Web Services and Data Caching for Java Mobile Clients

Alexandru Caracas, Iulia Ion, Mihaela Ion
International University in Germany, Bruchsal

BTW 2007, Aachen
6th March 2007

Outline

- Introduction
- Motivation and Challenges
- Related Work
- Data Caching
- Web Services
- Architecture
- Data Storing Implementation
- Demo
- Evaluation
- Conclusions and Future Work

Introduction and Motivation

- Mobile telephony market:
  - Tremendous increase in the usage of mobile phones
- Mobile services and applications:
  - Mobile devices become excellent platforms for many types of applications and services
- Target application:
  - We want to consult train schedule from the device
  - Karsruher Verkehrsverbund: http://www.kvv.de/kvv/

Challenges of Mobility

- Wireless connection:
  - Air data transmissions are energy-intensive and reduce the operating time of the device.
  - Physical effects like shielding, reflection, refraction etc.
  - Impossible to guarantee a permanent availability
  - Solution: Data caching
- Limited computational capabilities and battery lifetime:
  - Solution: move computationally intensive operations to a server, for example, use Web Services

Related Work

- *Deutsche Bahn (DB)* personal scheduler http://persoenlicherfahrplan.bahn.de
  - Data is static, downloaded once and not up-to-date
- *Actuan Mobile*: complete dynamic schedule for the New Jersey Transit rail system http://www.actuanmobile.com/documentation/train_schedule_manual-online-version-v0_1.pdf
  - No cashing is used
  - The schedule information queried from online servers for each request

Data Caching

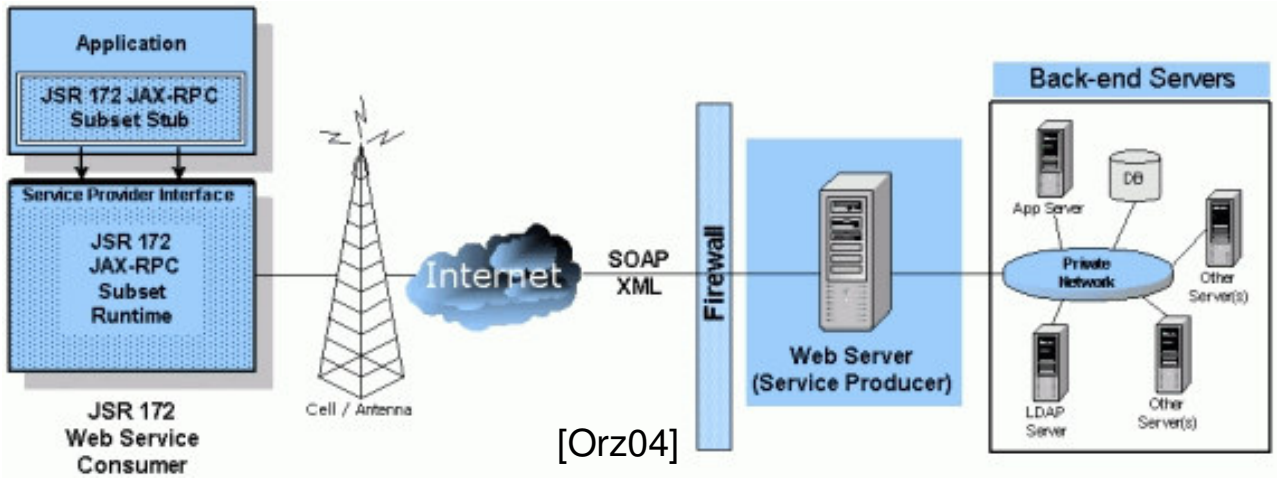
- Storing and reusing received data in order to reduce volume of data transferred
- Creates redundant data that needs to be maintained by the system
- Publications on caching, hoarding or replicating data in MIS lack an usable implementation and evaluations are based on simulations.

Data Caching

- Data synchronization:
  - Clients do not allow modifications of cached data
  - → synchronization techniques are not required
- Coherency:
  - Must guarantee that stored data is up to date
  - → use an attribute that specifies how long a data item is valid
- Replacement:
  - Devices have limited amount of memory
  - → FIFO replacement strategy
- Efficient lookup:
  - → we use unique identifiers instead of query indexes and request complete data items

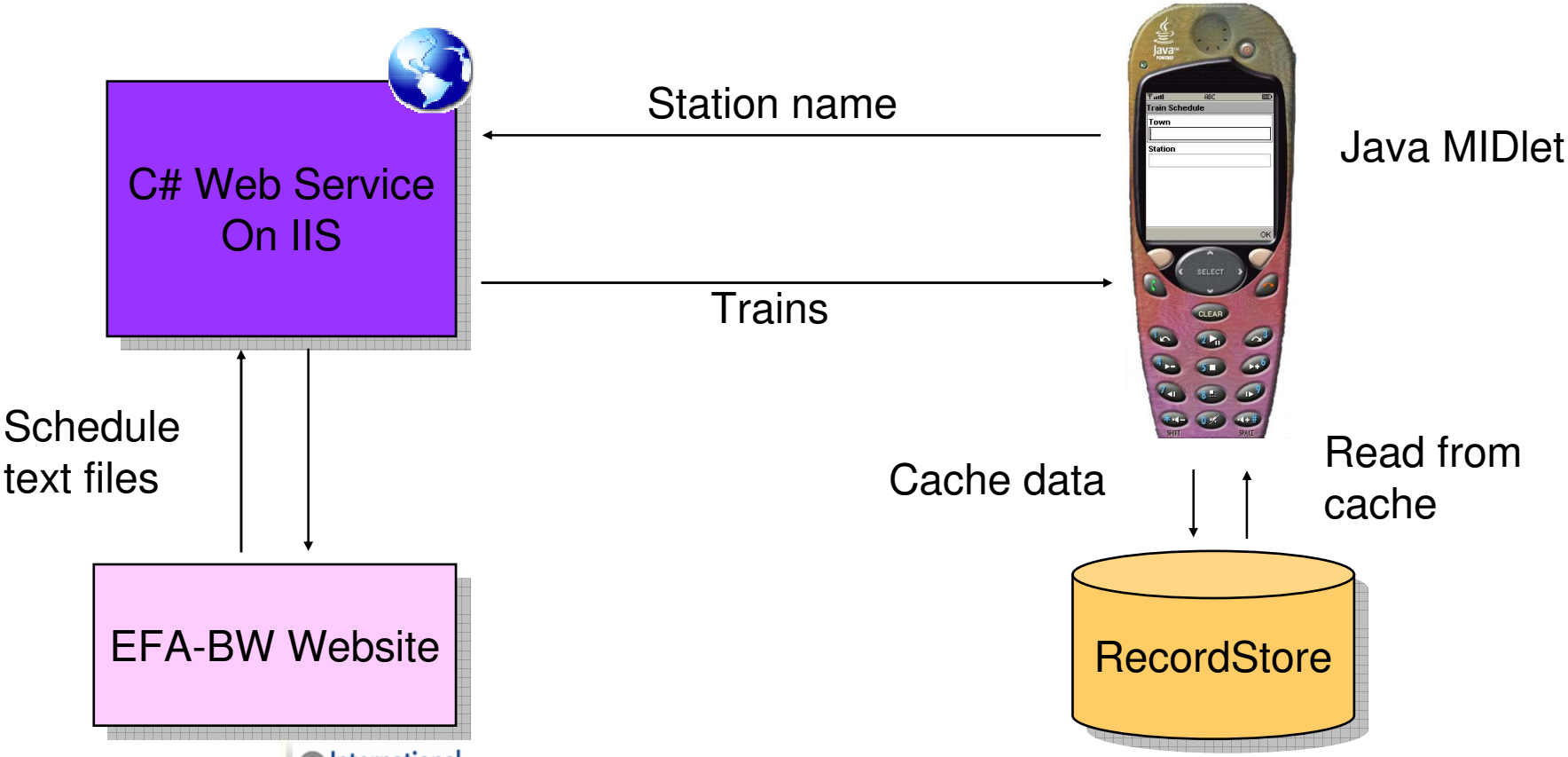
Web Services

- Optional package **JSR 172** for MIDP 2.0: J2ME Web Services Specification
  - Defines API for Web Service communication
  - Enables mobile clients devices to access web services
- The application on a WSA-enabled wireless device includes:
  - The JSR 172 runtime
  - A JSR 172 stub to communicate with the network.



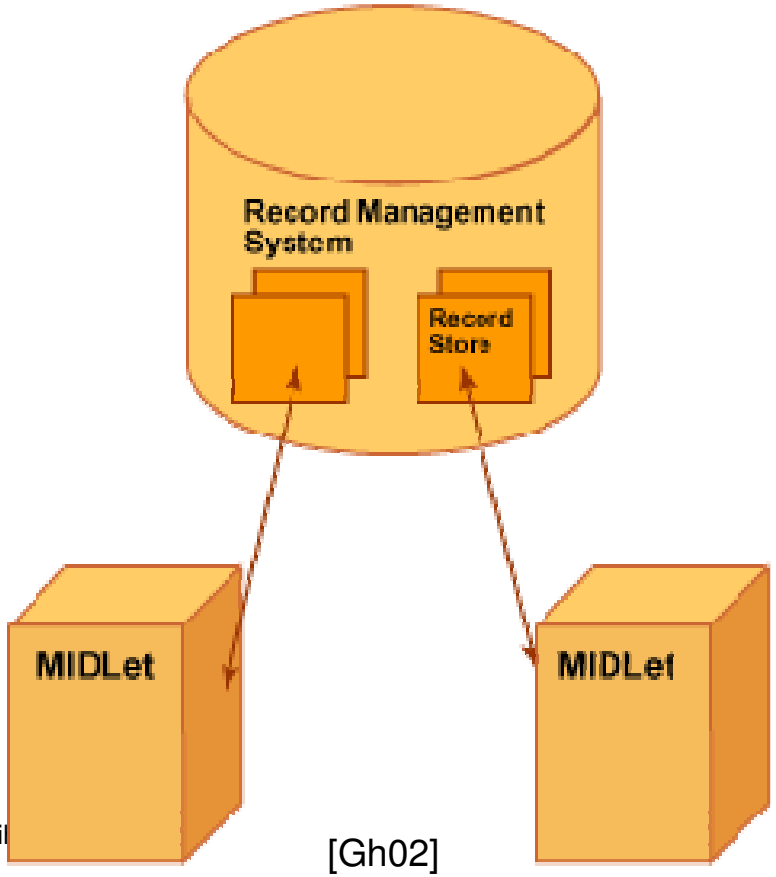
[Orz04]

Architecture



MDIP RecordStore

- Offers *persistent* storage in the form of a collection of records
- Is created in platform-dependent locations, (e.g. non-volatile device memory), which are not directly exposed to the MIDlets
- Ensures atomic, synchronous, versions and timestamps and serialized operations
- Application has to ensure thread safe behaviour



Storing Data

…In a structured manner

BRUCHSAL-BAHNHOF

	Valid until
00h – 01h	Train Train ....
01h – 02h	Train Train ....
	.....
23h – 24h	.....

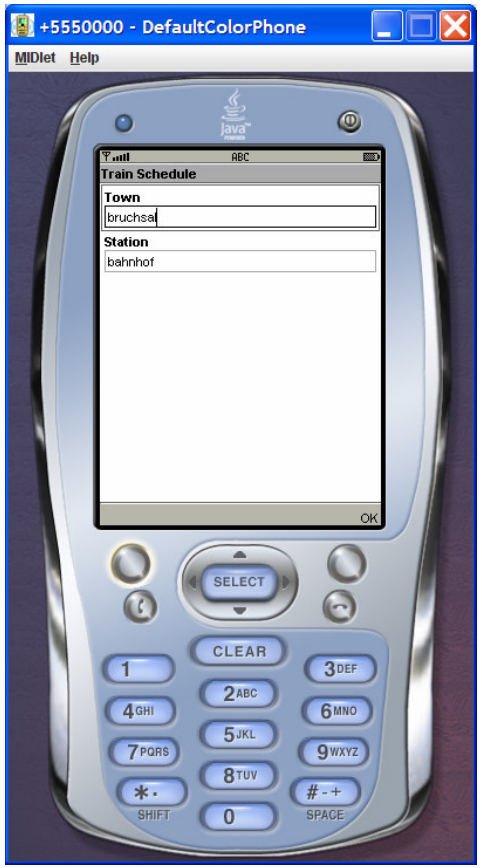
KARLSRUHE-BAHNHOF

	Valid until
00h – 01h	Train Train ....
01h – 02h	Train Train ....
	.....
23h – 24h	.....

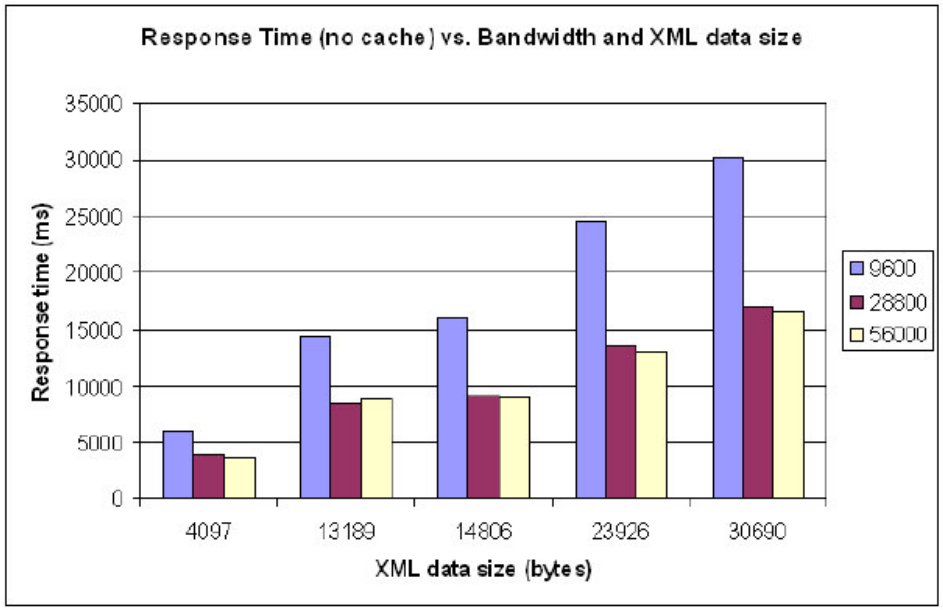
Train

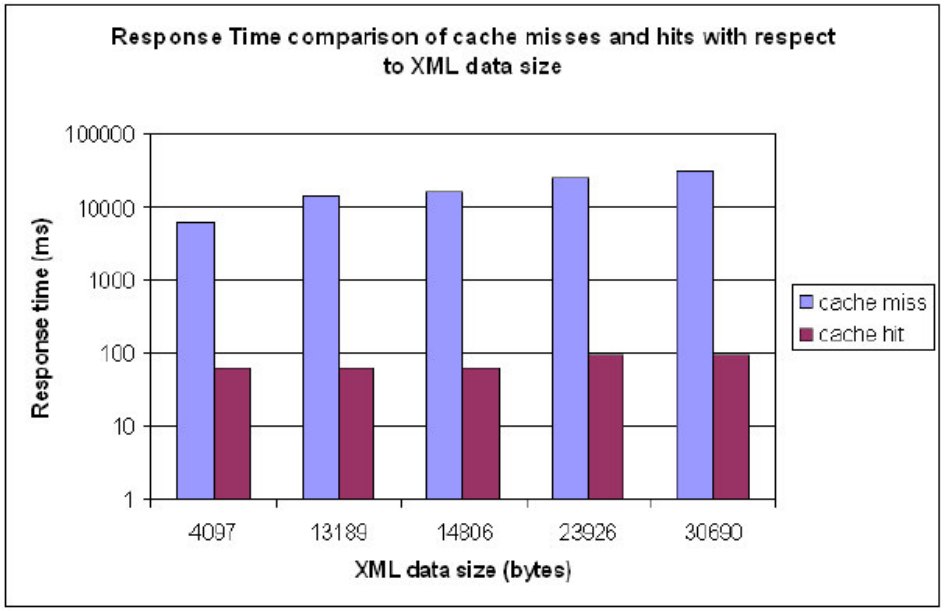
len	**type**	len	**time**	len	**destination**

Demo


+5550000 - DefaultColorPhone
MIDlet Help
Train Schedule
Town
bruchsal
Station
bahnhof
OK
SELECT
CLEAR
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
*
SHIFT
0
#
SPACE

Cache Performance





- Answering a query locally is much faster than receiving the data over network
- The influence of the size of nearly requested data disappears if data is provided locally

Conclusions

- Implementation substitutes caching issues discussed in journals and conferences by pragmatic approaches
  - Wireless connection challenges → data caching
  - Limited device capabilities → Web Services
- Results show:
  - Caching on mobile phones is really possible
  - There is a huge potential to support the integration on mobile devices into information systems
- Future work:
  - Implement more efficient caching strategies
  - Evaluate and compare in detail the implementations

Thank you for your attention!

Questions?

References

- [EN04] Elmasri, R. and Navathe, S. B.: Fundamentals of Database Systems. Addison Wesley. 4th. 2004.
- [EY03] Ellis, J. and Young, M.: J2ME Web Services 1.0. Sun Microsystem, Inc. Santa Clara, CA, USA. Final draft. October 2003. Available at http://jcp.org/aboutJava/communityprocess/final/jsr172/index.html.
- [Gh02] Ghosh, S. J2ME record management store – Add data storage capacities to your MIDlet apps. Online article at IBM developerWorks. May 2002. Available at http://www-128.ibm.com/developerworks/library/wi-rms/.
- [GHOS96] Gray, J., Helland, P., O'Neil, P. E., and Shasha, D.: The Dangers of Replication and a Solution. In: Jagadish, H. V. and Mumick, I. S. (Eds.), Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996. volume 25 of SIGMOD Record. pp. 173–182. New York, NY, USA. June 1996. ACM Press.
- [Gi04] Giguere, E. Databases and MIDP, Part 1: Understanding the Record Management System. Online article at Sun Developer Network. February 2004. Available at http://developers.sun.com/techtopics/mobility/midp/articles/databaserms/.
- [HTKR05] Hopfner, H., Turker, C., and K¨onig-Ries, B.: Mobile Datenbanken und Informationssysteme —Konzepte und Techniken. dpunkt.verlag. Heidelberg, Germany. July 2005. in German.
- [LLS99] Lee, K. C. K., Leong, H. V., and Si, A.: Semantic query caching in a mobile environment. ACMSIGMOBILE Mobile Computing and Communications Review. 3(2):28–36. April 1999.