

PERSONAL INFORMATION GUIDE – EINE PLATTFORM MIT LOCATION BASED SERVICES FÜR MOBILE POWERED E-COMMERCE

Sebastian Herden, Claus Rautenstrauch, André Zwanziger, Marco Plack

Metop GmbH
Sandtorstr. 23
D-39106 Magdeburg

Abstract: Im folgenden Beitrag wird das Fach- und Technologiekonzept eines Personal Information Guide (PIG) vorgestellt, mit dem konventionelle Internet-Dienste für mobile Anwendungen auf der Grundlage von Location-based Services verfügbar gemacht werden. Ausgehend von einer kurzen Situationsanalyse zum Stand der mobilen Anwendungen und den sich daraus ergebenden Anforderungen wird zunächst das Konzept des PIG mobile Client erläutert. Der Schwerpunkt liegt dann auf der Vorstellung der PIG-Server-Infrastruktur, die auf Agenten-, XML- und Peer-to-Peer-Technologie basiert. Es wird gezeigt, dass sich hiermit kostengünstige und hochgradig flexible Hochverfügbarkeitslösungen implementieren lassen. Der Beitrag schließt mit einer Zusammenfassung und einen Ausblick auf weitere Entwicklungen.

1 M-Commerce und mobile powered E-Commerce

Elektronischer Handel über drahtlose Kommunikation mit Mobiltelefonen, Personal Digital Assistants (PDAs) oder anderen ortsunabhängigen Geräten (mobile devices) wird *Mobile Commerce* oder *M-Commerce* genannt (aktuelle Werke hierzu sind z. B. [GeGo01; MiSc01; NiPe01; Zobe01]). Damit ist M-Commerce ein Spezialfall von E-Commerce, für den folgende Besonderheiten gelten:

- Die Endgeräte (auch mobile Devices wie z. B. Mobiltelefone oder PDAs) haben ein begrenztes Display und nur eingeschränkte Grafikfähigkeiten.
- Die Übertragungsraten zu den Endgeräten über Mobilfunknetze sind verglichen mit Festnetzverbindungen (noch) gering und die Übertragungskosten hoch.
- Die Endgeräte können jederzeit an jedem Ort eingesetzt werden.
- Zusätzlich zum E-Commerce können nicht nur personalisierte, sondern auch ortsabhängige Dienste angeboten werden.

M-Commerce löst die Nutzung von Internet-Diensten von räumlichen Zwängen. Weitere Mehrwerte lassen sich durch *Location Based Services* (LBS) erschließen. Hierbei werden Internet-Dienste mit Ortungsinformationen des Dienstinhabers, die mit GPS-Satellitenortung (*GPS* = Global Positioning System) oder den seit November 2001 freigeschalteten GSM-Ortungsdiensten (*GSM* = Global System for Mobile Communication) ermittelt werden, gekoppelt.

In diesem Beitrag werden Konzept und Architektur des Personal Information Guide (PIG), einer agentenbasierten Service-Plattform für Location Based Services, vorgestellt und gegen bestehende Entwicklungen abgegrenzt. Die Grundidee von PIG ist es, bestehende Internet-Dienste über eine gemeinsame Plattform für mobile Nutzer effizient verfügbar zu machen. Daher wird hier von „mobile powered E-Commerce“ gesprochen.

2 Beispielszenario für die Anwendung von LBS

Ein Außendienstmitarbeiter entschließt sich kurzfristig zu einem Kundenbesuch in einer entfernten Stadt. Der Besuch dauert länger als geplant, so dass er sich entschließt, dort zu übernachten. Nun braucht er ein Hotelzimmer und ein Restaurant, in das er den Kunden abends einladen möchte. Er wählt den Service „Hotelsuche“ seines Navigationssystems. Dieses ermittelt die GPS-Koordinate und leitet sie über ein Mobilfunknetz an einen Dienstanbieter weiter, der den Nachfrager anhand seiner Mobilfunknummer identifiziert. Dieser ermittelt ein freies Hotelzimmer, das den im Benutzerprofil gespeicherten individuellen Anforderungen bezogen auf Preis, Lage oder auch Sondervereinbarungen zwischen Arbeitgeber und Hotelketten entspricht bzw. sehr nahe kommt. Der Außendienstler wählt die Option „Buchen“ und das Hotelzimmer wird für ihn reserviert. Die Hoteladresse wird an das Navigationssystem weiter gegeben, das ihn dann zum Ziel führt. In gleicher Weise verfährt er beim Service „Restaurantsuche“, der ihm allerdings mehrere Restaurants zur Auswahl stellt und bei dem in der Funktion „Tischreservierung“ noch die Anzahl Personen angegeben werden muss. Dieses Beispiel stellt folgende Kennzeichen heraus:

- Der Nutzer kommt mit nur zwei primitiven Eingaben zum Ziel.
- Auf der Server-Seite liegt detailliertes Wissen über die Nutzeranforderungen bezogen auf den Service vor.
- Als Client wird das Navigationssystem bzw. Car-PC im PKW verwendet. Die Kommunikation zwischen Navigationssystem und Mobile Service beschränkt sich dabei auf die Übergabe der GPS-Koordinate und der Übernahme der Zieladresse.
- Die Kommunikation über Mobilfunk beschränkt sich auf die Übergabe der GPS-Koordinate und der Anfrage nach einem Hotel und die Übernahme der Hotelinformation.

Am nächsten Tag stellt der Außendienstler fest, dass er tanken muss. Der Bordcomputer berechnet die Reichweite aus Tankinhalt und Durchschnittsverbrauch und der Mobile Service „Tankstellensuche“ übermittelt die nächste günstige Tankstelle innerhalb der Reichweite. Dieses Beispiel zeigt einen sinnvollen Integrationsansatz mit der Fahrzeugelektronik.

Danach beschließt der Außendienstler einen spontanen Kundenbesuch. Zur Vorbereitung schreibt er auf seinem PDA ein Angebot. Dieses wird über den mobile Service „Brief drucken“ im Corporate Design des Unternehmens beim Kunden ausgedruckt, so dass bereits beim Eintreffen des Außendienstlers eine aussagefähige Verhandlungsgrundlage beim Kunden vorliegt. In diesem Beispiel wird als mobile Eingabeeinheit ein PDA verwendet. Ortungsinformationen sind hier nicht relevant.

3 Anforderungen an eine Plattform für LBS

Aus dem oben skizzierten Szenario und weiteren Überlegungen lassen sich folgende allgemeine Anforderungen für die Gestaltung einer Plattform für LBS ableiten:

- *Endgeräteunabhängigkeit:* Die Nutzung von LBS soll über alle verfügbaren mobile Devices wie Notebooks, PDAs, Navigationssysteme, Mobiltelefone oder auch Bordcomputer möglich sein. Hardwarelösungen haben zwar prinzipiell den Vorteil der Ressourceneffizienz, bieten aber keine hinreichende Flexibilität, wenn die Benutzerschnittstelle beispielsweise mit einem Zugang zu neuen Services ergänzt werden soll.
- *Einfache Benutzerschnittstelle:* Die Benutzerschnittstelle muss so einfach wie möglich gestaltet werden, da zum Einen davon auszugehen ist, dass der Benutzer nur gelegentlicher Benutzer mit geringen Fachkenntnissen ist und zum Anderen auch die Nutzung der Dienste während der Fahrt möglich sein soll. Das Beispiel der Hotelbuchung mit zwei Knopfrücken zeigt wie diese Anforderung erfüllt werden kann. Voraussetzung hierfür ist, dass auf der Server-Seite detailliertes Wissen über den Benutzer vorhanden ist und die Recherche-Operationen vollständig auf der Server-Seite ablaufen.
- *Minimale Kommunikation über Mobilfunknetze:* Auch wenn UMTS vor der Tür steht und die Tage von GSM und GPRS gezählt sind, sind zwar deutliche Verbesserungen bei den Übertragungsraten und neue Abrechnungsmodelle von den Netz-Providern zu erwarten, es ist aber angesichts der enormen Investitionen in diese neue Technologie zumindest zweifelhaft, ob die Kommunikationskosten signifikant sinken. Es ist damit auch langfristig sinnvoll, die Systeme so zu gestalten, dass nur minimale Datenmengen über Mobilfunknetze übertragen werden.
- *Integration der mobile Devices:* Das Beispielszenario zeigt, dass verschiedene Devices relevante Informationen für die Nutzung von Diensten beisteuern können (z. B. der Bordcomputer die Reichweite, das Navigationssystem die GPS-Koordinate und der PDA einen Text). Weiterhin können Ortungsinformationen präzisiert werden, wenn GPS- und GSM-Ortungen kombiniert werden.
- *Einfache Integration vorhandener Internet-Dienste:* Eine Vielzahl von für LBS nutzbaren Services ist im Internet bereits verfügbar. Beispiele hierfür sind HRS (Hotel Reservation Service), Clever Tanken oder Verkehrsinformationen der verschiedenen im Internet vertretenen Radiosender. Es ist daher wenig sinnvoll neue Services zu entwickeln, statt dessen müssen vorhandene Dienste mit geringem Aufwand integrierbar sein. Dies muss mittels einfacher Schnittstellen sowohl für komplexe Such- und Reservierungssysteme von Meta-Service-Providern wie auch einfachen Internet-Sites von Content-Providern möglich sein.
- *Hochverfügbarkeit der Dienste auch bei hohen Lasten:* Eine Plattform für LBS muss grundsätzlich für die Bedienung eines Massenmarktes ausgerichtet sein, da jeder Nutzer von Mobiltelefonen, PDAs oder Navigationssystemen ein potenzieller Nutzer von LBS ist. Location Based Services müssen zu jeder Tages- und Nachtzeit zur Verfügung stehen. Da sie insbesondere in nicht-vorhersehbaren Situationen eingesetzt werden, ist eine Verfügbarkeit nahe 100% gefordert.

- *Skalierbarkeit und Offenheit:* Die Server-Infrastruktur muss leicht ausbaubar sein, damit Anbieter wie Autohändler oder Hotels kundenbezogene Services auch auf eigenen Servern in Eigenregie anbieten können.
- *Niedrige Kosten:* Während die Zielgruppe der Automobilhersteller (noch) eher die Kunden in oberen Marktsegmenten sind, ist ein Erschließen des Massenmarkts sinnvoll und möglich. Daher muss eine Plattform von vorne herein so gestaltet werden, dass die Services kostengünstig angeboten werden können. Dies gilt sowohl für den Betrieb der Server-Infrastruktur wie auch für die Nutzung des Systems.

4 Stand der Technik

Die Grundidee, mit LBS mobile Anwendungssysteme für Autofahrer zu entwickeln ist prinzipiell nicht neu. Exemplarisch werden hier die Konzepte der Projekte DRiVE und MCP kurz vorgestellt.

4.1 DRiVE

Ziel des DRiVE (Dynamic Radio for IP-Services in Vehicular Environments) (<http://www.ist-drive.org/>) Projekts ist es, eine Verbindung zwischen (Mobil-)Funknetzen und Broadcast Netzwerken zu schaffen, um eine möglichst kosteneffiziente und hochverfügbare Basis der IP Übertragung zu realisieren. Dabei werden die Funkspektren der Übertragungstechniken (bspw. GSM, GPRS, UMTS, DAB, DVB-T) dynamisch allkiiert, um die Auslastung der knappen Ressource Bandbreite zu optimieren. Applikationen und Netzwerkelemente sollten auf einander abgestimmt sein und sich verschiedenen Situationen anpassen können. Die geplanten Anwendungen reichen von IP-basierten Multimedia Diensten, Informationsrecherchen über das Internet, Bildungsplattformen bis hin zu Entertainment Diensten im automobilen Umfeld.

Aus dem DRiVE Projekt heraus ist in Zusammenarbeit mit DaimlerChrysler ein Prototyp entstanden, der die beabsichtigten Ziele umgesetzt hat. Der Benutzer kann multimediale Informationen über den Bordcomputer aus dem Internet abrufen. Dabei wird die optimale Bandbreite, sowie das Spektrum dynamisch vom System gewählt. Die Komponenten von Drive wurden in Java™ entwickelt. Die Meilensteine des Projekts sind Folgende:

- Spezifikation für die spektrumseffiziente Kooperation von Funk- und Broadcast Netzwerken
- IP-basierte mobile Infrastruktur für optimierte Zusammenarbeit zwischen Funknetzen, die multimediale Dienste in Autos bereitstellt
- Anpassbare Services für Multi-Funksysteme
- Prototyp
- Standardisierungen und wissenschaftliche Beiträge

DRiVE ist damit stark basistechnologisch orientiert, da die Forschungsarbeiten in erster Linie die Schaffung einer mobilen IP-Infrastruktur in Mobilfunknetzen betrifft.

4.2 MCP – Multimedia Car Platform

Broadcasting-Dienste und mobile Dienste sollen in der Multimedia Car Platform (MCP) integriert werden (<http://mcp.fantastic.ch>). Das Ziel ist es einen transparenten Zugang zu multimedialen Diensten auf Basis existierender und neuer Funktopologien, wie zum Beispiel GSM/GPRS, DAB, DVB-T und UMTS zu bieten, um den speziellen Bedürfnissen in der Automotive-Umgebung gerecht zu werden. Die Interoperabilität zwischen Broadcasting und mobilen Kommunikationsnetzwerken eröffnet dabei neue Möglichkeiten für Anwender.

Das erste Modell eines mobilen Multimediaterminals ermöglicht es den Nutzern aus einer Auswahl von Applikationen und Daten aus verschiedenen Netzwerken und Diensten auszuwählen. Die Multimedia Car Plattform verwaltet die verschiedenen Netzwerkzugänge, das Terminal und die Benutzerpräferenzen in Hinblick auf die Dauer, Qualität und die Art der Informationsbeschaffung. Hauptziele des Projekts sind:

- Entwurf von innovativen Multimedia Diensten inkl. der Anwendungsfälle
- Spezifizierung und Implementation der Architektur für interoperable hybride Netzwerke
- Komplette technische Beschreibung anbieten, um das System transparent zu gestalten
- Integration von Kommunikation, Entertainment und Navigation in der MCP
- Aufbau eines Prototypen
- Die Entwicklung und Konvergenz zwischen Broadcasting und mobiler Kommunikationsnetzwerken schaffen

Meilensteine des Projekts sind:

- Definition kommerzieller und technischer Anforderungen an MCP bzgl. Diensten und Netzwerkarchitektur
- Spezifikation eines MCP-Terminals
- Implementierung der Dienste und der Architektur
- Aufbau des Hardwaredemonstrators für MCP

Das Projekt wurde erfolgreich abgeschlossen. Eine Implementierung in Java mit einer offenen API-Schnittstelle liegt vor. Dieses Projekt ist sowohl basistechnologisch (Netzwerkarchitektur) wie auch dienstorientiert ausgelegt.

4.3 Daypath

Auf der CeBIT 2001 stellte SUN Microsystems mit seinen Partnern Jentro und Siemens das „Daypath“ Projekt vor. Hierbei wurden mit der offenen SUN ONE Architektur (intelligente) Webservices implementiert, die geräteunabhängig gesteuert werden können. Über eine Mobilfunkverbindung kann sich der Benutzer navigieren lassen, Informationen aus dem Internet abrufen, oder auf das Intranet seiner Firma zugreifen. Kernpunkt des Projektes ist die Kopplung elementarer Funktionen (Navigation, Hotelsuche, Terminplanung) zu einem neuen Dienst. Das Projekt wurde mit einem Prototypen abgeschlossen, der die SUN ONE Architektur demonstriert.

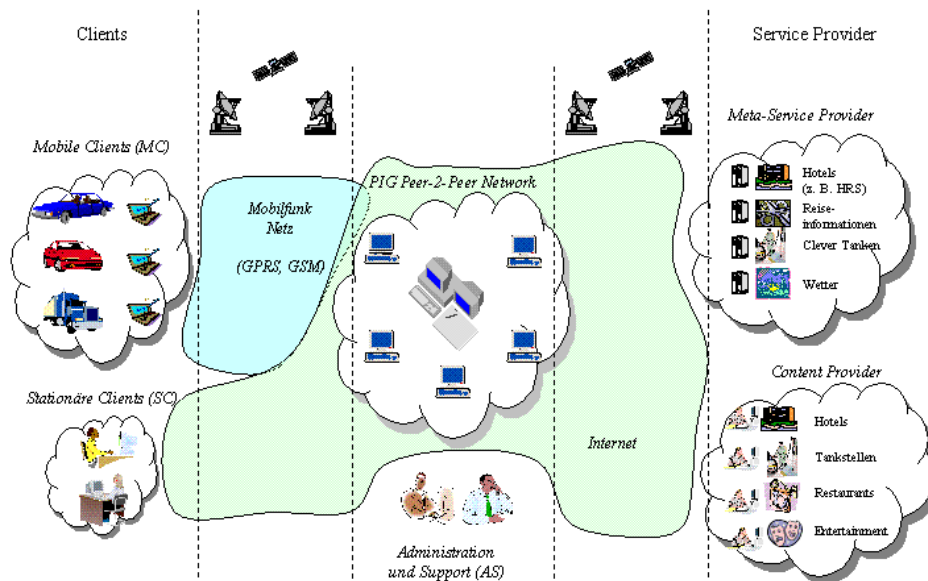


Abbildung 1: Szenario

5 Projekt PIG

Zielsetzung des Projekts PIG (Personal Information Guide) ist die Schaffung einer allgemeinen und nicht-proprietären Server-Plattform für LBS auf Basis bestehender Internet-Dienste, welche die weiter oben genannten Anforderungen so weitgehend wie möglich erfüllt, und die prototypische Entwicklung eines mobile Clients. Damit liegt der Schwerpunkt auf der Integration vorhandener Internet-Dienste (und nicht spezifischer Dienste wie bei MCP) und die Schaffung einer Server-Infrastruktur, die auf die in MCP und DRiVE entwickelten Basistechnologien aufsetzen kann. Das Gesamtszenario ist in Abbildung 1 dargestellt. Der mobile Benutzer verfügt auf seinem Endgerät über eine Software, dem PIG-Client. PIG ermöglicht die Abfrage mobiler Dienste auf Basis der Ortung und ggf. weiterer Informationen sowie die Übernahme von Rechercheergebnissen. Die Anfrage erreicht über ein Mobilfunknetz die Server-Infrastruktur, auf der die Plattform für LBS im

plementiert und in einem Peer-to-Peer Netzwerk verteilt ist. Diese greift via Internet auf die angebotenen Dienste der Meta-Service- und Content-Provider zurück. Im Folgenden die Architektur von PIG genauer beschrieben.

6 PIG Clients

Die PIG Clients sind die Benutzerschnittstelle des PIG Systems. Sie basieren auf einer dreistufigen Softwarearchitektur. Die oberste Schicht der PIG-Architektur bildet die Präsentationsschicht mit der Benutzeroberfläche, die vollständig mit HTML implementiert ist. Hier werden die einzelnen Oberflächenkomponenten, zur Nutzung der Dienste bereitgestellt. In der Anwendungsschicht werden die zur Dienstnutzung notwendigen Funktionen gekapselt. Diese setzen auf Hardware und Betriebssystem (Schicht 3) des Client-Rechners auf. Stationäre und mobile Clients unterscheiden sich durch ihre Aufgabenstellung, allerdings ist diese Trennung aus technischer Sicht künstlich. Sofern entsprechende Ressourcen verfügbar sind, kann der mobile Client alle Funktionen der stationären Clients übernehmen und umgekehrt. Abbildung 2 zeigt die Architektur des PIG-Clients im Überblick.

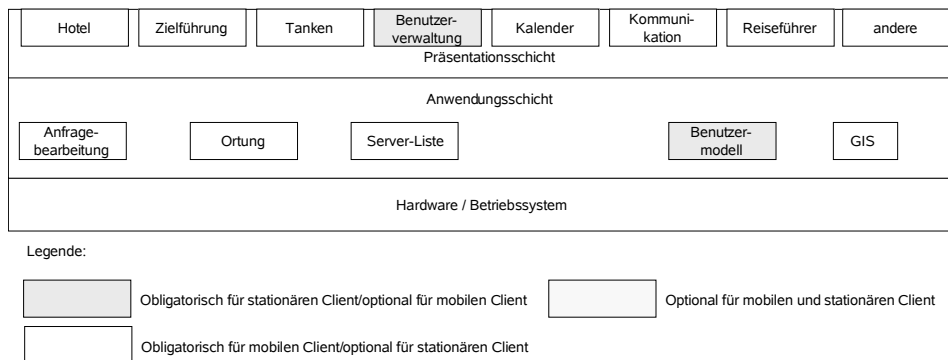


Abbildung 2: PIG-Client-Architektur

6.1 Architektur des stationären PIG Clients

Der stationäre PIG-Client dient der Einrichtung und Verwaltung der Benutzereinstellungen. Hierbei kann der Benutzer PIG-Dienste abonnieren und seine dienstbezogenen Benutzerrollen einrichten. Als *PIG-Dienst* wird hier ein Dienst bezeichnet, der von den PIG-Servern zur Verfügung gestellt wird, wie z. B. Hotelsuche, Tankstellenrecherche usw. Die Internet-Angebote (für den PIG Dienst Hotelsuche z. B. hrs.de oder varta.de), die von einem PIG-Dienst ausgewertet werden, werden *Web-Dienste* genannt.

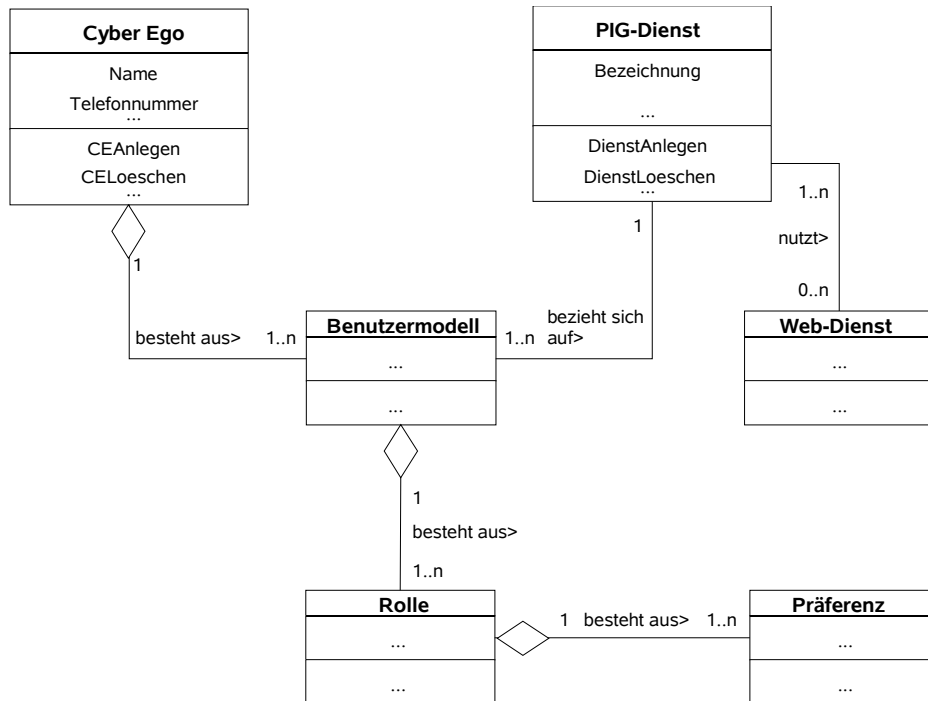


Abbildung 3: Cyber-Ego

Auf der Präsentationsschicht werden die notwendigen Funktionen als HTML-Anwendung zur Verfügung gestellt. Dabei wählt der Benutzer zunächst die PIG-Dienste aus, die er mobil nutzen will. Für jeden Dienst wird ein Benutzermodell verwaltet, in dem die dienstspezifischen Benutzerpräferenzen spezifiziert sind. Solche Präferenzen sind bezogen auf den Dienst „Hotelsuche“ beispielsweise Hotelkategorie, Preisobergrenze, Lage, Zimmerausstattung etc. Da nicht immer davon ausgegangen werden kann, dass ein genau den Präferenzen entsprechendes Hotel gefunden wird, können die Präferenzen priorisiert werden. Die Präferenzen zu einem Dienst werden in einer Rolle zusammengefasst. Dabei kann ein Benutzer zu einem Dienst mehrere Rollen anlegen, da die Präferenzen eines Benutzers abhängig davon, ob er z. B. privat oder dienstlich reist, unterschiedlich sein können. Für jeden Dienst muss eine Rolle als gültig voreingestellt werden. Alle einem Dienst zugeordneten Rollen bilden ein Benutzermodell und die Benutzermodelle über alle Dienste das Cyber-Ego des Benutzers. Die Benutzermodelle werden auf mindestens zwei PIG Servern abgespeichert, um sicherzustellen, dass bei Ausfall eines Servers ein anderer als Dienstanbieter einspringen kann. Alternativ kann ein Cyber-Ego auf dem Client abgelegt werden, um es bei Bedarf einem Agenten zuzuordnen. Die Zusammenhänge zwischen Cyber-Egos und Diensten sind in Abbildung 3 als UML-Klassendiagramm skizziert.

Da der stationäre Client eine reine HTML-Anwendung ist, kann jedes Betriebssystem und jede Hardware, auf dem ein Internet-Browser verfügbar ist, genutzt werden.

6.2 Architektur des mobilen PIG Clients

Die Benutzeroberfläche des mobilen Clients stellt die mit den stationären Client ausgewählten Dienste zur Verfügung. Auf der Anwendungsschicht sind dann folgende Funktionen realisiert:

- *Ortung*: Die Ortung erfolgt über den mobile Client über GPS oder GSM. Die Information wird in Form einer geographischen Koordinate übertragen. Verfügt der mobile Client über ein lokales GIS (Geographisches Informationssystem) wie es insbesondere bei Navigationssystemen der Fall ist, kann der Standort auch als Adresse übermittelt werden.
- *Server-Liste*: Der mobile Client verfügt über eine Liste der Server, auf den sein Cyber-Ego gespeichert ist. Die Liste wird bei der Einwahl solange abgearbeitet, bis sich ein verfügbarer Server meldet.
- *Benutzermodellverwaltung (optional)*: Die Benutzermodelle können auch auf einem mobilen Client installiert sein. Die Verwaltung auf einem mobilen Client erlaubt die Offline-Änderung des Benutzermodells, das jederzeit – auch im mobilen Betrieb – mit der Server-Version abgeglichen werden kann. Das geänderte Benutzermodell wird dann an den ersten in der Server-Liste aufgeführten Netzwerkknoten übertragen und von dort an die anderen Knoten der Liste per Snapshot-Verfahren weiterverteilt. Das aktuelle Benutzermodell wird dabei anhand des Zeitstempels identifiziert. Voraussetzung für die Bearbeitung von Benutzermodellen im mobilen Client ist, dass dieser hierfür hinreichend Speicherkapazität hat und ein Internet-Browser hierauf lauffähig ist. Ist dies nicht der Fall, reicht es aus, wenn sich der Benutzer durch seine persönliche Authentifikation (in der Regel seine Mobilfunknummer), den gewählten Dienst und die hierfür die gültige Rolle gegenüber den PIG Servern ausweist.
- *Anfragebearbeitung*: Beim Anfragevorgang wird das PIG Server Netzwerk zunächst über ein Mobilfunknetz ausgewählt. Danach wird die Anfrage unter Angabe des nachgefragten Dienstes, die ID der relevanten Rolle (die Rollen selbst sind auf dem Server bekannt) und die ggf. je nach Dienst noch manuell eingegebenen Eingabeparameter. Die Anfragen werden vollständig auf dem Server bearbeitet. Beim Datenempfang sind die Trefferlisten zu übernehmen. Zwischen Anfrage und Empfang wird die Verbindung physisch unterbrochen, um die kostspielige Airtime so gering wie möglich zu halten.
- *GIS (optional)*: Zu den optionalen Funktionen gehört ein GIS, das die Funktion einer elektronischen Landkarte wahrnimmt und mit dessen Hilfe als GPS-Koordinaten geographische Informationen ermittelt werden können. Derartige Systeme befinden sich z. B. auf den CDs der Navigationssysteme, können aber auch per Internet abgerufen werden (z. B. unter <http://www.tegaron.de>).

Der heute verfügbare Prototyp des PIG mobile Client ist unter Windows CE, Windows 2000, Windows 95/98, Linux, UNIX und MacOS verfügbar. Mit dem Compaq iPAQ und dem HP Jornada wurden mobile Endgeräte getestet. Der nächste Schritt ist die Portierung auf einen Car-PC. Ein Car-PC ist ein vollständiger PC im Format eines Autoradios mit integriertem Navigationssystem, Digitalradio, MP3-Player und Anschluss an den Bordcomputer. Implementierungen auf Basis von Navigationssystemen oder Bordcomputern

liegen noch nicht vor, da dies geschlossene Systeme sind, an die man ohne Herstellerunterstützung nicht herankommt.

7 PIG-Server-Netzwerk

Das PIG-Server-Netzwerk basiert auf Agenten-, XML- und Peer-to-Peer- (P2P)-Technologie. Im Folgenden werden die technischen Konzepte von PIG vorgestellt.

7.1 Agentenkonzept

Gemäß der Klassifikation von [NwNd99] lassen sich die auf den Servern implementierten Agenten als stationär und kollaborativ klassifizieren. Sie sind stationär, da sie stets auf den Servern verweilen. Kollaborative Agenten dienen der Agent-to-Agent (A2A-) Kommunikation. Ein solcher Agent bekommt von einem Benutzer oder anderem Agenten eine Aufgabe und bearbeitet sie selbstständig. Hierfür muss er in der Lage sein, die eigenen Aktivitäten zu überwachen und Abläufe zu steuern. Weiterhin muss er wissen, welche anderen Agenten (Teil-)Aufgaben lösen können und diese in den Lösungsprozess einbinden. Da im Falle des PIG die Arbeitsabläufe feststehen, kann hier auf Lernfähigkeit und KI-Verfahren zur Ablaufkontrolle verzichtet werden.

Für die Kommunikation zwischen den Agenten ist ein Protokoll implementiert, das sich an der Knowledge Query and Manipulation Language (KQML) [FFMM94] orientiert. Der Nachfrager eröffnet die Kommunikation mit einem Anfrage (request) an den (Leistungs-)Anbieter. Der Anbieter prüft den nachgefragten Servicetypen, sowie die Identität des Nachfragers. Bei positiver Bewertung beider Sachverhalte schickt der Anbieter ein Angebot an den Nachfrager (offer), bei negativer Bewertung lehnt er die Kommunikation ab (reset). Im Falle eines Angebotes kann der Nachfrager auf das Angebot mit einer neuen Anfrage reagieren, eine Buchung auslösen (booking) oder den Vorgang bei Desinteresse abbrechen (cancel). Stimmt die Buchungsnachricht mit dem Angebot überein, wird eine Buchungsbestätigung vom Anbieter zum Nachfrager gesendet (ack), ansonsten wird eine Fehlermeldung zum Nachfrager verschickt (error). Tritt der Nachfrager von einer Buchung zurück, sendet er eine Stornierungsnachricht (cancel). Abbildung 4 zeigt das Kommunikationsprotokoll für Agenten als UML-Aktivitätsdiagramm (N=Nachfrager, A=Anbieter).

Jeder Benutzer wird auf einem PIG-Server durch einen Agent repräsentiert. Dieser *Benutzeragent* übernimmt das Cyber-Ego des jeweiligen Benutzers und aktiviert die für den vom Benutzer ausgewählten Dienst gültige Rolle. Weiterhin übernimmt er ggf. Ortungsinformation und die vom Benutzer übermittelten Parameter.

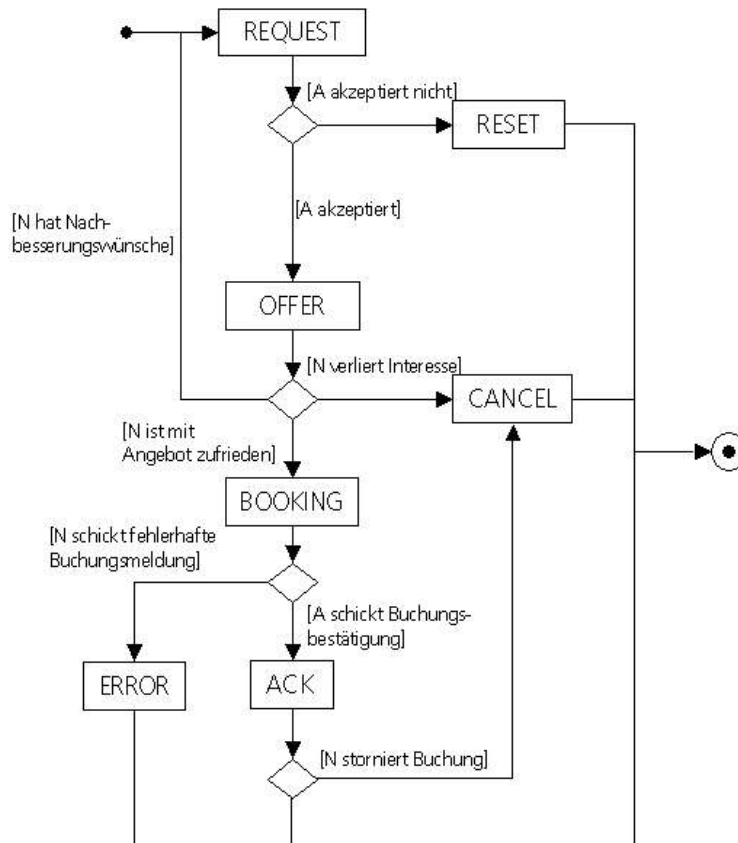


Abbildung 4: Kommunikation zwischen Agenten

Ein Knoten des PIG-Server-Netzwerks wird Host genannt. Auf jedem Host gibt es genau einen *Host-Agenten*, der zwischen Benutzeragenten und Dienstagenten (Provider) vermittelt. *Provider* sind die Akteure beim Zugriff auf Internet Dienste. Der Host-Agent besitzt damit allein das gesamte Wissen über die lokalen Gegebenheiten auf seinem Host, damit Benutzeragenten und Provider auf einfache Weise im Netz verteilt werden können. Weiterhin kapselt der Host die Benutzeragenten, in den viel für Außenstehende interessantes Wissen über den Benutzer gespeichert ist, gegenüber der Außenwelt, so dass die Privatsphäre des Benutzers gewahrt bleibt.

Provider und Benutzeragenten können redundant auf die Server im Netzwerk verteilt werden. Welche Agenten auf welchen Servern residieren, lässt sich anhand so genannter *Kontaktlisten*, die repliziert auf allen Knoten geführt werden, nachvollziehen. Wird ein Agent auf einem Host eingerichtet, dann nimmt dieser mit seinem Registration Service selbstständig den entsprechenden Eintrag in allen Kontaktlisten des Netzwerks vor. Die Konsistenz der Kontaktlisten wird über ein Snapshot-Verfahren sichergestellt. Wird z. B. ein Agent von einem Knoten entfernt, wird ein mit Zeitstempel versehener Löschermerk in die lokale Kontaktliste eingetragen. Diese Änderung wird dann so lange an die anderen

Server propagiert, bis alle Kontaktlisten auf dem gleichen Stand sind. Die Kontaktliste für Cyber-Egos besteht aus einer Tabelle mit den Spalten Name des Agenten (korrespondiert mit dem Benutzernamen), Telefonnummer (die hier zur Authentifizierung verwendet wird) und dem Servernamen, auf dem das Cyber-Ego angelegt ist. In der Kontaktliste für Dienstagenten sind der Name des Agenten, der Name des PIG-Dienstes, der Name des aus dem Internet nachzufragenden Dienstes und der Servername, auf dem der Agent residiert, gespeichert. Abbildung 5 zeigt Beispiele für Kontaktlisten. Das Beispiel der Kontaktliste „Dienste“ zeigt, dass ein Provider auch verschiedene PIG-Dienste bedienen kann. Dies ist sinnvoll, wenn die verschiedenen Dienste bezüglich ihrer Zugriffsmethoden so ähnlich sind, dass eine Aufteilung in verschiedene Agenten ineffizient ist oder ein Anbieter mehrere Dienste offerieren möchte.

Kontaktliste Benutzeragenten			Kontaktliste Dienste			
Agenten-name	Telefon-nummer	Server	Agenten-name	PIG-Dienst	Dienstanbieter	Server
Rauten	0172123456	piggy	Varta	Hotelsuche	Varta.de	piggy
Rauten	0172123456	statler	HRS	Hotelsuche	HRS.de	Statler
20er	0172654321	piggy	Orakel	Verkehr	MDR.de	Statler
20er	0172654321	Waldorf	Orakel	Tanken	Clever-Tanken.de	Statler
...			...			

Abbildung 5: Kontaktlisten

Die Vermittlung zwischen Benutzeragent und Provider erfolgt über den Routing-Service des Host-Agenten. Dieser übernimmt die Anfrage des Benutzeragenten als XML-Dokument, wobei für jeden Dienst eine eigene Document Type Definition (DTD) bzw. ein XML-Schema definiert ist, und ermittelt mithilfe des Resolver-Service, der die Kontaktlisten aus dem Directory-Service auswertet, die zutreffenden Provider. Die ausgewählten Provider erhalten alle das XML-Dokument mit der Suchanfrage. Die Suchergebnisse werden wieder als XML-Dokument an den Routing-Service des Host-Agenten zurückgemeldet und von dort an den nachfragenden Benutzeragenten weitergegeben. Dieser bereitet ggf. das Suchergebnis auf (z. B. wenn aus mehreren Treffern der Beste herausselektiert werden muss) und sendet es an den nachfragenden PIG-Client zurück. Das Konzept der Routing- und Resolver-Services wurde aus der Architektur des JXTA-Search-Hubs entliehen [Wate01].

Die Web-Dienste, die für die PIG-Dienste angezapft werden, haben zumeist dynamisch generierte Web-Sites. Für die Kommunikation zwischen Providern und den Anbietern der Web-Dienste gibt es prinzipiell zwei Möglichkeiten:

- Der Datenaustausch erfolgt über ein eigenes PIG-Protokoll, das über eigene Agenten auf beiden Seiten realisiert wird, die XML-Dokumente austauschen.
- Der Provider analysiert und parst die HTML-Seiten des Web-Dienstes (diese Methode wird HTML-Scraping genannt) und wandelt die Ergebnisse in das PIG-XML-Dokumentformat um.

Die erste Methode ist die elegantere, da der Datenaustausch zwischen Provider und Web-Dienst über eine standardisierte Schnittstelle abläuft. Weiterhin kann der Anbieter des Web-Dienstes seine Seiten verändern, ohne dass dies Auswirkungen auf die Implementierung des Providers hat. Im vorliegenden PIG-Prototyp wird allerdings HTML-Scraping

verwendet, da hier keinerlei Eingriffe auf der Seite des Web-Dienstes notwendig sind. Allerdings muss der Code des Providers geändert werden, sobald sich die HTML-Implementierung beim Web-Dienstanbieter ändert. Außerdem lassen diese Art des Zugriffs nicht alle Anbieter zu.

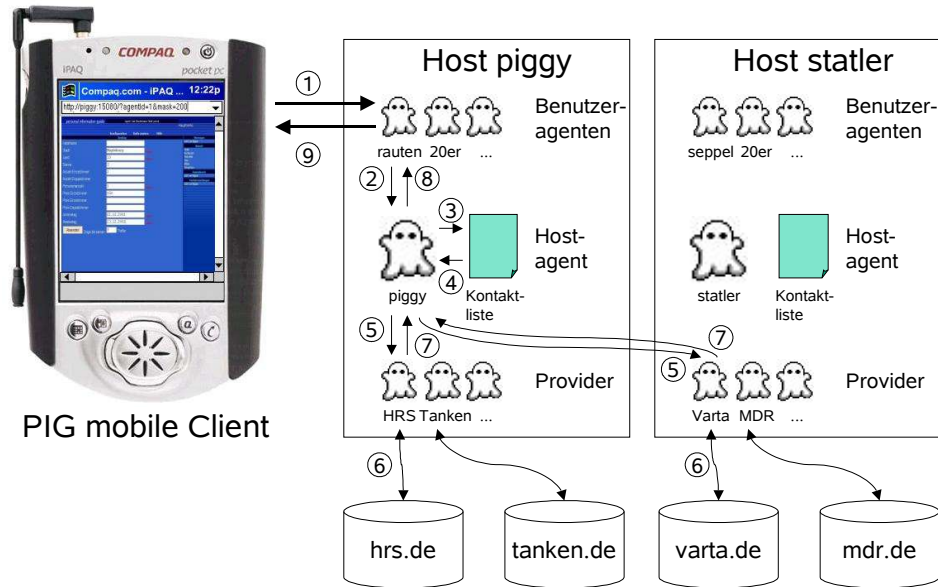


Abbildung 6: PIG Kommunikationsfluss

Abbildung 6 fasst die Kommunikation im PIG-Netzwerk anhand eines einfachen Beispiels zusammen:

1. Der Benutzer selektiert mit seinem mobilen Endgerät einen Dienst (in diesem Fall Hotelsuche), wählt sich auf dem Server piggy ein und übermittelt ein XML-Dokument mit der Dienstanfrage und den Suchparametern. Danach wird die Telefonverbindung beendet, wenn keine permanente Internetverbindung besteht.
2. Der Kundenagent rauten übernimmt die Anfrage und kontaktiert den Host-Agenten piggy über dessen Routing Service.
3. Der Host-Agent sucht dann mit seinem Resolver-Service in der Kontaktliste (Directory-Service) nach Providern für den PIG-Dienst Hotelsuche.
4. Er findet die Provider hrs auf Server piggy und varta auf Server statler.
5. Die gefundenen Provider werden mit der Suche über den Routing-Service beauftragt, indem Ihnen das XML-Dokument mit der Benutzeranfrage übermittelt wird.
6. Die Provider recherchieren auf den Web-Diensten
7. Die Rechercheergebnisse werden in ein XML-Dokument verpackt und an den Routing-Service des Host-Agenten zurückgemeldet.
8. Der Host-Agent reicht das Dokument über den Routing-Service an den Benutzeragenten weiter.
9. Der Benutzeragent bereitet das Ergebnis ggf. auf, ruft den PIG-Client an und sendet das Ergebnis an den PIG-Client zurück.

7.2 Das PIG-P2P-Netzwerk

Das PIG-Netzwerk basiert auf der pure P2P-Architektur, d. h. es existieren zur Laufzeit weder Software- noch Hardware-seitig zentrale Instanzen [MiHe01]. Die Server sind direkt miteinander über P2P-Verbindungen miteinander verbunden. Pure P2P-Netzwerke sind durch folgende Eigenschaften charakterisiert [P2PW02; Jata029]:

- Alle Netzknoten sind gleichberechtigt.
- Jeder Netzknoten kann die Initiative zur Kommunikation ergreifen.
- Die Netzknoten kommunizieren direkt miteinander.
- Zugriffskontrolle und Berechtigungen werden im P2P-Netz verwaltet.
- Das Internet oder Intranet dient lediglich als Übertragungsmedium.
- Die am P2P-Netz beteiligten Rechner können sich zu Rechnerfarmen zusammenschließen, um die Leistung der Einzelrechner zu einem leistungsstarken virtuellen Gemeinschaftssystem zu verbinden. Damit hat jeder Netzknoten auch die Möglichkeit, Dienste aller anderen Rechner anzubieten.

Die Peer-to-Peer Working Group hat in einem White Paper die Taxonomie für P2P-Architekturen zusammengefasst [NVSB01]. Folgende grundlegende Eigenschaften zeichnen demnach eine P2P-Architektur aus:

- Die *Identität* bezieht sich auf den Namen oder die Legitimation einer Entität im P2P-Netzwerk. Im Fall von PIG sind die Entitäten die Agenten.
- *Discovery* bedeutet, dass man nach Maschinen, Diensten, Ressourcen bzw. im PIG-System Agenten suchen oder feststellen kann, ob diese online sind oder nicht.
- *Authentifizierung* beschreibt den Prozess zur sicheren Identitätsprüfung einer Entität.
- *Autorisierung* bestimmt, ob eine Entität mit einer gegebenen Identität Zugriff auf bestimmte Ressourcen hat oder bestimmte Funktionen ausführen darf.
- Die *Funktionalität* ist die Zusammenfassung aller anwendungsspezifischen Operationen, die außerhalb der oben genannten Grundeigenschaften auftreten.

Die Identität der PIG-Agenten wird über eine eindeutige Agenten-ID gesichert. Die in Abbildung 6 verwendeten Agentennamen sind Aliasse. Da eine zentrale Vergabe von IDs auch eine zentrale Vergabeinstanz voraussetzen würde, was der Philosophie des P2P-Netzes entgegensteht, gibt es zwei Varianten zur lokalen Vergabe eindeutiger IDs:

- Die Agenten werden lokal durchnummeriert. Die ID ergibt sich dann aus der Konkatenation der IP-Adresse und der lokal eindeutigen Nummer. Das Verfahren funktioniert nur, wenn die IP-Adressen statisch zugewiesen sind, was jedoch nicht immer gegeben ist.
- Bei der zweiten Variante wird anstelle der IP-Adresse der Name des Rechners mit einem voll qualifizierten Domänenname (Fully Qualifying Domain Name – FQDN) [Kirc96, 30] und Port sowie Portnummer konkateniert, wie z. B. piggy.iff.fhg.de:port:nummer. Diese Variante ist etwas lesbarer als die erste und wird daher bei PIG verwendet.

Mit den Kontaktlisten und dem Registration Service sind die Voraussetzungen dafür geschaffen, dass alle Agenten netzweit bekannt sind und gefunden werden können. Für die Authentifizierung eines Benutzers kann dessen Mobilfunknummer, ein Benutzername und das zugehörige Passwort, sowie das Public-Key Verfahren (Digitale Signatur) verwendet werden. Damit hat er die Verantwortung für Korrektheit der Authentifizierung und deren Schutz im wahrsten Sinne des Wortes in der Hand. Autorisierungsmechanismen sind Gegenstand zukünftiger Weiterentwicklungen. Die Funktionalität von PIG wurde bereits oben beschrieben.

8 Geschäftsmodell

Für den kommerziellen Einsatz von PIG ist ein Geschäftsmodell erforderlich, das angesichts der potenziellen Vielzahl von Beteiligten am in Abbildung 6 dargestellten Geschäftsprozess einerseits den Belangen der einzelnen Partner gerecht, andererseits aber auch nicht zu kompliziert wird. Abbildung 7 zeigt die möglichen Zahlungsflüsse in Pfeilrichtung zwischen den potenziell beteiligten Geschäftspartnern [Raut02]. Der Nutzer zahlt dabei für die Ermittlung seiner Position (wobei momentan GPS-Ortungen kostenlos sind), für die Mobilfunkverbindung und die abonnierten PIG-Dienste. Die prinzipiell gerechteste Lösung für die Nutzung von PIG-Diensten ist die Zahlung pro Dienstnutzung. Dies erfordert jedoch die Integration eines aufwändigen Billing- oder Micropayment-Systems, was einen Overhead verursacht, der die Handhabbarkeit sowohl für Nutzer wie auch Betreiber einschränkt. Daher ist hier eine pauschale Abrechnung z. B. in Form eines Monatsabonnements pro Dienst sinnvoller.

Bei der Beziehung zwischen PIG- und Web-Diensteanbieter sind Zahlungsflüsse in beide Richtungen denkbar. Hier hängt es davon ab, auf welcher Seite das Geschäftsinteresse zur Beteiligung an PIG-Diensten größer ist. Beispielsweise kann ein Anbieter von Hotelreservierungen mit der Verfügbarkeit seines Dienstes im PIG-System neue Kundenkreise erschließen und sein Geschäftsfeld erweitern. Er hat damit ein vitales Interesse an der Bereitstellung seines Dienstes in PIG, so dass eine Zahlungsbereitschaft unterstellt werden kann. Umgekehrt kann die Bereitstellung eines kostenlosen Dienstes in PIG wie z. B. CleverTanken für den PIG-Betreiber eine höhere Relevanz als für den Web-Diensteanbieter haben. Damit kehrt sich auch die Zahlungsbereitschaft um. Weiterhin zahlt der PIG-Betreiber die Mobilfunkgebühren für die Anrufe zur Übermittlung der Abfrageergebnisse. Bei den Zahlungsflüssen mit den Werbepartnern zeigt sich ein spezielles Detailproblem: Werbebanner, die auf den Seiten von Web-Diensteanbietern erscheinen und für diese eine wesentliche Einnahmequelle sind, werden nicht zu den PIG-Clients durchgereicht. Dieses „Ausblenden“ von Werbung widerspricht oftmals den Verträgen zwischen Web-Diensteanbietern und ihren Werbepartnern. Hierfür sind dann Ausgleichszahlungen zu leisten.

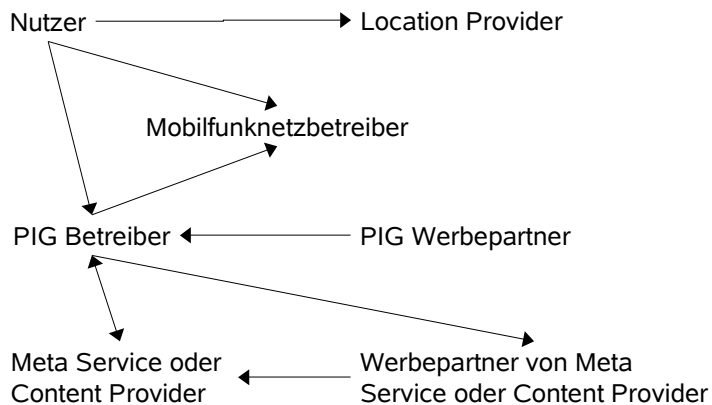


Abbildung 7: Zahlungsflüsse im PIG-Geschäftsprozess

9 Zusammenfassung und Ausblick

Mit PIG ist ein System geschaffen worden, das konventionelle Internet-Dienste für mobile Anwendungen verfügbar macht. Die konsequente Nutzung von Agenten- und P2P-Technologie sorgt dafür, dass das System hochgradig flexibel bezogen auf die Einbindung von Diensten, die Skalierbarkeit der Server-Infrastruktur und die Anpassbarkeit an individuelle Benutzeranforderungen ist. Weiterhin ist eine hohe Verfügbarkeit der Server realisierbar, auch wenn die einzelnen Server nicht redundant ausgelegt sind bzw. über nur über konventionelle Plattensysteme verfügen. Damit können kostengünstige Rechner als Server – im der Beispielkonfiguration des Fraunhofer IFF PCs mit Linux-Betriebssystem – eingesetzt werden. Dies geht so weit, dass als PIG-Server normale Arbeitsplatz-PCs genutzt werden können, bei denen die Server-Dienste im Hintergrund laufen. Selbst kleine Unternehmen sind damit in der Lage, PIG-Server zu betreiben und eigene Dienste anzubieten.

PIG ist heute ein vollumfänglich einsetzbarer Prototyp, der von Mitarbeitern des Fraunhofer IFF im Pilotbetrieb genutzt wird. Die Weiterentwicklung von PIG betrifft folgende Punkte:

- Die Benutzerschnittstelle wird von HTML auf XML umgestellt. Der Grund hierfür ist, dass eine wesentliche Voraussetzung für die kommerzielle Nutzung des Systems die Bereitstellung einer Benutzeroberfläche im Corporate Design des Anwendungsunternehmens ist. Diese Anforderung lässt sich erfüllen, wenn die jeweiligen Corporate Designs mit XSL-Stylesheets in der Präsentationsschicht implementiert und von der Geschäftslogik getrennt werden.
- Eine bislang nicht umgesetzte Anforderung ist die Integration verschiedener mobile Devices. Der Grund hierfür ist die Geschlossenheit von Bordcomputern und Navigationssystemen. Mit der Portierung des PIG mobile Client auf einen Car-PC, der diese Schnittstellen offen legt, wird diese Integration vollzogen.

- Eine weitere Aufgabe ist die Absicherung der PIG Server gegen Angriffe von außen. Hierfür werden die Host-Agenten um eine Security-Service erweitert, mit dem die Kommunikation zwischen PIG-Servern und Firewall-Systemen möglich wird.
- Der Resolving- und Directory-Service muss um Optimierungsalgorithmen zur Verwaltung der Kontaktlisten erweitert werden. Für die Verbreitung der XML Nachrichten durch den Routing-Service soll das Simple Object Access Protocol (SOAP) genutzt werden.

Literatur

- [FFMM94] *Finin, T., Fritzon, R., McKay, D., McEntire, R* KQML as an Agent Communication Language. In: Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM '94). ACM Press, Gaithersburg (Maryland) 1994, pp. 456-463.
- [GeGr01] *Geer, R., Gross, R.*: M-Commerce – Geschäftsmodelle für das mobile Internet. Vmi, Bonn 2001.
- [Jata02] *Jatalite Systems*: Definitionen Peer-to-Peer (P2P). <http://www.jatalite.de/html/definitionen/index.html>. Abfrage am 26. April 2002.
- [Kirc96] *Kirch, O.*: LINUX Wegweiser für Netzwerker, Band 1. O'Reilly/Thomson, Bonn u. a. 1996.
- [MiHe01] *Milnar, N., Hedlung, M.*: A Network of Peers: Peer-to-Peer Models Through the History of the Internet. In: Oram, A. (ed): Harnessing the Power of Disruptive Technologies. O'Reilly, Sebastopol 2001, pp. 341-353.
- [MiSc01] *Michelsen, D., Schaale, A.*: Handy Business: M-Commerce als Massenmarkt. Financial Times Prentice Hall, Upper Saddle River 2001.
- [NVSB01] *Ngo, T., Venkat, J., Stolarz, D., Barkai, D.*: Taxonomy of Peer-to-Peer Architectures. Technischer Bericht, Peer-to-Peer Working Group 2001.
- [NiPe01] *Nicolai, A. T., Petersmann, T.*: Strategien im M-Commerce. Schäffer, Stuttgart 2001.
- [NwNd99] *Nwana, H. S., Ndumu, D. T.*: A Perspective on Software Agents Research. In: The Knowledge Engineering Review 14 (1999) 2, pp. 1-18.
- [P2PW02] *Peer-to-Peer Working Group*. What is Peer-to-Peer? <http://www.p2pwg.org/whatis/index.html>, Abfrage am 26. April 2002.
- [Raut02] *Rautenstrauch, C.*: New Business Models for Electronic Commerce. Keynote auf der III. Conferencia Internacional de Ciencias Empresariales (Proceedings auf CD), Santa Clara (Kuba) 2002.
- [Wate01] *Waterhouse, S.*: JXTA Search: Distributed Search for Distributed Networks. Technischer Bericht, SUN Microsystems, Inc. 2001.
- [Zobe01] *Zobel, J.*: Mobile Business und M-Commerce. Hanser, München 2001.