Akonadi
The KDE4 PIM Framework

Tobias Koenig
Overview

- Why a new PIM Framework
- Akonadi
  - History
  - Concepts
  - Current State
  - The Future
- Questions & Answers
Problems with old Framework

- Bad Performance
- Synchronous Access
- Memory Consumption
- Missing Notifications
- Missing Locking
Akonadi

- A general storage for PIM data
- Aims to solve the problems of kabc/kcal
- Modular design for robustness and easy integration
History

- First ideas of a PIM daemon at Osnabrück meeting 2005
- Sample code for address book daemon
- General agreement for PIM service at Osnabrück meeting 2006
- 'Akonadi' was born
Concepts
Concepts
The Storage

- Accessible via extended IMAP
  => high performance on data delivery
- Caches all PIM items of the resources
  (depending on the policy)
- Informs the Control about changes
- Provides basic search features (IMAP)
- Delegates extended search requests to SearchProviders
The Resources

- Applications which synchronize data between the Storage and an external data source (e.g. groupware server or file)
- Asynchronous communication
- Profiles for grouping Resources together
The Control

- Starts and monitors the Storage and resource processes
- Provides D-Bus API for managing resources and profiles
The SearchProviders

- External applications which process more complex search queries
- One SearchProvider for every PIM item type
- Store search results in collections inside the Storage
Collections

- Virtual containers inside the Storage
- Contain PIM items/references
- Root collection ’/’
- Every Resource provides at least one collection
- Easy to realize virtual folders

/  
/resource1  
/resource1/calendar  
/resource1/addressbook  
/resource2  
/search
Current State
Current State

- **Storage**
  - Provides all needed IMAP commands
  - Stores items in Collections

- **Control**
  - Lifetime management for Storage and Resources
  - D-Bus API nearly completed

- **Resources**
  - Simple iCalendar file
  - Dummy resource 'Knut'
Demo...
TO DO

- Implementing SearchProviders
  (mail, calendar, address book)
- Defining asynchronous client library
- Implementing full featured resources
  (remote and local file)
- Writing documentation!!!
The Future
The Future

- Goal: Components based PIM library
- Components as standalone views and editors of PIM data
- Plasma applets for calendar and address book
We need developers!
Questions?
Thank you