

PIM in IVI

Contacts, your phone, your car

Patrick Ohly
Intel GmbH

TIZEN[™]
**DEVELOPER
CONFERENCE**
2014
SAN FRANCISCO

Content

- Use cases
- Tizen IVI:
 - 1 Evolution Data Server
 - 2 SyncEvolution
 - 3 libphonenumber
 - 4 folks
- First steps and links



CC BY 2.0, Jim D. Woodward

Head Unit as a Better UI for Multiple Phones

- Fast caller ID lookup for incoming call (**not the same as text phone number search!**), local photo data
- Look up a contact via searching or browsing and initiate a phone call
- Find address and start navigation
- Manage data from more than one phone: driver and passenger
- Unified address book: no duplicates, merge data from all available sources; may include dynamic information (online presence)

Connected car

- **Direct synchronization with cloud services:**
 - Google Contacts
 - iCloud
 - Enterprise (= Exchange)

The GNOME PIM Stack for Tizen IVI: Overview

- **Evolution Data Server:** store contacts in sqlite
- **libphonenumber:** parsing and normalization of phone numbers
- **folks:** unified address book in memory
- **SyncEvolution:**
 - Phone and cloud syncing
 - Hosts the unified address book
 - “IVI PIM Manager” D-Bus API
- **Bluez obexd: Phone Book Access Protocol**

Overview, cnt.

- **LGPL 2.1 or more liberal**
- **Minimal additional dependencies (no GTK):
ICU, glib, sqlite, vala, libgee, libsecret, libgcr, protobuf, gtest;
optionally for cloud sync also neon + gSSO**
- **Locale aware:**
 - **Phone number parsing**
 - **Sorting, special case Pinyin (transliterate, then mix with Western names)**
 - **Support systemd localed for dynamic change of locale**



IVI Features in the GNOME-based Stack



Evolution Data Server: “Per-device access”

One address book per phone

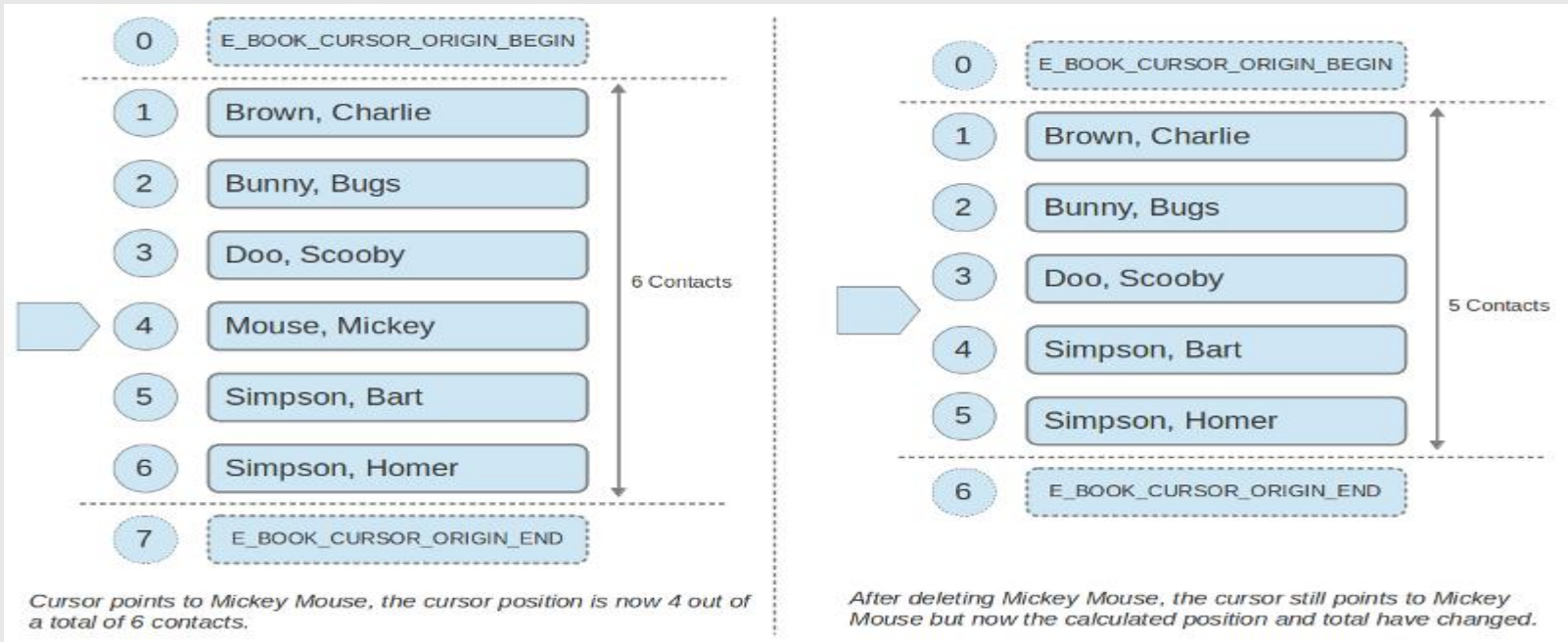
Traditional EDS:

- Abstract API, framework, storage provided by specific backends
- File backend: Berkley DB + sqlite index

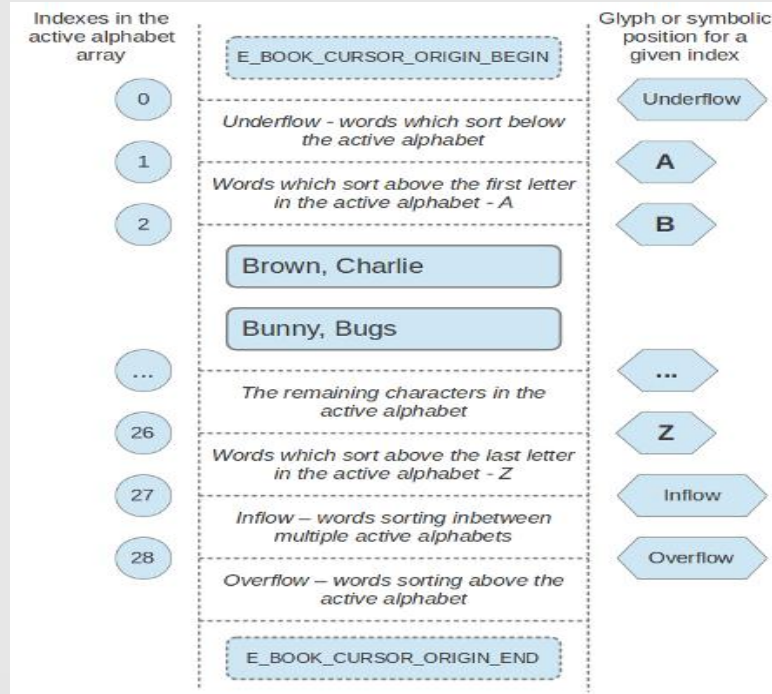
Enhanced EDS (3.6 and later plus Openismus patches, almost all included in 3.10):

- All string data in sqlite, configurable indices to reflect searches made by UI
- Photos as separate files, managed by EDS
- Normalize phone numbers with libphonenumber:
find “089-1234” when looking for “+49891234”
- Regular expression support for fuzzy phone number search and keypad search
- Writing in EDS daemon, reading in clients:
same API, just different open method
- Efficient browsing through all or some contacts:
sorted results, cursor marks current position, reading only returns next *n* contacts
- Locale-aware alphabetic index

Evolution Data Server: Cursor



Evolution Data Server: Alphabetic Index



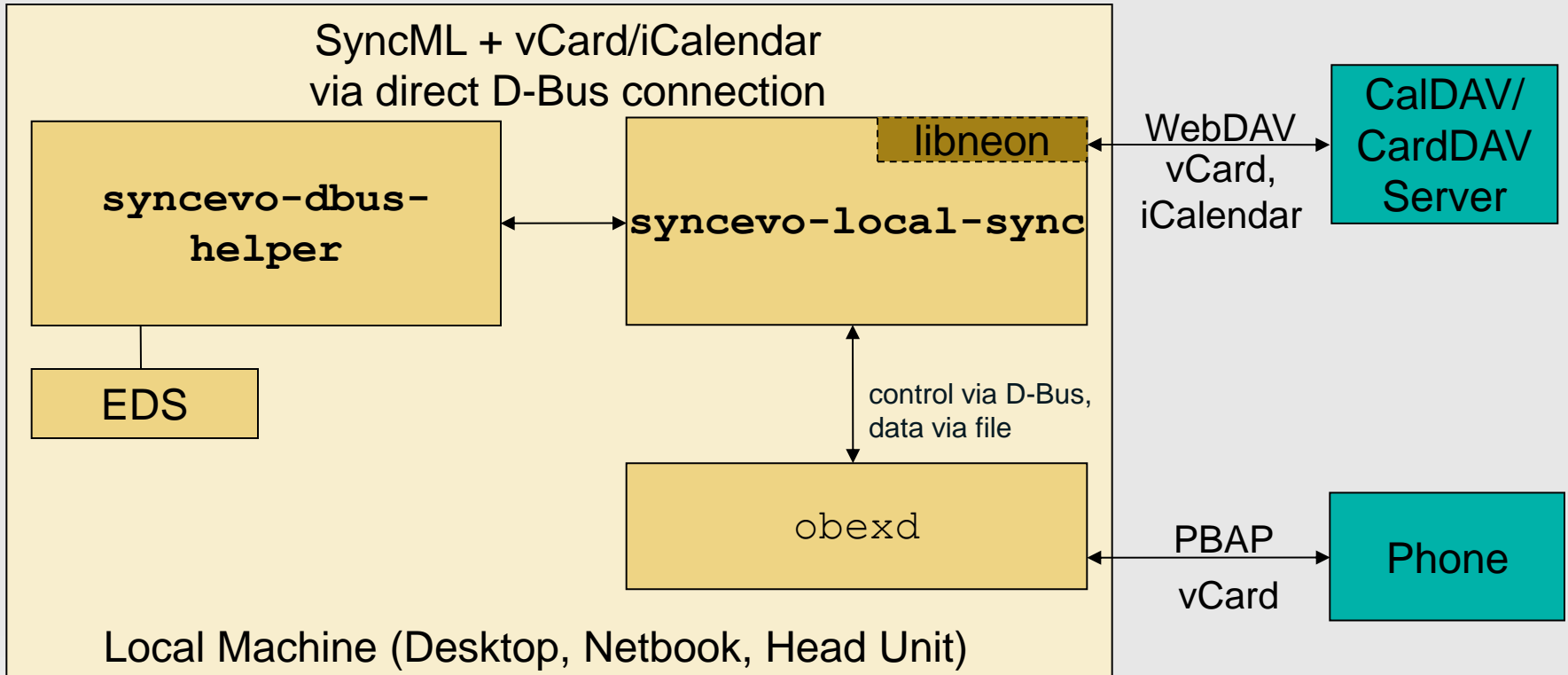
SyncEvolution + folks: “Unified address book”

- Configurable set of enabled address books
- No disk writes when reconfiguring
- Might include transient information (presence status)
- Kept in memory
- Sorting, searching, fast caller ID lookup
- Model/view/controller principle for results

SyncEvolution + obexd: PBAP syncing

- **Take full dump of phone address book, detect changes and apply them to local cache**
- **No writes (logs, DB, sync meta data) if nothing changed on phone**
- **Incremental syncing:**
 - First (or only) text: must not modify local photos
 - Then everything: must add/update/remove local photos
- **Overlap download and processing**
- **Goal is to finish sync shortly after finishing PBAP download (“processing at wire(less) speed”)**

PBAP + CalDAV/CardDAV Sync





Getting Started



Installation

- **Core PIM Stack:**

```
# zypper install syncevolution-ebook \  
evolution-data-server
```

- **PBAP:**

```
# zypper install obexd bluez-test syncevolution-pbap
```

- **CardDAV:**

```
# zypper install signond gsignond-plugin-oauth \  
signonui-efl \  
libgsignon-glib-devel \  
syncevolution-dav  
# signonui-efl might not be available yet (TINF-588)
```

- **PIM API examples:**

```
# zypper install syncevolution-test
```

PBAP Contact Caching

- **Run as normal user in a regular user session (session D-Bus is needed)**

- **Pair a phone:**

```
$ bluetoothctl
[bluetooth]# power on
[bluetooth]# scan on
[bluetooth]# pair A0:4E:04:1E:AD:30
```

- **Configure and cache address book:**

```
$ /usr/lib/syncevolution/test/sync.py \
    --bt-mac=A0:4E:04:1E:AD:30 \
    --configure \
    --progress --sync
```


Accessing Contacts

- **PBAP:**

```
$ syncevolution --export -- \
    backend=pbap database=bt-obex://A0:4E:04:1E:AD:30
```

- **EDS:**

```
$ syncevolution [--print-databases|
    --print-items|--export -|--import <file>] \
    backend=evolution-contacts [database=<DB name>]
```

- **Unified address book:**

```
$ /usr/lib/syncevolution/test/search.py \
    -a '' -a peer-a04e041ead30 \
    --search '[]' \
    --read-all
```

Google Contacts Syncing with Username/Password

- Configure and test access to Google Contacts:

```
$ syncevolution --configure username=john.doe@googlemail.com \  
    password=foobar \  
    sslverifyhost=0 sslverifyserver=0 \  
    syncurl=https://www.googleapis.com/.well-known/carddav \  
    backend=carddav \  
    target-config@google addressbook
```

sslverify*=0 works around PTF-190

```
$ syncevolution --print-databases target-config@google addressbook  
$ syncevolution --export - target-config@google addressbook
```

- Configure and run sync:

```
$ syncevolution --configure --template SyncEvolution_Client \  
    syncURL=local://@google google addressbook  
$ syncevolution --sync slow google
```

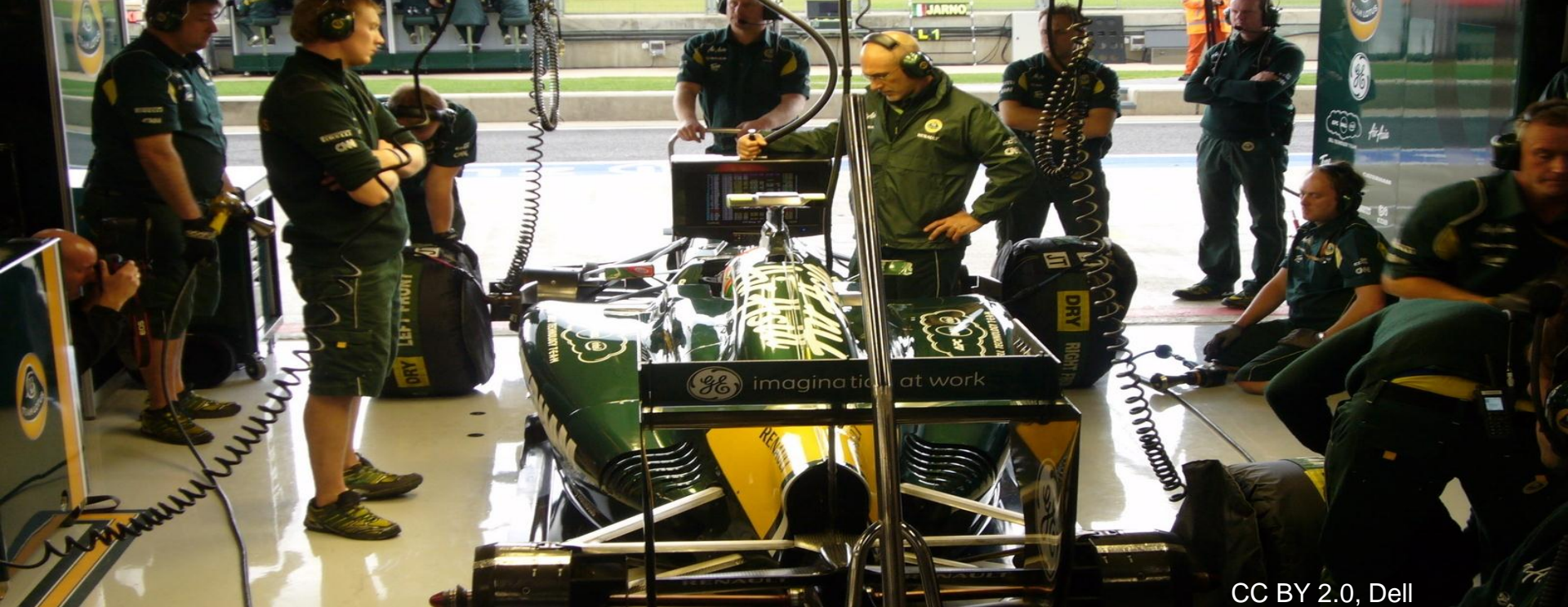
Google Contacts via OAuth2

- **Configure identity in gSSO:**

```
$ gsignond & # D-Bus auto-activation fails at the moment (TINF-588)
$ gss-example --create-identity=google-for-syncevolution \
  --identity-method=oauth --identity-realms=google.com
$ ID=1
$ for i in /usr/bin/syncevolution /usr/libexec/syncevo-*; do \
  gss-example --add-context=$ID --system-context=$i --application-context=; \
done # Depends on gSSO using file-based ACL in Tizen; may change.
```

- **Use username=signon:<parameters> and no password:**

```
$ syncevolution --print-databases \
"username=signon:{'identity': <uint32 $ID>, 'method': <'oauth'>, 'mechanism': <'oauth2'>,
'session': <{'TokenHost': <'accounts.google.com'>, 'ForceClientAuthViaRequestBody': <true>,
'Scope': <'email https://www.googleapis.com/auth/carddav'>, 'UiPolicy': <uint32 0>, 'ClientId':
<'73652887053-2ciia00v5fseed7s0sudggdu3oao2re.apps.googleusercontent.com'>, 'AuthPath':
<'/o/oauth2/auth'>, 'ResponseType': <'code'>, 'AuthHost': <'accounts.google.com'>,
'ClientSecret': <'2YDYzyI6HWSJFd5dOqz0uZGj'>, 'Realms': <['google.com']>, 'RedirectUri':
<'http://localhost'>, 'TokenPath': <'/o/oauth2/token'>}}" \
syncurl=https://www.googleapis.com/.well-known/carddav \
sslverifyhost=0 sslverifyserver=0 \
backend=carddav
# ClientSecret/Id are for syncevolution.org - get your own from Google!
```



CC BY 2.0, Dell

Getting Involved

Next Steps

- **In development:**
 - PBAP backend enhancements (PBAP 1.3, transfer in chunks)
 - PIM Manager windowed search
- **Ideas for IVI:**
 - Calendar support
 - Transparent access to contacts without caching
- **Needs community help:**
 - KDE
 - GTK UI

References

SyncEvolution:

- <https://syncevolution.org>
- <http://cgit.freedesktop.org/SyncEvolution/syncevolution/tree/src/dbus/server/pim/pim-manager-api.txt>
- <http://cgit.freedesktop.org/SyncEvolution/syncevolution/tree/src/dbus/server/pim/README>
- <http://cgit.freedesktop.org/SyncEvolution/syncevolution/tree/src/backends/pbap/README>
- <http://cgit.freedesktop.org/SyncEvolution/syncevolution/tree/src/backends/webdav/README>

Evolution Data Server:

- <https://developer.gnome.org/libebook/stable/EBookClient.html>

Other projects:

- <https://wiki.gnome.org/Folks/>
- <http://code.google.com/p/libphonenumber/>
- <http://code.google.com/p/googletest/>
- <http://code.google.com/p/protobuf/>



TIZEN™
DEVELOPER
CONFERENCE
2014
SAN FRANCISCO