

Архитектура и реализация сервис-ориентированной научной среды MathCloud

О.В. Сухорослов
Центр грид-технологий и распределенных
вычислений ИСА РАН

Service-Oriented Science

- ...So-called service-oriented architectures define standard interfaces and protocols that allow developers to encapsulate information tools as services that clients can access without knowledge of, or control over, their internal workings. Thus, tools formerly accessible only to the specialist can be made available to all...
- **service-oriented science** refers to scientific research enabled by distributed networks of interoperating services

I. Foster. Service-Oriented Science //
Science. – 2005. – V. 308, N. 5723. - P. 814-817.

Ключевые моменты

- Почему именно сервисы?
- Интероперабельность
- Композиция сервисов
- Масштабируемость
- Безопасность
- Мотивация
- Технологический барьер

Интероперабельность

- Протоколы
- Интерфейсы
- Форматы
- Семантика

Стандарты и технологии

- Протокол HTTP
- Форматы XML, JSON
- Web-сервисы
 - WS-* (WSDL, SOAP, UDDI ...)
 - REST

Композиция сервисов

- Многие прикладные задачи допускают **декомпозицию** на несколько типовых подзадач
- Процесс решения задачи в распределенной среде сводится к **композиции** сервисов
- Полученный **сценарий** может быть опубликован в виде нового сервиса среды

Масштабируемость

- Обработка запроса может потребовать значительных вычислительных ресурсов
 - Большой объем обрабатываемых данных
 - Большое количество клиентов/запросов
-
- Грид
 - Cloud Computing

Безопасность

- Контроль доступа к сервисам
 - Аутентификация пользователей и сервисов
 - Управления группами пользователей (ВО)
 - Делегация прав
-
- Грид

Мотивация

- Удобство пользователей
- Упрощение поддержки
- Научная кооперация
- Публикация результатов
- Коммерциализация

Технологический барьер

- Разработчик сервиса
 - Квалификация
 - Технологии разработки, размещения, регистрации и защиты сервисов (middleware)
 - Технологии параллельных и распределенных вычислений, грид-технологии
 - Физические ресурсы
 - Хостинг сервиса
 - Вычислительная инфраструктура

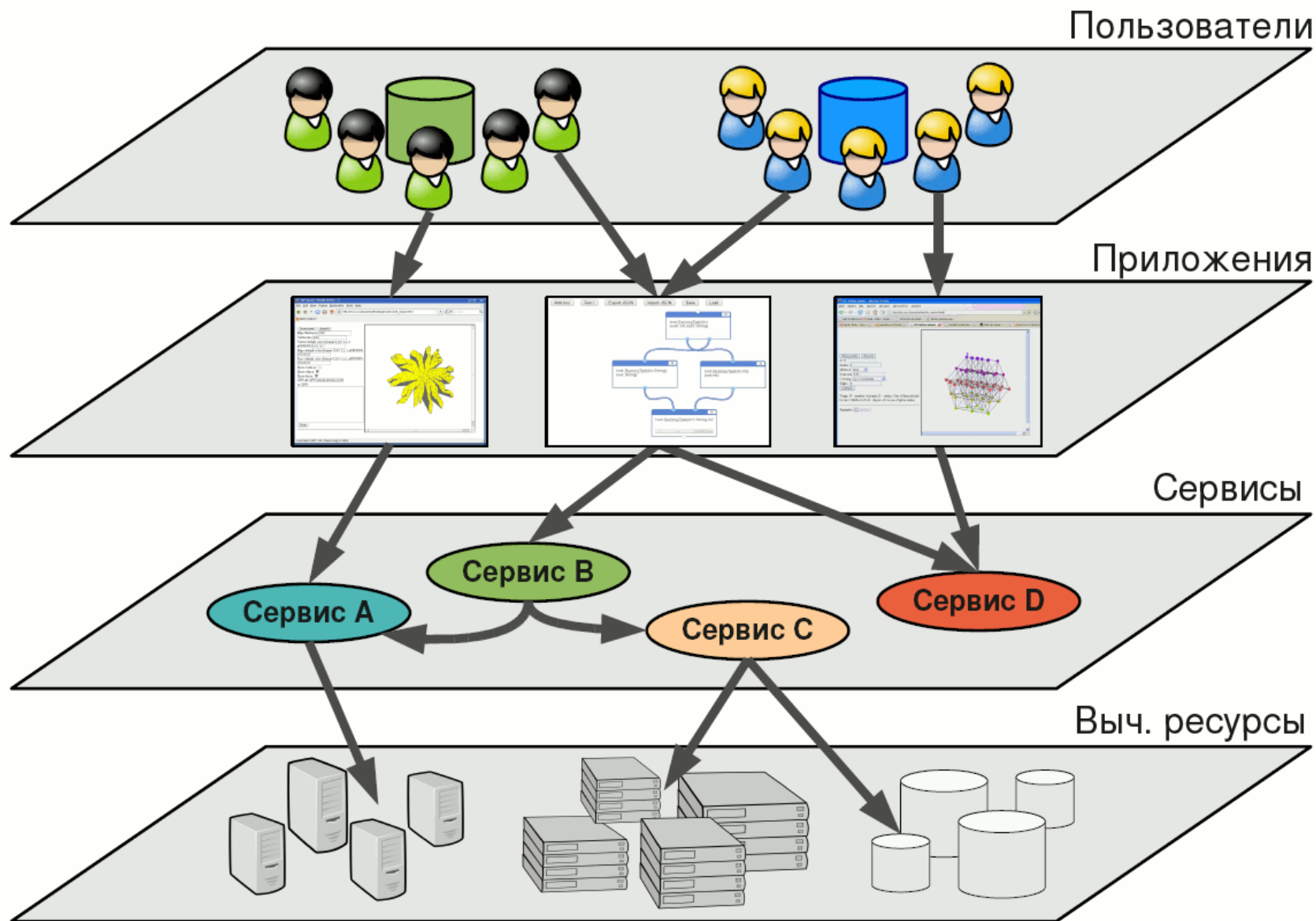
Технологический барьер

- Пользователь сервиса
 - Доступ к привычному и понятному интерфейсу
 - Отсутствие необходимости вникать в техническую сторону функционирования сервиса
 - Единое “рабочее пространство”
- Problem Solving Environment (PSE)
- Scientific Gateways
- Workflow

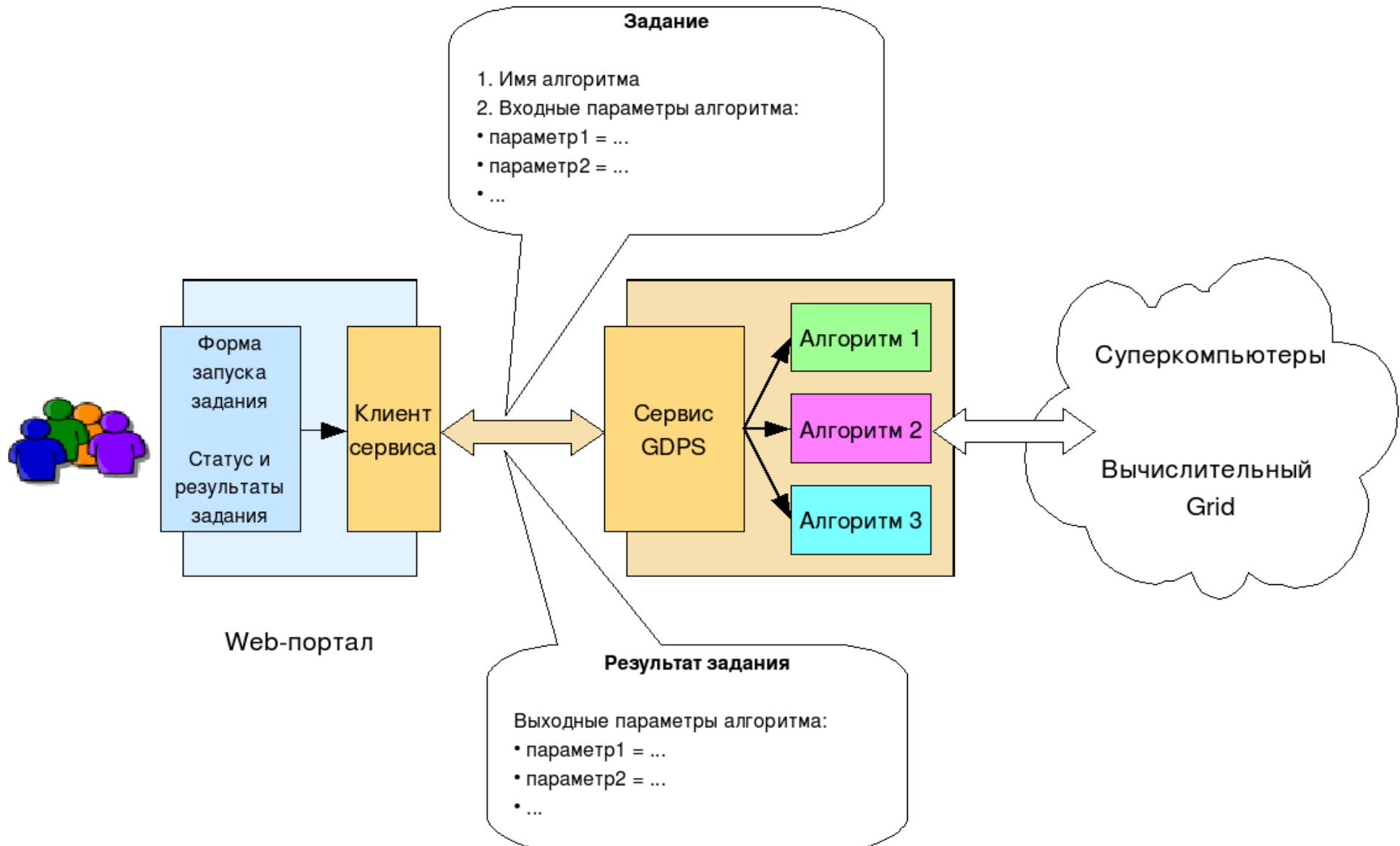
MathCloud

- Сервис-ориентированная научная среда
 - предоставление унифицированного доступа к проблемно-ориентированным вычислительным сервисам
 - поддержка интеграции данных сервисов при решении прикладных задач
- Ключевые требования
 - Простота, легкость использования, открытость
- Базируется на веб- и грид-технологиях

Архитектура MathCloud



Geo Data Processing Service



Алгоритмический сервис

- Доступный по сети программный компонент, поддерживающий решение задач определенного класса по запросу клиентов
- Запрос содержит параметризованное описание задачи (задание), формулируемое в виде конечного набора входных параметров
- Результат обработки запроса оформляется в виде конечного набора выходных параметров
- $ServiceX (in1, in2) \rightarrow \{out1, out2, out3\}$

Важное ограничение

- Результат запроса к сервису определяется исключительно значениями передаваемых клиентом входных параметров и не зависит от результатов других запросов
- Позволяет
 - упростить интерфейс и реализацию сервисов
 - повысить масштабируемость и отказоустойчивость

Важные требования

- Обработка запроса может быть сопряжена с проведением длительных вычислений
 - сервис не может сразу вернуть результат клиенту
- Сервис может принимать на вход или генерировать большие объемы данных
 - нельзя передавать непосредственно внутри параметров запроса или результата

Принципы реализации

- Каждый сервис MathCloud оформляется в виде веб-сервиса, реализующего унифицированный интерфейс
- Данный интерфейс основан на
 - спецификации протокола HTTP
 - архитектурном стиле REST
 - распространенных форматах представления данных (JSON, HTML)

Почему не WS-*?

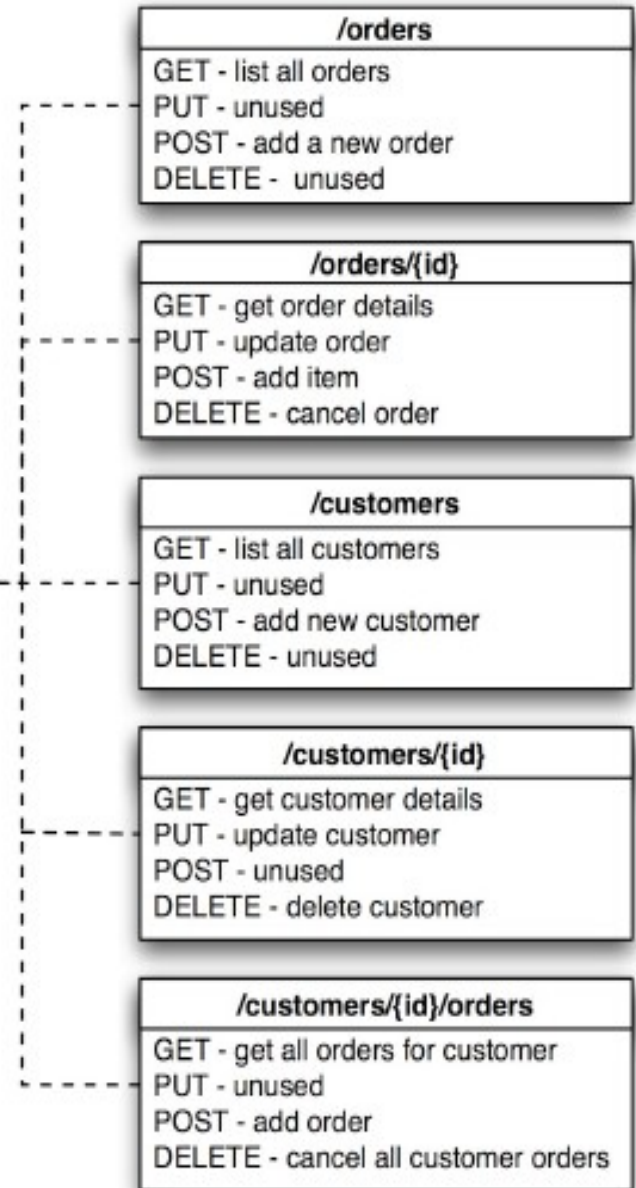
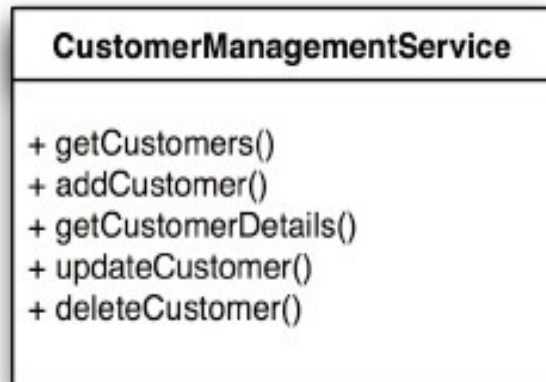
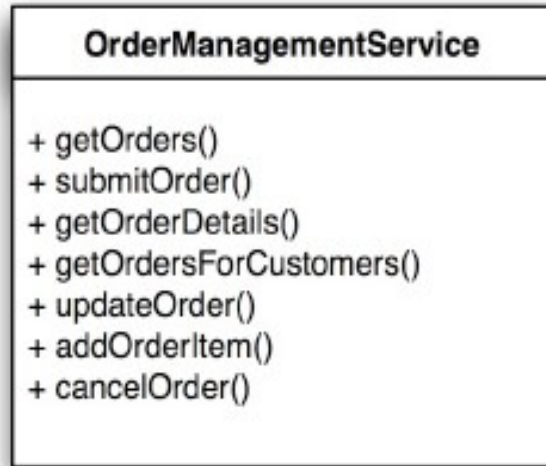
REST

- Representational State Transfer
- Архитектурный стиль, лежащий в основе HTTP (Roy Fielding, 2000)
- Определен через набор ограничений
 - Клиент-сервер
 - Состояние приложения хранит клиент
 - Кэширование
 - Многоуровневая архитектура
 - Унифицированный интерфейс

Унифицированный интерфейс HTTP

- Идентификация ресурсов с помощью URI
- Стандартные методы
 - GET: получить состояние ресурса (идемпотентный)
 - PUT: запись состояния ресурса (идемп.)
 - DELETE: удаление ресурса (идемп.)
 - POST: создание/расширение ресурса (не идемп.)
- Ресурс может иметь несколько представлений
 - Клиент и сервер согласуют формат представления

RPC vs REST



Возможности HTTP

- Кэширование
- Согласование форматов
- Перенаправление запросов
- Стандартные коды ответов
- Компрессия

Интерфейс сервиса MathCloud

Ресурс	GET	POST	DELETE
Сервис	Получение описания сервиса	Запрос к сервису	
Задание	Получение статуса и результатов задания		Отмена задания, удаление результатов задания
Файл результата задания	Получение файла		
Сервер	Получение списка сервисов, размещенных на сервере		

Форматы представления данных

Ресурс	GET	POST
Сервис	Accept: application/json text/html	Content-Type: application/json multipart/form-data Accept: application/json text/html
Задание	Accept: application/json text/html	
Файл результата задания	Content-Type: соотв. типу файла	
Сервер	Accept: application/json text/html	

Описание сервиса

- Имя
- Аннотация
- Список входных параметров
- Список выходных параметров
- Для описания параметров используется JSON Schema
 - имя, тип, аннотация, обязательный, значение по умолчанию, допустимые значения ...
 - <http://json-schema.org/>

Пример описания сервиса (JSON + JSON Schema)

```
{  
  "name": "echo",  
  "description": "Test service echoes back input params",  
  "inputs": {  
    "input1": {"type": "integer", "title": "Integer input"},  
    "input2": {"type": "string", "title": "String input",  
      "optional": true, "description": "Not required"},  
    "input3": {"type": "file", "title": "File input"}  
  },  
  "outputs": {  
    "output1": {"type": "integer", "title": "Integer output"},  
    "output2": {"type": "string", "title": "String output"},  
    "output3": {"type": "file", "title": "File output"}  
  }  
}
```

Типы параметров

- String
- Number
- Integer
- Boolean
- Object
- Array
- Any
- File

Допустимые значения параметра

- minimum
- maximum
- enum
- pattern
- format
- ...

Пример запроса к сервису

POST <http://somehost.com/echo>

Content-Type: application/json

```
{  
  "input1":1637673,  
  "input2":"some string",  
  "input3":{"uri":"http://otherhost.com/test.bin"}  
}
```

Пример ответа на запрос

202 Accepted

Location: <http://somehost.com/echo/job2706049717>

Content-Type: application/json

```
{  
  "state": "WAITING"  
}
```

Пример получения результата

```
GET http://somehost.com/echo/job2706049717  
Accept: application/json
```

```
200 OK  
Content-Type: application/json
```

```
{  
  "state": "DONE",  
  "result": {  
    "output1": 1637673,  
    "output2": "some string",  
    "output3":  
      {"uri":  
        "http://somehost.com/echo/job2706049717/out.bin"}  
      }  
  }  
}
```

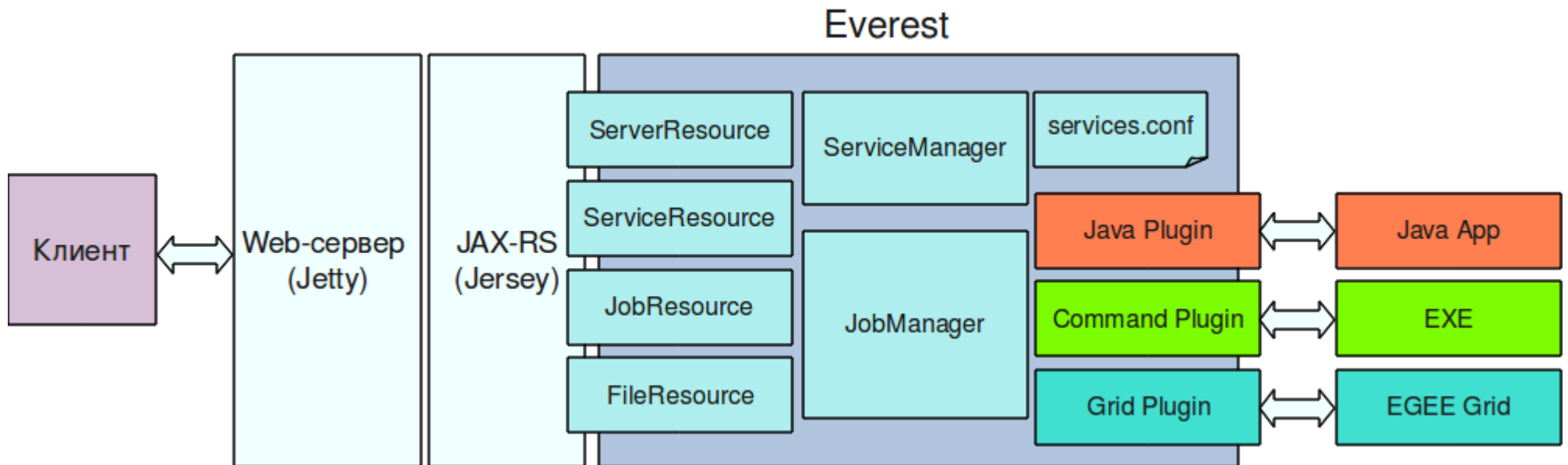

Возможные состояния задания

- **WAITING** - выполнение задания еще не началось
- **RUNNING** - задание выполняется
- **DONE** - задание выполнено успешно
- **FAILED** - выполнение задания завершилось ошибкой

Программные реализации

- “Обертка”
 - Perl
 - Преобразование в сервисы приложений с интерфейсом командной строки
- Everest
 - Java
 - Преобразование в сервисы: приложений с интерфейсом командной строки, Java-программ и грид-приложений
 - Поддержка реализации сервисов на Python

Архитектура Everest



command

```
{
  name: povray,
  description: "POV-Ray raytracer.",
  inputs: {
    scene: {type: file, title: "Scene file"},
    ini: {type: file, title: "Ini file", optional: true},
    params: {type: string, title: "Rendering parameters",
             optional:true, description: "Try +W320 +H240"}
  },
  outputs: {
    image: {type: file, title: "Rendered image"},
    log: {type: file, title: "Runtime log",
         description: "This is the raw output of POV-Ray"}
  },
  implementation: {
    type: command,
    command: "povray $ini +I$scene +Oimage.png +FN $params",
    input2file: {
      scene: scene.pov, ini: scene.ini
    },
    file2output : {
      image.png: image, stderr: log
    }
  }
}
```

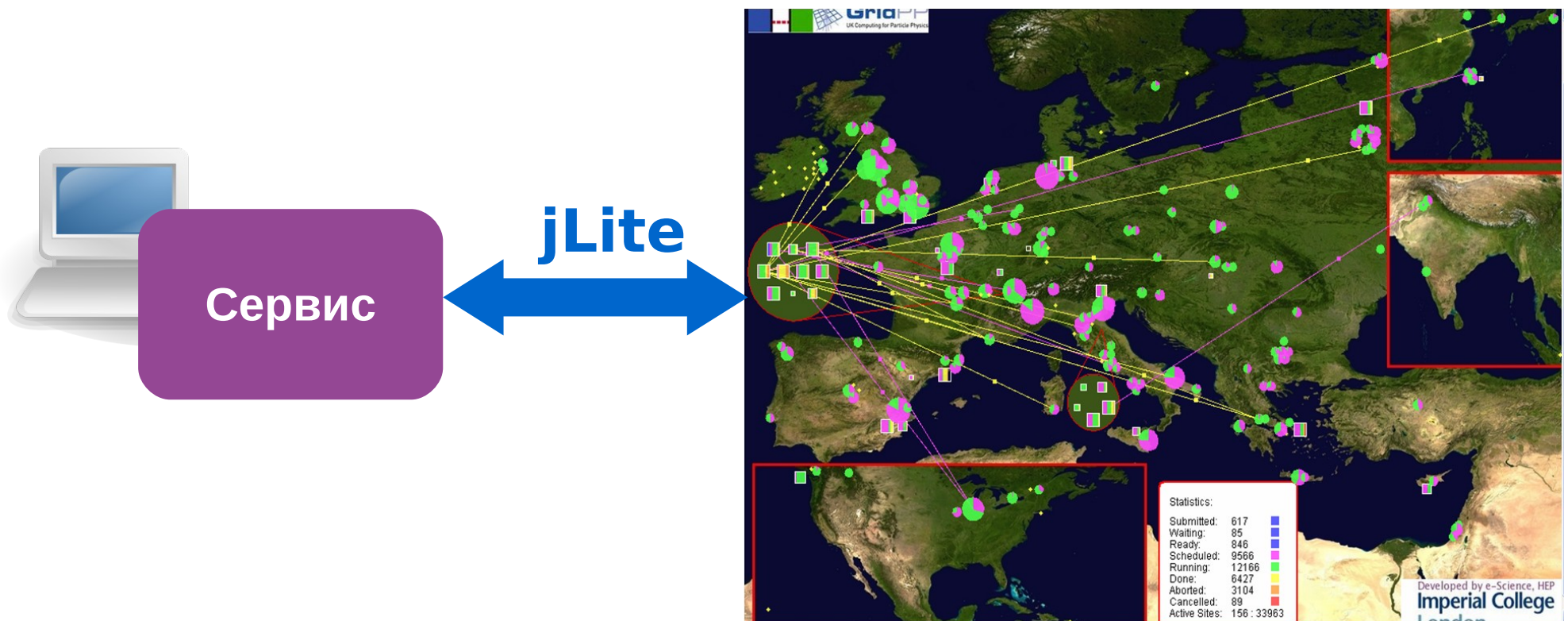
java

```
{
  name: fib,
  description: "Fibonacci numbers calculator.",
  inputs: {
    n: {type: integer, title: Number}
  },
  outputs: {
    fib: {type: integer, title: Fib(N)}
  },
  implementation: {
    type: java,
    class: org.mathcloud.everest.test.FibService
  }
}
```

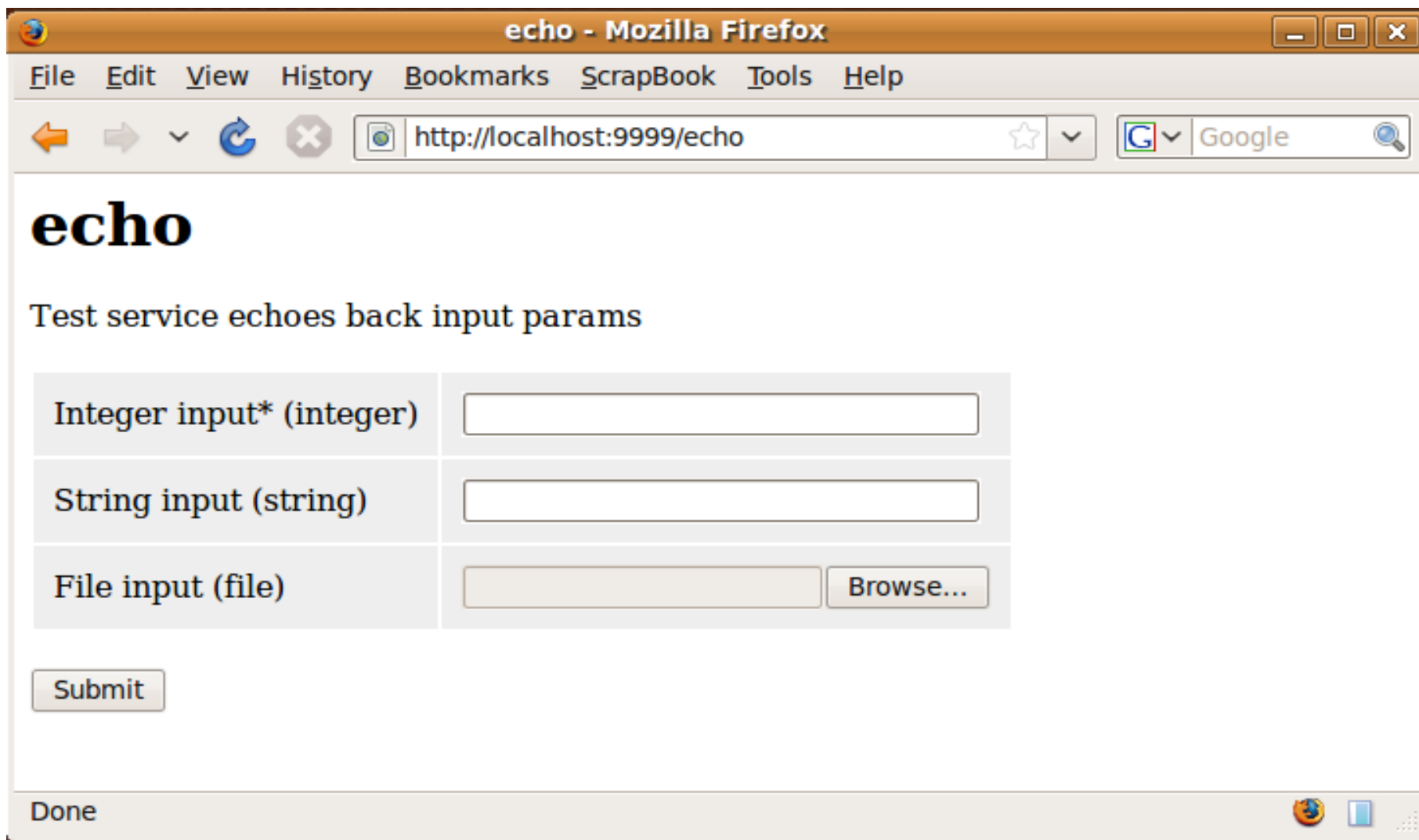
```
{
  name: gridtest,
  inputs: {
    executable: {type: string, title: Executable,
                 default: "/usr/bin/wc"},
    arguments:  {type: string, title: Arguments,
                 optional: true, default: "-w input"},
    input_file: {type: file,   title: "Input file",
                 description: "Saved to file 'input'"}
  },
  outputs : {
    stdout: {type: file, title: StdOutput},
    stderr: {type: file, title: StdError}
  },
  implementation : {
    type: glite,
    vo: gilda,
    job-file: "services/gridtest/test.jdl",
    input2file: {input_file: input},
    file2output: {
      stdout.txt: stdout, stderr.txt: stderr
    }
  }
}
```

Интеграция с грид

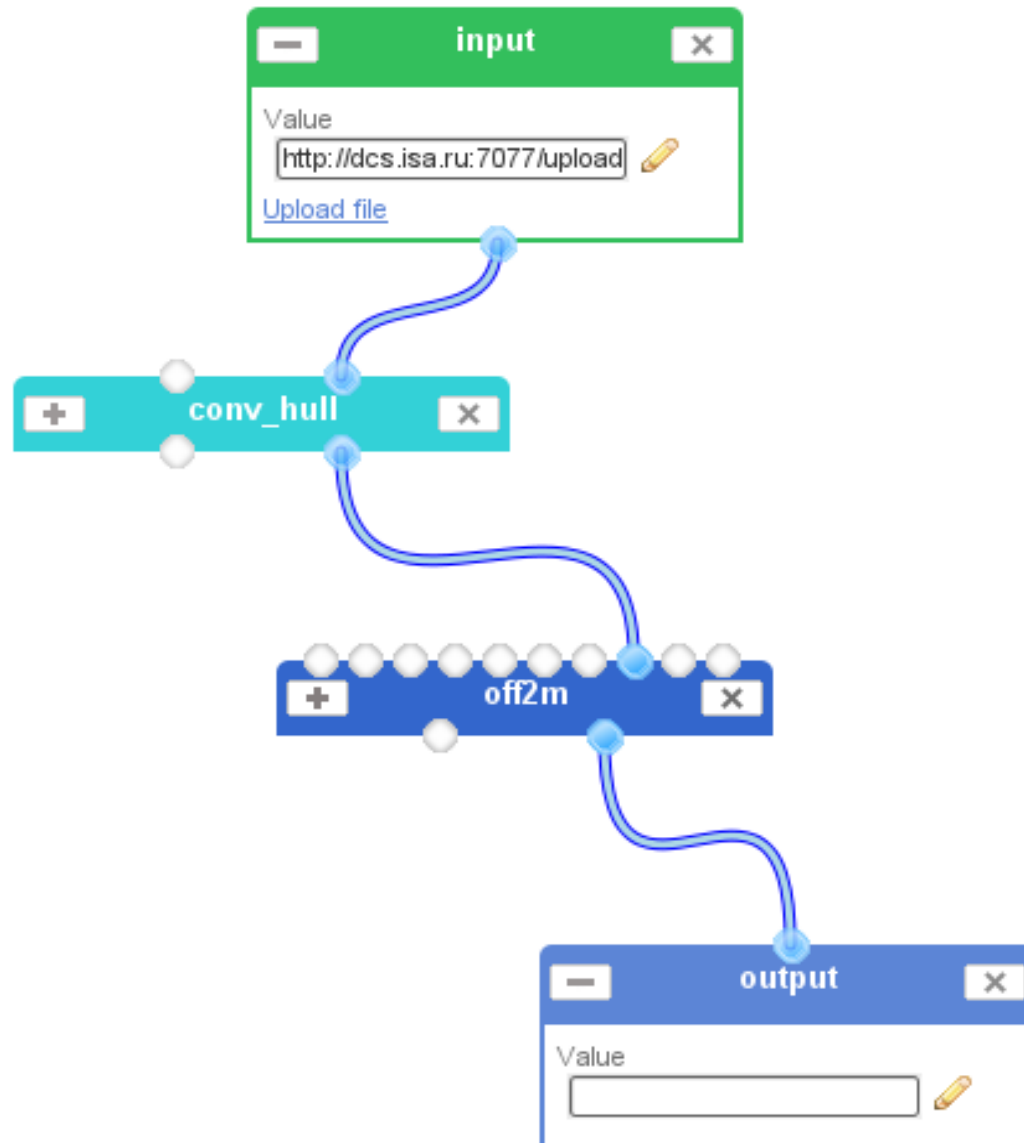
- jLite - переносимый программный интерфейс для запуска вычислительных заданий в грид-инфраструктуре EGEE



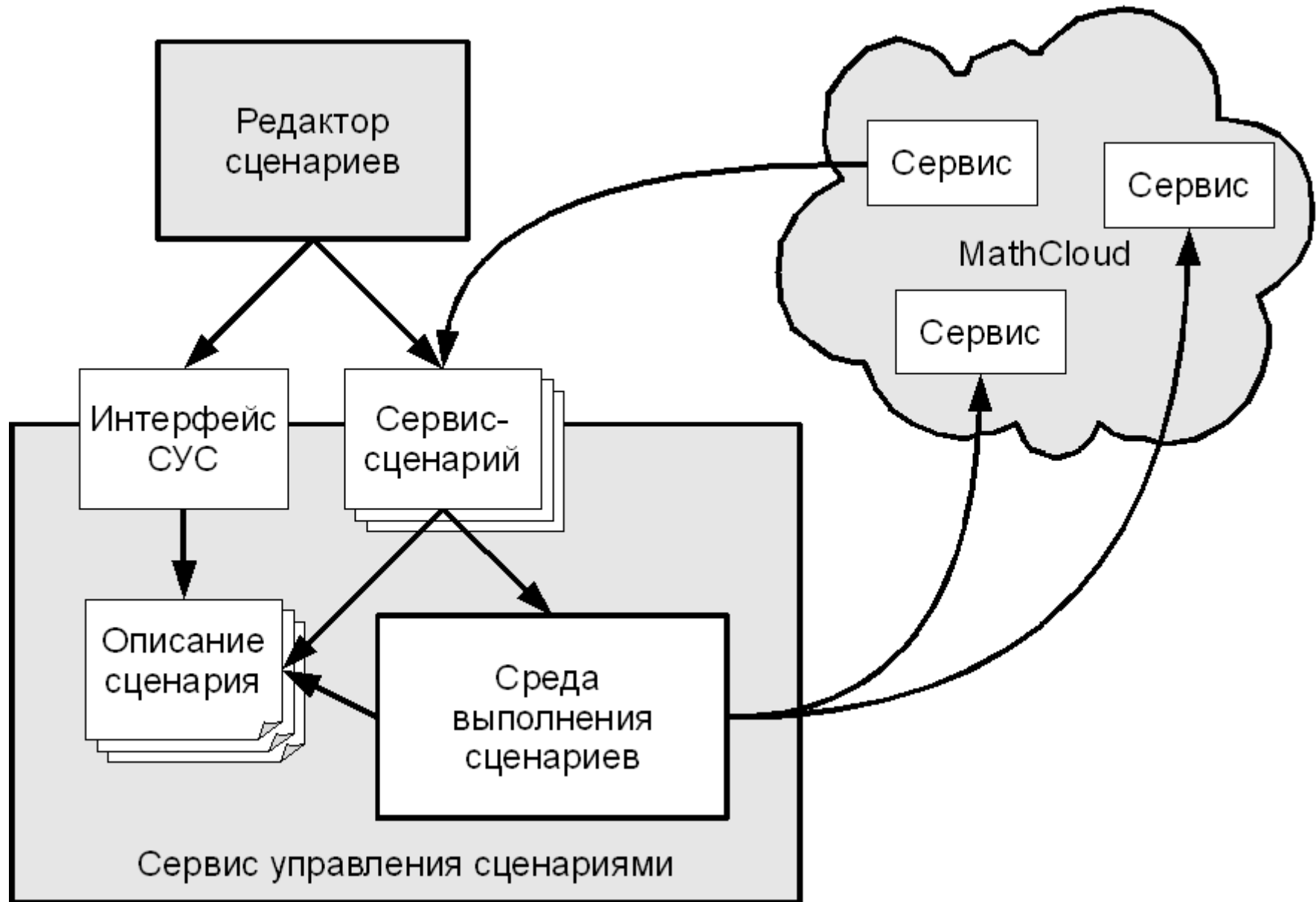
Генерация Web-интерфейса



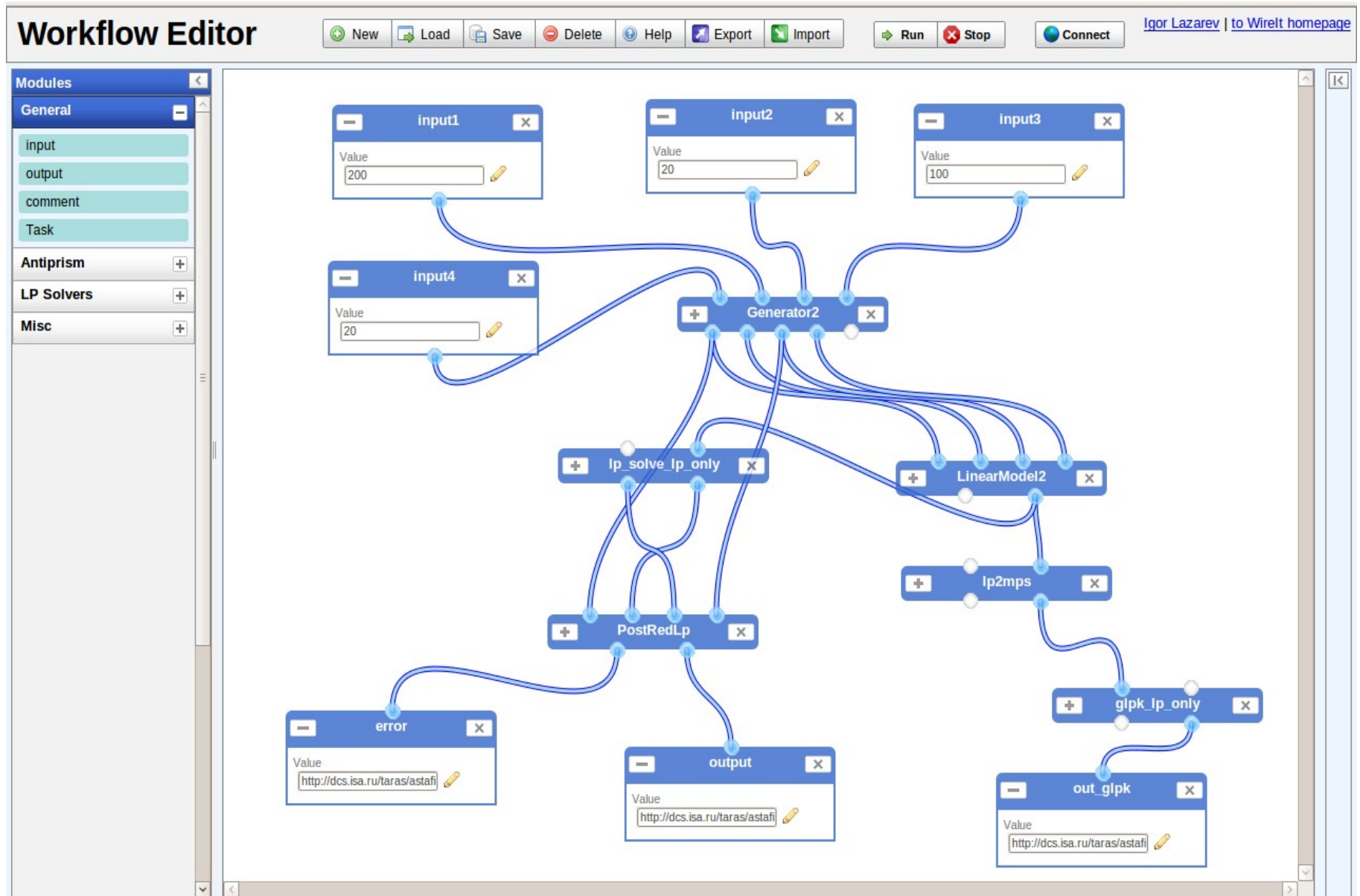
Композиция сервисов



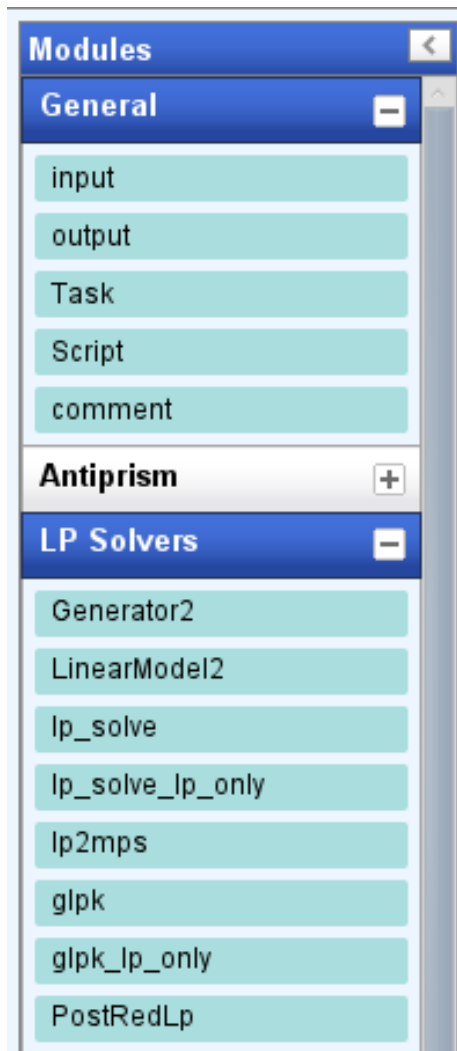
Сервис управления сценариями



Workflow-редактор

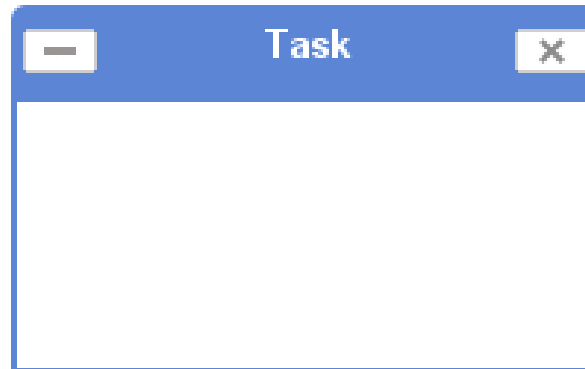


Список блоков

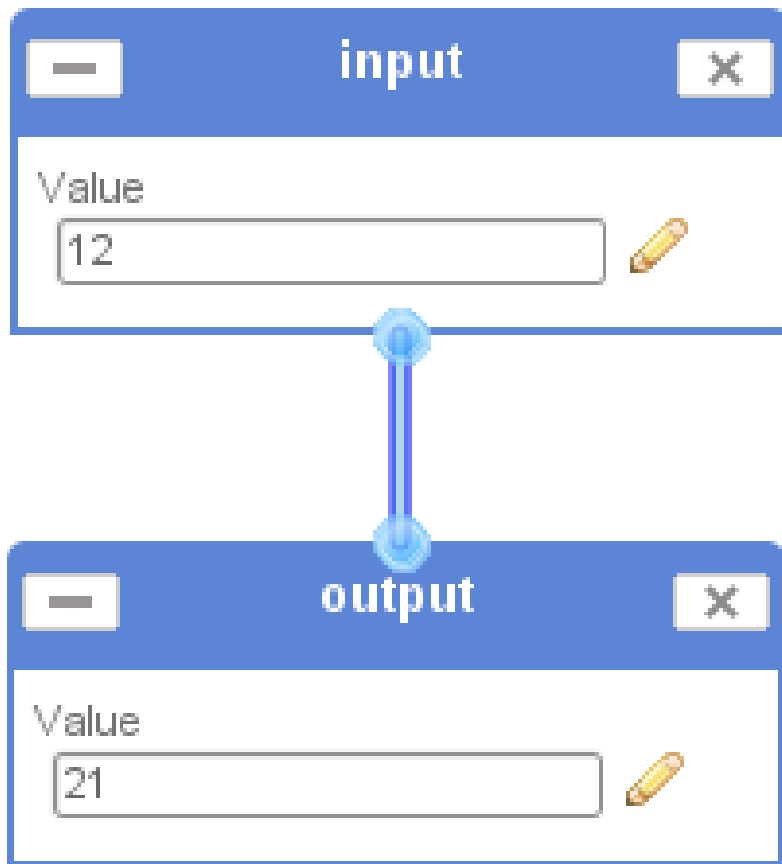


- Входы и выходы
- Задания (сервисы)
- Скрипты
- Комментарии
- Сервисы, сгруппированные по категориям
- Сервисы-сценарии

Блок Task

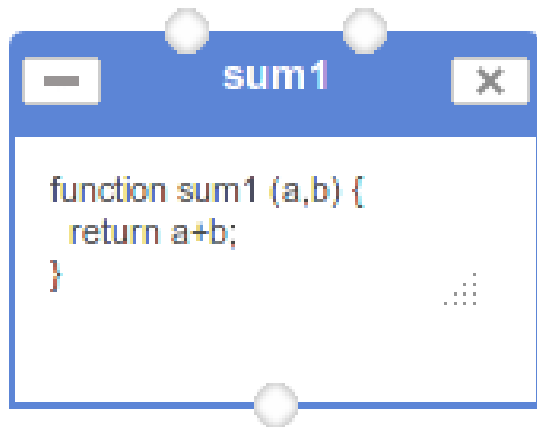


Блоки Input/Output

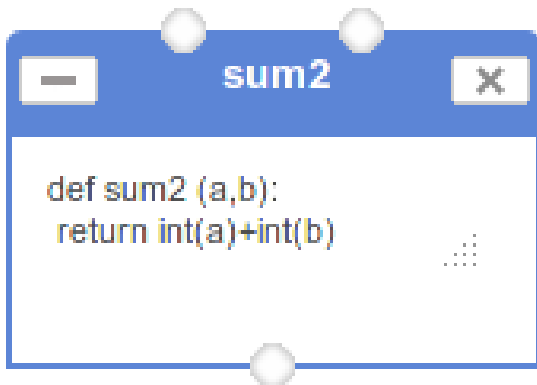


- Ввод входных параметров
- Загрузка файлов
- Вывод конечных результатов
- Визуализация результатов
- Генерируются автоматически

Блоки Script/Python



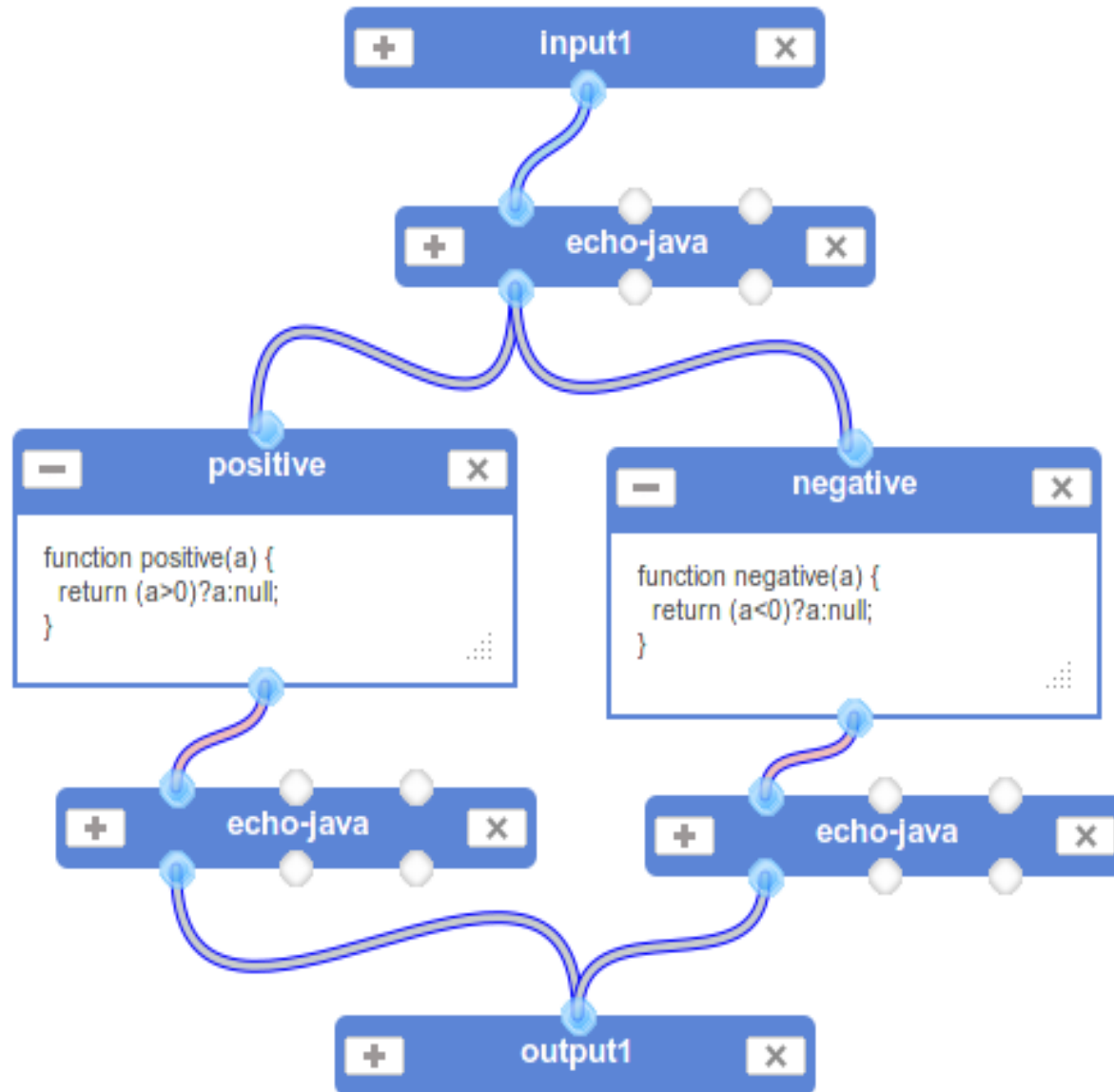
```
function sum1 (a,b) {  
  return a+b;  
}
```



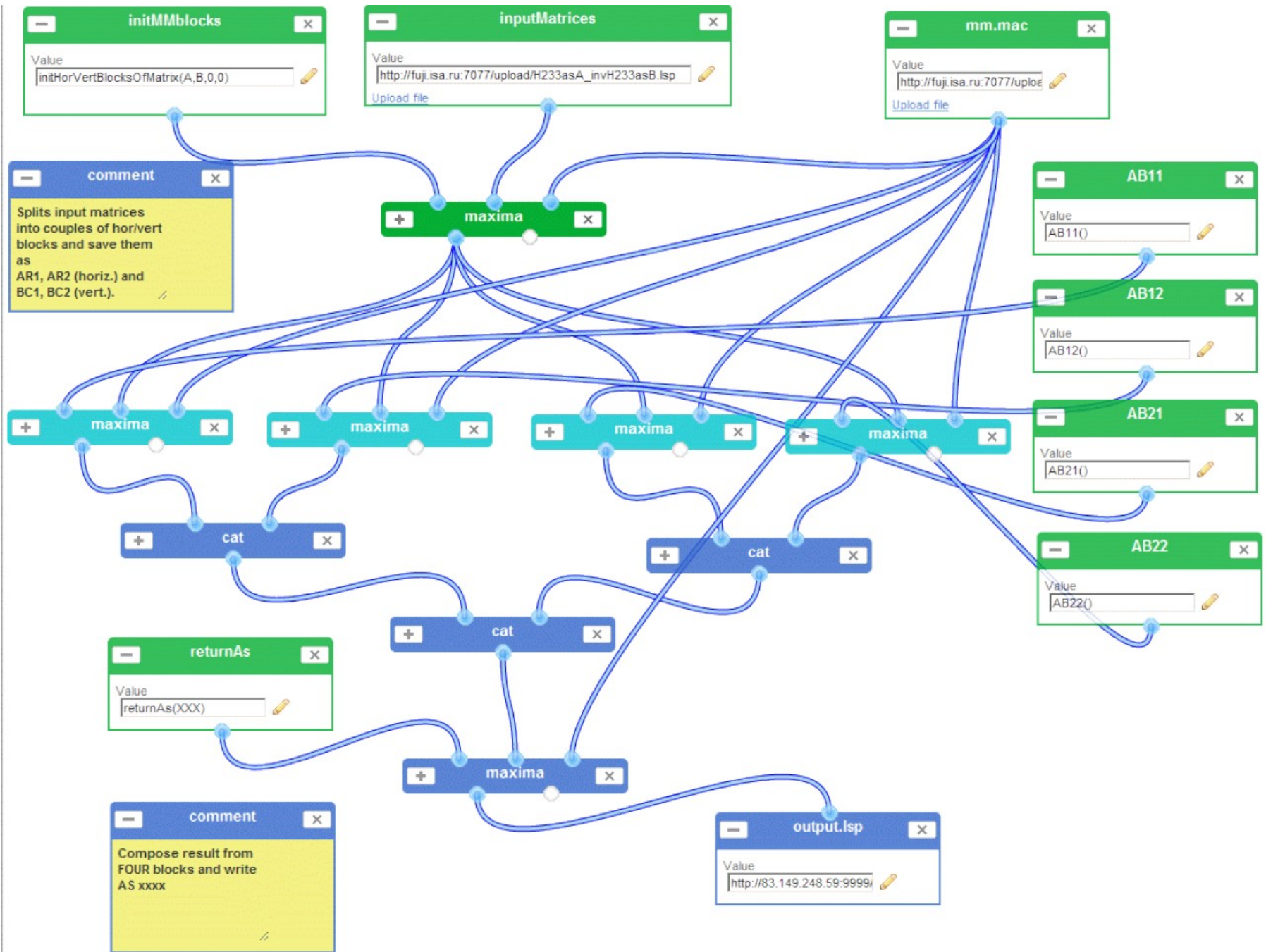
```
def sum2 (a,b):  
  return int(a)+int(b)
```

- Функции на JavaScript и Python
- Вносят большую гибкость в сценарий
- Порты, название блока генерируются на основе сигнатуры функции
- Скрипт выполняется на серверной стороне

Условное ветвление



Выполнение сценария



REST API

Ресурс	GET	POST	PUT	DELETE
Сервис СУС	Получение списка хранимых сценариев	Создание нового сценария	—	—
Сценарий	Получение описания сценария	—	Обновление сценария	Удаление сценария
Сервис сценария	Получение описания сервиса	Запуск сценария	—	—
Прокси	Получение описания сервиса с указанным URI	—	—	—

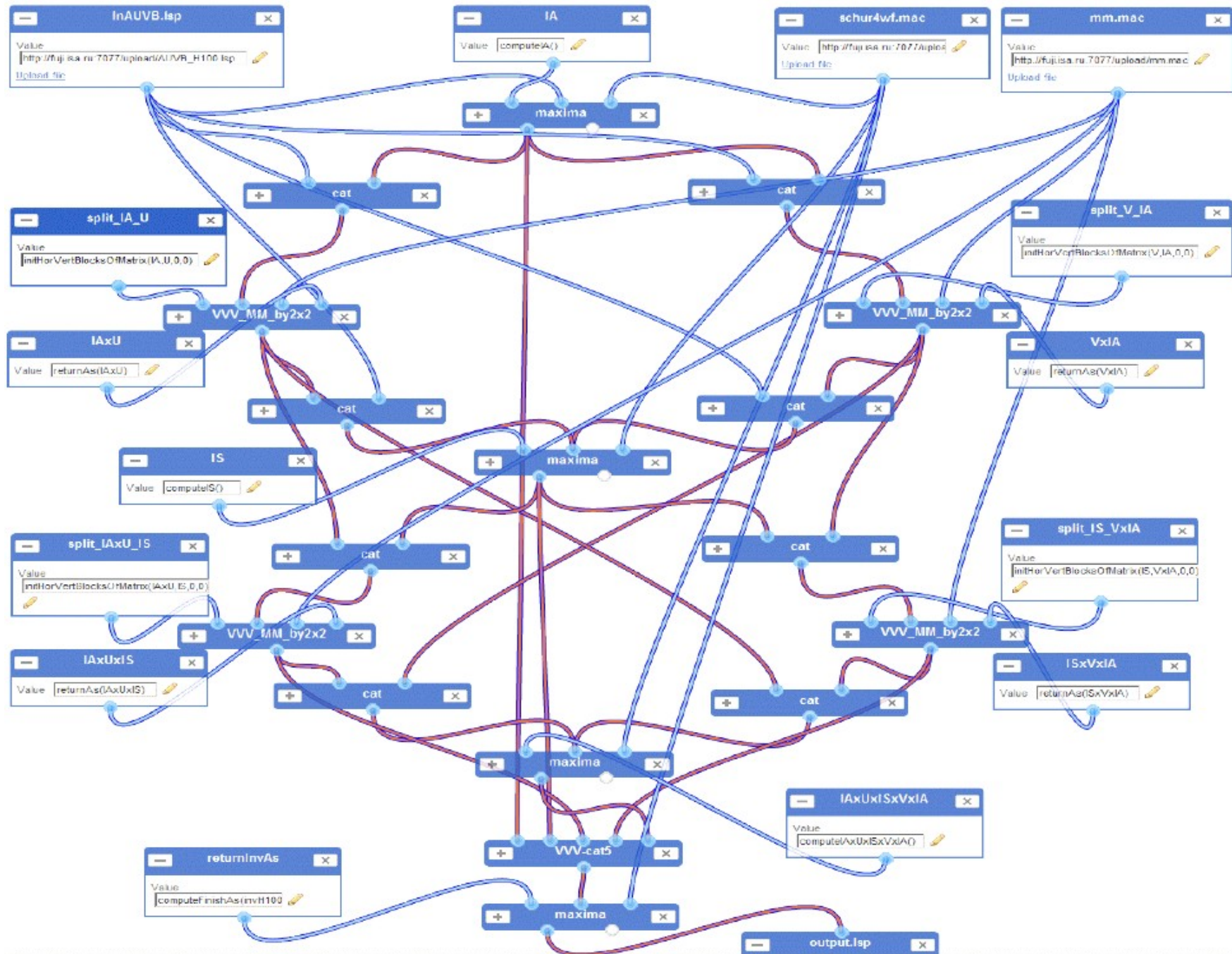
Безопасность

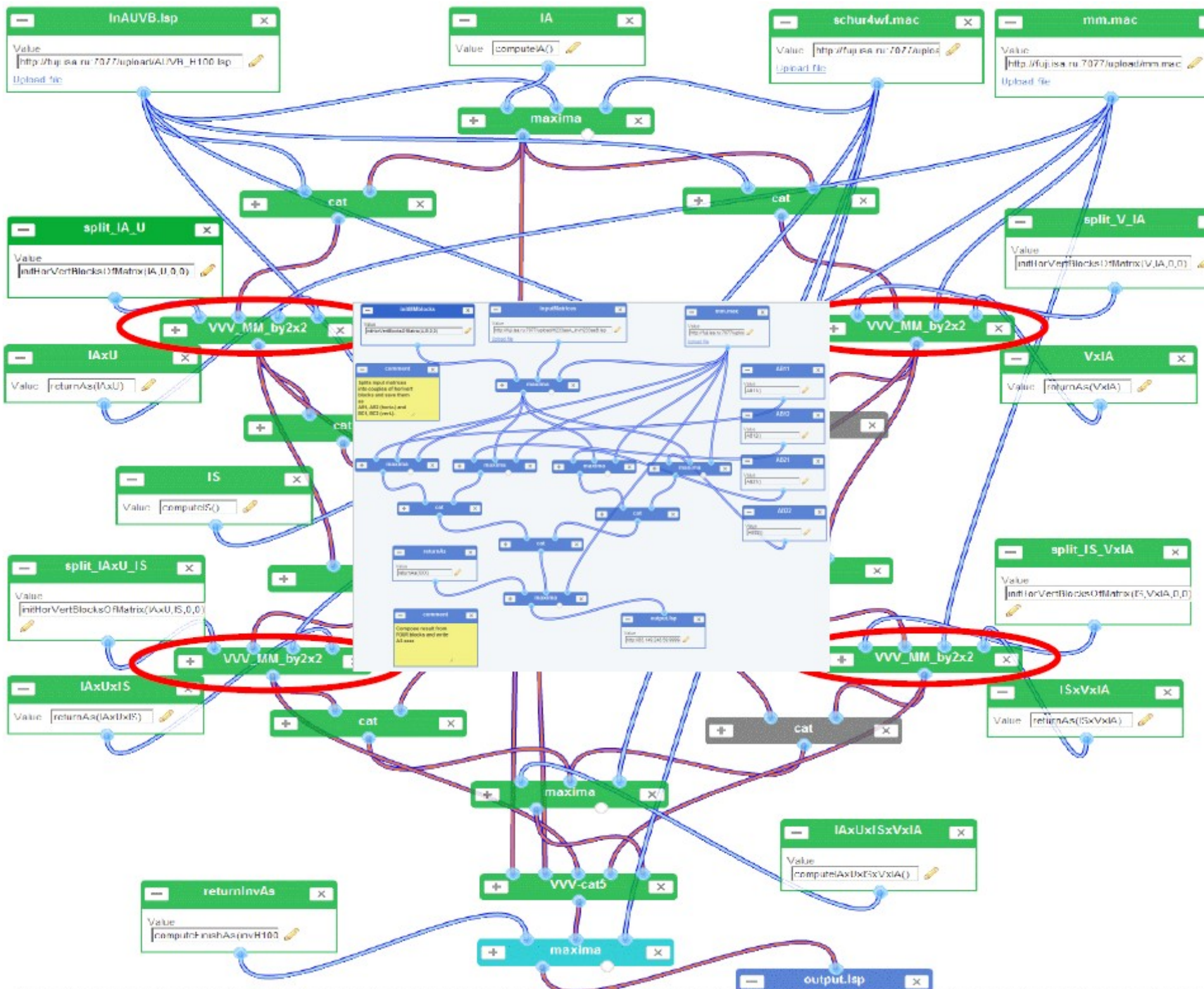
- Аутентификация сервера и защита данных
 - Цифровые сертификаты
 - Протокол HTTPS
- Аутентификация пользователей
 - Цифровые сертификаты
 - Публичные провайдеры учетных записей
 - OpenID, Google, Yandex, Facebook, Vkontakte...
 - Только для браузерного интерфейса

Приложения

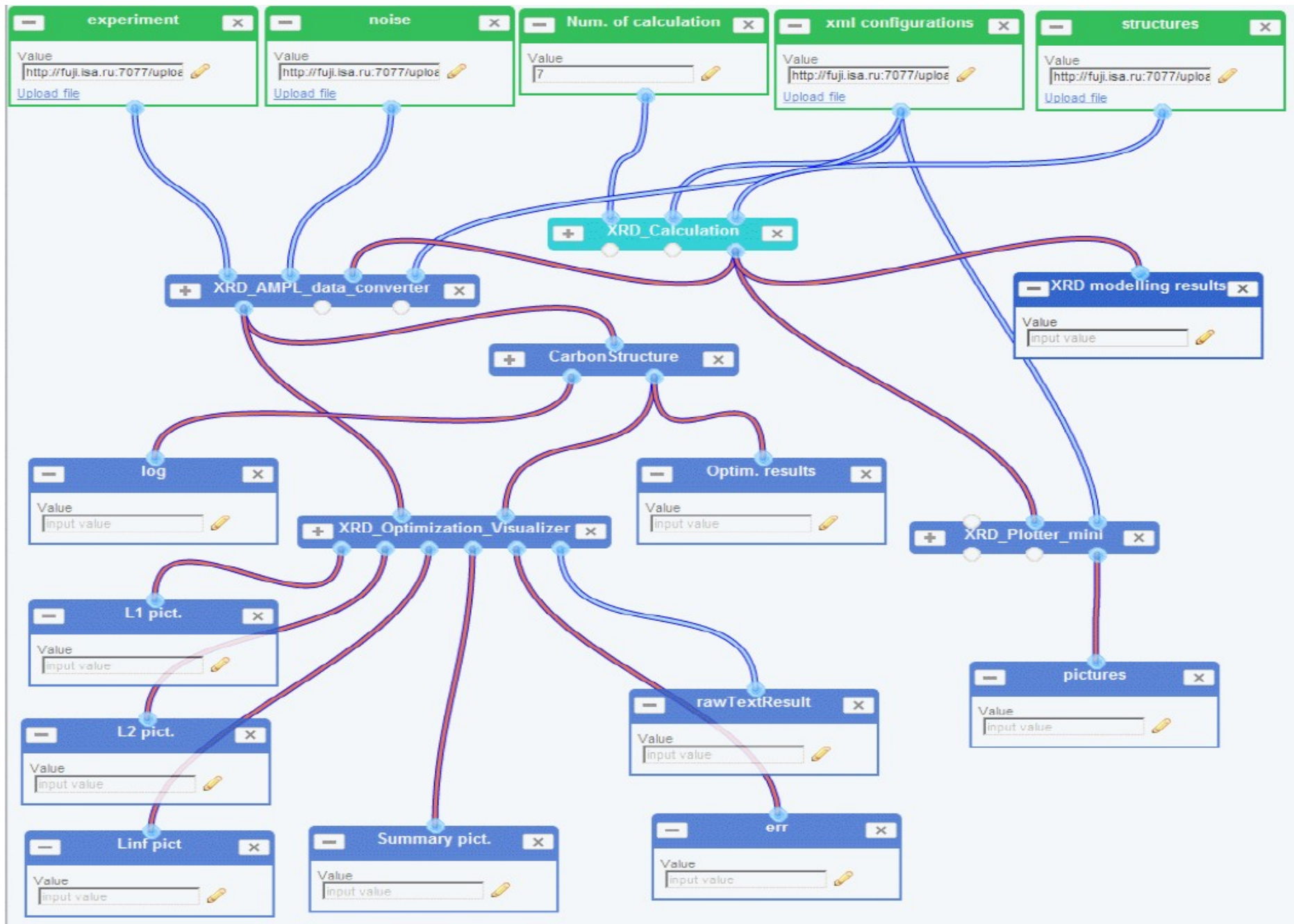
- Безошибочные вычисления в среде сервисов компьютерной алгебры
 - V.V. Voloshinov, S.A. Smirnov
Error-free Inversion of Ill-conditioned Matrices in Distributed Computing System of RESTful services of Computer Algebra
- Моделирование рентгеновской дифракции на углеродных наноструктурах и определение их состава в осаждённых плёнках из ТОКАМАКА T-10
 - V.S. Neverov, V.V. Voloshinov, A.P. Afanasiev, A.B. Kukushkin, N.L. Marusov, I.B. Semenov, V.G. Stankevich, N.Yu. Svechnikov, A.S. Tarasov, A.A. Veligzhanin, Ya.V. Zubavichus
Distributed Computing and Optimization Algorithms for Interpretation of X-ray Scattering by Carbon Nanostructures in the Deposited Films from TOKAMAK T-10

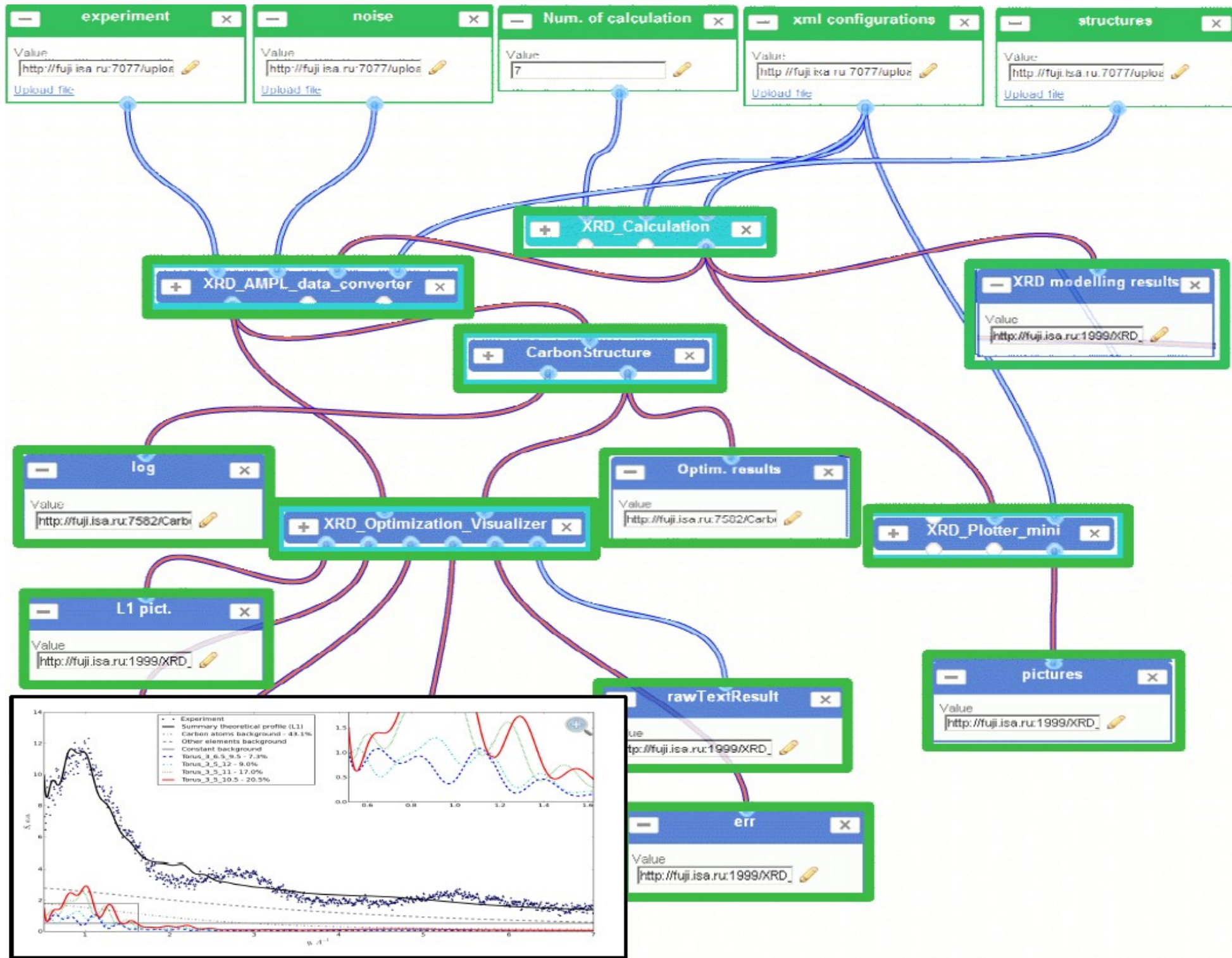
Безошибочное обращение матриц Гильберта





Анализ наноструктур углерода





Дальнейшее развитие

- Каталог сервисов
- Безопасность (авторизация, делегация)
- Управление группами пользователей (ВО)
- Плагины к вычислительным ресурсам и инфраструктурам
- Хостинг сервисов

Дополнительная информация

- Публикации

<https://www.assembla.com/wiki/show/mathcloud/Publications>

- Открытая разработка

- <https://www.assembla.com/spaces/mathcloud/>

- Apache License, Version 2.0

Спасибо за внимание