Auch eine Transaktionsverbindung und die Backend-Simulation sind im PI-Mobile vorhanden.

Die Business Process Engine von jBPM ist in Java SE realisiert. Das Spektrum an Mobilgeräten, die durch das PI-Mobile unterstützt werden, enthält jedoch Java ME (MIDP oder Personal Profile). Daher ist eine Anpassung der Business Process Engine an Java ME notwendig, um das Spektrum weiterhin zu erhalten.

Für einen funktionierenden Prototypen wurde auf eine komplette Business Process Engine verzichtet. Stattdessen wird die benötigte Funktion für den Handwerksprozess nachgebildet.

Zum Simulieren ist es notwendig unterscheiden zu können, ob die Material- oder Zeiterfassung beendet wurde, oder erhalten bleibt. Zusätzlich sind die erfassten Daten (Variablen) zu sichern. Nach dem Beenden der Erfassung wird die Prozessinstanz auf dem Server aktualisiert. Dazu werden die Variablen auf dem Server gesetzt und der Prozessschritt beendet.

Das Einlesen der XML-Data durch das PI-Mobile erfordert ein Schema. Ein zur Prozessinstanz passendes Schema wurde erstellt. Das PI-Mobile hat aus dem Schema die Klassen für Prozessdefinition, Prozessinstanz und Aufgabeninstanz erstellt und die relation zwischen den Klassen hergestellt. Das Binden des Beispiels aus Abbildung 33 hat fehlerfrei funktioniert.

Zum Aktualisieren des Servers, ruft der Client über ServiceCalls API-Funktionen von jBPM auf.

## 5.5 Backend

Durch den Backend-Simulator im PI-Mobile Framework ist bereits eine Implementierung vorhanden, durch die der Softwareentwickler entscheidet, welche Teile des Backends offline zur Verfügung stehen sollen und für welche Teile

eine Verbindung benötigt wird. Dadurch kann sparsamer mit Speicher auf dem Mobilgerät umgegangen werden. Die Anwendung sieht ein vollständiges Backend und greift immer auf den Backend-Simulator zu.

Das Aktualisieren eines Backends auf dem Mobilgerät erfolgt entweder durch eine Synchronisierung des Clients oder vom Server aus. Dem Entwickler sind viele Freiheiten gesetzt, da Backends sich verschieden verhalten. Denn nicht immer ist es dem Server möglich, bei einem Backend-Update informiert zu werden oder die genauen Änderungen zu erfahren. Daher ist eine Flexibilität notwendig.

## 6 Zusammenfassung

Die Bachelorarbeit analysiert die Integration von Mobilgeräten in Geschäftsprozessen. Geschäftsprozesse mit Außendienstmitarbeitern kommen beispielsweise im Service, Vertrieb oder im Handwerk vor. Der in der Bachelorarbeit erarbeitete Beispielprozess aus dem Handwerk ermöglicht, dass ein Handwerker vor Ort beim Kunden die verwendeten Materialien erfassen und seine Zeiterfassung auf dem Mobilgerät durchführen kann. Ein wichtiger Aspekt war eine zu realisierende Offline-Fähigkeit und die begrenzte Hardware von Mobilgeräten.

Es wurde ein Konzept erarbeitet, das die Anforderungen einer Offline-Fähigkeit erfüllt. Realisiert wurde das Konzept durch das Abbilden einer Business Process Engine auf den Mobilgeräten. Die Business Process Engine ermöglicht die Ausführung von Geschäftsprozessen ohne Netzverbindung zum Server. Die Aktualisierung der Geschäftsprozesse auf dem Server erfolgt, sobald wieder eine Netzverbindung besteht. Das vom Geschäftsprozess benötigte Backend (z.B. SAP) wird zum Teil zur Offline-Fähigkeit auf den Mobilgeräten benötigt. Der mobil benötigte Abschnitt des Backends ist in seinem Umfang winzig. Deshalb wurde das Backend durch ein auf das notwendigste reduzierte Abbild an die beschränkten Ressourcen von Mobilgeräten angepasst.

Für die Realisierung wurde als Basis die bei der Firma PI-Data vorhandene Software genutzt, wobei die Implementierung auf bestehenden Code aufsetzt. Darin ist bereits eine Lösung vorhanden, die ein reduziertes Abbild des Backends ermöglicht. Im produktiven Betrieb hat sich die Lösung seit Jahren bewährt und gehört mittlerweile zu den Erfolgsfaktoren der PI-Data.

## 7 Ausblick

Die informationstechnische Unterstützung von Geschäftsprozessen unterliegt derzeit einer schnellen Entwicklung. Viele große IT-Unternehmen wie IBM, Oracle und SUN bieten Business Process Management-Systeme an. Der Drang vieler Unternehmen, Geschäftsprozesse zu modellieren und automatisieren, bringt diesem Bereich hohe Aufmerksamkeit. Mobilgeräte können aufgrund ihrer Verbreitung und Möglichkeiten eine größere Rolle spielen.

Es ist ungewiss, welche Technik in der Geschäftsprozessmodellierung sich durchsetzen wird. Wie die Untersuchung zeigt, ist für die Spezifikation BPMN 2.0 in Zukunft mit einer weiten Verbreitung zu rechnen.

Innerhalb des vorgegebenen Zeitrahmens wurde die Implementierung nicht vollständig durchgeführt. Das Exportieren einer Prozessinstanz in eine XML-Datei und das Binden im PI-Mobile funktionieren fehlerfrei. Alle kritischen Aspekte der Implementierung wurden somit erfolgreich implementiert und lassen den Schluss zu, dass einer weiteren Implementierung nichts im Weg steht.

Aufgrund einer fehlenden Portierung von jBPM, ist das Laufzeitverhalten auf Mobilgeräten nicht nachgewiesen. Durch eine Verbindung mit PI-Mobile wird ein für den Anwender angenehmes Laufzeitverhalten erwartet.

Anhand der gewonnenen Erkenntnisse wird der abschließende Teil der Implementierung nach der Bachelorarbeit mit Zuversicht fortgeführt. Von einem Erfolg ist auch die PI-Data überzeugt.

## Literatur

- [All] Thomas Allweyer. BPMN 2.0 macht Fortschritte. <http://kurze-prozesse.de/?p=176> Stand: 12.10.2009.
- [BL03] Joerg Becker and Holger Luczak. *Workflowmanagement in der Produktionsplanung und -steuerung: Qualität und Effizienz der Auftragsabwicklung steigern*. Springer Verlag, erste edition, 2003.
- [can09] Smart phones defy slowdown, August 2009. <http://www.canalys.com/pr/2009/r2009081.htm> Stand: 13.12.2009.
- [cld03] JSR 139: Connected Limited Device Configuration 1.1 (Final Release), März 2003. <http://jcp.org/en/jsr/detail?id=139> Stand: 13.12.2009.
- [FG08] Jakob Freund and Klaus Götzer. *Vom Geschäftsprozess zum Workflow: Ein Leitfaden für die Praxis*. HANSER Verlag, erste edition, 2008.
- [For09] JBoss Issue Tracker, 2009. <https://jira.jboss.org/jira/browse/JBPM-2516> Stand: 26.11.2009.
- [Fre08] Jakob Freund. Zwölf Fragen zum Business-Process-Management, Juli 2008. [http://wiki.computerwoche.de/doku.php/soa\\_bpm/bpm\\_faq](http://wiki.computerwoche.de/doku.php/soa_bpm/bpm_faq) Stand: 13.12.2009.
- [Gal06] Christian Gallant. Mobile Learning - Analyse und Entwurf einer mobilen Anwendung zur Unterstützung von Mitarbeitern im Außendienst. Master's thesis, Fachhochschule für Technik Esslingen, 2006. [http://www.pidata.de/resources/download/DA\\_Mobile\\_Learning.pdf](http://www.pidata.de/resources/download/DA_Mobile_Learning.pdf) Stand: 13.12.2009.
- [GHR] Christopher Gill Gregory Hackmann, Mart Haitjema and Gruia-Catalin Roman. Sliver: A BPEL Workflow Process Execution Engine for Mobile Devices. <http://students.cec.wustl.edu/~gwh2/publications/icsoc2006.ppt> Stand: 13.12.2009.

- [Hac06] Gregory Hackmann. Sliver, 2006. <http://mobilab.cse.wustl.edu/projects/sliver/> Stand: 12.11.2009.
- [Kuj06] Holger Kujath. Grafische Modellierung mit anschließender Generierung mobiler Anwendungen. Master's thesis, Fachhochschule Furtwangen, 2006. [http://www.pi-data.de/resources/download/DA\\_Grafische\\_Modellierung.pdf](http://www.pi-data.de/resources/download/DA_Grafische_Modellierung.pdf) Stand: 13.12.2009.
- [mid02] JSR 118: Mobile Information Device Profile 2.0 (Final Release), November 2002. <http://jcp.org/en/jsr/detail?id=118> Stand: 13.12.2009.
- [MVC] Model View Controller. [http://de.wikipedia.org/wiki/Model\\_View\\_Controller](http://de.wikipedia.org/wiki/Model_View_Controller) Stand: 13.12.2009.
- [net09] Palm pulls the plug on Palm OS, bets the future on Pre's webOS, Februar 2009. <http://www.networkworld.com/news/2009/021109-palm-kills-palm-os-bets-on-webos.html?hpg1=bn> Stand: 13.12.2009.
- [Sch07] Klaus-Dieter Schmatz. *Java Micro Edition: Entwicklung mobiler JavaME-Anwendungen mit CLDC und MIDP*. dpunkt Verlag, zweite edition, 2007.
- [Sch08] David Schumm. Grafische Modellierung von BPEL Prozessen unter Verwendung der BPMN Notation. Master's thesis, Universität Stuttgart, 2008. [ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart\\_fi/DIP-2720/DIP-2720.pdf](ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/DIP-2720/DIP-2720.pdf) Stand: 13.12.2009.
- [Str08] Stefan Strohmeier. *Informationssysteme im Personalmanagement*. Vieweg+Teubner Verlag, 2008.



## Abbildungsverzeichnis

1	Mobiltelefon . . . . .	10
2	PDA . . . . .	10
3	Smartphone . . . . .	11
4	Notebook . . . . .	12
5	Eine Bildschirmtastatur blendet einen Teil des Inhaltes aus . .	13
6	Im Querformat wird der Bildschirminhalt von unten beschnitten	14
7	Ausgelieferte Betriebssysteme im zweiten Quartal 2009 . . . .	15
8	Architektur von Java ME . . . . .	17
9	Beispiel eines groben Geschäftsprozesses . . . . .	20
10	Aufbau eines Business Process Management-Systems [BL03] .	22
11	jBPM Webconsole . . . . .	23
12	Darstellung der Business Process Modeling Notation . . . . .	25
13	Darstellung von jPDL (Version 4.0) . . . . .	27
14	Möglichkeiten für die PI-Data . . . . .	30
15	Handwerksprozess in jPDL . . . . .	33
16	Beispielhafter Aufbau einer IT-Landschaft . . . . .	35
17	Aufbau des Servers . . . . .	38
18	Konzept ohne Business Process Engine auf dem Mobilgerät . .	39
19	Schematischer Aufbau des Model-View-Controller-Konzept . .	41
20	Konzept mit Business Process Engine und einem Backend- Replikate auf dem Mobilgerät . . . . .	42
21	Konzept mit Business Process Engine und einem Backend- Simulator auf dem Mobilgerät . . . . .	44
22	Backend zu Geschäftsprozessen . . . . .	46

23	Beispielhaftes SAP-Backend . . . . .	47
24	Sequenzdiagramm der Client-Server Kommunikation bei Übertragung der Prozessinstanz durch den Server . . . . .	48
25	Sequenzdiagramm der Client-Server Kommunikation zur Auswahl und Übertragung einer Prozessinstanz . . . . .	49
26	Sequenzdiagramm der Client-Server Kommunikation bei durchgehender Netzverbindung . . . . .	50
27	Sequenzdiagramm der Client-Server Kommunikation bei ausfallender Netzverbindung . . . . .	52
28	Sequenzdiagramm der Client-Server Kommunikation bei keiner Netzverbindung . . . . .	54
29	Sequenzdiagramm der Client-Server Kommunikation bei zwei Clients . . . . .	55
30	Übersicht über jBPM <a href="http://www.jboss.com">www.jboss.com</a> . . . . .	60
31	Eine Prozessdefinition zu Java-Objekten geparsed und über Hibernate in der Datenbank gesichert . . . . .	61
32	In jBPM vorliegende Prozessstruktur . . . . .	63
33	Exportierte Aufgabe aus jBPM . . . . .	64
34	In jBPM vorhanden Klassen, die Informationen zur Prozessinstanz beinhalten . . . . .	66

## Tabellenverzeichnis

1	Hardware-Anforderungen von CLDC und MIDP . . . . .	18
2	Übersicht verschiedener Prozessengines . . . . .	28

## Abkürzungsverzeichnis

API	Application Programming Interface
BPEL	Business Process Execution Language
BPMN	Business Process Management Notation
BPMS	Business Process Management System
CLDC	Connected, Limited Device Configuration
CRM	Customer-Relationship-Management
EDGE	Enhanced Data Rates for GSM Evolution
ERP	Enterprise Resource Planning
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HSCSD	High Speed Circuit Switched Data
HSDPA	High Speed Downlink Packet Access
JCP	Java Community Process
JDBC	Java Database Connectivity
jPDL	jBPM Process Definition Language
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
MIDP	Mobile Information Device Profile
PDA	Personal Digital Assistant

PIM	Personal Information Management
UMTS	Universal Mobile Telecommunications System
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

## Glossar

**Backend** Eine Datenbank oder ein System, das zur Datenhaltung oder Verwaltung verwendet wird.

**Business Process Engine** Die Business Process Engine dient der Ausführung von Geschäftsprozessen.

**Business Process Execution Language** BPEL ist eine ausführbare Beschreibungssprache für Geschäftsprozesse und verfügt über keine standardisierte grafische Notation.

**Business Process Management** Umfasst Methoden zur Gestaltung von Geschäftsprozessen und muss nicht informationstechnisch gelöst sein.

**Business Process Management System** Umfasst ein komplettes System zum modellieren, ausführen und erfassen von Geschäftsprozessen.

**Business Process Modeling Notation** BPMN dient der visuellen Darstellung von Geschäftsprozessen.

**Node** Ein in UML verwendeter Begriff für einen Knoten. In der Modellierung von Geschäftsprozessen unter dem Begriff Prozessschritt anzutreffen.

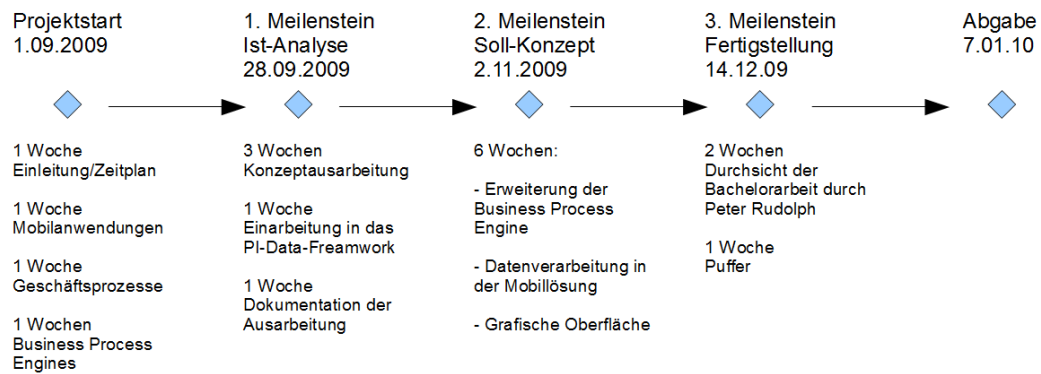
**Geschäftsprozess** Ein Geschäftsprozess ist ein Ablauf von Tätigkeiten in einem Unternehmen, mit einer Ausrichtung auf ein Ziel. Im Ablauf tritt ein kontinuierliches Muster auf.

**Prozessinstanz** Eine Prozessinstanz repräsentiert ein konkreter Vorgang eines Geschäftsprozesses.

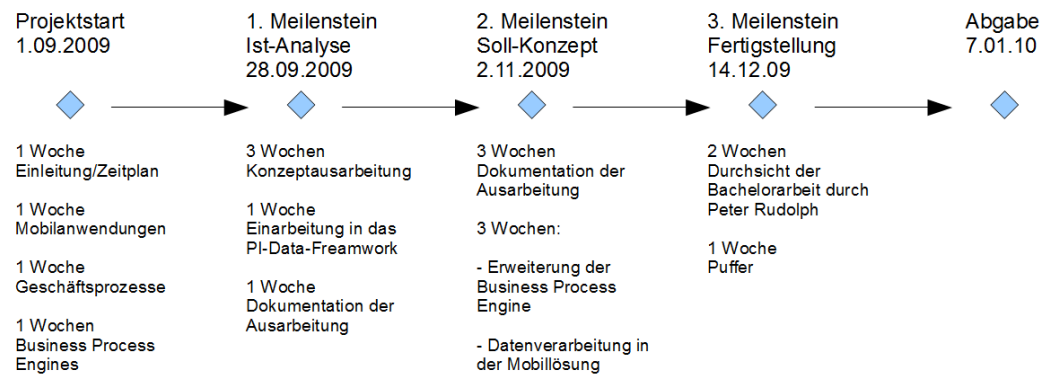
**Prozessschritt** Eine Arbeitsschritt aus einem Geschäftsprozess.

# Anhang

## Geplanter Zeitplan



## Tatsächliche Zeitverteilung



## Handwerksprozess aus Kapitel 3.8

```

<?xml version="1.0" encoding="UTF-8"?>

<process-definition xmlns="urn:jbpm.org:jbpm-3.2"
  name="handwerk2">

  <swimlane name="Bauleitung">
    <assignment actor-id="manager"></assignment>
  </swimlane>

  <swimlane name="Office">
    <assignment actor-id="manager"></assignment>
  </swimlane>

  <start-state name="Auftrags-Start">
    <task swimlane="Office"/>
    <transition to="Anzahlung vereinbart"></transition>
  </start-state>

  <decision name="Anzahlung vereinbart">
    <transition to="Termin vereinbaren" name="nein">
      <condition expression="#{anzahlung==&quot;nein&quot;}" />
    </transition>
    <transition to="Rechnung mit Anzahlung drucken" name="ja">
      <condition expression="#{anzahlung==&quot;ja&quot;}" />
    </transition>
  </decision>

  <task-node name="Rechnung mit Anzahlung drucken">
    <task swimlane="Office"/>
    <transition to="Anzahlung eingetroffen"></transition>
  </task-node>

  <task-node name="Anzahlung eingetroffen">
    <task swimlane="Office"/>
    <transition to="Termin vereinbaren"></transition>
  </task-node>

  <task-node name="Termin vereinbaren">
    <task swimlane="Office"/>
    <transition to="fork1"></transition>
  </task-node>

  <fork name="fork1">
    <transition to="Auswahl" name="erfassen"></transition>
    <transition to="Erfassung freischalten"
      name="Erfassung beginnen"></transition>
  </fork>

  <task-node name="Zeiterfassung">
    <task>
      <assignment actor-id="#{user}"></assignment>
    </task>
    <transition to="Zeiterfassung" name="weiter"></transition>
    <transition to="join1" name="beenden"></transition>
  </task-node>

  <join name="join1">
    <transition to="Rechnung drucken"></transition>
  </join>

```

```

<task-node name="Rechnung drucken">
  <task swimlane="Office"/>
  <transition to="Auftrags-Ende"></transition>
</task-node>

<task-node name="Materialerfassung">
  <task>
    <assignment actor-id="{user}"></assignment>
  </task>
  <transition to="Materialerfassung" name="weiter"></transition>
  <transition to="join1" name="beenden"></transition>
</task-node>

<decision name="Auswahl">
  <transition to="Zeiterfassung" name="Zeit">
    <condition expression="{auswahl=='"zeit"}' />
  </transition>
  <transition to="Materialerfassung" name="Material">
    <condition expression="{auswahl=='"material"}' />
  </transition>
  <transition to="join1" name="to join1"></transition>
</decision>

<task-node name="Erfassung freischalten">
  <task swimlane="Bauleitung"/>
  <transition to="Erfassung erledigt"></transition>
</task-node>

<decision name="Erfassung erledigt">
  <transition to="join1" name="ja">
    <condition expression="{ende=='"ja"}' />
  </transition>
  <transition to="fork1" name="nein"></transition>
</decision>

<end-state name="Auftrags-Ende"></end-state>

</process-definition>

```



## Schema der exportierten Prozessinstanz

```

<?xml version="1.0" encoding="ASCII" ?>
<xsd:schema id="DemoFactory" targetNamespace="http://www.pidata.de/res/Demo.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.pidata.de/res/Demo.xsd">

  <xsd:element name="taskInstance">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="contextInstance" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="node" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="actorid" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:long"/>
    </xsd:complexType>

    <xsd:key name="key_contextInstance">
      <xsd:selector xpath="contextInstance"/>
      <xsd:field xpath="@id"/>
    </xsd:key>
  </xsd:element>

  <xsd:element name="contextInstance">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="processInstance" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="variable" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:long"/>
    </xsd:complexType>

    <xsd:key name="key_processInstance">
      <xsd:selector xpath="processInstance"/>
      <xsd:field xpath="@id"/>
    </xsd:key>
  </xsd:element>

  <xsd:element name="processInstance">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="processDefinition" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:long"/>
    </xsd:complexType>

    <xsd:key name="key_processDefinition">
      <xsd:selector xpath="processDefinition"/>
      <xsd:field xpath="@id"/>
    </xsd:key>
  </xsd:element>

```

```

<xsd:element name="processDefinition">
  <xsd:complexType>
    <xsd:attribute name="id" type="xsd:long"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="variable">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="entry" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="entry">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="key" type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="value" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

</xsd:schema>

```