Stephan Körnig

User Needs and Digital Libraries Design (2):

Design Principles for Digital Library Services

Objectives

- Switching to the aspects of technology / modeling of applications
- Focus on user driven development for “Scientific Networks”
- Stress on the aspects of DL-service provision

Outline

- User driven development
- Design principles for Digital Library Services
- Protocols and their usage
- Problems of technology driven development
- Conclusions

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

User driven development

- Roles in behalf of systems and applications
- Use Case modeling
- The “Unknown user”
- Exercise

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Types of Roles

- Three different types of roles
  - Content provider
  - Developers
  - Users
- A single person or institution will interact in different roles

User driven develop-ment

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Roles: different views

Service	Provider	Developer	User
Notification / Profile Service	How to reach possible interested users, costs of metadata, …	Scalability, database model, heteroge-neity of systems, …	How to configure, no loss off interesting material, security of personal information,…

Use case modeling – Background (1)

- Analysis of „using the system“
  - Workflow / business processes
  - User behavior and requirements
  - Dependencies between system components

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Use case modeling – Background (2)

- Traditional: paperwork as part of the first analysis
- OO-development: the „user“ as part of the project team
  - Shaping system elements as objects (abstract data types) according to the user‘s view [idealiter]
  - Understandable diagrams with a real meaning for developers

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Use case modeling in UML (1)

- UML = Unified modeling Language
- Different diagram techniques
  - Type systems
  - Sequences
  - Interactions
  - State machines
  - …
  - and Use Cases

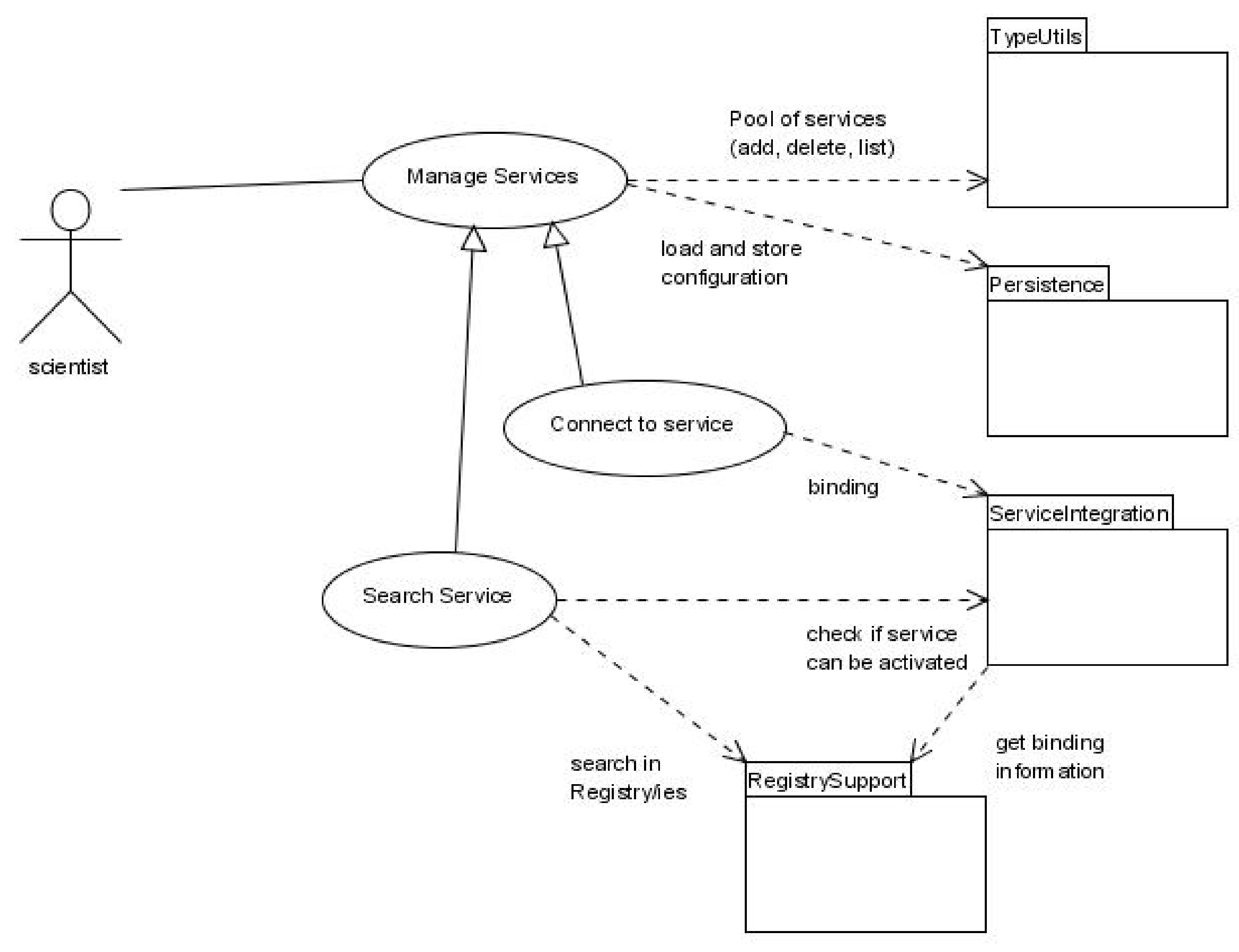
User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion



User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Use case modeling in UML (2)

- Some of UML’s offers
  - Supports the whole process from design to implementation
  - Documentation according to different views
  - Reengineering of existing code
  - Refactoring
  - Code Generators
- Is this the ultimate solution to meet the user‘s needs?

User driven develop-ment

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Use case modeling in UML (3)

- Developers
  - No **red button**: Code generators need as much „input“ as coding
  - Re-engineering and refactoring works good
- „User‘s view“
  - Use cases are informal
  - Correct implementation is not guaranteed
- Integration of users in project

User driven develop-ment

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

The „unknown user“

- Users know very well
  - What they need for their work
  - If a working system is what they need
- Users often do not know
  - How to shape an innovative and more powerful system that replaces the existing one

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Exercise

Literature:
Alistair Cockburn, „Writing Effective Use Cases“, 2001

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Design principles for DL-Services

- SOA
- „Web Services“
- REST
- How to model a distributed landscape?
- Discussion: Divergence of Web Services specs and REST

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SOA Background

- Service Oriented Architecture
- New buzzword
- Promoted by the „big players“
- Closely related to the „Web Services“ activity of the W3C (driven by the same players)

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SOA Basics

- Services as a Component
  - Platform-independent „interface contract
  - Dynamic service localization
  - Self-containg: service maintains its own state
- Communication via messages
- W3C „Web Services“ as a framework for SOA

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Web Services

- Just another hype?
- Basics
- Perspectives
- Drawbacks and limitations

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Web Services – the Hype

- What does this term imply?
  - Every service in the web is a web service
  - Generic approach
- Support by the big players
- Open standardization process
- Self-describing capabilities
- XML based

User driven development

Design principles

Protocols and their usage

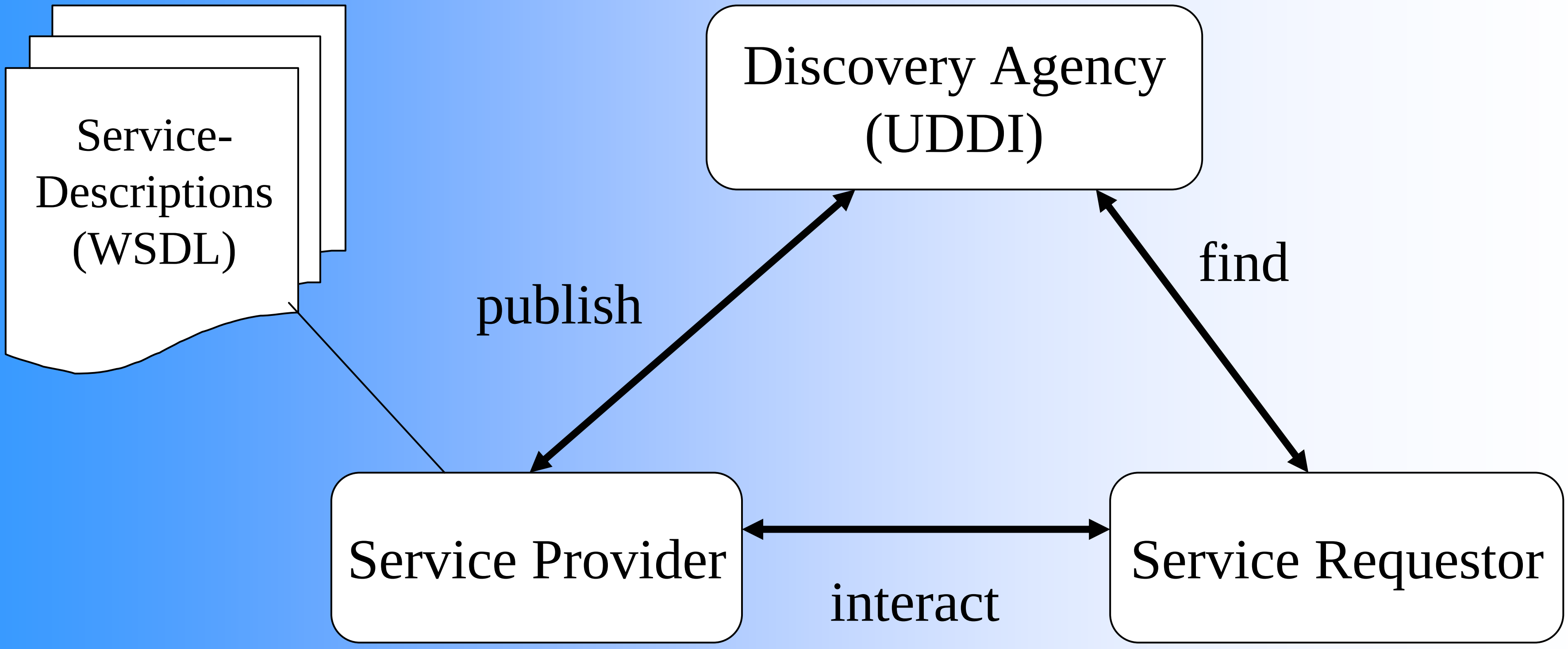
Layered technology and the Semantic Web

Conclusion

Web Services Basics

- RPC (Remote Procedure Calls) via HTTP
- Interaction of loosely coupled and reusable components
- Integration of legacy systems
- Machine readable interface descriptions (WSDL)

Provision of Web Services



User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Web Services - Perspectives

- Generic, all-purpose specification based on XML
- Possibilities for automated workflows
- Allows visibility of offers (services, content,...)
- Needs frameworks to be useful for a specific domain

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Web Services – Open Problems

- Still undergoing changes
- WSDL tries to support divergent concepts
  - Object oriented
  - Functional programming
  - Relational database modeling

➡ Automated processing of „any to any“ not possible

User driven develop-ment

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

REST - Background

- „Representational State Transfer“
- Roy T. Fielding 2000: „Architectural Styles and the Design of Network-based Software Architectures“ (Dissertation)
- Model for the modern Web architecture

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

REST – Concepts (1)

- Utilizes the widely accepted standards
  - URI (addressing, localization)
  - HTTP (communication)
  - HTML (links)
- Basic concept: resource
  - Representation, not the object itself
  - Accessible via URI

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

REST – Concepts (2)

- Stateless Client-Server interactions
  - Each request contains all necessary information to be executed
  - Server status is always unknown
- Resources marked as "cacheable" or "non-cacheable"
- Code on demand

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

REST - Perspectives

- Not technology-dependent
- URI-space instead of layered architectures
- Allows straightforward implementations (no protocol stack overhead)
- Not compliant to the vision of the “machine-readable” Web

User driven develop-ment

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Modeling the Landscape

- Vendors try to do so – we should even not try…
- Prescribing a technology will not be accepted
- Investments in metadata and conversion to XML-formats
- Available technology offers an infrastructure for DL-Services

User driven development

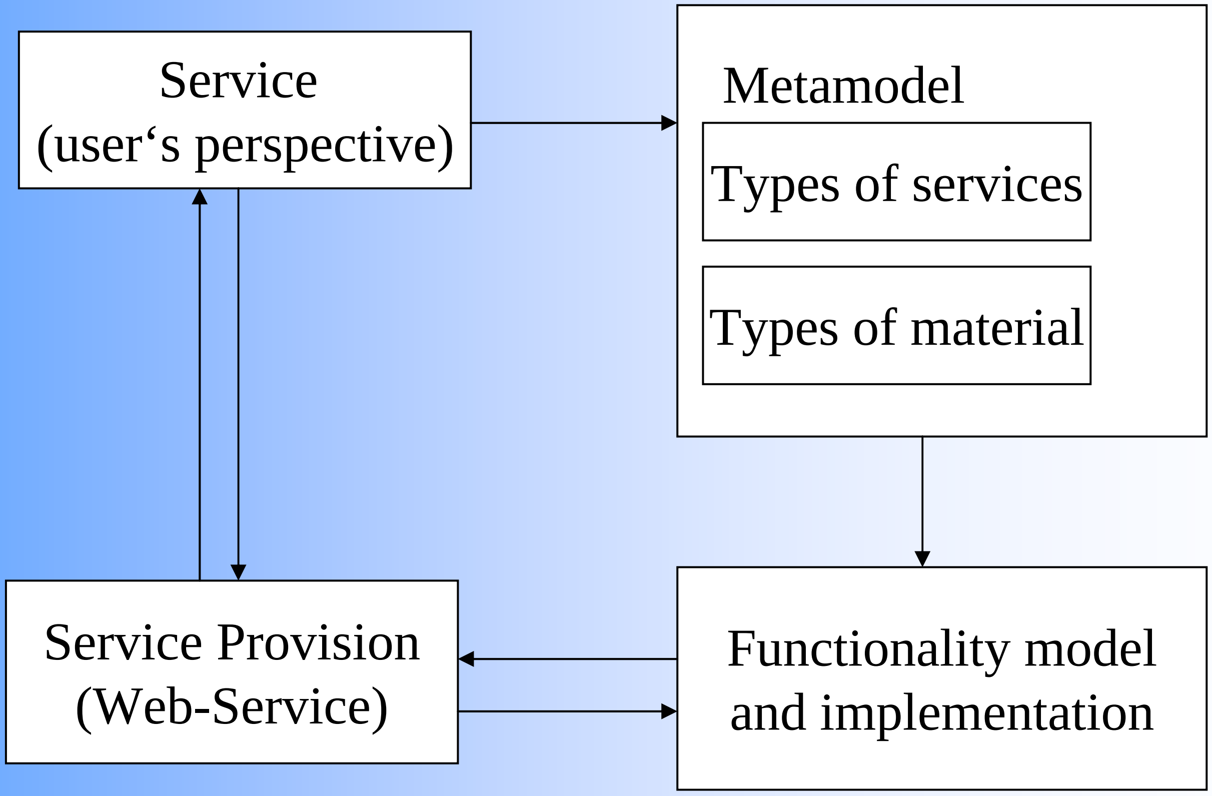
Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

DL-Services (logical view)



User driven development

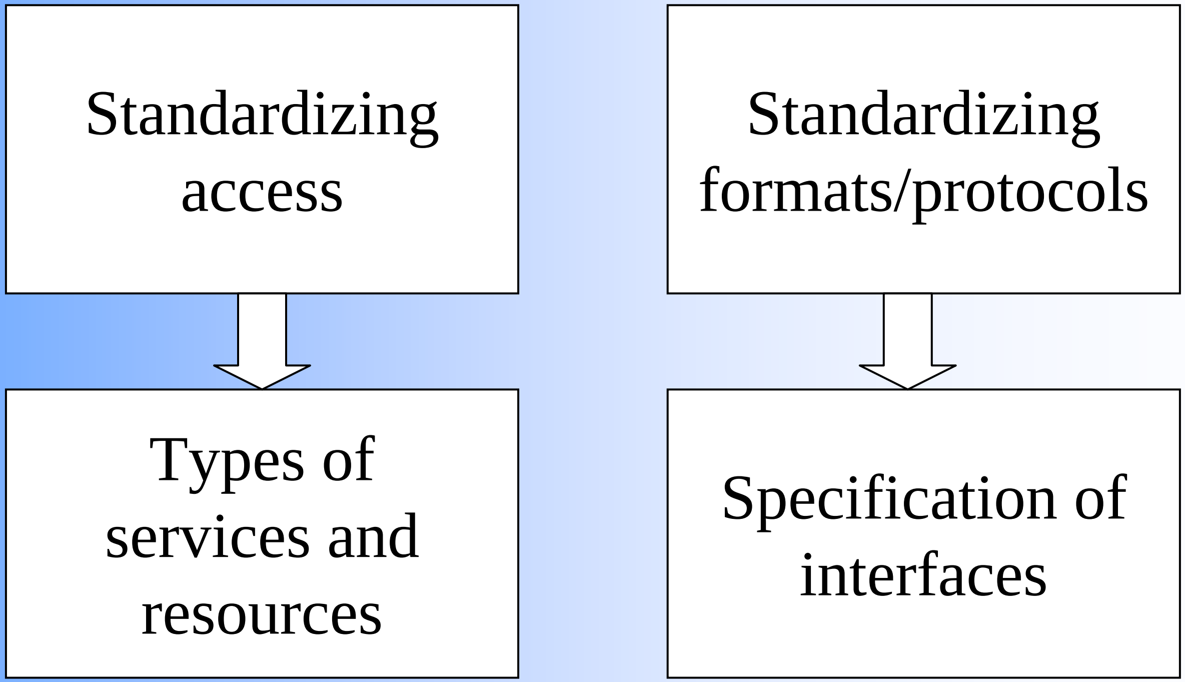
Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Access and Formats



Standardizing access

Types of services and resources

Standardizing formats/protocols

Specification of interfaces

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Type System: Services

- Authentication/Identification
- Search for services/content
- Management of structured research/education related information
- Communication
- Serializing/Rendering
- Archiving

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Type System: Types of Resources

- Application specific data
- Data structures (aggregates of basic types)
- Documents
- Metadata
- References (links, identifiers, handles, …)

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

REST-style Interfaces

- Foundation is the REST model
  - HTTP: GET, PUT, DELETE und POST
- General operations
  - Insert
  - Read
  - Update
  - Search
  - Delete

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

REST-style Interfaces

- Services are accessible via an URL
- Explain service as starting point
- Example: Typekit (www.typekit.org)

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Protocols and Usage

- Protocols and Standardization
- Z39.50
- OAI-PMH
- XML-RPC
- SOAP
- SRW / ZING
- WebDAV
- On „Layered Technologies“

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Protocols and Standardization (1)

- Supporting standards nowadays is a „conditio sine qua non“
- Standards as a part of the problem
  - Standardizing of own products is an advantage in the market
  - Too high/low level of details, due to strategic considerations
  - Standardizing is research

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Protocols and Standardization (2)

- Protocol = standard?
- Open and de facto standards
- Conclusion:
  - Use the most promising standards not in technological but in user‘s perspective
  - Reuse available standards whenever possible
  - Make the usage of standards explicit (e.g. Registries, Typekit, Explain-Services, etc.)

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Z39.50 (1)

- Protocol to access OPACs (Online Public Access Catalog)
- Specifies services from the user‘s (librarians) point of view
- Copes with heterogeneity
- Can be utilized for distributed services
- Can support distributed systems / huge amount of (bibliographic) datasets

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Z39.50 (2)

- Connects client stateful to server
- ASN-1 for data exchange
- Core services for search and retrieval
- Extended services to support the use case “OPAC”
- Explain Service
- Supports diverse bibliographic formats

User driven development

Design principles

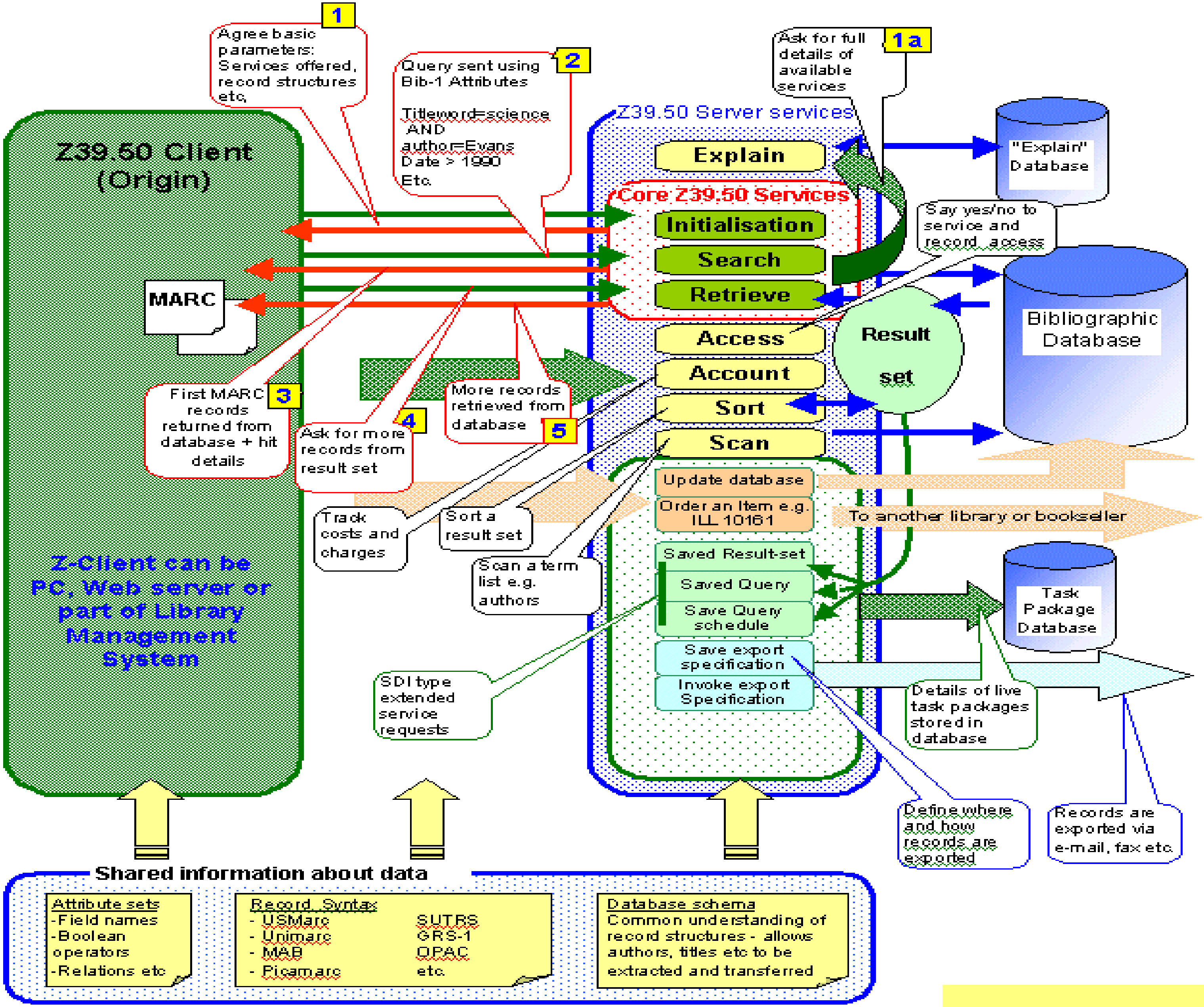
Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Z39.50 (3)

- ASN-1 is obsolete
- Heavy-weight protocol
  - Very rich functionality
  - Stateful protocol
  - Hard to implement
- Not widely supported
- Incomplete implementations


1
Agree basic parameters: Services offered, record structures etc.
2
Query sent using Bib-1 Attributes
Titleword=science AND author=Evans Date > 1990 Etc.
1a
Ask for full details of available services
Z39.50 Client (Origin)
MARC
Z39.50 Server services
Explain
"Explain" Database
Core Z39.50 Services
Initialisation
Search
Retrieve
Say yes/no to service and record access
Bibliographic Database
Access
Account
Sort
Scan
Result set
3
First MARC records returned from database + hit details
4
Ask for more records from result set
More records retrieved from database
5
Track costs and charges
Sort a result set
Scan a term list e.g. authors
Update database
Order an Item e.g. ILL 10161
To another library or bookseller
Saved Result-set
Saved Query
Save Query schedule
Task Package Database
Save export specification
Invoke export Specification
Details of live task packages stored in database
Z-Client can be PC, Web server or part of Library Management System
SDI type extended service requests
Define where and how records are exported
Records are exported via e-mail, fax etc.
Shared information about data
Attribute sets
-Field names
-Boolean operators
-Relations etc
Record Syntax
- USMarc
- Unimarc
- MAB
- Picamarc
SUTRS
GRS-1
OPAC
etc.
Database schema
Common understanding of record structures - allows authors, titles etc to be extracted and transferred

Any questions?

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Open Archives Initiative (1)

- Metadata harvesting of open archives
- OAI-PMH 2.0 (Protocol for Metadata Harvesting)
- Keep it simple
  - REST-style interface
  - Response in XML
  - DC metadata (unqualified)
- Widely accepted
- Generic approach

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Open Archives Initiative (2)

- Problems
  - Unqualified DC demands normalization
  - Missing mechanisms required for distributed services
  - No specs for use cases beyond bibliographic metadata harvesting (generic service model)

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

XML-RPC Basics

- Remote procedure calls HTTP, encoded in XML
  - HTTP-request and its response
- Supports most common types and aggregates

User driven develop-ment

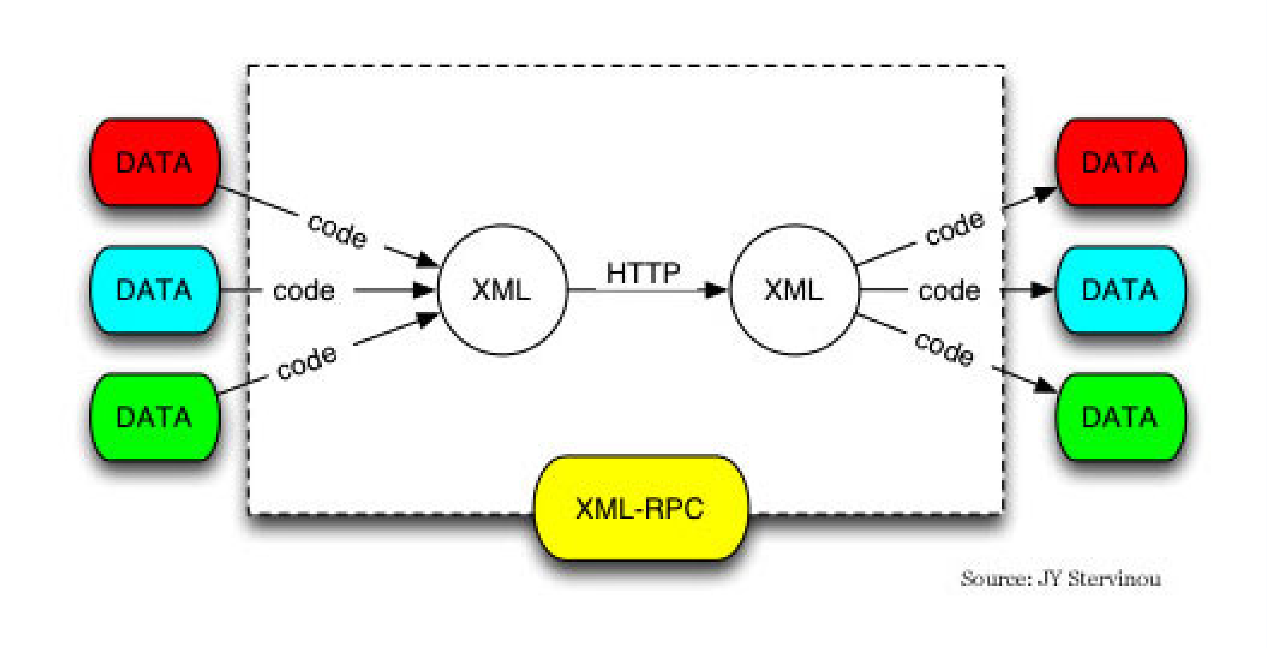
Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

XML-RPC



Source: JY Stervinou

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

XML-RPC - Request

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT) Host
betty.userland.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
    <methodName>examples.getStateName</methodName>
    <params>
        <param>
            <value><i4>41</i4></value>
        </param>
    </params>
</methodCall>
```

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

XML-RPC - Response

```
HTTP/1.1 200 OK
 Connection: close
 Content-Length: 158
 Content-Type: text/xml
 Date: Fri, 17 Jul 1998 19:55:08 GMT
 Server: UserLand Frontier/5.1.2-WinNT

<?xml version="1.0"?>
<methodResponse>
     <params>
              <param>
                            <value><string>South
     Dakota</string></value>
              </param>
     </params>
</methodResponse>
```

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

XML-RPC Perspectives

- Libraries for many programming languages available
- Fairly simple and fast
- Implicit knowledge about interfaces required

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SOAP - Basics

- „Simple Object Access Protocol“ (W3C standard)
- Communication between applications via messages
- Platform and language independent
- Based on XML

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SOAP-Messages

- SOAP Envelope
- Header with meta information e.g. on
  - Routing
  - Security
  - Transactions
- Body contains the payload, compliant with XML Schema

User driven develop-ment

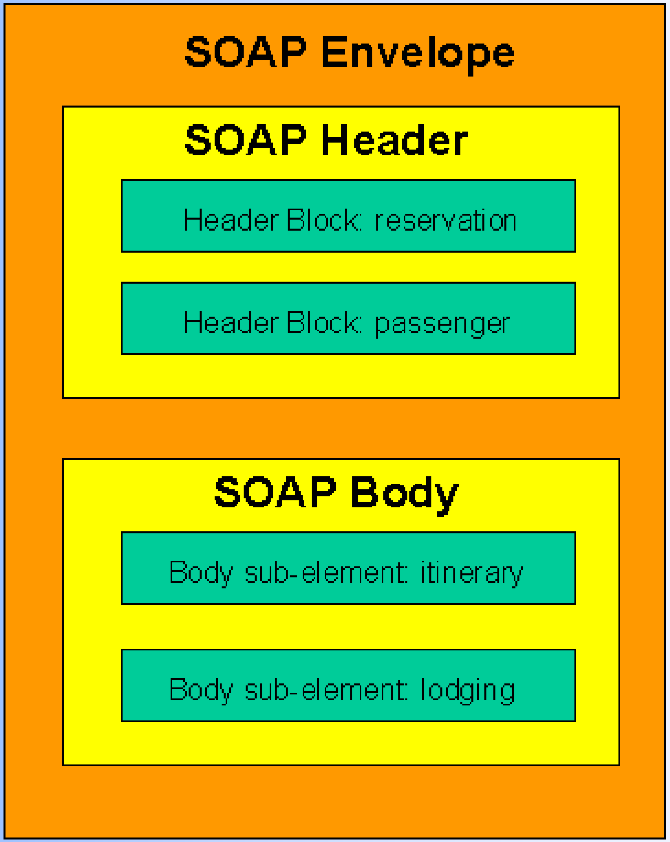
Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SOAP-Messages



SOAP Envelope

SOAP Header

Header Block: reservation

Header Block: passenger

SOAP Body

Body sub-element: itinerary

Body sub-element: lodging

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SOAP Perspectives

- Supported by the vendors
- Cornerstone of „Web Services“
- Human-readable?
- Depends on XML Schema type system
- Overhead
- Vendor specific tools
- Vendor specific type systems

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

SRW / ZING

- “Search/Retrieve Web Service” of the ZING Initiative, hosted by the Library of Congress
- ZING = Next Generation of Z39.50
- SOAP based implementation
- Leverages CQL query language
- Supports the librarians needs

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

WebDAV - Background

- Web Distributed Authoring and Versioning Access Protocol
- Microsoft, Netscape, Xerox, IBM, Novel…
- Extensions to HTTP protocol
- XML based
- Generic

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

WebDAV - Services

- Overwrite prevention
- Properties
- Namespace management
- Version management
- Advanced collections
- Access control

WebDAV - Perspectives

- Supported by
  - Office software packages
  - Content management Systems
  - Webservers
  - Databases
  - Browsing facilities
  - …

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Layered Technologies and the Semantic Web

- Example: OSI
- Transparency of layers
- Semantic Web
- Transparency in Scientific Workflow and Networks

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

OSI - Background

- „Open Systems Inter-connection Reference Model“ (OSI Model or OSI Reference Model)
- Framework for standards of the ISO
- Specifies seven layers for communications and computer network protocols

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

OSI - Layers

- Physical layer
- Data link layer
- Network layer
- Transport layer
- Session layer
- Presentation layer
- Application layer

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

OSI – Transparent layers

- Independence of layers allows replacing implementations
- Extensions don‘t interfere with other layers
- Users or developers are concerned only with „their“ layers

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

OSI – In Real Life

- Vendors reduced the number of layers
- Early implementations very instable
- Real solutions (like TCP/IP) established de facto standards

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Semantic Web – Vision I

- Global database
- Democratic access to information (vs. information divide)
- Web of knowledge

User driven develop-ment

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

User‘s Perspective on Semantic Web

- Missing support for semantics
- Real use is behind possibilities
- Problem of standardization
  - Standards as a part of the problem
  - Domain specific solutions required
  - Technology driven development

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Semantic Web today

- Generic specification (plus vendor-specific features)
- B2B – supported by detailed frameworks
- Weak support for scientific networks

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Semantic Web – Vision II

- Coupling of distributed information systems
- Support of information flow
- Foundation for (new) services
- Machine-machine communication
- Support for machine reasoning

User driven development

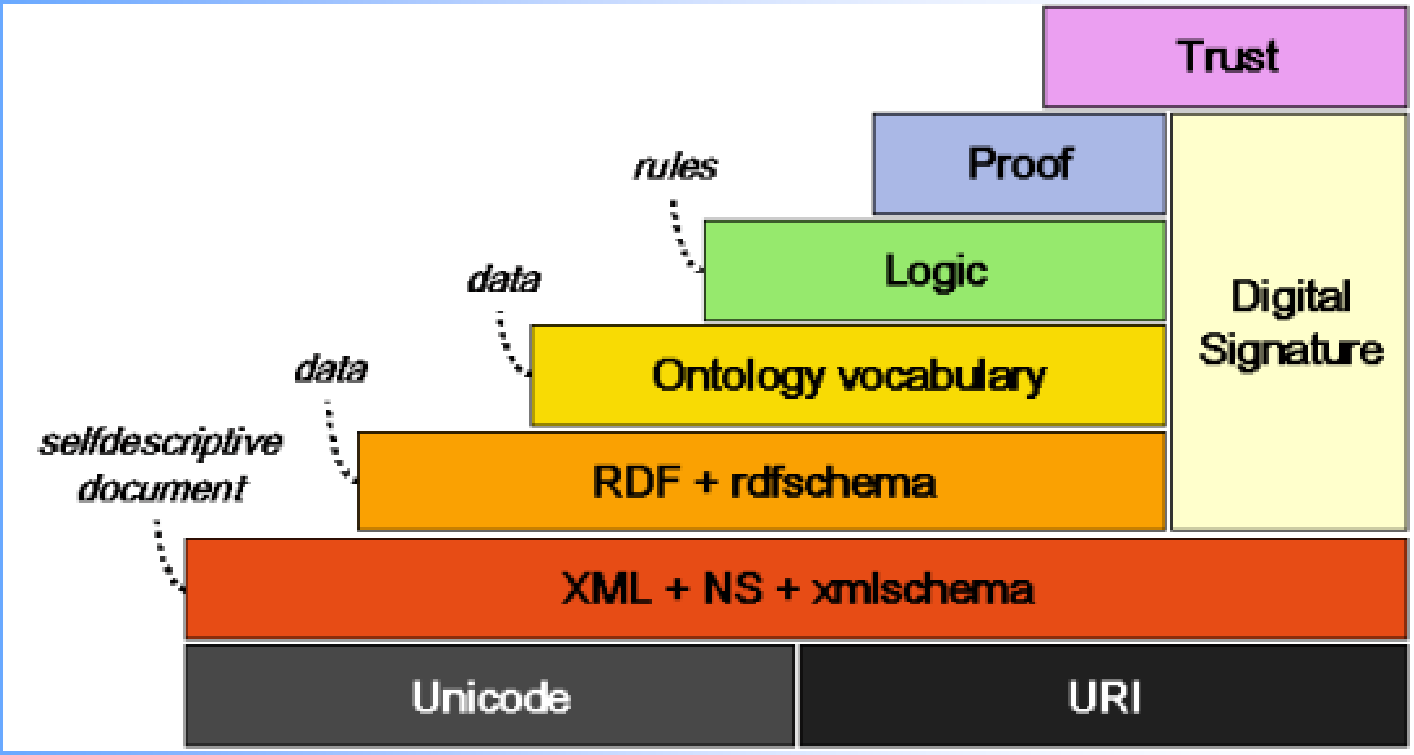
Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Layers of the Semantic Web



User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Layers of Web Services Standard (W3C)

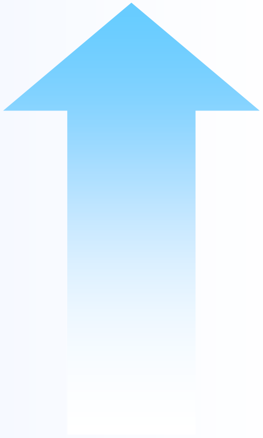
Service Discovery Layer

Service Description Layer

XML Messaging Layer

Service Transport Layer

Informal / high level of semantics



formal / low level of semantics

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Problems of “semantic” layers

- Implicit assumptions and assertions of layered technology
  - basic vs. higher functions
  - completeness
  - independent layers
- Independence of semantic layers is not given
- Conceptualization is domain specific

User driven development

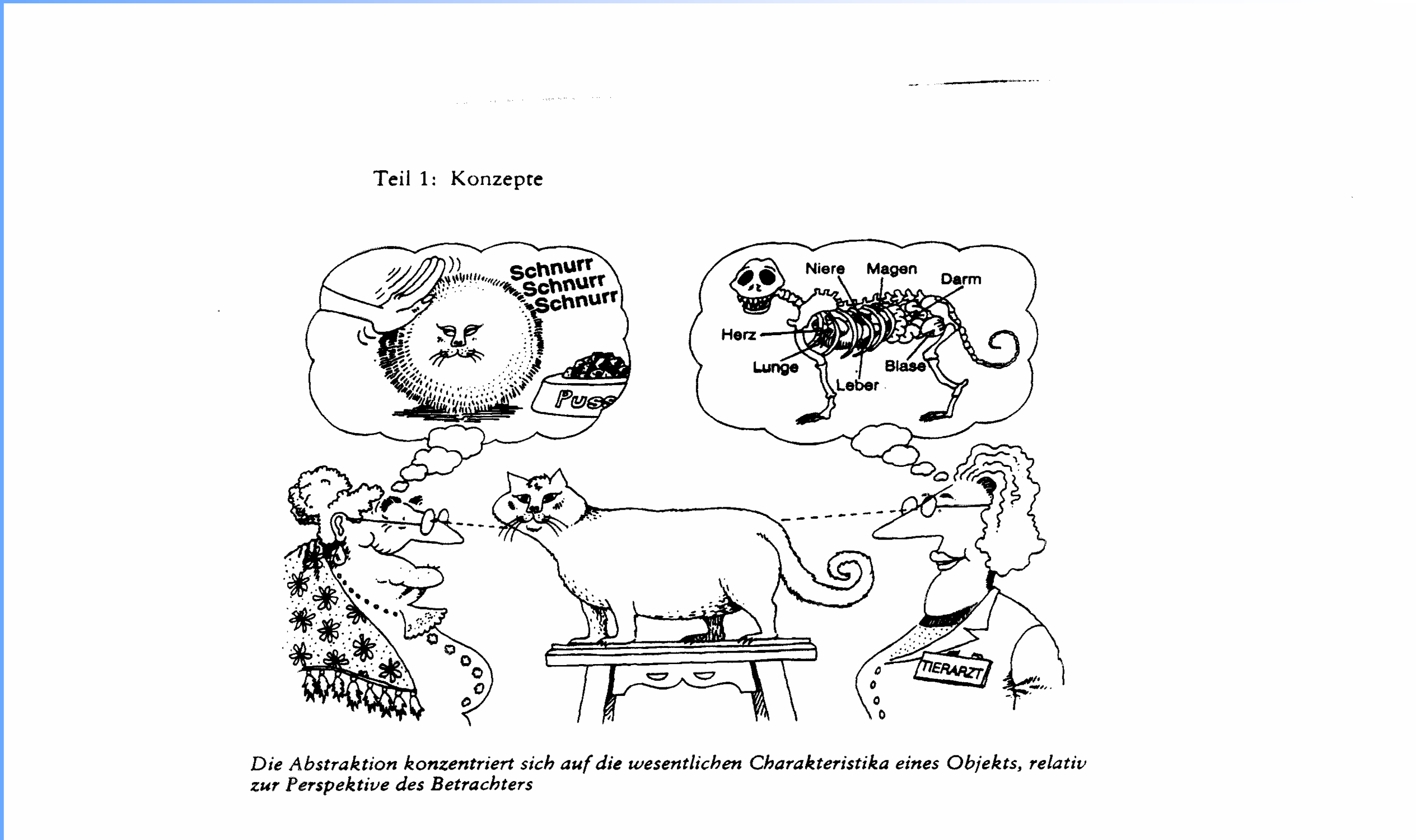
Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Conceptualization and Common Sense



Teil 1: Konzepte

Die Abstraktion konzentriert sich auf die wesentlichen Charakteristika eines Objekts, relativ zur Perspektive des Betrachters

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Requirements for Scientific DL-Services

- Support of scientific workflow
- Support of new emerging roles or change of roles
- Information on information needed
- Support of distributed generation of content

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Technology and Solutions

- „We do what we have“
- „One size fits all“ – the generic way
- Semantic GRID: we have the technology – now we invent the problems to solve

User driven development

Design principles

Protocols and their usage

Layered technology and the Semantic Web

Conclusion

Conclusion

- Best „return on investment“: metadata
- Information on Information
- Specification on usage of standards
- Transparency in science and education
- People behind solutions