

OpenMath in SCIENCE: SCSCP and POPCORN

Peter Horn

Dan Roozmond

8th International Conference on
Mathematical Knowledge Management
July 11th, 2009

www.symbolic-computation.org

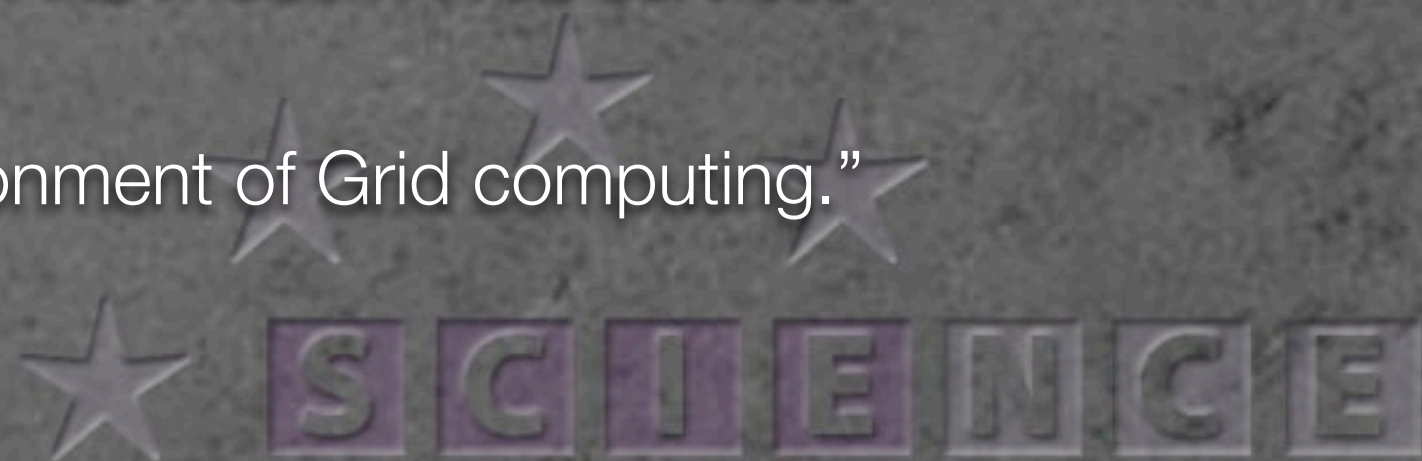


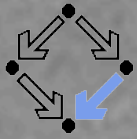


- European “Framework 6” programme,
- Started April 2006, runs for 5 years,
- Main purpose:

“to unite the European community of researchers in, and users of, symbolic computation. SCIENCE aims to promote the development of new software that is

- made more efficient by sharing components and expertise;
- made more interoperable in the modern Web services environment; and
- ready for the coming environment of Grid computing.”





U N I K A S S E L
V E R S I T Ä T



TU/e Technische Universiteit
Eindhoven
University of Technology



Maplesoft



- The Centre for Interdisciplinary Research in Computational Algebra,
University of St Andrews, Scotland
- Research Institute for Symbolic Computation,
Linz, Austria
- Ecole Polytechnique,
Centre National de la Recherche Scientifique, Paris, France
- Computational Mathematics Group,
Universität Kassel, Germany
- The KANT group,
Technische Universität Berlin, Germany
- Discrete Algebra and Geometry group,
Technische Universiteit Eindhoven, Netherlands
- Institute e-Austria Timisoara,
Romania
- Maplesoft,
Waterloo, Canada
- The Dependable Systems Research Group,
Heriot-Watt University, Edinburgh, Scotland

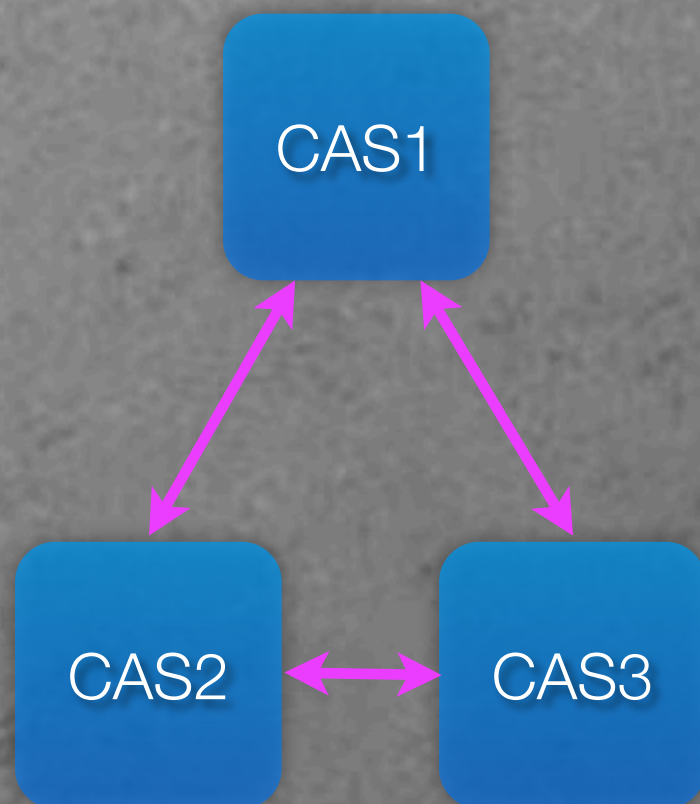


What's it all about?



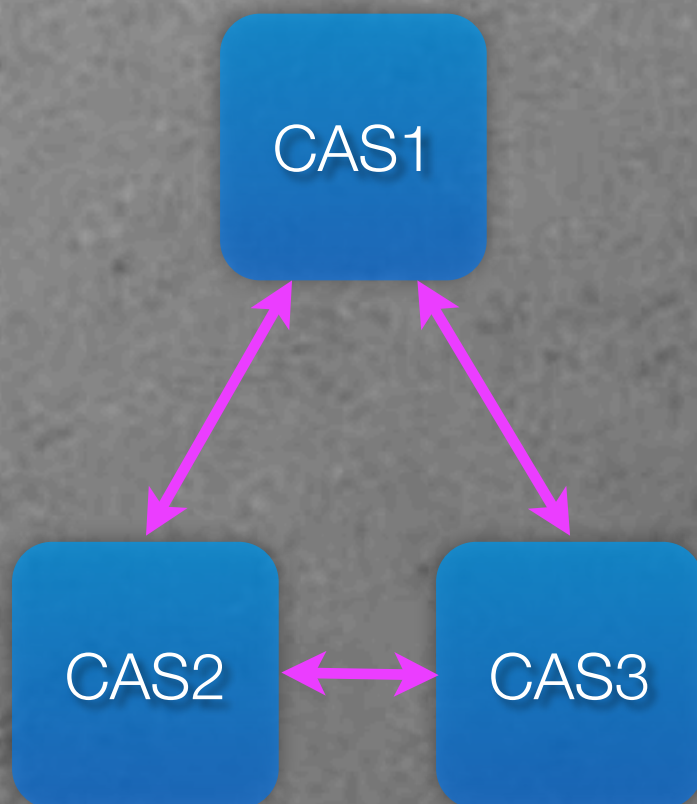
What's it all about?

Directly linking
Symbolic
Software

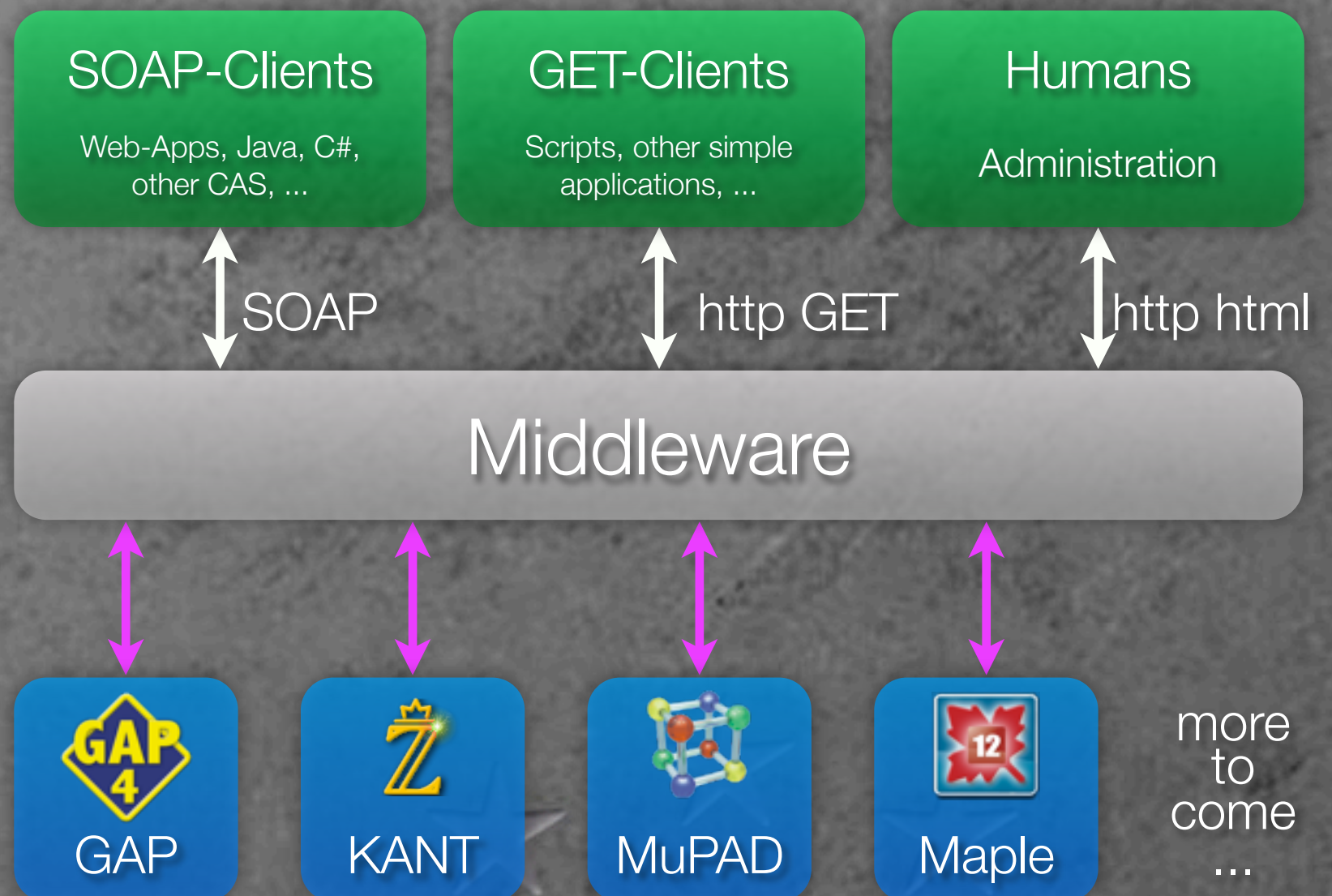


What's it all about?

Directly linking
Symbolic
Software



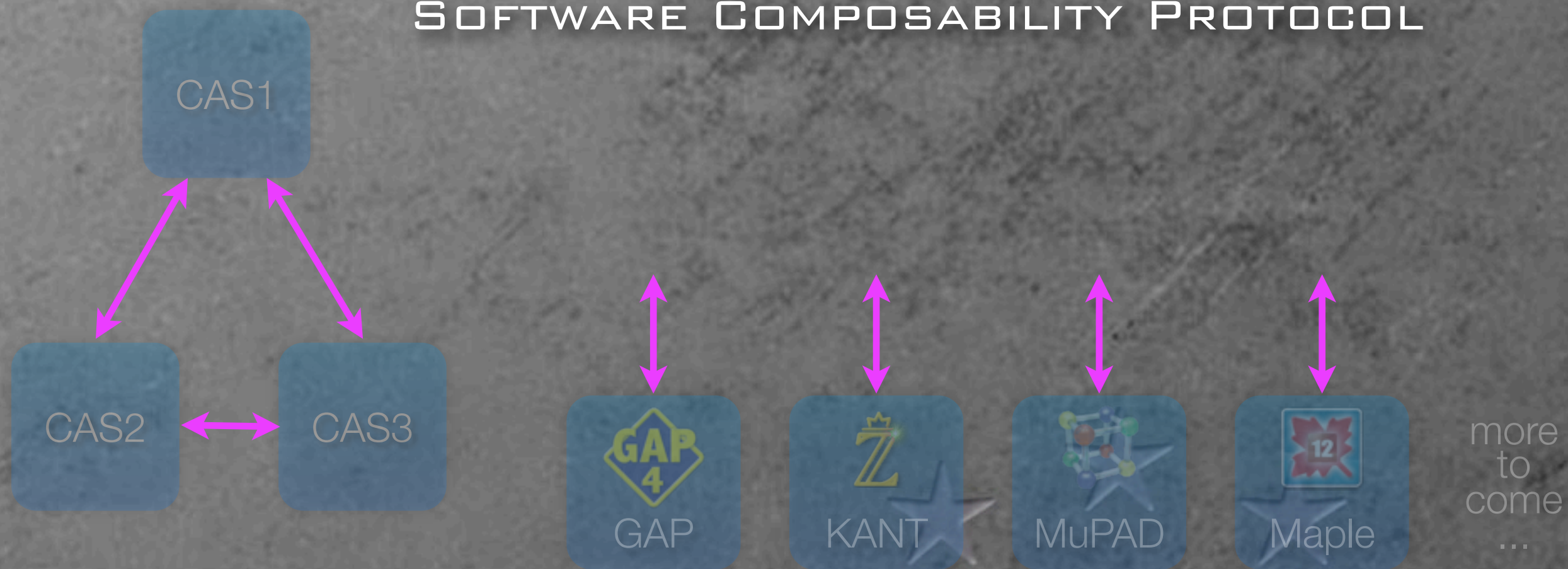
Link Symbolic Software to
other Systems



What's it all about?

SCSCP

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL



★ SCIENCE

SCSCP

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

- ✦ Protocol for communication between CASes
- ✦ OpenMath based
- ✦ Lightweight, simple sockets
- ✦ Basis for symbolic computation on Clusters and Grids
- ✦ Described in the “SCSCP standard (version 1.3)”, and `scscp1` and `scscp2` Content Dictionaries

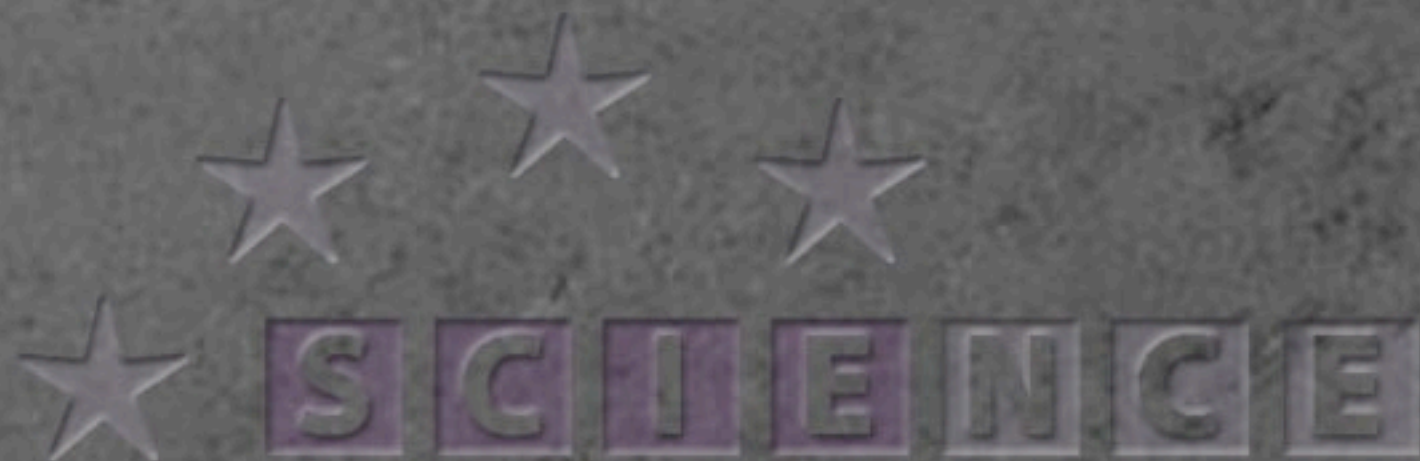
★ ★ ★ ★ ★
★ SCIENCE

SCSCPT

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

Possible applications

- Cross-program: Software A can do things B can't,

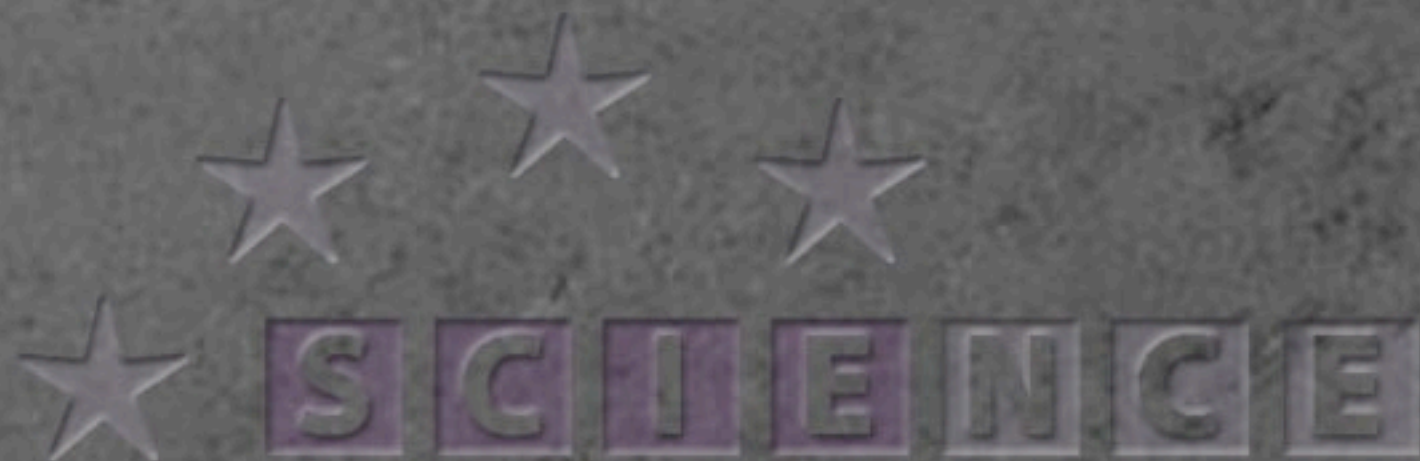


SCSCPT

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

Possible applications

- ✦ Cross-program: Software A can do things B can't,
- ✦ Cross-program: A can do things much faster than B,

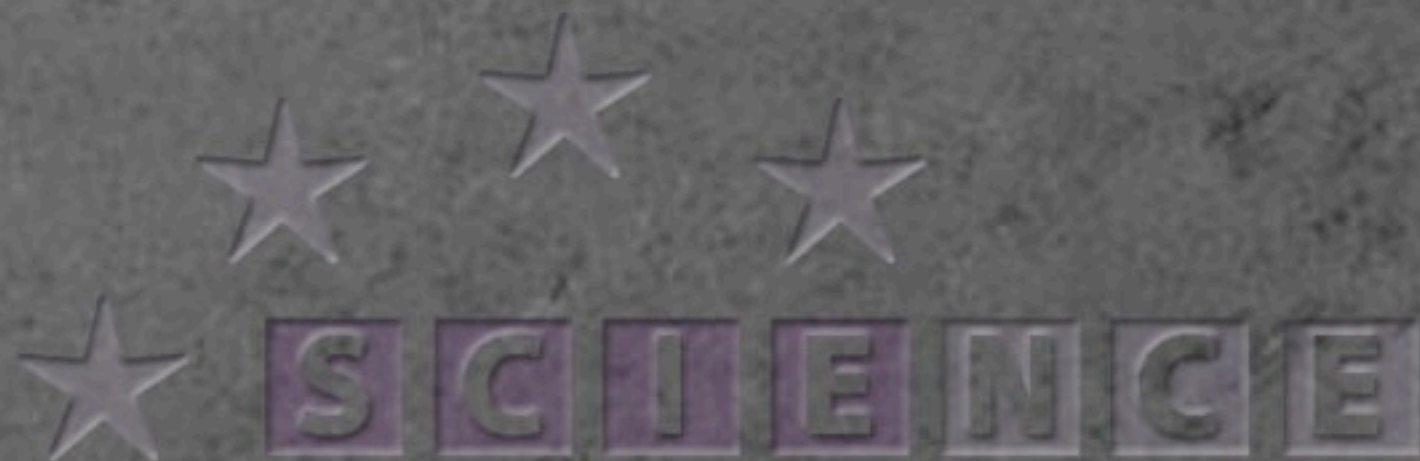


SCSCPT

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

Possible applications

- ✦ Cross-program: Software A can do things B can't,
- ✦ Cross-program: A can do things much faster than B,
- ✦ Cross-platform: A is only available on *nix,

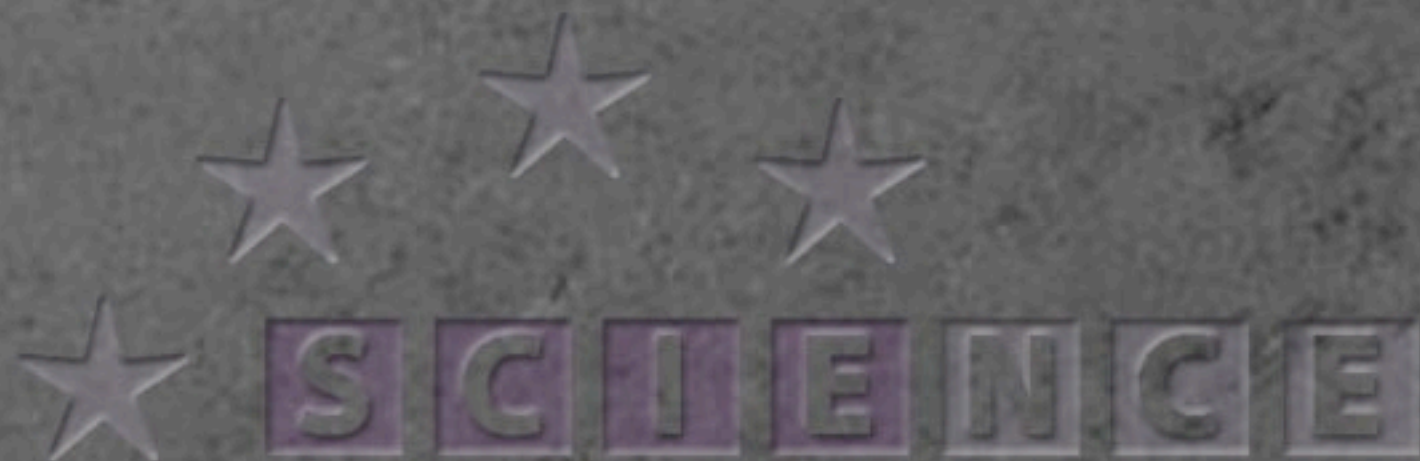


SCSCPT

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

Possible applications

- ✦ Cross-program: Software A can do things B can't,
- ✦ Cross-program: A can do things much faster than B,
- ✦ Cross-platform: A is only available on *nix,
- ✦ A is a pain to compile and install,

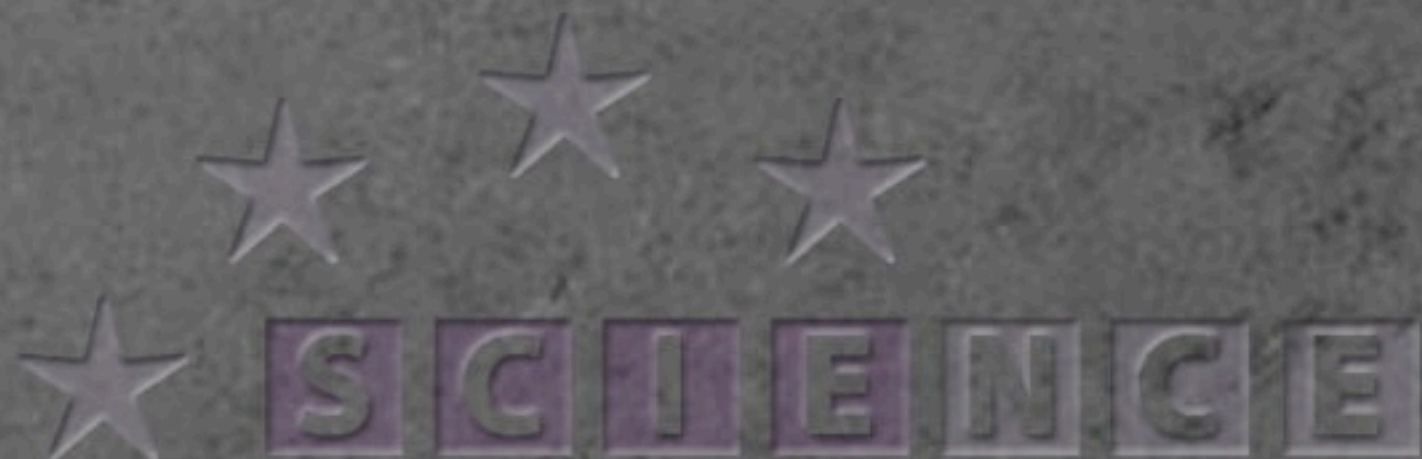


SCSCPT

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

Possible applications

- ✦ Cross-program: Software A can do things B can't,
- ✦ Cross-program: A can do things much faster than B,
- ✦ Cross-platform: A is only available on *nix,
- ✦ A is a pain to compile and install,
- ✦ etc...





- ✦ Standard for representing mathematical objects
- ✦ Focused on semantics
- ✦ Extensible

1+2	<pre><OMOBJ><OMA><OMS cd="arith1" name="plus"/> <OMI>1</OMI><OMI>2</OMI></OMA></OMOBJ></pre>
$x \rightarrow x \cdot \pi$	<pre><OMOBJ><OMBIND><OMS cd="fns1" name="lambda"/> <OMBVAR><OMV name="x"/></OMBVAR> <OMA><OMS cd="arith1" name="times"/><OMV name="x"/> <OMS cd="nums2" name="pi"/> </OMA></OMBIND></OMOBJ></pre>





- ✦ Very simple, only 12 language elements:
 - ✦ Integers, Floats, Strings, Variables, References, Symbols
 - ✦ Binary, Foreign,
 - ✦ Application, Binding, Error, Attribution
- ✦ All semantics is the Symbols, described by “Content Dictionaries” (CDs)





CD/Symbol Example

OpenMath Content Dictionary: arith1

Canonical URL:

<http://www.openmath.org/cd/arith1.oed>

CD Base:

<http://www.openmath.org/cd>

CD File:

[arith1.oed](#)

CD as XML Encoded OpenMath:

[arith1.omcd](#)

Defines:

[abs](#), [divide](#), [gcd](#), [lcm](#), [minus](#), [plus](#), [power](#), [product](#), [root](#), [sum](#), [times](#), [unary minus](#)

Date:

2004-03-30

Version:

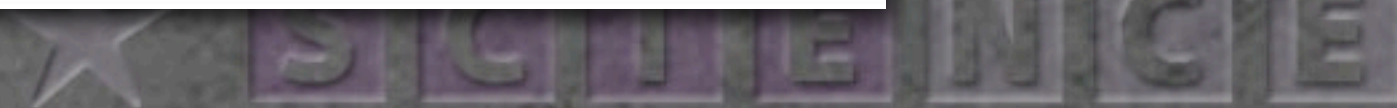
3

Review Date:

2006-03-30

Status:

official





CD/Symbol Example

OpenMath Content Dictionary: arith1

plus

Role:

application

Description:

The symbol representing an n-ary commutative function plus.

Commented Mathematical property (CMP):

for all $a, b \mid a + b = b + a$

Formal Mathematical property (FMP):

xml

prefix

mathml

$\forall a, b . a + b = b + a$

Signatures:

[sts](#)



- ✦ OpenMath allows for different representations:
 - ✦ XML
 - ✦ OpenMath Binary

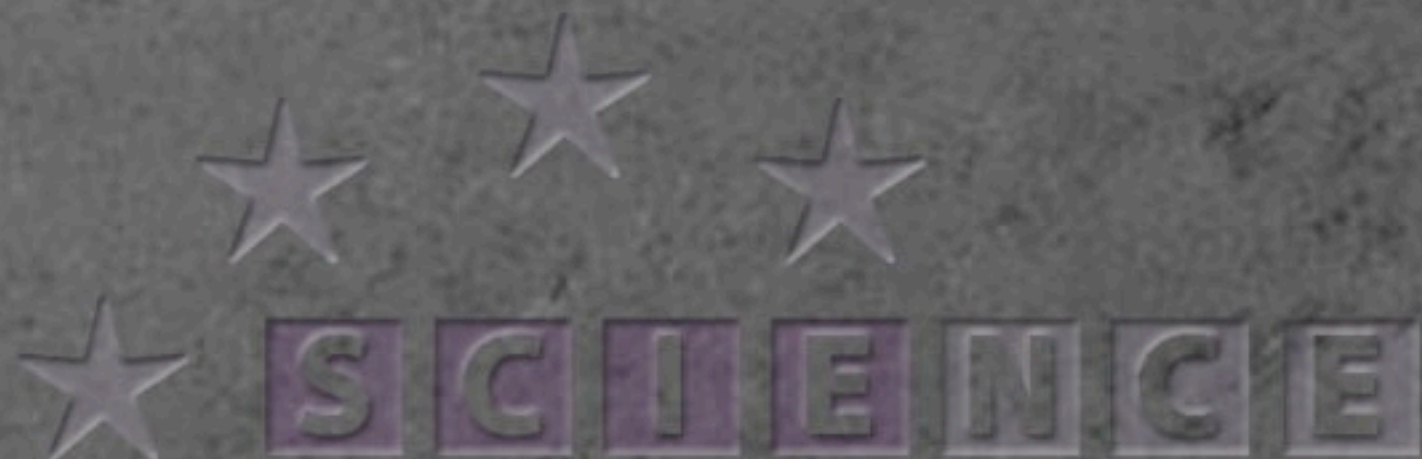




- ✦ OpenMath allows for different representations:
 - ✦ XML
 - ✦ OpenMath Binary

1+2

$x \rightarrow x \cdot \pi$





- ✦ OpenMath allows for different representations:
 - ✦ XML
 - ✦ OpenMath Binary

 $1+2$

```
<OMOBJ><OMA><OMS cd="arith1" name="plus"/>
<OMI>1</OMI><OMI>2</OMI></OMA></OMOBJ>
```

 $x \rightarrow x \cdot \pi$

```
<OMOBJ><OMBIND><OMS cd="fns1"
name="lambda"/><OMBVAR><OMV name="x"/></
OMBVAR><OMA><OMS cd="arith1" name="times"/
><OMV name="x"/><OMS cd="nums2" name="pi"/> </
OMA></OMBIND></OMOBJ>
```





- ✦ OpenMath allows for different representations:
 - ✦ XML
 - ✦ OpenMath Binary

 $1+2$

```
<OMOBJ><OMA><OMS cd="arith1" name="plus"/>
<OMI>1</OMI><OMI>2</OMI></OMA></OMOBJ>
```

```
18 10 08 06 04 61 72 69 74
68 31 70 6c 75 73 01 01 01
02 11 19
```

 $x \rightarrow x \cdot \pi$

```
<OMOBJ><OMBIND><OMS cd="fns1"
name="lambda"/><OMBVAR><OMV name="x"/></
OMBVAR><OMA><OMS cd="arith1" name="times"/
><OMV name="x"/><OMS cd="nums2" name="pi"/> </
OMA></OMBIND></OMOBJ>
```

```
18 1a 08 04 06 66 6e 73 31
6c 61 6d 62 64 61 1c 05 01
78 1d 10 08 06 05 61 72 69
74 68 31 74 69 6d 65 73 05
01 78 08 05 02 6e 75 6d 73
31 70 69 11 1b 19
```





- ✦ OpenMath allows for different representations:
 - ✦ XML
 - ✦ OpenMath Binary

 $1+2$

```
<OMOBJ><OMA><OMS cd="arith1" name="plus"/>
<OMI>1</OMI><OMI>2</OMI></OMA></OMOBJ>
```

```
18 10 08 06 04 61 72 69 74
68 31 70 6c 75 73 01 01 01
02 11 19
```

 $x \rightarrow x \cdot \pi$

```
<OMOBJ><OMBIND><OMS cd="fns1"
name="lambda"/><OMBVAR><OMV name="x"/></
OMBVAR><OMA><OMS cd="arith1" name="times"/
><OMV name="x"/><OMS cd="nums2" name="pi"/> </
OMA></OMBIND></OMOBJ>
```

```
18 1a 08 04 06 66 6e 73 31
6c 61 6d 62 64 61 1c 05 01
78 1d 10 08 06 05 61 72 69
74 68 31 74 69 6d 65 73 05
01 78 08 05 02 6e 75 6d 73
31 70 69 11 1b 19
```

Bulky

SCIENCE



- OpenMath allows for different representations:
 - XML
 - OpenMath Binary

Unintelligible

1+2

```
<OMOBJ><OMA><OMS cd="arith1" name="plus"/>
<OMI>1</OMI><OMI>2</OMI></OMA></OMOBJ>
```

```
18 10 08 06 04 61 72 69 74
68 31 70 6c 75 73 01 01 01
02 11 19
```

$x \rightarrow x \cdot \pi$

```
<OMOBJ><OMBIND><OMS cd="fns1"
name="lambda"/><OMBVAR><OMV name="x"/></
OMBVAR><OMA><OMS cd="arith1" name="times"/
><OMV name="x"/><OMS cd="nums2" name="pi"/> </
OMA></OMBIND></OMOBJ>
```

```
18 1a 08 04 06 66 6e 73 31
6c 61 6d 62 64 61 1c 05 01
78 1d 10 08 06 05 61 72 69
74 68 31 74 69 6d 65 73 05
01 78 08 05 02 6e 75 6d 73
31 70 69 11 1b 19
```

Bulky



POPCORN

Possibly Only Practical Convenient OpenMath Replacement Notation

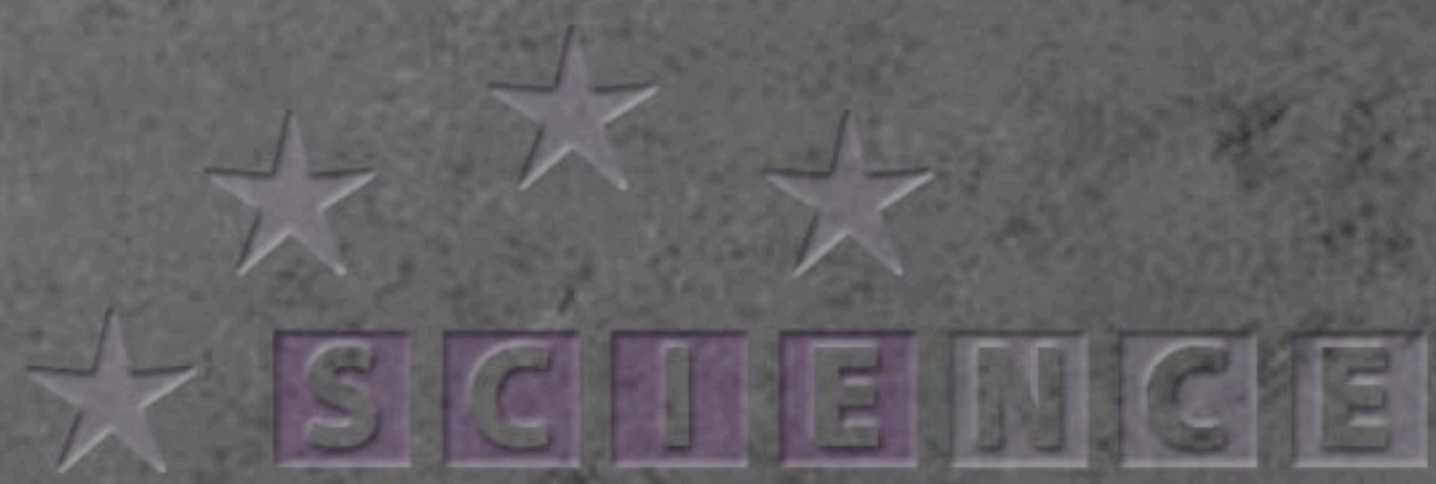
POPCORN offers an OpenMath-representation for humans:

- ✦ Integers, Floats and Strings as you expect:
18, 0.6, 2.09e3, "Hi, MKM 2009!"
- ✦ Symbols: `cdname.symbolname`
- ✦ Variables: `$name`, References: `#name`
- ✦ Application: `arith1.plus(1,2,3)`
- ✦ Binding: `fns1.lambda[$x -> $x + 1]`
- ✦ Attribution: `some.thing{aa.bb -> 1}`
- ✦ Some abbreviations and infix operators

SCIENCE

POPCORN

Possibly Only Practical Convenient OpenMath Replacement Notation

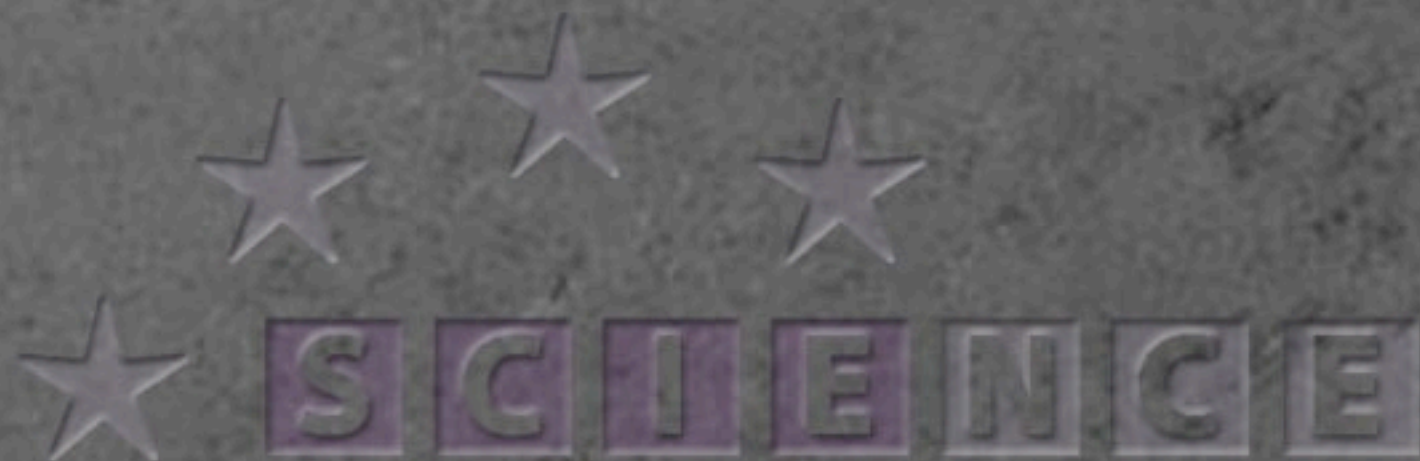


POPCORN

Possibly Only Practical Convenient OpenMath Replacement Notation

1 + 2

1+2



POPCORN

Possibly Only Practical Convenient OpenMath Replacement Notation

```
<OMA><OMS
cd="arith1"
name="plus"/
><OMI>1</
OMI><OMI>2</OMI></
OMA>
```

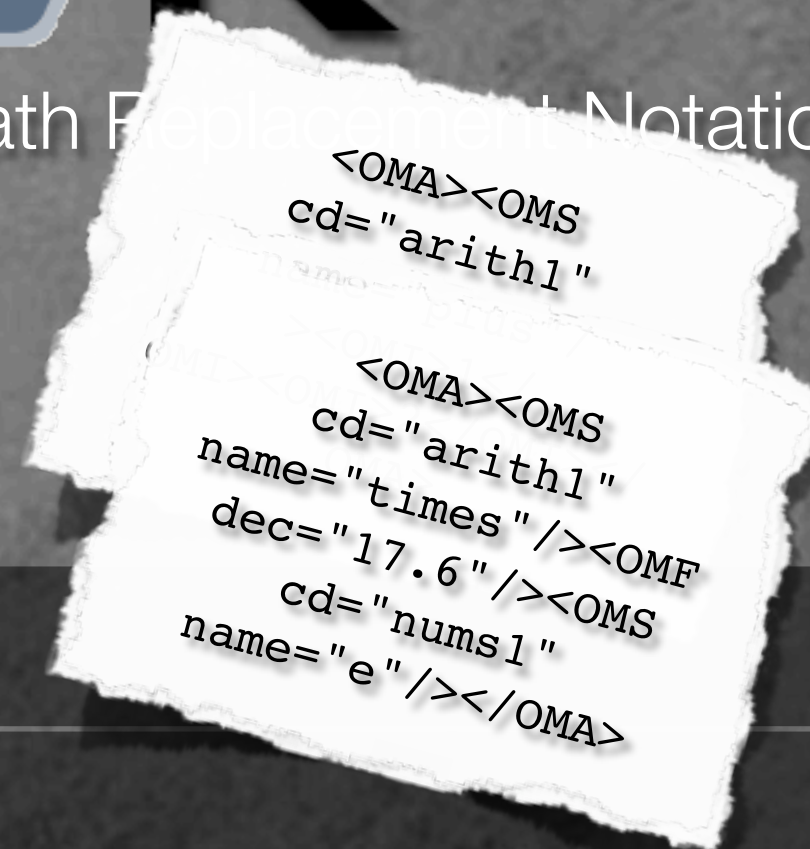
1 + 2

1+2



POPCORN

Possibly Only Practical Convenient OpenMath Representation Notation



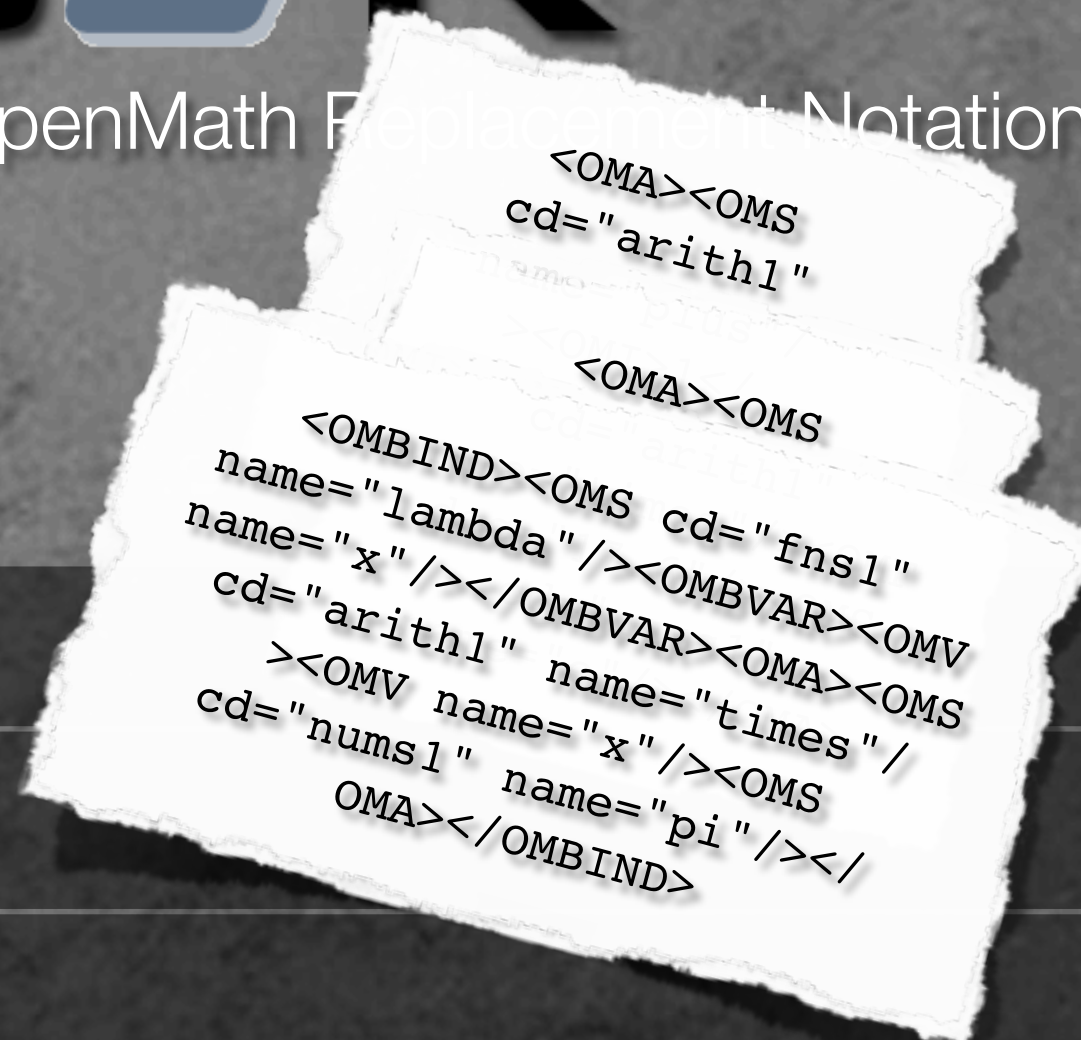
$1 + 2$	$1+2$
$17.6 \cdot e$	17.6^*e



POPCORN

Possibly Only Practical Convenient OpenMath Format for TeX Notation

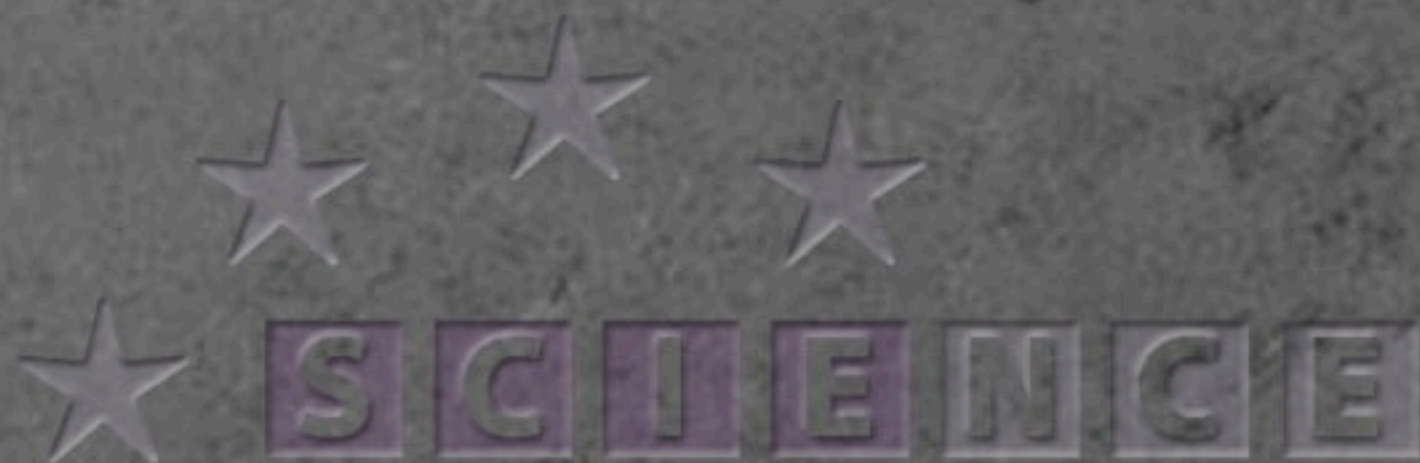
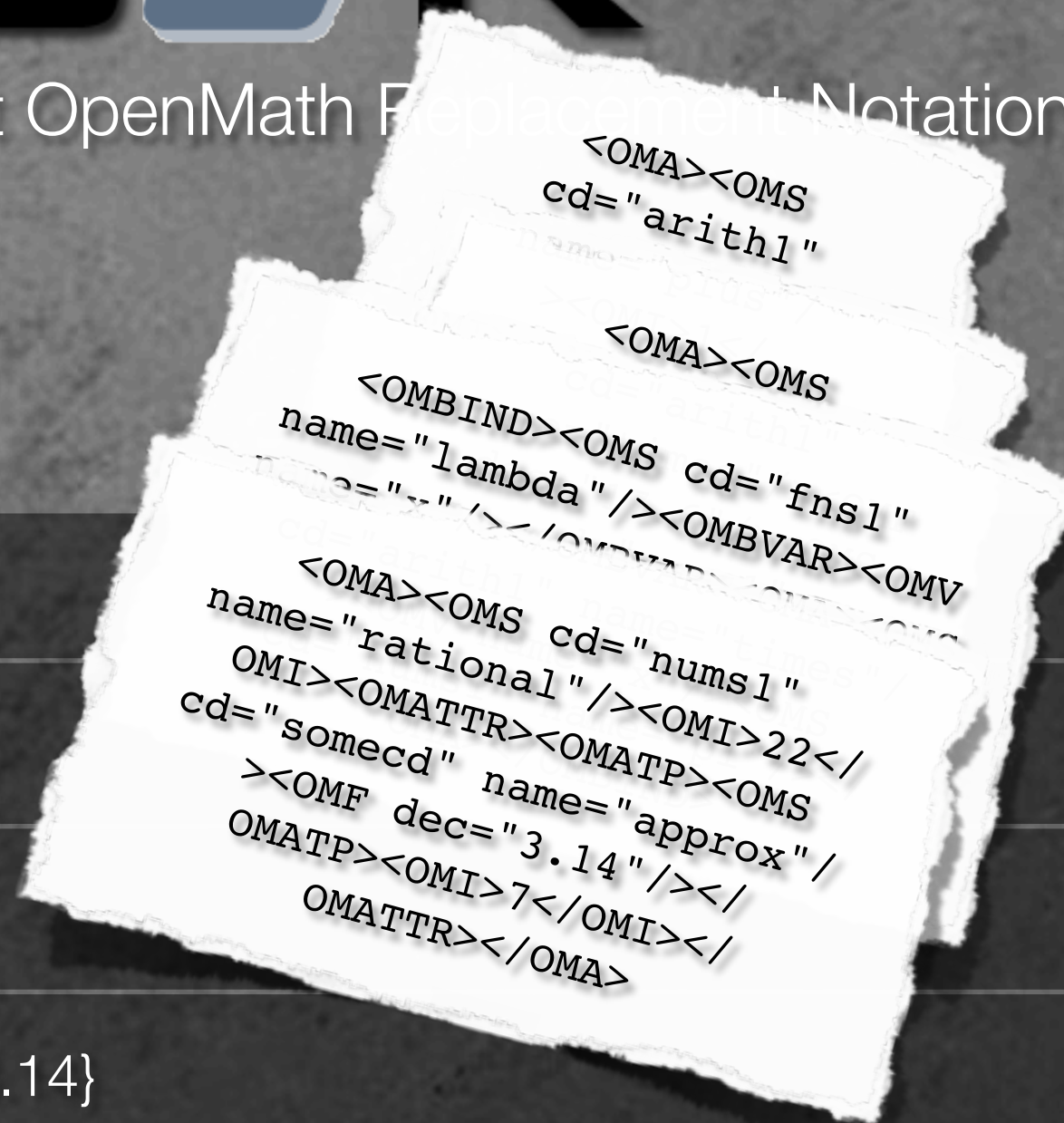
$1 + 2$	1+2
$17.6 \cdot e$	17.6*e
$x \rightarrow x \cdot \pi$	lambda[\$x -> \$x*pi]



POPCORN

Possibly Only Practical Convenient OpenMath Format Notation

$1 + 2$	1+2
$17.6 \cdot e$	17.6*e
$x \rightarrow x \cdot \pi$	lambda[\$x -> \$x*pi]
$\frac{22}{7}$	22//7{somecd.approx -> 3.14}



POPCORR

Possibly Only Practical Convenient OpenMath Format Notation

```

<OMA><OMS cd="calculus1" name="defint" /
><OMA><OMS cd="interval1" name="interval" /
><OMI>0</OMI><OMI>1</OMI></OMA><OMBIND><OMS
cd="fns1" name="lambda" /><OMBVAR><OMV name="x" /
></OMBVAR><OMA><OMS cd="arith1" name="divide" /
><OMI>1</OMI><OMA><OMS cd="arith1" name="plus" /
name="x" /><OMI>3</OMI></OMA><OMA><OMS
cd="transc1" name="sin" /><OMV name="x" /></OMA></
OMA></OMA></OMBIND></OMA>

```

22//7{somecd.approx -> 3.14}

$$\int_0^1 \frac{1}{x^3 + \sin x} dx$$

calculus1.defint(interval1.interval(0,1), lambda[\$x -> 1/(\$x^3 + sin(\$x))])

```

<OMA><OMS
cd="arith1"

```

```

<OMA><OMS

```

```

<OMBIND><OMS cd="fns1"
name="lambda" /><OMBVAR><OMV
name="x" /></OMBVAR><OMA>

```

```

<OMA><OMS cd="nums1"
="rational" /><OMI>22</
><OMATTR><OMATP><OMS
somecd" name="approx" /
<OMF dec="3.14" /></
OMATP><OMI>7</OMI></
OMATTR></OMA>

```

SCIENCE

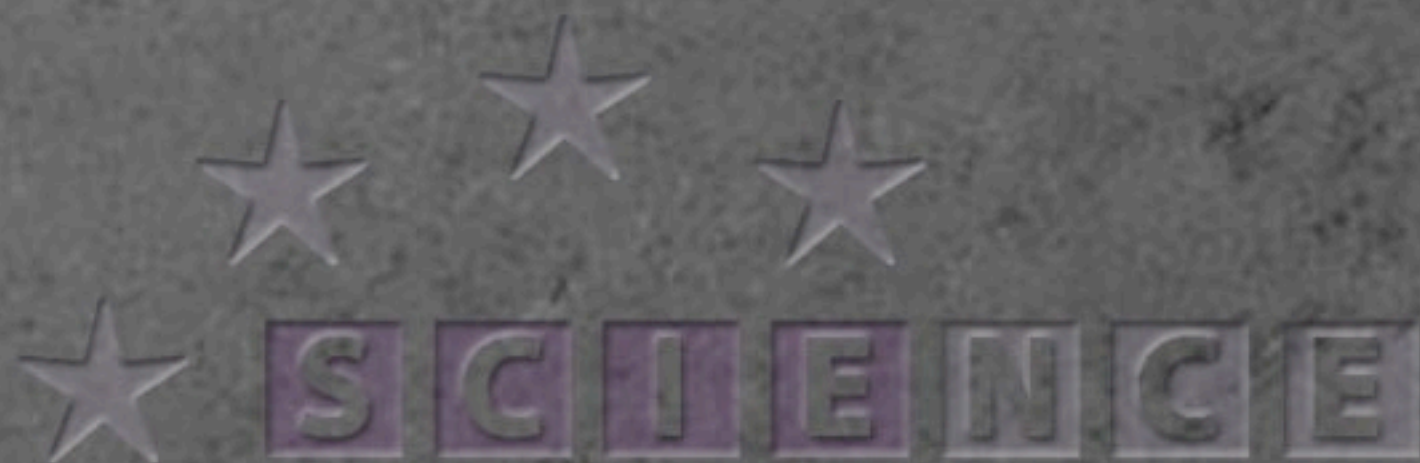
Java Libraries for OpenMath and SCSCP



Java Libraries for OpenMath and SCSCP



org.symcomp.openmath



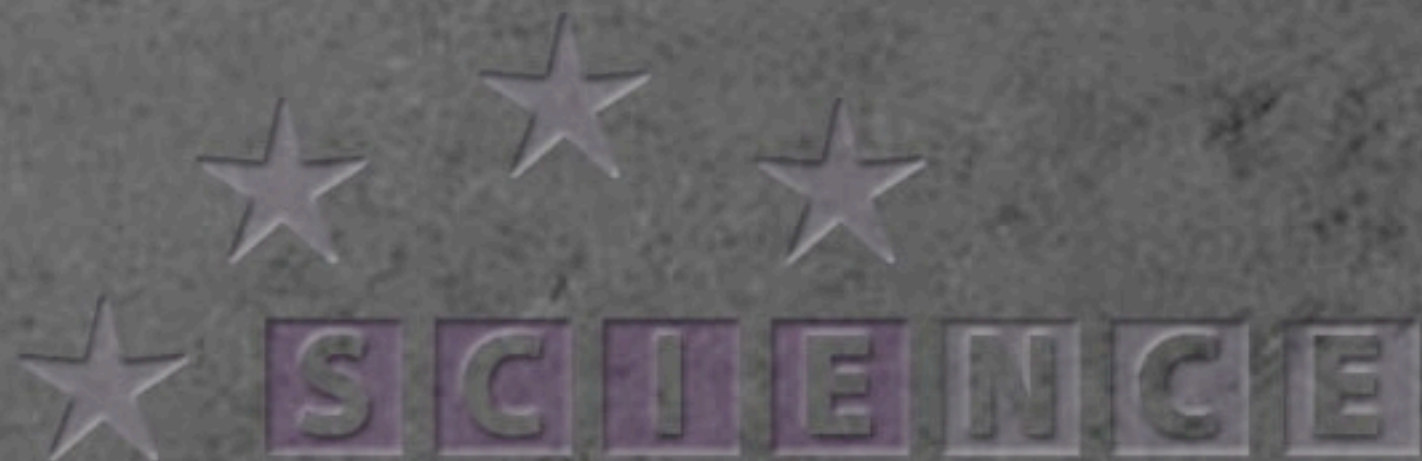
Java Libraries for OpenMath and SCSCP



org.symcomp.openmath



org.symcomp.scscp



Java Libraries for OpenMath and SCSCP

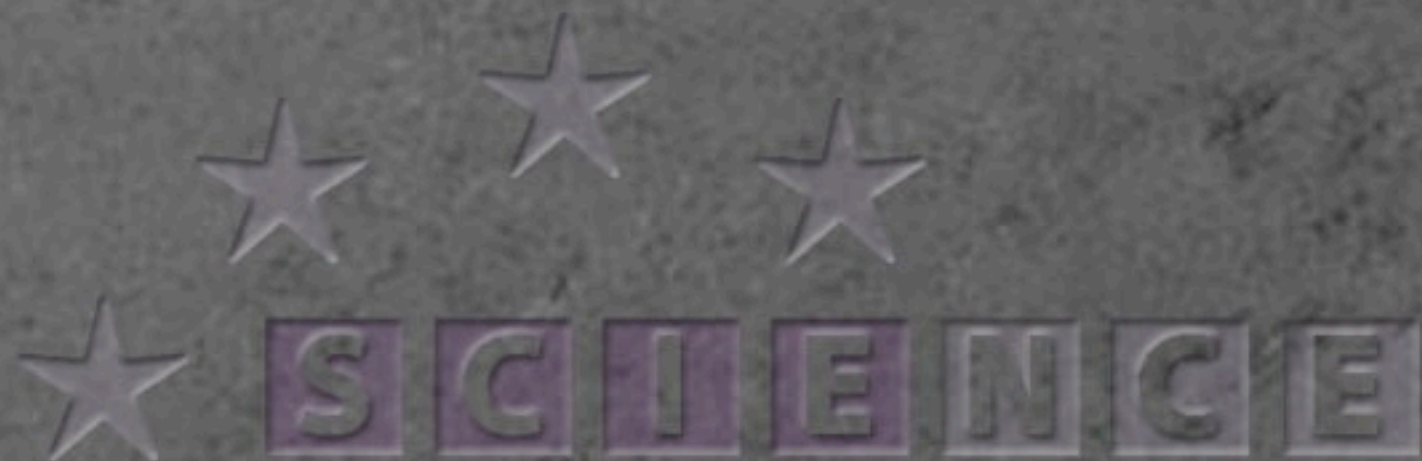


org.symcomp.openmath



org.symcomp.scscp

Standard
Libraries



Java Libraries for OpenMath and SCSCP

SCSCP enabled Java-Application

SCSCP
SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

org.symcomp.scscp



org.symcomp.openmath

Standard
Libraries



Java Libraries for OpenMath and SCSCP

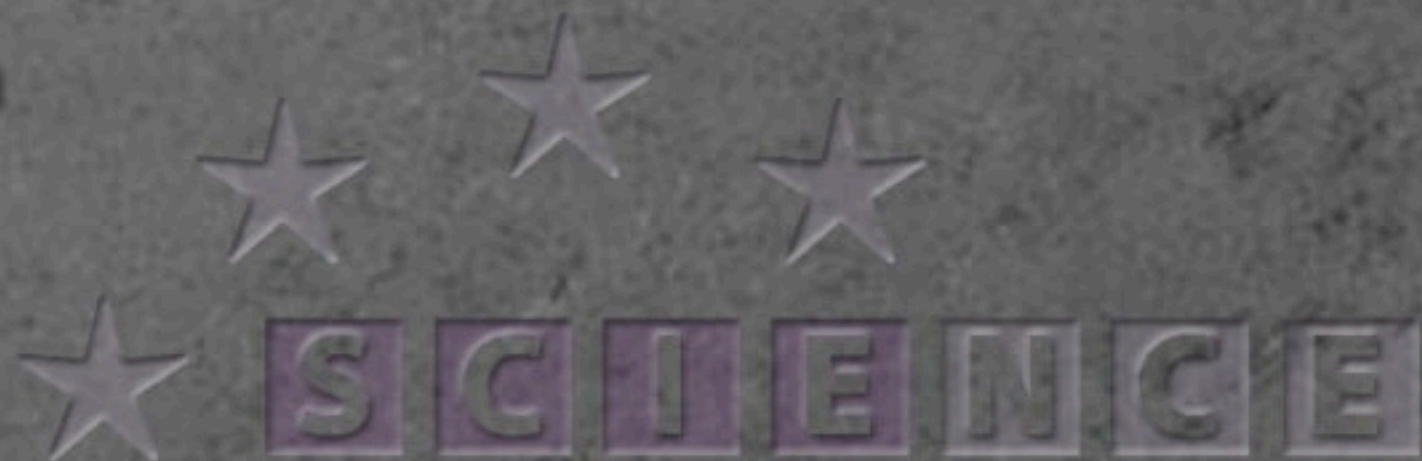


org.symcomp.openmath



org.symcomp.scscp

- ✦ Representation and Manipulation of OM
- ✦ Many convenience methods
- ✦ Reads and writes different formats: XML, Binary, POPCORN, LaTeX
- ✦ Very extensible, e.g. Custom Renderers



Java Libraries for OpenMath and SCSCP



org.symcomp.openmath

- ✦ Representation and Manipulation of OM
- ✦ Many convenience methods
- ✦ Reads and writes different formats: XML, Binary, POPCORN, LaTeX
- ✦ Very extensible, e.g. Custom Renderers



SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

org.symcomp.scscp

- ✦ Wraps all SCSCP functionality
- ✦ Turning a Java-application into a SCSCP server/client is (almost) a one-liner
- ✦ Comes with many examples to help you get started



Java Libraries for OpenMath and SCSCP



org.symcomp.openmath

- ✦ Representation and Manipulation of OM
- ✦ Many convenience methods
- ✦ Reads and writes different formats: XML, Binary, POPCORN, LaTeX
- ✦ Very extensible, e.g. Custom Renderers

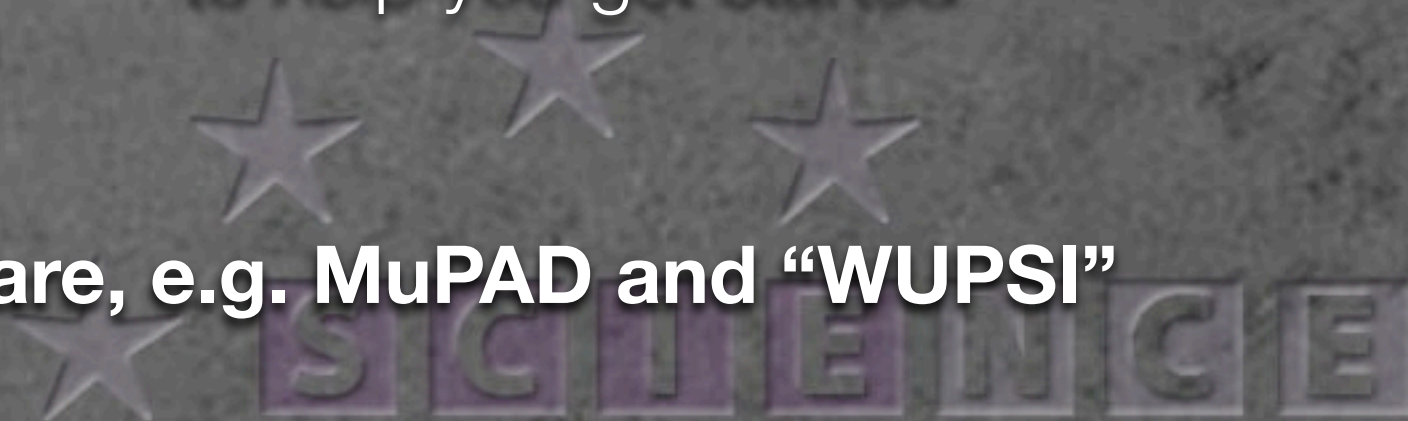
✦ **Used for our own software, e.g. MuPAD and “WUPSI”**



SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

org.symcomp.scscp

- ✦ Wraps all SCSCP functionality
- ✦ Turning a Java-application into a SCSCP server/client is (almost) a one-liner
- ✦ Comes with many examples to help you get started

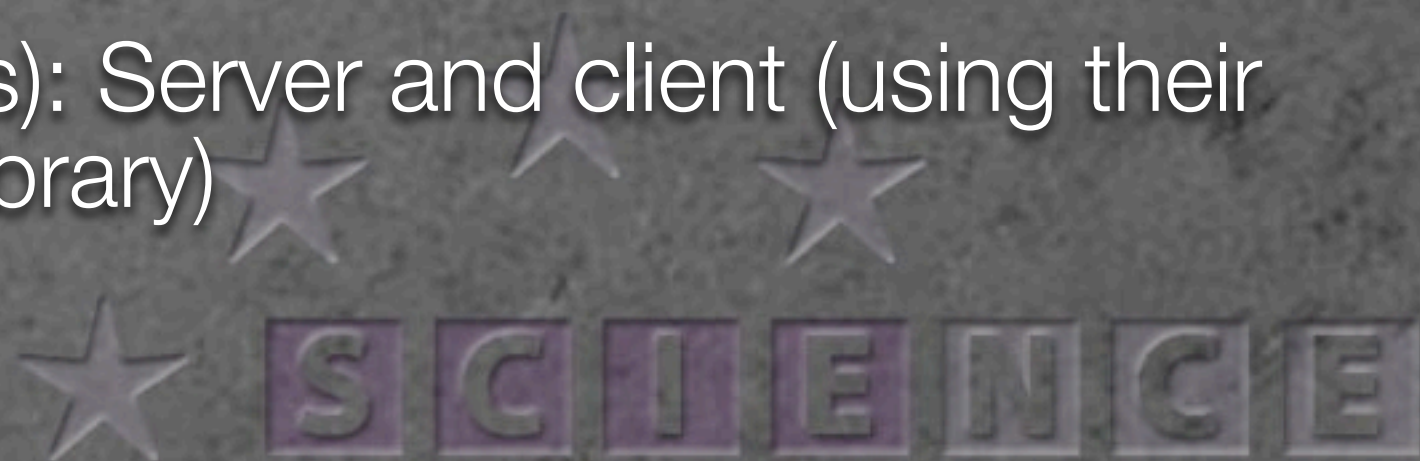


SCSCP

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

State of the systems

- ✦ GAP: Client and (single threaded) server
- ✦ KANT: Client and server
- ✦ MuPAD: Client and server (using Java libs)
- ✦ Maple: (First prototype of) client and server
- ✦ Magma: Server (using Java libs)
- ✦ TRIP (celestial mechanics): Server and client (using their own, public, SCSCP C-library)
- ✦ more to come

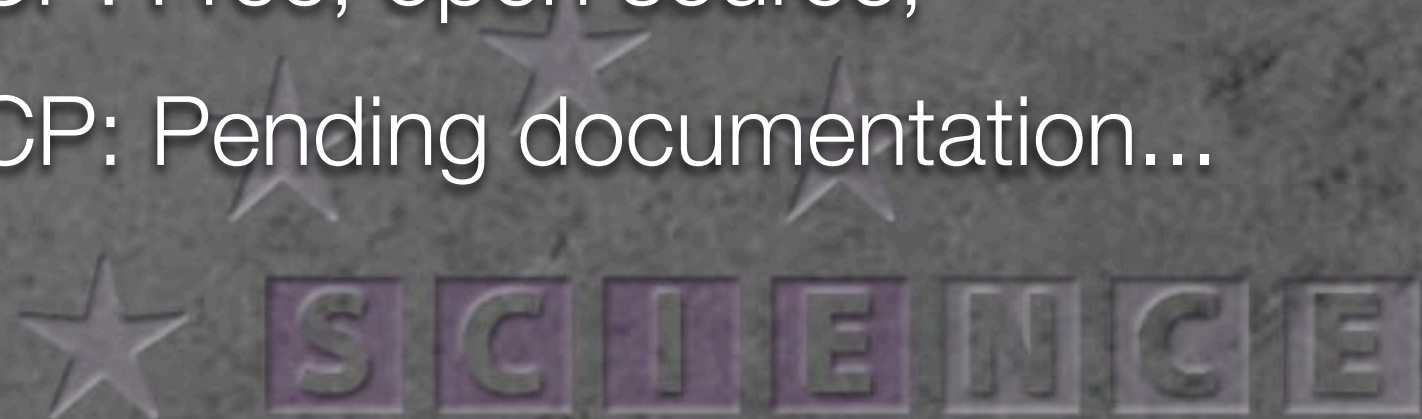


SCSCP

SYMBOLIC COMPUTATION
SOFTWARE COMPOSABILITY PROTOCOL

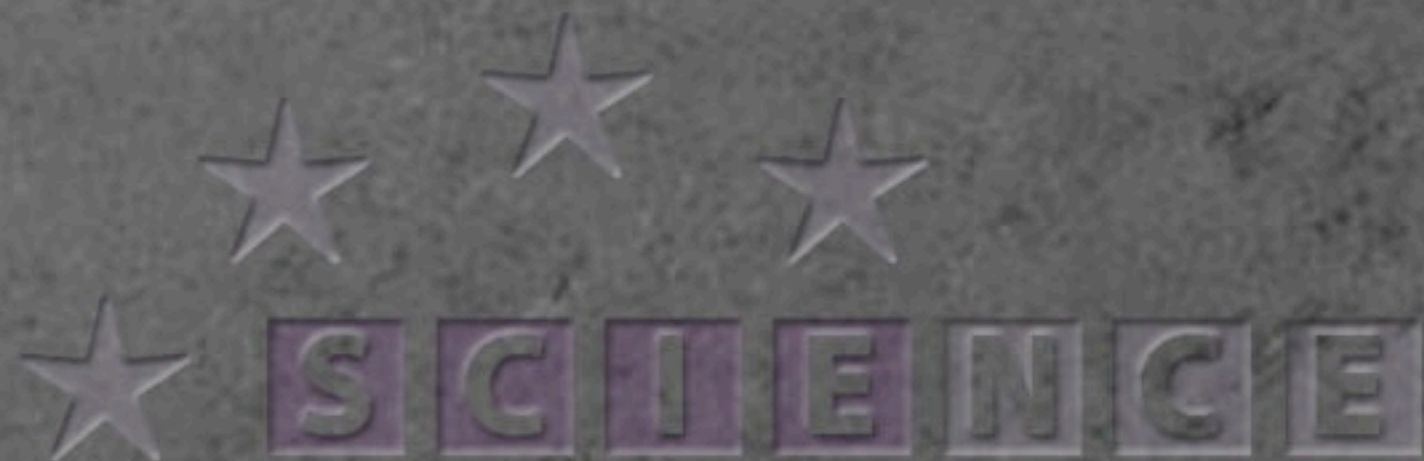
Licensing and availability

- ✦ GAP: Free and open source; SCSCP/OpenMath libraries now included with GAP distribution
- ✦ KANT: Free; binaries from KANT homepage
- ✦ TRIP SCSCP C-library: Open source, free
- ✦ Java libraries: Free, open source
- ✦ MuPAD OpenMath/SCSCP: Free, open source;
- ✦ Magma OpenMath/SCSCP: Pending documentation...



What we have...

- ✦ SCSCP has been implemented and is working for a range of systems,
- ✦ Our OpenMath/SCSCP Java libraries are quite usable,
- ✦ POPCORN is a convenient tool for demo and debugging.

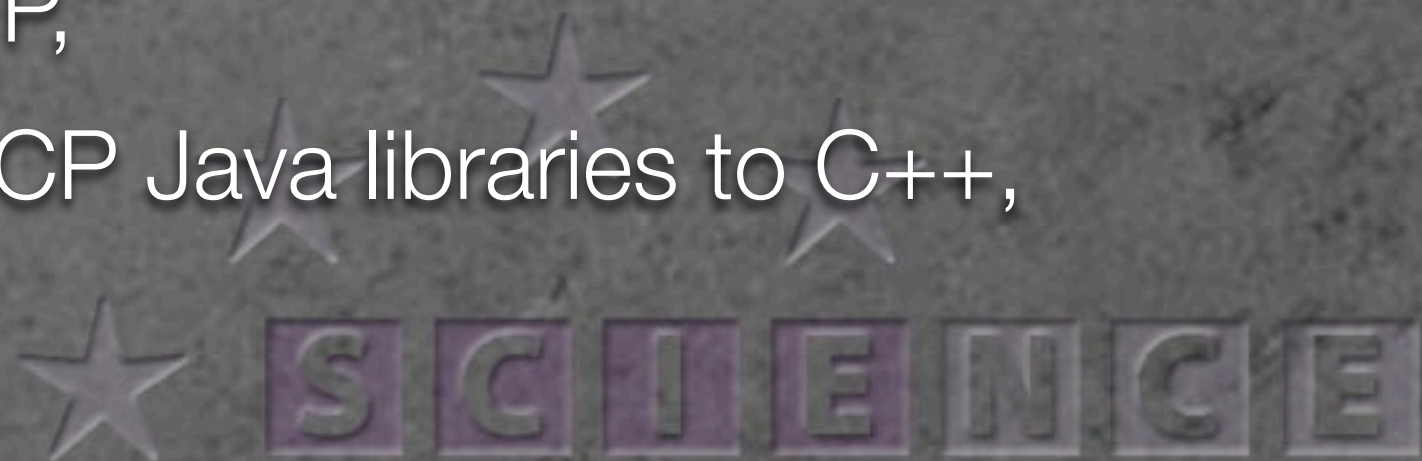


What we have...

- ✦ SCSCP has been implemented and is working for a range of systems,
- ✦ Our OpenMath/SCSCP Java libraries are quite usable,
- ✦ POPCORN is a convenient tool for demo and debugging.

What we want...

- ✦ Have SCSCP implemented and working for MORE systems,
- ✦ Enable e.g. MathML output,
- ✦ Enable SCSCP over HTTP,
- ✦ Port our OpenMath/SCSCP Java libraries to C++,
- ✦ ...



Thank you!

SCIENCE homepage

<http://www.symbolic-computation.org/>

The java libraries are available at
<http://java.symcomp.org/>

