Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SEBASTIAN HEIDL, RALF STAUDEMEYER

Implementation von WAP-Diensten für die SGI/CRAY T3E des ZIB

ZIB-Report 00-41 (November 2000)

Studienarbeit

Implementation von WAP-Diensten für die SGI/CRAY T3E des ZIB

vorgelegt von

Sebastian Heidl und Ralf Staudemeyer

am Fachbereich Informatik
der Humboldt-Universität Berlin

betreut durch

Prof. Dr. Alexander Reinefeld
Dr. Jens Simon

November 2000

Danksagung:

Diese Arbeit ist im Rahmen einer Studienarbeit am Fachbereich Informatik der Humboldt-Universität zu Berlin entstanden.
Unser Dank gilt Prof. Dr. Alexander Reinefeld, Leiter des Bereichs Computer Science am Konrad-Zuse-Zentrum für Informationstechnik Berlin, und unserem Betreuer Dr. Jens Simon für die Unterstützung beim Entstehen dieser Arbeit. Desweiteren danken wir Anja Weigel für die tatkräftige Hilfe beim Überarbeiten der im Dokument enthaltenen Abbildungen. Außerdem danken wir Florian Schintke für viele hilfreiche Ratschläge, die dem Vorankommen beim Schreiben dieser Arbeit oft dienlich waren. Nicht zu vergessen danken wir den Entwicklern von TeX und LaTeX, wodurch unsere Arbeit in dieser Form präsentiert werden konnte.

Inhaltsverzeichnis

1 Einleitung

1.1 Aufgabenstellung

Durch die erheblichen Kosten, die bei der Anschaffung und beim Unterhalt von Hochleistungsrechnern entstehen, besteht das Ziel, die Rechenkapazitäten möglichst optimal zu nutzen. Dafür soll den Benutzern des Konrad-Zuse-Zentrums für Informationstechnik Berlin (ZIB[1]) ein individueller Service angeboten werden, der ihnen die Möglichkeit bietet, über ein drahtloses Telefon Informationen über ihre auf dem Hochleistungsrechner SGI/CRAY T3E „`berte`" laufenden Rechenjobs zu beziehen. Weiterhin soll es möglich sein, Ausgaben von abgearbeiteten Prozessen (über von ihnen erzeugte Dateien) zu kontrollieren und neue Jobs zu starten. Für die Realisierung wurde das „Wireless Application Protokoll" (WAP) ausgewählt.

Ziel dieser Arbeit ist es, einen Überblick über den aktuellen Stand der drahtlosen Übertragungstechniken zu geben und eine Beispielanwendung zu entwickeln, die die Möglichkeiten von WAP demonstriert. Dabei soll die Implementation nicht nur auf den Zugriff des aktuellen Hochleistungsrechner beschränkt sein, sondern so flexibel angelegt werden, daß auch später zur Verfügung stehende Systeme problemlos angebunden werden können.

1.2 Warum WAP ?

Kaum ein Unternehmen oder eine Institution kann in der heutigen Zeit seine Aktivitäten auf einen Standort beschränken. Durch die wachsenden Möglichkeiten der Kommunikation verstärkt sich die Konkurrenz nicht nur zu inländischen, sondern auch zu ausländischen Einrichtungen und Unternehmen.
Diese zunehmende Verteilung der Aktivitäten bedingt auch eine steigende Mobilität der Mitarbeiter. Um möglichst effizient zu arbeiten, werden Reisezeiten ebenfalls als Arbeitszeit genutzt. Aus diesem Grund gibt es eine starke Nachfrage nach Möglichkeiten der drahtlosen Kommunikation, um auch im Auto oder Zug erreichbar zu sein und Daten auszutauschen.
Der erste Schritt zur mobilen Kommunikation waren und sind die mobilen Telefone (Handys). Bald reichte die bloße Möglichkeit zu telefonieren nicht mehr aus und auf Basis der GSM-Technologie wurde später der Short Message Service (SMS) etabliert, der es ermöglicht, kurze Nachrichten (160 Zeichen) an andere Teilnehmer zu schicken.
Das Wireless Application Protocol (WAP) beschreibt Möglichkeiten über Verbindungen mit sehr geringer Bandbreite, wie es bei drahtlosen Verbindungen der Fall ist, Daten zu übertragen und auf Geräten mit sehr beschränkten Anzeigekapazitäten auszugeben. Damit wurden die eingeschränkten Möglichkeiten des SMS wesentlich erweitert. Jetzt können auch größere Dokumente für drahtlose Geräte angeboten werden (zu den Einschränkungen der aktuellen Geräte siehe Kapitel 4).

[1] `http://www.zib.de`

2 Die WAP-Architektur

Der Aufbau der WAP-Architektur ähnelt dem ISO/OSI-7-Schichtenmodell sehr stark und basiert auch auf einzelnen Layern. Diese stehen über „Service Access Points“ (SAP) miteinander in Verbindung. Die WAP-Architektur bietet eine skalierbare und erweiterbare Umgebung für die Entwicklung von Applikationen auf drahtlosen Endgeräten. Jeder Layer bietet den darüberliegenden Schichten über entsprechende SAP’s Zugriff auf seine Services. Darüber hinaus ist es externen Services und Programmen möglich, direkt auf tiefere Schichten zuzugreifen.

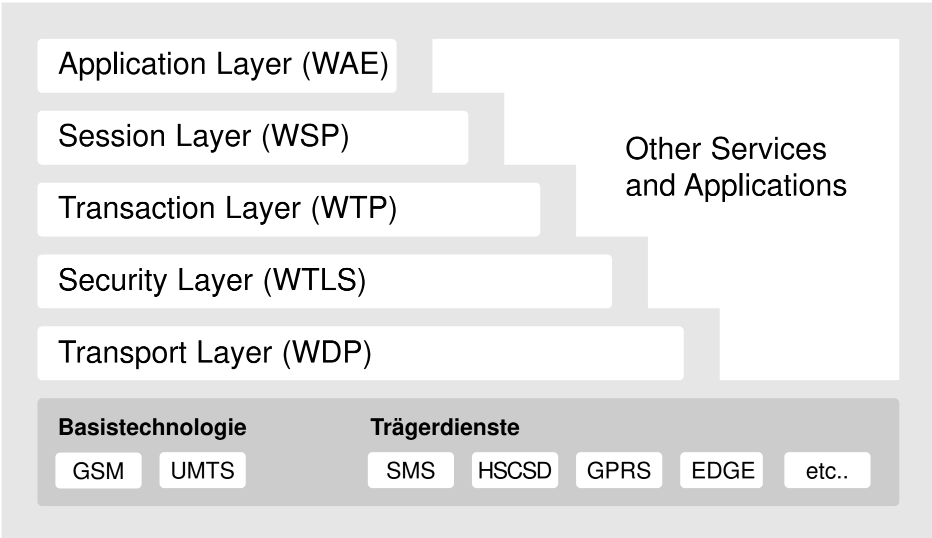

Abbildung 1: Die WAP-Architektur

In Abbildung 1 sieht man einen kleinen Überblick über die WAP-Architektur. In den folgenden Abschnitten werden nun die einzelnen Layer genauer vorgestellt.

2.1 Der Network Layer

Im WAP-Standard ist nicht vorgeschrieben, welche Art von Träger oder auch Bearer-Service zur Datenübertragung eingesetzt wird. Es sind verschiedene Datenübertragungsarten mit unterschiedlicher Dienstgüte (QoS: Quality of Service) möglich. Beim GSM-Netz sind dies im wesentlichen Short Message Service (SMS), Cellbroadcast (CB), General Paket Radio Service (GPRS) und Enhanced Data-Rates for GSM Evolution (EDGE). Somit können sich Durchsatz, Latenzzeiten, Fehlerrate und Echtzeitverhalten stark unterscheiden. Die über dem Network Layer liegenden WAP-Schichten, mit den in ihnen enthaltenen WAP-Protokollen, sind äußerst tolerant gegenüber den unterschiedlichen „Quality of Service “ (QoS)-Eigenschaften. Bis zur Einführung von GPRS wird die Leitung über den Circuid Switched Data (CSD) Kanal aufgebaut. Dazu

ist es notwenig, sich mit einem GSM-Modem bei einem Service-Provider einzuwählen. Über das Point-to-Point Protokoll (PPP) werden dann die Verbindungsparameter ausgehandelt. Nach erfolgreichem Verbindungsaufbau kann dann WAP aufgesetzt werden. Wenn der Wechsel zu GPRS, bzw. generell zu paketvermittelnden Diensten erfolgt ist, wird dies wegfallen, weil die Endgeräte dann in ständiger Verbindung zu den Providern stehen.
Im folgenden Abschnitt wird auf die unterschiedlichen Netze und Bearer Services genauer eingegangen.

2.2 Die Bearer Services

Die unterste Schicht, auf der die WAP-Architektur aufsetzt, wird von den in der Netzwerkschicht liegenden Bearer Services gebilden. Zwei Dinge werden hier grundsätzlich unterschieden. Zum Einen, welche Art von Mobilfunknetz bzw. Übertragungssystem genutzt wird und zum Anderen, welche Trägerdienste genutzt werden. Die momentan wichtigsten Mobilfunknetze und die dazugehörigen Trägerdienste sind:

2.2.1 Basistechnologien

GSM Der „Global System of Mobile Communication“-Standard wurde in seiner ersten Fassung 1990 verabschiedet. Er wurde von der „Groupe Speciale Mobile“ im Auftrag der Europäischen Union entwickelt. Ziel war es ein länderübergreifendes digitales Mobilfunknetz zu entwickeln.
Die Mobiltelefone kommunizieren hier über Funk mit der jeweils nächstgelegenen Basisstation. Das Gebiet einer Basisstation wird als Zelle bezeichnet. Durch Basiskontrollstationen werden die Basisstationen miteinander gekoppelt und einige wenige Mobilvermittlungsstationen verbinden diese dann. Über die Mobilvermittlungsstationen wird dann in andere Netze weitervermittelt. Dabei kann es sich um das Festnetz oder auch um andere Mobilfunknetze handeln.

Das GSM-Netz arbeitet auf drei verschiedenen Frequenzbereichen. In Deutschland wird das 900 MHz Band von den Mobilfunkanbietern D1 und D2 genutzt, das 1800 MHz Band teilen sich E-Plus und VIAG Interkom. Aktuelle Handys sind überwiegend Dualbandhandies und in beiden Netzen einsatzfähig. Im nordamerikanischen Bereich wird auf 1900 MHz gefunkt. Dies verhinderte lange Zeit den internationalen Einsatz von Handys. Inzwischen gibt es Triband Geräte, die auf allen Frequenzbereichen einsatzfähig sind. Als weiterer Frequenzbereich ist das 400 MHz Band bereits angekündigt (GSM 450).
Es werden Verbindungsgeschwindigkeiten von 2.4, 4.8 und 9.6 Kbps unterstützt. In heutigen GSM-Netzen sind teilweise auch auf Kosten der Fehlerkorrektur 14.4 Kbps möglich. Wenn die Gegenstelle einen ISDN-Anschluß mit aktiver ISDN-Karte besitzt, sind ebenfalls Beschleunigungen in diesem Bereich möglich. Durch die Einführung von GSM 2+ als Ausbaustufe von GSM sollen die Übertragungsraten durch Kanalbündelung auf 115.2 Kbps gesteigert werden. Das entspricht dann etwa zwei gebündelten

ISDN-Leitungen.

UMTS „Universal Mobile Telecommunikation System“ oder auch „Wideband Code Division Multiple Access“ (W-CDMA), so die technische Bezeichnung, soll das derzeit genutzte GSM-basierende System in nächster Zeit ablösen bzw. ergänzen. Es wird hierfür sowohl beim Versorger als auch beim Teilnehmer komplett neue Soft- und Hardware benötigt. Im gesamten System wird durchgehend das IP-Protokoll eingesetzt. Sprache wird als „Voice-over-IP“ übertragen. Das UMTS-System besteht aus einem drahtlosen Funknetz und einem trägerabhängigen Festnetz. Das Funknetz besteht aus den Mobilgeräten und einer Basisstation zwischen denen kommuniziert wird. Die Basisstationen werden von dem Trägernetz untereinander verbunden. Die Übertragungstechnik ist, wie bei GPRS, paketorientiert. Über das Trägernetz lassen sich auch Verbindungen in das konventionelle ISDN-Festnetz und zum Internet aufbauen. Die erhöhte Übertragungskapazität wurde durch eine 25fach größere Bandbreite pro Kanal (5 MHz), im Vergleich zu GSM (200 kHz), erreicht.
Das UMTS-Funknetz wird in Versorgungsebenen unterteilt, die mit unterschiedlichen Bandbreiten arbeiten. Die Ebenen sind stark abhängig von der Bewegungsgeschwindigkeit des Netznutzers. Die Datenraten varrieren zwischen 144 Kbps, bei Teilnehmern mit einer Fortbewegungsgeschwindigkeit von bis zu 500 km/h, und 2 Mbps, bei stationären Teilnehmern. Entsprechend wird dann Frequenzmultiplexing (Frequency Division Datamultiplex (FDD)) oder Zeitmultiplexing (Time Division Datamultiplex (TDD)) eingesetzt.

2.2.2 Trägerdienste

SMS Short Message Service ist ein paketorientierter GSM-Dienst. Die Nachricht wird vom Funknetz entgegengenommen und an das Ziel weitergeleitet, sobald dieses eingeschaltet ist. Die Abwicklung wird von einem „Short Message Service Center“ übernommen, das vom Netzanbieter betrieben wird. Es lassen sich bis zu 160 Zeichen in einem SMS-Paket transportieren. Normalerweise wird bei SMS der 7 Bit ASCII-Zeichensatz verwendet. Wenn allerdings Daten verschickt werden, dann wird der erweiterte 8 Bit ASCII-Zeichensatz genutzt. Es lassen sich längere Nachrichten verschicken, indem man mehrere miteinander verkettet. Im Moment sind dies bis zu vier. Mit der Einführung von GSM 2+ werden es dann aber bis zu 255 sein.

HSCSD „High-Speed Circuit Switched Data“ ist eine Lösung, die die Übertragungsrate durch das Multiplexen von zwei bis maximal vier GSM-Zeitschlitzen erhöht. Dies geht allerdings stark auf Kosten der knappen Funkressourcen, da die Leitungen aufgrund der leitungsorientierten Übertragung während der gesamten Sitzung für einen Nutzer alleine reserviert werden. Allerdings regelt das Funknetz die Kanalbündelung dynamisch in Abhängigkeit von der Verkehrslast.
Wie bei neueren GSM-Netzen bereits möglich, wird die Datenrate pro Zeitschlitz von 9.6 Kbps auf 14.4 Kbps erhöht, indem effizientere Fehlerkorrekturverfahren eingesetzt

werden und von einer besseren Verbindungsqualität ausgegangen wird. Durch die Nutzung von 4 Zeitschlitzen lassen sich 57.6 Kbps erreichen. Allerdings ist auch dies nur ein theoretischer Wert, weil praktisch 3 Zeitschlitze für den Downlink (43.2 Kbps) und 1 Zeitschlitz für den Uplink (14.4 Kbps) zur Verfügung stehen.

GPRS “General Paket Radio Service“ ist eine Technik, die bis Ende 2000 flächendeckend verfügbar sein wird und wesentliche höhere Datenraten als GSM bietet. Der Ausbau des GSM-Netzes auf GPRS wird in zwei Schritten vonstatten gehen. Im ersten Ausbauschritt werden maximal 53 Kbps zur Verfügung stehen. Die darauf folgende Stufe wird Geschwindigkeiten über 100 Kbps ermöglichen.

Der GPRS-Standard bietet vier verschiedene Kodierungsverfahren mit unterschiedlichen Datenraten. Es können theoretisch bis zu 8 Zeitschlitze genutzt werden, wobei in der ersten Ausbaustufe nur vier für den Downlink und einer für den Uplink zur Verfügung stehen. Dies kann sich allerdings anbieterabhängig noch verändern.

Coding Scheme	vorhanden ab Ausbaustufe	Datenrate pro Zeitschlitz	theo. max. Datenrate (8 Zeitschlitze)
CS1	1	9,05 Kbps	72,4 Kbps
CS2	1	13,4 Kbps	107,2 Kbps
CS3	2	15,6 Kbps	124,8 Kbps
CS4	2	21,4 Kbps	171,2 Kbps

Tabelle 1: GPRS Coding Schemata

Die in Tabelle 1 dargestellten „Coding Schemata“ stehen zur Verfügung, wobei zu berücksichtigen ist, daß diese Übertragungsraten nur unter idealen Bedingungen gelten.

GPRS basiert auf paketorientierten Übertragungstechniken. Dies kommt dem Einsatz von dem Internetprotokoll TCP/IP sehr entgegen, da dieses ebenfalls paketbasierend ist. Zusätzlich setzt GPRS, genauso wie HSCSD, eine Multislot-Technik ein, wobei sich hier mehrere Teilnehmer die Zeitschlitze teilen.

EDGE „Enhanced Data-Rates for GSM Evolution“ soll die Verfahren HSCSD und GPRS ergänzen und basiert auf einem neuen Modulationsverfahren mit dem die Datenübertragungsgeschwindigkeit eines GSM-Kanals auf 48 Kbps erhöht wird. Bis zu acht GSM-Kanäle können gleichzeitig für die Übertragung genutzt werden. Dies führt zu einer maximalen Datenrate von 384 Kbps. Die Einführung von EDGE ist der letzte Schritt, der noch auf dem GSM-Netz basiert, bevor UMTS eingeführt wird.

	GSM	HSCSD	GPRS	EGPRS**	UMTS FDD (beweglich)	UMTS TDD (stationär)
TDMA	√	√	√	√	–	√
FDMA	√	√	√	√	√	–
max. Datenrate *	9,6	76,8	160	473,6	384	2000
min. Datenrate *	9,6	38	60	170	240	1000
Einführung ab	3Q ’91	1Q ’00	3Q ’00	1Q ’02	1Q ’02	?Q ’03

* Datenrate in Kbps

** GPRS erweitert mit EDGE

Tabelle 2: Die verschiedenen Übertragungsverfahren.

2.3 Der Transport Layer

Vom „Wireless Datagramm Protokoll„ (WDP) werden die Dienste des Transport Layers integriert. WDP bildet den unteren Abschluss der WAP-Protokollfamilie. Es dient der Verbindung zwischen den darüber liegenden Schichten und der darunter liegenden Netzwerkschicht. Abhängig von den verwendeten Mobilfunknetzen (GSM, UMTS) und den verwendeten Transportmechanismen (SMS, HSCSD, GPRS, EDGE), über die die Übertragung gerade stattfindet sind verschiedene Anpassungen notwendig. Dadurch haben Protokolländerungen innerhalb der Netzwerkschicht keinen direkten Einfluß auf die anderen Layer. Es ist optional möglich WDP mit dem „Wireless Control Message Protokoll“ (WCMP) zu ergänzen. Dieses übernimmt die Benachrichtigung der Kommunikationspartner im Falle eines Fehlers. Bei Einsatz des Internetprotokolls (IP) übernimmt dies das Internet Control Message Protokoll (ICMP). Für den Einsatz außerhalb des Carriernetzes, wie beispielweise im Internet, entfällt WDP und wird durch Standard-UDP ersetzt. Das auf einer Trägerfrequenz basierende Carriernetz wird beim Einsatz von paketorientierter Übertragung wegfallen. Bei GPRS und UMTS wird dies dann der Fall sein, was dann dazu führt das WDP vollständig durch UDP ersetzt wird.

2.4 Der Transport Security Layer

Der „Security Layer“ ist optional. Er dient der sicheren Übertragung der Daten über das unsichere Funknetz. Die Aufgaben werden von der „Wireless Transport Layer Security“ (WTLS) übernommen. Es basiert auf der, für drahtlose Kommunikation optimierten, Transport Layer Security (TLS) v1.0. Diese ist auch bekannt als „Secure Socket Layer“ (SSL)-Verschüsselung. Im Wesentlichen sorgt WTLS für drei Dinge:

1. Authentizität
   Dadurch wird sichergestellt, daß die Kommunikationsteilnehmer wirklich diejenigen sind, als die sie sich ausgeben.

2. Anonymität
   Anonymität ist garantiert, wenn die Daten die ausgetauscht werden von anderen weder inhaltlich, noch die Quelle betreffend, ausgewertet werden können.

3. Datenintegrität
   Diese ist gewährleistet, wenn eine Veränderung der übertragenen Daten von Dritten ausgeschlossen werden kann.

Außerdem gibt es noch eine Denial-of-Service Protektion zum Schutz der oberen Schichten, die verhindert, daß ein Service mutwillig blockiert wird.

Es gibt momentan noch keine absolute Sicherheit, da keine End-to-End Verschlüsselung erfolgt. Im Gateway entsteht eine Sicherheitslücke, weil die Daten dort im Klartext vorliegen. Die WTLS gesicherten Pakete werden hier entschlüsselt, um wiederum SSL-verschlüsselt zum Server übertragen zu werden. Es liegt also im Ermessen des Providers die Daten im Gateway angemessen zu schützen.

2.5 Der Transaction Layer

Durch das „Wireless Transaction Protocol“ (WTP) werden die Dienste im Transaction Layer implementiert. WTP ist ein schlankes, an geringen Ressourcenverbrauch angepaßtes Transportprotokoll, über das zuverlässige Point-to-Point-Verbindungen realisiert werden können. Dies beinhaltet die Möglichkeit, auf niedriger Ebene Bestätigungen für einzelne Pakete zu erzwingen. Es kann über sichere und unsichere Datagrammnetzwerke kommuniziert werden. Da-raus ergibt sich folgendes Angebot an Diensten:

- unzuverlässige Einweganfrage
  Diese müssen vom Dienstanbieter nicht bestätigt werden.

- zuverlässige Einweganfrage
  Hier erfolgt eine Bestätigung des Dienstanbieters.

- zuverlässige Zweiweganfrage
  Zusätzlich zur Bestätigung des Dienstanbieters wird diese vom Client wiederum quittiert.

Des weiteren kann durch das Zusammenfassen von Protokolldateneinheiten und durch die gezielte Verzögerung von Bestätigungsmeldungen die Zahl der Nachrichten reduziert werden. Es ist quasiparallele Kommunikation möglich, indem asynchrone Transaktionen unmittelbar nacheineinander beim Client initiert werden, wobei Ergebnismeldungen in beliebiger Reihenfolge eintreffen dürfen.

2.6 Der Session Layer

Das „Wireless Session Protocol“ (WSP) ist das Gegenstück zum Hypertext Transfer Protokoll (HTTP) 1.1 und liefert auch dessen Funktionalität. Es ist verantwortlich für

den organisierten Austausch von komprimierten Inhalten zwischen zusammenarbeitenden Client/Server-Applikationen. Dazu gehört auch der kontrollierte Aufbau und Abbau von Sessions und das Pausieren und Wiederherstellen (Suspend/Resume) von Verbindungen.
Es wurden Optimierungen für relativ große Verzögerungen vorgenommen, die in drahtlosen Netzen häufig vorkommen. Zusätzlich zu den HTTP 1.1-Methoden wurden einige erweiterte Verfahren, sogenannte Push-Techniken, und langlebige Sitzungen eingeführt. Die „Data Push“-Technik dient dazu, Informationen ohne Interaktion des Benutzers an den Client zu senden.
Es gibt zwei Modi und insgesamt drei Sitzungsarten, die jeweils mit oder ohne den optionalen Security Layer (WTLS) betrieben werden können:

1. Verbindungsorientiert:
   Hier wird ein Übertragungskanal in beide Richtungen geöffnet. Dies wird vom Wireless Transaction Protokol (WTP) übernommen. WTP ist ein zuverlässiges Protokoll, da es jede Anfrage bestätigt.

2. Verbindungslos:
   Wenn keine Bestätigung vorliegen muß, wird das Wireless Datagramm Protokol (WDP) genutzt. Es wird dann allerdings unterschieden, ob sich die Verbindung innerhalb oder außerhalb einer bereits existierenden Session befindet.

2.7 Der Application Layer

Der Application Layer wird durch das Wireless Application Environment (WAE) realisiert. Er basiert auf einer Kombination von WWW- und Handytechnologien. Sie sollen gewährleisten, daß Applikationen und Dienste für zukünftige drahtlose Plattformen einfach und effizient implementiert werden können.

Das WAE-Modell hat sich dicht an bestehenden Internetstandards, wie dem HTTP-Protokoll und dem auf OSI basierenden Internetmodell, orientiert. Inhalte und Services lassen sich so auf bereits vorhandenen Web-Servern hosten und dieselben Technologien können genutzt werden.

WAE erweitert einige der WWW-Standards in Bezug auf die Geräte- und Netzwerkcharakteristiken. Diese Erweiterungen beziehen sich in erster Linie auf drahtlose Netzwerkservices zur Anruf- und Nachrichtensteuerung. Besonderer Wert wurde auf den Verbrauch von Ressourcen des Clients gelegt. Arbeitsspeicher und CPU-Leistung sind in den WAP-Geräten der ersten Generation nur sehr begrenzt verfügbar. Desweiteren wurden hohe Latenzzeiten und geringe Bandbreiten bei der Übertragung berücksichtigt.

Wie in Abbildung 2 zu sehen ist, setzt WAE das Vorhandensein eines Gateways voraus. Es übernimmt die Binärkodierung der Daten, die vom Server zum drahtlosen Client und umgekehrt übertragen werden, mittels Kodier- und Dekodierfunktionen.

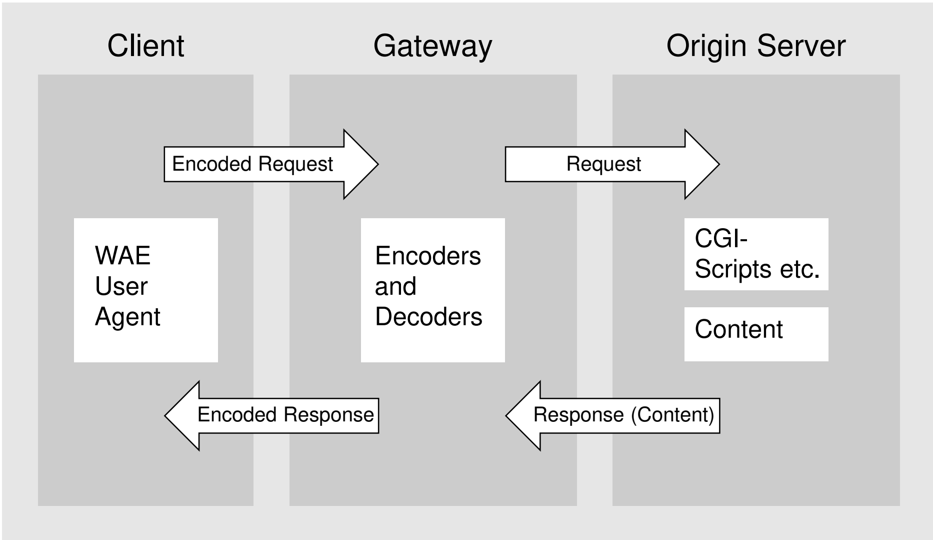

Abbildung 2: Das WAE-Modell

Dadurch reduziert sich die Menge der anschließend benötigten Ressourcen, um die Daten zu verarbeiten.

Das „Wireless Application Environment“ wird, wie in Abbildung 3 ausführlich dargestellt, in folgende Teile unterteilt:

1. Die WAE-User-Agenten

    - Der erste Agent ist ein clientseitiger Microbrowser, der das Navigieren im Internet ermöglicht. Der Microbrowser wird als WML-User-Agent bezeichnet. Dieser ist verantwortlich für die Interpretation und Darstellung der kodierten WML-Seiten, sowie für das Ausführen von vorkompiliertem WML-Script Bytecode.
    - Den zweiten Agenten bilden die Wireless Telephone Applikations (WTA). Sie basieren auf dem WML-User-Agent und bieten Zugriff auf die erweiterten Wireless Telephone Applikations (WTAI). Zu den Services gehört die Speicherung von Inhalten auf dem Endgerät oder die Möglichkeit auf Anrufe serviceabhängig zu reagieren.

2. Die WAE Services und Formate

    - Um Text und Bilder darstellen zu können, wurde die Wireless Markup Language (WML) entwickelt. Da auf den WAP-Endgeräten die Möglichkeiten der Darstellung sehr eingeschränkt sind, werden die Informationen in Decks unterteilt, die wiederum aus einzelnen Cards bestehen. Es wird ein vollständiges Deck in den Speicher geladen und eine Card dargestellt. Der

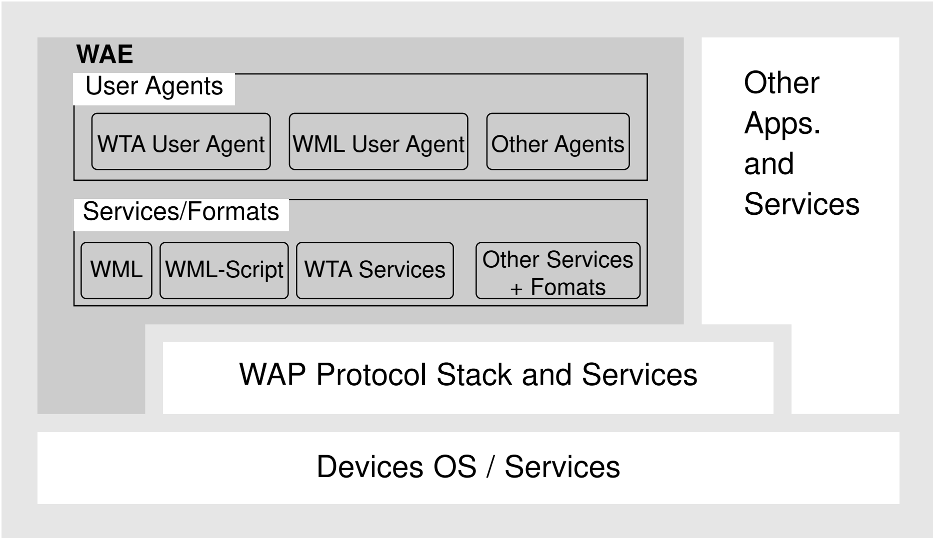

Abbildung 3: Die WAE-Clients

Anwender kann dann durch die einzelnen Cards navigieren.
Bilder werden in einem speziellen Format abgespeichert.

- Desweiteren steht mit WMLScript eine Scriptsprache zur Verfügung, die es ermöglicht direkt auf dem Client Programmcode auszuführen. Dadurch lassen sich aufwendige Dienste, wie die Überprüfung von Benutzereingaben auf ihre Gültigkeit oder der Zugriff auf Zusatzgeräte, realisieren.

3 Das WAP-Modell

Der Anfang des Hype der drahtlosen, digitalen Datenkommunikation war im Wesentlichen durch die intensive Nutzung des Short Message Service gekennzeichnet. Diese Art der Übertragung läßt sich auch für Rechnerdaten nutzen. Um das Internet auf das GSM-Netz zu bringen, wurde im April 1998 das Wireless Application Protokoll verabschiedet. WAP entwickelt sich zu einem industrieweitem Standard für mobilen, drahtlosen Zugriff auf das Internet und auf andere, erweiterte Informationsdienste. Dabei ist allerdings zu berücksichtigen, daß WAP mit hoher Warscheinlichkeit nur eine Zwischenlösung sein wird, solange bis angemessene Bandbreiten und leistungsstarke Clients zur Verfügung stehen. Zu den erweiterten Informationsdiensten gehört auch das hier vorgestellte WAP-Interface zur CRAY T3E. Der WAP-Standard bietet dem Nutzer eines WAP-fähigen Gerätes eine Vielzahl von Diensten. Diese sind an jedem beliebigen Ort ohne zusätzliches Equipment nutzbar. Zu den WAP-Geräten gehören in erster Linie mobile Telefone, aber auch Pager und PDAs. Die Entwicklung von WAP-Applikationen wird explizit von der WAP-Architektur durch diverse Schnittstellen unterstützt.

Das WAP-Forum[2] vereinigt die wichtigsten Hersteller von Hard- und Software, sowie Mobilfunkanbieter. Die Aufgabenstellung an das WAP-Forum war umfassend, denn eine übermittlungsdienstunabhängige Übertragung mußte gewährleistet sein. Es sollte garantiert werden, daß WAP mit allen zukünftigen Mobilfunkstandards kompatibel ist. Heute wird im europäischen Raum überwiegend GSM für den Fernbereich und DECT für den Nahbereich (innerhalb eines Gebäudes) eingesetzt. Im amerikanischen Raum wird CDMA genutzt, der sich in wesentlichen Punkten vom europäischen Standard stark unterscheidet. Die Technologien der kommenden Generation, wie GPRS, UMTS im Fernbereich und Bluetooth im Nahbereich, unterscheiden sich jedoch grundlegend von den heutigen.
Die Architektur von WAP basiert auf dem klassischen ISO/OSI-Schichtenmodell (OSI-Referenzmodell). Für den Aufbau der Seiten wurde die Beschreibungssprache Wireless Markup Language (WML) und die Scriptsprache WML-Script entwickelt. WML basiert auf der Extensible Markup Language (XML[3]) mit angepaßten Definitionen für die drahtlose Umgebung und ist der Hypertext Markup Language (HTML) sehr ähnlich. WML-Script[4] basiert auf ECMAScript[5] und ist somit JavaScript sehr ähnlich. Für den Zugriff auf erweiterte Telefondienste gibt es die „Wireless Telephone Applications“ (WTA). Es wird ein gesicherter Zugang auf das „Telefon Application Programmers Interface“ (API) des Endgerätes bereitgestellt. Das ermöglicht differenzierte Kostenkontrolle, Unified Messaging, FAX-Zugriff oder Banktransaktionen.

Primäre Ziele bei der Entwicklung von WAP waren und sind die Anbindung des In-

[2] `http://www.wapforum.com`
[3] siehe Abschnitt 5.3
[4] siehe Abschnitt 3.2.2
[5] `http://www.ecma.ch`

ternets an drahtlose Netze und deren Endgeräte, die Definition eines global einheitlichen Standards, die Ermöglichung der Erstellung von Inhalten und Applikationen, die auf einer großen Anzahl von Netzen und Endgeräten verfügbar sind und die Unterstützung und Erweiterung vorhandener Standards und Technologien. Die WAP-Spezifikation spart die Präsentation bzw. die Benutzerschnittstelle aus und überläßt diese den Endgeräteherstellern.

Die Standardaustattung eines WAP-Gerätes umfaßt eine Implementierung eines WAP-Stacks für die Kommunikation, einen Microbrowser für die Darstellung der WML-Seiten und eine Wireless Telephony Applications (WTA) Umgebung für die Anfertigung spezieller Lösungen (WTA erlaubt über ein spezielles Interface (WTAI) typische Funktionen eines Handys durch WML oder WMLScript aufzurufen.).

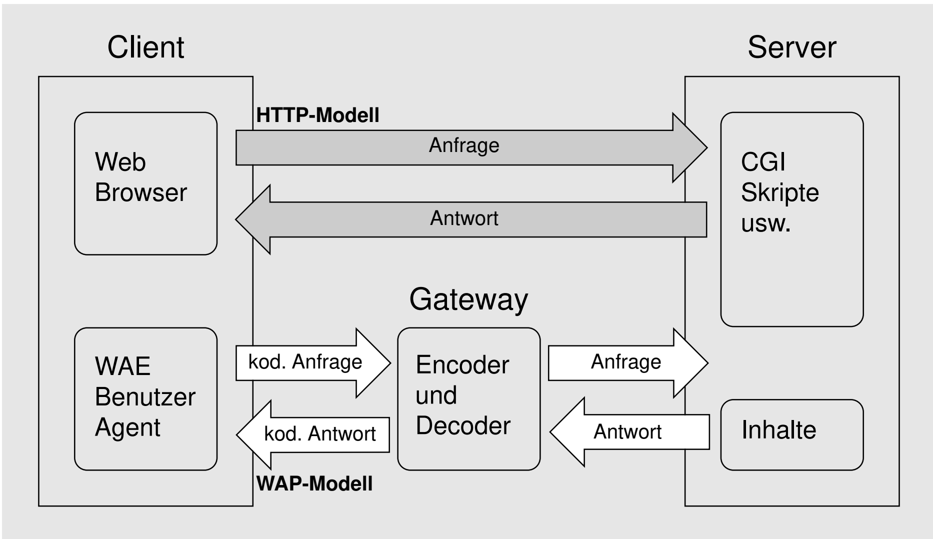

Abbildung 4: Das WAP-Modell im Vergleich zum HTTP-Modell.

Soweit es möglich war, wurde für das WAP-Modell auf bestehenden Standards aufgesetzt. Das WAP-Modell ist mit dem WWW-Modell identisch, mit der Ausnahme, daß zwischen Server und Client ein Gateway notwendig ist, der die Daten entsprechend aufbereitet. In Abbildung 4 wird dies deutlich.

Auf Basis des Wireless Session Protocols (WSP) wird eine Sitzung zwischen Endgerät und dem WAP-Gateway aufgebaut. Das Gateway setzt die WAP-Anfrage in eine Standard-HTTP-Anfrage (HTTP 1.1) um und schickt diese an den WWW-Server. Auf dem Web-Server können nun entsprechende WML-Seiten hinterlegt sein, die dann an den Gateway geschickt und anschließend wieder an das Endgerät weitergeleitet werden. Die Seiten werden zwischen Client und Gateway in einem komprimierten Binärformat übertragen. Ein „Microbrowser“ im drahtlosen Client koordiniert die Steuerung und ist vergleichbar mit einem „WWW-Browser“. Die eingesetzte Namensstruktur (URLs) ist ebenfalls identisch mit der im World Wide Web bereits vorhandenen.

3.1 Das Dokumentformat

WML bietet die Möglichkeit zur Darstellung von Text und Grafik auf Clients mit geringer Leistungsfähigkeit. Der Aufbau von WML ähnelt HTML sehr stark. Es basiert auf XML, wurde aber den Belangen der drahtlosen Kommunikation angepaßt. So sind beispielsweise die Anzeigemöglichkeiten von Handys und anderen mobilen Endgeräten aus naheliegenden Gründen sehr eingeschränkt. Die größte Herausforderung, die sich dadurch ergibt, ist die Informationsreduzierung. Deswegen ist es nicht sinnvoll, obwohl es technisch einfach umzusetzen ist, HTML einfach direkt in WML zu übersetzen.

3.2 WAP 1.1/1.2

Im April 1998 wurde die Version 1.0 der WAP-Spezifikation verabschiedet und dem World Wide Web Consortium (W3C) vorgelegt. Dabei wurde versucht, soweit wie möglich, auf bestehende Standards aufzubauen. Dennoch wurde an der ersten WAP-Version die mangelnde Kompatibiltät zu XML bemängelt. Daraufhin wurde der Standard komplett überarbeitet. Im Juni 1999 wurde der WAP 1.1-Standard verabschiedet. WAP 1.1 ist nicht abwärtskompatibel zum 1.0 Standard und mittlerweile die Basis, auf der weiterentwickelt wird. Wenige Monate später, im März 2000, wurde dann die aktuelle WAP 1.2-Fassung veröffentlicht. Der WAP-Standard 1.2 ist vollständig kompatibel zu dem 1.1 Standard und ist im Wesentlichen eine Erweiterung. Die neue Spezifikation änderte nur wenig an WML und WMLScript, jedoch sind im gesamten Bereich kleine Änderungen zu finden. Hinzugekommen sind wesentliche Möglichkeiten der Endgerätedifferenzierung (Unterstützung zusätzlicher Träger) und die Push-Technik. Damit werden sogenannte „Real Time Alerts“ für das Senden von Daten von einem Server zu interessierten Clients ermöglicht. Das ist beispielsweise für die Umsetzung von Newstickern via WAP notwendig.

3.2.1 WML

Mit WML lassen sich Textabsätze, Tabellen, Formulare und, in eingeschränktem Rahmen, auch Bilder anzeigen. Für die Bilddarstellung wurde das Wireless Bitmap Format (WBMP) entwickelt. Momentan unterstützt dieses allerdings nur eine Farbtiefe von einem Bit und eine maximale Bildgröße von 150x150 Pixeln. Die Seite, ein sogenanntes Deck, wird in mehrere Cards unterteilt, um diese auf dem kleinen Display darstellen zu können. Das Deck wird in den Arbeitsspeicher des Gerätes geladen. Die Cards werden dann einzeln auf dem Display dargestellt und sind durch Links miteinander verknüpft. Man unterscheidet dabei Links, die innerhalb eines Decks verknüpfen und solche, die auf andere Decks verweisen. Die Links werden analog wie in HTML umgesetzt. Die mit jedem beliebigen Editor erstellbaren WML-Seiten lassen

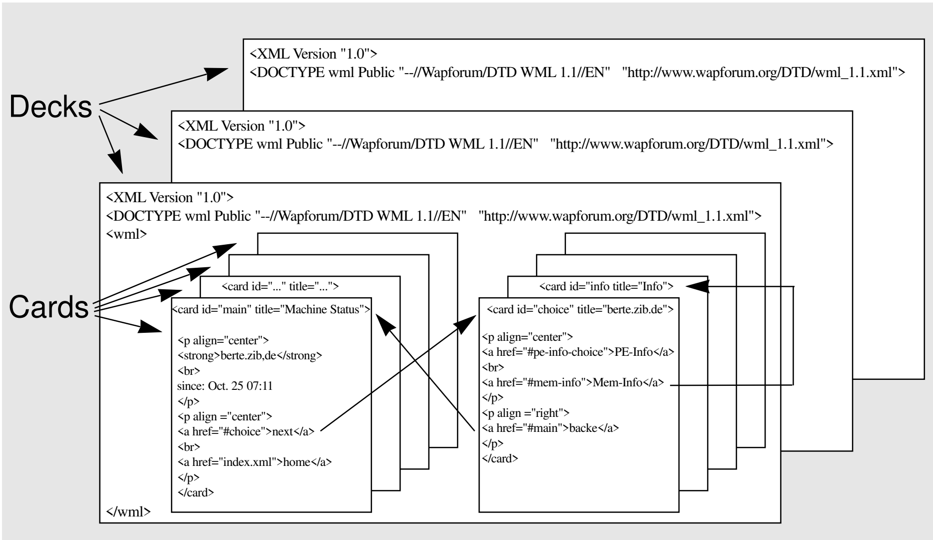

Abbildung 5: WML-Dokument mit seinen Decks und Cards

sich nur mit einem speziellen WAP-Microbrowser darstellen. Solch ein Microbrowser ist in jedem WAP-fähigen Handy enthalten. In Abbildung 5 ist die Struktur eines WML-Dokumentes dargestellt.

3.2.2 WMLScript

Neben WML, gibt es noch die Scriptsprache WMLScript. Sie ist speziell für die Förderung von Interaktionen zwischen WML-Seite und Nutzer gedacht. Das beinhaltet die Kontrolle von Eingaben, das Anzeigen von Dialogboxen, das Ansprechen von Peripheriegeräten oder kleine Applikationen. WMLScript ermöglicht es, Programmcode direkt auf den Endgeräten auszuführen.

4 Existierende WAP-Systeme

4.1 Verfügbare Geräte

Die Palette der WAP 1.1-fähigen Geräte ist mittlerweile recht groß und wächst beständig. Wir stellen in Tabelle 3 daher nur eine Übersicht der im Oktober 2000 in Deutschland verfügbaren Geräte dar.

Es zeigt sich, daß bis auf wenige Ausnahmen die Displays der Geräte sehr klein sind (15-34 Zeichen in 4-6 Zeilen). Die Displaygröße ist deshalb das entscheidende Kriterium bei der Entwicklung der WML-Seiten. Eine weitere Einschränkung besteht darin, daß der überwiegende Teil der Displays nur die monochrome Darstellung unterstützen. Es gibt momentan noch kein Gerät, welches Farben darstellen kann. Da WML und WBMP aber in der aktuellen Spezifikation ebenfalls keine Farbinformationen enthalten, bedeutet das keinen echten Nachteil (siehe 3.1).

4.2 Verfügbare Software

In vielen Geräten wird der UP.Browser von phone.com[6] eingesetzt. Er ist vollständig WAP 1.1 kompatibel und unterstützt die Anzeige von WML-Seiten und WBMP-Graphiken. Die Sicherheitsfunktionen des WTLS wurden ebenfalls implementiert.
Nokia hat ebenfalls einen WAP-Browser entwickelt, der die gleichen Möglichkeiten wie der UP.Browser bietet. Er wird in den WAP-fähigen Geräten von Nokia verwendet.

Es existieren verschiedene WAP-Server auf dem Markt, z.B. von Nokia, phone.com, Infinite oder Ericsson. Diese Lösungen sind jedoch meist ganz speziell auf E-Commerce über WAP oder ähnliche Anwendungsgebiete zugeschnitten und werden kommerziell vertrieben. Daher konnten sie hier keiner eingehenden Prüfung unterzogen werden.

Für die Unterstützung von WAP-Entwicklern bieten einige Unternehmen (z.B. Nokia und Ericsson oder auch phone.com) WAP-Entwicklungsumgebungen an, die einen Simulator eines WAP-Gerätes enthalten. Da uns zunächst kein WAP-fähiges Gerät zur Verfügung stand, haben wir zum Testen der WML-Seiten auf einige im Folgenden vorgestellte Entwicklungsumgebungen zurückgegriffen.

[6] www.phone.com (ursprünglich wireless Planet)

Hersteller	Modell	WAP	Displaygröße in Pixel/Zeilen	Extra AT	IrDA	sonst.
3Com	**Palm III / V**	**1.1**	?×? / ?×?			
Alcatel	**OT VIEW db @**	**1.1**	?×? / 15 × 5			
	OT 30x	**1.1**	96 × 49 / 5			
Benefon	**Twin Dual SIM**		?×? / ?×?	√		
Bosch	**1886**	**1.1**	104 × 64 / 34 × 5			**HSCSD**
	820	**1.1**	104 × 64 / 34 × 5			
Ericsson	**A2618s**	**1.1**	?×? / ?×?			
	R320s	**1.1**	? × 7	√	√	
	R380s	**1.1**	?×? / 15 × 5	√	√	
	MC218	**k.A.**	640 × 200			
Motorola	**Talkabout v.2288**	**1.1**	?×? / ?×?			
	Talkabout T2288	**1.1**	?×? / ?×?			
	P.7389	**1.1**	?×? / ?×?	√	√	
Nokia	**6210**	**1.1**	96 × 60 / 16 × 4	√	√	**HSCSD**
	7110	**1.1**	96 × 65 / 19 × 6	√	√	
	9110i	**1.1**	84 × 48 / 16 × 5			
			640 × 200 / 80 × 24			
Sagem	**MC939 WAP**	**1.1**	? × 8			
Samsung	**SGH-A110**	**1.1**	128 × 64	√	√	
Siemens	**C35i**	**1.1**	? × 5	√		
	S35i	**1.1**	? × 7	√	√	
	M35i	**1.1**	? × 5			**Outdoor**
	IC35	**1.1**	240 × 160			
Sony	**CMD-Z5**	**1.1**	?×? / ?×?	√		**HTML, E-Mail**

http://www.palm-europe.com/de
http://www.alcatel.com/consumer/mobilephone/index.htm
http://www.benefon.com/
http://www.bosch.de/uc/eg/produkte/
http://www.ericsson.de/mobilekommunikation/mobiltelefone/index.html
http://www1.motorola.de/mobiltel/public/produkte/produktinfos.shtml
http://www.nokia.de
http://www.sagem.com/en/produits-en/telephonie-en.htm
http://samsungelectronics.com/mobile/products/reg-germ.html
http://www.siemens.de/mysiemens/deutsch/mobile/content.html
http://www.sony.de/produkt/index.html15

Tabelle 3: Marktübersicht von aktuell verfügbaren Handys

4.2.1 WAP-Enwicklungsumgebungen

Die folgenden Tools lassen sich unterstützend beim Programmieren von WAP-Seiten einsetzen:

WAPAlizer Bei gelon.net[7] gibt es einen Emulator, dessen Online-Nutzung kostenlos ist, lediglich die Offline-Version ist kostenpflichtig. Hier läßt sich ein WAP-fähiges Handy aussuchen, welches dann simuliert wird. Die Navigation erfolgt dann gemäß des ausgesuchten Handys. Teilweise ist WAPAlizer noch ziemlich instabil. Für schnelle Tests ist er jedoch gut geeignet.

YOURWAP YOURWAP[8] kann WAP-Seiten und HTML-Seiten parallel darstellen. Es werden verschiedene Handys zur Auswahl angeboten. Die Navigation erfolgt hier vollständig über die entsprechenden Knöpfe und Schalter. Dadurch ist die Simulation sehr realitätsnah.

WINWAP WINWAP[9] ist Freeware und einer der leistungsstärksten Emulatoren. Allerdings werden die WAP-Seiten teilweise anders als auf einem Handy angezeigt. Damit ist er für die Entwicklung ungeeignet. Als Browser an sich, bietet er vieles was man von HTML-Browsern gewohnt ist.

Nokia WAP Toolkit Das Nokia WAP Toolkit[10] läßt sich nach einer Registrierung kostenlos herunterladen und ist das mit Abstand mächtigste kostenlos verfügbare Paket. Es simuliert ein WAP-fähiges Mobiltelefon und stellt eine Entwicklungsumgebung für WAP Projekte zur Verfügung. Dazu gehören u.a. ein WML-Browser mit den dazugehörenden WMLScript-Interpretern und den WMLScript-Bibliotheken, eine WAP-Telefonsimulation mit unterschiedlichen Geräten und Editoren für WML, WMLScript und WBMP-Bilder. In der aktuellen Version 2 werden die WAP V1.2 Standards unterstützt. Dazu gehört ein WAP-Server Simulator für Geräte, die Zugriff zu einem WAP-Gateway benötigen und ein WAP-Push Simulator.

4.3 Das Gateway

Für das Abrufen von WAP-Seiten ist immer ein spezieller WAP-Gateway notwendig. Er läßt sich funktionell mit Gateways aus dem Internet vergleichen, bietet jedoch zusätzliche Funktionen. Dazu gehört die Funktion eines Proxy-Servers der zwischen WAP-Client und dem WAP-Server steht. Neben den klassischen Aufgaben eines Proxy-Servers, wird der Datenstrom vom WAP-Gateway komprimiert, um den ressourcenschwachen Clients gerecht zu werden. Hier wird WML bzw. WMLScript in kompakten Bytecode übersetzt.

[7] `http://www.gelon.net/about/wapalizer.html`
[8] `http://www.yourwap.com/de/download/download.html`
[9] `http://www.winwap.org/winwap/download.html`
[10] `http://www.forum.nokia.com`

Für die eigentliche Kommunikation mit dem Endgerät wird auf niederer Ebene eine Umsetzung der HTTP- in die für drahtlose Kommunikation optimierten WAP-Protokolle vorgenommen.
Für die WAP-Gateways wurden Minimalanforderungen festgelegt, die erfüllt werden müssen. Zu den Layern die zumindestens teilweise implementiert sein sollten, gehören der Session-, der Transaktion- und der Transport-Layer.
Den Herstellern wurden bei der Implementierung größtmögliche Freiheiten gelassen. Die Betreiber verschiedener Gateways favorisieren unterschiedliche Hersteller gleichermaßen. Das hat zur Folge, daß sich der erzeugte Bytecode von Gateway zu Gateway stark unterscheidet. Dadurch ist es möglich das WAP-Gateways in einem Netz funktionieren und in einem anderen nicht. Auch ist es momentan noch nicht ungewöhnlich, daß der eine oder andere Client wegen des Bytecodes abstürzt, oder weil ihm der Speicher zu knapp wird. 1500 Bytes ist momentan eine magische Speichergrenze, deren Überschreitung zu schwer identifizierbaren Problemen führen kann. Ein weiterer Fehlerfaktor ist das Caching, welches auf unterschiedlichste Weise implementiert wurde.

4.3.1 Kommerzielle Gateways

Der Markt an Gateways ist zur Zeit noch überschaubar. Neben den großen Anbietern (Nokia, Ericsson, Materna, Phone.com) gibt es eine Anzahl kleinerer Anbieter, die oftmals Teilrealisierungen vorgenommen haben. Diese leisten nicht alles, sind dafür aber auch kosten- und ressourcenschonender. Hier werden vorerst einige der großen kommerziellen Gateways vorgestellt, bis später auf die in unserem Projekt favorisierte Lösung eingegangen wird:

- Der erste WAP-Gateway wurde von der kalifornischen Firma Phone.com (damals Unwired Planet) 1996 entwickelt. Das UP.Link benannte Programm enthält ein ganzes Paket von Diensten für mobile Anwendungen. Die aktuelle Version beinhaltet neben dem eigentlichen Gateway eine Administrationssoftware und bietet außerdem noch spezielle erweiterte Dienste (UPMail, UPOrganize usw.). Inzwischen ist der aktuelle WAP 1.1. Standard vollständig implementiert worden. Dazu gehören Möglichkeiten zur sicheren Anbindung an Content-Server mit SSL/TLS. Die Push-Services sind noch nicht ausgereift, da sie nicht mit aktuellen WAP-Telefonen zusammenarbeiten.

- Der Nokia-Gateway ist vollständig in Java programmiert, wodurch dieser sehr leicht auf viele Plattformen portierbar ist. Durch die integrierte Servlet Engine[11] lassen sich Inhalte direkt im Gateway generieren. Die aktuelle Version unterstützt WTLS, SSL-Verschlüsselung und Teile des vor kurzem verabschiedeten WAP 1.2 Standards.

- WAPLite von Infinite Technologies zeichnet sich durch seine geringe Größe aus. Das gesamte Paket nimmt gerade mal ein halbes Megabyte auf der Festplatte

[11] Eine Servlet Engine dient der on-the-fly generierung von WAP-Seiten.

ein, obwohl es in der aktuellen Version den WAP 1.1 Standard eigentlich so gut wie vollständig implementiert (auch WTLS).

Dr. Materna	www.materna.de
Ericsson	www.ericsson.com/wireless/products/mobsys/... .../gsm/subpages/wise/wap_prox.shtml
Kannel	www.kannel.org
Nokia	www.nokia.com/corporate/wap/gateway.html
Phone.com	www.phone.com/products/uplink.html
WAPLITE	www.waplite.com

Tabelle 4: Links zu Gateway-Entwicklern

4.3.2 Kannel

Kannel[12] ist ein OpenSource (BSD-Lizenz) Gateway, der von der finnischen Firma WAPit finanziert wird. Die Firma aus der WAPit hervorgegangen ist, beschäftigt sich seit geraumer Zeit mit der Realisierung von Diensten über SMS. 1998 erkannte man das Potenzial von WAP und beschloß einen eigenen freien Gateway zu entwickeln. Primäre Motivation war hier die Preisgestaltung der vorhandenen Gateway-Anbieter, die es kleinen Dienstanbieter unmöglich machte, Nischenlösungen anzubieten. WAPit erkannte genau diese Gruppe als Kundenpotenzial. Kannel orientiert sich am WAP 1.1 Standard und befindet sich immer noch in einem frühen Entwicklungsstadium, läuft allerdings schon stabil.

Der Aufbau von Kannel: Wie in Abbildung 6 dargestellt, besteht Kannel im Wesentlichen aus zwei Komponenten:

- Die Bearer Box verbindet sich mit dem SMS-Center und dem CSD-Router und bietet den anderen Boxen somit eine einheitliche Schnittstelle. Dadurch ist die Implementation des WDP-Protokolls des Transport Layers realisiert.
- Die WAP Box beinhaltet die oberen Schichten der WAP-Architektur. Dazu gehört in erster Linie WTP und WSP (Optional auch WTSL). Sie übernimmt die WTP-Kommunikation zwischen Handy und dem Contentserver.

Die Bearer Box empfängt die UDP-Pakete von dem CSD-Router. Bei dem CSD-Router kann es sich um eine ISDN-Karte, ein Modem oder um einen Modempool handeln. Von der Bearer Box werden die UDP-Pakete dann zur WAP-Box geroutet. Diese empfängt die kodierte Anfrage vom Handy, dekodiert diese und macht eine HTTP-Anfrage beim Webserver. Die Antwort wird nun wieder kodiert und zum Handy übertragen.

[12] eigentlich ein finnisches Saiteninstrument

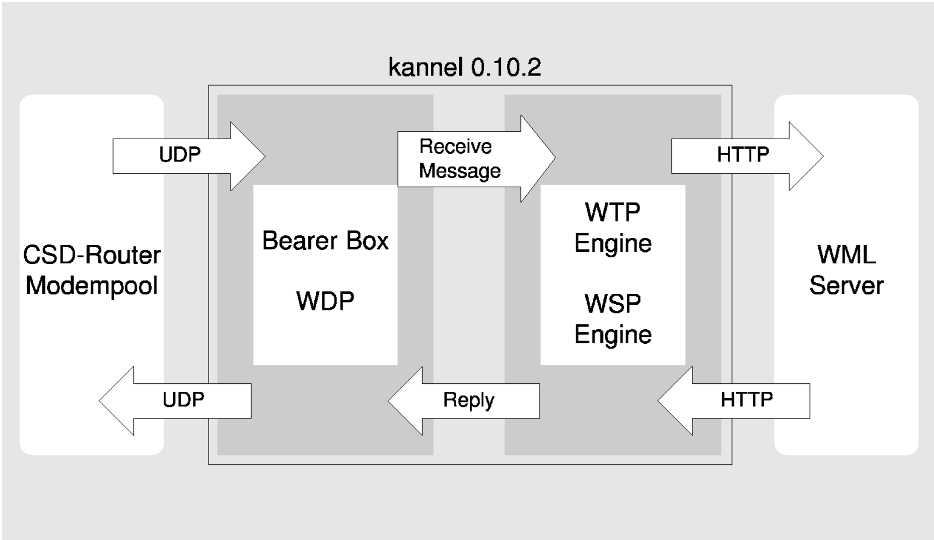

Abbildung 6: Die Kannel-Architektur

WTLS: Die aktuelle Version von Kannel 0.13 besitzt keine Implementation der Wireless Transaction Layer Security (WTLS). Da Bedarf für einen WTLS-Layer besteht, existiert eine OpenSouce WTLS-Implementation von 3iu.com[13] die speziell für Kannel entwickelt wurde. Die während der Entwicklung des WAP-Dienstes zur Verfügung stehende Version war jedoch nur auf einer veralteten Kannelversion (0.8.2) lauffähig und deshalb für unser Projekt nicht geeignet. Aus diesem Grund haben wir uns für ein anderes Sicherheitskonzept entschieden, auf das im Implementationsabschnitt (Abschnitt 6) näher eingegangen wird.
Seit kurzer Zeit existiert eine aktuelle Version der WTLS-Implementation für Kannel 0.10.3. Diese konnte im Rahmen dieser Arbeit nicht mehr untersucht werden, wird aber voraussichtlich im weiteren Verlauf des Projektes zum Einsatz kommen.

[13] `http://www.3ui.com`

5 Anwendungskonzept

Das Ziel unserer Arbeit war die Erstellung einer Anwendung, die den Zugriff auf Prozeßinformationen von Hochleistungsrechnern, in unserem Fall eine SGI/CRAY T3E, über ein drahtloses Endgerät (z.B. ein Handy) erlaubt. Die Applikation sollte außerdem als Modell für weitere Anwendungen im Bereich der Serviceleistungen des ZIB dienen. Auch sollte das Anbinden neuer Module bzw. das Einbringen von Erweiterungen ohne größere Probleme möglich sein.
Aufgrund dieser Anforderungen haben wir eine Reihe von Architekturentscheidungen getroffen, die im Folgenden erläutert werden.

5.1 Architekturüberblick

Das Projekt diente der Erprobung der neuen Technologien (WAP, WML, drahtlose Netze). Wir haben ausschließlich frei verfügbare Software verwendet, was verschiedene, entscheidende Vorteile bot:

- Wie die Bezeichnung schon nahe legt ist sie kostenlos.
- Die Autoren freier Software pflegen in der Regel einen engen Kontakt zu den Benutzern ihrer Programme, da sie an der raschen Weiterentwicklung interessiert sind. Außerdem sind die Autoren auf die Meldung von Fehlern angewiesen und auch deshalb an Rückmeldungen interessiert.
- Diese Art von Software wird oft als *Open Source* verbreitet, d.h. der Quellcode steht den Anwendern zur Verfügung, die ihn dann auch nach den eigenen Anforderungen anpassen können.

Als Format zum Datenaustausch wird die Extensible Markup Language[14] (XML, siehe Abschnitt 5.3) verwendet. Diese Entscheidung wurde vor dem Hintergrund der geforderten Erweiterbarkeit des Systems getroffen. XML bietet zum Einen die Möglichkeit die Strukturierung der Daten problemlos zu ändern (falls sich die Möglichkeiten der Datengewinnung unter UNICOS ändern sollten), andererseits ist XML ein universelles Format, das in verschiedenste Anwendungen exportiert werden kann. Da in unserem Fall die Daten in einem WML-Browser, d.h. von einem WAP-fähigen Client dargestellt werden sollen, wird die Transformation von XML in WML auf dem Applikationsserver (WAP-Server) durchgeführt. Diese Vorgehensweise bietet darüberhinaus die Möglichkeit, die gleichen Daten auch für andere Clients aufzubereiten, z.B. für herkömmliche WWW-Browser.

Als WAP-Server kam ein *Apache*[15] zum Einsatz, der um die Möglichkeit WML-Seiten zu verschicken erweitert wurde. Als WAP-Gateway verwendeten wir *Kannel*[16] (siehe

[14] `http://www.w3c.org/XML/`
[15] `http://www.apache.org`
[16] `http://www.kannel.org`

auch Abschnitt 4.3.2). Es befindet sich noch im Entwicklungsstadium (Version 0.10.2) und bietet insbesondere keine Unterstützung für WTLS, d.h. die Verbindungen zwischen Client und Server können nicht verschlüsselt werden. Diese Einschränkung spielt in unserer Umgebung allerdings keine Rolle, da sich die Nutzer direkt im ZIB einwählen. Dementsprechend werden die Nutzer vom Einwahlserver authorisiert. Der WAP-Server wiederum läßt nur Verbindungen zu, die von Rechnern innerhalb des ZIBs kommen. Somit werden die Daten nur durch das lokale Netz unverschlüsselt übertragen, was zu Testzwecken durchaus tolerierbar ist.

Die erforderliche Infrastruktur zur XML-Verarbeitung bzw. Aufbereitung bietet ein spezielles Modul für den verwendeten WWW-Server Apache: *Cocoon*[17]. Es ist ein Java-basierendes Framework zur XML-Bearbeitung und benötigt eine Java-Laufzeitumgebung.

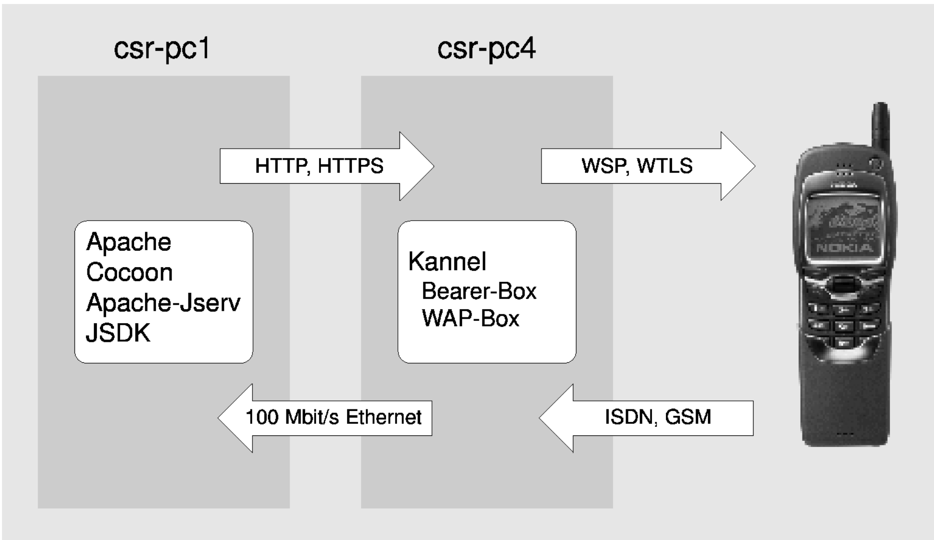

Abbildung 7: Konfiguration der WAP-Testumgebung

Die Abbildung 7 zeigt die Verteilung der einzelnen Software-Pakete auf den zur Verfügung stehenden Rechnern. Daraus ergab sich folgende Konfiguration:

- **2 Linux-Rechner**: Pentium III 800 MHz, 256 MB Hauptspeicher, SuSE Linux 6.4, Kernel 2.2.16
- **WAP-Gateway**: Kannel 0.10.2
- **WAP-Server**: Apache 1.3.12, Java Servlet Development Kit 2.0 (JSDK), Apache-JServ 1.1.2, Cocoon 1.7.4

[17] `http://xml.apache.org/cocoon/`

Die scheinbare Häufung der Pakete auf dem Rechner `csr-pc1` hängt damit zusammen, daß das JSDK, Apache-Jserv und Cocoon Komponenten sind, die in den Apache-Webserver integiert werden. Eine ausführliche Darstellung zum Testaufbau und -ablauf befindet sich im Abschnitt 6.

5.2 UNICOS/mk

Beim Hochleistungsrechner „`berte`“ handelt es sich um einen Parallelrechner vom Typ T3E der Firma SGI/CRAY. Er gehört zu den massiv parallelen Prozessorsystemen (MPP) und ist mit insgesamt 408 DEC EV5.6 Alpha Prozessoren ausgestattet, von denen 384 für parallele Applikationen zur Verfügung stehen und 24 für Betriebssystemaufgaben reserviert sind.

Anzahl PEs	**Taktrate**	**lokaler Speicher**
48	600 MHz	512 MB
88	600 MHz	128 MB
264	450 MHz	128 MB
8	450 MHz	256 MB

Tabelle 5: PE-Konfiguration der T3E

Die Tabelle 5 zeigt die Konfiguration der Prozessoreinheiten (PEs). Dies ist auch ein Teil der Informationen, die auf der Maschinenstatus-Seite (siehe Abschnitt 6.2.2) ausgegeben werden.
Als Betriebssystem wird UNICOS/mk verwendet. Hierbei handelt es sich um ein verteiltes Betriebssystem, bei dem die PEs in Betriebssystem-, Kommando- und Applikations-PEs eingeteilt werden. Der Namenszusatz „mk“ weist darauf hin, daß das Betriebssystem auf einem Microkernel basiert, der auf jedem Knoten (ein Prozessor mit zugehörigem Hauptspeicher und I/O-Hardware) ausgeführt wird und nur eine gewisse Basisfunktionalität zur Verfügung stellt.

Die Konfiguration der PEs und ihre Zuordnung zu den einzelnen Gruppen (Betriebssystem, Kommando...) nimmt in UNICOS/mk der Global Ressource Manager (GRM) vor. Er ist auch für die Anzeige von Statusinformationen über den aktuellen Betriebszustand verantwortlich. Dazu existiert das Kommando `grmview`. Es gibt, je nach Kommandozeilenoptionen, detailiert Auskunft, welcher Gruppe die einzelnen PEs angehören, über wieviel lokalen Speicher sie verfügen, usw. (siehe Abbildung 8).
Für die Ausführung von Programmen im Hintergrund (als *Batchjobs*) ist auf dem Parallelrechner das Network Queuing System (NQS) installiert. Batchjobs werden in sogenannte Warteschlangen (*queues*) eingereiht und ohne Eingreifen des Benutzers abgearbeitet. Daher müssen alle Informationen, die zur Ausführung des Programms benötigt werden, in einem Skript abgelegt werden. Hier werden z.B. die notwendigen

```
bzfheidl@berte /home/bzfheidl/> grmview -ldm
PE Map: 408 (0x198) PEs configured(normal scheduling)
             Ap. Size  Number Aps.  Abs.           <<<< Lists >>>>>
 Type    PE  min  max running limit limit    Label svc uid gid acid  x  y  z Clock
 UsrMem FreMem 1PE-Lim 1PE-Run MPP-Lim
+ APP     0    2  396       1     1     2    Stime  -   -   -    -   0  0  0   450
 119     26       0       0       1
+ APP     1    2  396       1     1     2    Stime  -   -   -    -   1  0  0   450
 119     28       0       0       1
+ APP     2    2  396       1     1     2    Stime  -   -   -    -   0  1  0   450
 119     27       0       0       1
+ APP     3    2  396       1     1     2    Stime  -   -   -    -   1  1  0   450
 119     27       0       0       1
+ APP     4    2  396       1     1     2    Stime  -   -   -    -   0  2  0   450
 119     27       0       0       1
 ...
```

Abbildung 8: Ausgabe von `grmview`

Betriebsmittel (PEs, Speicher, Rechenzeit usw.) spezifiziert.

Um den aktuellen Zustand der eigenen Batchjobs zu erfragen oder ganz allgemein den Zustand der Batch-Queues zu untersuchen, stellt das NQS das Kommando `qstat` zur Verfügung. Es informiert über den „Füllzustand“ der Warteschlangen, die Benutzer, die Batchjobs abgeschickt haben (bei entsprechender Berechtigung) oder auch, welche Jobs gerade ausgeführt werden (siehe Abbildung 9).

```
bzfheidl@berte /home/bzfheidl> qstat -bam
--------------------------------------
NQS 3.3.0.15 BATCH REQUEST MPP SUMMARY
--------------------------------------
IDENTIFIER    NAME    USER        PE'S     REQUEST-TIME  P-TIME  ST
                              LIM USED     LIM   USED     LIM
------------- ------- -------  ---- ----  ------ ------  ------  ---
34857.berte   uhca246 bvit09mf   26   26   28000   1252   28000   R03
34925.berte   H2      bvkk01ce  202  202   36000  11403  <43200>  R03
34956.berte   visk-ein nhkfmei1    4    4    7200   1343    7200   R04
...
34928.berte   CYL14X  bvit15ss   80        28800          28800   Qqr
34929.berte   CYL14X  bvit15ss   80        28800          28800   Qqr
34930.berte   CYL14X  bvit15ss   80        28800          28800   Qqr
34936.berte   dc_wf_qu bvch12vb  100         3600           3600   Qqr
...
```

Abbildung 9: Ausgabe von `qstat`

5.3 XML

Die Extensible Markup Language ist ein Dokumentverarbeitungsstandard, der vom World Wide Web Consortium[18] (W3C) vorgeschlagen wurde. Obwohl die Spezifikation noch nicht vollständig abgeschlossen ist, wird XML als die Technologie angesehen, die zukünftig für dynamisch generierte Inhalte verwendet wird. Tatsächlich gibt es bereits einige WWW-Browser die die Anzeige von XML-Daten teilweise unterstützen[19]. Die historischen Wurzeln von XML liegen, ebenso wie bei HTML, in der Standard Generalized Markup Language (SGML). Der wesentliche Unterschied zwischen XML und HTML ist, daß XML bereits selbst eine Metasprache zur Entwicklung von Auszeichnungssprachen ist. Jedoch ist XML gegenüber SGML soweit vereinfacht, daß sich damit keine weiteren Metasprachen erzeugen lassen. HTML läßt sich also durch XML erzeugen. Dies gilt auch für WML. HTML und WML liegen, wie in Abbildung 10 zu sehen, beide auf der Ebene der Auszeichnungssprachen.

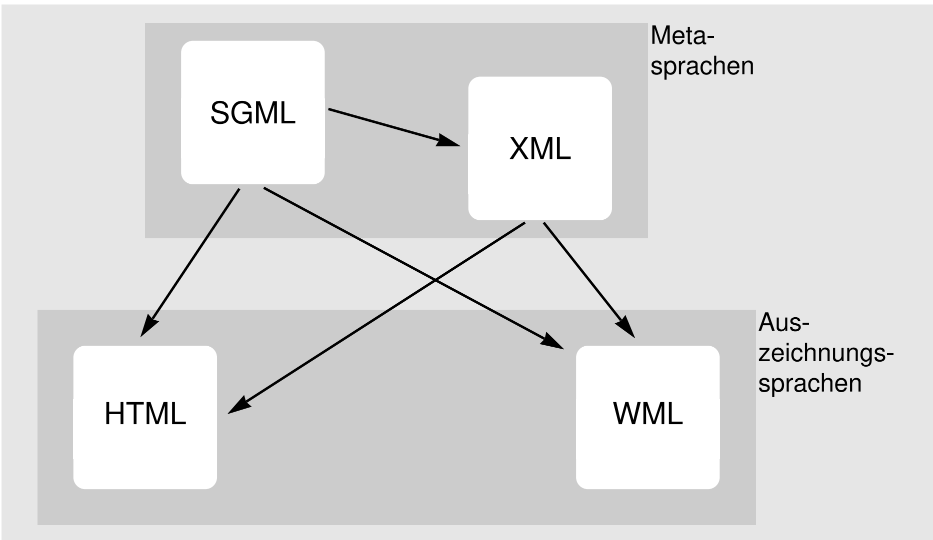

Abbildung 10: Abgrenzung von Metasprachen und Auszeichnungssprachen

XML ist unabhängig vom Ausgabemedium. Der Autor konzentriert sich ausschließlich auf den Text und dessen logische Strukturierung. Die Darstellung wird dann, in Abhängigkeit vom Darstellungsmedium, von entsprechender Software übernommen. In XML werden die Dokumente in hierarchisch geordnete Abschnitte strukturiert, die in Tags eingeschlossen sind. Anders als bei HTML sind diese Tags jedoch nicht statisch, sondern können vom Benutzer nach gewissen Regeln selbst definiert werden. In einer Document Type Definition[20] (DTD) wird festgelegt, welche Tags in diesem

[18] http://www.w3c.org

[19] http://www.opera.com/de

[20] siehe XML-Spezifikation: http://www.w3.org/TR/REC-xml\#dt-doctype

XML-Dokument vorkommen dürfen, wie sie ineinander geschachtelt werden können und mit welchen Attributen die Tags erweiterbar sind. Die DTD definiert demzufolge eine Auszeichnungssprache.
Zur Erstellung von WML-Seiten ist es zur Strukturierung erforderlich, die in der WML-DTD definierten Tags zu verwenden. Bei XML kann der Text durch eigene Tags ergänzt werden.
Der Autor muß allerdings einige Regeln bei der Erstellung seiner Dokumente beachten. Hier wird eine stark vereinfachte Einführung gegeben. Für eine genaue Definition sei der Leser auf die XML Spezifikation[21] verwiesen.:

- XML-Dokumente müssen *well-formed* (wohlgeformt) sein. Dazu können eigene Tags verwendet werden. Es ist nicht notwendig, daß diese in einer DTD erklärt werden.
- Ein XML-Dokument ist *well-formed*, wenn alle seine Elemente ein öffnendes und ein schließendes Tag haben (z.B. `<bold>Text</bold>`), die Attributwerte der Elemente in Anführungszeichen stehen (z.B. `<p align="left">Text</p>`) und die Elemente korrekt ineinander verschachtelt sind.
- Ein XML-Dokument ist zusätzlich *valid*, wenn es zu allen verwendeten Tags dazugehörige DTDs gibt und das gesamte Dokument den darin definierten Regeln entspricht.

Im letzten Abschnitt wurde bereits kurz angedeutet, welche Gründe dafür sprechen, in dieser Umgebung XML als Datenformat zu verwenden. Im Wesentlichen sind das:

- die Möglichkeit, nahezu beliebige Ausgabeformate zu erzeugen, z.B. WML, HTML, PDF
- und die Universalität dieses Formats: Es kann leicht an neue Anforderungen angepaßt und in viele Anwendungen integriert werden.

Viele der verfügbaren XML-fähigen Programme unterstützen lediglich die baumartige Anzeige der logischen Dokumentstruktur mit ihren verschachtelten XML-Tags und den darin enthaltenen Daten. Der Transformationsschritt in die eigentliche „Anzeigesprache“ wird nicht vollzogen.
Daher haben wir einen anderen Ansatz gewählt: Die komplette XML-Verarbeitung erfolgt bereits auf dem Server und nur das fertige Ausgabedokument wird, im richtigen Format, zum Client übermittelt. Cocoon bietet die dafür notwendige Infrastruktur.

5.3.1 Das XML-Verarbeitungsmodell

Im Folgenden sollen die Verarbeitungsschritte vorgestellt werden, die von der Erzeugung des XML-Dokuments bis zur Anzeige beim Client-Programm durchgeführt wer-

[21] http://www.w3.org/TR/REC-xml

den (Abbildung 11). Optionale Teile sind gestrichelt dargestellt.

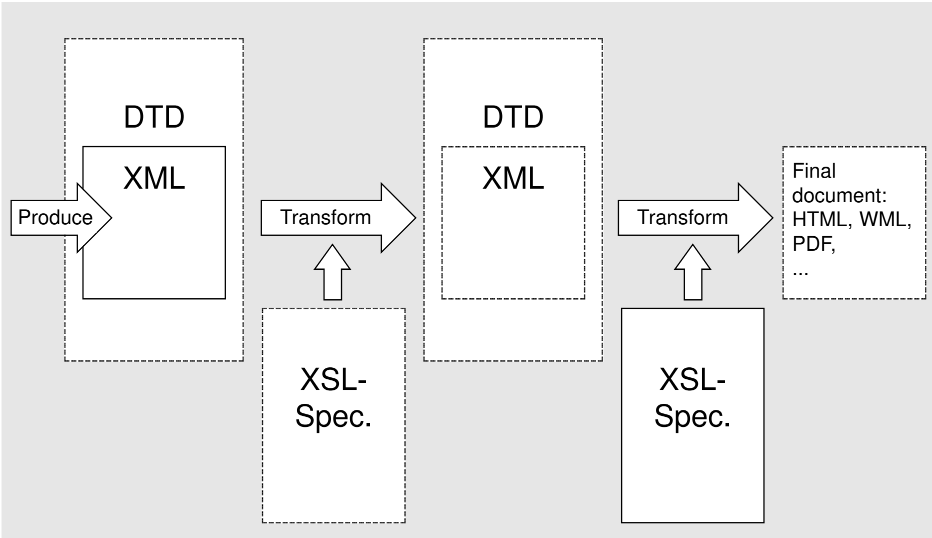

Abbildung 11: Das XML-Verarbeitungsmodell

Zuerst wird ein XML-Dokument von einer Anwendung erzeugt. Dieses Dokument kann einer bestimmten DTD entsprechen. Das ist jedoch nicht zwingend vorgeschrieben. Dieses Dokument kann nun beliebig oft transformiert werden. Diese Transformation wird über ein XSL-Stylesheet[22] (Extensible Stylesheet Language) gesteuert. Die XSL bietet verschiedenste Möglichkeiten XML Dokumente ineinander umzuwandeln. Das entstehende Dokument *kann* natürlich auch wieder einer DTD entsprechen.
Der letzte Schritt ist üblicherweise die Transformation in das „Zielformat“, das der Client darstellen kann.

Es ist natürlich nicht zwingend erforderlich, daß all diese Schritte auf dem Server durchgeführt werden. Wenn ein Client in der Lage ist, die XSL Transformation(en) selbst durchzuführen, kann er das selbstverständlich auch tun. Der derzeitige Stand der XML-Software läßt allerdings keine Alternative zur serverseitigen Verarbeitung zu.

5.3.2 Das Cocoon-Verarbeitungsmodell

Cocoon erweitert das XML-Verarbeitungsschema um die Unterstützung der XML Erzeugung. Dazu wurde von den Entwicklen das Konzept der Extensible Server Pages (XSP)[23] eingeführt. Dies ermöglicht es dem Designer der XML Dokumente, Java-Code in die XML Dateien einzubetten, der zur Laufzeit (d.h. wenn die Seite abgerufen wird)

[22] siehe XSL-Spezifikation: `http://www.w3.org/TR/xsl/`
[23] `http://xml.apache.org/cocoon/xsp.html`

ausgeführt wird. Damit ist es z.B. möglich, die Ergebnisse von Datenbankanfragen dynamisch in das XML Dokument einzufügen.

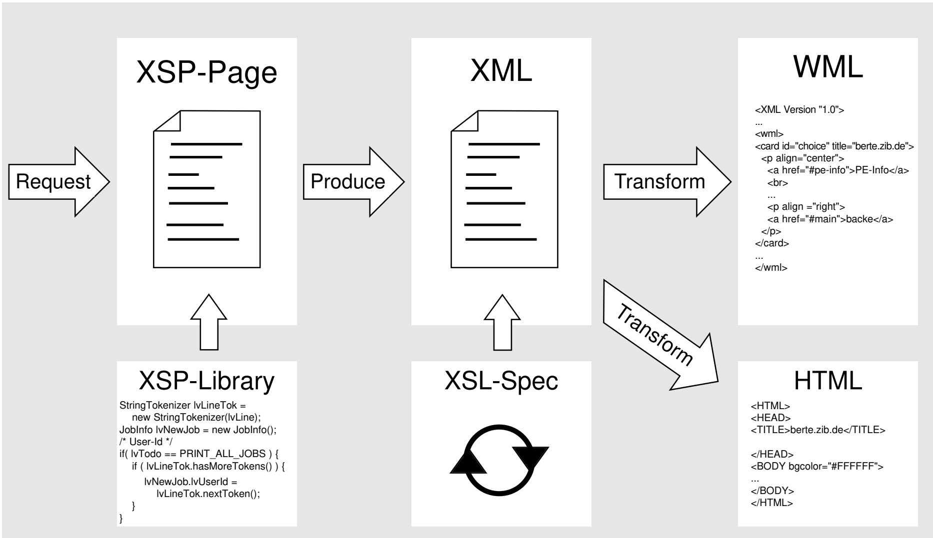

Abbildung 12: Das Cocoon-Verarbeitungsmodell

Die XSPs können außerdem noch modularisiert werden. Oft benötigte Codeteile lassen sich in sogenannten *Tag-Libraries* oder auch *Logicsheets* auslagern. Wie der Name schon suggeriert handelt es sich hierbei um selbstdefinierte Tags, die dann durch den entsprechenden Java-Code ersetzt werden.
Das Ergebnis dieser Vorverarbeitung ist ein XML-Dokument, das sich in das normale XML-Verarbeitungsmodell einfügen läßt. Cocoon unterstützt jedoch auch die weiteren Stufen der XML-Verarbeitung. Es lassen sich verschiedene XSL-Stylesheets auf das Dokument anwenden, bis schließlich das Ausgabeformat erreicht ist, das dann an den Client übermittelt wird.

Da sich unsere Anwendung nicht besonders komplex gestaltete, haben wir auf das Konzept der XSP-Tag-Libraries verzichtet. Die anderen Verarbeitungsschritte aus Abbildung 12 wurden jedoch implementiert. Zur Demonstration der Flexibilität der Architektur wurden auch XSL-Stylesheets zur HTML-Ausgabe erzeugt.

6 Implementation

Ziel unserer Arbeit war es, die Möglichkeiten der mobilen Kommunikation, insbesondere des WAP-Modells, zu untersuchen und anhand einer Beispielanwendung zu demonstrieren. Dafür sollte der Zugriff auf verschiedene Statusinformationen eines Hochleistungsrechners über ein mobiles Gerät implementiert werden. Abbildung 13 verdeutlicht den von uns gewählten Testaufbau.

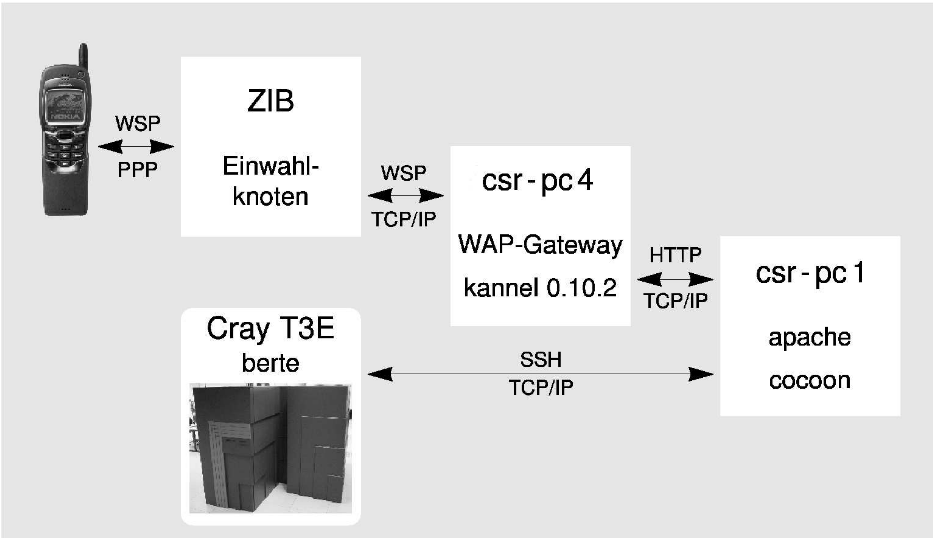

Abbildung 13: Testaufbau des WAP-Service im ZIB

Zum Testen der generierten WML Daten haben wir die WAP-Simulatoren von Nokia (Nokia WAP-Toolkit) und Phone.com (UP.SDK) verwendet. Zeitweise standen uns auch ein Siemens c35i, ein Nokia 7110, sowie ein Sony Z5 zur Verfügung.
In den weiteren Teilen dieses Kapitels werden die einzelnen Komponenten aus der Abbildung 13 und die Kommunikation zwischen ihnen erläutert.

6.1 Verbindungsaufbau

Damit ein mobiles Gerät Daten von einem WAP-Server anzeigen kann, braucht es eine Verbindung zum Internet. Die Netzbetreiber bieten ihren Kunden dafür eigene WAP-Portale an, die auch ein WAP-Gateway beinhalten. Für unsere Anwendung wäre es natürlich möglich gewesen, eines dieser Gateways zu benutzen. Das hätte allerdings verschiedene Nachteile bzw. Risiken mit sich gebracht:

- Die zumindest als sensibel zu betrachtenden Statusinformationen werden unnötigerweise über das Internet übertragen, was das Risiko des Abhörens der Informationen in sich birgt.

- Die Autorisierung und Authentisierung der Benutzer müßte der WWW- bzw. WAP-Server übernehmen. Die dafür zur Verfügung stehenden Mechanismen sind jedoch als äußerst unsicher zu betrachten (HTTP Basic Authentication). Außerdem hätte diese Vorgehensweise nicht unerheblichen Administrationsaufwand auf dem WAP-Server bedeutet (Nutzerkennungen einrichten und verwalten).

Aus den genannten Gründen haben wir uns entschieden, am ZIB einen WAP-Zugang einzurichten. Das Konrad-Zuse-Zentrum stellt für seine Mitarbeiter bereits die Möglichkeit zur Verfügung, sich über ein analoges Modem oder mit einer ISDN-Karte im ZIB einzuwählen, um z.B. auch von zu Hause Zugriff auf den eigenen Arbeitsplatzrechner zu haben.
Auf diesem Einwahlserver wurde eine spezielle Kennung für den WAP-Zugang eingerichtet. Gleichzeitig wurden das Gateway und der WAP-Server so konfiguriert, daß sie nur Verbindungen aus der Domäne `*.zib.de` akzeptieren. Auf diese Art und Weise wurde ein einfaches und trotzdem effektives Sicherheitskonzept eingeführt, da die Authentisierung der Nutzer durch den Einwahlserver erfolgt und die Daten nur über das lokale Netz des ZIB übertragen werden[24].

Ein Nutzer des WAP-Services meldet sich mit seinem mobilen Gerät am Einwahlserver des ZIB an. Dabei wird über die vom GSM-Netz hergestellte Verbindung eine Point-to-Point (PPP)-Verbindung aufgebaut. Damit kann der WAP-User-Agent (in den meisten Fällen ein WML-Browser) über UDP Daten mit dem Gateway austauschen.
Das Gateway war in unserem Fall eine Instanz von Kannel 0.10.2, die auf dem Rechner `csr-pc4` installiert wurde. Die Kommunikation zwischen Einwahlserver und Gateway erfolgt über normales UDP, d.h. der Einwahlserver reicht die UDP-Pakete, die vom mobilen Gerät übermittelt werden einfach an das WAP-Gateway durch. In die andere Richtung funktioniert der Datenaustausch natürlich genauso. Die UDP-Pakete vom Gateway werden durch den Einwahlserver in PPP-Pakete verpackt und zum Mobilgerät geschickt. Aus diesem Szenario ist schon ersichtlich, daß Kannel mit UDP als Bearer-Service konfiguriert wurde. Die Möglichkeit, SMS als Träger zur Kommunikation zu benutzen, kam für uns nicht in Frage, da uns kein SMS-Gateway in die Funknetze zur Verfügung stand.

Für das WAP-Gateway wurde demzufolge folgende Konfiguration gewählt:

- *Software*: Open Source Gateway, **kannel**, Version 0.10.2, kompiliert für Linux 2.2.16

- *Hardware*: PC mit Intel Pentium III Prozessor, 800 MHz, 256 MB Hauptspeicher, Fast-Ethernet Anschluß (100 Mbit/s)

- *Bearer-Service*: UDP

[24]Auf der Funkstrecke können sie natürlich nach wie vor abgehört werden. Hier werden jedoch standardmäßig Verschlüsselungsverfahren verwendet.

Vom Gateway werden die kodierten WSP-Anfragen (siehe Abschnitt 2.6) in normale HTTP-Anfragen umgesetzt und an den WAP-Server weitergereicht. Mit der Bearbeitung dieser Anfragen beschäftigt sich der nächste Abschnitt.

6.2 Der WAP-Service

Wie bereits dargelegt wurde, kann ein gewöhnlicher WWW-Server auch dazu benutzt werden, WML Dateien zu verschicken. In unserem Fall haben wir einen Apache-WWW-Server als WAP-Server eingesetzt. Im Kapitel 5 haben wir bereits erläutert, daß als internes Datenformat XML verwendet wird, welches dann durch entsprechende Transformationsschritte in HTML oder WML umgewandelt wird. Für die XML-Verarbeitung ist Cocoon verantwortlich.

6.2.1 Cocoon

Cocoon ist ein Java-Servlet[25]. Servlets sind Java-Programme, die ähnlich den CGI-Skripten, auf dem WWW-Server ausgeführt werden, wenn eine bestimmte Seite angefordert wird. Sie bieten gegenüber den herkömmlichen Perl- oder PHP-Skripten unter anderem den Vorteil, daß nicht für jede Anfrage ein neuer Prozeß erzeugt werden muß, sondern die einzelnen Anfragen als Threads abgearbeitet werden, d.h. Servlets gehen etwas schonender mit den Ressourcen des WWW-Servers um. Ein weiterer Vorteil der Servlets ist, daß der Programmierer sämtliche Möglichkeiten von Java zur Verfügung hat, z.B. Datenbankzugriff über JDBC (Java Database Connectivity) oder Zugriff auf entfernte Objekte über RMI (Remote Method Invocation) oder CORBA (Common Object Request Broker Architecture; Für weitere Erläuterungen sei der Leser auf die ausführliche Dokumentation unter `http://java.sun.com/products/` verwiesen).
Cocoon wurde nach dem *Reactor*-Designschema konzipiert (siehe Abbildung 14). Es sieht folgende Verarbeitungsschritte vor:

1. Die vom Client kommende Anfrage für eine XML-Seite wird in ein *Request*-Objekt verpackt. Es enthält den Typ des Clients, welche URL angefordert wurde und welcher *Producer* die Anfrage bearbeiten soll.

2. Der *Producer* erzeugt die XML-Daten. Handelt es sich um eine statische XML-Datei, so kann das Dateisystem als Producer betrachtet werden. Dynamische Seiten (XSP siehe Abschnitt 5.3.2) werden vom Cocoon-Framework in einen Producer, d.h. in ein ausführbares Programm übersetzt, welches dann die XML-Daten erzeugt. (Genauer gesagt ist der *Formatter* für die Erzeugung des Producers aus einer XSP zuständig.)

3. Im nächsten Schritt entscheidet der *Reactor*, welcher *Processor* die weitere Verarbeitung der XML-Daten übernehmen soll. Diese Entscheidung wird durch im

[25]siehe Servlet-Dokumentation: `http://java.sun.com/products/servlet/index.html`

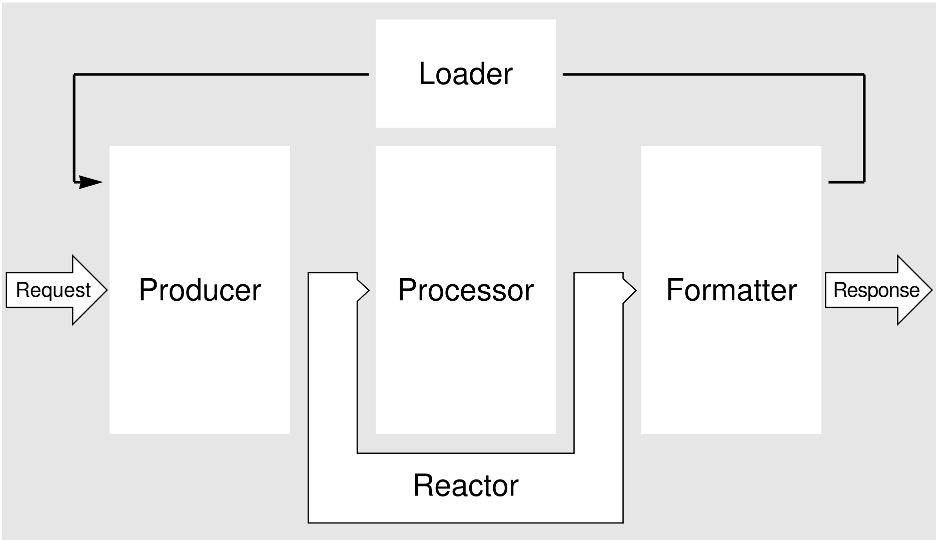

Abbildung 14: Das Cocoon-Verarbeitungsschema

XML-Dokument enthaltene XML-Processing-Instructions[26] gesteuert. Dadurch sind die Verarbeitungsschritte dynamisch konfigurierbar und es werden auch nur notwendige Schritte durchgeführt.

4. Der *Formatter* erzeugt aus der im Speicher angelegten XML-Datenstruktur ein Dokument im vom Client geforderten Format. Er kann jedoch auch ausführbaren Code erzeugen (im Fall von XSPs), welcher dann vom *Loader* wieder in das System eingebracht wird.

5. Wird vom *Formatter* ein ausführbares Programm erzeugt, so wird dieses vom *Loader* wieder geladen und als Producer ausgeführt.

6.2.2 Bearbeitung der Anfragen

Die Konfiguration des WAP-Servers umfaßte folgende Komponenten:

- *Software*: Apache Version 1.3.12, Apache JServ Version 1.1.2 (benötigt zusätzlich ein Java Development Kit und ein Java Servlet DevelopmentKit, es wurden JDK 1.1.8 und JSDK 2.0 installiert) zum Ausführen von Servlets, Cocoon Version 1.7.4

- *Hardware*: PC mit Intel Pentium III Prozessor, 800 MHz, 256 MB Hauptspeicher, Fast-Ethernet Anschluß (100 Mbit/s)

[26]wiederum sei der Leser auf die XML-Spezifikation (`http://www.w3.org/TR/REC-xml`) und die Cocoon-Dokumentation (`http://xml.apache.org/cocoon/guide.html`) verwiesen. Hier soll es genügen zu wissen, daß XML-Processing-Instructions die weitere Verarbeitung steuern.

Es wurden zwei verschiedene Dienste implementiert, die von Rechnern aus der Domäne `*.zib.de` über `http://csr-pc1.zib.de/heidl/index.xml` erreichbar sind:

- **Maschinenstatus**: Diese Seite gibt darüber Auskunft, wie lange `berte` online ist (Zeit seit dem letzten Reboot) und wie stark der Rechner ausgelastet ist. Dafür werden die Anzahl belegter und freier Prozessoren (Prozessoreinheiten - PEs), der belegte Speicher und noch einige andere Informationen ausgegeben.

- **Jobstatus**: Hier bekommt der Anwender zuerst eine Liste mit Namen von Nutzern, die Jobs in der Batch-Queue (siehe Abschnitt 5.2) haben, präsentiert. Danach kann er auswählen, ob er nur Informationen über die Jobs eines Nutzers oder über alle Jobs in der Queue angezeigt bekommen möchte.

Jede Seite wird dabei durch 3 Dateien repräsentiert: Eine XML-Datei, die die Daten enthält und zwei XSL-Dateien, in denen die Transformation der XML-Daten in ein Ausgabeformat (HTML oder WML) beschrieben ist, z.B. `jobstatus.xml`, `jobstatus.xsl`, `jobstatus-wml.xsl`. Den beiden eigentlichen Statusseiten ist noch eine Startseite (`index.xml`) vorgeschaltet, auf der sich der Nutzer für eine der beiden Statusanzeigen entscheiden kann.

Die Verarbeitung eines Requests soll hier am Beispiel der Seite zur Anzeige des Maschinenstatus demonstriert werden:

1. Der WAP-Server (Apache) erhält vom Gateway eine Anfrage für die URL `http://csr-pc1.zib.de/heidl/machine-status.xml`. Wegen der Konfiguration werden alle Anfragen für `.xml` Dateien an das Cocoon-Servlet weitergeleitet.

2. Cocoon überprüft zuerst die syntaktische Korrektheit (Wohlgeformtheit, Gültigkeit siehe Abschnitt 5.3) des XML-Dokuments. Sollte hier ein Fehler in der Dokumentstruktur gefunden werden, wird die weitere Verarbeitung abgebrochen und eine Fehlermeldung ausgegeben. Ist die XML-Datei hingegen fehlerfrei, so wird sie in eine Objektstruktur übersetzt, die im Speicher manipuliert bzw. bearbeitet werden kann.

3. Sollte es sich um eine dynamische Seite handeln (was bei `machine-status.xml` der Fall ist), d.h. eine XSP, so wird das erzeugte DOM (Document Object Model) der Cocoon-XSP-Engine übergeben. Die XSP-Engine ist ein Cocoon *Processor*, der XSP-Seiten verarbeiten kann. Beim ersten Aufruf der Seite wird sie in ein Java-Programm (genauer gesagt einen *Producer*) übersetzt, das die XML-Struktur des Dokuments als DOM erzeugt und den eingebetteten Java-Code ausführt. Dieses Programm wird kompiliert und im Cocoon-Repository (ein spezielles Verzeichnis) als Datei abgelegt. Beim nächsten Aufruf der Seite kann dann die kompilierte Version benutzt werden.

4. Der kompilierte *Producer* wird ausgeführt und erzeugt die XML-Struktur, d.h. die Daten für die weitere Verarbeitung. Um die Daten für den Maschinen-

status zu ermitteln, wird ein Shell-Kommando ausgeführt, das über eine SSH-Verbindung[27] auf `berte` ein Perl-Skript anstößt. Dieses Skript wertet die Ausgabe des Kommandos `grmview` (siehe Abschnitt 5.2) aus und erzeugt eine Ausgabe, die vom Java-Programm leicht einzulesen ist.

5. Die Ausgabe des Perl-Skripts wird in die XML-Struktur eingebunden. Damit ist die Arbeit des Producers (der XSP) beendet. Anhand des erzeugten XML-Dokuments entscheidet nun der *Reactor*, welche weiteren Verarbeitungsschritte durchzuführen sind. In unserem (einfachen) Fall ist nur noch die Transformation in das „Zielformat“ notwendig. Dafür gibt es in `machine-status.xml` zwei XML-Processing-Instructions, die festlegen, daß für bestimmte Clients (die Handys und Handysimulatoren) WML ausgegeben werden muß und daß alle Anderen eine normale HTML Ausgabe erhalten. Die Identifikation des Clienttyps erfolgt anhand des „User-Agent“-Headers in der Anfrage.

6. Je nach Clienttyp nutzt der XSL-*Processor* die Datei `machine-status.xsl` für die HTML Ausgabe oder die Datei `machine-status-wml.xsl` für die WML Ausgabe.

Die erzeugte Ausgabe wird, im Falle von HTML, an den Client geschickt oder, im Falle vom WML, an das Gateway ausgeliefert, welches dann wieder die Übersetzung in den komprimierten Bytecode vornimmt.
Im folgenden Abschnitt soll noch kurz auf die Kommunikation zwischen dem WAP-Server und der SGI/CRAY T3E eingegangen werden.

6.3 Verbindung zu `berte`

Wie im vorigen Abschnitt schon kurz erwähnt, wird die Verbindung zum Rechner `berte` über SSH hergestellt. Da Cocoon mit den gleichen Rechten und auch mit der gleichen User-ID (`wwwroot`) wie der WAP-Server (Apache) ausgeführt wird, wurde für diese User-ID ein SSH-Schlüssel erzeugt, der (wie für solche Schlüssel üblich) in einem Verzeichnis abgelegt wurde, welches nur für diesen Nutzer zugänglich ist. Für die SSH-Verbindung zur T3E wurde die Nutzerkennung `bzfheidl` verwendet. Da das NQS (siehe Abschnitt 5.2) so konfiguriert ist, daß jeder zugangsberechtigte Nutzer auch die Jobs der anderen Nutzer „sehen“ darf, konnte für alle Anfragen (insbesondere die nach den Status der Jobs eines bestimmten Nutzers) diese Nutzerkennung verwendet werden.
Damit für die SSH-Verbindung keine Passworteingabe notwendig ist, wurde der öffentliche SSH-Schlüssel von `wwwroot` in die Datei `authorized_keys` von `bzfheidl` aufgenommen[28].
Auf der SGI/CRAY T3E wurden zwei Perl-Skripte zur einfacheren Bearbeitung der Anfragen implementiert.

[27] Secure Shell
[28] siehe Manual-Page von `ssh`

- `mppstat.pl` ist für die Sammlung der Daten für die Maschinenstatus-Anzeige zuständig. Dazu wird die Ausgabe des Kommandos `grmview` ausgewertet und entsprechend aufbereitet.
- `jobstat.pl` stellt die Daten für die Jobstatus-Seite(n) zur Verfügung. Hier wird auf die Ausgabe des Kommandos `qstat` zurückgegriffen.

6.4 Aufgetretene Probleme

Hier soll kurz geschildert werden, welche Probleme während der Arbeit an diesem Projekt aufgetreten sind und wie sie gelöst wurden.

- **verschiedene WML-Browser**:
  Die von uns zum Testen der erzeugten WML-Seiten benutzten WML-Browser verhielten sich zum Teil sehr unterschiedlich. Insbesondere die Anzeige vom Titel der Cards die Unterstützung von Absatzformatierungen (zentriert usw.) oder von Tabellen bereiteten Probleme. Dabei traten die Probleme sowohl bei den WAP-Simulatoren, als auch bei den realen Handys auf, die wir verwendeten. Dadurch wurde die Entwicklung der WML-Vorlagen sehr erschwert. Letzlich haben wir uns auf ein Minimallayout zurückgezogen, welches auf allen von uns verwendeten Browsern zumindest lesbar ist.
- **WTLS-Implementation für Kannel**:
  Da Kannel sich noch im Entwicklungsstadium befindet, gibt es zur Zeit von den Kannel-Entwicklern selbst keine Unterstützung für die Verschlüsselung der Verbindungen (siehe Abschnitt 4.3.2 bezüglich der Erweiterungen von 3ui.com). Dies ist als ein Sicherheitsrisiko zu betrachten. Deshalb haben wir ein anderes, restriktiveres Sicherheitskonzept etabliert (siehe Abschnitt 6.1).

7 Zusammenfassung

Es ist uns gelungen, einen WAP-Service für die Anzeige von Status-Informationen einer SGI/CRAY T3E zu implementieren. Die entwickelte Applikation ist durch die gewählte Architektur und die Verwendung von XML als internes Dokumentformat so flexibel gestaltet, daß sie ohne großen Aufwand an neue Anforderungen (Zugriff auf einen anderen Höchstleistungsrechner, Erzeugung eines anderen Ausgabeformats) angepaßt werden kann. Da uns keine *funktionstüchtige* WTLS-Implementation für das Gateway zur Verfügung stand, mußten wir auf die Implementation einer *Jobsteuerung* auf Grund von Sicherheitsbedenken verzichten. Zur Zeit werden nur Informationen über den aktuellen Status der Maschine inklusive der laufenden bzw. auf ihre Ausführung wartenden Jobs zur Verfügung gestellt. Allerdings traten bei der Entwicklung verschiedene Probleme auf, die sich unterschiedlichen Ursachen zuordnen lassen.
Die WAP-Spezifikation räumt den Implementatoren zum Teil sehr große Freiheiten bei der Umsetzung einzelner Elemente ein. Daraus resultieren z.B. WML-Browser, die sich in der Anzeige bestimmter Elemente stark unterscheiden. Dies erschwert natürlich die Entwicklung von WML-Seiten, da bei der Bereitstellung eines WAP-Dienstes ein einheitliches Layout gewünscht ist. Diese „Inkompatibilitäten“ gehen so weit, daß bestimmte Informationen von einigen Browsern angezeigt, von anderen aber einfach weggelassen werden.
Um nicht für jeden WML-Browser eigene WML-Seiten generieren zu müssen, mußten wir die Komplexität der generierten Seiten stark einschränken, was zu einem recht einfach gehaltenen, aber auf allen getesteten Browsern funktionierenden Layout führte.
Eine weitere Schwierigkeit war die sehr unzureichende Debug-Unterstützung der getesteten Handys. Fehlermeldungen wie:*„Unknown Content Type“* oder *„Es ist ein Fehler aufgetreten.“* sind für einen Entwickler nicht besonders hilfreich. Von den Handys, die uns zeitweise zur Verfügung standen (Siemens C35i, Nokia 7110, Sony Z5) ist es uns nur mit zwei Modellen (Siemens und Nokia) gelungen, eine Verbindung zum ZIB-Einwahlknoten aufzubauen. Das Sony Z5 hatte offenbar Probleme beim Verbindungsaufbau, die wir wegen der unzureichenden Fehlermeldung (*„Es ist ein Fehler aufgetreten.“*) und fehlendem Zugriff auf die Logfiles des Einwahlservers auch nicht beheben konnten.

Abschließend läßt sich feststellen, daß die WAP-Technologie trotz der fortgeschrittenen Standardisierung noch in den Kinderschuhen steckt. Die Einschränkungen, denen die Datenübertragung in den aktuellen Funknetzen unterworfen ist, und die sich durch die beschränkten Möglichkeiten der Eingabe und Anzeige in den aktuellen Handys ergeben, lassen alle Dienste, die über eine reine Statusanzeige hinausgehen illusorisch erscheinen.
Es bleibt zu hoffen, daß die neuen Netztechnologien, wie GPRS und UMTS mit ihren höheren Übertragungsraten, neue Geräte mit vollständigen, einheitlichen Implementationen des WAP-Standards und aufkommende Entwicklungen, wie zum Beispiel Spracherkennungssysteme, die Leistungsfähigkeit und den Bedienkomfort der mobi-

len Geräte erheblich steigern werden. Das würde auch einen Leistungssprung in den WAP-Angeboten ermöglichen.

A Glossar

Glossar

Bearer Als Bearer wird die Schnittstelle zwischen WAP und physikalischen Netzen wie GSM- oder TCP/IP-Netzen bezeichnet.

CB Durch Cellbroadcast lassen sich Broadcastnachrichten verschicken. Alle empfangsbereiten Geräte die einen entsprechenden Kanal aktiviert haben und sich innerhalb des Ausstrahlungsgebietes befinden erhalten dieselbe Nachricht. Die CB-Nachrichtengröße beträgt nur 93 Zeichen. Es lassen sich allerdings sogenannte Macromessages aus bis zu maximal 15 Messages zusammensetzen. Zur Datenübertragung läßt sich dieses Verfahren selbstverständlich ebenfalls einsetzen.

CDMA „Code Division Multiple Access" ist der vorherschende Mobilfunkstandard in den USA.

CSD Circuit Switched DATA ist ein Protokoll für Handys, das bei der Datenkommunikation eingesetzt wird.

DECT „Digital Enhanced Cordless Telekommunikations" ist das Gegenstück zu GSM für Übertragungen in geschlossenen Gebäuden. DECT und GSM wurden so konzipiert, daß sich beide zu einem neuen Gerätetyp verbinden lassen. (weitere Informationen unter:
`http://www.dect.ch/aboutdect/whatis.htm`)

DTD Eine Document Type Definition beschreibt, welche Tags in einem XML-Dokument erlaubt sind, welche Attribute sie enthalten dürfen und wie sie ineinander geschachtelt werden können.

ECMAScript ECMAScript ist eine von der European Computer Manufacturers Association (ECMA) als Standard ECMA-262 veröffentlichte Scriptsprache.

EDGE Das in der letzten Zeit entwickelte „Enhanced Data-Rates for GSM Evolution" stellt eine auf einem neuen Modulationsverfahren basierende Erweiterung der GSM-Dienste dar. Damit wird eine maximale Datenrate von 384 Kbps möglich sein.

Gateway Ein Gateway ermöglicht die Kommunikation von Rechnern die sich in unterschiedlichen Netzen befinden.

GPRS „General Paket Radio Service" ist die letzte bis jetzt mögliche Ausbaustufe von GSM. Es läßt die Nutzung von bis zu 8 Funkzeitschlitzen (max. 171,2 Kbps) zu und unterstützt 4 Dienstgüteklassen. Durch paketorientierte Übertragungstechnik ist der Mobilteilnehmer ständig mit dem Internet verbunden. Dadurch entfallen die bei GSM recht aufwendigen Verbindugsaufbau- bzw. abbauzeiten.

GSM GSM ist die Abkürzung für „Global System of Mobile Communikation". GSM ist der europäische Standard für digitale, drahtlose Kommunikation. Pro Kanal ist eine maximale Übertragungsrate von 9.6 Kbps vorgesehen. In neueren GSM Netzen sind auch 14.4 Kbps möglich.

HSCSD Durch „High-Speed Circuit Switched Data" wird eine auf GSM basierende Kanalbündelung ermöglicht. Die Softwarelösung steigert die Übertragungsrate in einem Kanal von 9.6 Kbps auf 14.4 Kbps. Durch das Zeitmultiplexen von maximal 4 Funkkanälen läßt sich die Bandbreite damit auf 57.6 Kbps steigern. In HSCSD wird, wie bei der Sprachkommunikation, leitungsvermittelnde Technik eingesetzt.

HTML HTML ist eine auf SGML basierende Scriptsprache. Sie wurde für die Veröffentlichung von Dokumenten im Internet entwickelt.

IP-Protokoll Der Internet Protokoll ist teil der TCP/IP-Protokollfamilie. Es definiert Regeln und Vereinbarungen nach denen der Informationsfluß in einem Komminikationssystem gesteuert werden kann.

ISDN Integrated Services Digital Network ist der digitale Telefondienst in Deutschland. Es stehen zwei Basiskanäle mit jeweils 64Kbps und ein Steuerkanal mit 16Kbps zur Verfügung.

Multiplexen Die Mehrfachbelegung eines Kommunikationskanals wird als „Multiplexing" bezeichnet. Dies läßt die gleichzeitige Übertragung mehrerer Signale zu, obwohl nur eine physikalische Leitung vorhanden ist. Es wird zwischen Zeit- und Frequenzmultiplexing unterschieden.

NQS Das Network Queuing System ist ein Verwaltungssystem für Prozesse, die in Warteschlangen eingereiht sind. Es ist in der Lage die Verwaltung der Warteschlangen für ein ganzes Netzwerk von Rechnern zu übernehmen.

OSI-Referenzmodell „Open Systems Interconnect" ist ein offenes 7-Schichtenmodell für computervermittelte Kommunikation. Die Problembereiche werden hierachisch in verschiedene Schichten unterteilt die aufeinander aufsetzen. Das OSI-Modell besteht aus sieben Schichten die alle wesentlichen Dienste und Funktionen definieren.

Pager Pager sind kleine Geräte die SMS-Nachrichten empfangen und darstellen können.

PDA PDA heißt „personal digital assistent". Es handelt sich dabei um kleine Computer im Taschenformat. Sie werden im deutschsprachigen Raum auch als Handheldcomputer bezeichnet.

PDF Das Portable Document Format ist ein von Adobe Systems Incorporated (`http://www.adobe.com`) entwickeltes Format für elektronische Dokumente.

PPP Point to Point Protocol: Protokoll zur Übertragung von Daten über serielle Leitungen. Es erlaubt die gleichzeitige Verwendung verschiedener Netzwerkprotokolle (z.B. TCP/IP, IPX) und bietet verschiedene Authentisierungsmechanismen.

QoS „Quality of Service" ist die garantierte Verfügbarkeit von Bandbreite als Qualitätsmerkmal paketvermittelter Leitungen und zukünftig GPRS.

SAP An Service Access Points werden Dienste in Schichtenmodellen bereitgestellt. Die SAP's einer Schicht n sind die Stellen, an denen die Schicht n+1 auf das Dienstangebot zugreifen kann. Jeder SAP hat eine Adresse die ihn eindeutig auszeichnet.

SGML Die Standard Generalized Markup Language ist eine sehr komplexe Metasprache zur Generierung von Auszeichnungssprachen. Seit 1986 ist diese standardisiert. Ziel war es unter anderem, die Unabhängigkeit von Inhalt und Layout zu gewährleisten. HTML ist solch eine Auszeichnungssprache und somit eine Anwendung von SGML.

SMS Der Short Message Service ist ein auf GSM basierender Dienst zur Übertragung von Kurznachrichten. Diese dürfen maximal 160 Zeichen lang sein. Die maximale Datenübertragungsrate liegt bei 14,4 Kbps. Der Service läßt sich nutzen indem man für die Datenübertragung Pakete von 160 Zeichen überträgt.

SSH Secure Shell ist ein Ersatz für RSH (Remote Shell) zum Starten einer Shell auf einem entferten Rechner. Bei SSH werden im Gegensatz zu RSH die übertragenen Daten verschlüsselt.

SSL SSL (Secure Socket Layer) ist ein als sicher zu betrachtendes Übertragungsverfahren das von Netscape entwickelt wurde, um sichere Übertragung über das Internet zu gewährleisten. SSL benutzt eine privaten Schlüssel um die zu übertragenen Daten zu verschlüsseln.

UDP User Datagram Protocol: Netzwerkprotokoll zur unbestätigten verbindungslosen Übertragung von Daten. Setzt auf IP auf.

UMTS Universal Mobile Telecommunikation System oder auch Wideband Code Division Multiple Access (W-CDMA) ist der neue Mobilfunkstandard der GSM ablösen wird. UMTS bietet eine deutlich höhere Bandbreite pro Kanal (5MHz statt 200kHz bei GSM) und verspricht so unter günstigen Bedingungen Datenraten bis zu 2Mbps.

Voice-over-IP Bei der IP-Telefonie werden die Sprachsignale in digitalen Datenpaketen über das Kommunikationsnetz geschickt.

W-CDMA siehe UMTS

WAP WAP steht für Wireless Application Protokoll und ist eine Technik, um Internet-Inhalte und -Dienste auch auf (mobilen) Geräten mit kleinen Bildschirmen, insbesondere auf Handys, anzubieten.

WDP Der Wireless Datagram Protocol ist für die Kommunikation zwischen Bearer und den darüber liegenden Schichten zuständig.

WML Die Wireless Markup Language ist eine in XML definierte Beschreibungssprache zur Darstellung von Internetinhalten auf Handys und anderen Kleinstgeräten.

WMLScript Ähnlich wie Javascript ist WMLScript eine Scriptsprache zur Beschreibung von Abläufen in Web-Seiten. WMLScript ist jedoch den Bedürfnissen von WML angepasst.

WSP Der Wireless Session Protocol sorgt für die Bereitstellung von verbindungsorientierten und verbindungslosen Services.

WTLS Die Funktion der Wireless Transport Security besteht in der Sicherung von Datenintegrität, Privatsphäre und Authentifizierung.

XML Die Extensible Markup Language ist ein Dokumentverarbeitungsstandard, der vom World Wide Web Consortium unter Anderem für die Generierung dynamischer Inhalte im WWW vorgeschlagen wurde.

XSL Die Extensible Stylesheet Language ist eine Sprache zur Beschreibung von Stylesheets. Ein Stylesheet spezifiziert die Präsentation einer Klasse von XML-Dokumenten durch die Beschreibung der Umwandlung einer Instanz dieser Klasse in das Ausgabeformat. Sie wird von der XSL Working Group des W3C (World Wide Web Consortium) entwickelt.

XSP Eine Extensible Server Page ist ein XML-Dokument, welches Java-Code enthält, der zur Laufzeit (beim Abruf der Seite) ausgeführt wird und zum Beispiel Daten aus einer Datenbank in das Dokument einbettet. Auf diese Art und Weise können XML-Dokumente dynamisch erzeugt werden.

Zeitschlitz Ein „Zeitschlitz“ entsteht durch die Aufteilung der Übertragungszeit in Zeiteinheiten. Die Kanäle werden in die einzelnen Zeiteinheiten sortiert.

ZIB Das Konrad-Zuse-Zentrum für Informationstechnik Berlin ist eine außeruniversitäre Forschungseinrichtung des Landes Berlin. Es betreibt in enger, fächerübergreifender Kooperation mit den Hochschulen und wissenschaftlichen Einrichtungen in Berlin Forschung und Entwicklung auf dem Gebiet der Informationstechnik.

B Literatur

Literatur

[1] Brett McLaughlin, *JAVA and XML*, O'Reilly, Cambridge, 2000

[2] Diego Kuonen, Armin Roehrl, Stefan Schmiedl und Thomas Verasani *WAPux - der moderne Pinguin*, Linux Magazin, 5/2000

[3] Lars Röwekamp *Handy HTML*, heise, iX, 2/2000, S.52

[4] Lars Röwekamp *Dynamik mobil*, heise, iX, 3/2000, S.192

[5] Nokia *Nokia Wap Toolkit - Users Guide V2*, Nokia, 6/2000

[6] 3ui.com *WAP WTLS layer Users Guide for Kannel v0.8*, 3ui.com, 2000

[7] Lars Wirzenius, Kalle Marjola *Kannel: Open Source WAP and SMS Gateway 0.10.3, Architecture and Design*, wapit, 10/2000

[8] Rico Keller, Peter Angehrm *Datenbankanfragen via WAP*, HSR, 6/99

[9] Dr. Dirk Nikolai, Klaus Daniel, Dr. Edgar Kühn *Turbolader für Funk-Bits*, heise, ct, 19/2000, S. 190

[10] Duŝan Ẑivadinović *Pakete per Funk*, heise, ct, 21/2000, S. 152

[11] Lars Röwekamp *Drahtlos die Dritte - WAP V1.2*, heise, iX, 5/2000, S.104

[12] Stefan Müller-Wilken *Gateway to Heaven*, heise, iX, 5/2000, S.134

[13] Bernhard Bablok *XML: Jedem sein Format*, linux-magazin, 9/1999

[14] Stefan Minert, Rainald Menge *XML verpuppt*, heise, ct, 10/2000, S. 222

[15] Henning Behme *dummy*, heise, iX, 4/2000

[16] Robert Eckstein, *XML Pocket Reference*, O'Reilly, Cambridge, 1999

[17] Bernhard Bablok, *Eingesponnen - XML für Server mit Cocoon*, Linux Magazin 10/2000, S. 144

[18] Duŝan Ẑivadinović *Drahtlose Mehrspur-Infobahn*, heise, ct, 22/2000, S. 214

[19] Winfried Dulz *WAP - Wireless Application Protocol*, Informatik Spektrum, 23/2000, S. 271