



NFDI for and with Computer Science



**Prof. Dr. Agnes Koschmider
(Universität Bayreuth)**

NFDIxCS Mission Statement

- Discussion and standardization of CS research data formats, metadata formats, and semantics
- Implementation of the related research data management infrastructure, tools and services:
 - to promote the implementation of the FAIR Data Principles in the CS community for research data as well as software artifacts,
 - to simplify the citability of software and CS data
 - to modernize the publication processes and culture in both CS and its applications
- Support all subdisciplines of the CS community in handling of their research data
- Invoke data sets from other disciplines to further develop genuine CS (research) methods
- Share experience and knowledge of CS (system architectures, processes, standards for interoperability, data-oriented scientific publishing, communication systems, etc.) with others



Gesellschaft für Informatik | www.gi.de

Digitales Kulturerbe

Die Digitalisierung hat unsere Kultur tief durchdrungen. Musik, Videos, Fotos werden inzwischen überwiegend digital hergestellt und verbreitet, Bücher werden digitalisiert, Kommunikation findet über Handys, E-Mails oder Chats statt. Manches wie z. B. Computerspiele hat nicht einmal mehr eine Entsprechung in der analogen Welt. Wenn digitale Informationen nun nahezu von überall her und für jeden zugänglich sind, birgt dies zwar Chancen, stellt uns aber auch vor technische Herausforderungen: Wie lässt sich unsere digitale Kultur dauerhaft bewahren? Wie können virtuelle Objekte angemessen (re)präsentiert und zugänglich gemacht werden?

Wir benötigen Konzepte, um Kulturgüter auch für künftige Generationen begreifbar und erlebbar zu machen. Und ohne nachhaltige Langzeitbewahrung ist unsere digitale Kultur unwiederbringlich verloren. Nur mit Strategien zur Langzeitarchivierung können wir unser digitales Kulturerbe erhalten und ein „Zeitalter ohne Gedächtnis“ vermeiden.

1
DIE
GRAND CHALLENGES
DER INFORMATIK

Timetable

- ▷ Incremental build up
- ▷ Pattern of regular assessment and tuning

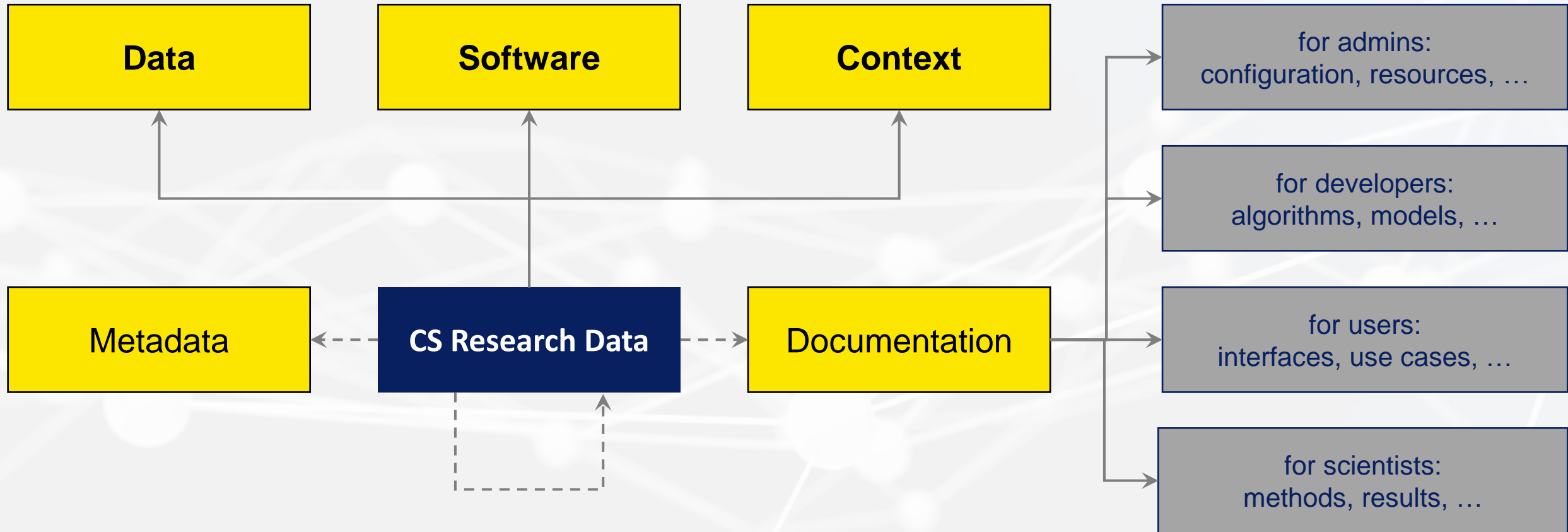


Consortium and Services

- balance of different interest groups:
 - research
(quick response to heterogenous and dynamic requirements)
 - operations
(stable, safe, efficient and maintainable infrastructure)
- disciplinary expertise + powerful tools
from different perspectives
- re-use, adaption and connection of existing services
as far as possible
- high degree of automation for sustainable operation



Research Data in Computer Science



Examples for Computer Science Research Data

```

fun
  fetch :: instr list => state => cell => instr
  where
    fetch p 0 b = (Nop, 0)
    | fetch p (Suc s) Bk =
      (case nth_of p (2 * s) of
        Some i => i
        | None => (Nop, 0))
    | fetch p (Suc s) Oc =
      (case nth_of p ((2 * s) + 1) of
        Some i => i
        | None => (Nop, 0))

lemma fetch_Nil [simp]:
  shows fetch [] s b = (Nop, 0)
  (proof)

```

```

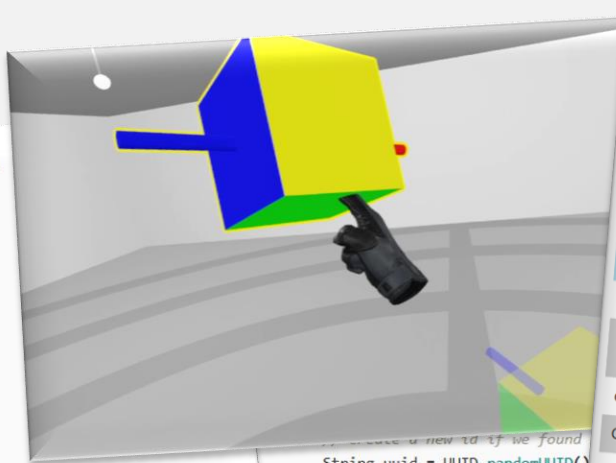
fun
  update :: action => tape => tape
  where
    update WO (l, r) = (l, Bk # (tl r))
    | update WI (l, r) = (l, Oc # (tl r))
    | update L (l, r) = (if l = [] then [], Bk # r) else (tl l, (hd l) # r)
    | update R (l, r) = (if r = [] then (Bk # l, []) else ((hd r) # l, tl r))
    | update Nop (l, r) = (l, r)

```

```

abbreviation
  read r == if (r = []) then Bk else hd r

```



System	OC	FUY	Screenshots	TG							
			P	W	C	H	I	M	S	U	X
checkpoint [0] [1] [2]	GBR	2005	•								
cloudcoder [0]	USA	2013	•	•							
COALA [0]	ESP	2009	•								
code [0] [1]	MKD	2012	•								
code hunt [0] [1] [2]	USA	2014	•								
codeinsghts [0]	PRT	2018	•								
CodeLab [0]	USA	2016	•								
codeOcean [0] [1]	DEU	2016	•								
CodeQ [0]	SVN	2018	•								
CodeRunner [0] [1]	NZL	2016	•								

```

String uuid = UUID.randomUUID();
String requestCommand = "INSERT";

if (!Strings.isNullOrEmpty(fullSubmissionPostRequest) &&
    !Strings.isNullOrEmpty(fullSubmissionPostRequest.requestCommand)) {
    requestCommand = "REPLACE";
}

String request = String.join(" ", requestCommand.trim(),
    "fullsubmissions ('id', 'version', 'groupId', 'header',
    connection.connect();
// build and execute request
connection.issueInsertOrDeleteStatement(request, uuid, version,
    fullSubmissionPostRequest.getGroupId(),
    fullSubmissionPostRequest.getHeader(),
    fullSubmissionPostRequest.getText(),
    fullSubmissionPostRequest.getProjectName(),
    fullSubmissionPostRequest.getFileRole().toString(),
    fullSubmissionPostRequest.getUserEMail(),
    fullSubmissionPostRequest.getVisibility().toString());

```



Publication search results

found 12,169 matches

2021

- Michele Flammini, Manuel Mauro, Matteo Tonelli: **On fair price discrimination in multi-unit markets.** *Artif. Intell.* 290: 103388 (2021)
- Jiyeon Ham, Soohyun Lim, Kyeng-Hun Lee, Kee-Eung Kim: **Corrigendum to 'Extensions to Hybrid Code Networks for FAIR Dialog Data' Computer Speech & Language volume 53 (2019) Pages 80-91.** *Comput. Speech Lang.* 65: 100961 (2021)
- Zhaoxi Wu, Liqun Fu: **Optimizing job completion time with fairness in large-scale data centers.** *Future Gener. Comput. Syst.* 114: 563-573 (2021)
- Tiantian Li, Wei Ren, Yuexin Xiang, Xianghan Zheng, Tianqing Zhu, Kim-Kwang Raymond Choo, Gautam Srivastava: **FAPS: A fair, autonomous and privacy-preserving scheme for big data exchange based on oblivious transfer, Ether cheque and smart contracts.** *Inf. Sci.* 544: 469-484 (2021)
- Santosh Kumar Bhal, P. Danumjaya, Graeme Fairweather: **The Crank-Nicolson orthogonal spline collocation method for one-dimensional parabolic problems with interfaces.** *J. Comput. Appl. Math.* 383: 113119 (2021)

Data Category	Data Subcategory	Examples for technical specifications, protocols, formats, languages etc.
Data sets	strongly structured	SQL, RDF, SPARQL, GraphQL, JSON, XML
	semi-structured	CSV, TSV, NetCDF, HDF5, Apache Parquet
	unstructured	txt, mp3, mp4, jpg
Software	formal models	UML, BPMN, DMN, ArchiMate, ERM
	scripts	Python, R, MatLab, Visual Basic
	web/microservices	HTTP, REST, JSON, AMQP, MQTT
	complex systems	YAML
Context	operating systems	MS Windows, macOS, iOS, Android, various Unix versions and derivatives
	Programming models & runtimes	C, C++, Fortran, Java OpenMP, MPI, CUDA
	supporting services	CI, CD, container services, server virtualization
	hardware	Configuration files, specifications, emulators if available/necessary
	Physical location	GPS, Address, location in rack, room number

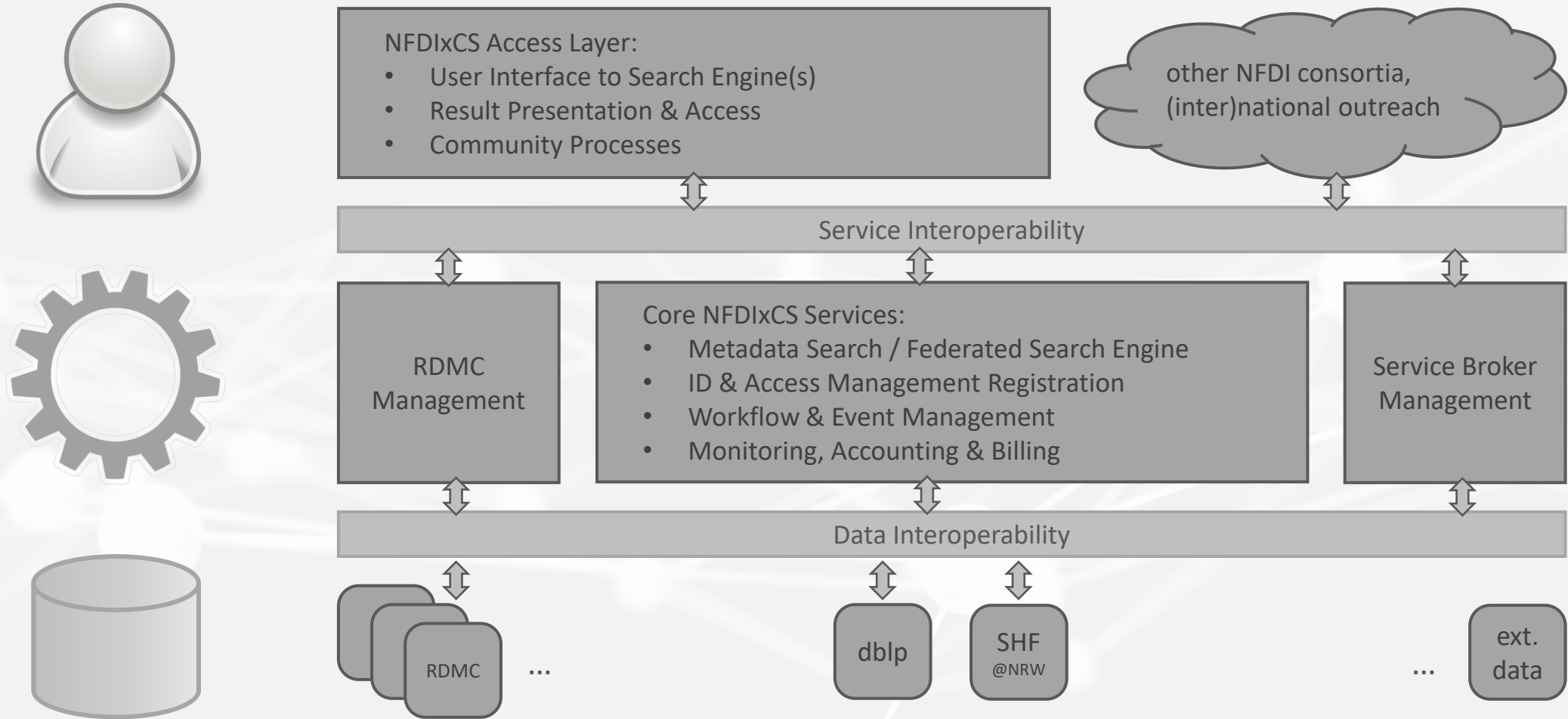
FAIR DO: Couple Data and Context

The persistent encapsulation of data and different types of metadata into such a knowledge unit should significantly **improve the level of trust** with which it can be understood across disciplines and modes of sharing. For such a transformation be achieved, it is essential to employ a set of robust conceptual and technological models that enable **sharing and reuse of data-in-context**.

Smedt, K. de; Koureas, D.; Wittenburg, P.: FAIR Digital Objects for Science: From Data Pieces to Actionable Knowledge Units. Publications 2/8, p. 21, 2020.

DOI: [10.3390/publications8020021](https://doi.org/10.3390/publications8020021)

NFDI_XCS Architektur



Stage 1: Prototype and implement Research Data Management Container

Data

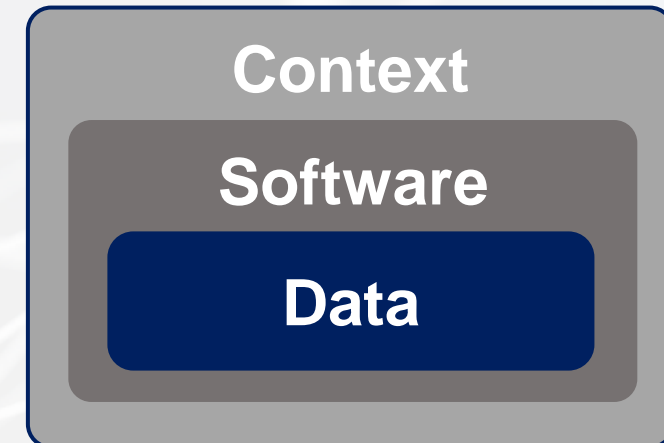
Unstructured, Semi-Structure, Strongly Structured

Software

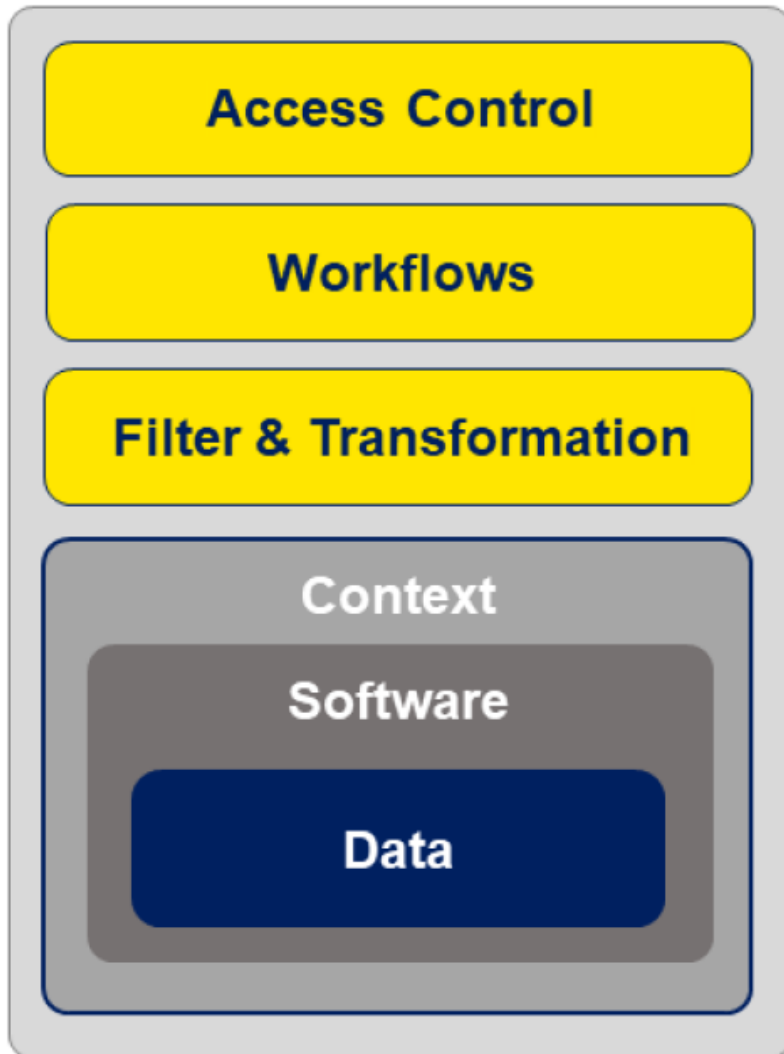
Formal Models, Scripts, Web/Microservices, ...

Context

Operating Systems, Programming models & runtime, hardware...

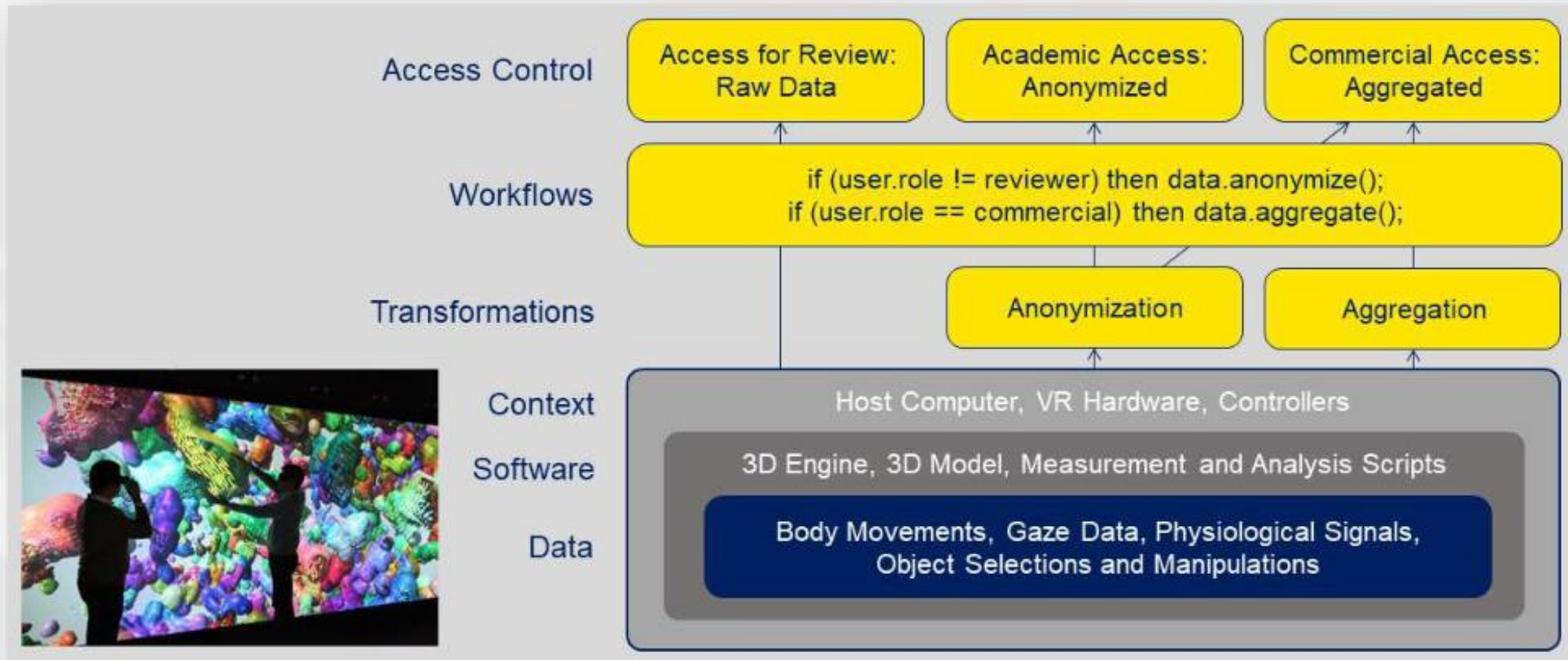


Research Data Management Container (RDMC)



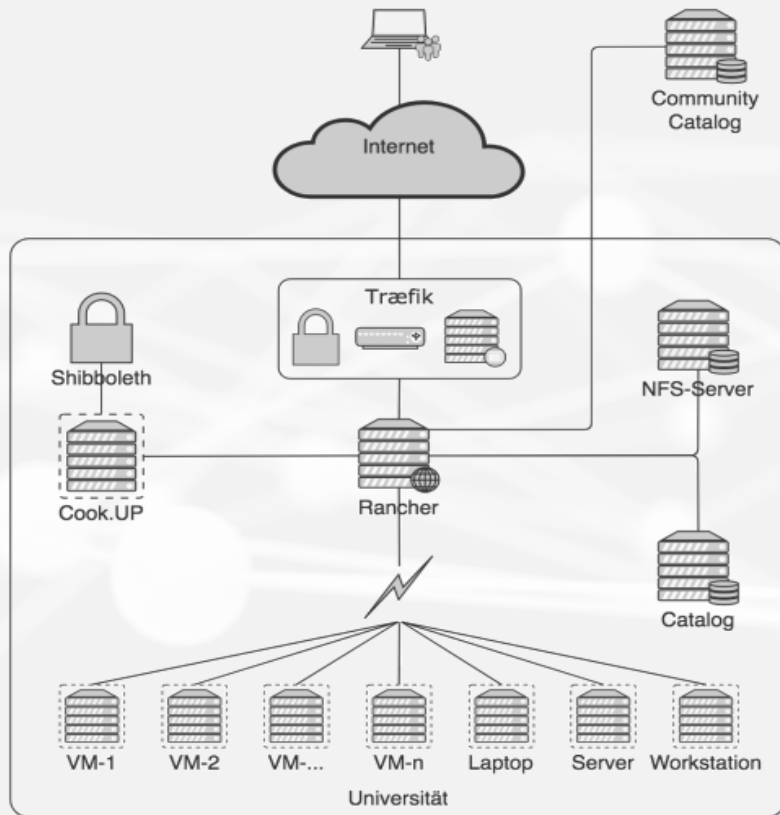
- Encapsulating research data in a container/packaged format
- portable object that manages itself
 - access
 - specific workflow and all the data
 - Software
- “bring the data back to life”
- RDMCs will be hosted by trusted service providers
- Examples: Docker, ..., code ocean4, Fair Data Objects6, ...

Example of an RDMC



Example: Docker / Rancher Infrastructure

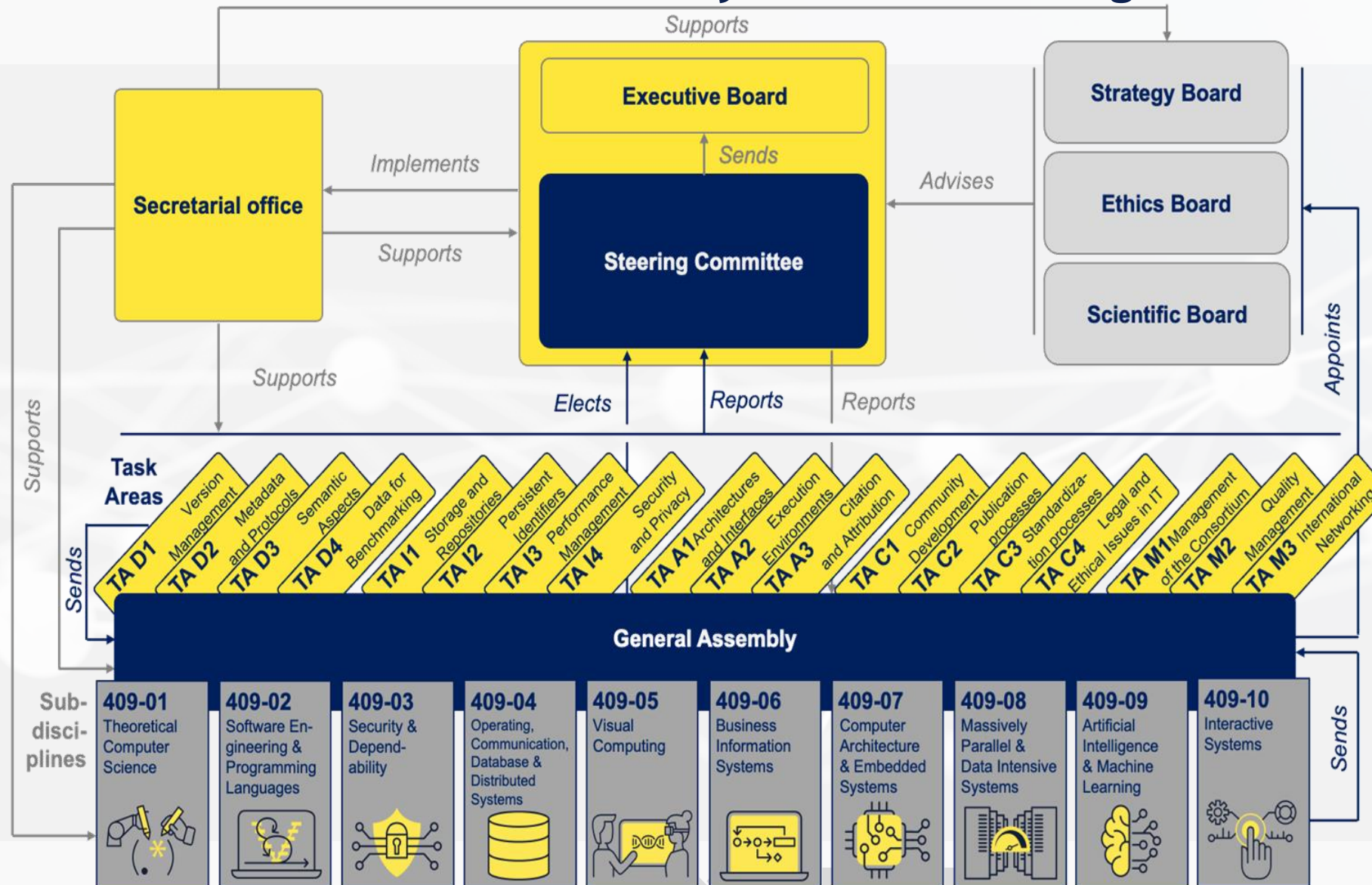
Treat your software like cattle, not pets!



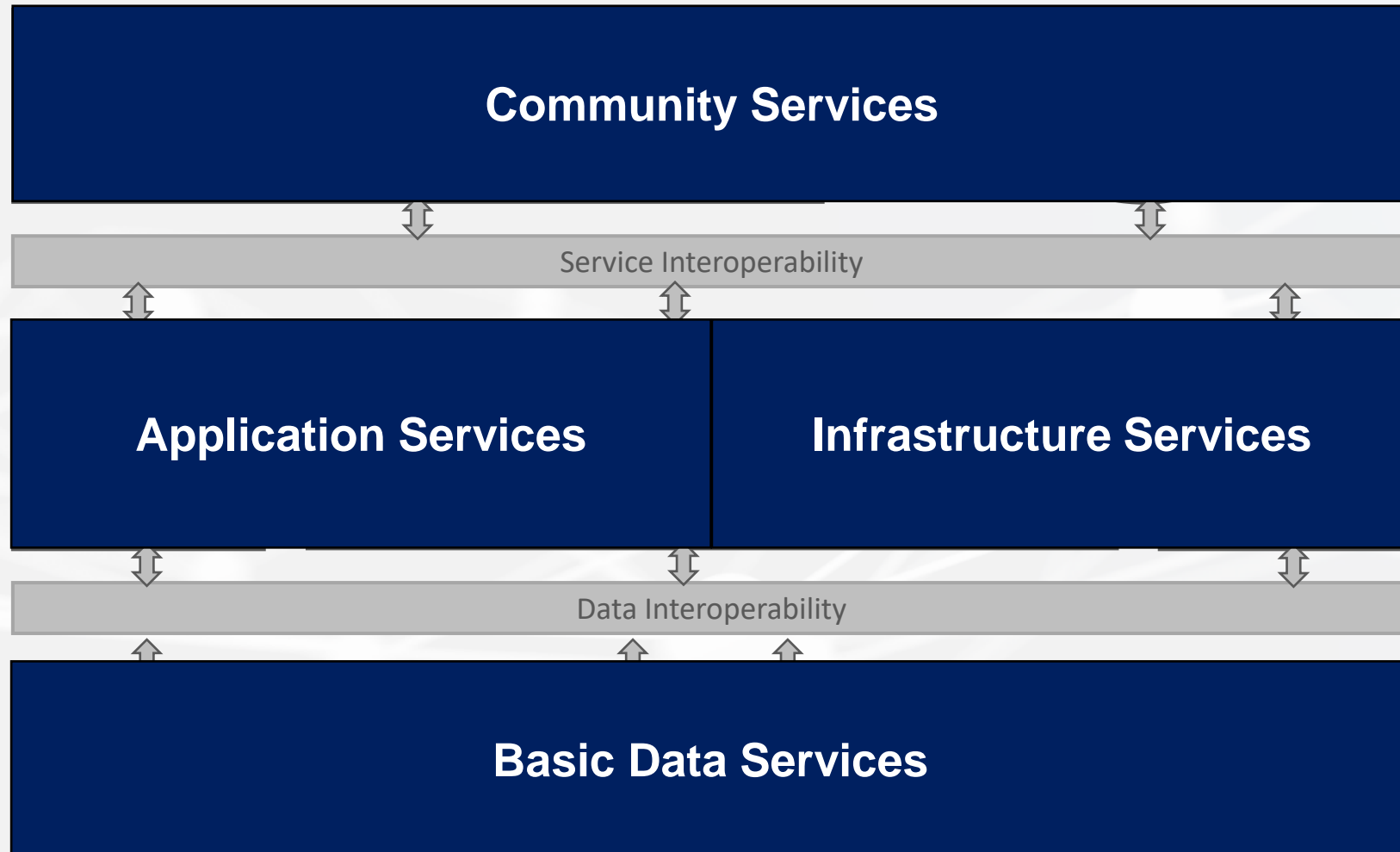
The screenshot shows the 'Cook.UP' web interface for the University of Bayreuth. The header includes the university logo, the name 'Cook.UP', and navigation links for 'Uni-Startseite', 'Uni A-Z', and 'Deutsch'. A navigation bar contains 'Rezepte' (selected), 'Dienste', 'Hallo Dr. Manfred von Mustermann!', and 'Logout'. The main content area is titled 'Dienstrezepte' and lists '10 Verfügbare Rezepte'. The recipes are displayed in a grid:

- DokuWiki**: DokuWiki ist eine einfach zu benutzende und höchst vielseitige Open Source Software, die vollkommen ohne Datenbank auskommt.
- Ghost**: Ghost ist eine einfache Blogplattform.
- Limesurvey**: Limesurvey ist eine Online-Umfrage-Applikation, die es ermöglicht, ohne Programmierkenntnisse Online-Umfragen zu entwickeln, zu veröffentlichen sowie deren Ergebnisse in einer Datenbank zu erfassen.
- Matomo**: Matomo, ehemals Piwik, ist eine Open-Source-Webanwendung für Webanalytik. Es ist eine Alternative zu Google Analytics.
- MediaWiki**: MediaWiki ist eine einfach zu benutzende und höchst vielseitige Open Source Software.
- phpbb**: phpbb ist eine Forumsoftware.
- Redmine**: Redmine ist eine freie, webbasierte Projektmanagement Anwendung.
- RocketChat**: RocketChat ist eine Alternative zum Slack online chat.
- Wekan**: Wekan ist eine open-source Trello Alternative.
- Wordpress**: Wordpress ist eine freie Blogging Plattform, mit der Sie eine schöne Website, einen Blog oder eine App erstellen können.






General Assembly and how all fits together



NFDIxCS Layers of Task Areas



Task Areas


	409-01	409-02	409-03	409-04	409-05	409-06	409-07	409-08	409-09	409-10
Management Layer		TAM1 Management of the Consortium		TAM2 Quality Management		TAM3 International Networking				
Community Layer		TAC1 Community Development		TAC2 Publication Processes		TAC3 Standardization Processes		TAC4 Legal and Ethical Issues in IT		
NFDI Application Layer		TAA1 Architectures and Interfaces		TAA2 Reusable Exec. Environment		TAA3 Citation and Attribution				
Infrastructure Layer		TAI1 Storage and Repositories		TAI2 Persistent Identifiers		TAI3 Performance Management		TAI4 Security and Privacy		
Basic Data Layer		TAD1 Version Management		TAD2 Metadata and Protocols		TAD3 Semantic Aspects		TAD4 Data for Benchmarking		

Layer

Task Area


Sub-Discipline

For each Subject...

<p>409-01 Theoretical Computer Science</p>	
<p>Characterization of the field</p>	
<p>Theoretical computer science (TCS) primarily focuses on mathematical aspects of CS, historically emerging from a specialized sub-field of mathematics addressing basic notions of computability, algorithms and their computational complexity. Modern TCS is a well-established field of research in its own right with a large and broad spectrum of topics, including (but not limited to) data structures, computational geometry, cryptography, automata theory, formal methods, parallel and distributed computing and computational logic. Research in TCS is characterized by mathematical rigor, applications of formal mathematical proof techniques, and, usually, its foundational character. In particular, the sub-discipline of automated reasoning and theorem proving can be described in a broader sense as applied mathematical logic. It deals with the theory, implementation and application of software systems for automatic or interactive proofs of formal logical statements from mathematics and CS. The central goal is the automation of formal mathematical reasoning on the one hand and, on the other hand, the practical support of its application to concrete problems from mathematics and CS. The applications hereby range from basic theoretical results, e.g., the proof of Kepler's conjecture in mathematics, to applied academic and industrial applications, e.g., the verification of software and cyber-physical systems.</p>	

<p>Examples for research data</p>	
<p>Formal proofs for interactive theorem provers. Each repository is associated with a specific theorem prover {A2}. Repositories are directories of text files {I1}. Current repository sizes are of the order of 1GB. Three widely used repositories {M3} are listed below.</p>	<p>Proofs are generated interactively by researchers with the help of a theorem prover. Proofs are checked by the theorem prover, archived {C2} and referenced {I2, A3} from other proofs and from publications. They can be updated continuously {D1}. Proofs are written in a formal language (specific to the theorem prover) {D2, A1}; the formal language may change from version to version {D1} and the libraries may change {A2}. The repositories are versioned {D1}, online and open access {C2, C4}.</p>
<p>Data for testing, evaluation and training of theorem provers. Proof problems in standardized syntax formats {C3} (e.g. for classical first- and higher-order logic) are used to test and evaluate automated theorem provers {D4, M2}. A prominent library is the TPTP library of proof problems; version v7.5.0 expands into 6.9GB (problems, axiom sets, documents, and utilities). Such problem repositories {I1} are increasingly</p>	<p>Proof problems are submitted by the community. They are assessed according to criteria such as novelty and difficulty (how many provers can solve them) {M2}. Only a portion of submissions is selected for inclusion in the TPTP library {C2}, which is the basis for benchmarking {D4} and yearly ATP competitions {M3}. The problems are analysed and formatted with additional tools from the TPTP infrastructure {A2}. TPTP is maintained and stored at Univ. of Miami, but a long-term solution is currently sought (one current option includes a move to Germany) {A1}. TPTP is online and open access {C2, C4}.</p>

For each Subject...

<p>409-05 Visual Computing</p>	
<ul style="list-style-type: none"> • Albrecht Schmidt • Enrico Rukzio • Bernt Schiele 	
<p>Characterization of the field</p>	
<p>Visual computing is the key discipline to enable effective and efficient interaction between people or with their environment. The research also considers the impact of system and visual design as well as architectural decisions in the development of information and communication technology on the experience of people's use of technology. The field includes computer graphics, computer vision, interactive visualization, and human-computer interaction. It is concerned with questions of how information is presented for human users and also how computing systems can acquire knowledge about their context and the situation around them. Novel interaction and interface technologies are researched from a technical perspective (object recognition, 3D registration, efficient rendering) as well as from a user perspective (acceptability, efficiency of interaction, user experience). New knowledge and scientific progress are generated from research into algorithms, from experimental system research, and from empirical research. Systematic studies are used to validate and evaluate visual computing solutions. Such studies can be formative or summative and the approaches can be qualitative or quantitative. The results are new algorithms, methods, tools, and approaches that are experimentally and empirically validated, the development of new interaction techniques and technologies in software and hardware, and theories and models for visual computing.</p>	

Examples for research data	
<p>Datasets for image recognition or activity recognition. This includes, besides the multimodal media (e.g. images, sensor data, videos, audio recordings) itself, descriptions of the data set {D2} and the experimental context {A2} as well as labels for the media data {D3} (e.g. time codes).</p>	<p>In visual computing systems the recognition of the environment is one key challenge, like the classification of images (such as indoor vs. outdoor) or the description of what is shown in a video. Such datasets may also include further modalities {D3} such as sensor value (e.g. gyro, acceleration) or audio recordings. Typically, a research group creates a dataset and initial algorithm, and publishes it {C2} along with the recognition results {D4}. Based on this, other research groups publish superior algorithms {A3} using the same dataset {D4}.</p>
<p>Documentation of research systems and their use. This includes software (e.g. programs in different programming languages and interface descriptions) {D2}, hardware descriptions (e.g. boards, components, architectures) {A2}, descriptions of physical forms (e.g. CAD files, models for 3D printing) and multimedia data (e.g. videos) that document the operation and usage. These data can depend on or relate to each other {A2, D3}.</p>	<p>Experimental systems are developed and functional prototypes are implemented. Besides algorithms, such implementations are often complex {A2} including specific hardware (e.g. power wall display, tangible interaction controller, GPU computing setup), newly developed or adapted system software (or firmware), as well as test applications {I3}. Such prototypes are set up and operated for the specific research project or for a study and they are not permanent {C2}. Hence it is required to document and reproduce these setups {M2}. There are no standards yet {C3}, often researchers conserve this information as an add-on to publication {C2} in a ZIP-archive.</p>

Mehr Informationen

NFDixCS.org

Prof. Dr. Michael Goedicke (Universität Duisburg-Essen)
Sprecher
michael.goedicke@paluno.uni-due.de

Prof. Dr. Ulrike Lucke (Universität Potsdam)
stellvertretende Sprecherin
ulrike.lucke@uni-potsdam.de

Vortrag von

Prof. Dr. Agnes Koschmider (Universität Bayreuth)
agnes.koschmider@uni-bayreuth.de

