# Mobile Transactions and Synchronization

Can Türker

ETH Zurich

Seminar "Ubiquitous Information", 28.11.2001

# ... towards Pervasive Computing

- Gartner Group: *"by 2003 more than 137 million business users will be involved in some form of remote work"*
- Accenture: *"by 2005 over 500 million mobile devices will offer Internet access"*

- *Conclusions*
  - mobile hosts (MH; laptops, palmtops, smart phones, etc.) outstrips fixed hosts (FH; personal computers, desktops, etc.)
  - the way information is created and processed will change within this increasingly ubiquitous network

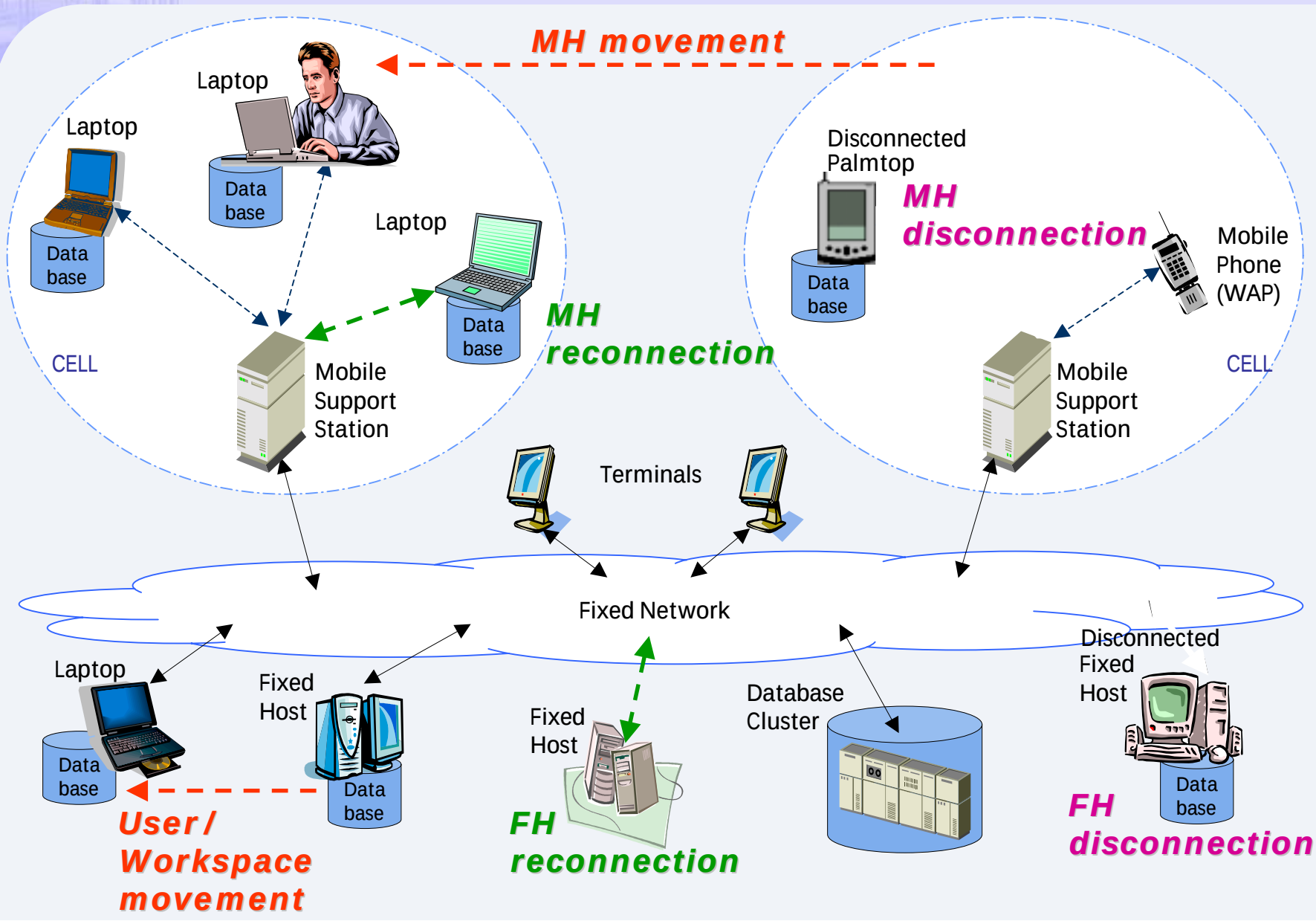
- *Need*
  - infrastructure to coordinate concurrent information access and processing in the presence of mobile hosts and users

# Characteristics of Mobility

- Mobile information sources and consumers
  - physical access point to the network may change: sources as well as consumers may move
  - sources as well as consumers may be disconnected
- User and Context Awareness
  - tracking/monitoring information sources and consumers
  - consumer's information needs may shift with location change
- Data management techniques have to be revisited

| | |
|---|---|
| Resource Limitations (Bandwidth, Memory, Computing Power, ... ) | Optimization + Careful Resource Sharing |
| Scalability | |
| Correctness Concerns | Transactional Guarantees |
| Combining Many Sources | Data Integration |

# Movements and Disconnections



# Abstractions

| | |
|---|---|
| **Abstraction of Data Storage** | (Relational) DBMS |
| **Abstraction of Concurrency & System Failures** | DBMS & TP-Monitors with Concurrency Control & Recovery |
| **Abstraction of Method Implementations** | Object-relational DBMS with Object Methods, Triggers & Stored Procedures |
| **Abstraction of Distribution, Heterogeneity & Autonomy** | Distributed & Federated DBMS, Data Integration, Conflict Resolution |
| **Abstraction of Movements & Disconnections** | Mobile DBMS, Context Maintenance, Replication & Synchronization, Profiling |

# Transparency of Mobility

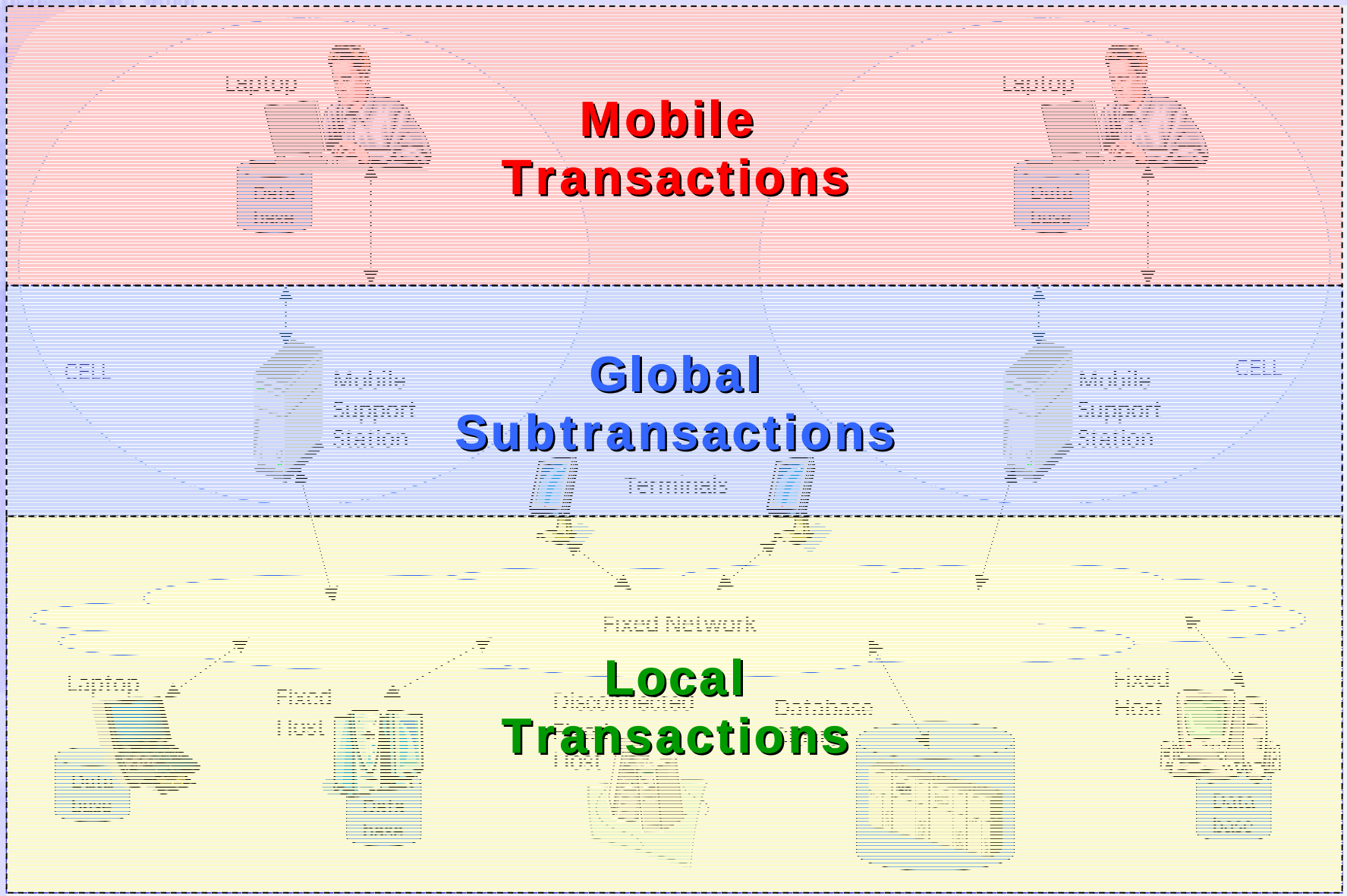
- Challenge: information access and processing everywhere and at anytime while supporting
  - *transparent disconnections* and
  - *transparent movements of users and information components*

- How much transparency is indeed needed and reasonable?

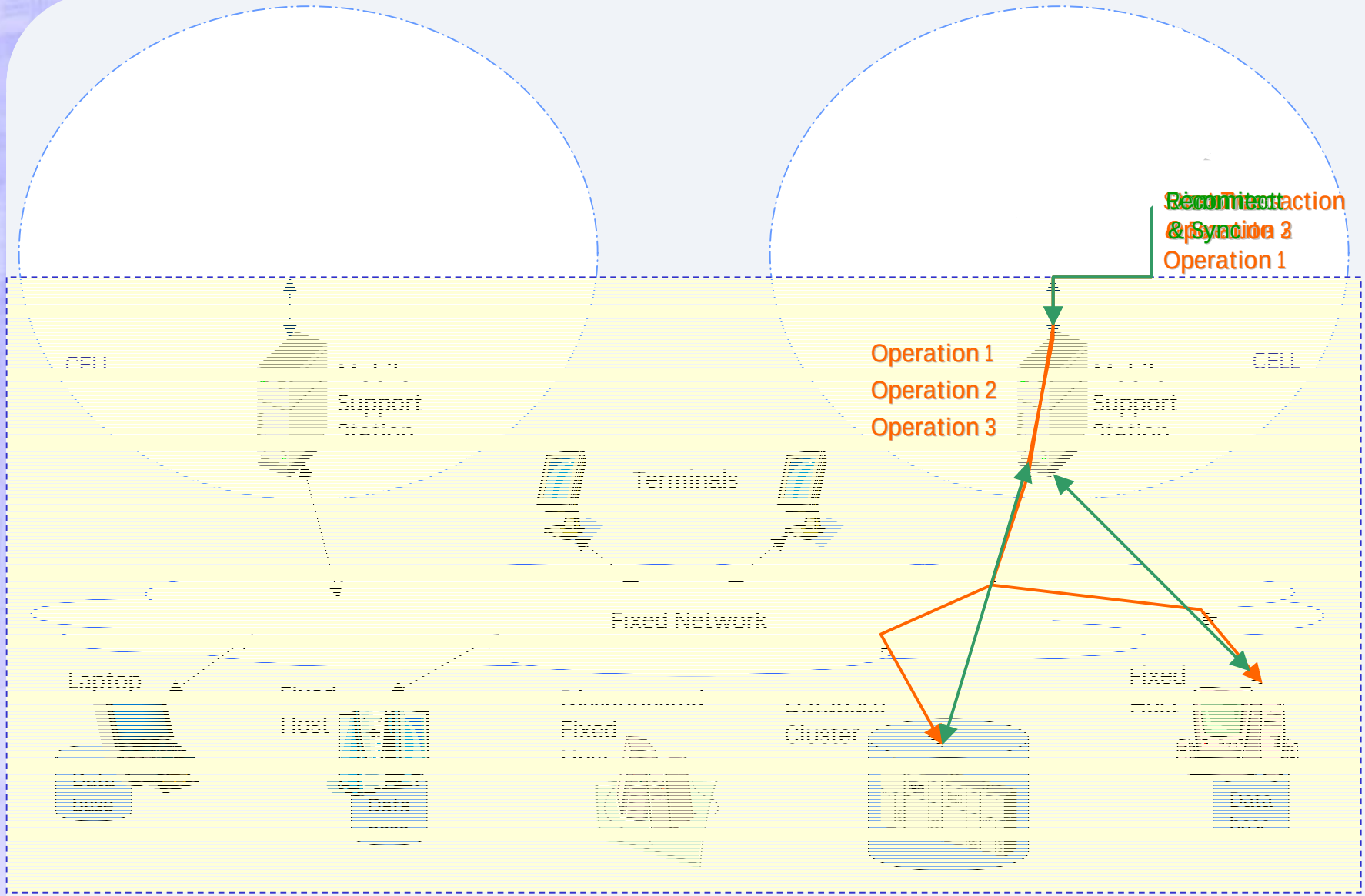
# Characteristics of Transactions

- Logically related set of operations executed under certain guarantees, e.g.,
  - atomicity
  - consistency
  - isolation
  - durability
- Advanced transaction models rely on multi-tier transactions
  - increase parallelism by exploiting application semantics
  - parent-child relationships
- Mobile Transactions involve execution/initiation on MH
  - ACID cannot be supported generally
  - nevertheless: certain transactional guarantees shall be ensured always and everywhere

# Transaction Tiers



# Disconnection-resistant Transactions



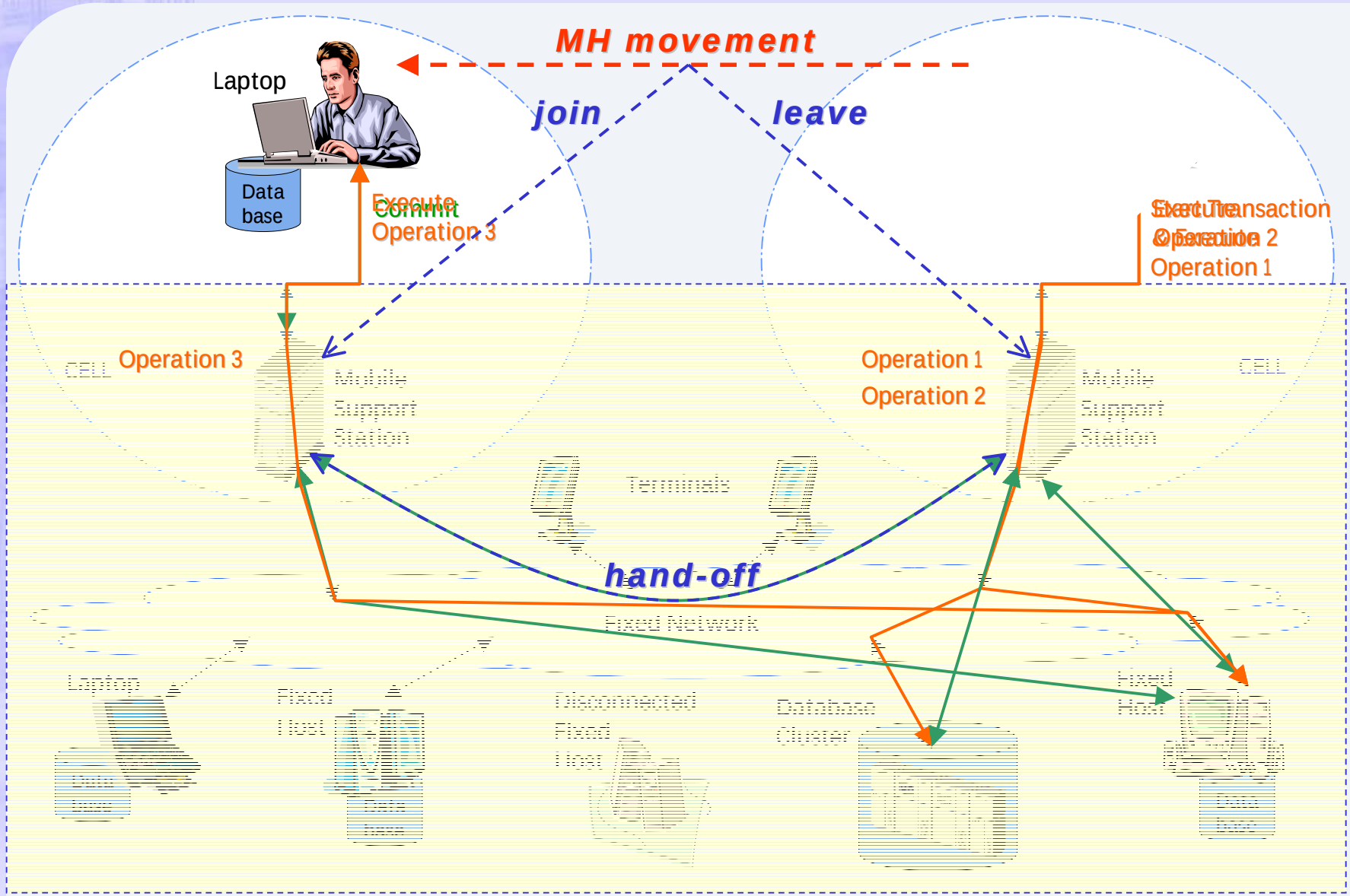
# Disconnections

- "The flight worker": Working in the "intended" disconnected mode requires some lazy replication techniques
  - updates are precommitted locally
  - precommitted updates are propagated asynchronously when reconnected to the network
    - conflicts may occur
    - conflict resolution when a conflict arises

transparent to the user?

- Conflict detection via timestamps, version vectors, etc.
- Conflict handling
  - optimistic (resolution): function-based, manual
  - pessimistic (avoidance): primary copy, ROWA, quorum

# Movement-resistant Transactions



# Movements

- "The train/tram/bus worker": Working while physically moving requires transparent support of cell migration and "unintended" disconnections
  - create subtransactions on several mobile support stations
  - coordinate these substranactions correctly
  - wireless communication and cost issues

transparent to the user?

- "The home worker": Resume and/or continue work at another host (mobile or fixed)
  - continue transactions
  - create new subtransactions (within the existing workspace / transaction sphere)

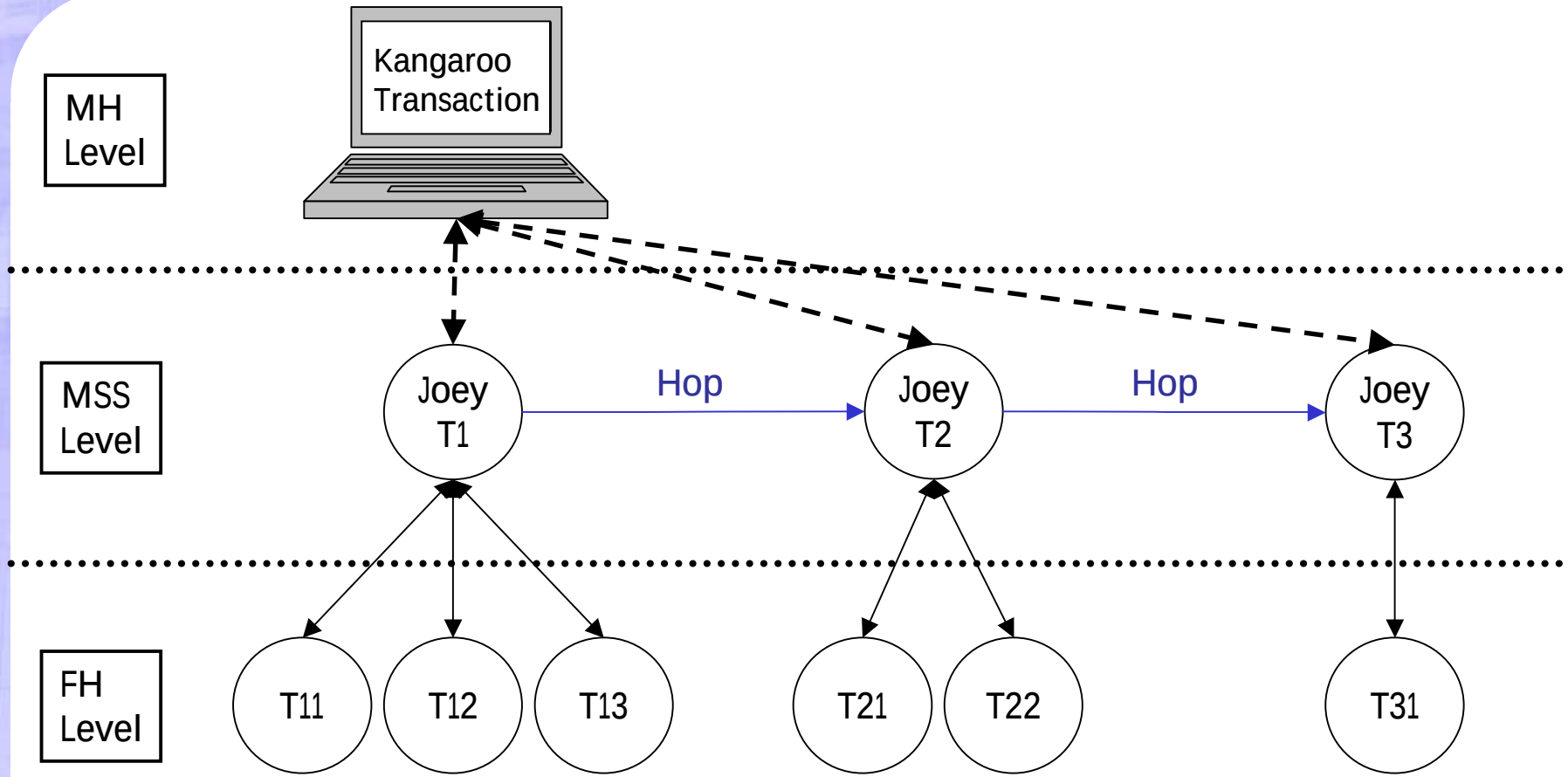
# Issues of Mobile Transaction Models

- MH may have transaction processing capabilities
  - MH may run transactions locally
  - MH may only initiate transactions on FH
- MH may change location and network connection while transactions are being executed
  - split computations
- MH may disconnect while transactions are being executed
  - long-lived processes
  - data replication

# Characteristics of Advanced Transaction Models

- closed vs. open
  - child's results are visible to the parent only or to all?
- vital vs. non-vital
  - parent's commitment depends on the child's commitment?
- dependent vs. independent
  - child's commitment depends on the parent's commitment?
- substitutable vs. non-substitutable
  - does there exists an alternative transaction?
- compensatable vs. non-compensatable
  - are the results semantically undoable?

# An Example of a Mobile Transaction Model: Kangaroo Transactions




- On disconnect: JTs and Ts can complete but no more Ts are created
- On reconnect: the T resumes
- Hand-offs: new JT runs on the new MSS (transaction split)

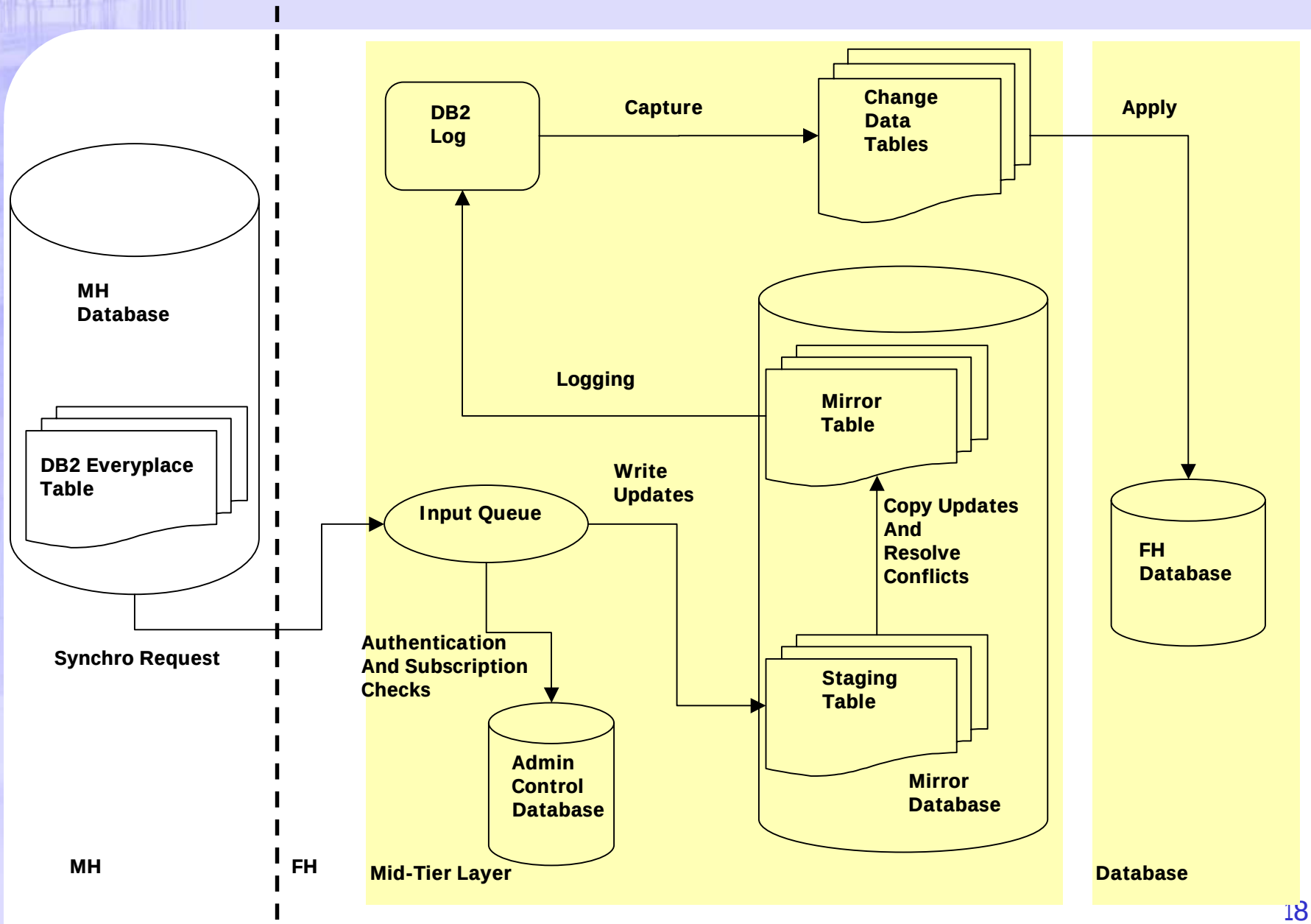
# An Example of a Mobile Transaction Model: Kangaroo Transactions

- Builds on open nested and split transaction models
- Supports mobility and disconnections
- MH starts a *Kangaroo* Transaction (KT)
  - A *Joey* Transaction (JT) is started at the connected MSS
  - T run on the FH (as open nested transaction)
  - If the MH changes location, the previous JT is split and a new JT runs on the new location MSS
  - JT1 can commit independently from JT2
- If JT1 fails:
  - Compensating mode: undoes the entire KT
  - Split mode:
    - Previously committed JTs are not compensated
    - No new JTs are initiated
    - Current child transactions are committed or aborted upon decision of the local DBMS

# Overview of Mobile Transaction Models

| Subtransaction Types & Mobility Support | Open | Closed | Non-vital | Independent | Substitutable | Compensatable | Temporal | MH Disconnection | MH Movement | MH Usage | User Profiling |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reporting-/Co-Txs<br>Chrysanthis 93 | ✓ | ✓ | | | ✓ | ✓ | | | ✓ | H | |
| Isolation-Only Txs<br>Lu & Satyanarayanan 94 | | ✓ | | | | | | ✓ | | H | |
| MDSTPM Txs<br>Yeo & Zaslavsky 94 | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | | L | |
| Weak/Strict Txs<br>Pitoura & Bhargava 94 | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | H | |
| Kangaroo Txs<br>Dunham et al. 97 | ✓ | | | | | ✓ | | ✓ | ✓ | L | |
| Pro-Motion<br>Walborn & Chrysanthis 97 | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | H | ✓ |
| Toggle Txs<br>Dirckze & Gruenwald 98 | ✓ | | ✓ | | ✓ | ✓ | | ✓ | ✓ | L | |
| Moflex Txs<br>Ku & Kim 00 | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | L | |

# IBM DB2 Everyplace



# An Example of a Commercial Mobile Database Approach: IBM DB2 Everyplace

- *DB2 Everyplace*, relational database residing on the MH
- *DB2 Everyplace Sync Server*
- Master database on the FH
- Replication is asynchronous and on demand
  - entire tables, subsets of columns/rows, views, joins, unions
- Synchronization by publish and subscribe
  - *Refresh propagation*: all data are sent to subscribers
  - *Incremental propagation:* only changes are propagated to the subscribing MHs
- Conflicts are handled by checking the version of each record in each table in the replication subscription
- No savepoints, No transaction nesting

# Overview of Commercial Mobile DB Approaches

| Subtransaction Types & Mobility Support | Savepoints | Tx Nesting |
|---|---|---|
| IBM DB2 Everyplace | | |
| Informix Cloudspace | | |
| Microsoft SQL Server CE | ✓ | ✓ |
| Oracle Lite | ✓ | |
| Sybase Anywhere | ✓ | ✓ |

- only a few supports nested Txs (closed, vital, dependent)
- no compensating / alternative Tx
- "basic" data replication and synchronization techniques
- no transaction mobility (hand-offs)

# HotSync

- MH stores data as records in PalmDBs
  - each PalmDB is associated with an application
  - each record has a set of status bits indicating whether the record has been created, modified, or deleted since last sync
- FH maintains it's own copies of the PalmDBs, including it's own versions of the status bits
  - also maintains a snapshot of each PalmDB taken immediately after most recent sync
  - runs HotSync Manager
- MH initiates synchronization (fast vs. slow sync)
- Problem: No support for synchronizing with multi-user concurrent data sources
  - no notion of "*interest*" in a *subset* of the records in a database
  - no notion of *transaction* at all

# SyncML Standard

- Industry Consortium with most major players: Ericsson, Nokia, Motorola, Palm, Psion, IBM, ...
- Goal: enabling cross-format, cross-system synchronization
- Simple architecture
  - MH is intermittently connected und FH is continuously available
- Consists of a standard set of message types, each represented as an XML document
- Conflict resolution is dealt with abstractly
  - only standard status codes to implement typical policies are given

| | |
|---|---|
| Two-way | MH and FH exchange only modified records, MH sends first |
| Slow-sync | MH sends all records |
| One-way, MH only | MH sends only modified records to FH |
| Refresh, MH only | MH sends entire DB to FH |
| One-way, FH only | FH sends only modified records to MH |
| Refresh, FH only | FH sends entire DB to MH |
| FH Alerted | FH initiates sync |

# Summary

- Mobile transactional coordination has to deal with
  - weak connectivity and frequent disconnections
    - asynchronous, dynamic replication with profiling
    - publish & subscribe for data recharging & propagation
  - large-scale replication
  - user interaction / feedback
  - long-running tasks and decentralized commitments
  - real-time constraints
- Commercial mobile database approaches mostly neglect the latter issues

# Open Questions

- Where to implement the abstraction of disconnections and movements?
  - Do we really need extensions to transaction models?
  - Can we model mobility issues as additional steps of an overall transactional process?
  - Which part is only "network staff"?
  - ...
- Design of mobile transactional applications
  - How to appropriately express mobility issues?
    - connection duration and costs
    - disconnection times
    - data and function interests
    - consistency requirements
  - ...