Silenus

Max Berger <max@berger.name>,
Michael Sobolewski <sobol@cs.ttu.edu>
SORCER Research Group, Texas Tech University <sorcer@cs.ttu.edu>

Presentation for CE2005

Overview

Technical Background:

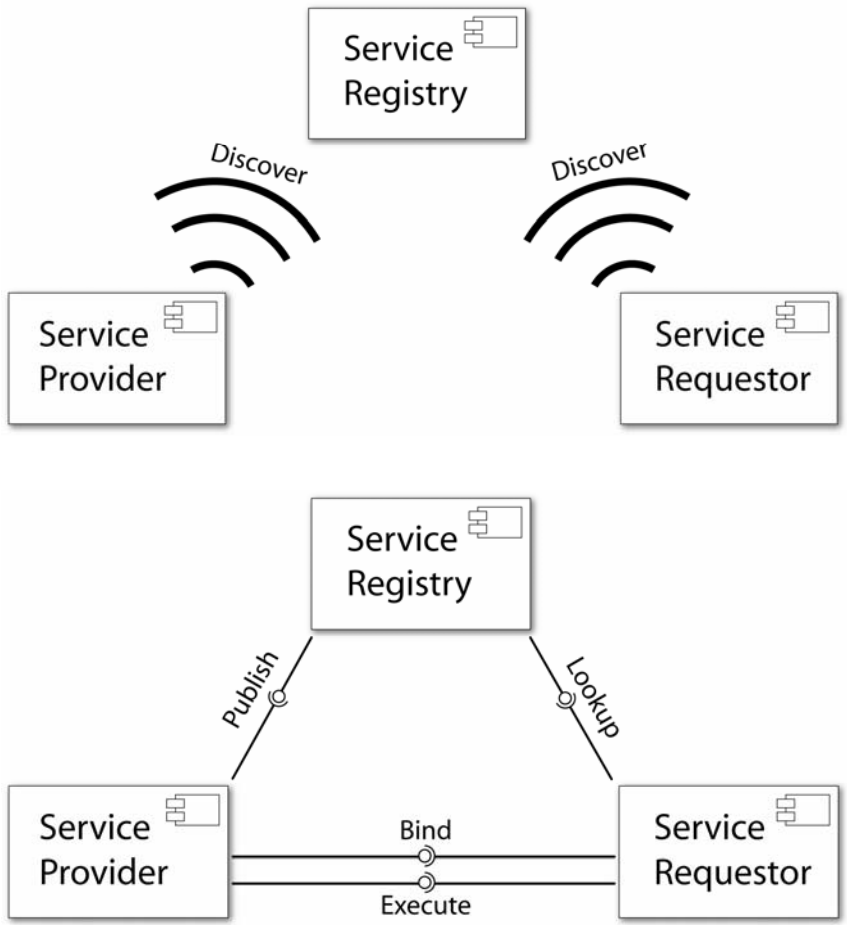
- Service Oriented Architecture
- SORCER

Silenus:

- Silenus Components
- Silenus in action

Service Oriented Architecture

- Discovery mechanism
- Service Registry
- Registers a Smart Proxy
- Lookup by Interface
- Protocol independent



SORCER Infrastructure

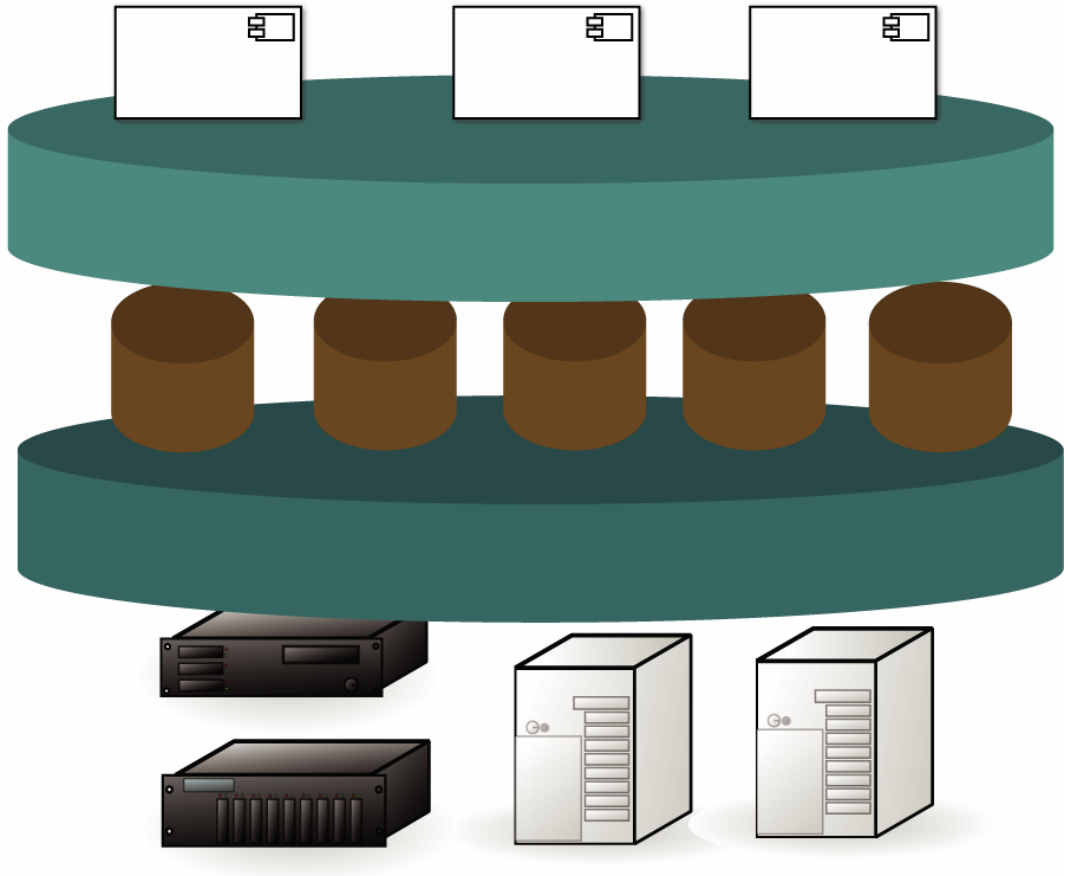
Virtual Services

Dynamic Provisionig Grid

Cybernodes

Grid Resource Management

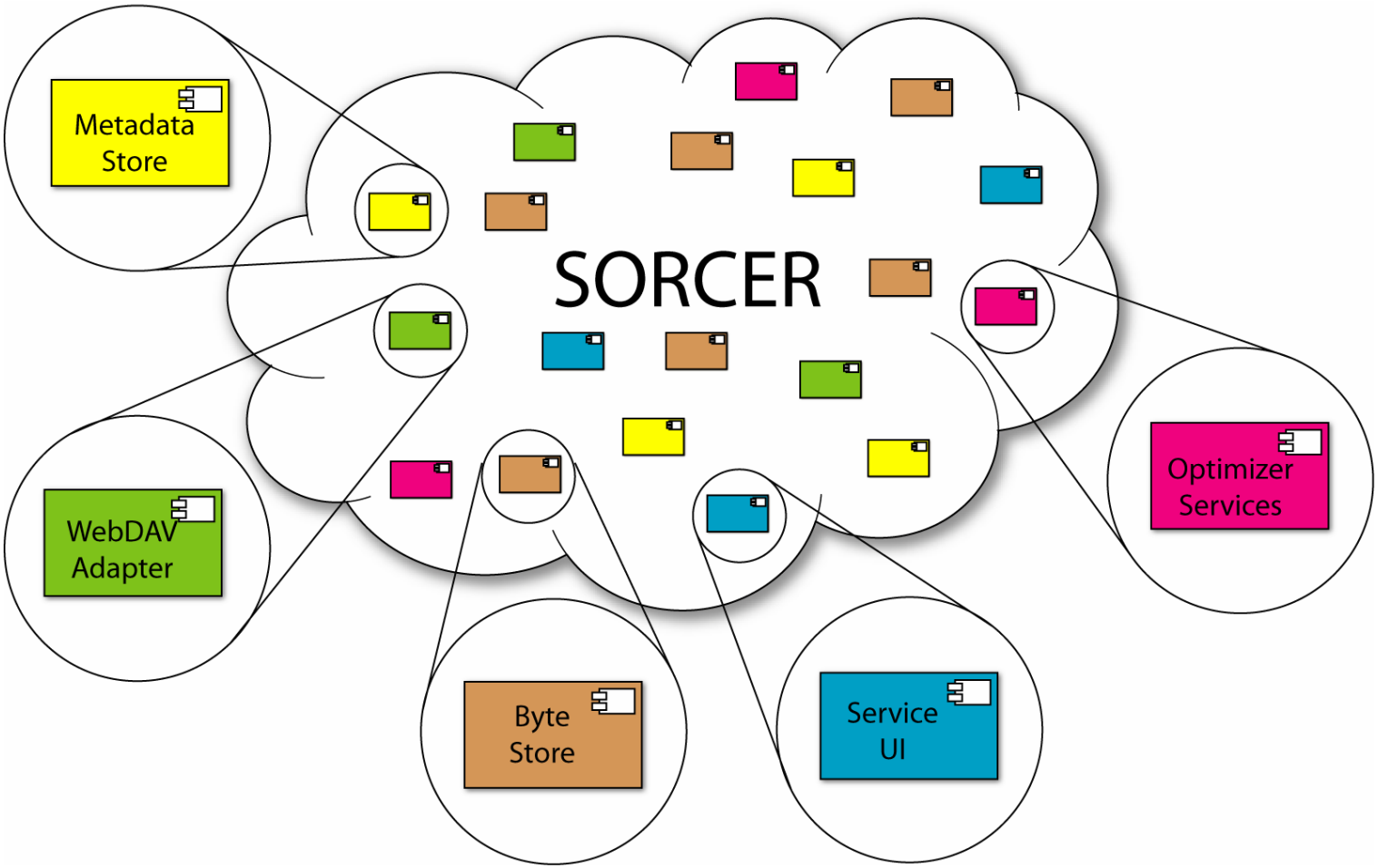
Computing Resources



Sorcer's Integrated Local Enhanced New Storage

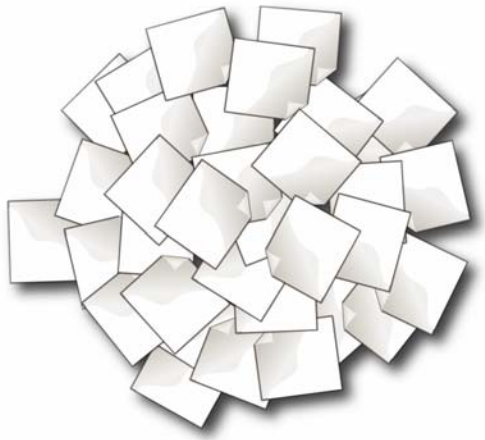
A Service Oriented Application

SILENUS Components



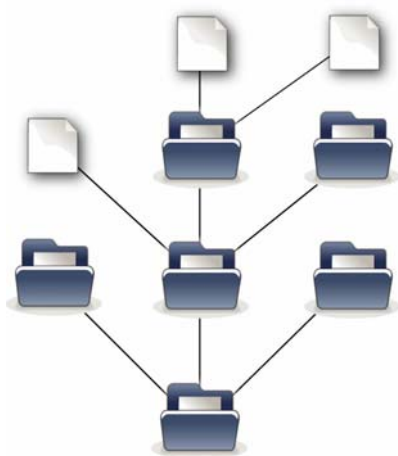
Byte-store

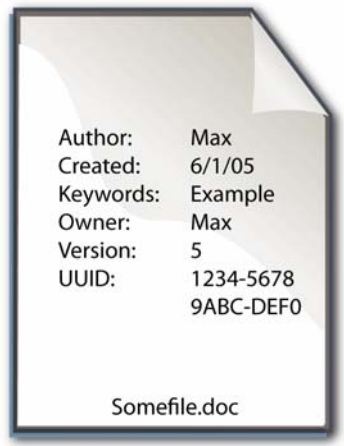
- Unordered Collection of Files
- Files are Identified by UUID
- Files are Stored encrypted
- Byte-store is optimized for fast transfer
- Byte-stores can contain different files



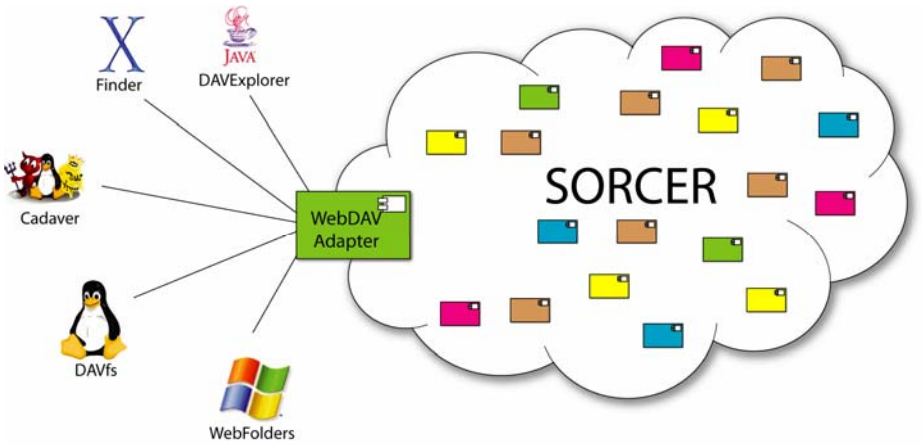
Metadata-store

- Orders Files in hierarchical structure
- Supports Files, Folders and Links
- Stores metadata information
- Information across all connected metadata-stores is synchronized



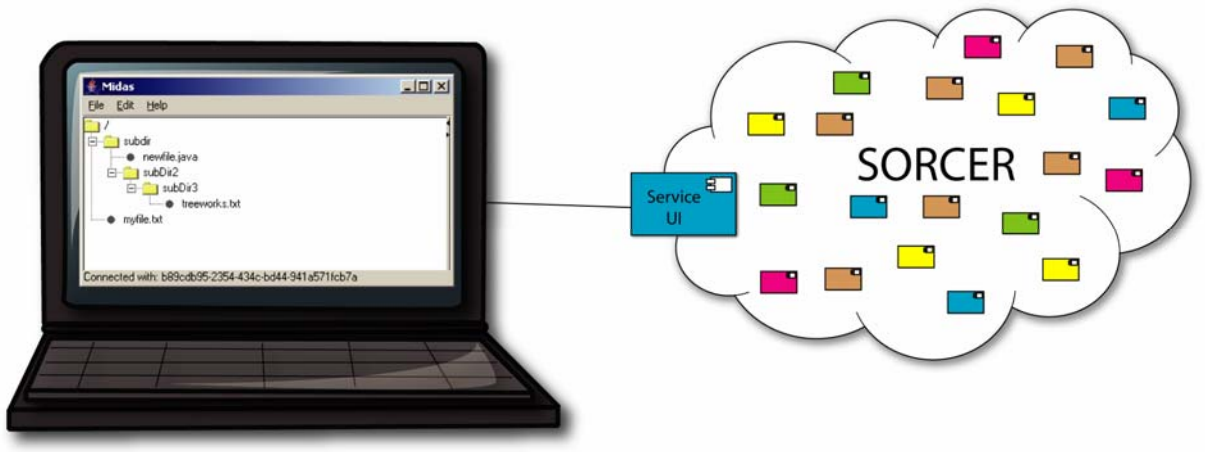


WebDAV adapter



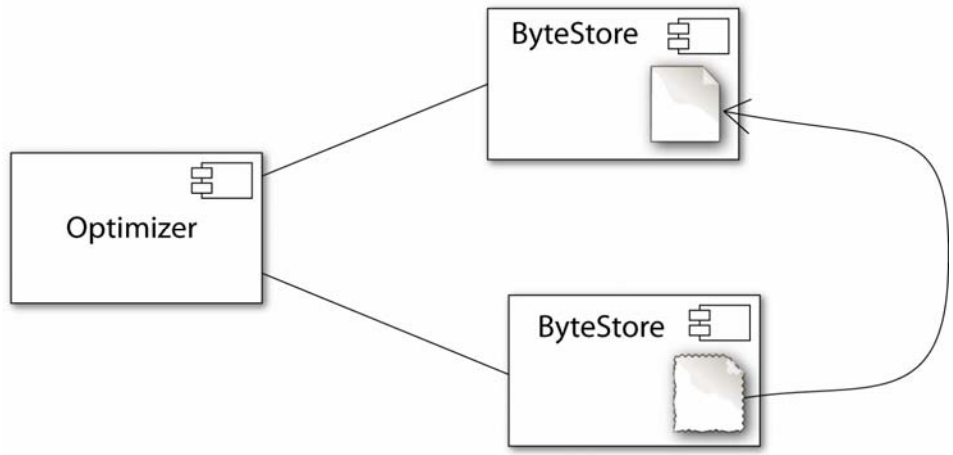
- WebDAV is an Internet standard (RFC 2518)
- Well supported in most common operating systems
- No need to write special drivers
- Accessible like local file system

Service UI



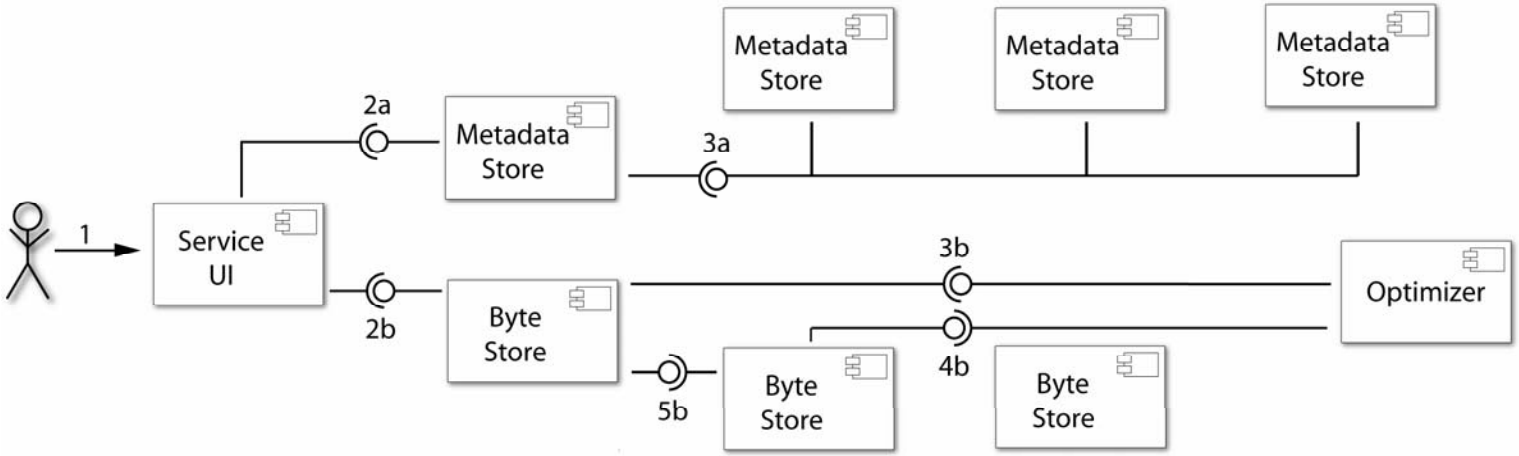
- Service UI is accessible via service browser
- Offers full functionality
- No need to install SILENUS specific software
- No need to configure client computer

Optimizer services



- Not one, but several optimizer services
- Keep the network “in good shape”
- Can auto deploy services among nodes
- Can move / replicate files among bytestores

File Upload



1 User initiated file upload

2a Upload metadata information

3a Publish metadata information to other metadata stores

2b Upload file

3b Inform optimizer about new file

4b Tell other byte-store to initiate transfer

5b Transfer files directly between byte-stores

Conclusions

The system provides

- Flexibility through discovery
- Scalability through independent providers
- Architecture independence through Java
- Privacy through encryption
- Fast data transfer through caching
- Protection against data loss through replication