OpenOffice.org

http://www.openoffice.org

- Open Source Office Suite
- ~1.5 million downloads a week
- ~800,000 daily unique visitors
- ~100 localizations
- ~7 million LOC (98% C++)
- ~120 million installations
- 10 platforms
- 8 machine architectures
- Various C++ Compilers
Architectural Challenges

• Huge code base
• Monolithic architecture
• Different programming languages
• Big problems because of C++ / C++ ABI
• Accidental incompatible changes

Required: Stable API for 3rd party developers
Architecture
Solution: Divide (into Software Components) and Conquer
Component Architecture

- Have a component model API&ABI only (rest described e.g. by IDL)
- Create a program out of components only
- Recursively refine the architecture
Current Flavors of Components

http://extensions.services.openoffice.org

- 3rd Party Functionality
- Input/Output (UCB)
- Filters
- Remote Communication
- Mathematical Functions
- BASIC Libraries
- Uno Services
- New Chart Types
- Spellcheckers
- Templates
- Cliparts
- System Integration

New stuff developed as components

3rd Party components are mostly Java
Universal Network Objects (I)

http://udk.openoffice.org

- IDL based
- Exceptions
- Multi threaded
- Multiple inheritance
- No overhead in case of co-location
- Interoperable between various programming languages
- In process / over process, machine boundaries -> remote transparent
Universal Network Objects (II)

• Independent (standalone usable)
• Cross Platform
• Extensible
  > Bridges
  > Services
  > Interfaces
• Uses / supports components
• Unicode based

Reported to be fast :-(
Uno Architecture

UNO

Specifications
- Services
- Remote
- Type Sys
- Runtime
- Types

Bindings
- Bin-UNO
- Java-UNO
- Py-UNO
- C++-UNO

Adapters
- BASIC
- OLE/COM
- CLI (.Net)
- Web Serv.

Languages
- C
- C++
- Python
- Java
- OLE
- CLI
Component Problems

- Different semantics of competing components
- Semantic changes
- Interface Inflation
- Who owns the process?
- Who owns an object?
- Bypassing
- Deployment: System vs. User
Software Components

• To unitize a big software program
• To enable distributed development
• To enable independent / 3rd party developers
• To enable competitive implementations
• To design an architecture
SOFTWARE COMPONENTS IN OPENOFFICE.ORG

Kay Ramme
> kay.ramme@sun.com