Perm State University
Bukireva Str. 15, 614990, Perm, Russia

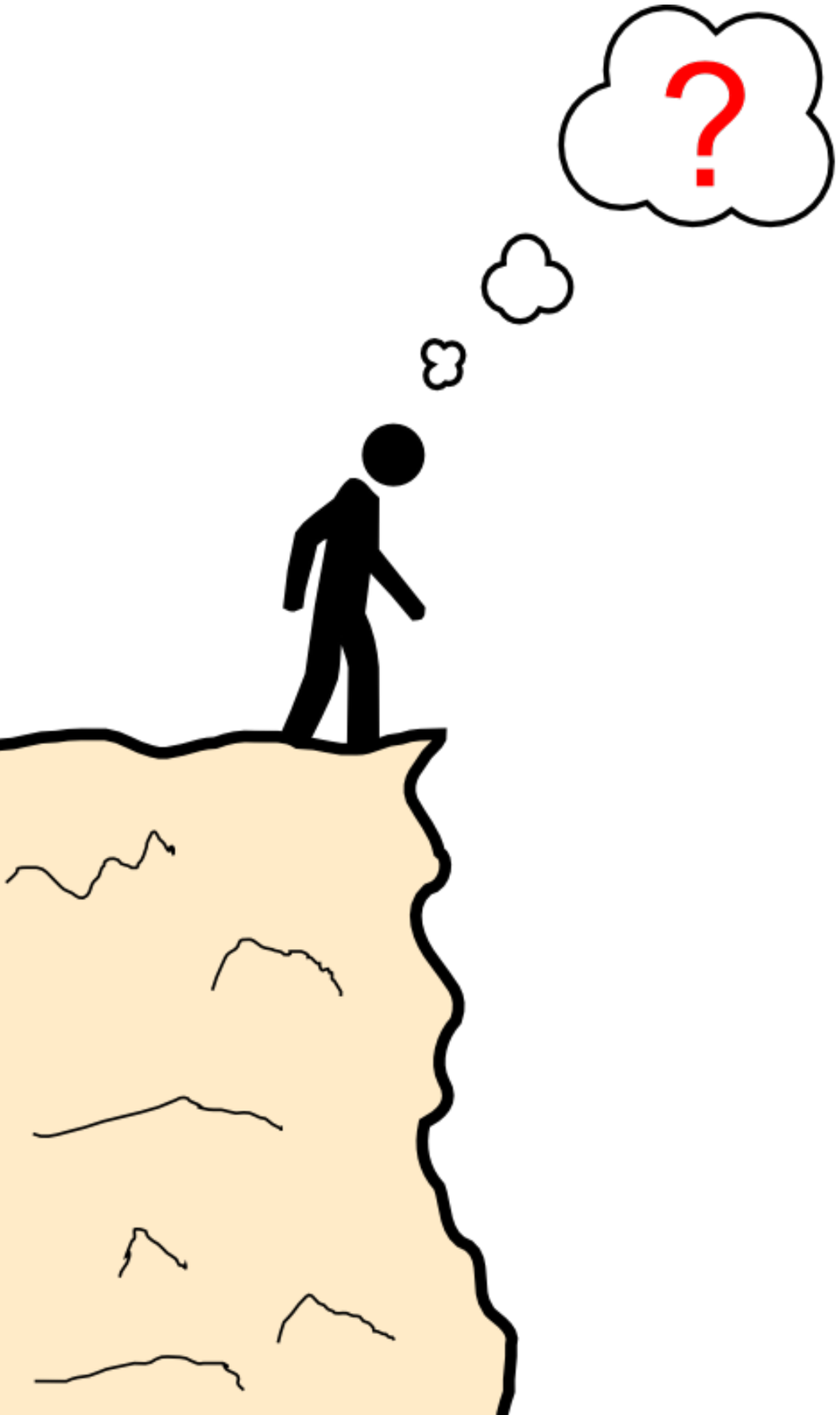
Using Scientific Visualization Tools to Bridge the Talent Gap

Konstantin Ryabinin
e-mail: kostya.ryabinin@gmail.com

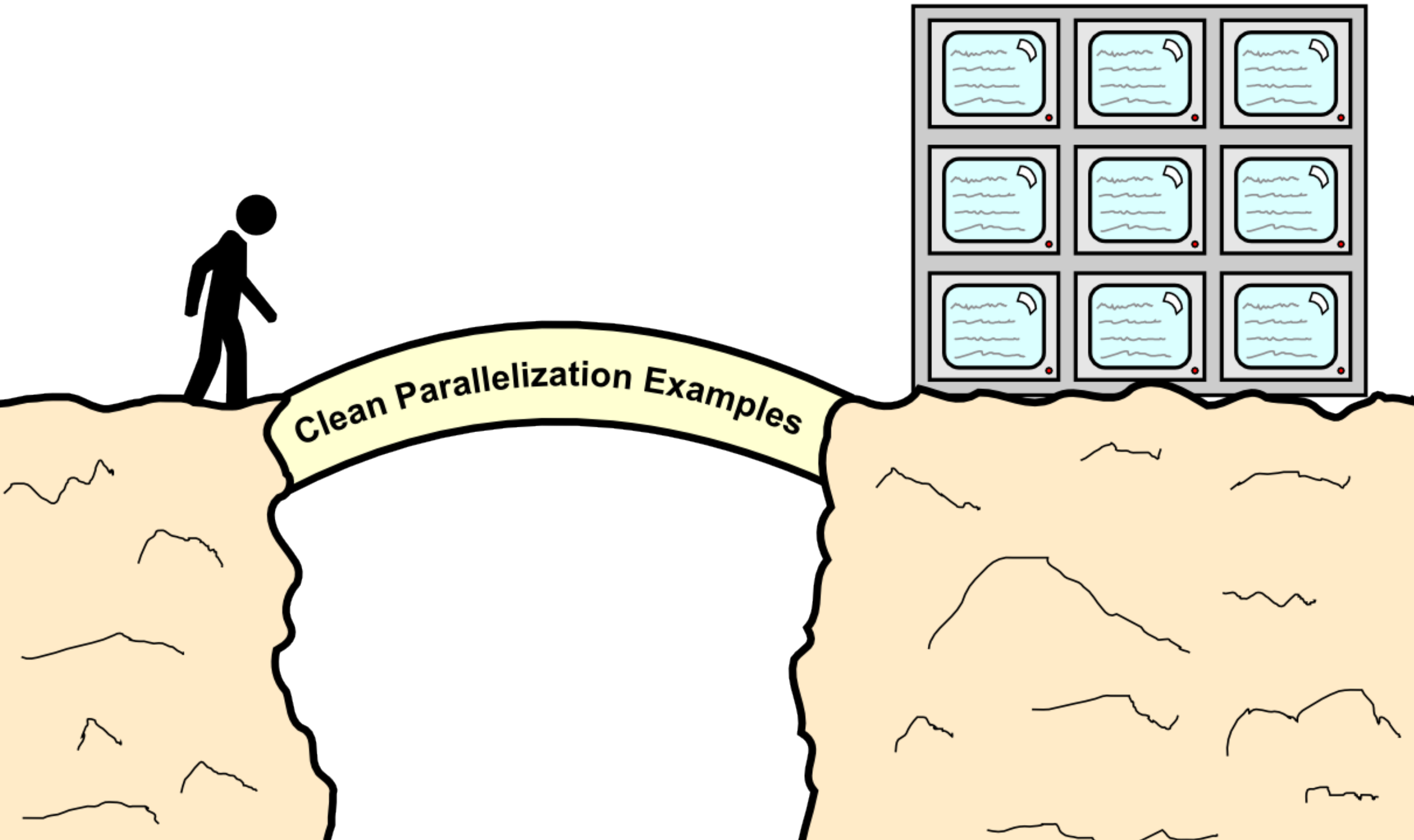
Svetlana Chuprina
e-mail: chuprinas@inbox.ru

Reykjavik – 2015

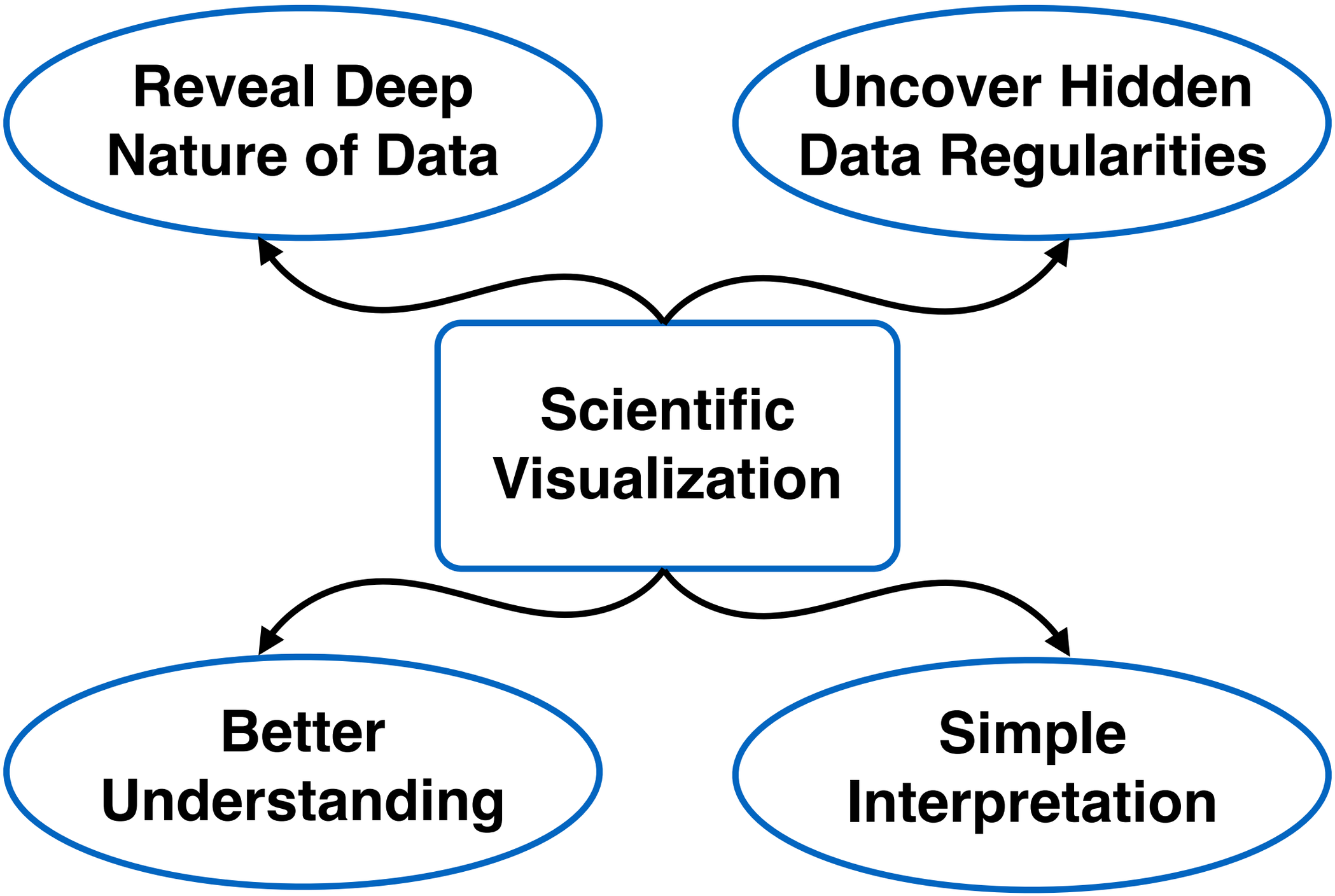
Science and HPC Talent Gap



Bridging Science and HPC Talent Gap



Benefits of Visualization



Scientific Visualization Tools Requirements

1. Easy to use with third-party data and data generators (solvers)
2. Easy to install
3. Easy to distribute data and results
4. High visual quality of rendering

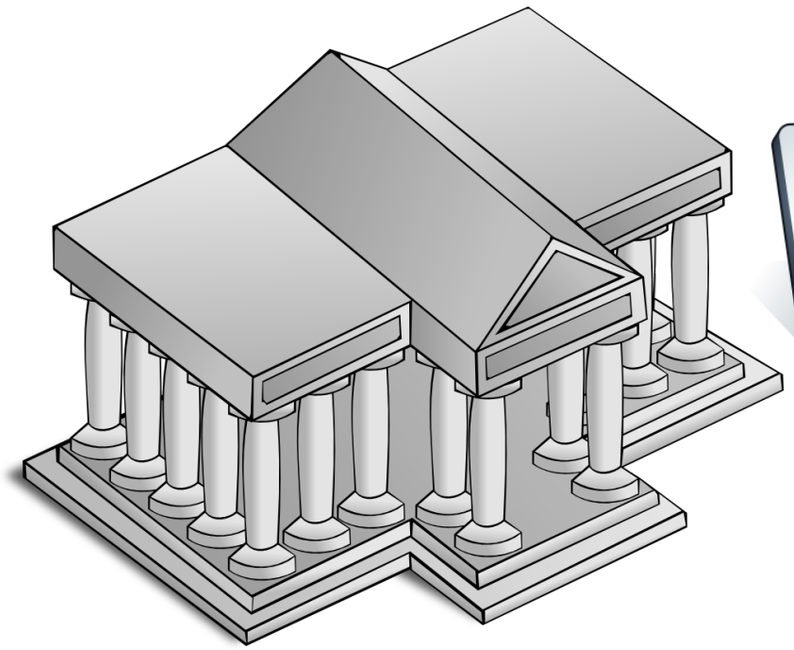
1. Automated high-level integration with third-party solvers based on ontologies

2. Multiplatform portability

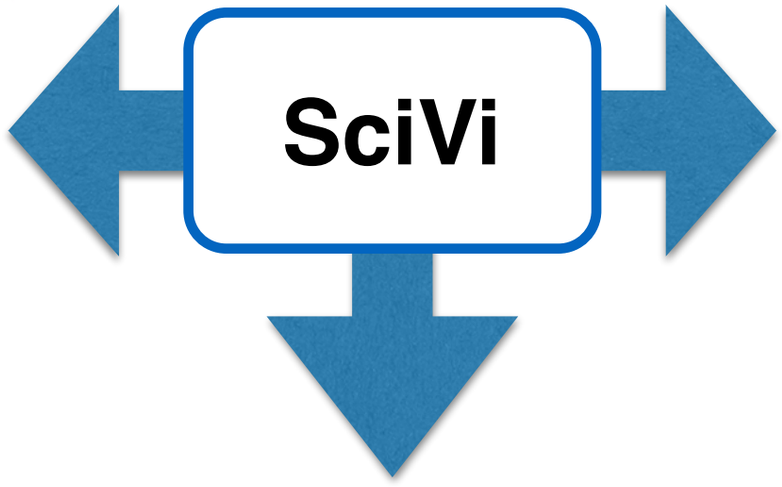
3. Client-server architecture and distributed rendering support

4. High visual quality of rendering due to adaptive anti-aliasing algorithm

Benefits of Multiplatform Portability



University



Home

Outdoor

Collaboration with Help of SciVi

Server-side (Teacher)

Network

Client-Side (Students)

Server-side (Supervisor)

Network

Client-Side (Researchers from different fields of science)

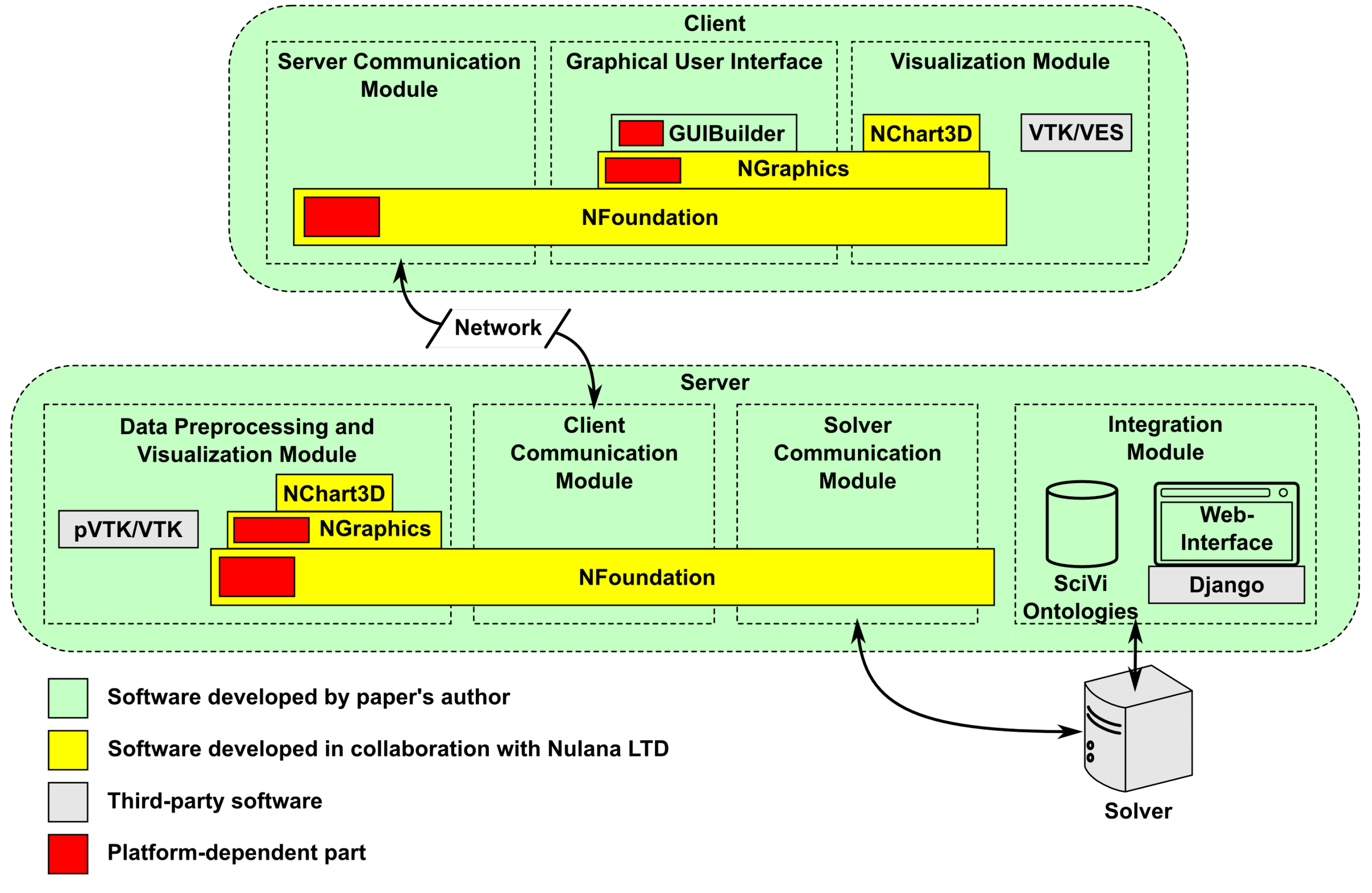
How Students Use SciVi

1. **Students develop the solvers and use SciVi as visualizer**

   OR

2. **Students use the given solver and use SciVi to analyze its output**

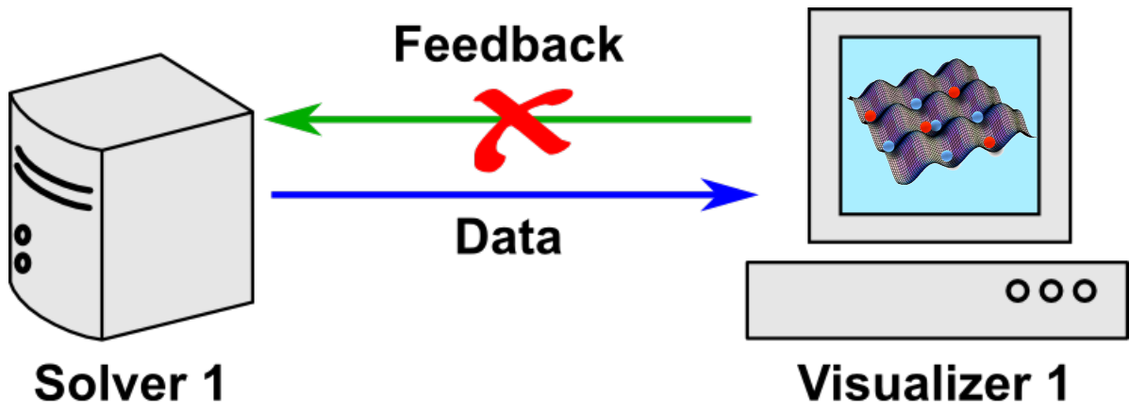
SciVi Architecture



Problems of Integration with Solver

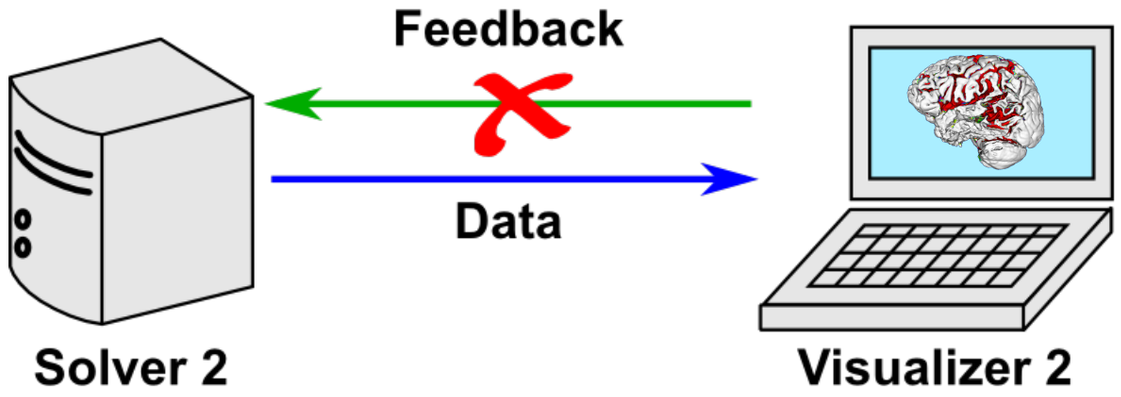
Well-known scientific visualization systems:

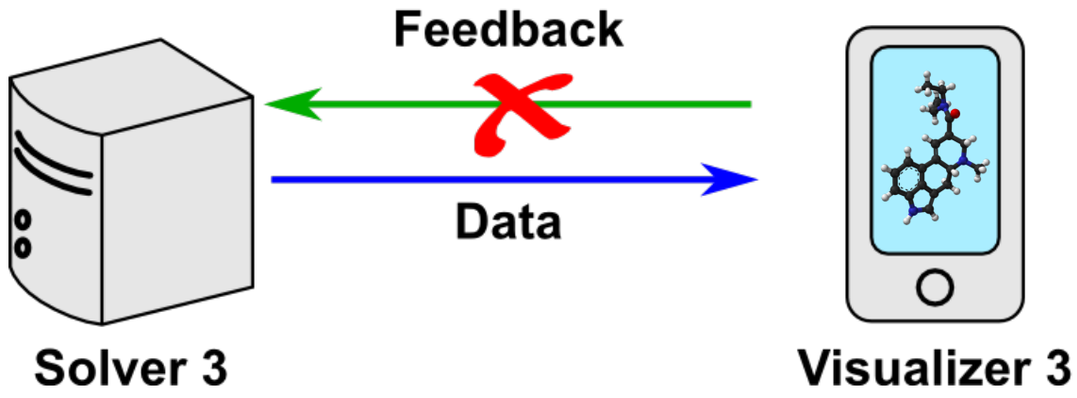
1. No unified way to adapt to third-party solver
2. No feedback



Well-known solutions:

1. Create visualizer from scratch
2. Require solver to generate data in standard format
3. Develop data format convertors





Tackle the Integration Problems

Adaptive integration:

1. Model-based architecture

2. Ontology engineering

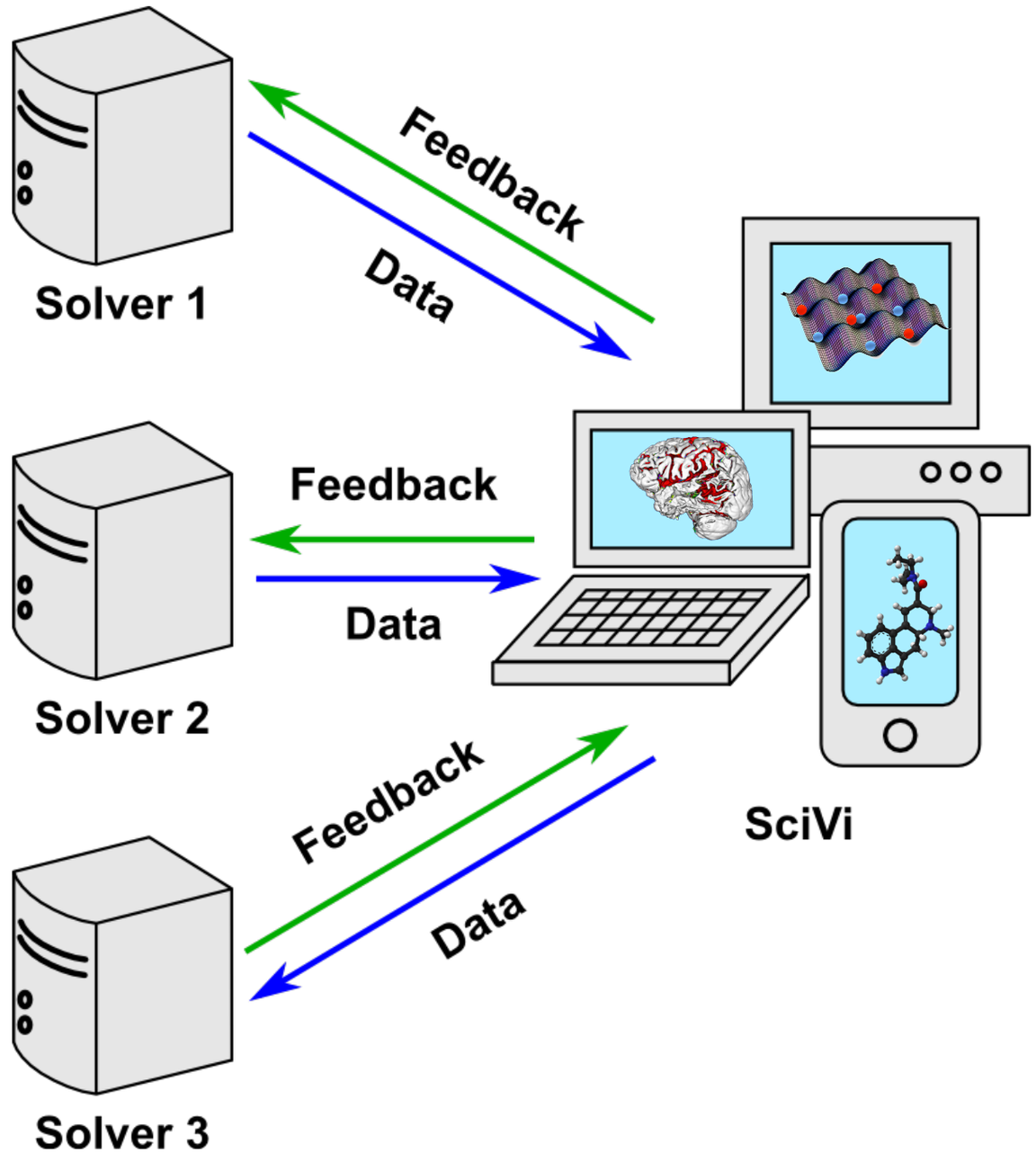
Ontology – formal model of the application domain

$$O = \langle T, R, A \rangle$$

T **– thesaurus of application domain concepts**

R **– set or relations between concepts**

A **– set of axioms**



Tackle the Integration Problems

We suggest to use the damain-specific ontologies without axioms to ensure high-level integration of visualizer with third-party solvers:

$A = \varnothing$

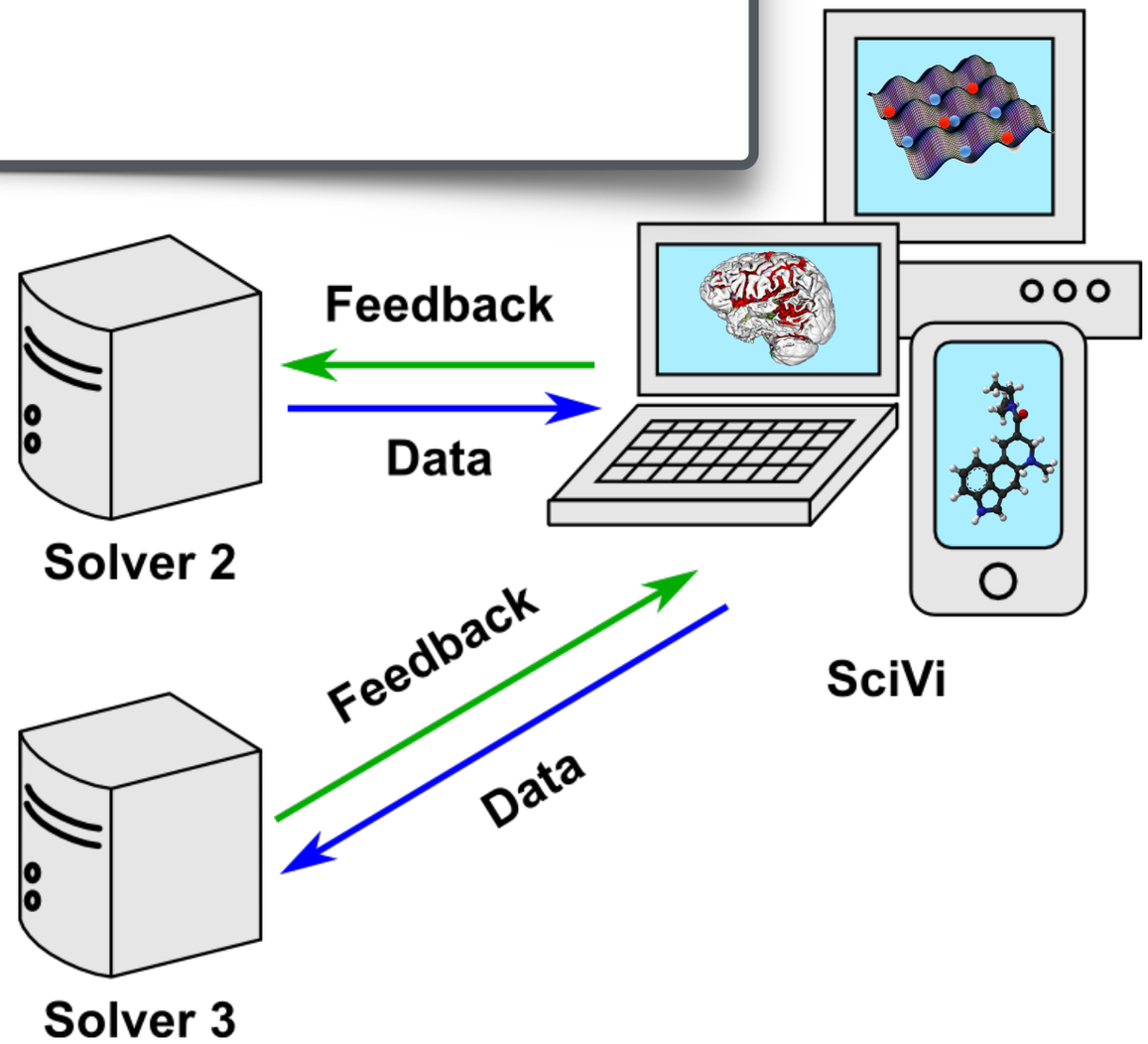
Ontology – formal model of the application domain

$O = \langle T, R, A \rangle$

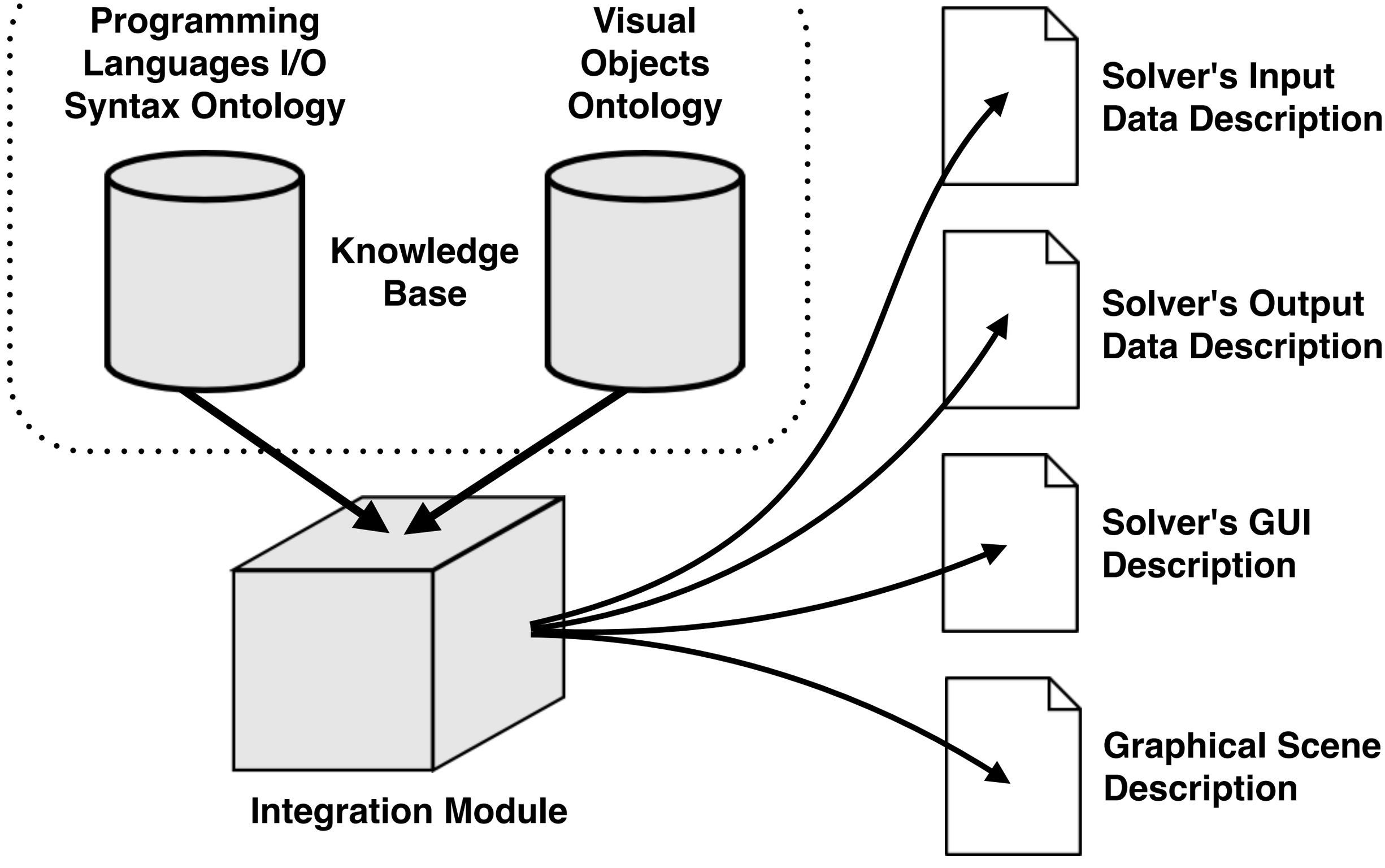
T **– thesaurus of application domain concepts**

R **– set or relations between concepts**

A **– set of axioms**



SciVi Integration Module



SciVi Integration Module


```
<?xml version="1.0"?>


<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>


<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.semanticweb.org/icosaeder/ontologies/2014/6/untitled-ontology-20"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     ontologyIRI="http://www.semanticweb.org/icosaeder/ontologies/2014/6/untitled-ontology-20">
    <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Declaration>
        <Class IRI="#2d-chart"/>
    </Declaration>
    <Declaration>
        <Class IRI="#3d-chart"/>
    </Declaration>
    <Declaration>
        <Class IRI="#background"/>
    </Declaration>
    <Declaration>
        <Class IRI="#borderColor"/>
    </Declaration>
    <Declaration>
        <Class IRI="#caption"/>
    </Declaration>
```

Integration Module

Solver's Input Data Description

Solver's Output Data Description

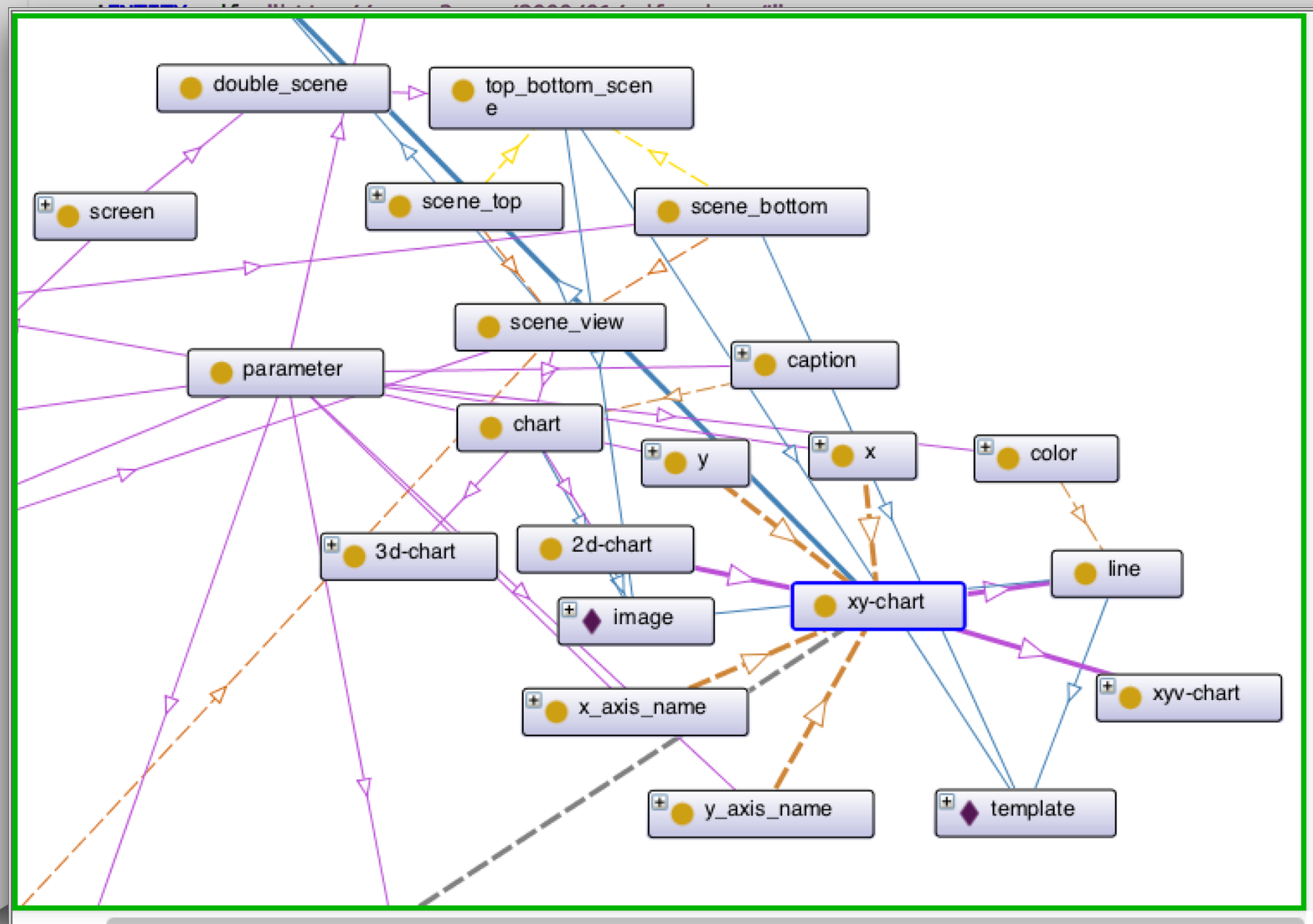
Solver's GUI Description

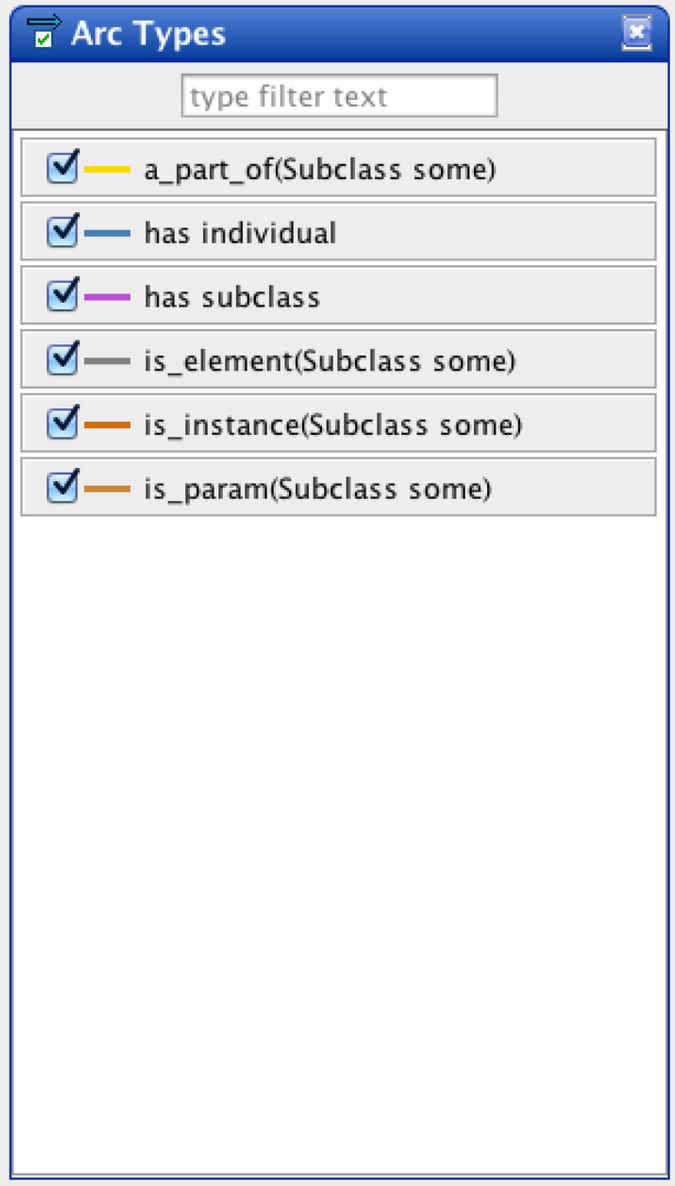
Graphical Scene Description

SciVi Integration Module


```
<?xml version="1.0"?>


<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
```





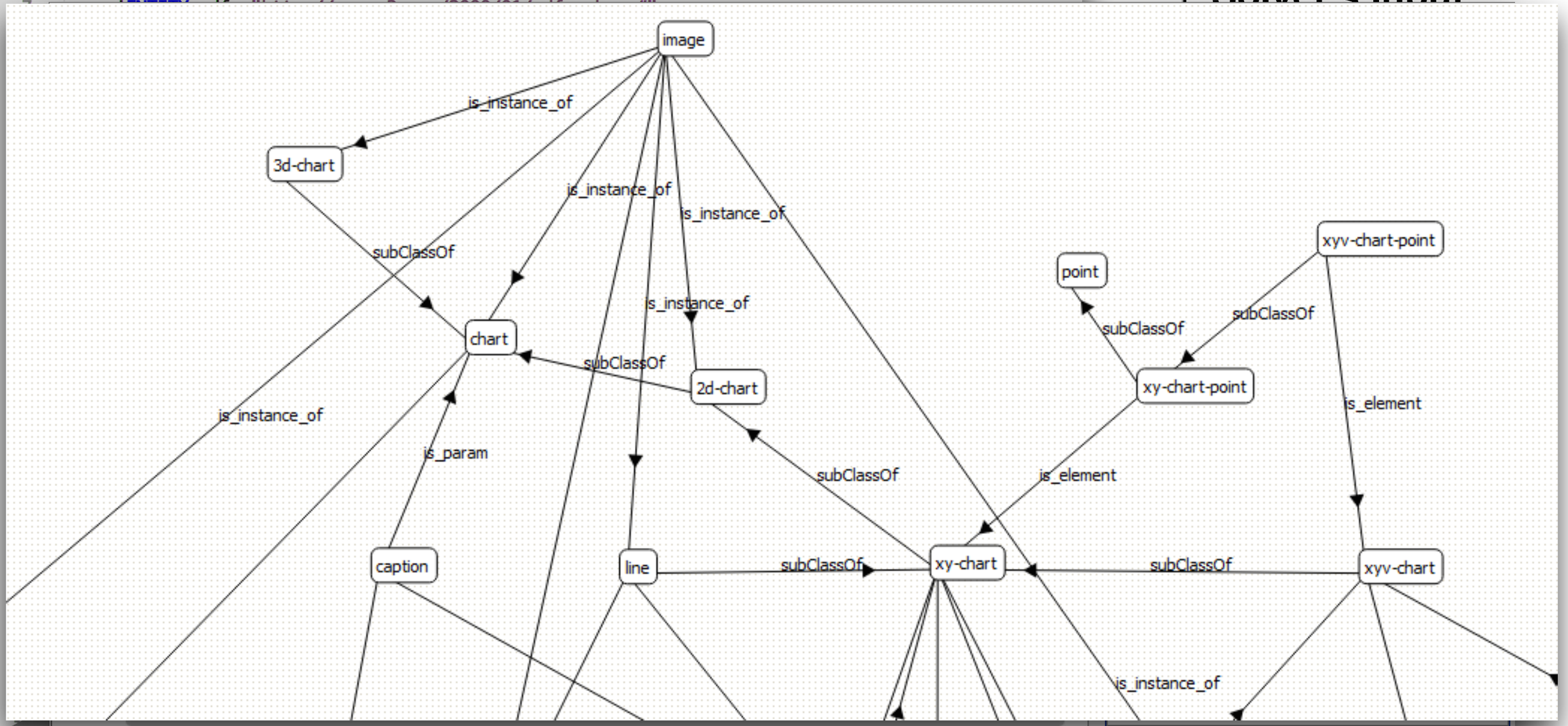
Solver's Input

Integration Module

Description

SciVi Integration Module


```
1  <?xml version="1.0"?>
2
3
4  <!DOCTYPE Ontology [
5      <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
6      <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
```



Integration Module

Solver's Input

Description

I/O Syntax Ontology

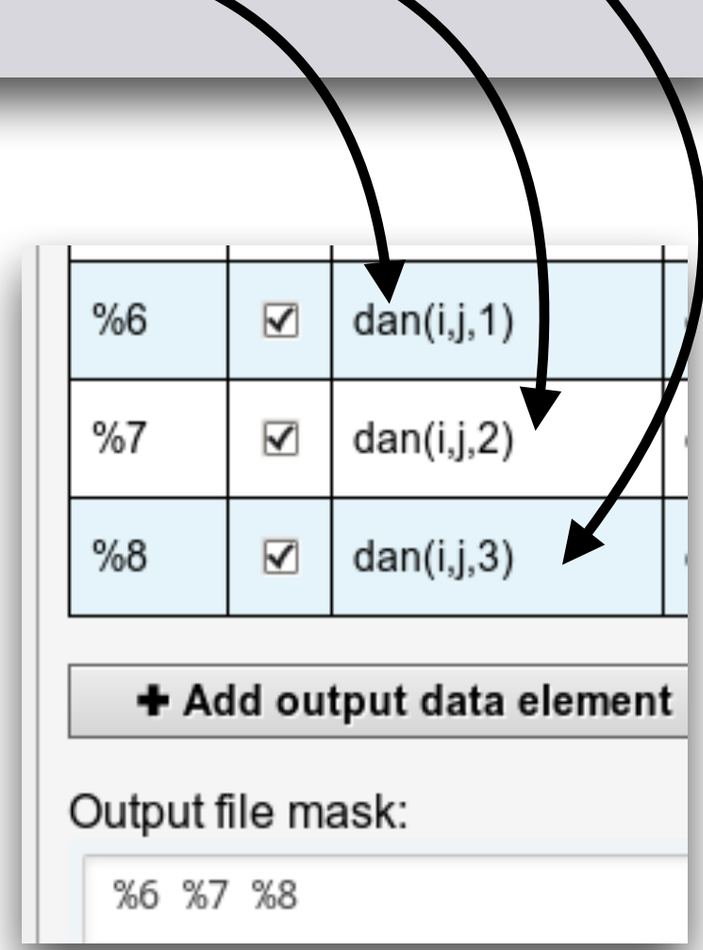
- **Input/Output statements description**
  - **C, C++, Fortran, Java**

- **Solver's source code is available:**
  - **Automatic parser generation**
  - **Automatic Input/Output data description generation**

- **Solver's source code is not available:**
  - **High-level graphical user interface for Input/Output data description**

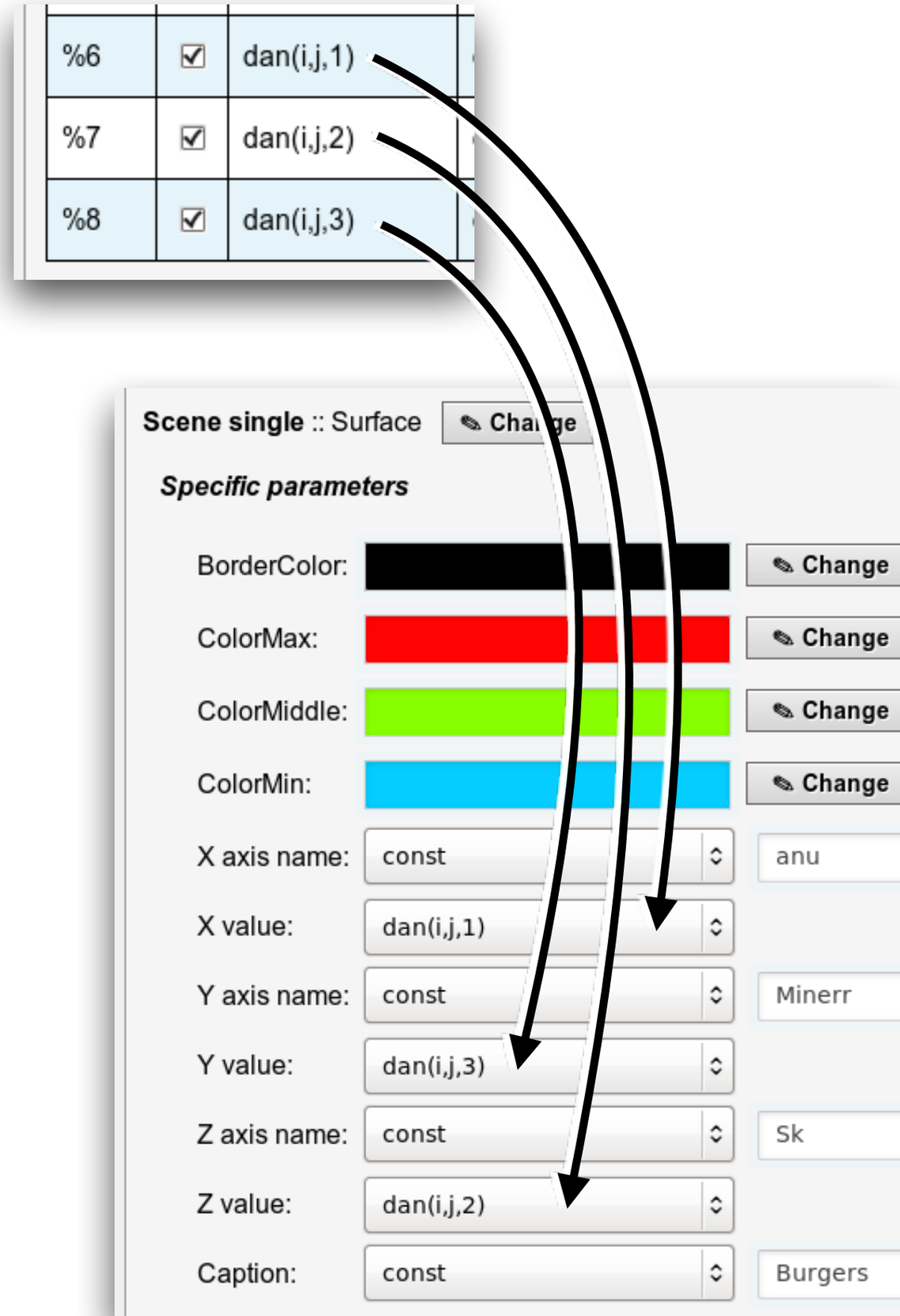
Hybrid finite-difference schemes optimization for solving hydrodynamic problems

```
149
150   open(1,file='result.txt')
151   do 16 j=1,n
152    do 17 i=1,m
153     dan(i,j,1)=anu1(i)
154     dan(i,j,2)=sk(j)
155     dan(i,j,3)=proc1(i,j)
156     write(1,*) dan(i,j,1), dan(i,j,2), dan(i,j,3)
157   17 continue
158   16 continue
159
```



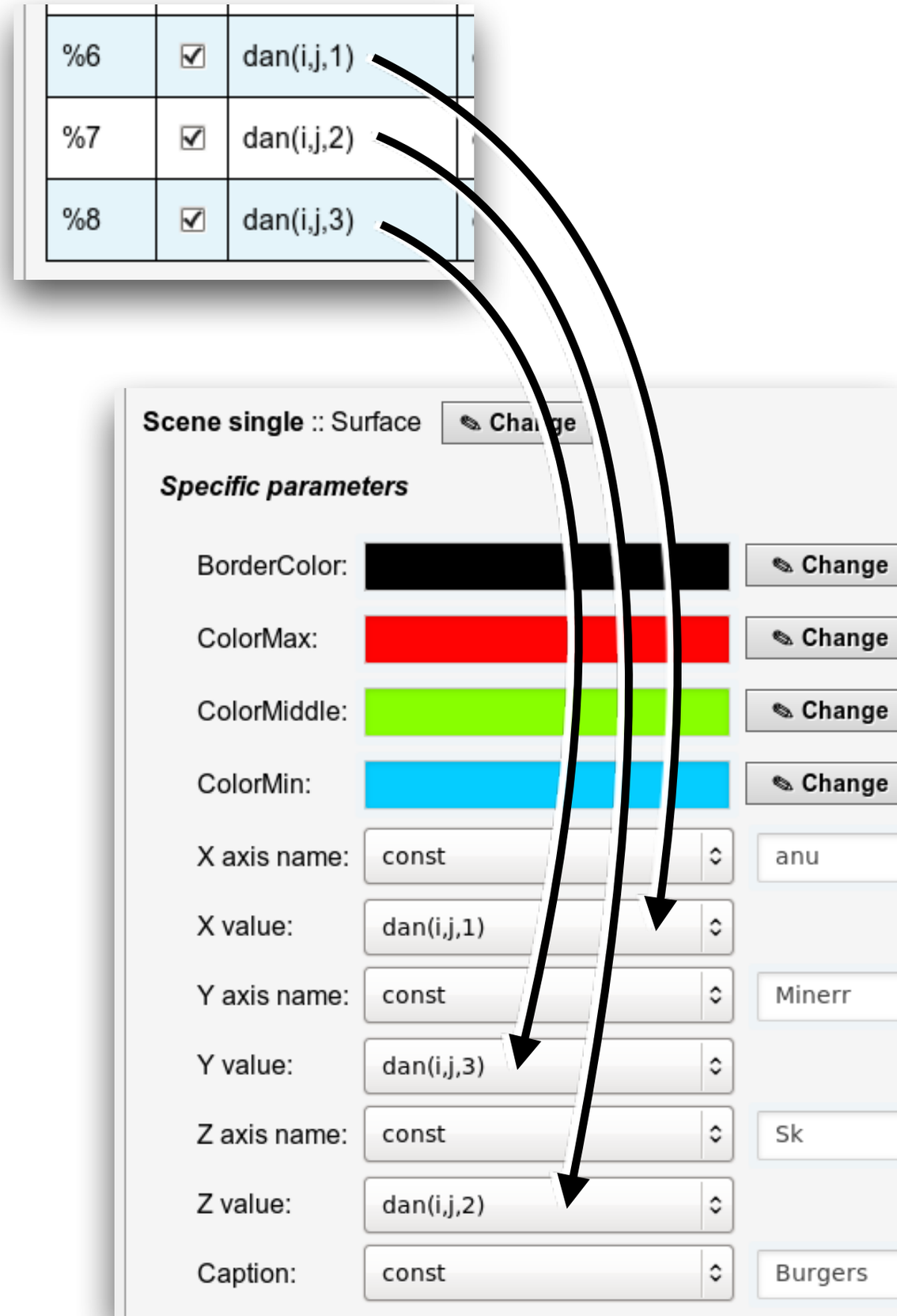
Visual Objects Ontology

- Visual objects description:
  - 3D-models, charts, images, ...
- Visualization settings:
  - Mapping Input/Output data to visual objects' properties
- Solver's executable is available:
  - Set up feedback from visualizer to solver
- Solver's executable is not available:
  - Data visualization only



```
1.00000005E-03  0.100000001   2.84658074
1.99999986E-03  0.100000001   1.29388571
2.99999979E-03  0.100000001   0.443625450
3.99999972E-03  0.100000001   0.199139118
4.99999989E-03  0.100000001   0.527486205
6.00000005E-03  0.100000001   0.795495510
7.00000022E-03  0.100000001   1.00262165
8.00000038E-03  0.100000001   1.16453767
9.00000054E-03  0.100000001   1.29425526
1.00000007E-02  0.100000001   1.40078068
1.00000005E-03  0.200000003   2.83433199
1.99999986E-03  0.200000003   1.28409863
2.99999979E-03  0.200000003   0.446271896
3.99999972E-03  0.200000003   0.187385082
4.99999989E-03  0.200000003   0.530013442
6.00000005E-03  0.200000003   0.799649954
7.00000022E-03  0.200000003   1.00648999
8.00000038E-03  0.200000003   1.16815567
9.00000054E-03  0.200000003   1.29763484
1.00000007E-02  0.200000003   1.40398741
1.00000005E-03  0.300000012   2.82210708
1.99999986E-03  0.300000012   1.28102303
2.99999979E-03  0.300000012   0.448900461
3.99999972E-03  0.300000012   0.180268288
4.99999989E-03  0.300000012   0.532552600
6.00000005E-03  0.300000012   0.803792477
7.00000022E-03  0.300000012   1.01032257
8.00000038E-03  0.300000012   1.17175579
9.00000054E-03  0.300000012   1.30103230
1.00000007E-02  0.300000012   1.40718818
1.00000005E-03  0.400000006   2.80989408
```

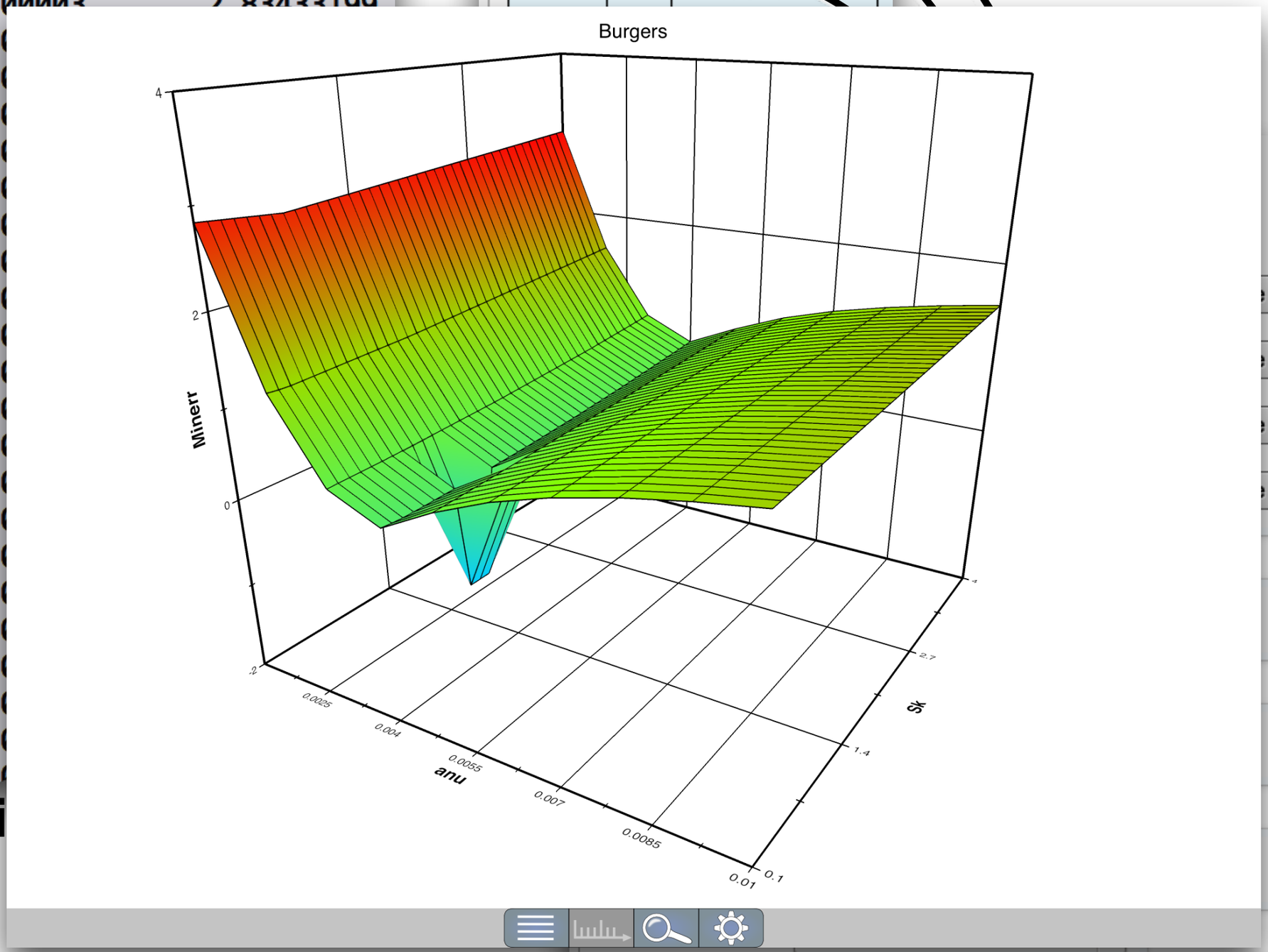
- **Data visualization only**



1	1.00000005E-03	0.100000001	2.84658074
2	1.99999986E-03	0.100000001	1.29388571
3	2.99999979E-03	0.100000001	0.443625450
4	3.99999972E-03	0.100000001	0.199139118
5	4.99999989E-03	0.100000001	0.527486205
6	6.00000005E-03	0.100000001	0.795495510
7	7.00000022E-03	0.100000001	1.00262165
8	8.00000038E-03	0.100000001	1.16453767
9	9.00000054E-03	0.100000001	1.29425526
10	1.00000007E-02	0.100000001	1.40078068
11	1.00000005E-03	0.200000003	2.83433199
12	1.99999986E-03	0.2000
13	2.99999979E-03	0.2000
14	3.99999972E-03	0.2000
15	4.99999989E-03	0.2000
16	6.00000005E-03	0.2000
17	7.00000022E-03	0.2000
18	8.00000038E-03	0.2000
19	9.00000054E-03	0.2000
20	1.00000007E-02	0.2000
21	1.00000005E-03	0.3000
22	1.99999986E-03	0.3000
23	2.99999979E-03	0.3000
24	3.99999972E-03	0.3000
25	4.99999989E-03	0.3000
26	6.00000005E-03	0.3000
27	7.00000022E-03	0.3000
28	8.00000038E-03	0.3000
29	9.00000054E-03	0.3000
30	1.00000007E-02	0.3000
31	1.00000005E-03	0.4000
	1.99999986E-03	0.4000

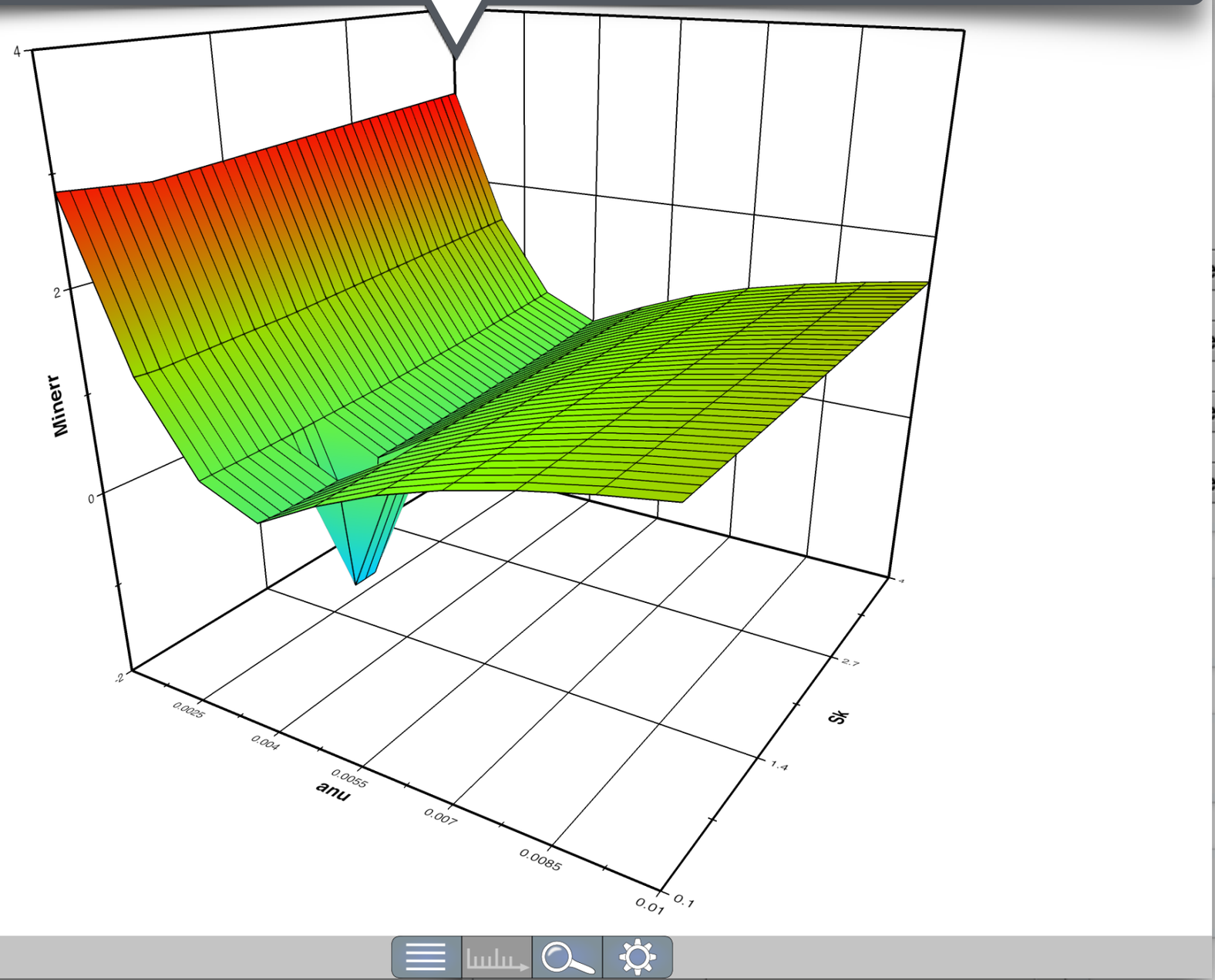
- Data visualizati

%6	☑	dan(i,j,1)
%7	☑	dan(i,j,2)
%8	☑	dan(i,j,3)



1	1.00000005E-03	0.100000001	2.84658074
2	1.99999986E-03	0.100000001	1.29388571
3	2.99999979E-03	0.100000001	0.443625450
4	3.99999972E-03	0.100000001	0.199139118
5	4.99999989E-0		
6	6.00000005E-0		
7	7.00000022E-0		
8	8.00000038E-0		
9	9.00000054E-0		
10	1.00000007E-0		
11	1.00000005E-0		
12	1.99999986E-0		
13	2.99999979E-03	0.20000	
14	3.99999972E-03	0.20000	
15	4.99999989E-03	0.20000	
16	6.00000005E-03	0.20000	
17	7.00000022E-03	0.20000	
18	8.00000038E-03	0.20000	
19	9.00000054E-03	0.20000	
20	1.00000007E-02	0.20000	
21	1.00000005E-03	0.30000	
22	1.99999986E-03	0.30000	
23	2.99999979E-03	0.30000	
24	3.99999972E-03	0.30000	
25	4.99999989E-03	0.30000	
26	6.00000005E-03	0.30000	
27	7.00000022E-03	0.30000	
28	8.00000038E-03	0.30000	
29	9.00000054E-03	0.30000	
30	1.00000007E-02	0.30000	
31	1.00000005E-03	0.40000	
	1.999999	0.40000	

Visual representation of the computational scheme error depending on its parameters helps students to understand complex computation method



- Data visualizati

A picture paints a thousand words

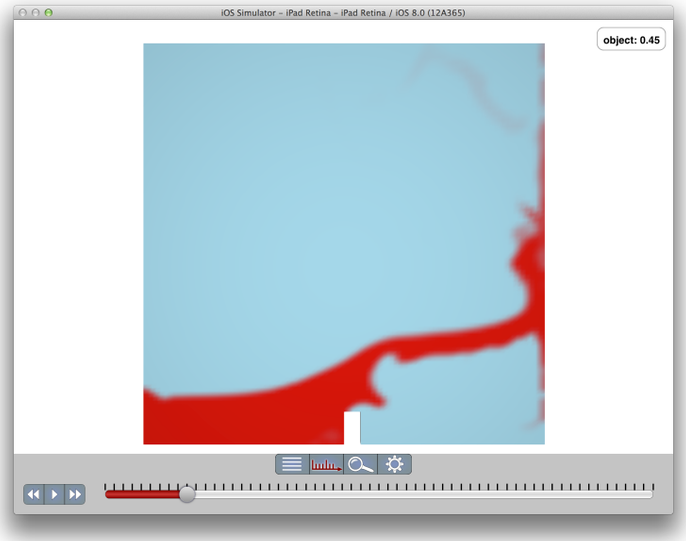
Distinctive features of the method:

1. **High-level integration with third-party solvers**
2. **Support of feedback**
3. **Independency of solver's architecture**
4. **Independency of solver's application domain**

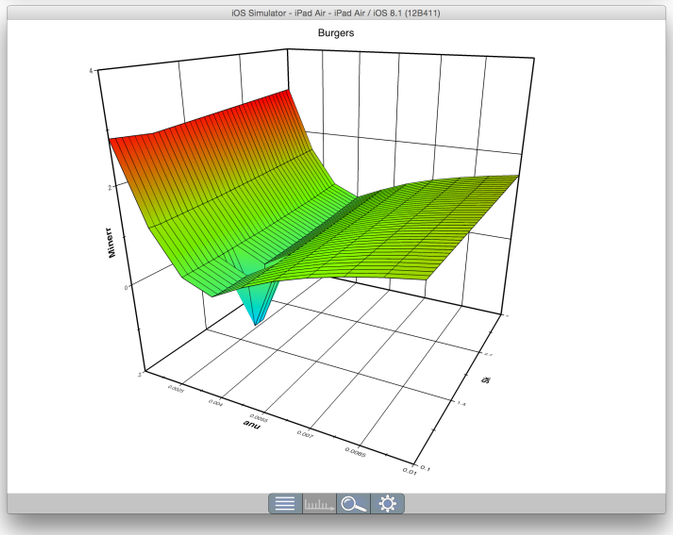
Benefits for teachers:

1. **High-quality demonstration of processes, methods and algorithms**
2. **Ability to use in any field of science**
3. **No need to hire third-party developers**

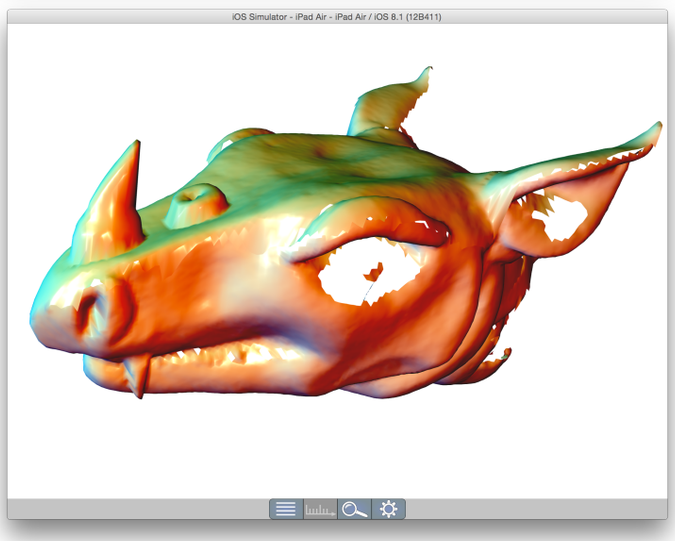
SciVi Usage



1. Fluid simulation
   Solver: OpenFOAM
   Language: C++



2. Hybrid finite-difference schemes optimization
   Solver: Burgers2
   Language: Fortran



3. Digital art object
   Solver: 3D-scanner

How SciVi Helps to Train Specialists

A lot of disciplines assume students to develop solvers (Mathematical Physics, Computational Science, Geo-Information Systems, ...)

SciVi provides:

1. **High-quality demonstration of results**
2. **Verification of results through visualization**
3. **Collaboration by using laptops and mobile devices students have wherever they want to**

How SciVi Helps to Study Parallel Algorithms

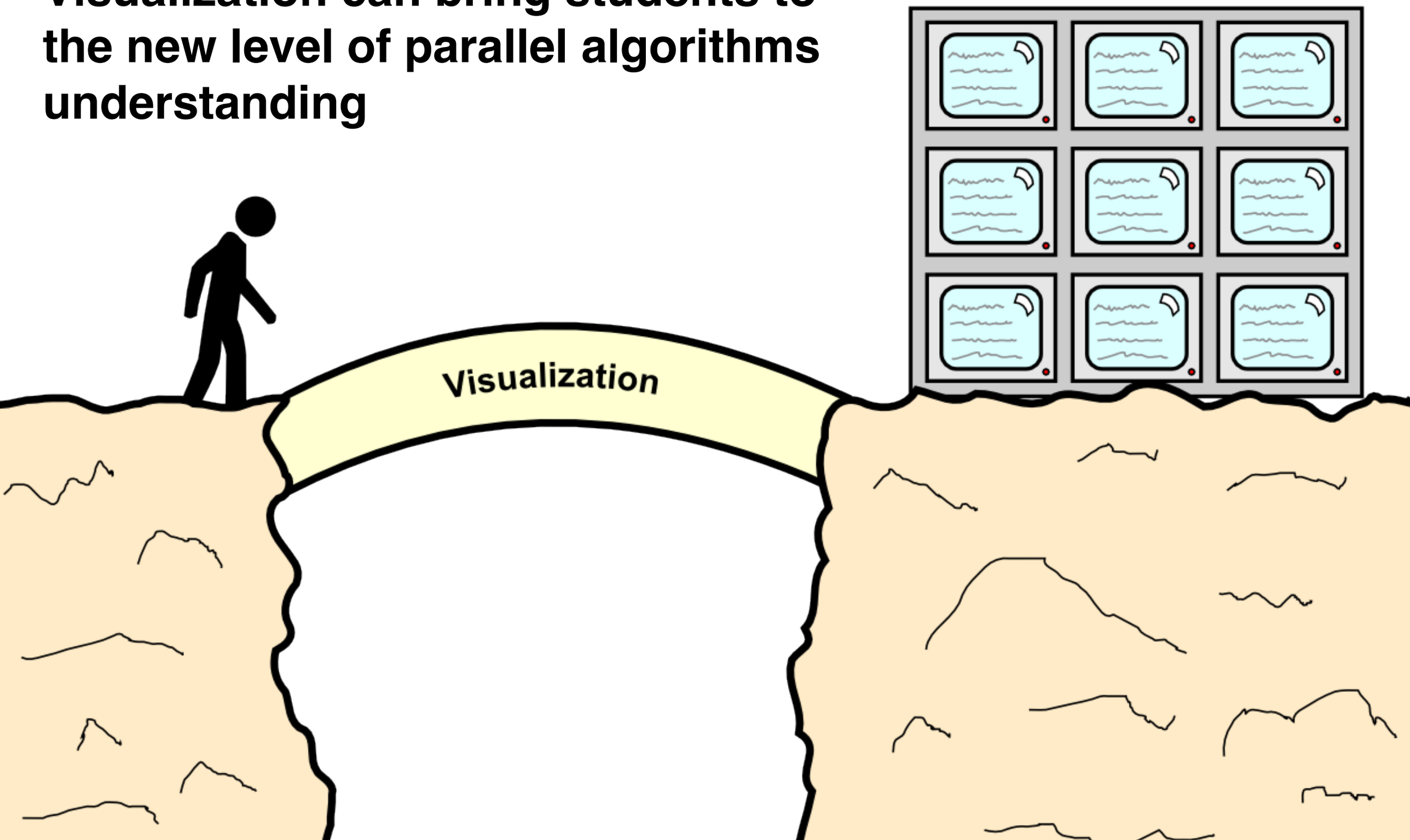
1. **Visualize intermediate calculations (uncover the data relations to perform efficient data decomposition)**

2. **Visualize statistics obtained from solver (parallel algorithms evaluation)**

3. **Visualize both serial and parallel algorithms results side by side (verification of parallelization)**

Science and HPC Talent Gap

Visualization can bring students to the new level of parallel algorithms understanding



Future Work

1. **Integrate SciVi in educational process as a part of pedagogical learning environment**
2. **Encourage students to achieve complex higher-order skills in parallel programming through scientific visualization**
3. **Use SciVi and Perm State University supercomputer facilities to tackle complex interdisciplinary scientific problems**

Perm State University
Bukireva Str. 15, 614990, Perm, Russia

Thank you for your attention!

Konstantin Ryabinin
e-mail: kostya.ryabinin@gmail.com

Svetlana Chuprina
e-mail: chuprinas@inbox.ru

Reykjavik – 2015