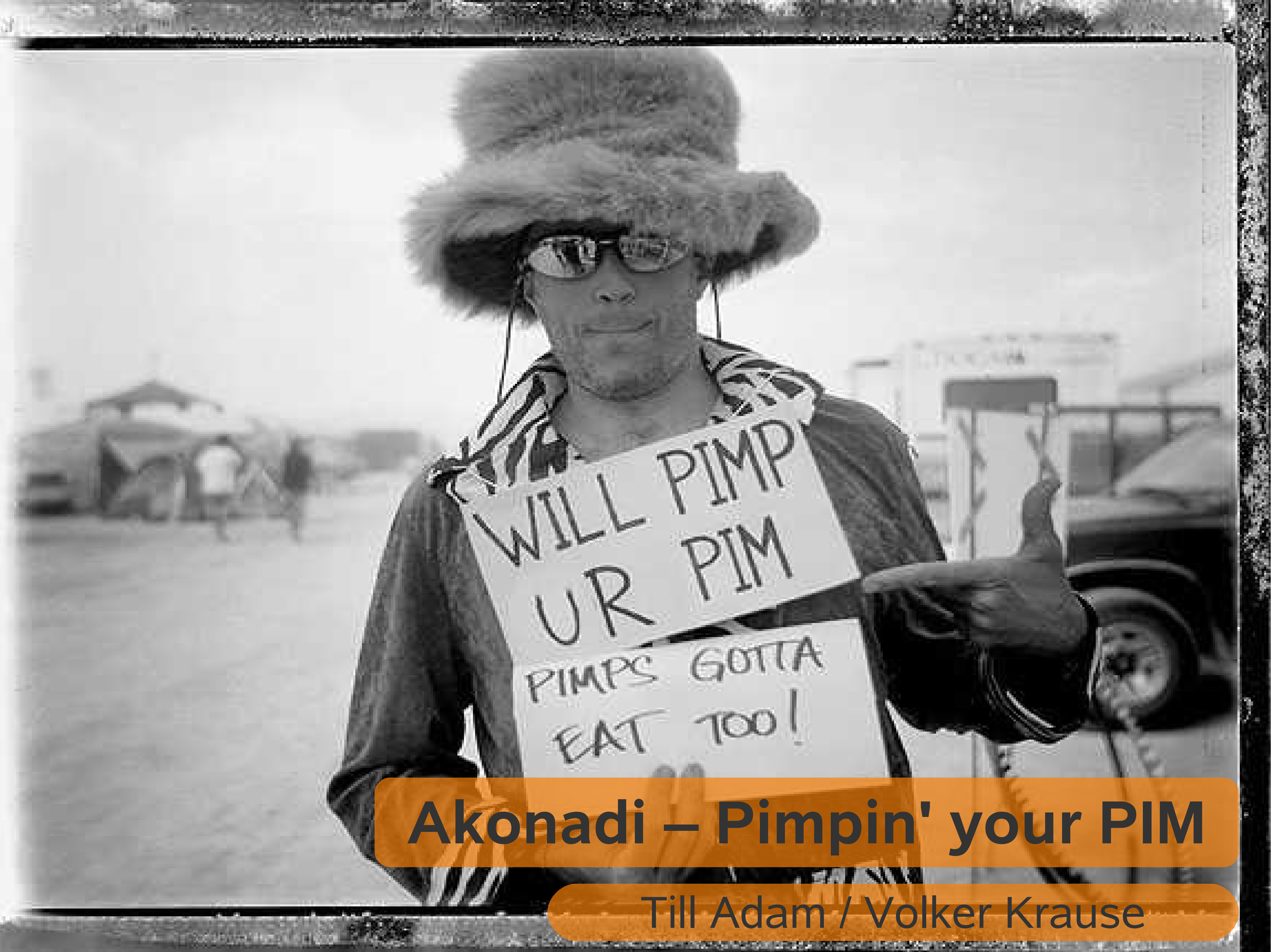


Akonadi – Pimpin' your PIM

Till Adam / Volker Krause

Topics

- Akona-what?
- Design Overview
- What we give you
- What you give us

A bit of history

Why?

Limitations of KDE3

- KResource framework limitations:
  - Data is not shared
  - Designed for synchronous access
  - Hard to extend to other data types
  - Basically no shared common code
- KMail limitations:
  - Only limited backend abstraction

➔ **Designed for small amounts of local data**

Why?

Goals

- As much as possible shared, type independent functionality
- Easy to extend to new data types
- Unified API to access PIM data, independent of the actual data source
- Scalability

Why?

Goals

- One synchronization point for mobile devices
- Reliable, desktop wide notification
- Clean model/view separation (UI-less data access)
- Easy to write access libraries for

➔ **Usable for the whole free desktop**

Why?

Enabling new use cases

“show me the log of the last IRC chat I had with the person who send me this mail”

Why?

Enabling new use cases

“show me all mails with pdf attachments mentioning my hamster 'cookie' right here inside my IM client, whenever someone mentions chicken curry”

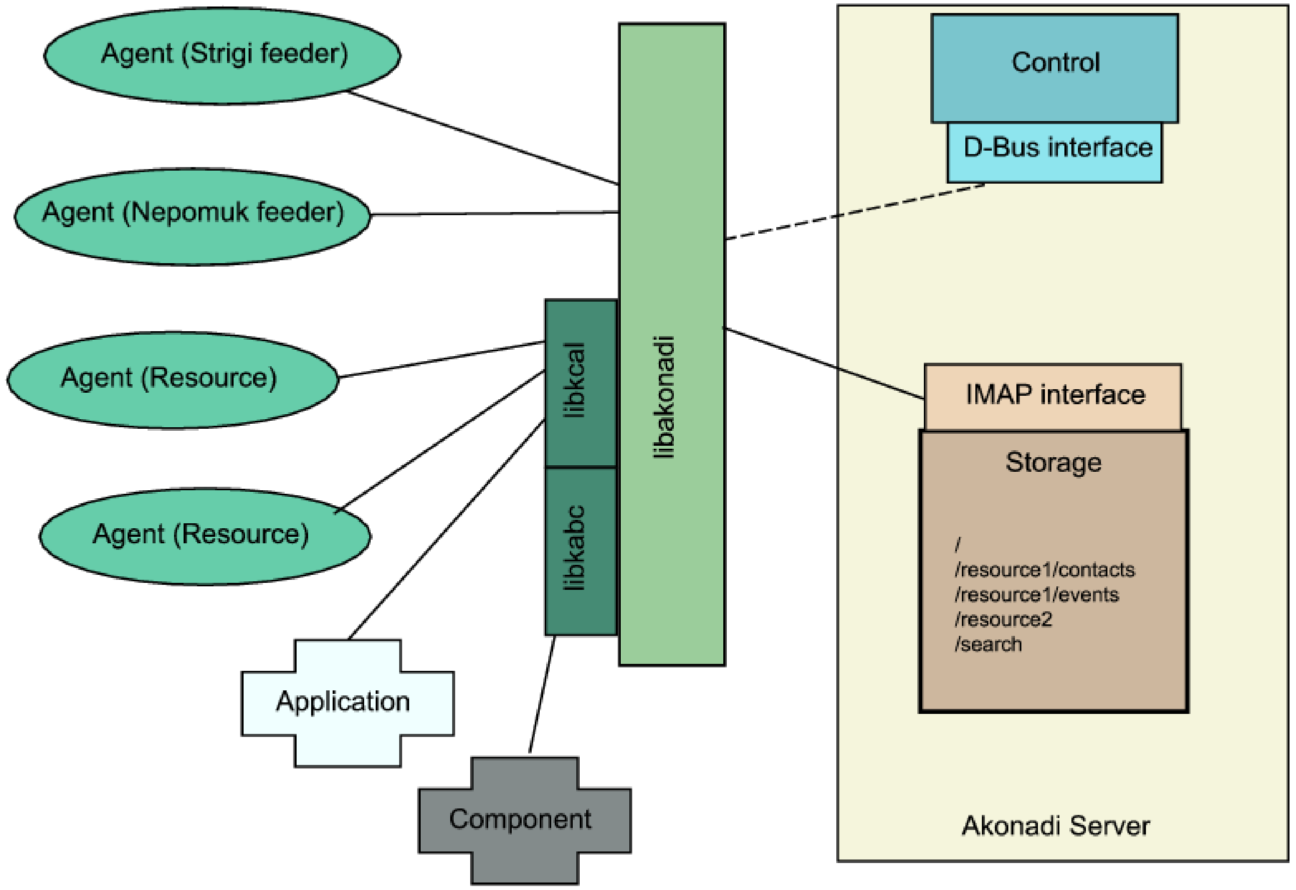
Why?

Enabling new use cases

“tell me when I get new mail in this folder and this other folder, and show it on the desktop, but only if it's not from my mom. show me a picture of the person next to it, and when I have an appointment with them, if I do. allow me to cancel that appointment by dragging it to the trash”

Design

Design Overview



Design

Server

- Fully type independent
- Cache for remote data with variable cache policies
- Change notification
- Conflict detection

Design

Basic Objects

- Filesystem-like structure:
  - Collections
  - Items
- Items can consist of multiple parts so clients can access only the actually needed data

Design

Client/Server Communication

- Two communication channels:
  - D-Bus for control data
  - IMAP-like protocol for content data
- Standard formats for content data (MIME, iCal, vCard, etc.)
- **Toolkit and language independent interface**

Client Libraries

- Currently only one: libakonadi, C++/KDE
- Consists of type-independent part and type specific plugins
- Provides low-level access to Akonadi objects as well as high-level components

Resource Agents

- Connect Akonadi to external data sources
  - local files (maildir, iCal, vCard, ...)
  - mail- or groupware servers
  - web services
- Translate data formats
- Replay offline changes

Other Agents

- Implement functionality not limited to one application as separate agents
- Existing agents:
  - Search index feeder
  - Mail threading
- Planned agents:
  - Filtering

What we give you

Is it ready?

The KDE project is proud to announce the immediate availability of the first public developer preview of Akonadi, code named

आकाशवाणी (Akashwani).

anonsvn.kde.org/home/kde/trunk/KDE/kdepim/akonadi

What we give you

Requirements

- Server:
  - D-Bus
  - Qt 4.3
  - MySQL Server binary, does not need to be configured and running
- Clients:
  - recent kdelibs + kdepimlibs

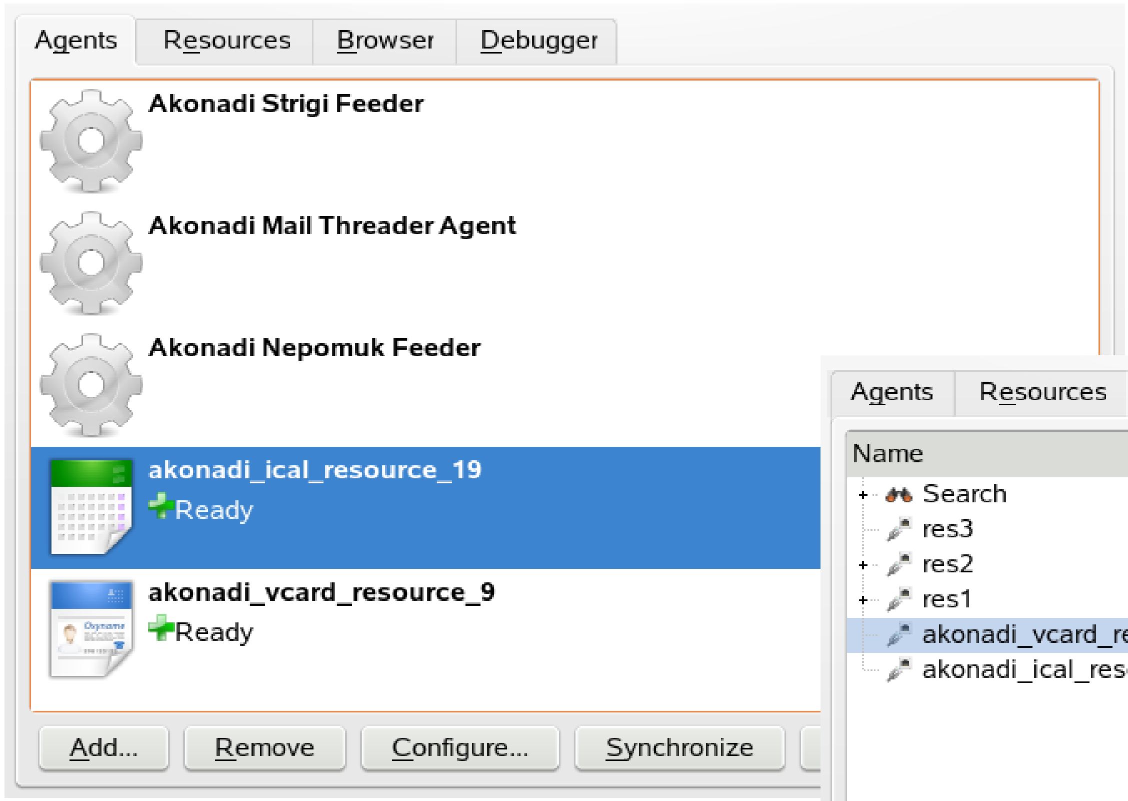
What we give you

How do I use it?

- Starting/stopping Akonadi:

  akonadictl start/stop
- Akonadi Console:
  - Manage resource agents
  - Browse content
  - Watch client/server communication

What we give you



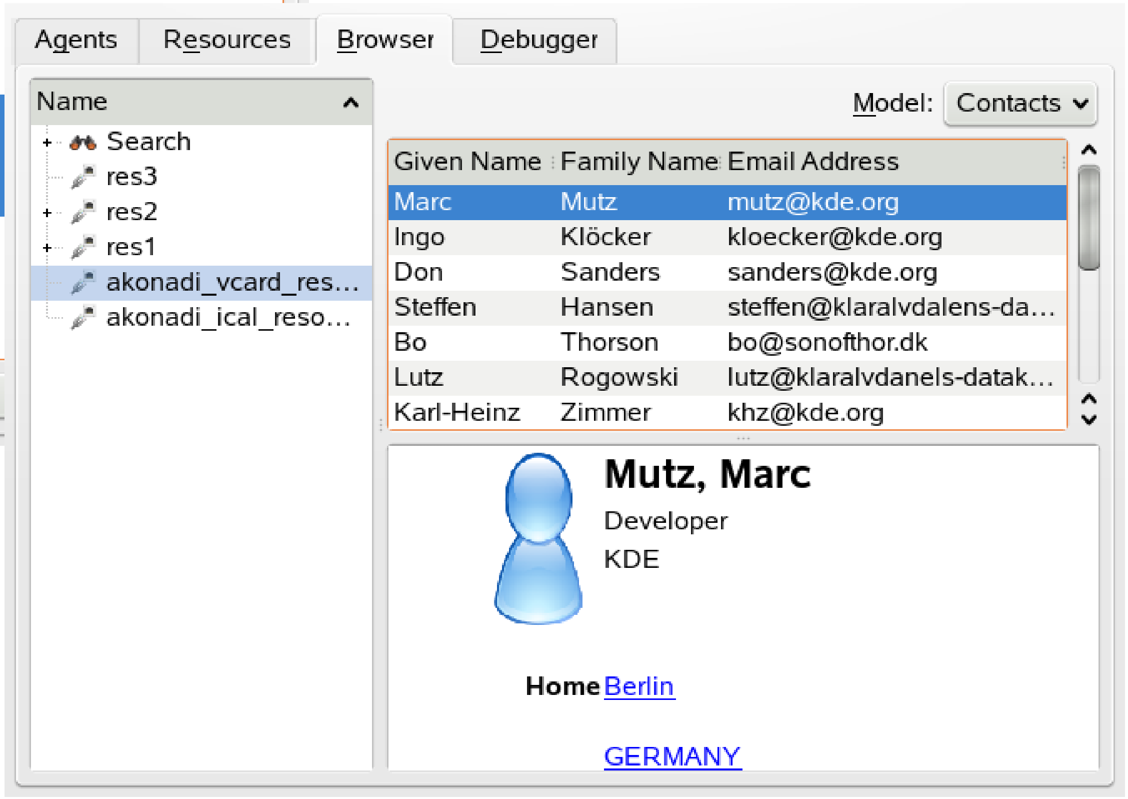
Akonadi Strigi Feeder

Akonadi Mail Threader Agent

Akonadi Nepomuk Feeder

akonadi_ical_resource_19
Ready

akonadi_vcard_resource_9
Ready



Given Name	Family Name	Email Address
Marc	Mutz	mutz@kde.org
Ingo	Klöcker	kloecker@kde.org
Don	Sanders	sanders@kde.org
Steffen	Hansen	steffen@klaralvdalens-da…
Bo	Thorson	bo@sonofthor.dk
Lutz	Rogowski	lutz@klaralvdanels-datak…
Karl-Heinz	Zimmer	khz@kde.org

Mutz, Marc
Developer
KDE

Home Berlin

GERMANY

Roadmap

- Finish the basics
  - Add missing features needed by exiting applications
  - Performance optimizations
- Port existing KDE PIM applications
- Extend Akonadi further beyond KDE3 possibilities: eg. virtual collections/search

Contribute

What can I do?

- Contributing to the Akonadi server
- Additional Client library implementations
- Language bindings
- Add resource agents for new backends
- Extend Akonadi to support new types
- Using Akonadi in applications

Contribute

Akonadi Server

- Fulltext/Metadata indexing & search:
  - Virtual collections
  - Search in backends (IMAP, LDAP, etc.)
- Additional features:
  - Local subscriptions / Subscription profiles
  - Filtering
- Performance/Database optimization

Client Libraries

- Currently only one: KDE/C++
- Possible approaches:
  - Native implementations:
    - Native data types, easy integration
  - Language bindings:
    - Scripting languages, RAD

Extending Akonadi

- Support additional backends: groupware servers, web services, ...
- Support for additional data types: Notes, IM messages, RSS feed entries, ...
- See “Developing PIM applications with Akonadi” in room “A” right after this talk

Using Akonadi in Applications

- Port existing applications
- New possibilities:
  - Integrate PIM data wherever useful:
    - Every mail address can be linked to your addressbook
    - Every date can be linked to your calendar
  - Plasma applets / Desktop widgets

Further Information

- IRC: #kontact on irc.freenode.org
- Mailinglist kde-pim@kde.org
- http://pim.kde.org/akonadi
- Talk: “Developing PIM Applications with Akonadi” in room “A” right after this talk