

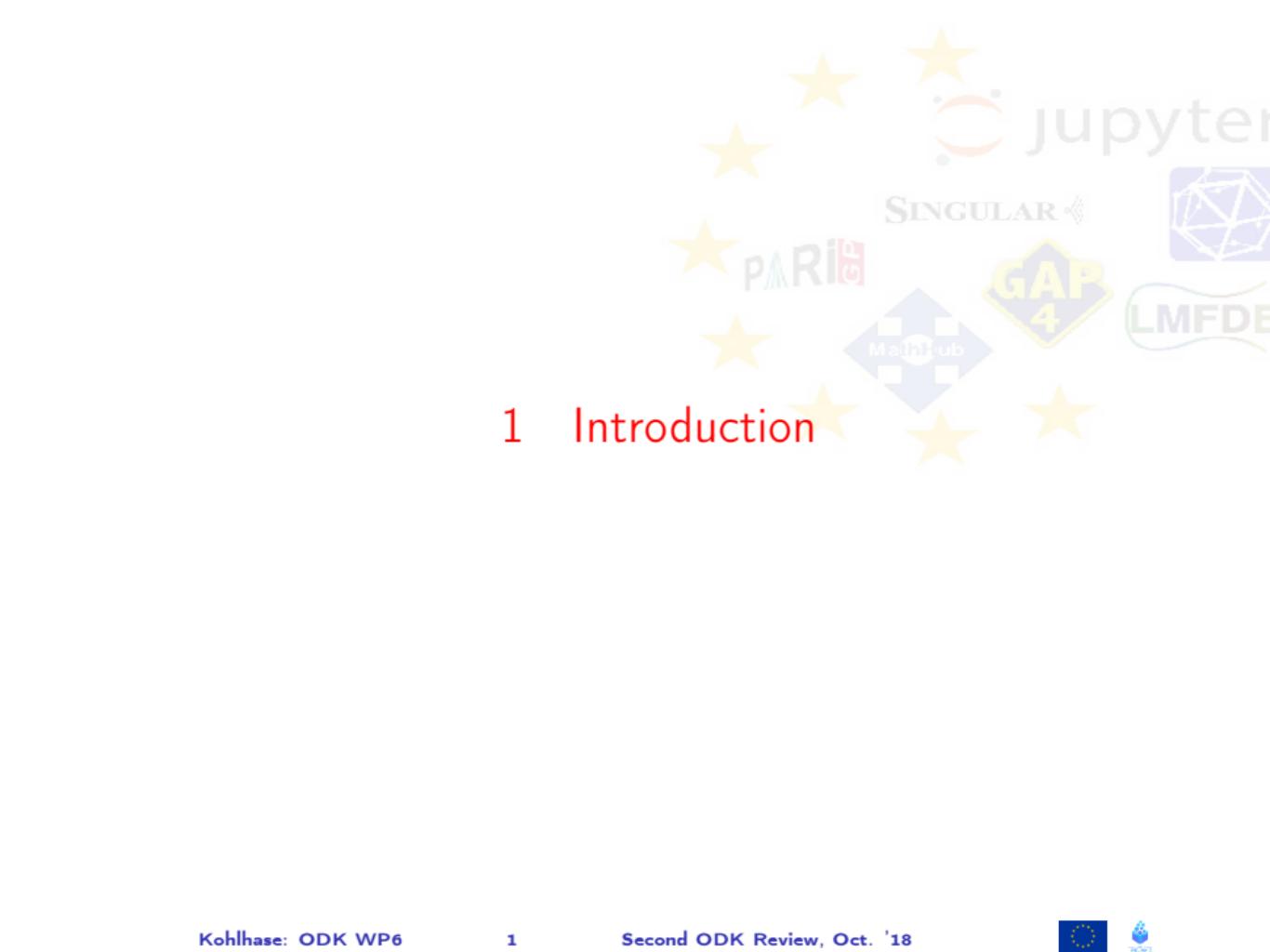
OpenDreamKit Work Package 6

Data/Knowledge/Software-Bases

Michael Kohlhase
FAU Erlangen-Nürnberg
<http://kwarc.info/kohlhase>

Second OpenDreamKit Review, Luxembourg, October 30. 2018



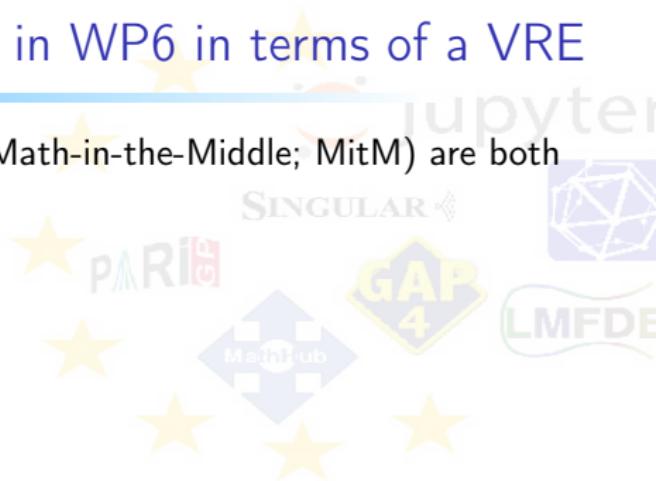


1 Introduction



Conclusion: What are we doing in WP6 in terms of a VRE

- ▶ SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.



- ▶ **overall pattern:** design – prototype – scale

Conclusion: What are we doing in WP6 in terms of a VRE

- ▶ SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- ▶ SageMath/CoCalc is **very successful**, because **integration is lightweight**:
 - ▶ It makes no assumption on the meaning of math objects exchanged.
 - ▶ Restricts itself to master-slave integration of systems into SageMath.

But there are safety, extensibility, and flexibility issues!
- ▶ **overall pattern**: design – prototype – scale

Conclusion: What are we doing in WP6 in terms of a VRE

- ▶ SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- ▶ SageMath/CoCalc is **very successful**, because **integration is lightweight**:
 - ▶ It makes no assumption on the meaning of math objects exchanged.
 - ▶ Restricts itself to master-slave integration of systems into SageMath.
But there are safety, extensibility, and flexibility issues!
- ▶ MitM tries to **take the high road** (make possible by OpenDreamKit)
 - ▶ **Safety**: by semantic (i.e. context-aware) objects passed.
 - ▶ **Extensibility**: any open-API system (i.e. with API CDs) can play.
 - ▶ **Flexibility**: full peer-to-peer possibilities. (future: service discovery)
But we have to develop a whole new framework! (Review 1 ~ Proof of Concept)
- ▶ **overall pattern**: design – prototype – scale

Conclusion: What are we doing in WP6 in terms of a VRE

- ▶ SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- ▶ SageMath/CoCalc is **very successful**, because **integration is lightweight**:
 - ▶ It makes no assumption on the meaning of math objects exchanged.
 - ▶ Restricts itself to master-slave integration of systems into SageMath.

But there are safety, extensibility, and flexibility issues!
- ▶ MitM tries to **take the high road** (make possible by OpenDreamKit)
 - ▶ **Safety**: by semantic (i.e. context-aware) objects passed.
 - ▶ **Extensibility**: any open-API system (i.e. with API CDs) can play.
 - ▶ **Flexibility**: full peer-to-peer possibilities. (future: service discovery)

But we have to develop a whole new framework! (Review 1 ~ Proof of Concept)
- ▶ **Review Period2: State of WP6 (MitM) Integration**
 - ▶ Developed mathematical use-cases (what do researchers want to do)
 - ▶ Extended middleware, grown MitM ontology, collected alignments
 - ▶ Jupyter integration into MathHub.info
- ▶ **overall pattern**: design – prototype – scale



Conclusion: What are we doing in WP6 in terms of a VRE

- ▶ SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- ▶ SageMath/CoCalc is **very successful**, because **integration is lightweight**:
 - ▶ It makes no assumption on the meaning of math objects exchanged.
 - ▶ Restricts itself to master-slave integration of systems into SageMath.

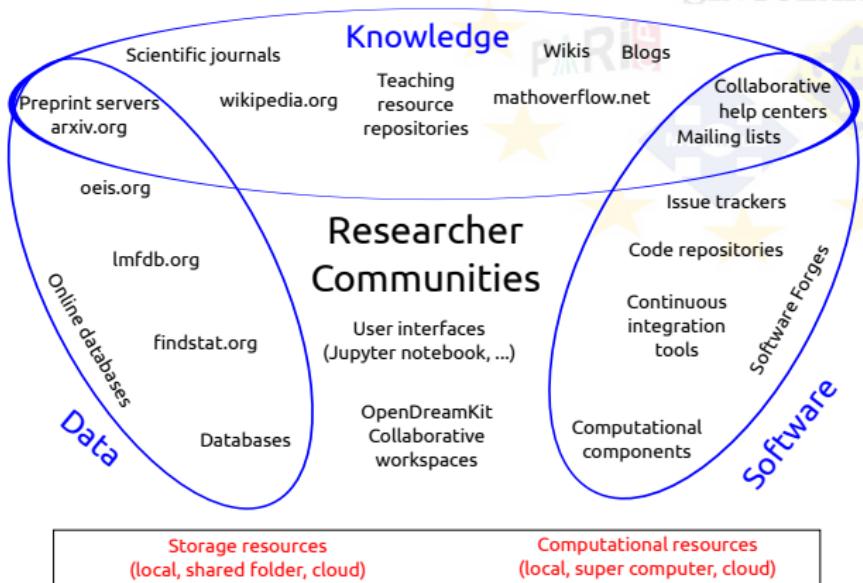
But there are safety, extensibility, and flexibility issues!
- ▶ MitM tries to **take the high road** (make possible by OpenDreamKit)
 - ▶ **Safety**: by semantic (i.e. context-aware) objects passed.
 - ▶ **Extensibility**: any open-API system (i.e. with API CDs) can play.
 - ▶ **Flexibility**: full peer-to-peer possibilities. (future: service discovery)

But we have to develop a whole new framework! (Review 1 ~ Proof of Concept)
- ▶ **Review Period2: State of WP6 (MitM) Integration**
 - ▶ Developed mathematical use-cases (what do researchers want to do)
 - ▶ Extended middleware, grown MitM ontology, collected alignments
 - ▶ Jupyter integration into MathHub.info
- ▶ **Plan for Review Period 3: Extend to external use-cases users** (scale and deploy publically)
- ▶ **overall pattern**: design – prototype – scale



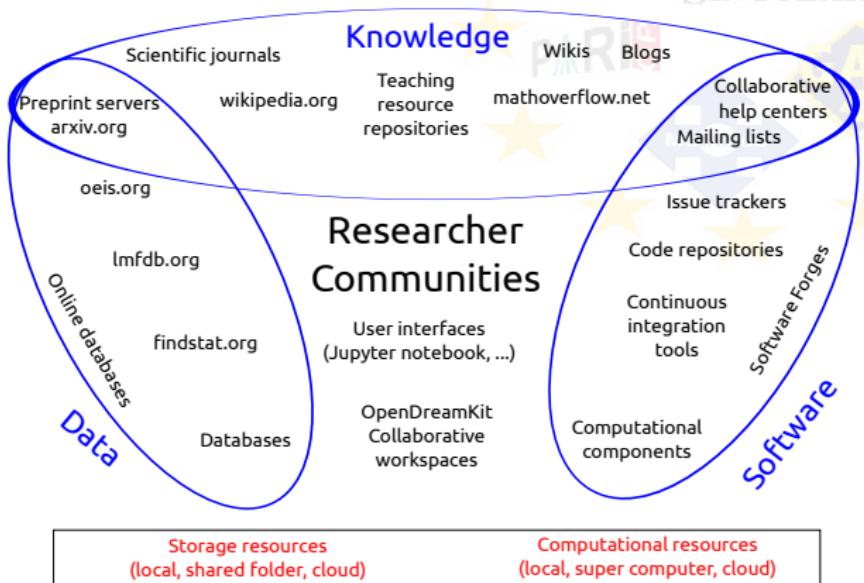
Background: WP6 (Data/Knowledge/Software-Bases)

► From the Proposal:



Background: WP6 (Data/Knowledge/Software-Bases)

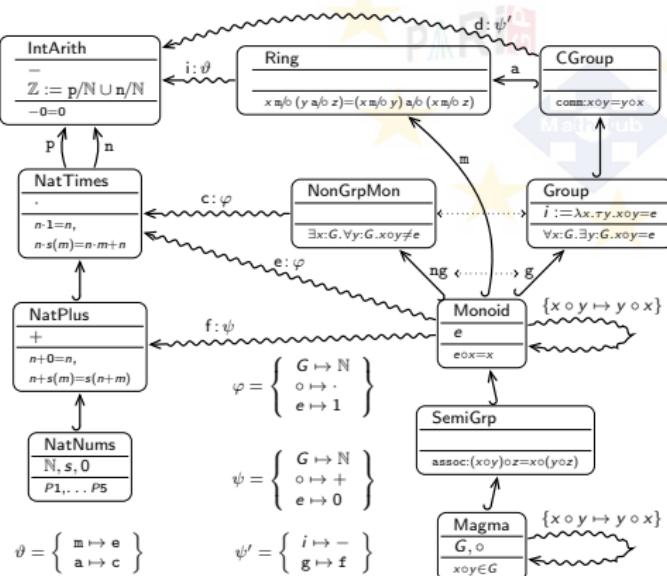
► From the Proposal:



► Proposed Focus: Supply this data to VRE components in an integrated fashion programmatically

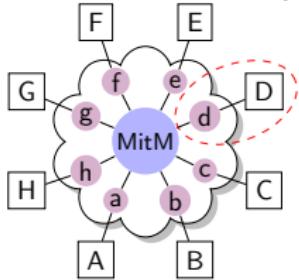
Results of the WP6 Workshops: Semantic Interoperability

- ▶ The WP6 group had a series of workshops
- ▶ Kickoff in Paris (Sep '15): strategies for joint knowledge representation



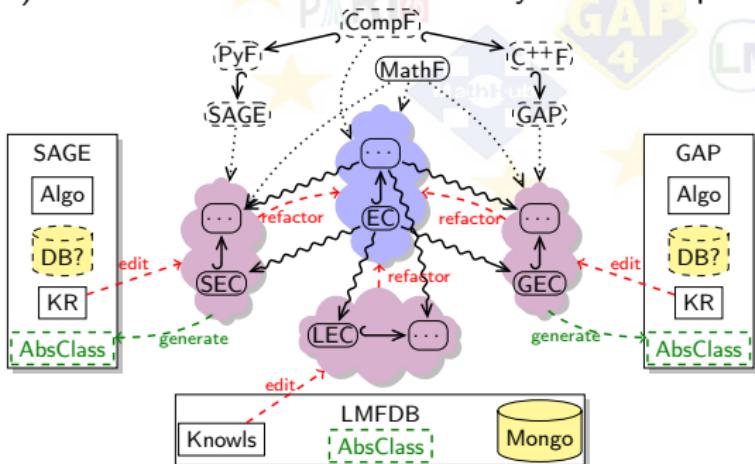
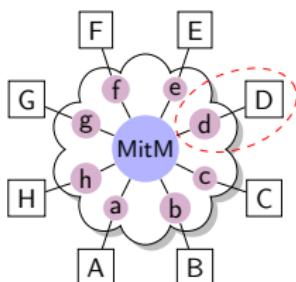
Results of the WP6 Workshops: Semantic Interoperability

- ▶ The WP6 group had a series of workshops
- ▶ Kickoff in Paris (Sep '15): strategies for joint knowledge representation
- ▶ WS in St. Andrews (Feb '16): **Math in the Middle Arch.** for System Interop.



Results of the WP6 Workshops: Semantic Interoperability

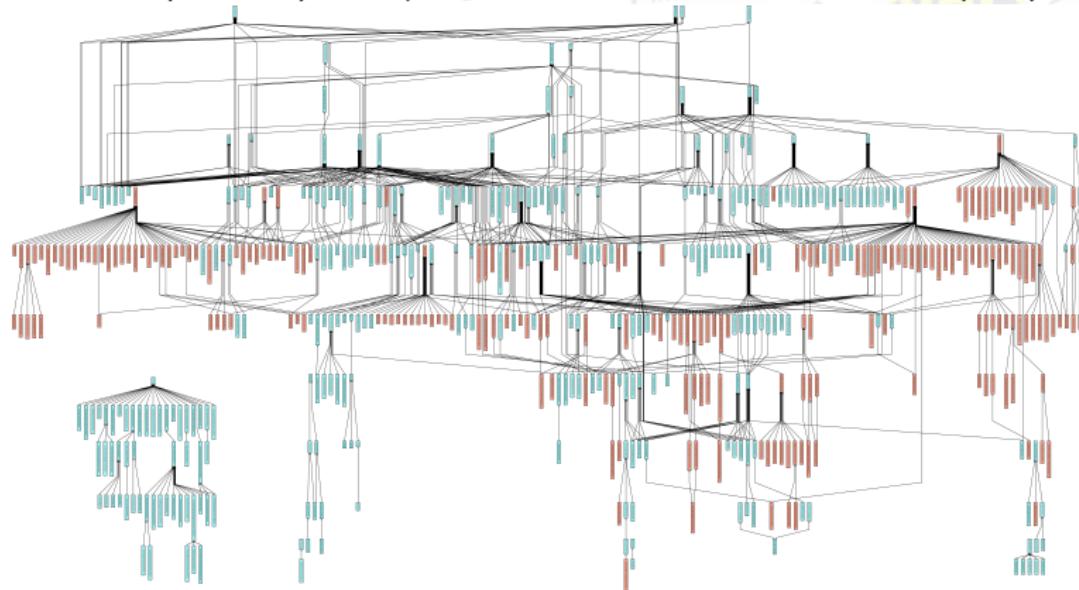
- ▶ The WP6 group had a series of workshops
- ▶ Kickoff in Paris (Sep '15): strategies for joint knowledge representation
- ▶ WS in St. Andrews (Feb '16): **Math in the Middle Arch.** for System Interop.



Paper: *Interoperability in the OpenDreamKit Project: The Math-in-the-Middle Approach* [CICM 2016]

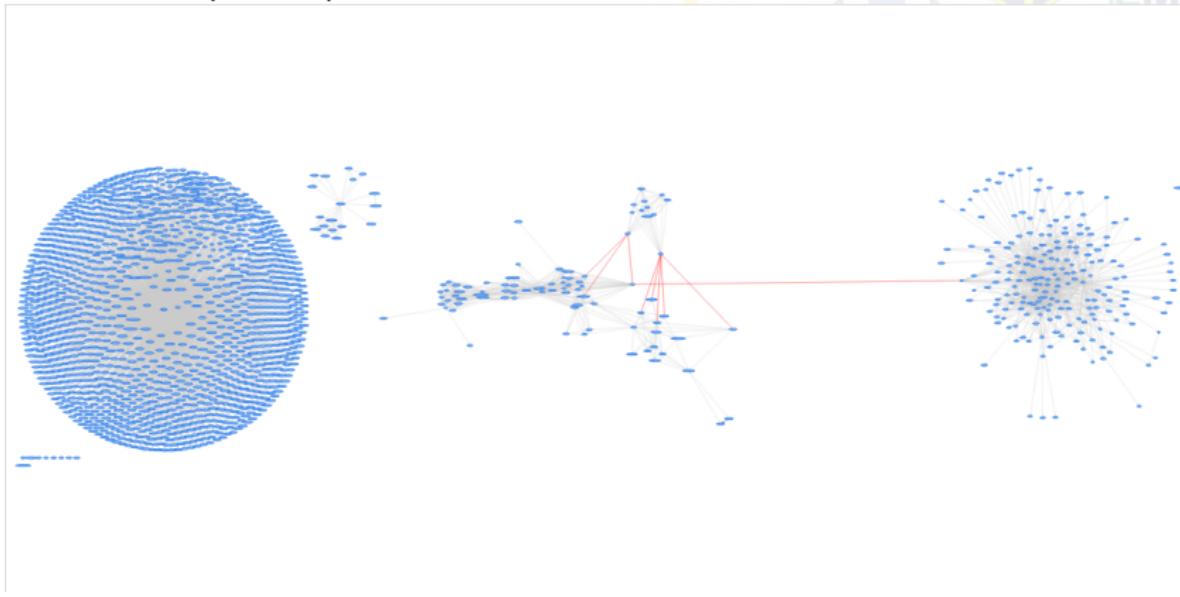
Results of the WP6 Workshops: Semantic Interoperability

- ▶ The WP6 group had a series of workshops
 - ▶ Kickoff in Paris (Sep '15): strategies for joint knowledge representation
 - ▶ WS in St. Andrews (Feb '16): Math in the Middle Arch. for System Interop.
 - ▶ WS in Bremen (June '16): GAP/SageMath API Content Dictionaries (CDs)



Results of the WP6 Workshops: Semantic Interoperability

- ▶ The WP6 group had a series of workshops
 - ▶ Kickoff in Paris (Sep '15): strategies for joint knowledge representation
 - ▶ WS in St. Andrews (Feb '16): Math in the Middle Arch. for System Interop.
 - ▶ WS in Bremen (June '16): GAP/SageMath API Content Dictionaries (CDs)
 - ▶ WS in Berlin (Feb '17): Math-in-the-Middle Ontology



2 Mathematical Use Cases



Running Example/Use Case: Jane's Invariant Experiments

- ▶ Jane wants to experiment with invariant theory of finite groups.
- ▶ She works in the polynomial ring $R = \mathbb{Z}[X_1, \dots, X_n]$.
- ▶ **Goal:** construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I .
- ▶ **Idea:** pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$.
- ▶ For effective further computation with I , she needs a Gröbner base of I .

Running Example/Use Case: Jane's Invariant Experiments

- ▶ Jane wants to experiment with invariant theory of finite groups.
- ▶ She works in the polynomial ring $R = \mathbb{Z}[X_1, \dots, X_n]$.
- ▶ **Goal:** construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I .
- ▶ **Idea:** pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$.
- ▶ For effective further computation with I , she needs a Gröbner base of I .
- ▶ Jane is a **SageMath** user and wants to receive the result in **SageMath**, but she wants to use **GAP**'s orbit algorithm and **Singular**'s Gröbner base algorithm, which she knows to be very efficient.

Running Example/Use Case: Jane's Invariant Experiments

- ▶ Jane wants to experiment with invariant theory of finite groups.
- ▶ She works in the polynomial ring $R = \mathbb{Z}[X_1, \dots, X_n]$.
- ▶ **Goal:** construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I .
- ▶ **Idea:** pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$.
- ▶ For effective further computation with I , she needs a Gröbner base of I .
- ▶ Jane is a **SageMath** user and wants to receive the result in **SageMath**, but she wants to use **GAP**'s orbit algorithm and **Singular**'s Gröbner base algorithm, which she knows to be very efficient.
- ▶ **Problem:** Jane has to learn the **GAP** and **Singular** languages and retype the results in them. (error-prone)

Running Example/Use Case: Jane's Invariant Experiments

- ▶ Jane wants to experiment with invariant theory of finite groups.
- ▶ She works in the polynomial ring $R = \mathbb{Z}[X_1, \dots, X_n]$.
- ▶ **Goal:** construct an ideal I in R that is fixed by a group $G \leq S_n$ acting on the variables, linking properties of G to properties of I and the quotient of R by I .
- ▶ **Idea:** pick some polynomial p from R and consider the ideal I of R that is generated by all elements of the orbit $O = \text{Orbit}(G, R, p) \subseteq R$.
- ▶ For effective further computation with I , she needs a Gröbner base of I .
- ▶ Jane is a **SageMath** user and wants to receive the result in **SageMath**, but she wants to use **GAP**'s orbit algorithm and **Singular**'s Gröbner base algorithm, which she knows to be very efficient.
- ▶ **Problem:** Jane has to learn the **GAP** and **Singular** languages and retype the results in them. (error-prone)
- ▶ For the sake of example, we will work with $n = 4$, $G = D_4$ (the dihedral group), and $p = 3 \cdot X_1 + 2 \cdot X_2$, but our results apply to arbitrary values.
- ▶ **Caveat:** G is called " D_4 " in **SageMath** but " D_8 " in **GAP** due to differing conventions in different mathematical communities

John's Use Case for LMFDB (slightly abridged)

- ▶ John wants to investigate the number fields which are generated by the coefficients of Hilbert modular forms (HMFs).
- ▶ LMFDB contains information about all HMFs of level $\mathcal{N} = 2, 3, 4, 5, 6$ (of parallel weight 2 and trivial character).
- ▶ Each HMF comes with a Hecke field K which is stored via a defining polynomial
(not canonical or minimal \leadsto difficult to study)
- ▶ **Example 2.1.** $K = \mathbb{Q}(\sqrt{2})$ may occur as $x^2 - 2$ and $x^2 - 2x - 1$.
- ▶ John would like to be able to
 1. extract these defining polynomials from the LMFDB,
 2. use them to define number fields in SageMath,
 3. find simpler polynomials defining the same fields, and
 4. study their arithmetic properties (e.g., their class numbers).



3 Realizing MitM Interoperability

– The Computational Group Theory Case Study –

A MitM Theory in MMT Surface Language

► Example 3.1. A theory of Groups

Declaration $\hat{=}$

name : type [= Def] [# notation]

Axioms $\hat{=}$ Declaration with type $\vdash F$

ModelsOf makes a record type from a theory.

```
theory group : base:?Logic =  
  theory group_theory : base:?Logic =  
    include ?monoid/monoid_theory |  
  
    inverse : U → U | # 1-1 prec 24 |  
    inverseproperty : ⊢ ∀ [x] x ∘ x-1 ≡ e |  
  
  group = ModelsOf group_theory |
```

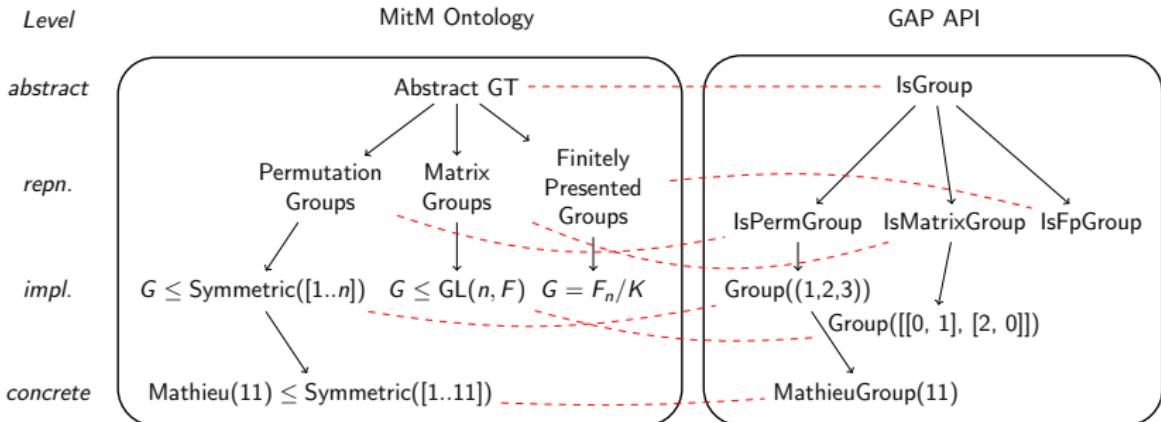
► MitM Foundation: optimized for natural math formulation

- ▶ higher-order logic based on polymorphic λ -calculus
- ▶ judgements-as-types paradigm: $\vdash F \hat{=}$ type of proofs of F
- ▶ dependent types with predicate subtyping, e.g. $\{n\}\{a \in \text{mat}(n, n) | \text{symm}(a)\}$
- ▶ (dependent) record types for reflecting theories

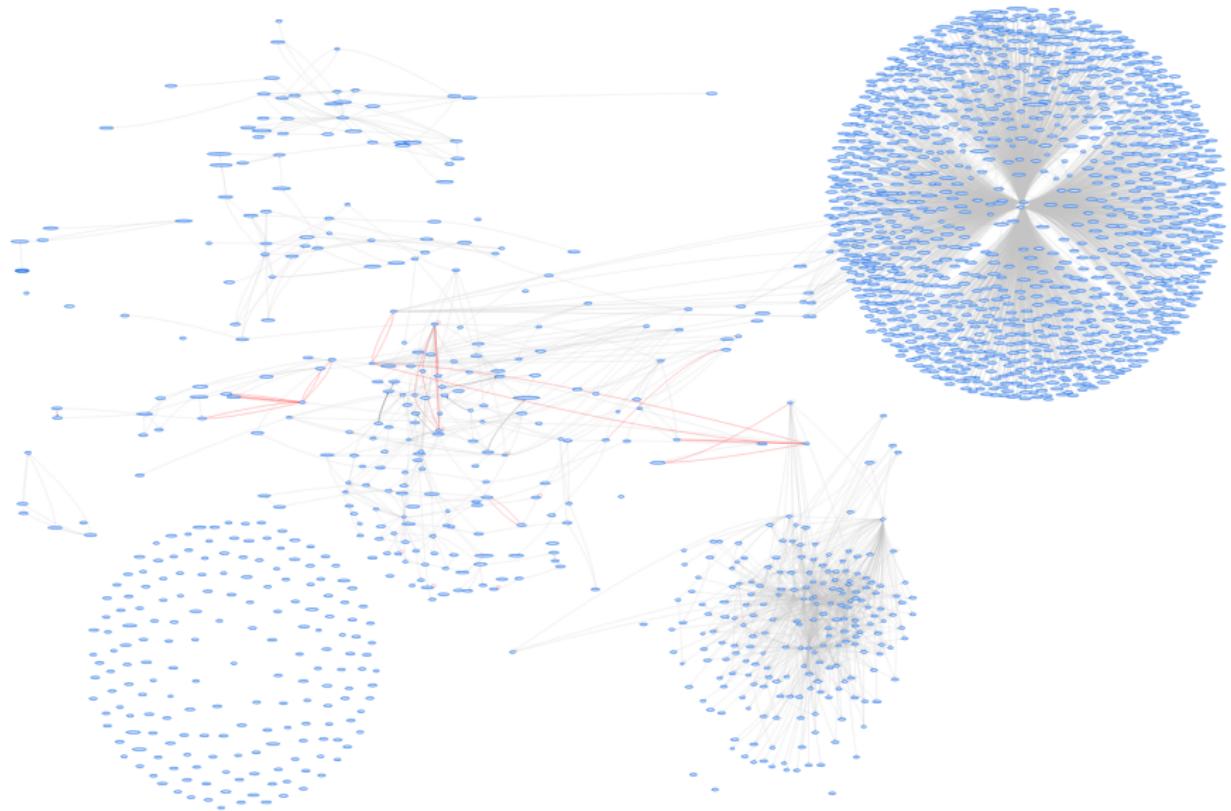
MitM Computational Group Theory

(Following the GAP template)

- ▶ Four levels of modeling
- ▶ **Abstract Level:** the group axioms, generating sets, homomorphisms, group actions, stabilisers, orbits, centralizers, normalizers.
- ▶ **Representation Level:** axiomatizations concrete objects suitable for computation – permutation groups, matrix groups, . . . , also group actions, group homomorphism
- ▶ **Implementation Level:** permutation groups as subgroups of $S_{\mathbb{N}^+}$, concretely $S_{[1, \dots, n]}$.
- ▶ **Concrete Level:** where actual computations happen.
- ▶ Alignments between the MitM Ontology and the GAP API



The Knowledge Graph for MitM, SageMath, GAP, Singular



Jane's Use case in the MitM System

- In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.

Jane's Use case in the MitM System

- ▶ In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.

- ▶ She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit  
i = MitM.Singular(o).Ideal() # the ideal  
g = i.Groebner().sage() # the Groebner basis
```

Jane's Use case in the MitM System

- ▶ In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- ▶ She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```
- ▶ The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.

Jane's Use case in the MitM System

- ▶ In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- ▶ She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```
- ▶ The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.
- ▶ GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$

Jane's Use case in the MitM System

- ▶ In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- ▶ She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```
- ▶ The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.
- ▶ GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$
- ▶ The MitM server translates `MitM.Singular(O).Ideal().Groebner()` to the Singular system dialect and sends it to Singular..

Jane's Use case in the MitM System

- ▶ In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- ▶ She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```
- ▶ The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.
- ▶ GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$
- ▶ The MitM server translates `MitM.Singular(O).Ideal().Groebner()` to the Singular system dialect and sends it to Singular..
- ▶ Singular returns the Gröbner base B .

Jane's Use case in the MitM System

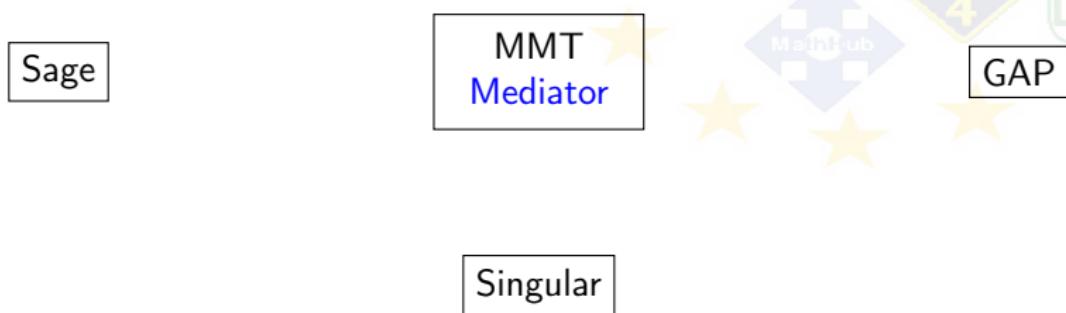
- ▶ In SageMath Jane has already built the ring $R = \mathbb{Z}[X_1, X_2, X_3, X_4]$, the group $G = D_4$, the action A of G on R that permutes the variables, and $p = 3 \cdot X_1 + 2 \cdot X_2$.
- ▶ She calls

```
o = MitM.Gap.orbit(G,A,p) # the orbit
i = MitM.Singular(o).Ideal() # the ideal
g = i.Groebner().sage() # the Groebner basis
```
- ▶ The MitM server translates `MitM.Gap.orbit(G,A,p)` to the GAP system dialect and sends it to GAP.
- ▶ GAP returns the orbit: $O = [3X_1 + 2X_2, 2X_3 + 3X_4, 3X_2 + 2X_3, 3X_3 + 2X_4, 2X_2 + 3X_3, 3X_1 + 2X_4, 2X_1 + 3X_4, 2X_1 + 3X_2]$
- ▶ The MitM server translates `MitM.Singular(O).Ideal().Groebner()` to the Singular system dialect and sends it to Singular..
- ▶ Singular returns the Gröbner base B .
- ▶ The MitM server translates B to the SageMath system dialect and sends it to SageMath, where the result is shown to Jane.

$$B = [X_1 - X_4, X_2 - X_4, X_3 - X_4, 5 * X_4].$$

Distributed Computational Group Theory

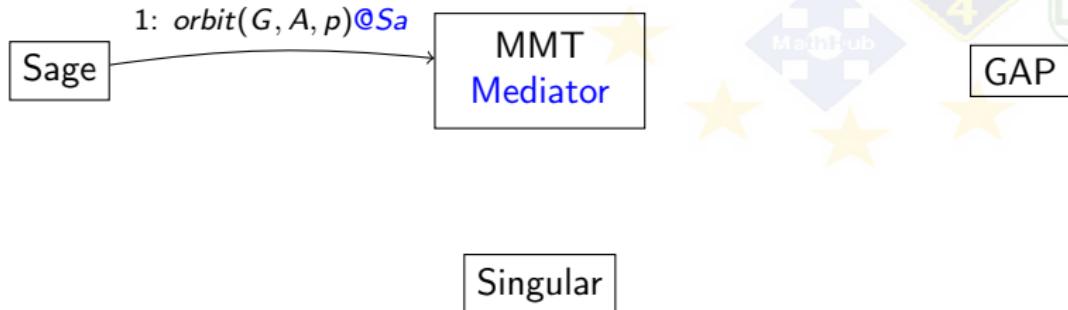
- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

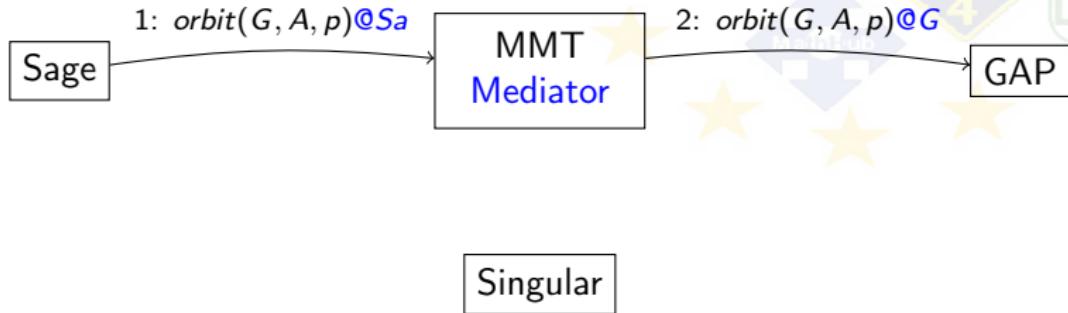
- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.

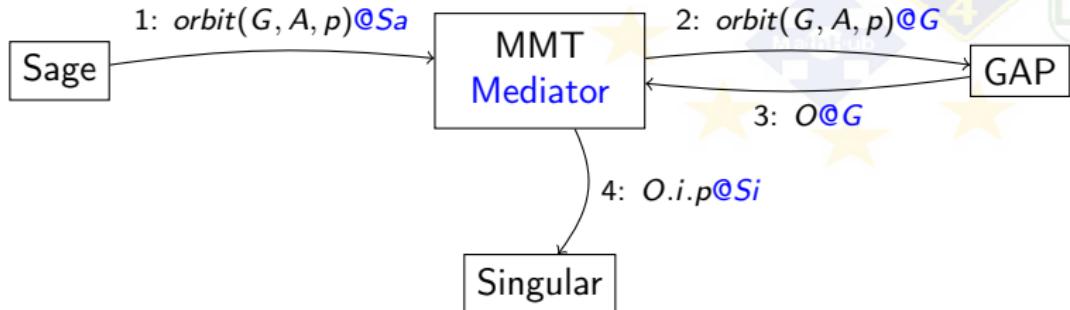


Singular

- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

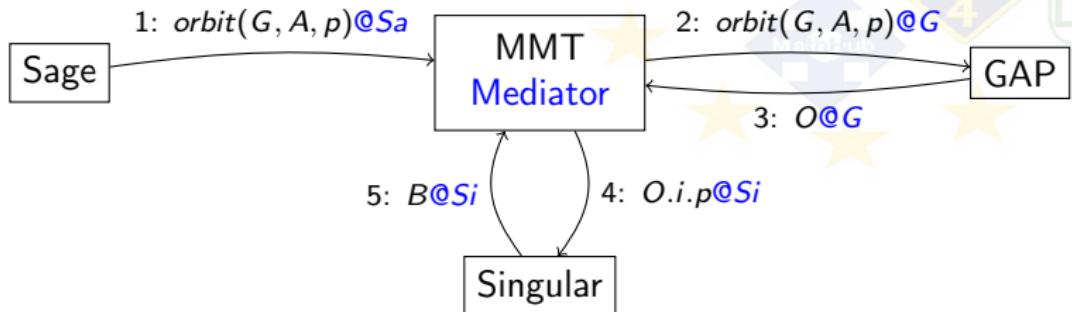
- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

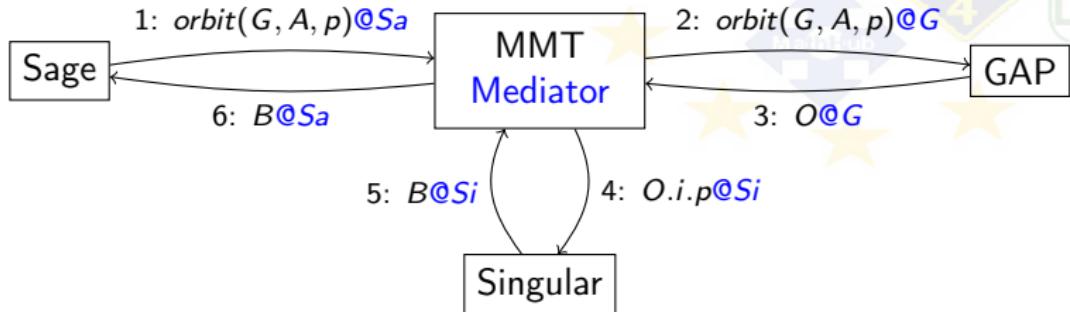
- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Distributed Computational Group Theory

- ▶ Combine SCSCP enabled GAP, SageMath, and Singular with MMT mediator.



- ▶ Nucleus of the OpenDreamKit interoperability layer.
Delegate computations between systems if exchanged objects are covered by the MitM ontology, the API theories, and the alignments

Future Use Case (Steve is Jane's Colleague)

- ▶ Steve prefers working in **GAP**, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- ▶ He discovers the **GAP** package `radiroot` (does not work for p)

Future Use Case (Steve is Jane's Colleague)

- ▶ Steve prefers working in **GAP**, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- ▶ He discovers the **GAP** package `radiroot` (does not work for p)
- ▶ Jane suggests **PariGP**: he calls (once that is MitM-connected)
`G := MitM("PariGP", "GaloisGroup", p)` from **PariGP** which gives him the desired Galois group as a **GAP** permutation group.

Future Use Case (Steve is Jane's Colleague)

- ▶ Steve prefers working in **GAP**, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- ▶ He discovers the **GAP** package `radiroot` (does not work for p)
- ▶ Jane suggests **PariGP**: he calls (once that is MitM-connected)
`G := MitM("PariGP", "GaloisGroup", p)` from **PariGP** which gives him the desired Galois group as a **GAP** permutation group.
- ▶ Steve repeats Jane's experiments on `G`, without leaving **GAP**.

Future Use Case (Steve is Jane's Colleague)

- ▶ Steve prefers working in **GAP**, and he wants to compute the Galois group of the rational polynomial $p = x^5 - 2$.
- ▶ He discovers the **GAP** package **radiroot** (does not work for p)
- ▶ Jane suggests **PariGP**: he calls (once that is MitM-connected)
`G := MitM("PariGP", "GaloisGroup", p)` from **PariGP** which gives him the desired Galois group as a **GAP** permutation group.
- ▶ Steve repeats Jane's experiments on **G**, without leaving **GAP**.
- ▶ Finally, Steve installs a **GAP** method by calling

```
InstallMethod(GaloisGroup, "for a polynomial", [IsUnivariatePolynomial],  
             p -> MitM("PariGP", "GaloisGroup", p))
```

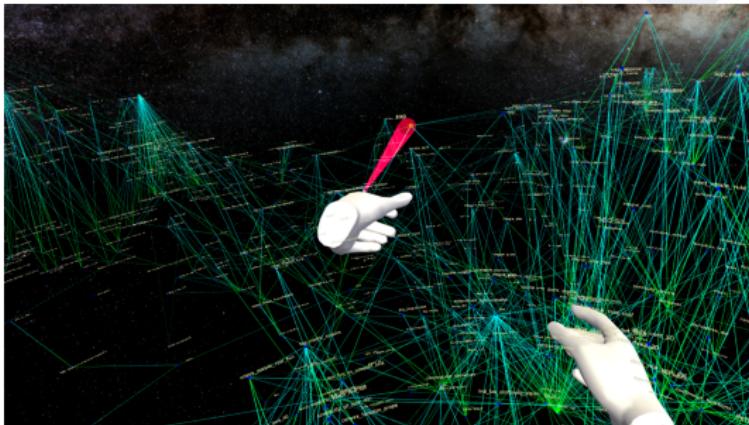
~> extends **GaloisGroup** to rational polynomials in **GAP**.
- ▶ This replaces a significant part of the 1800-LoC **radiroot** package (by **PariGP delegation**)



MitM-based Integration centers around the MitM Ontology

If you are Really interested in the Graphs

interact with them in 3D





4 MitM InterOperability for Mathematical Databases

Mathematical Knowledge Bases (MKS)

► State of the Art: mathematical object databases (GAP libraries, OEIS, LMFDB)

LMFDB

Elliptic Curve Isogeny Class 11.a (Cremona lab)

Introduction and more

Introduction Features Universe Future Plans News

L-functions

Degree: 1 2 3 4

ζ zeros

Modular Forms

Rank

The elliptic curves in class 11.a have rank 0.

Modular form 11.2.1.a

$q - 2q^2 - q^3 + 2q^4 + q^5 + 2q^6 - 2q^7 - 2q^8 + 2q^9 + q^{10} + q^{11} - 2q^{12} + 4q^{13} + 4q^{14} - q^{15} - 4q^{16} - 2q^{17} +$

Show more coefficients

Isogeny matrix

Varieties

Elliptic: /0 /NumberFields

Conics

Genus 2: /0

Isogeny graph

LMFDB label Cremona label Weierstrass coefficients Torsion order Modular degree O

11.a1	11a2	[0, -1, 1, -7820, -253580]	1	5	
11.a2	11a1	[0, -1, 1, -10, -20]	5	1	$\Gamma_0($
11.a3	11a3	[0, -1, 1, 0, 0]	5	5	

SINGULAR

This site is supported by donations to The OEIS Foundation.

THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES®

founded in 1964 by N. J. A. Sloane

Annual appeal: Please make a donation to keep the OEIS running! Over 6000 articles have referenced us, often saying "we discovered this result with the help of the OEIS".

Donate

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

A000045 Fibonacci numbers: $F(n) = F(n-1) + F(n-2)$ with $F(0) = 0$ and $F(1) = 1$.
Formerly M0992 N0356

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 823240, 1346269, 2178309, 3524578, 5702887, 9327465, 14930352, 24157817, 39088169 (list; graph; refs; index; history; search)

OFFSET 0,4
COMMENTS Also sometimes called Lame's sequence.
 $F(n+2) =$ number of binary sequences of length n that have no consecutive 0's.
 $F(n+2) =$ number of subsets of $\{1, 2, \dots, n\}$ that contain no consecutive integers.
 $F(n+1) =$ number of tilings of a $2 \times n$ rectangle by 2×1 dominoes.
 $F(n+1) =$ number of matchings (i.e., Hosoya index) in a path graph on n vertices: $F(5)=5$ because the matchings of the path graph on the vertices A, B, C, D are the empty set, $\{AB\}$, $\{AC\}$ and $\{AD\}$. $F(6) =$ [Révérin](#)

Mathematical Knowledge Bases (MKS)

- ▶ **State of the Art:** mathematical object databases (**GAP** libraries, **OEIS**, **LMFDB**)
- ▶ **Problem:** human-oriented interface, very limited programmatic API, no computation

The screenshot shows the LMFDB API interface for the `transitivegroups/groups` endpoint. The top navigation bar includes links for API documentation and search.

API - transitivegroups/groups

Formats: - [HTML](#) - [YAML](#) - [JSON](#) - 2017-11-14T20:02:56.693021 - [next page](#)

Query: /api/transitivegroups/groups/?_offset=0&cyc=1

Object 0: ObjectID('4e6db00aeb55b70c8000000')

```
{'ab': 1, 'arith_equiv': 0, 'auts': 1, 'cyc': 1, 'label': '1T1', 'n': 1, 'name': 'Trivial group', 'order': '1', 'parity': 1, 'pretty': 'Trivial', 'prim': 1, 'repns': [], 'resolve': [], 'solv': 1, 'subs': [], 't': 1}
```

Object 1: ObjectID('4e8df0cc0eb55b03cc000000')

```
{'ab': 1, 'arith_equiv': 0, 'auts': 12, 'cyc': 1, 'label': '12T1', 'n': 12, 'name': 'C(4)[x]C(3)', 'order': '12', 'parity': -1, 'pretty': 'C(4)[x]C(3)', 'prim': 0, 'repns': [], 'resolve': [[2, [2, 1]], [3, [3, 1]], [4, [4, 1]], [6, [6, 1]]], 'solv': 1, 'subs': [[2, 1], [3, 1], [4, 1], [6, 1]], 't': 1}
```

Object 2: ObjectID('4e68db140eb55b70c8000005')

```
{'ab': 1, 'arith_equiv': 0, 'auts': 9, 'cyc': 1, 'label': '9T1', 'n': 9, 'name': 'C(9)=9', 'order': '9', 'parity': 1, 'pretty': 'C(9)=9', 'prim': 0, 'repns': [], 'resolve': [[3, [3, 1]]], 'solv': 1, 'subs': [[3, 1]], 't': 1}
```

- ▶ **Idea:** can't we use MitM Technologies here to integrate?

John's Use Case for LMFDB (slightly abridged)

- ▶ John wants to investigate the number fields which are generated by the coefficients of Hilbert modular forms (HMFs).
- ▶ LMFDB contains information about all HMFs of level $\mathcal{N} = 2, 3, 4, 5, 6$ (of parallel weight 2 and trivial character).
- ▶ Each HMF comes with a Hecke field K which is stored via a defining polynomial
(not canonical or minimal \leadsto difficult to study)
- ▶ **Example 4.1.** $K = \mathbb{Q}(\sqrt{2})$ may occur as $x^2 - 2$ and $x^2 - 2x - 1$.
- ▶ John would like to be able to
 1. extract these defining polynomials from the LMFDB,
 2. use them to define number fields in SageMath,
 3. find simpler polynomials defining the same fields, and
 4. study their arithmetic properties (e.g., their class numbers).



MitM-based Integration of Math Knowledge Bases

► Requirements:

- a **uniform** programmatic API to multiple MKB
- interacting with MKB at the “mathematics Level”.

► Idea: use the Math-in-the-Middle Paradigm

- OMDoc/MMT-based API theories for the mathematical interface (~ MKB records as OM objects)
- alignments into MitM Ontology (for OM-dialect mediation)
- extend MMT's built-in query language **QMT** to general Math query language

► Problems:

- MKB tables become OMDoc/MMT theories (size problems)
- how to reconcile MKB records with OMDoc/MMT terms. (encoding/decoding)
- how to translate math-level queries to physical database queries



► Example 4.2 (A transitive group represented in LMFDB).

```
{  
    "ab": 1,  
    "arith_equiv": 0,  
    "auts": 1,  
    "cyc": 1,  
    "label": "1T1",  
    "n": 1,  
    ...  
}
```

Legend: for understanding them

(LMFDB improved documentation)

- the cyc field represents **being cyclic**
- the n field represents **degree**
- ...

(0 is **false**, 1 is **true**)

(IEEE Float 1 corresponds to $1 \in \mathbb{N}$)

Two Problems: that have to be solved for MitM integration

- data base schema is not at the mathematical level (let alone interoperable)
- values are encoded for MongoDB convenience (what do they mean?)

Codecs: Encoding and Decoding Database Values

- ▶ **Definition 4.3 (Codec).** A codec consists of two functions that translate between **semantic types** and **realized types**.

Codecs		MachHub
codec : type → type		
StandardPos	: codec \mathbb{Z}^+	JSON number if small enough, else JSON string of decimal expansion
StandardNat	: codec \mathbb{N}	
StandardInt	: codec \mathbb{Z}	
IntAsArray	: codec \mathbb{Z}	JSON List of Numbers
IntAsString	: codec \mathbb{Z}	JSON String of decimal expansion
StandardBool	: codec \mathbb{B}	JSON Booleans
BoolAsInt	: codec \mathbb{B}	JSON Numbers 0 or 1
StandardString	: codec \mathbb{S}	JSON Strings

- ▶ StandardInt decodes 1 into the float 1, but 2^{54} into the string "18014398509481984"

Elliptic Curve Code Operators



```
{  
    "degree": 1,  
    "x-coordinates_of_integral_points": "[5,16]",  
    "isogeny_matrix": [[1,5,25],[5,1,5],[25,5,1]],  
    "label": "11a1",  
    "_id": "ObjectId('4f71d4304d47869291435e6e')",  
    ...  
}
```

- ▶ Matrix in the `isogeny_matrix` field



$$\begin{bmatrix} 1 & 5 & 25 \\ 5 & 1 & 5 \\ 25 & 5 & 1 \end{bmatrix}$$

- ▶ represented as `[[1,5,25],[5,1,5],[25,5,1]]`

- ▶ **Definition 4.4 (Codec Operator).** A codec operator is a function which takes a codec, a set of parameters, and returns a codec.

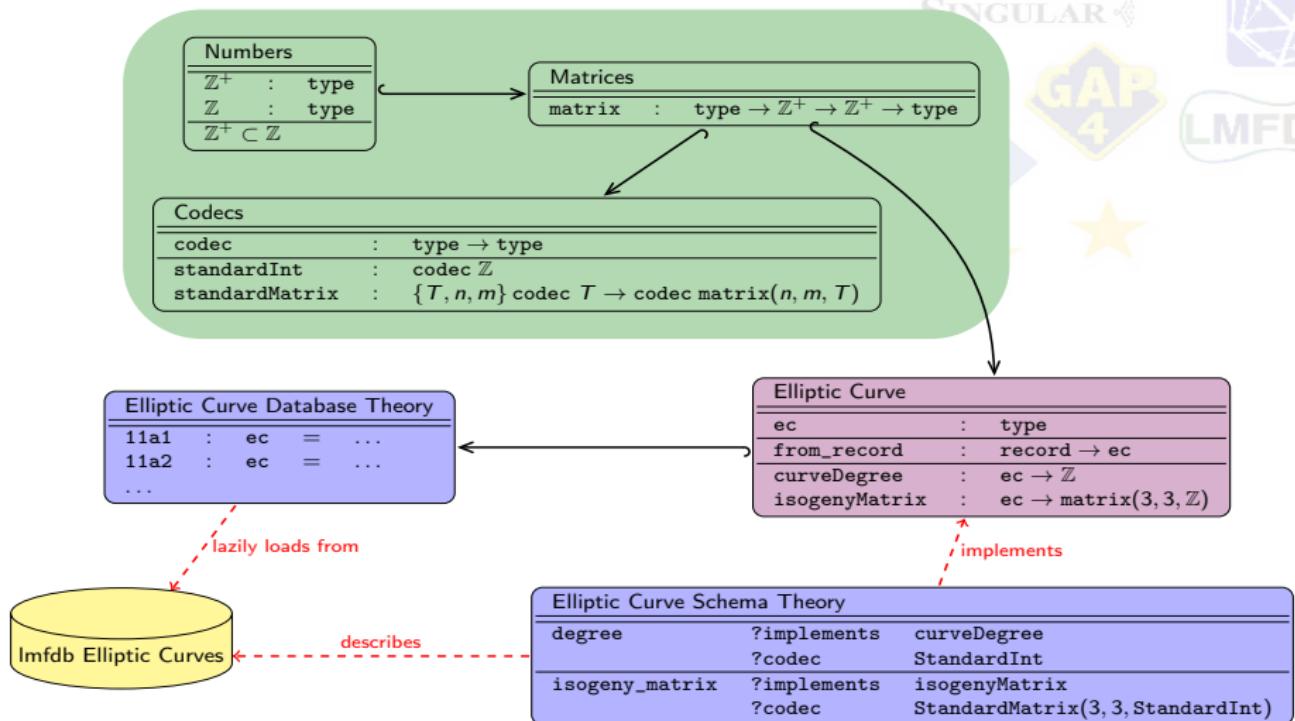


Codecs (continued)

StandardList	: codec $T \rightarrow \text{codec List}(T)$	JSON list, recursively coding each element of the list
StandardVector	: codec $T \rightarrow \text{codec Vector}(n, T)$	JSON list of fixed length n
StandardMatrix	: codec $T \rightarrow \text{codec Matrix}(n, m, T)$	JSON list of n lists of length m

- ▶ StandardMatrix(StandardInt, 3, 3) generates the codec we used for the isogeny matrix

Our approach: Virtual Theories



An Example of a Query

- ▶ **Example 4.5.** Finding all cyclic transitive groups in LMFDB (recall from above)

```
x in (related to ( literal 'lmfdb:db/transitivegroups?group ) by (object declares))  
| holds x (x cyclic x *==* true)
```

- ▶ This example does not rely on the internal structure of LMFDB
- ▶ can be translated into an LMFDB query using the just-defined **codecs theory**
- ▶ <http://www.lmfdb.org/api/transitivegroups/groups/?cyc=1>

Solving John's Hecke Fields Use Case

- ▶ Remember: John wanted to study number fields of HMFs via their Hecke field polynomials.
- ▶ John computes in SageMath and accesses LMFDB programmatically at the mathematical level
 - (directly in the MitM dialect)
- ▶ Build a query for LMFDB
 - lmfdb = MitM.lmfdb
 - algebra = MitM.smglom.algebra

```
# a MitM expression that returns all hmf_forms with degree 2
hmfs_query = lmfdb.hmf_forms.where(algebra.base_field_degree(2))
```

```
# a MitM expression that additionally extracts the Hecke polynomial
# from each hmf_form
polys_query = hmfs_query.map(lambda x: lmfdb.hecke(x))
```

- ▶ run the query via MitM and obtain the set of Sage polynomials
 - polys = MitM.run(polys_query)
- ▶ further processing in Sage
 - fields = [NumberField(p) for p in polys]

5 Jupyter Integration into MathHub

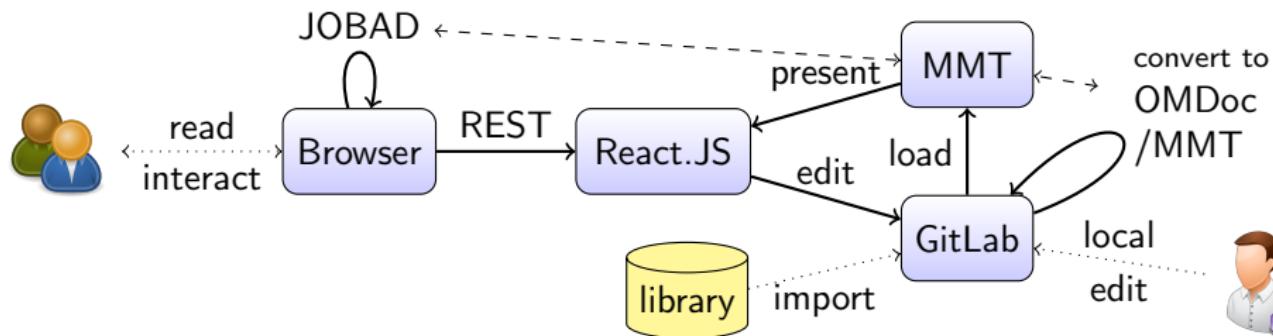


MathHub: A Portal and Archive of Flexiformal Maths

- ▶ **Idea:** learn from the open source community, offer a code repository with management support that acts as a hub for publication/development projects.
- ▶ **MathHub:** a collaborative development/hosting/publishing system of open-source, formal/informal math.
(See <http://mathhub.info>)

MathHub: A Portal and Archive of Flexiformal Maths

- ▶ **Idea:** learn from the open source community, offer a code repository with management support that acts as a hub for publication/development projects.
- ▶ **MathHub:** a collaborative development/hosting/publishing system of open-source, formal/informal math. (See <http://mathhub.info>)
- ▶ **MathHub Architeture:** Three core components (meet requirements above)
- ▶ **Representation:** OMDoc/MMT mechanized by the MMT system.
- ▶ **Repositories:** GitLab (git-based public/private repositories)
- ▶ **Front-End:** React.JS (all content served by MMT)



An OpenDreamKit Risk come True

- ▶ **Drupal Apocalypse:**
 - ▶ The MathHub front-end was based on Drupal
 - ▶ our Drupal server was repeatedly hacked and compromised → large maintenance overhead
- ▶ **Decision in April 2018:** Completely re-develop MathHub front-end using a web framework only.
This was planned anyway (Drupal too heavyweight), but cost us months developer time until now.
- ▶ The new architecture (Docker compose + JSON + React.JS) helped integrate with Jupyter.

KPIs and Deliverables for WP6

- ▶ MitM-connected Systems: four ([GAP](#), [Sage](#), [LMFDB](#), [Singular](#)) (See D6.5)
- ▶ Formal MitM Ontology: 55 files, 2600 LoF, 360 commits (See D6.8)
- ▶ Informal MitM Ontology: 815 theories, 1700 concepts in English, German, (Romanian, Chinese)
- ▶ MitM System API Theories (GAP, Sage, LMFDB, Singular): 1.000+ Theories, 22.000 Concepts.
- ▶ Multi-Site involvement of Researchers ([Mobility of Researchers](#))
 - ▶ PD. Dr. Florian Rabe (Joint appointment UPSud/FAU)
 - ▶ Felix Schmoll Summer Internship (From JacU to St.Andrews)
 - ▶ Prof. Nathan Carter (Bentley Univ.) in St. Andrews (Sabbatical)
- ▶ Heavy interest by the theorem proving community about MitM Ontology
- ▶ First ODK-external MitM “user” for the next months: Andrea Thevis, Saarbrücken

Conclusion: What are we doing in WP6 in terms of a VRE

- ▶ SageMath/CoCalc and WP6 approach (Math-in-the-Middle; MitM) are both attempts at making a VRE Toolkit.
- ▶ SageMath/CoCalc is **very successful**, because **integration is lightweight**:
 - ▶ It makes no assumption on the meaning of math objects exchanged.
 - ▶ Restricts itself to master-slave integration of systems into SageMath.

But there are safety, extensibility, and flexibility issues!
- ▶ MitM tries to **take the high road** (make possible by OpenDreamKit)
 - ▶ **Safety**: by semantic (i.e. context-aware) objects passed.
 - ▶ **Extensibility**: any open-API system (i.e. with API CDs) can play.
 - ▶ **Flexibility**: full peer-to-peer possibilities. (future: service discovery)

But we have to develop a whole new framework! (Review 1 ~ Proof of Concept)
- ▶ **Review Period2: State of WP6 (MitM) Integration**
 - ▶ Developed mathematical use-cases (what do researchers want to do)
 - ▶ Extended middleware, grown MitM ontology, collected alignments
 - ▶ Jupyter integration into MathHub.info
- ▶ **Plan for Review Period 3: Extend to external use-cases users** (scale and deploy publically)
- ▶ **overall pattern**: design – prototype – scale



References I

