Over The Air Sync

SyncML Server
Google, ScheduleWorld, Funambol, Memotoo, Mobical, Synthesis, ZYB ...

SyncML (OMA DS): protocol vCard/vCalendar/iCalendar: payload

SyncEvolution - Sync PIM
(Contacts, Events, Tasks, Notes)
History  “The Missing Link”

• SyncEvolution: spare time project for >3 years, full-time since 2009
• So far: SyncML client for
  • Evolution/Moblin
  • iPhone+Mac OS X (need maintainer)
  • Maemo (Ove Kaaven: Maemo 5)
  • KDE (under development)
• More to come in 1.0:  
direct synchronization
Synthesis

- Swiss company with >10y experience with SyncML, owned by the two main developers
- Synthesis SyncML Engine:
  - C++ implementation of client and server, data merging and conversion
  - Stable C API+ABI
  - LGPL 2.1+3.0
- Funambol: three different code bases, no stable ABI, AGPL v3
- libsyncml: only lower protocol layers, lacks synchronization engine
- OpenSync/Conduit: no/limited (?) data conversion
Components

- **SyncEvolution:**
  - User interface (command line, GTK GUI)
  - Configuration handling
  - Local data backends (Evolution Data Server, Mac OS X Address Book, files, XMLRPC, Akonadi)
  - Message transport (HTTP via libsoup/libcurl)
  - Test suite

- **libsynthesis:**
  - SyncML protocol (up to 1.2.1 very complete)
  - Data conversion, merging, database storage (ODBC)

- **libsmsltk (SyncML Toolkit):**
  - Message encoding/decoding (XML + WBXML)
SyncML Server

SyncML Clients

PIM Sync Challenges:
It can eat your data?!
1. No globally unique ID.
2. Different representation of data inside clients.
3. Offline updates allowed: what is the most recent data?
4. Must support standard formats, including legacy ones (vCalendar).
5. Must work with limited knowledge about peers (capabilities, behavior).
SyncML as Protocol

- “slow sync” - client sends all items, server compares, sends updates back
- “normal sync” - exchange of updates
- Resilient – suspend&resume (voluntarily or due to loss of connection), resend messages
- Avoid overloading peer: maximum message size, flow control
- Device Capability description
SyncML Client

• Easy to implement, less complex side
• List items and changes since last sync session, using locally unique ID
• Import and export items, either in standard format or Synthesis representation
• Data merging optional (supported by Synthesis engine)
SyncML Server

- Convert data between clients
- Map between local IDs
- Handle conflicts:
  - Fixed policy in HTTP servers
  - More flexible in desktop apps?! 
- Merge incomplete updates received from dumb clients with complete item on server
- Synthesis: same datastore API as on client side
Synthesis Data Handling

• Specified via XML config: new data formats can reuse existing mechanisms.
• Field list declares internal format and key properties for matching items.
• Conversion to and from external formats via one or more profiles; can be parameterized.
• Can use modification time stamps and different conflict resolution policies (duplicate, merge).
• Uses Device Capabilities: solves the “property removed or not supported?” problem.
Next Steps  + Getting Involved
Next Steps

- 1.0 release: beta available, release end of March
  - Sync in background + recovery
  - Direct device-to-device sync
- Beyond 1.0:
  - Device database ("config templates")
  - Support more local backends, including adaptors for other protocols (need contributors!)
  - Local sync between backends
  - Push sync
Getting Involved

- http://syncevolution.org
- Mailing list, also on Gmane
- bugzilla.moblin.org with enhancement ideas
- “SyncML Client Do-it-yourself Style”
- LWN.net article, long version on moblin.org