Akonadi – The independent solution for PIM data

Will Stephenson

Akonadi
the independent solution for PIM data

Will Stephenson
Topics

Akona-what?
Design Overview
What we give you
What you give us
The story so far

Monolithic apps
Own data storage
Limited if any external interfaces

E-D-S
Data storage service
Limited range of types supported
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
Scalability with KDE 3

|Data| = small

Contact applet

KMail

KAddressBook

Kopete
Scalability in KDE 3

|Data| = large

Contact applet

KAddressBook

KMail

Kopete
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
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
Scalability in KDE 4 with Akonadi

|Data| = large

- Contact applet
- KMail
- Kopete
- KAddressBook
- Akonadi
Enabling new use cases

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

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

Filesystem-like structure:
Collections
Items

Items can consist of multiple parts so clients can access only the actually needed data
Items can be polymorphic
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
Requirements

Server:
D-Bus
Qt 4.5
MySQL Server binary, does not need to be configured and running

Clients:
recent kdelibs + kdepimlibs
How do I use it?

Starting/停止 Akonadi:
  akonadictl start/stop

Akonadi Console:
  Manage resource agents
  Browse content
  Watch client/server communication
Akonadi – The independent solution for PIM data

GCDS 2009

What we give you

<table>
<thead>
<tr>
<th>Name</th>
<th>Model:</th>
<th>Contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given Name</td>
<td>Family Name</td>
<td>Email Address</td>
</tr>
<tr>
<td>Marc</td>
<td>Mutz</td>
<td><a href="mailto:mutz@kde.org">mutz@kde.org</a></td>
</tr>
<tr>
<td>Ingo</td>
<td>Klöcker</td>
<td><a href="mailto:kloecker@kde.org">kloecker@kde.org</a></td>
</tr>
<tr>
<td>Don</td>
<td>Sanders</td>
<td><a href="mailto:sanders@kde.org">sanders@kde.org</a></td>
</tr>
<tr>
<td>Steffen</td>
<td>Hansen</td>
<td><a href="mailto:steffen@klaralvdalens-datak.de">steffen@klaralvdalens-datak.de</a></td>
</tr>
<tr>
<td>Bo</td>
<td>Thorson</td>
<td><a href="mailto:bo@sonofthor.dk">bo@sonofthor.dk</a></td>
</tr>
<tr>
<td>Lutz</td>
<td>Rogowski</td>
<td><a href="mailto:lutz@klaralvdalens-datak.de">lutz@klaralvdalens-datak.de</a></td>
</tr>
<tr>
<td>Karl-Heinz</td>
<td>Zimmer</td>
<td><a href="mailto:khz@kde.org">khz@kde.org</a></td>
</tr>
</tbody>
</table>
Roadmap

Optimisation and server diversification
Port existing KDE applications
More native resources
More client libraries
Optimisation and server work

Optimisations in the API implementation
Making the client API richer based on porting experience
Postgresql backend
Sqlite backend
External mysql instance support (thin client)
Unit tests
KDE Porting progress (1)

Until now, bridge resources access Akonadi via KResource API
Some apps ported completely (KJots, Mailody)
Others in progress (Akregator, KPilot, KNode)
KAddressBook being reimplemented
(codename KContactManager)
KDE Porting progress (2)

Big apps (KMail/KOrganizer)
Refactoring to allow port
Simultaneous development on Model/View components for Akonadi
KDE Porting progress (3)

Mail migrator
Akonadi outbox agent – procmail

http://techbase.kde.org/Projects/PIM/Akonadi/PortingStatus
More native resources

SyncML agent
Google data resource
Exchange resource
IMAP resource
Semantic data extraction and semantic search
folders
Kolab groupware
More client libraries!

Additional Client library implementations
Language bindings
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: IM messages, [micro]blogs, CRM/ collaboration

See “How to write an Akonadi resource in 30 minutes” next!
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

Next talk!
KMail 2 – The Road to Akonadi – Mon 1715
http://techbase.kde.org/Projects/PIM/Akonadi/PortingStatus
Take away message

Engineered for performance
Engineered for flexibility
Engineered for independence