

# Unifying Mathematical Systems with



Georg Muntingh

Center of Mathematics for Applications, Oslo

October 1<sup>st</sup> 2007

Computers increasingly support mathematics

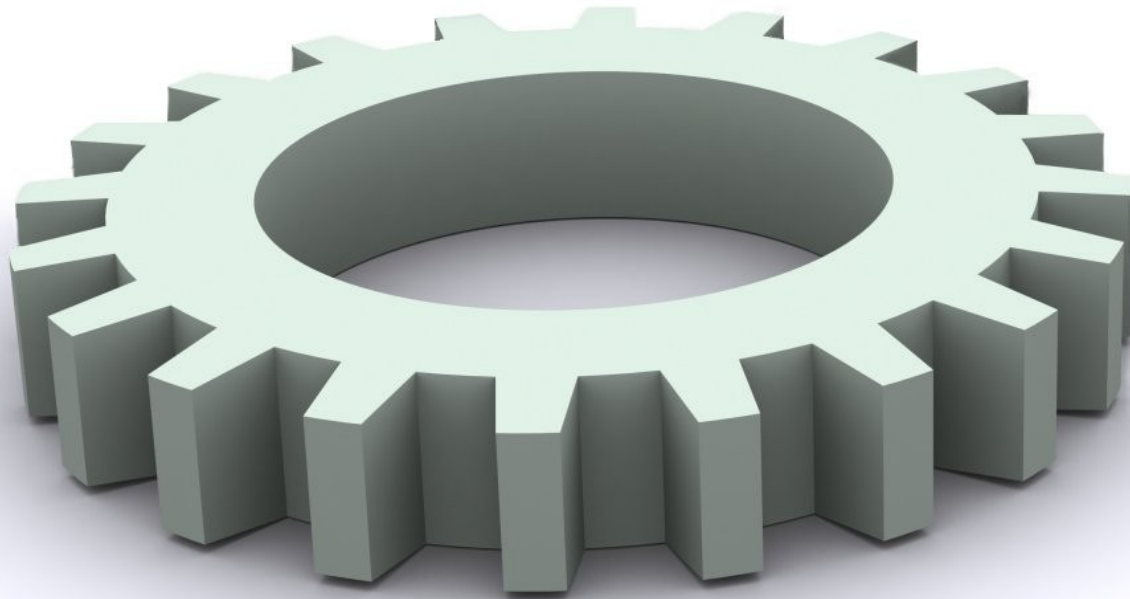


# Mathematicians have **too many** mathematical systems to choose from

4ti2 **Axiom** CVXOPT  
EC **GAP** genus2reduction Gfan  
Givaro GMP GMPY GSL LinBox  
**KASH/KANT** Lcalc Lie **Macaulay2**  
**Magma** **Maple**  
**Mathematica**  
**Matlab** **Maxima** **MuPAD**  
MWRANK MPFI MPFR NetworkX NTL  
Numpy NZMATH  
**Octave** PALP **Pari/GP** polymake  
PyCrypto Qsieve RealLib **REDUCE**  
SciPy **Singular** SYMPOW



Many systems need to **reinvent the wheel**



# Many systems need to **reinvent the wheel**

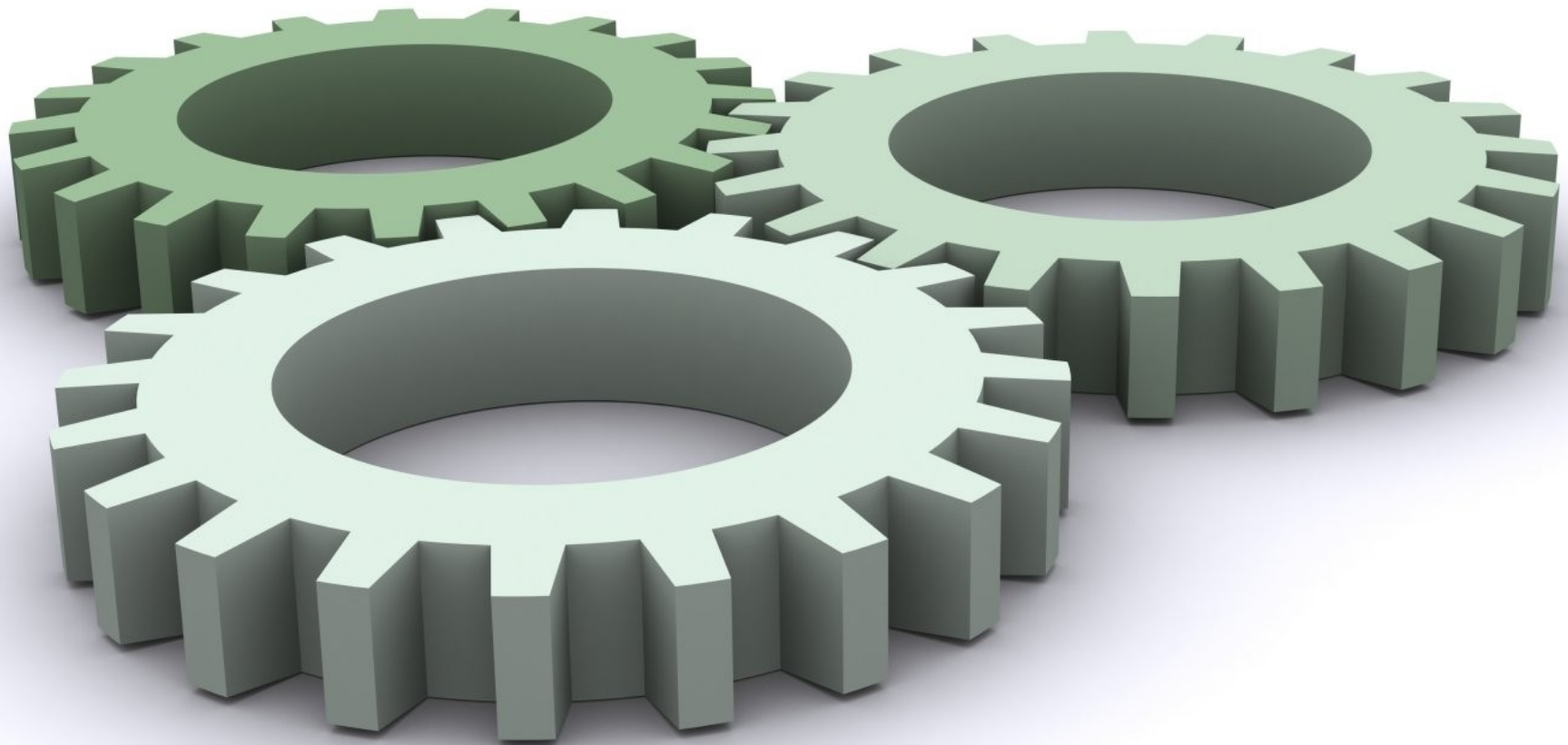
“

*Reproducing and redistribution of Maple code is a violation of the license agreement. [...] Without the express written permission of Maplesoft, Licensee shall not, and shall not permit any Third Party to: (a) reproduce, transmit, modify, adapt, translate or **create any derivative work** of, any part of the Software [...] (b) reverse engineer, disassemble, or decompile the Software, create derivative works based on the Software, or otherwise **attempt to gain access to its method** of operation or source;*

”

*Sincerely, Maplesoft Technical Support*

These systems **should cooperate**



Let's build a layer on top of these systems  
that **connects** them: **SAGE**



**Why?**

SAGE connects  
mathematical  
communities



**Free software** keeps SAGE **open**  
for anyone, forever



# **Free software** keeps SAGE **open** for anyone, forever

Freedom **0**: to run

Freedom **1**: to study and adapt

Freedom **2**: to redistribute copies

Freedom **3**: to improve and release

Anybody can use SAGE and **contribute**



Nobody can **run away** with your code



**Introspection** allows users to find out how each procedure works



**Introspection** allows users to find out how each procedure works

*Can we expect somebody to believe a result of a program that he is not allowed to see? — Joachim Neubüser, founder of GAP*





# The web notebook simplifies **sharing, collaborating** and **learning**

SAGE-Presentation1 (SAGE) - Mozilla Firefox

File Edit View History Bookmarks Tools Help

SAGE Notebook

georgm | [Home](#) | [Published](#) | [Log](#) | [Help](#) | [Sign out](#)

**SAGE-Presentation1**  
last edited on September 26, 2007 02:20 AM by georgm

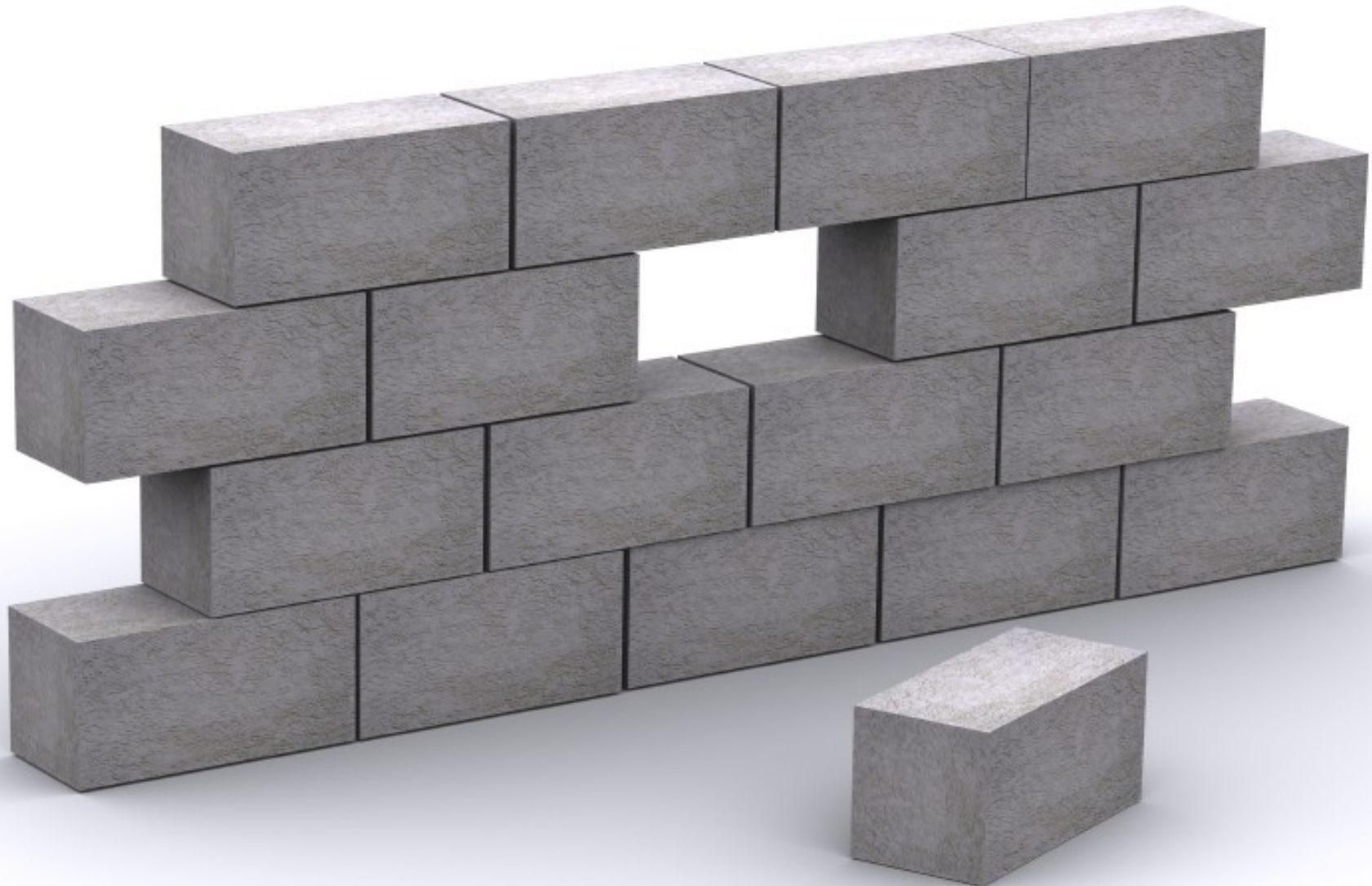
Save Save & close Discard changes

File... Action... Data... sage Print Use Edit Text Revisions Share Publish

```
E = EllipticCurve('389a')  
show(plot(E))
```

jsMath

The web interface **lowers the barrier** for participating



# The notebook encourages **sharing**



Active Worksheets	Owner / Collaborators	Last Edited
<b>ToricVarieties</b>	georgm <a href="#">Share now</a> <a href="#">(published)</a>	3 days ago by georgm
<b>SAGE-Presentation1</b>	georgm / meertn <a href="#">Add or Delete</a>	1 day ago by georgm
<b>(active) SAGE-Presentation2</b>	georgm / marian <a href="#">Add or Delete</a>	1 day ago by georgm
<b>SAGE-Presentation3</b>	georgm / meertn <a href="#">Add or Delete</a>	1 day ago by georgm

Interactive tutorials encourage  
**active learning**



**Why?**



SAGE enables us to  
**select, combine**  
and **create**

You **select** the implementation you need



You find the right function through  
(**command completion**), ( ) and ( )

```
E = EllipticCurve('389a')
```

```
E.
```

E.CPS_height_bound	E.dimension
E.Hom	<b>E.discriminant</b>
E.L1_vanishes	E.division_polynomial
E.L_ratio	E.divisor
E.Lambda	E.divisor_group
E.Lseries	E.divisor_of_function
E.Lseries_at1	E.dump
E.Lseries_deriv_at1	E.dumps
E.Lseries_dokchitser	E.eval_modular_form

You find the right function through ( ), (**help**) and ( )

```
E = EllipticCurve('389a')
```

```
E.discriminant?
```

```
File:      /usr/local/sage-2.6/local/lib/python2.5/site-pa
Type:      <type 'instancemethod'>
Definition: E.discriminant()
Docstring:

    Returns the discriminant of this elliptic curve.

EXAMPLES:
sage: E = EllipticCurve([0,0,1,-1,0])
sage: E.discriminant()
37
```

You find the right function through  
( ), ( ) and (**source**)

```
E = EllipticCurve('389a')
```

```
E.discriminant??
```

```
File: /usr/local/sage-2.6/local/lib/python2.5/site-packa  
Source Code (starting at line 779):
```

```
def discriminant(self):
```

```
    """
```

```
    Returns the discriminant of this elliptic curve.
```

```
EXAMPLES:
```

```
sage: E = EllipticCurve([0,0,1,-1,0])
```

```
sage: E.discriminant()
```

```
37
```

You can choose between  
a **fast** and a **provably correct** algorithm

You can choose between  
a **fast** and a **provably correct** algorithm

factor?

```
File:      /usr/local/sage-2.6/local/lib/python2.5/site-packages.  
Type:     <type 'function'>  
Definition: factor(n, proof=None, int_=False, algorithm='pari', v  
Docstring:  
  
Returns the factorization of the integer n as a sorted list of  
tuples (p,e).  
  
INPUT:  
  n -- an nonzero integer  
  proof -- bool or None (default: None)
```

You can choose between **different implementations** from different systems

# You can choose between **different implementations** from different systems

```
bernoulli?
```

```
File:      /usr/local/sage-2.6/local/lib/python2.5/site-package
Type:      <type 'function'>
Definition: bernoulli(n, algorithm='pari')
Docstring:

Return the n-th Bernoulli number, as a rational number.

INPUT:
  n -- an integer
  algorithm:
    'pari' (default) use the PARI C library, which is
            by *far* the fastest.
    'gap'  use GAP
    'gp'   use PARI/GP interpreter
    'magma' use MAGMA (optional)
    'python' use pure Python implementation
```



SAGE combines Magma, Macauley2, Kash,  
MuPad, GAP, Pari/GP, Singular, ...



Systems with a command line interface are  
opened in a **pseudo terminal**

# Systems with a command line interface are opened in a **pseudo terminal**

---

```
x = gp('9 + 6')
```

---

```
x.type()
```

---

```
<class 'sage.interfaces.gp.GpElement'>
```

---

```
x
```

---

```
15
```

---

```
x.name()
```

---

```
'sage[1]'
```

---

```
x.factor()
```

---

```
[3, 1; 5, 1]
```

---

```
factor(15)
```

---

```
3 * 5
```

Libraries are wrapped and compiled



**Automated testing** limits chaotic behavior



You can create your own functionality to  
**fill in the gap**



SAGE programmers use a fusion of **interpreted** Python and **compiled** Cython



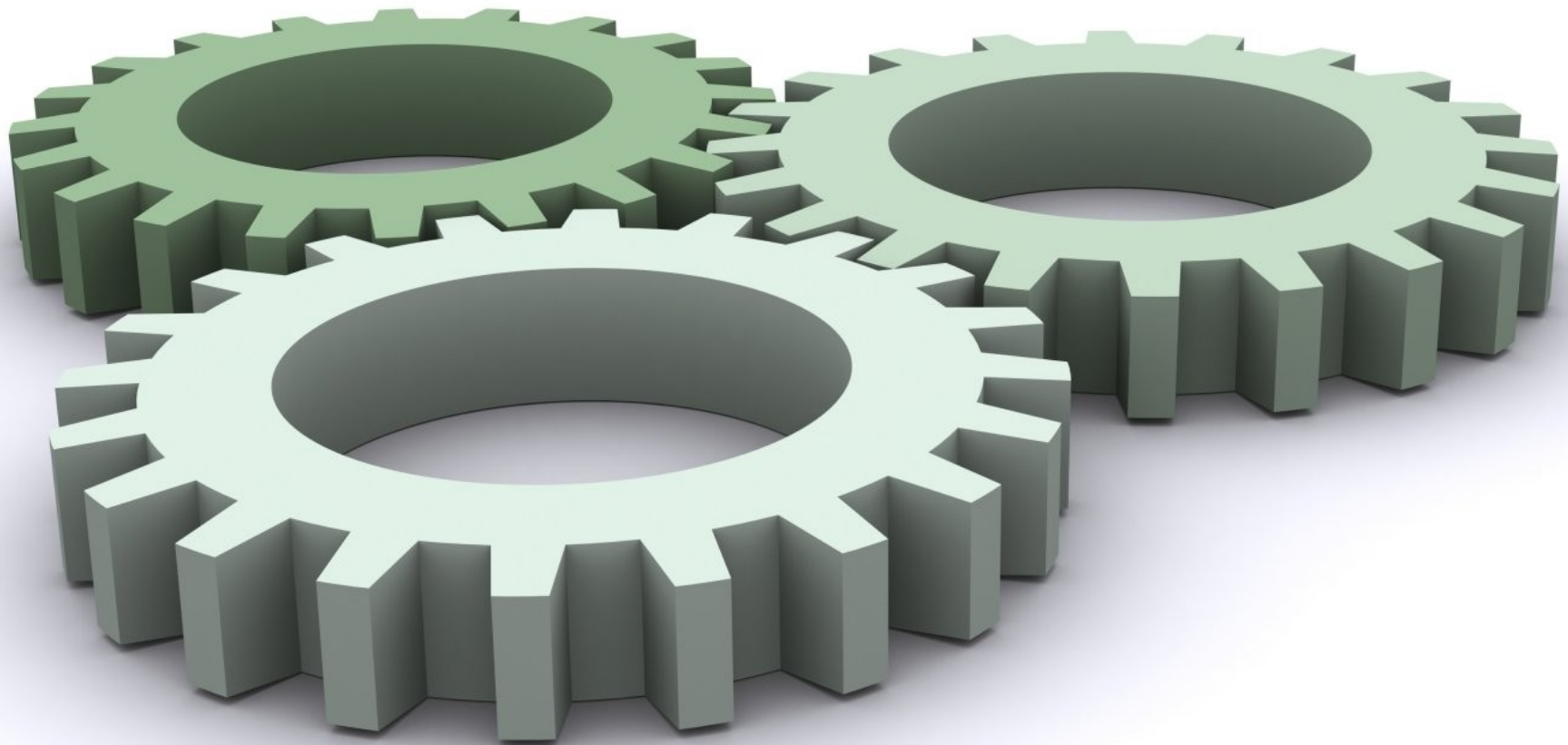
The Mercurial revisions control system  
makes it easy to experiment and contribute

The image shows the letters 'Hg' in a highly stylized, three-dimensional font. The letters are rendered in a dark grey or black color with a metallic sheen, featuring highlights and shadows that give them a sense of depth and volume. The 'H' is a classic serif style, while the 'g' is a lowercase, cursive-style letter with a prominent loop and a tail that curves downwards. The overall appearance is that of a polished, embossed metal or plastic surface.

The community has an active, helpful core



Systems that reinvent the wheel  
**should cooperate**

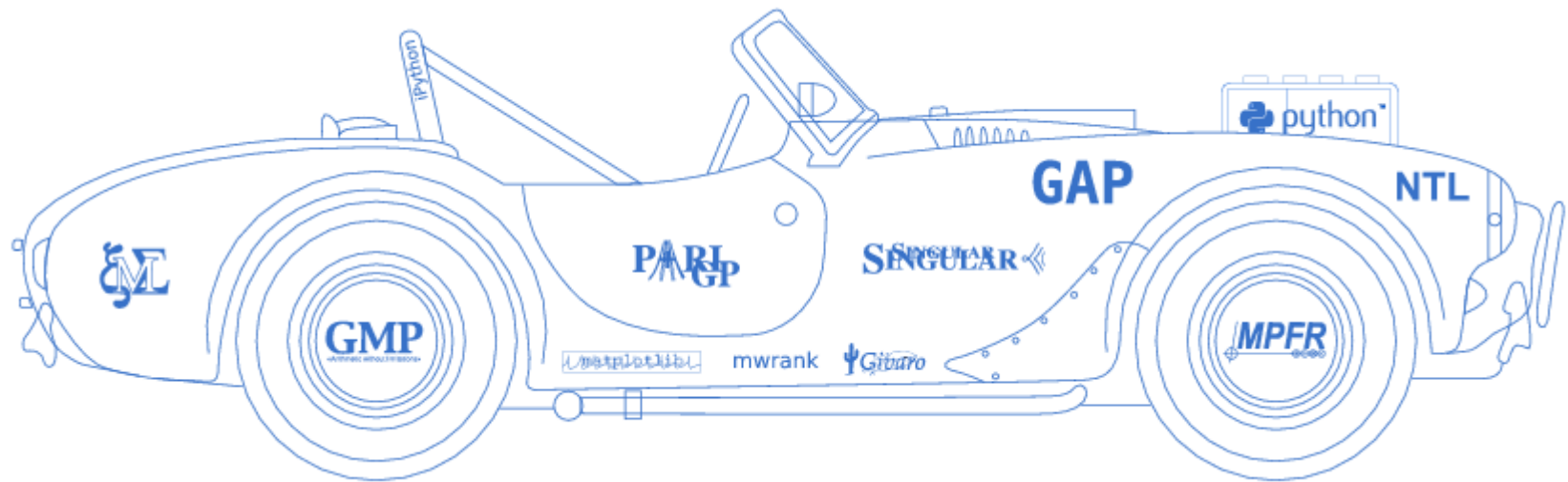


We need to build a layer on top of these systems that **connects** them: **SAGE**



SAGE can achieve its goals!





SAGE does not reinvent the wheel,  
but builds the car [8]

# References

- [1] David Joyner, *OSCAS*, <http://www.sagemath.org:9001/OSCAS>
- [2] Pierrick Gaudry, Alexander Kruppa, and Paul Zimmermann. *A GMP-based Implementation of Schönhage-Strassen's Large Integer Multiplication Algorithm*, <http://www.loria.fr/~gaudry/publis/fft.pdf>
- [3] William Stein. *Talk by Eric Raymond*, SAGE-development mailing list, [http://groups.google.com/group/sage-devel/browse\\_thread/thread/a8733b5db5e51738](http://groups.google.com/group/sage-devel/browse_thread/thread/a8733b5db5e51738)
- [4] William Stein. *SAGE: Software for Algebra and Geometry Experimentation*, <http://sagemath.org/talks/2007-01-11-uw-undergrads/sage.pdf>
- [5] Eric S. Raymond. *The Cathedral and the Bazaar*, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s04.html>
- [6] William Stein and David Joyner. *The SAGE Programming Guide, Interpreted and Compiled Code*, <http://sagemath.org/doc/html/prog/node35.html>
- [7] Mohamed Bendame and Darren McIntyre, *New generation of math software from Maple*, <http://www.youtube.com/watch?v=1I2QVRIf10A>
- [8] Image created by Martin Albrecht.