OpenDreamKit Work Package 6
The Knowledge-First Strategy for System Integration

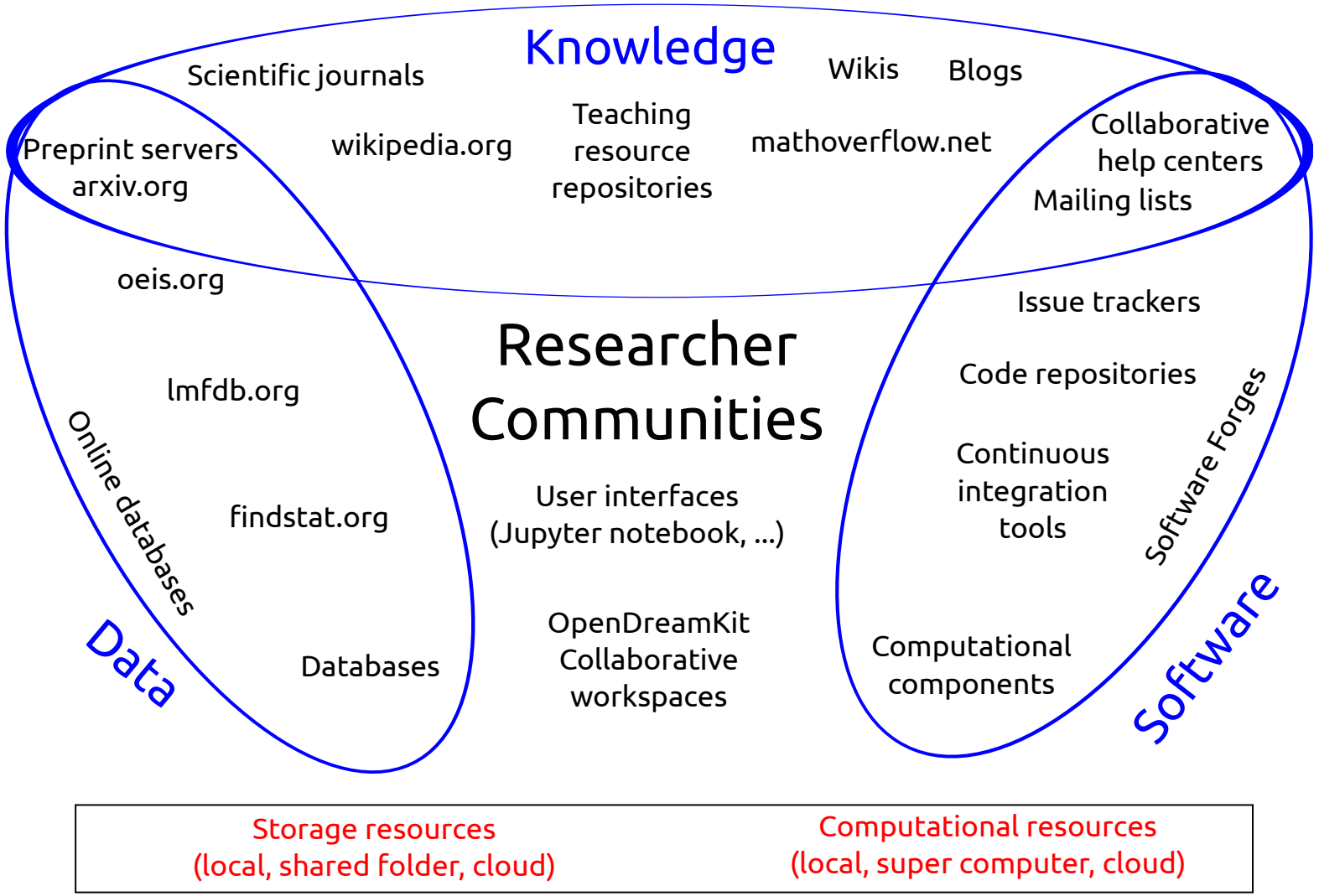
Michael Kohlhase

http://kwarc.info/kohlhase
Computer Science
Jacobs University Bremen, Germany

OpenDreamKit WP6 Workshop, St. Andrews, 25. January 2016

1 Work Package 6: Data/Knowledge/Software-Bases

OpenDreamKit: The Big Picture (from the Proposal)



- A math VRE where systems share Data ($\mathcal{D}$), Knowledge ($\mathcal{K}$), and Software ($\mathcal{S}$).

WP6 Objectives

- A VRE needs an infrastructure that supports the creation, management, access, and dissemination of $\mathcal{DKS}$-Structures. ($\mathcal{D}\hat{=}$ Data/$\mathcal{K}\hat{=}$ Knowledge/$\mathcal{S}\hat{=}$ Software)
- Observation: All ODK systems (GAP, SAGE, PARI, SINGULAR, LMFDB, OEIS, arXiv.org, . . .) already include data, knowledge, and software modules
- Limitation: low system Interoperability (Not a VRE yet)
- Root Cause: systems share the math, but represent $\mathcal{DKS}$ differently.
- WP6 Objectives:

  1) design metadata and representation formats for trans-system $\mathcal{DKS}$ structures as a basis for a math VRE,
  2) implement interfaces to existing systems for interoperability and compatibility with the RE, and
  3) implement a joint $\mathcal{DKS}$ infrastructure for, searches, documentation, traceability, versioning, provenance, visualisation and native dissemination of OpenDreamKit results (the latter three together with WP4).

WP6: Approach, Coverage

- WP6 Goal: Build a $\mathcal{DKS}$ repn. format, implement as a joint $\mathcal{DKS}$-base
- WP6 Approach: Build on a modular, foundation-independent, web-scalable $\mathcal{DKS}$-format/base $\rightsquigarrow$ OKDML/ODKBase
  1) for $\mathcal{K}$ use OMDoc/MMT as a basis (established interoperability format/base)
  2) for $\mathcal{S}$ extend it by computational foundations (prototype for Scala exists)
  3) for $\mathcal{D}$ develop scalable $\mathcal{KS}$-compatible data adaptors. (theory: $\mathcal{K} \supset \mathcal{D} \wedge \mathcal{S} \supset \mathcal{D}$)

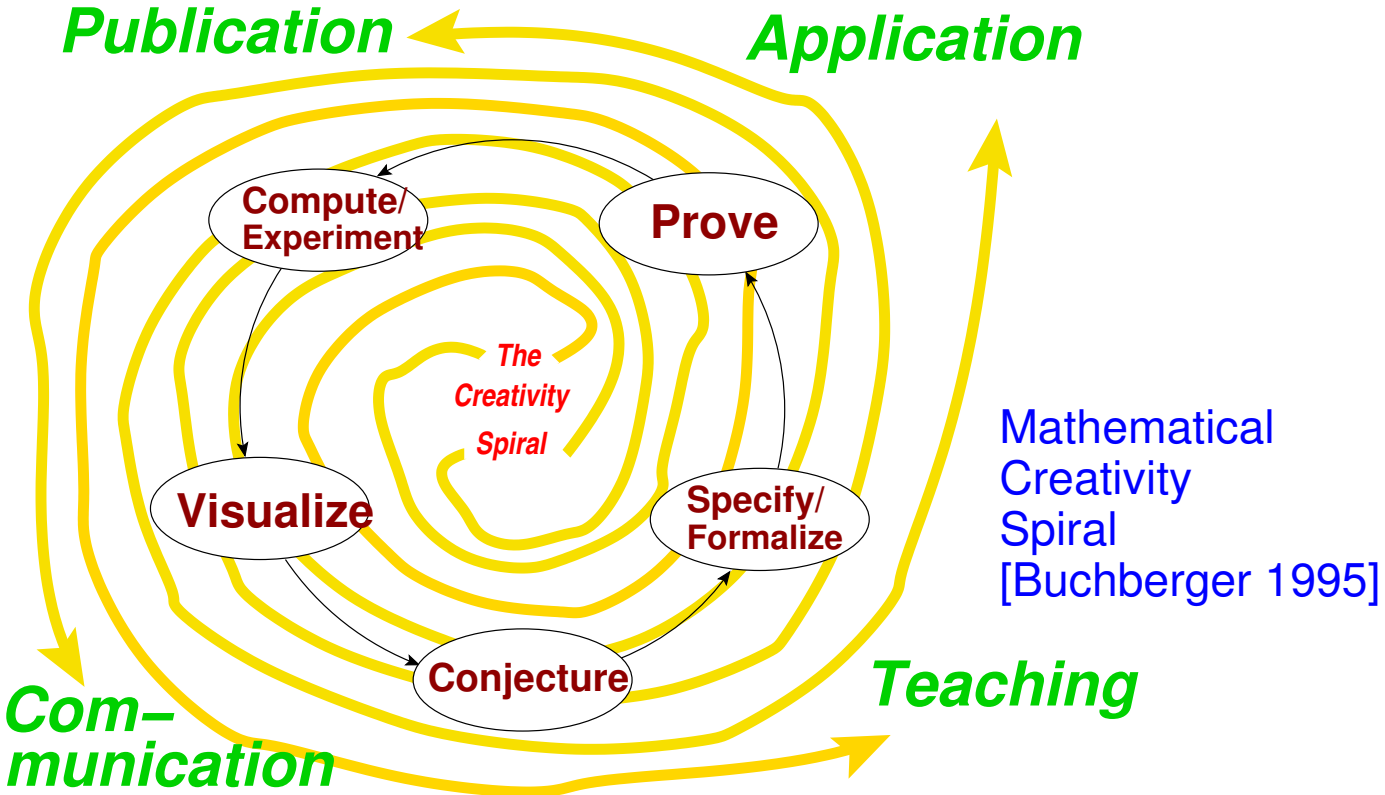
  Based on this make OpenDreamKit system/databases interoperable
  1) export existing databases into ODKML,
  2) specify ODK system foundations in ODKML
  3) build OKDML import/export facilities for ODK systems
  4) connect all up via ODKBase (acting as a $\mathcal{DKS}$ server and semantic context)
- Coverage: Start small/deep, extend, iterate (Mexican hat profile)

WP6 Participants/Efforts

- Sites involved in WP6: Data/Knowledge/Software-Bases
  1) JacobsUni (46 PM; lead) Survey, ODKML design, ODKBase implementation, OEIS, LMFDB, FindStat, Python/Sage Foundations, Search/query
  2) UPSud (37 PM), ODKbase design, CAS Integration, Python/Sage Foundations
  3) USTAN (10 PM), Survey, ODKbase design, Python/Sage Foundations, CAS Integration
  4) UWarwick (25 PM) LMFDB, ODKbase design, CAS Integration
  5) UZH (12 PM) Survey, ODKML design, LMFDB, FindStat, Python/Sage Foundations
  6) Logilab (2PM) ODKbase design
  7) USlaski (??? PM) CAS Integration, ODKBase design
- Total Effort: 132 PM ($\hat{=}$ 11 person years)

The way we do math will change dramatically

- **Definition 0.1 (Doing Math)** Buchberger's Math creativity spiral



- Every step will be supported by mathematical software systems
- Towards an infrastructure for web-based mathematics!

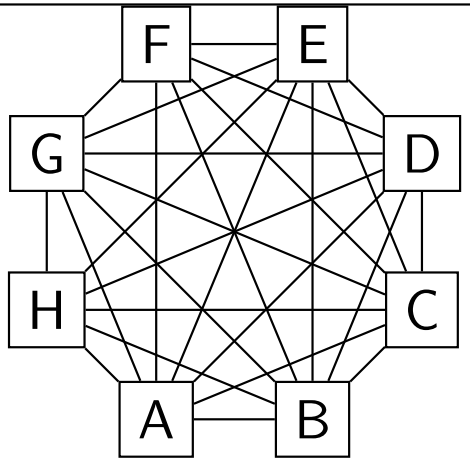
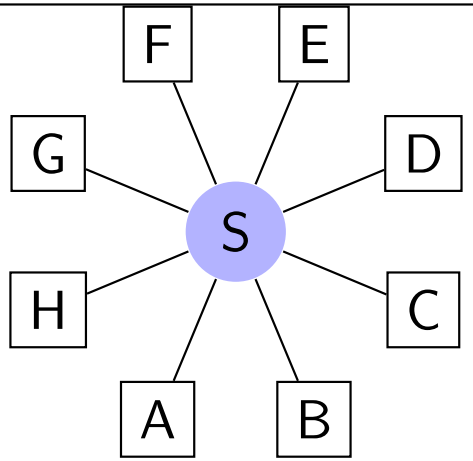
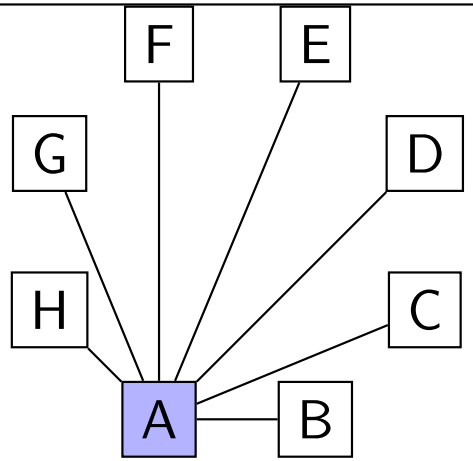
2 Towards a Math VRE
— Interoperability via a Joint Meaning Space —

Main Problem to solve for a VRE: Interoperability

- ODK Approach: build a VRE by connecting existing systems.(and improve them)
- Advantages: well-known Open Source Software
  1) Let the specialists do that they do best and like (and avoid what the don't)
  2) collaboration exponentiates results
  3) competition fosters innovation (+ no vendor lock-in)
- Problem: does an elliptic curve mean the same in GAP, SAGE, LMFDB?
  - otherwise delegating computation becomes unsound
  - storing data in a central KB becomes unsafe
  - the user cannot interpret the results in an UI
- Idea: Need a common meaning space for safe distributed computation in a VRE!

Obtaining a Common Meaning Space for our VRE

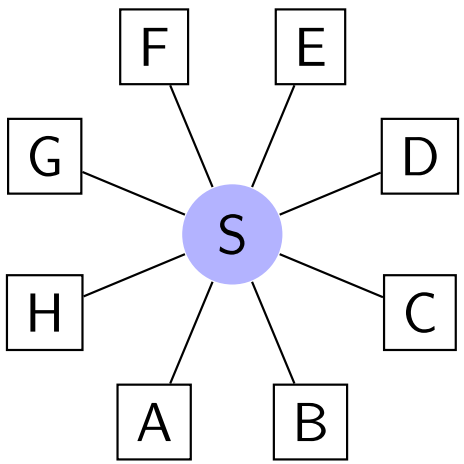
- Three approaches for safe distributed computation/storage/UIs

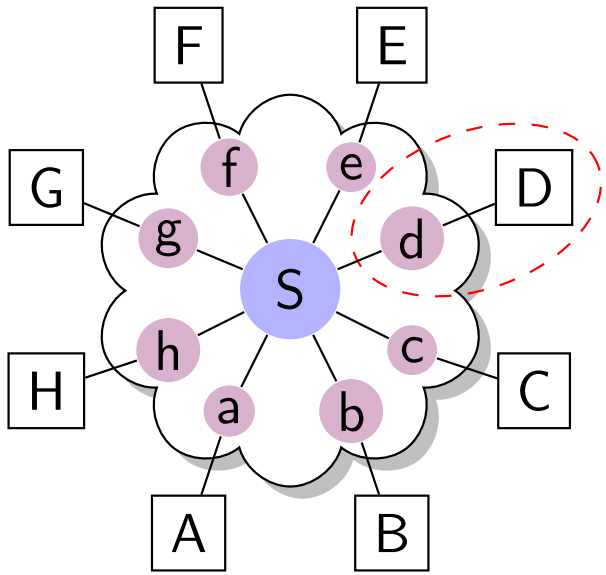
peer to peer	open standard	industry standard
		
$n^2/2$ translations symmetric	$2n$ translations symmetric	$2n-2$ translations asymmetric

- Observation: We already have a "standard" for expressing the meaning of concepts/objects/models: mathematical vernacular! (e.g. in math. documents)
- Problem: mathematical vernacular is too
  - ambiguous: need a human to understand structure, words, and symbols
  - redundant: every paper introduces slightly different notions.
- Knowledge First Paradigm: encode math knowledge in modular flexiformal format as a frame of reference for joint meaning (OMDoc/MMT)

Standardization with Interfaces

- Problem: We are talking about knowledge-based systems (large investment)
- Problem: Knowledge is part of both the
  - System $\rightsquigarrow$ system-specific representation requirements and release cycle
  - Interoperability Standard $\rightsquigarrow$ stability and generality requirements.
- Idea: Open standard knowledge base with interface theories



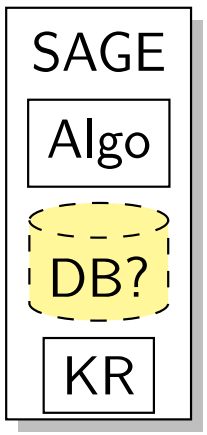


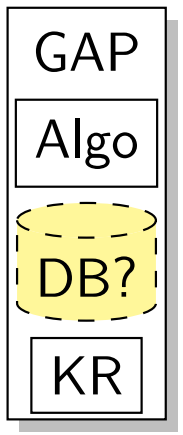
- **Definition 0.2** Interface theories are
  - system-near (import/export facilities maintained with system)
  - declarative, in standard format (refine general theories, relation documented)

3 Towards Realizing the Knowledge-First Paradigm in OpenDreamKit

The Knowledge-First Paradigm for ODK: Systems

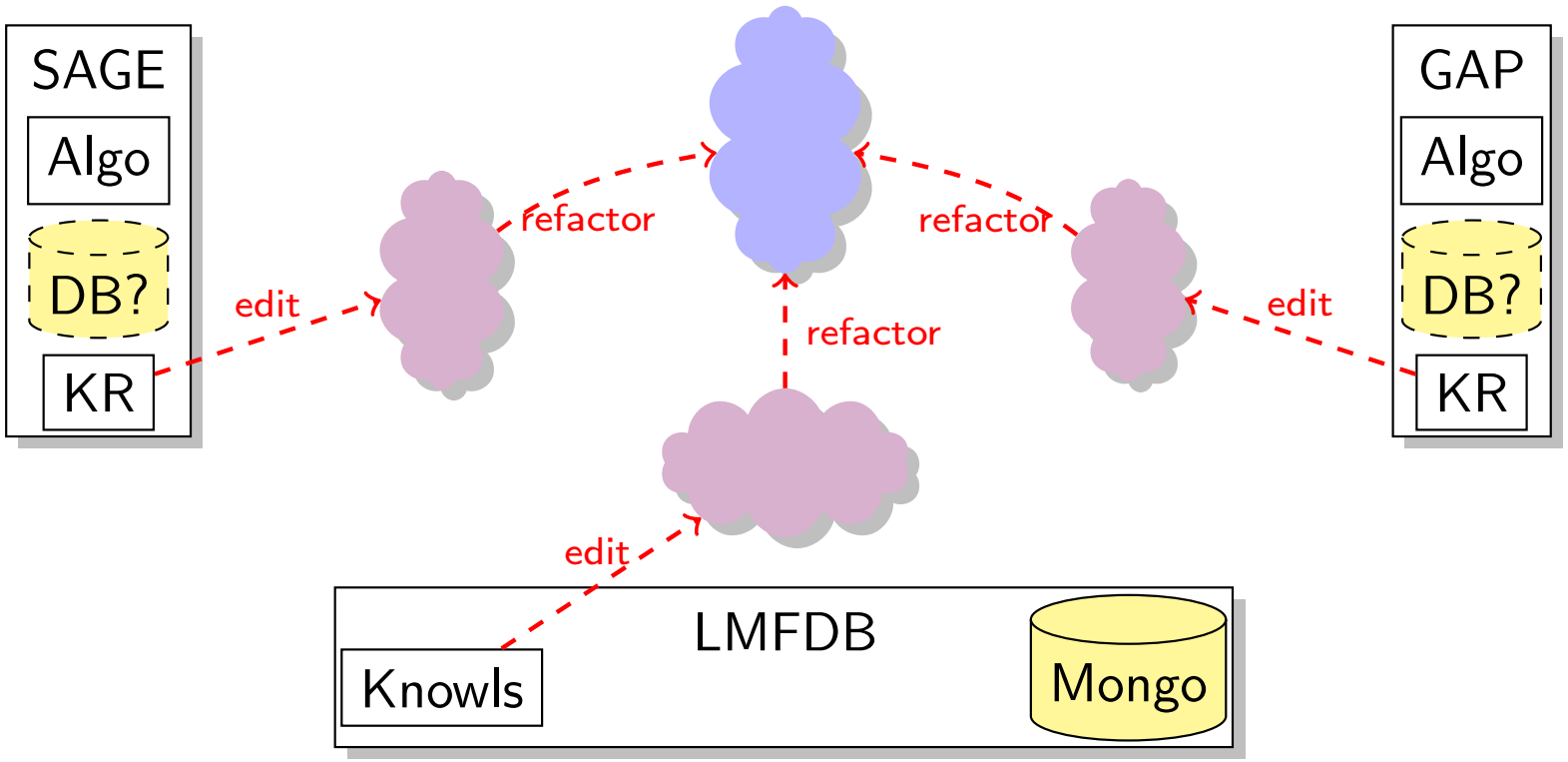
- Current State: Knowledge based systems with knowledge, algorithms, and data represented in program code.





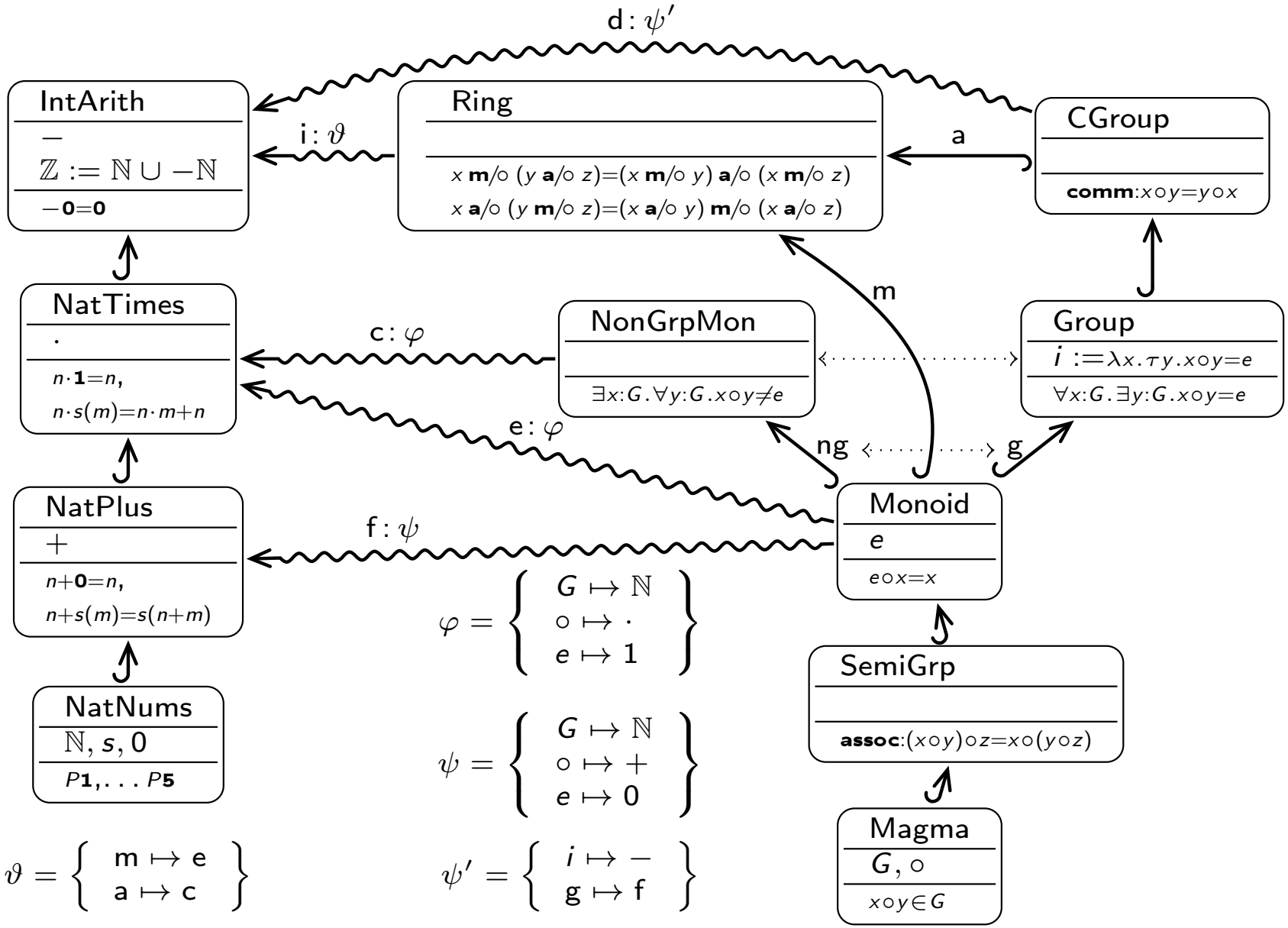
The Knowledge-First Paradigm for ODK: Interface Theories

- Build Standard and Interface Theories: Explicitly represent knowledge in joint KR frame framework



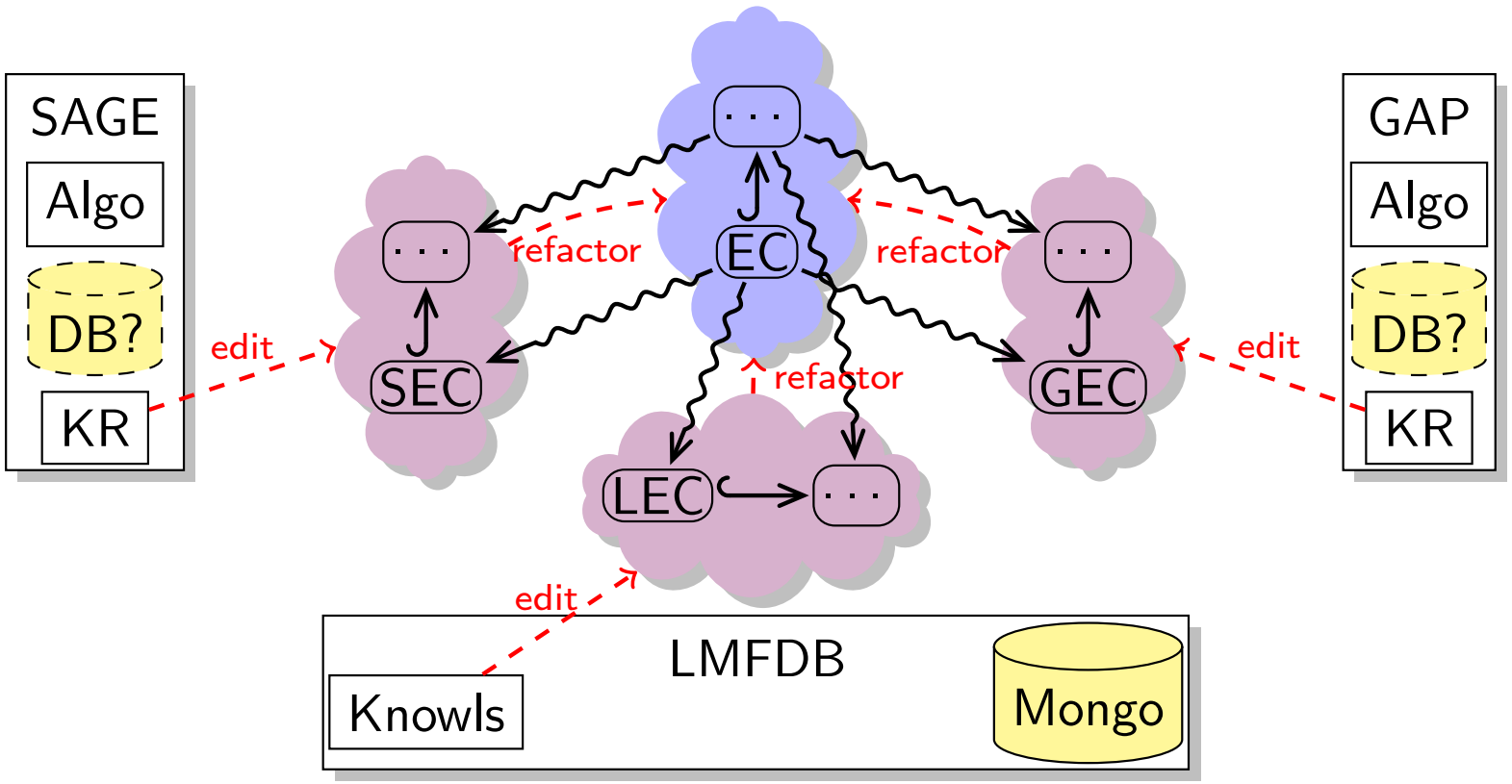
- Problem: Need an open, generic, modular KR Framework
- Solution: Use the OMDoc/MMT format and MMT system. (⇜ proposal)

Modular Representation of Math (MMT Example)



The Knowledge-First Paradigm for ODK: Theory Graphs

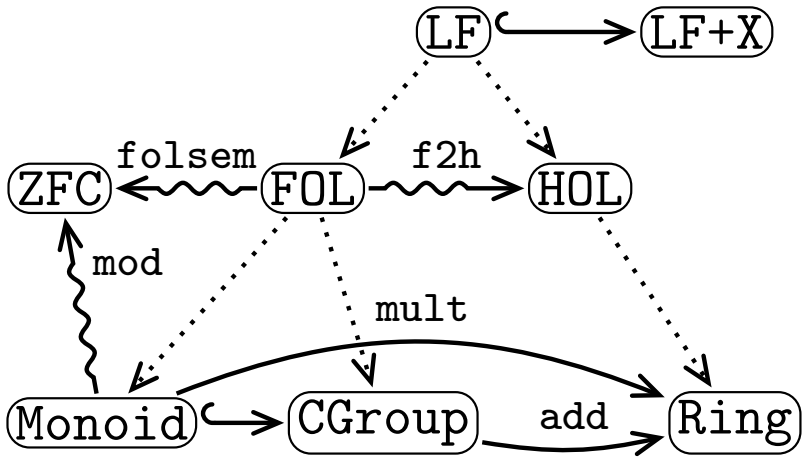
- Represent Knowledge in OMDoc/MMT: Theory graphs for math knowledge and interface theories connected by views



- Problem: Systems and Math have differing foundations. (representational primitives and assumptions)

Representing Logics and Foundations as Theories

- Logics and foundations represented as MMT theories (in the samlme graph)

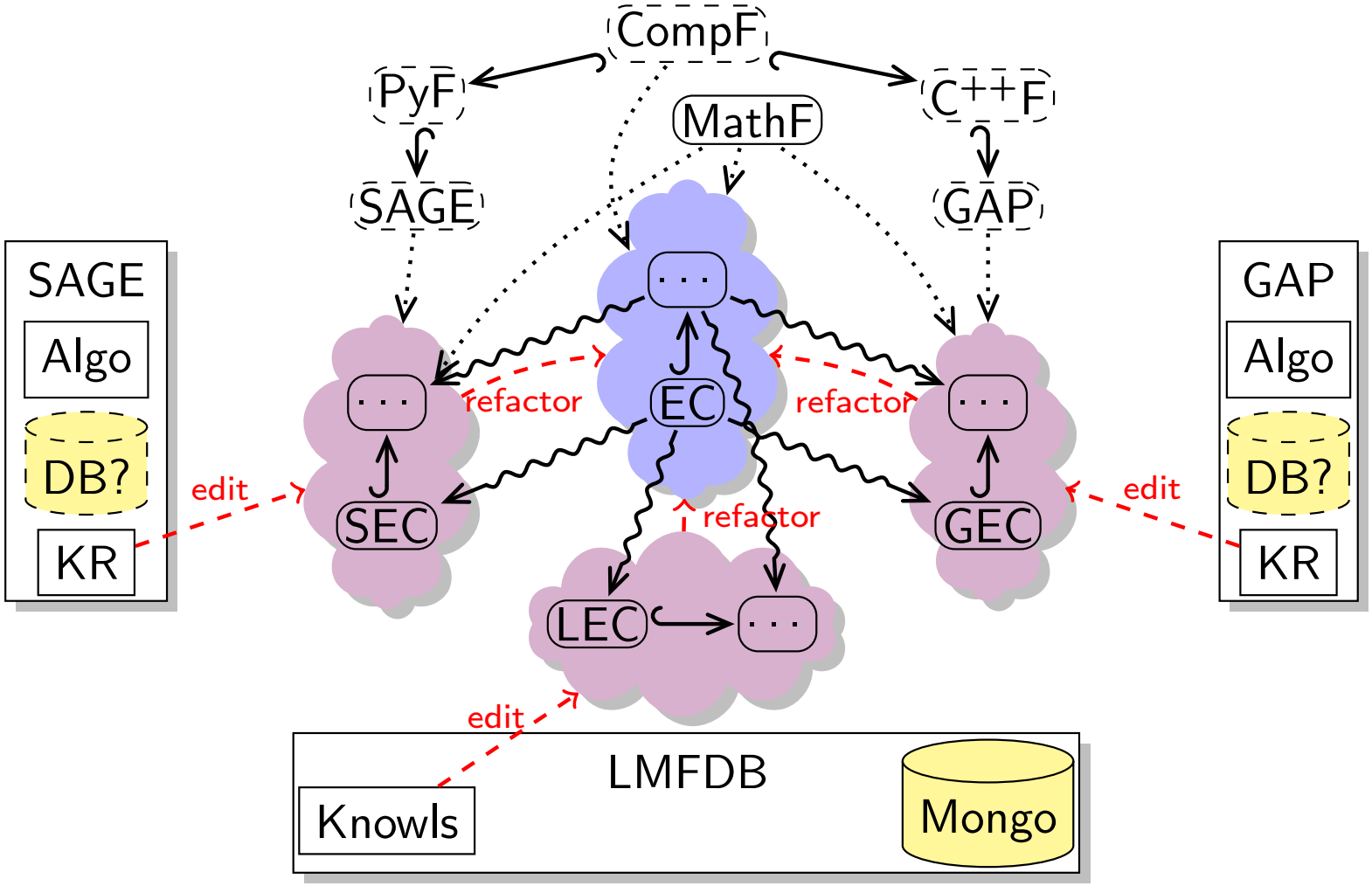


Meta-relation between theories – special case of inclusion (meta*-level)

- Uniform Meaning Space: morphisms between formalizations in different logics become possible via meta-morphisms.
- **Remark 0.3** *Semantics of logics as views into foundations, e.g.,* `folsem`.
- **Remark 0.4** *Models represented as views into foundations*
- **Example 0.5** $\texttt{mod} := \{G \mapsto \mathbb{Z}, \circ \mapsto +, e \mapsto 0\}$ interprets Monoid in ZFC.

The Knowledge-First Paradigm for ODK: Meta+Found.

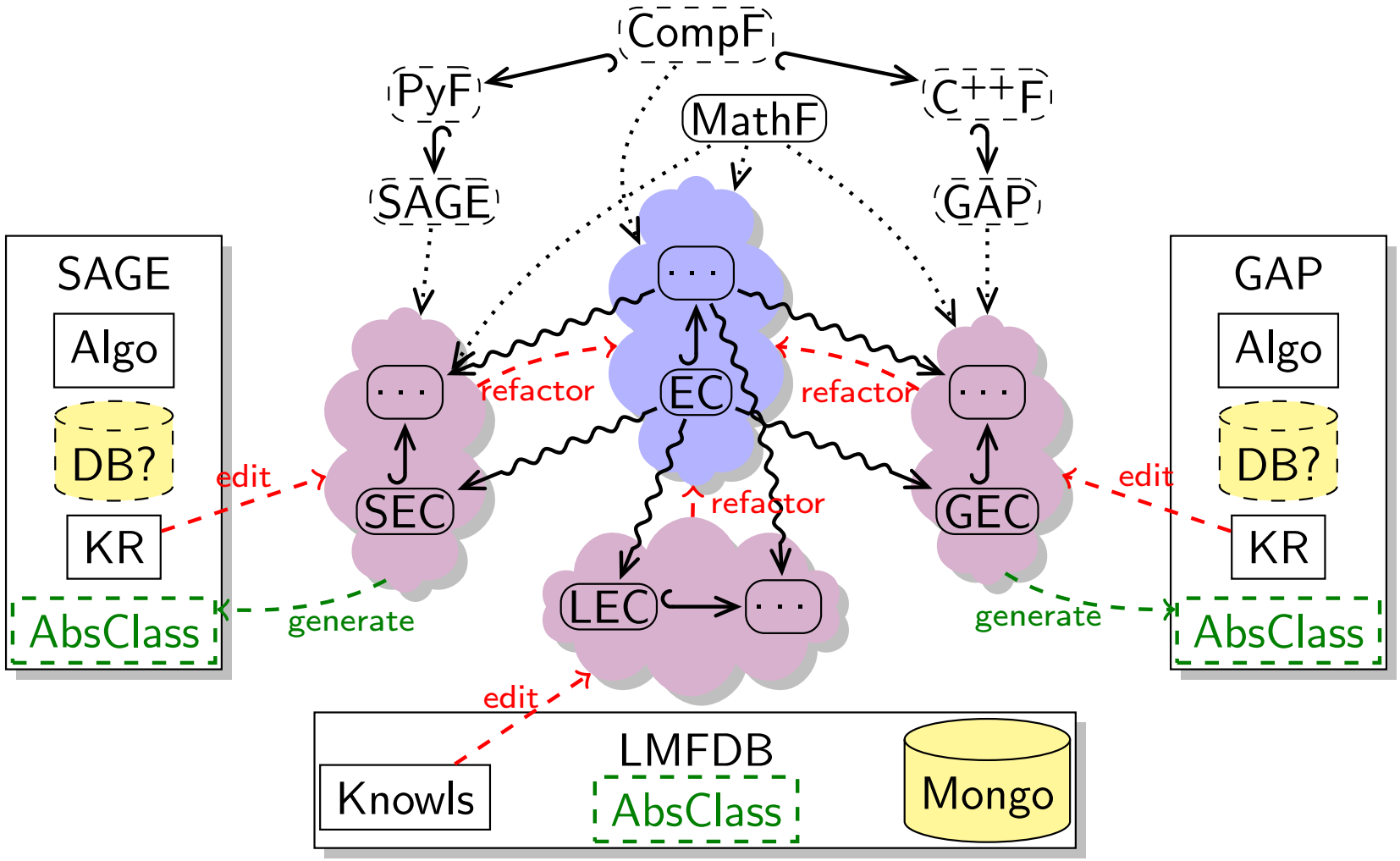
- Develop Meta-Levels: Languages and foundations for Maths and Computation



- Problem: This is a lot of work is there any benefit?

The Knowledge-First Paradigm for ODK: Integration

- Benefit: Given MMT codecs for basic data types and *OpenMath* phrasebooks, we can build ODK interoperability layer $\rightsquigarrow$ VRE (see the demo/tutorial)



Conclusion and Segway

- **Observation 0.6** *For a VRE from Open Sorce Systems we need a uniform meaning space.* *(promise/danger in the communication)*
- Idea: Center it around the shared math knowledge (Knowledge-First Pardigm)
- Idea: Represent it as OMDoc/MMT Theory graphs (profit from the MMT system)
- Demo: KWARC (JacU+Paul) has developed a first instance! (Demo/Tutorial)
- Next Talks: Find out the current state in the ODK systems