



TU Braunschweig
Institut für Betriebssysteme
und Rechnerverbund



Verteilte Systeme

Prof. Dr. Stefan Fischer

Kapitel 12: Mobilität

Überblick

- Die folgenden Folien stammen zu großen Teilen von Herrn Prof. Dr. Jochen Schiller, Berlin – herzlichen Dank!!.
- Inhalt:
 - Motivation – warum Mobilität in verteilten Systemen?
 - Mobile Technik – Netze, Rechner und Dienste
 - Mobilität in Anwendungen am Beispiel WWW
 - HTTP/HTML
 - WAP und iMode
 - J2ME

Computer für das nächste Jahrhundert?

- Computer sind integriert
 - klein, billig, beweglich, austauschbar - nicht mehr als eigenständige Einheit erkennbar
- Technik tritt in den Hintergrund
 - Computer erkennen selbst wo sie sind und passen sich an
 - Computer erkennen wo welcher Benutzer ist und verhalten sich entsprechend (z.B. Weiterleiten von Gesprächen, Fax)

Fortschritte in der Technik

- höhere Rechenleistung auf kleinerem Raum
- flache, leichte Anzeigen mit niedriger Leistungsaufnahme
- neue Schnittstellen zum Benutzer wg. kleiner Abmessungen
- mehr Bandbreite pro Kubikmeter
- vielfältige drahtlose Netzschnittstellen: lokale drahtlose Netze, globale Netze, regionale Telekommunikationsnetze etc.
(„Overlaynetzwerke“)

Mobilkommunikation

- Zwei Aspekte der Mobilität:
 - *Benutzermobilität*: Der Benutzer kommuniziert (drahtlos) “zu jeder Zeit, an jedem Ort, mit jedermann.”
 - *Gerätemobilität*: Ein Endgerät kann zu einer beliebigen Zeit, an einem beliebigen Ort im Netz angeschlossen werden.

- Wireless Mobile Beispiele



stationäre Arbeitsplatzrechner
Notebook im Hotel
Funk-LANs in nicht verkabelten Gebäuden
Personal Digital Assistants (PDA)

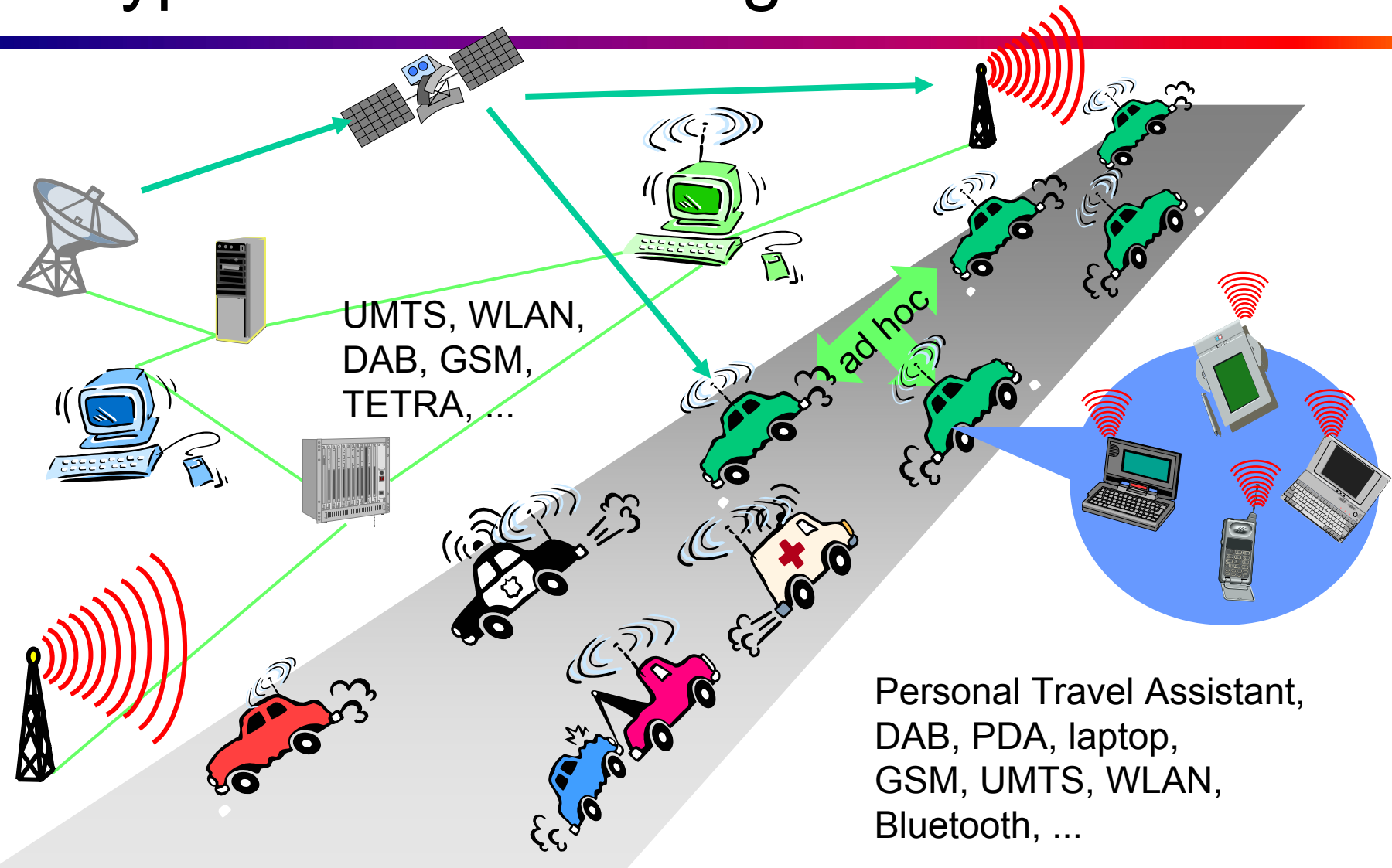
Festnetzintegration

- Der Wunsch nach mobiler Datenkommunikation schafft den Bedarf zur Integration von drahtlosen Netzen in bestehende Festnetze:
 - im lokalen Bereich: Standardisierung von IEEE 802.11, ETSI (HIPERLAN)
 - im Internet: Die Mobile IP-Erweiterung
 - im Weitverkehrsbereich: Anbindung an ISDN durch GSM

Anwendungen I

- Fahrzeuge
 - Empfang von Nachrichten, Straßenzustand, Wetter, Musik via DAB
 - persönliche Kommunikation über GSM
 - Positionsbestimmung über GPS
 - lokales Netz mit Fahrzeugen in der Umgebung zur Vermeidung von Unfällen, Leitsystem, Redundanz
 - Fahrzeugdaten (z.B. bei Linienbussen, ICE) können vorab in eine Werkstatt übermittelt werden, dann schnellere Reparatur
- Notfälle
 - Übermittlung von Patientendaten ins Krankenhaus vor der Einlieferung, aktueller Stand der Behandlung, Diagnose
 - Ersatz der festen Infrastruktur bei Erdbeben, Orkanen, Feuer etc.

Typische Anwendung: Straßenverkehr



Anwendungen II

- Handelsvertreter
 - direkter Zugriff auf Kundendaten in der Zentrale
 - konsistente Datenhaltung über alle Mitarbeiter
 - mobiles Büro
- Ersatz eines Festnetzes
 - abgeschiedene Messstationen, z.B. Wetter, Flusspegel
 - Flexibilität bei Messeständen
 - Vernetzung historischer Gebäude
- Freizeit, Unterhaltung, Information
 - Internet-Anschluss im Grünen
 - tragbarer Reiseführer mit aktuellen Informationen vor Ort
 - Ad-hoc Netzwerke für Mehrbenutzerspiele

Ortsabhängige Dienste

- **Umgebungsbewusstsein**
 - welche Dienste, wie Drucker, Fax, Telefon, Server etc. existieren in der lokalen Umgebung
- **Nachfolgedienste**
 - automatische Anrufweiterleitung, Übertragung der gewohnten Arbeitsoberfläche an den aktuellen Aufenthaltsort
- **Informationsdienste**
 - „push“: z.B. aktuelle Sonderangebote im Supermarkt
 - „pull“: z.B. wo finde ich Pizza mit Thunfisch
- **Nachfolgen der Unterstützungsdienste**
 - Caches, Zwischenberechnungen, Zustandsinformation etc. „folgt“ dem mobilen Endgerät durch das Festnetz
- **Privatheit**
 - wer soll Kenntnis über den Aufenthaltsort erlangen

Mobile Endgeräte

Pager

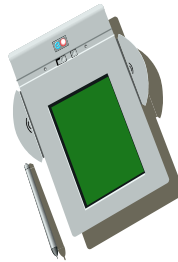
- nur Empfang
- sehr kleine Anzeigen
- einfache Textnachrichten



Sensoren,
embedded
systems

PDA

- einfache Grafikanzeigen
- Handschrifterkennung
- vereinfachtes WWW



Mobiltelefone

- Sprache, Daten
- einfache Textanzeigen

Palmtops

- kleine Tastatur
- einfache Versionen der Standardprogramme

L e i s t u n g

Auswirkungen der Endgeräteportabilität

- Leistungsaufnahme
 - begrenzte Rechenleistung, niedrigere Qualität der Anzeigen, kleinere Festplatten durch begrenzte Batterieleistung
 - CPU: Leistungsaufnahme $\sim CV^2f$
 - C: interne Kapazitäten, durch Hochintegration verringert
 - V: Betriebsspannung, wird kontinuierlich abgesenkt
 - f: Taktfrequenz, kann z.B. zeitweise gesenkt werden
- Datenverlust
 - muss von vornherein mit eingeplant werden (z.B. Defekte)
- Stark eingeschränkte Benutzungsschnittstelle
 - Kompromiss zwischen Fingergröße und Tragbarkeit
 - evtl. Integration von Handschrift, Sprache, Symbolen
- Eingeschränkter Speicher
 - Massenspeicher mit beweglichen Teilen nur begrenzt einsetzbar
 - Flash-Speicher als Alternative

Drahtlose vs. Festnetze

- Höhere Fehlerraten durch Interferenzen
 - Einstrahlung von z.B. Elektromotoren, Blitzschlag
- Restriktivere Regulierungen der Frequenzbereiche
 - Frequenzen müssen koordiniert werden, die sinnvoll nutzbaren Frequenzen sind schon fast alle vergeben
- Niedrigere Übertragungsraten
 - lokal einige Mbit/s, regional derzeit z.B. 9,6kbit/s mit GSM
- Höhere Verzögerungen, größere Schwankungen
 - Verbindungsaufbauzeiten via GSM im Sekundenbereich, auch sonst einige hundert Millisekunden
- Geringere Sicherheit Luftschnittstelle ist für jeden einfach zugänglich, Basisstationen können vorgetauscht werden
- sichere Zugriffsverfahren wegen shared medium

WWW und Mobilität

- Protokoll (HTTP, Hypertext Transfer Protocol) und Sprache (HTML, Hypertext Markup Language) des Web wurden nicht für mobile Anwendungen entworfen, daraus resultieren zahlreiche Probleme!
- Typische Datengrößen
 - HTTP request: 100-350Byte
 - Antworten typ. <10kByte, Kopf 160Byte, GIF 4,1kByte, JPEG 12,8kByte, HTML 5,6kByte
 - aber auch viele sehr große Dateien, nicht vernachlässigbar

Charakter des WWW

- Das Web ist kein Dateisystem!
 - Web-Seiten sind nicht nur einfach Dateien, die geladen werden
 - statische und dynamische Inhalte, Interaktion mit Servern über Formulare, Inhaltstransformation, push-Strategien
 - zahlreiche Querverweise, kein transparenter Datenstrom, automatisches Nachladen
 - ein Klick auf eine Datei kann große Konsequenzen haben!

WWW-Beispiel

- Anfrage an Port 80

```
GET / HTTP/1.0
```

- Antwort vom server

```
HTTP/1.1 200 OK
```

```
Date: Fri, 06 Nov 1998 14:52:12 GMT
```

```
Server: Apache/1.3b5
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Institut für Telematik</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#ffffff">
```

```

```


HTTP 1.0 und Mobilität I

- Eigenschaften
 - zustandslos, Client/Server, Anfrage/Antwort
 - erfordert verbindungsorientiertes Protokoll, eine Verbindung pro Anfrage (meist TCP eingesetzt)
 - primitive Caching- und Sicherheitsmodelle
- Probleme
 - entworfen für große Bandbreiten und geringe Verzögerungen
 - große und redundante Protokollköpfe (lesbar für Menschen, kein gespeicherter Zustand erfordert stets umfangreiche Köpfe)
 - unkomprimierte Übertragung von Inhalten
 - mit TCP
 - großer Overhead pro Anfrage durch 3-Wege-Handshake
 - slow-start Problematik
 - DNS-lookup durch Client erzeugt zusätzlichen Verkehr

HTTP 1.0 und Mobilität II

- Caching
 - oftmals von Dienst Anbietern *verhindert*, damit Benutzungsstatistiken erstellt werden können
 - dynamische Objekte können nicht im Cache abgelegt werden
 - zahlreiche Zähler, Datum, Personalisierungen, ...
 - Mobilität macht oft Caches unmöglich
 - Sicherheitsproblematik
 - wie verhält sich SSL im Zusammenhang mit Proxies
- POST (d.h. Senden *zum* Server)
 - kann i.allg. nicht gepuffert werden, damit problematisch falls gerade abgekoppelt
- Genügend ungelöste Fragen!

HTML und mobile Endgeräte

- HTML
 - entworfen für Endgeräte höherer Leistung, Farbdisplay, Maus
 - Optimierung der „Ästhetik“, nicht des Kommunikationsaufwandes
- Mobile Endgeräte
 - haben oft nur kleine Anzeigen niedriger Auflösung und sehr eingeschränkte Eingabemöglichkeiten
- weitere „Features“
 - animierte GIFs, Java AWT, Frames, ActiveX Controls, Shockwave, Filmclips, Audio, ...
 - heute schon werden Echtfarbendarstellung, Multimedia-Unterstützung, hohe Auflösungen beim Entwurf von Web-Seiten vorausgesetzt
- Die Mehrzahl der HTML-Seiten enthält Fehler
 - Browser müssen sehr leistungsfähig sein, um diese Fehler „auszugleichen“
- **Heterogenität der Endgeräte wird weitgehend ignoriert!**
 - also werden z.B. auch große Bilder automatisch zum Mobiltelefon mit geringer Auflösung unter hohen Kosten übertragen !

Ansätze in Richtung Web für mobile Endgeräte

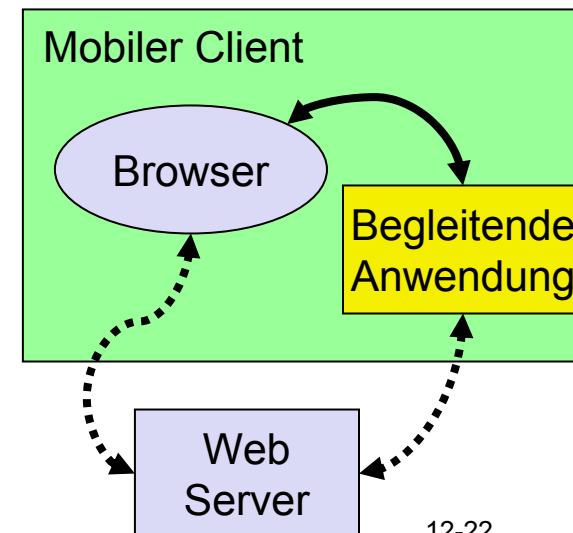
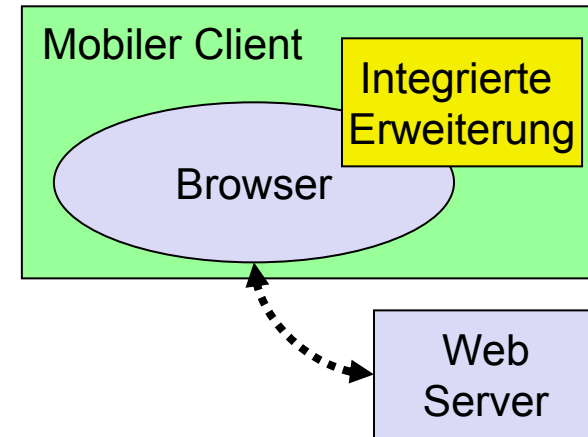
- Anwendungs-Gateways, erweiterte Server
 - einfache Clients, Vorberechnungen im Festnetz
 - Kompression, Filterung, Inhaltsextraktion
 - automatische Anpassung an Netzwerkgegebenheiten
- Beispiele
 - Skalierung von Grafiken, Farbreduktion, Änderung des Dokumentenformats (z.B. PS nach TXT)
 - Detailbetrachtungen, Ausschnitte, Zoom
 - Extraktion von Überschriften, Zusammenfassung des Textes
 - HDML: einfache, HTML-ähnliche Sprache mit speziellem Browser
 - HDTP: passendes Protokoll, entwickelt von Unwired Planet
- Probleme
 - proprietäre Ansätze, erfordern eine Menge spezieller Zusätze für Browser
 - große Geräteheterogenität erschwert Ansätze

Mobilitätsunterstützung

- HTTP/1.1
 - client/server nutzen die selbe Verbindung für mehrere Anfragen/Antworten
 - mehrere Anfragen zu Beginn einer Sitzung möglich, die Antworten kommen dann in der Anfragereihenfolge
 - erweiterte Zwischenspeicherung von Antworten möglich (falls Antworten identisch)
 - semantische Transparenz ist nicht immer machbar: abgetrennt, geringe Leistung, Verfügbarkeit
 - neue Tags und Optionen, um das Caching besser zu steuern (public/private, max-age, no-cache etc.)
 - Konsistenz kann auf Nutzerwunsch abgeschwächt werden
 - Kodierung/Kompression, Integritätsprüfung, Sicherheit von Proxies, Authentisierung, Authentifikation...

Systemunterstützung für WWW im Mobilien I

- **Erweiterte Browser**
 - Pre-fetching, caching und abgekoppelte Nutzung
 - z.B. Internet Explorer
- **Zusätzliche, begleitende Anwendung**
 - Pre-fetching, caching und abgekoppelte Nutzung
 - z.B. ursprünglicher WebWhacker



Systemunterstützung für WWW im Mobilien II

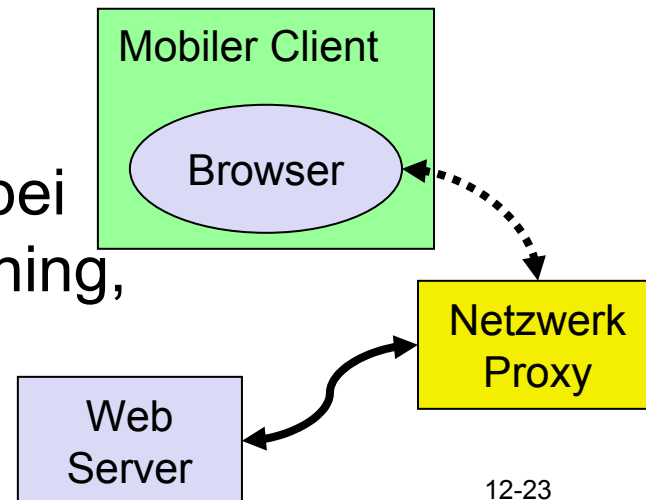
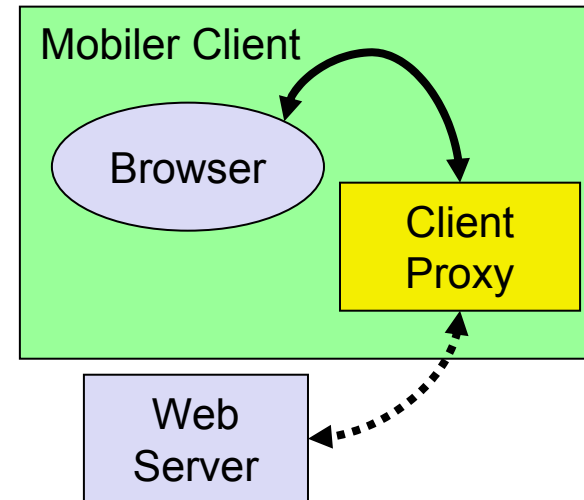
- Client Proxy

- Pre-fetching, caching und abgekoppelte Nutzung
- z.B. Caubweb, TeleWeb, Weblicator, WebWhacker, WebEx, WebMirror,

...

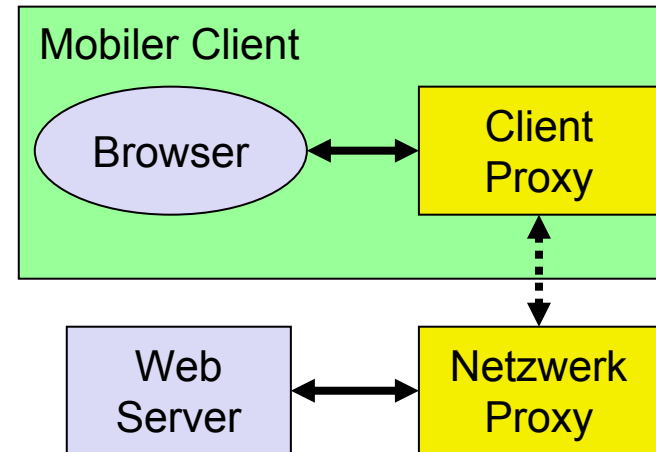
- Netzwerk Proxy

- Adaptive Inhaltstransformation bei schlechter Verbindung, pre-fetching, caching
- z.B. TranSend, Digestor

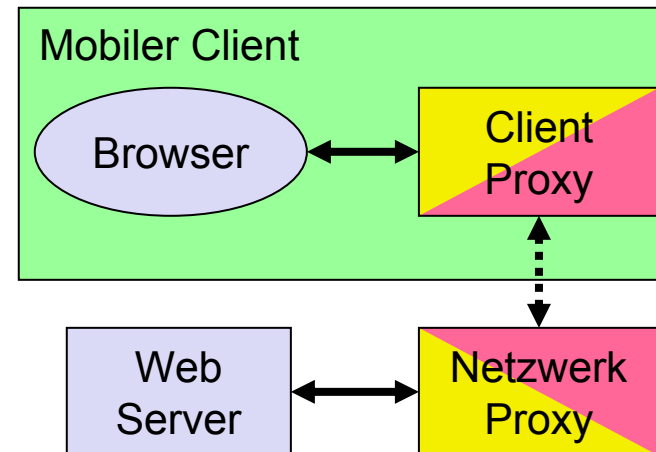


Systemunterstützung für WWW im Mobilien III

- Client und Netzwerk Proxy
 - Kombination der Nutzen plus Protokollvereinfachungen
 - z.B. MobiScape, WebExpress



- Spezielles Netzwerk Subsystem
 - Adaptive Inhaltstransformation bei schlechter Verbindung,
 - pre-fetching, caching
 - z.B. Mowgli



i-mode

- Zugang zu Internet-Diensten in Japan/NTT DoCoMo
 - Dienste
 - Email, Kurznachrichten, Web, Austausch von Bildern, Horoskope, ...
 - Sehr großer Erfolg, über 25 Mio. Nutzer (Juni 2001)
 - Für viele ein PC-Ersatz, PC-Dichte in Japan relativ niedrig
 - Ermöglichte für viele ersten Internet-Kontakt
 - Sehr einfach und bequem in der Handhabung
 - Technik
 - 9,6 kbit/s, paketorientiert
 - Compact HTML, keine Sicherheit

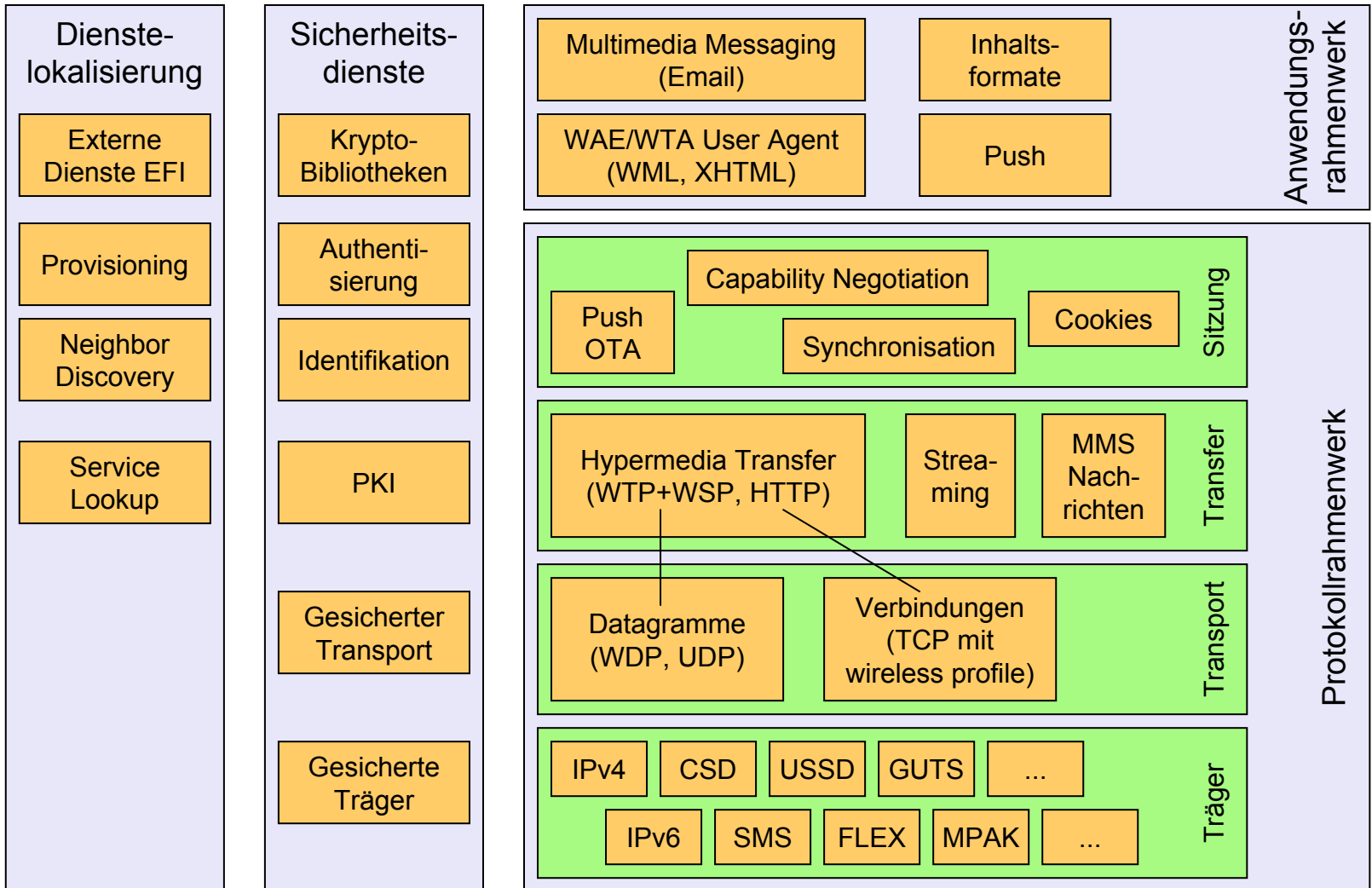


| | |
|--------|-------|
| i-mode | |
| Email | CHTML |
| HTTP | |
| TCP | |
| IP | |
| PDC-P | |

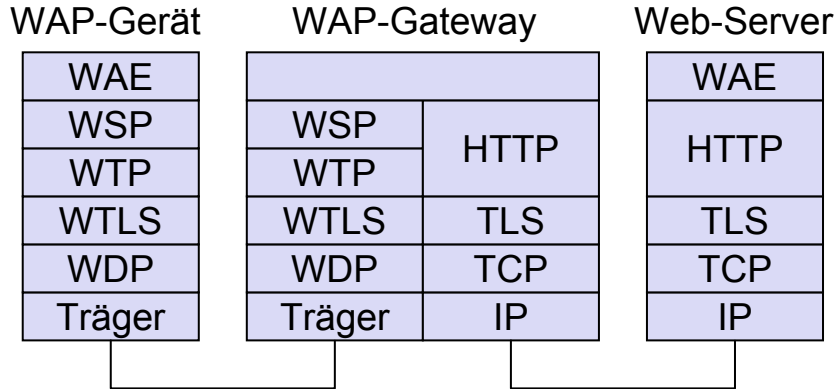
WAP 2.0 (Juli 2001)

- Neu für Entwickler
 - XHTML
 - TCP
 - HTTP
- Neue Anwendungen
 - Farbgrafik
 - Animation
 - Laden großer Dateien
 - Ortsabhängige Dienste
 - Synchronisation mit PIMs
 - Pop-up/kontextsensitive Menüs
- Ziel: Integration von WWW, Internet, WAP, i-mode

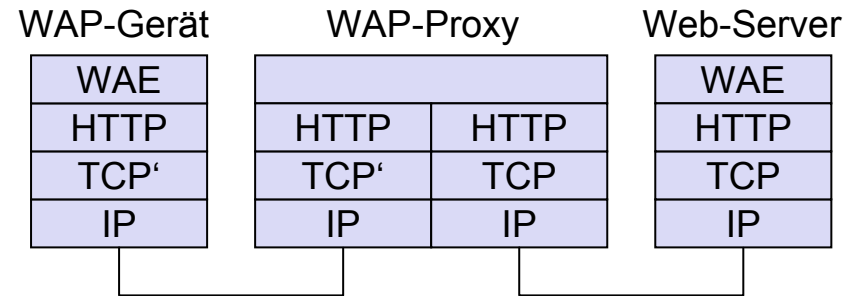
WAP 2.0 Architektur



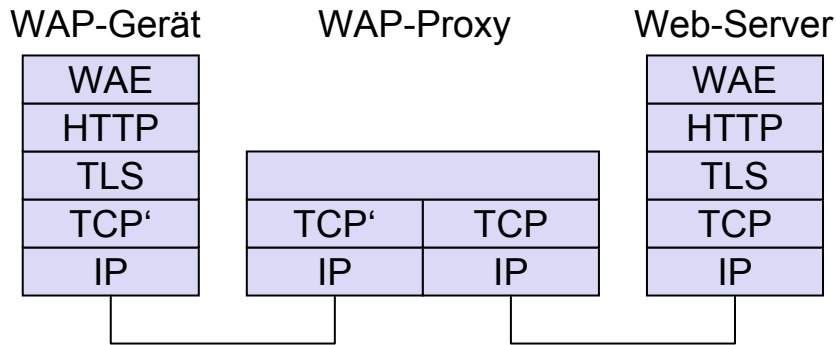
WAP 2.0 Protokollstapel



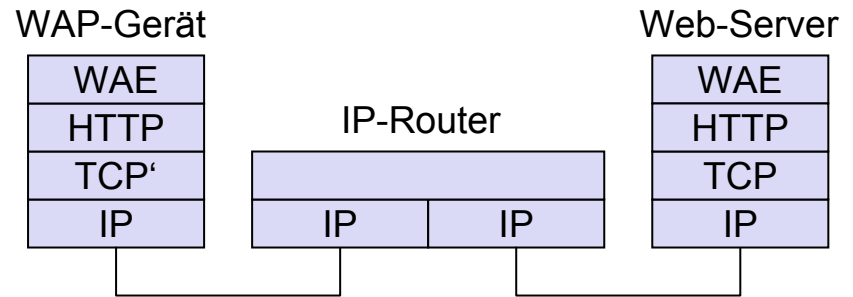
WAP 1.x Server/Gateway/Client



WAP HTTP Proxy mit angepasstem TCP



WAP Proxy mit TLS-Tunneling



WAP Direkter Zugriff

Java 2 Platform Micro Edition

- „Java-Boom erwartet“ (?)
 - Desktop-Bereich: über 90% Standard-PC-Architektur, Intel-x86-kompatibel, meist MS-Windows-Systeme
 - Was soll hier Plattformunabhängigkeit helfen?
- ABER: Heterogene, „kleine“ Geräte
 - Internet Appliances, Mobilfunktelefone, eingebettete Steuerungen, Autoradios, ...
 - Aus technischer Notwendigkeit (Temperatur, Platz, Leistungsaufnahme, ...) und Kostengründen unterschiedliche Hardware benötigt
- J2ME
 - Bereitstellung einer einheitlichen Plattform
 - Teilweise eingeschränkter Funktionsumfang im Vergleich zu Standard-Java-Plattform

Anwendungen von J2ME

- Beispiel Mobiltelefone
 - NTT DoCoMo führt iappli ein
 - Anwendungen auf PDA, Handy, ...
 - Laden von Spielen, Multimedia-Anwendungen, Verschlüsselung, Systemaktualisierungen
 - Zusatzfunktionen gegen Geld auf Knopfdruck laden
- Eingebettete Steuerungen
 - Hausgeräte, Fahrzeuge, Überwachungssysteme, Gerätesteuerungen
 - Vorrangig Systemaktualisierung



Kenndaten und Architektur

- Java Virtual Machine
 - Virtuelle Hardware (Prozessor)
 - KVM (K Virtual Machine)
 - Min. 128 kByte, typ. 256 kByte
 - Optimiert für leistungsschwache Geräte
 - Kann durch Coprozessor realisiert sein
- Konfigurationen
 - Untermenge der Bibliotheken bzgl. Standard-Java in Abhängigkeit der techn. Parameter (Speicher, CPU)
 - CLDC (Connected Limited Device Configuration, Basis-Konfiguration)
- Profile
 - Interoperabilität auch über heterogene Geräte hinweg, welche jedoch einer Kategorie angehören
 - MIDP (Mobile Information Device Profile)

Anwendungen

Profile
(**MIDP**)

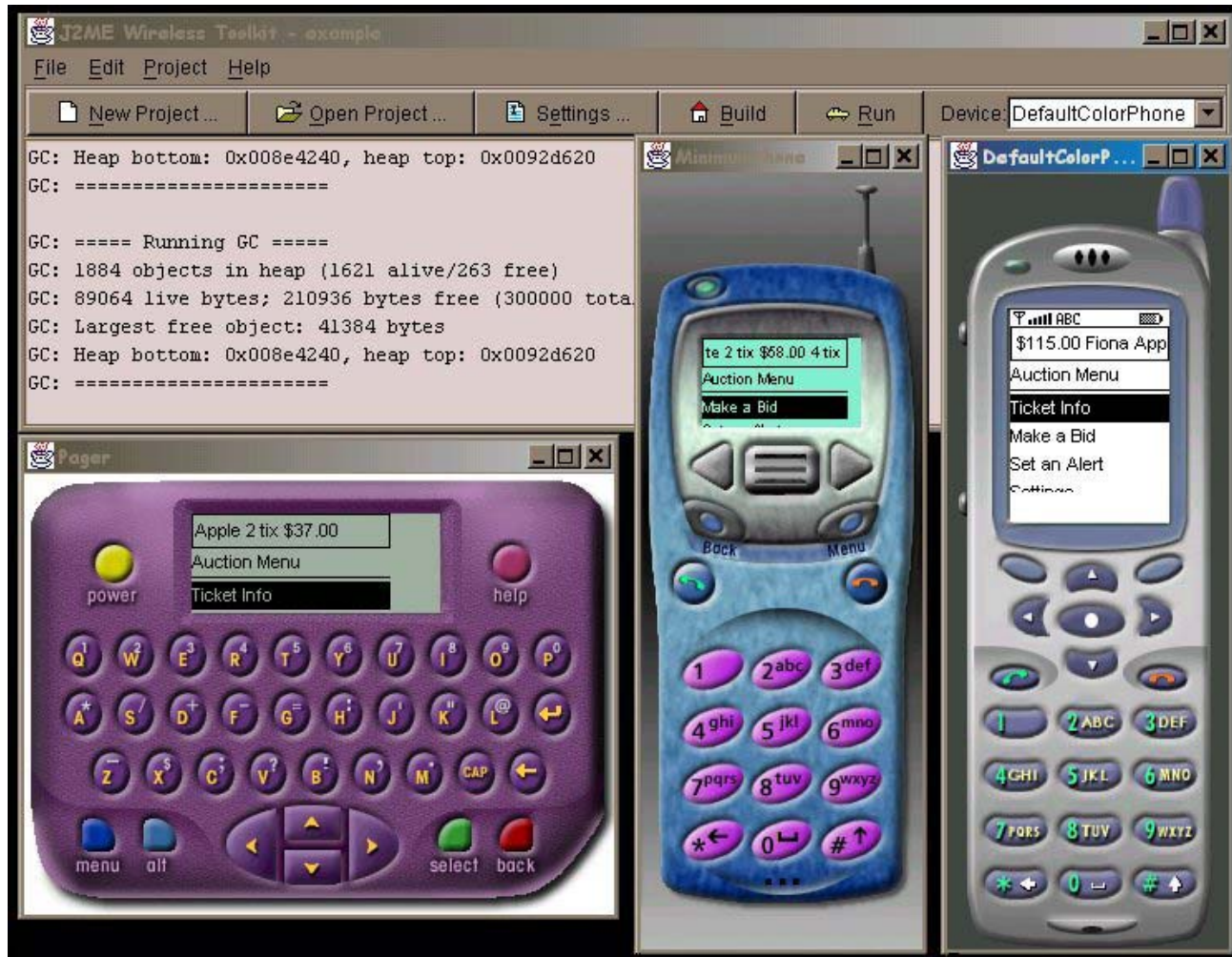
Konfigurationen
(CDC, **CLDC**)

Java Virtual Machine
(JVM, **KVM**)

Betriebssystem
(EPOC, Palm, WinCE)

Hardware
(SH4, ARM, 68k, ...)

Hardware-unabhängige Entwicklung



Zusammenfassung J2ME

- Idee geht über WAP 1.x oder i-mode hinaus
 - Vollständige Anwendungen auf Mobiltelefonen, nicht nur Browser
 - Auch Systemaktualisierungen, Ende-zu-Ende-Verschlüsselung
- Plattformunabhängigkeit durch Virtualisierung
 - Solange bestimmte Schnittstellen eingehalten werden
 - Nicht bei Hardware-spezifischen Funktionen
- Eingeschränkte Funktionalität im Vergleich zu JVM
 - Übergangslösung, bis auch eingebettete Systeme/Mobiltelefone etc. gleiche Leistungsfähigkeit wie heutige Desktopsysteme besitzen