

Реализация проблемно-ориентированных вычислительных сервисов в среде MathCloud

О.В. Сухорослов

Центр грид-технологий и распределенных
вычислений ИСА РАН

Московский физико-технический институт

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.

Композиция сервисов

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

Мотивация

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

Технологический барьер

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

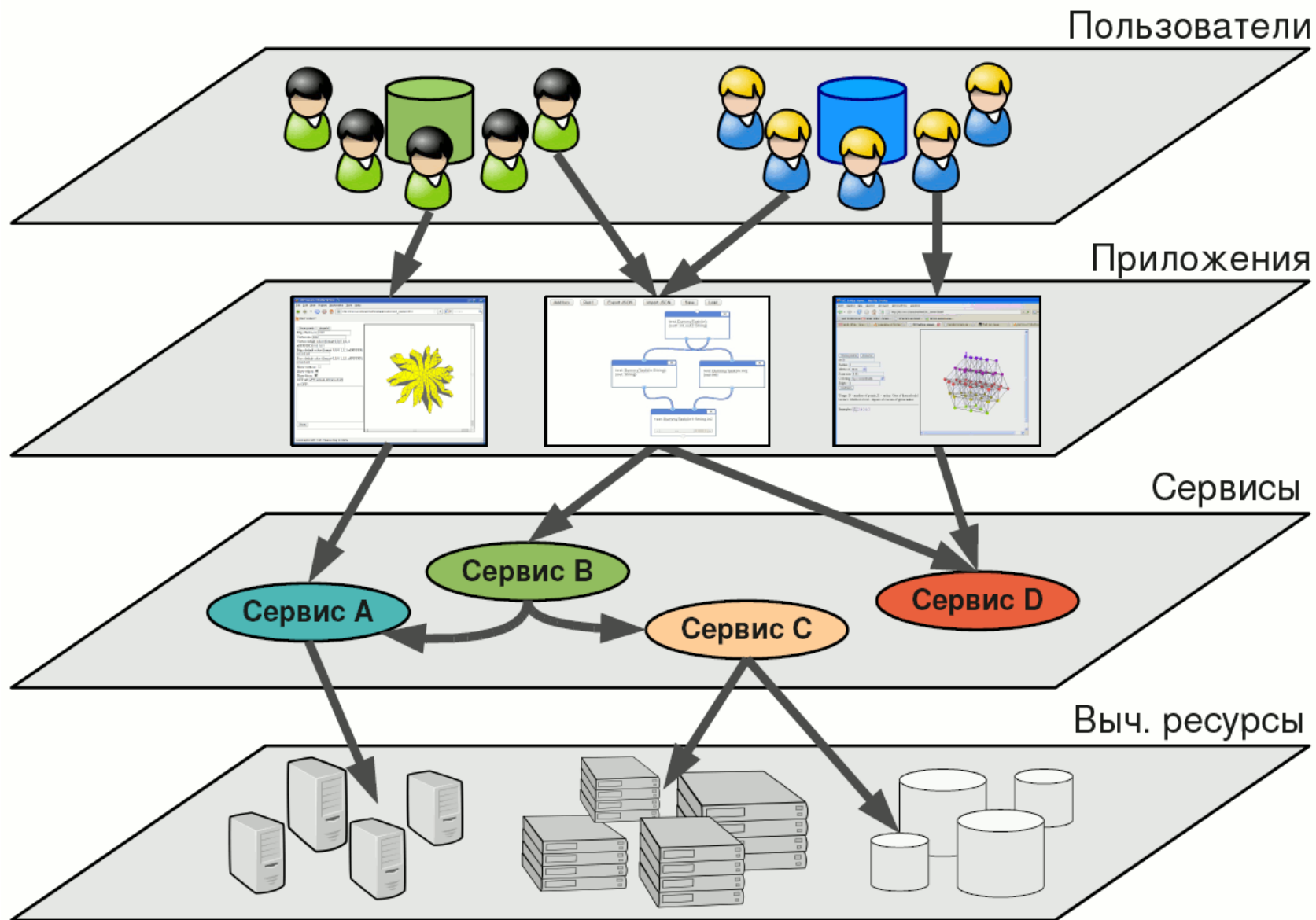
Технологический барьер

- Пользователь сервиса
 - Унификация доступа к сервисам
 - Привычный и понятный интерфейс
 - Отсутствие необходимости вникать в техническую сторону функционирования сервиса
 - Единое “рабочее пространство”

MathCloud

- Сервис-ориентированная научная среда
 - предоставление унифицированного доступа к проблемно-ориентированным вычислительным сервисам
 - поддержка композиции данных сервисов при решении прикладных задач
 - ... в рамках виртуальных сообществ поставщиков и пользователей сервисов (“распределенное облако сервисов”)
- Требования
 - Простота, легкость использования, открытость
- Базируется на веб- и грид-технологиях

Архитектура MathCloud



Сервис MathCloud

- 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)

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

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": "myservice",  
  "description": "My service",  
  "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"}  
  }  
}
```

Пример запроса к сервису

```
POST http://somehost.com/myservice  
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/myservice/job2706049717>

Content-Type: application/json

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

Пример получения результата

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

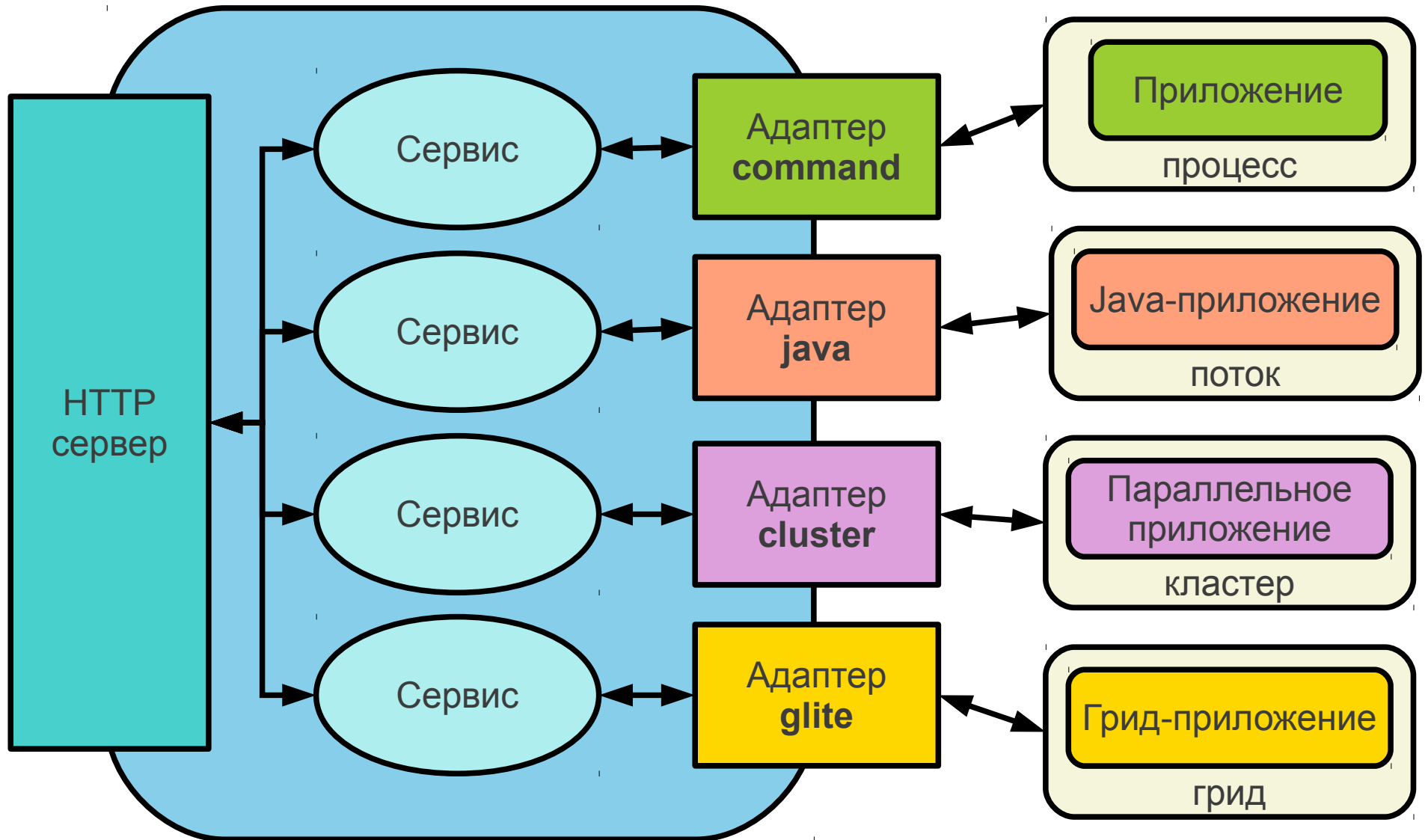
```
200 OK  
Content-Type: application/json
```

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

Программные реализации (среды выполнения сервисов)

- “Обертка”
 - Perl
 - Преобразование в сервисы приложений с интерфейсом командной строки
- Everest
 - Java
 - Механизм подключаемых адаптеров
 - преобразование запроса к сервису в вызов внешнего приложения или запуск приложения в вычислительной инфраструктуре

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: {
  adapter: 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,
        description: "Maximum value: 70"}
  },
  outputs: {
    fib: {type: integer, title: Fib(N)}
  },
  implementation: {
    adapter: java,
    class: websolve.everest.test.FibService
    config: {maxN: 70}
  }
}
```

cluster

```
{
  name: clustertest,
  description: "Submits job to cluster via cluster adapter",
  inputs: {
    sleep_time: {type: integer, title: "Sleep time in
                  seconds", default: 60},
    wc_input:   {type: file, title: "Input file for wc
                  command" }
  },
  outputs : {
    stdout: {type: file, title: StdOutput},
    stderr: {type: file, title: StdError}
  },
  implementation : {
    adapter: cluster,
    job-file: "services/clustertest/test.job",
    input2file: { wc_input: wc_input },
    iile2output: { stdout: stdout, stderr: stderr }
  }
}
```

```
{
  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
    }
  }
}
```

Генерация веб-интерфейса

Neverov-XRD-Cluster-calc - Mozilla Firefox

File Edit View History Bookmarks ScrapBook Tools Help

Neverov-XRD-Cluster-calc

isa.ru http://isa.ru/neverov-xrd-cluster-calc/ Google

Neverov-XRD-Cluster-calc

XRD MPI calculation on cluster

⚠ Number of Nodes* [integer] Each node has 4 threads

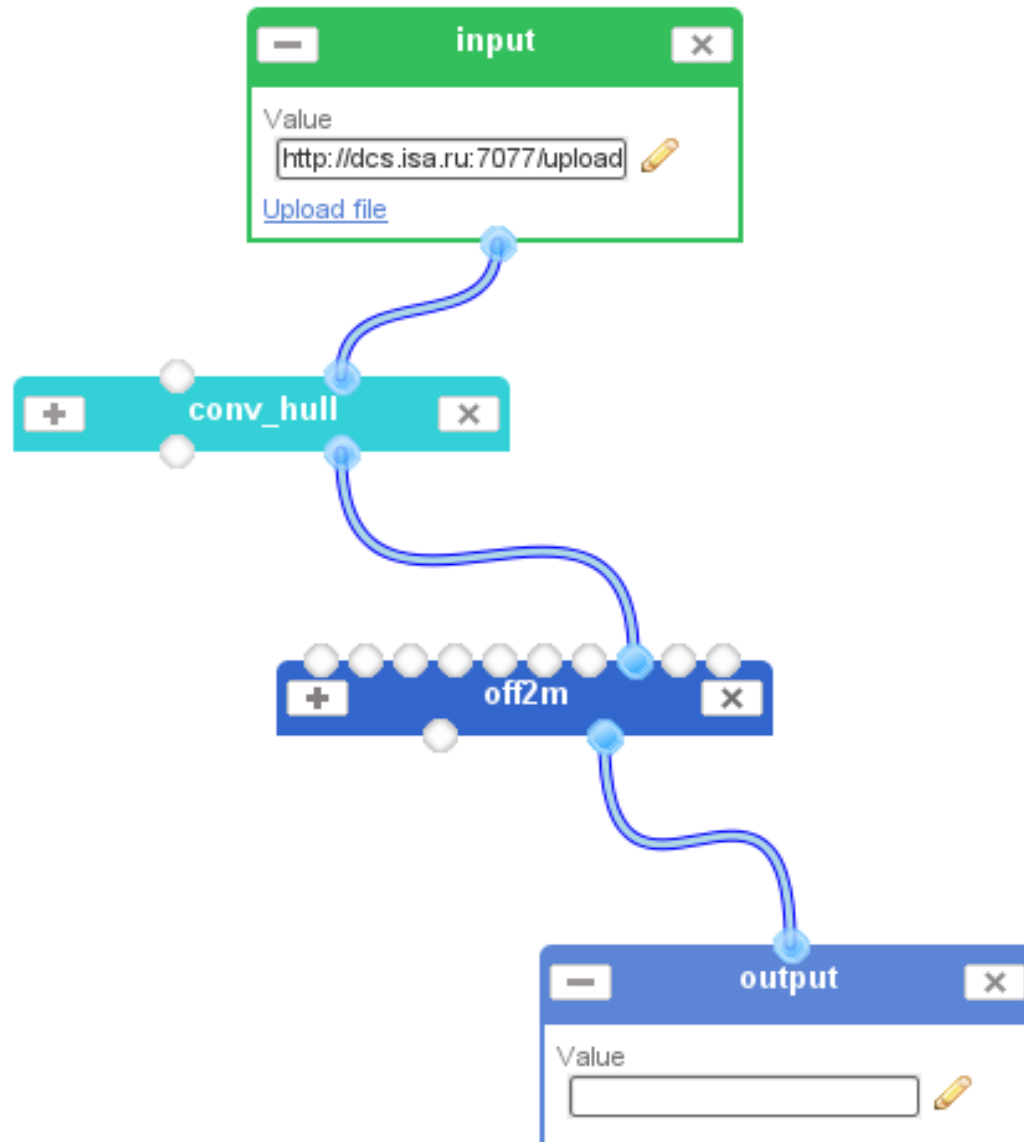
Config xml* Browse... [file] Single xml file

Structures archive* Browse... [file] tar.gz archive

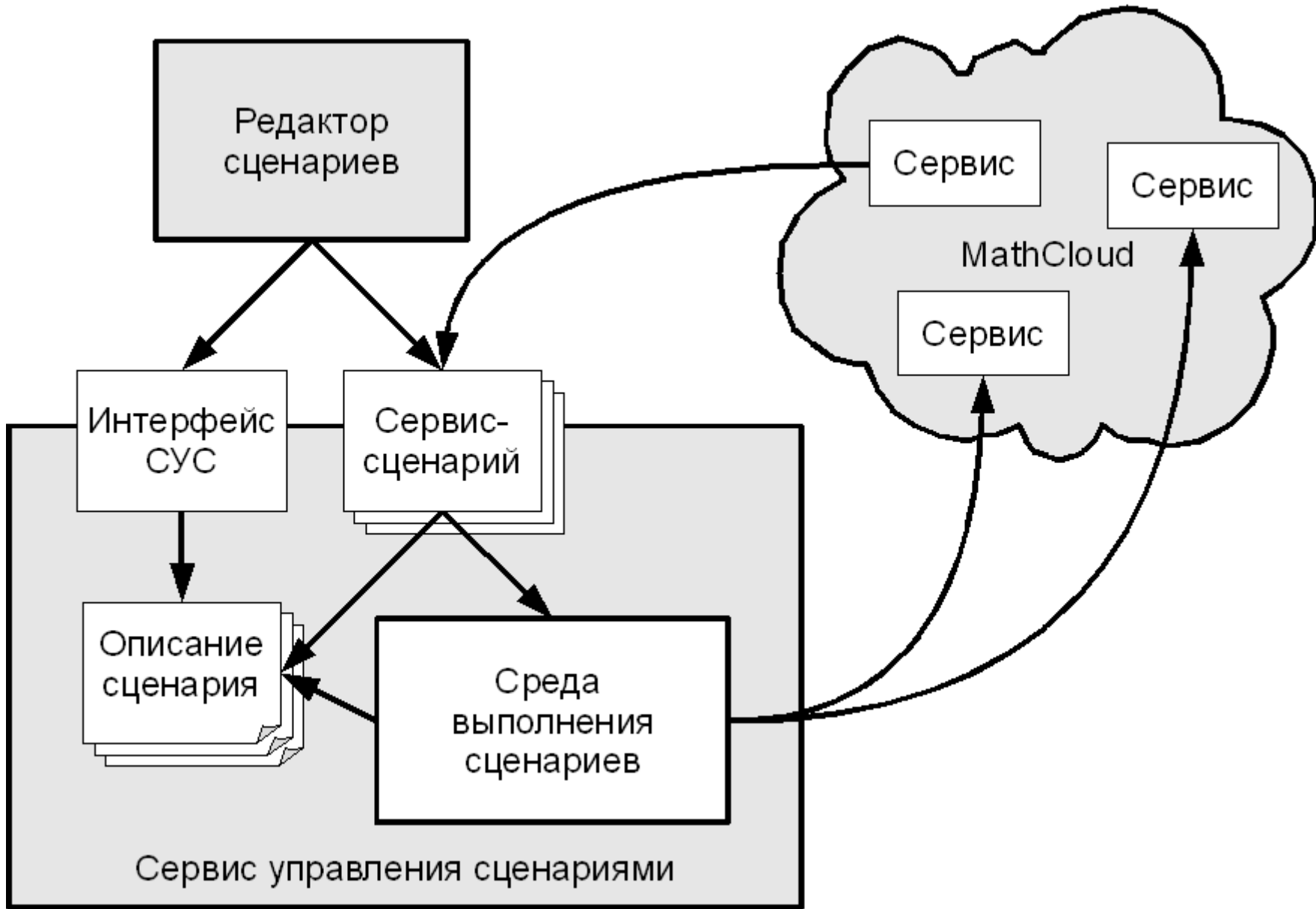
Command line options [string] Add parameters here.

Submit

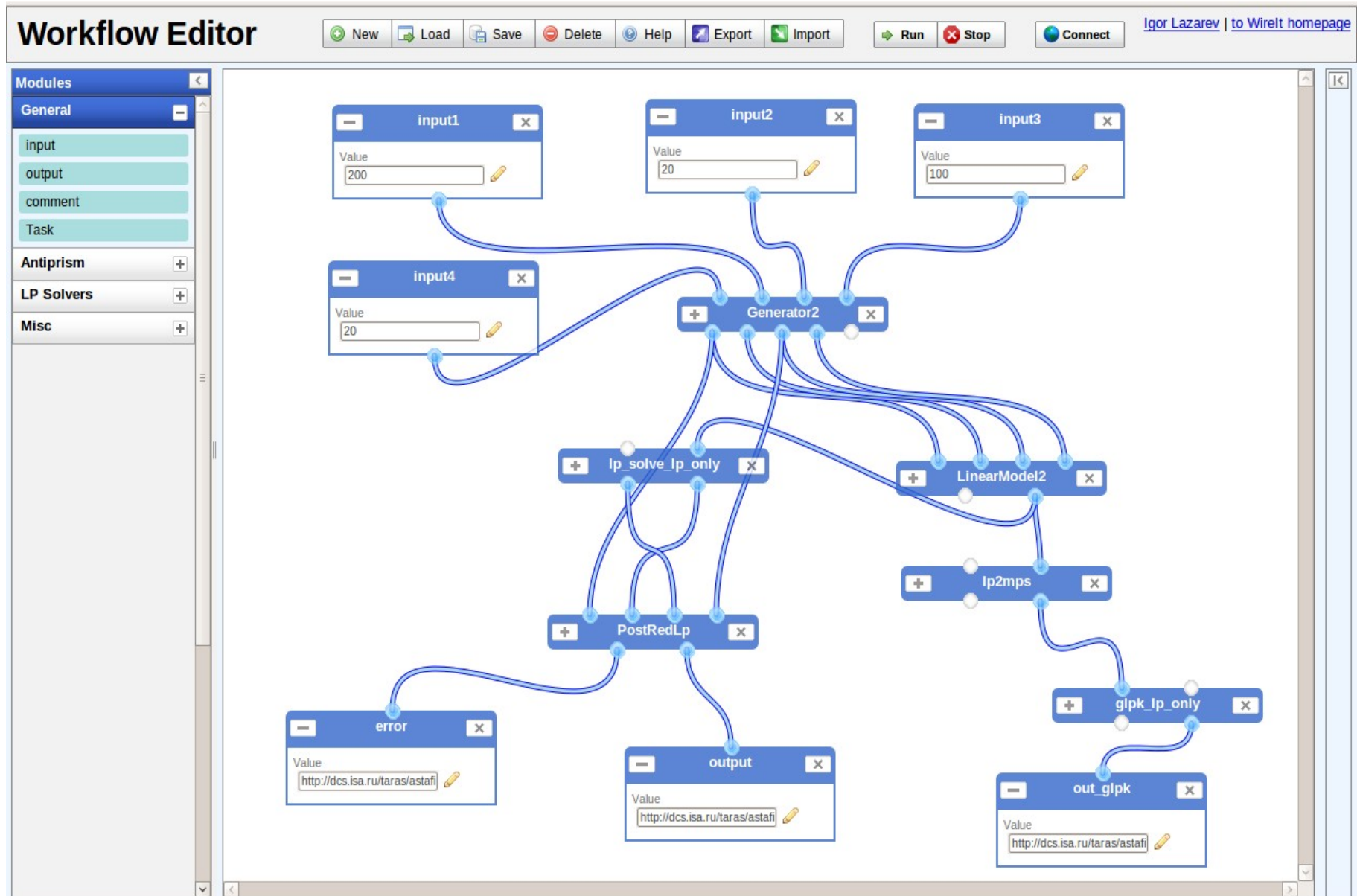
Композиция сервисов



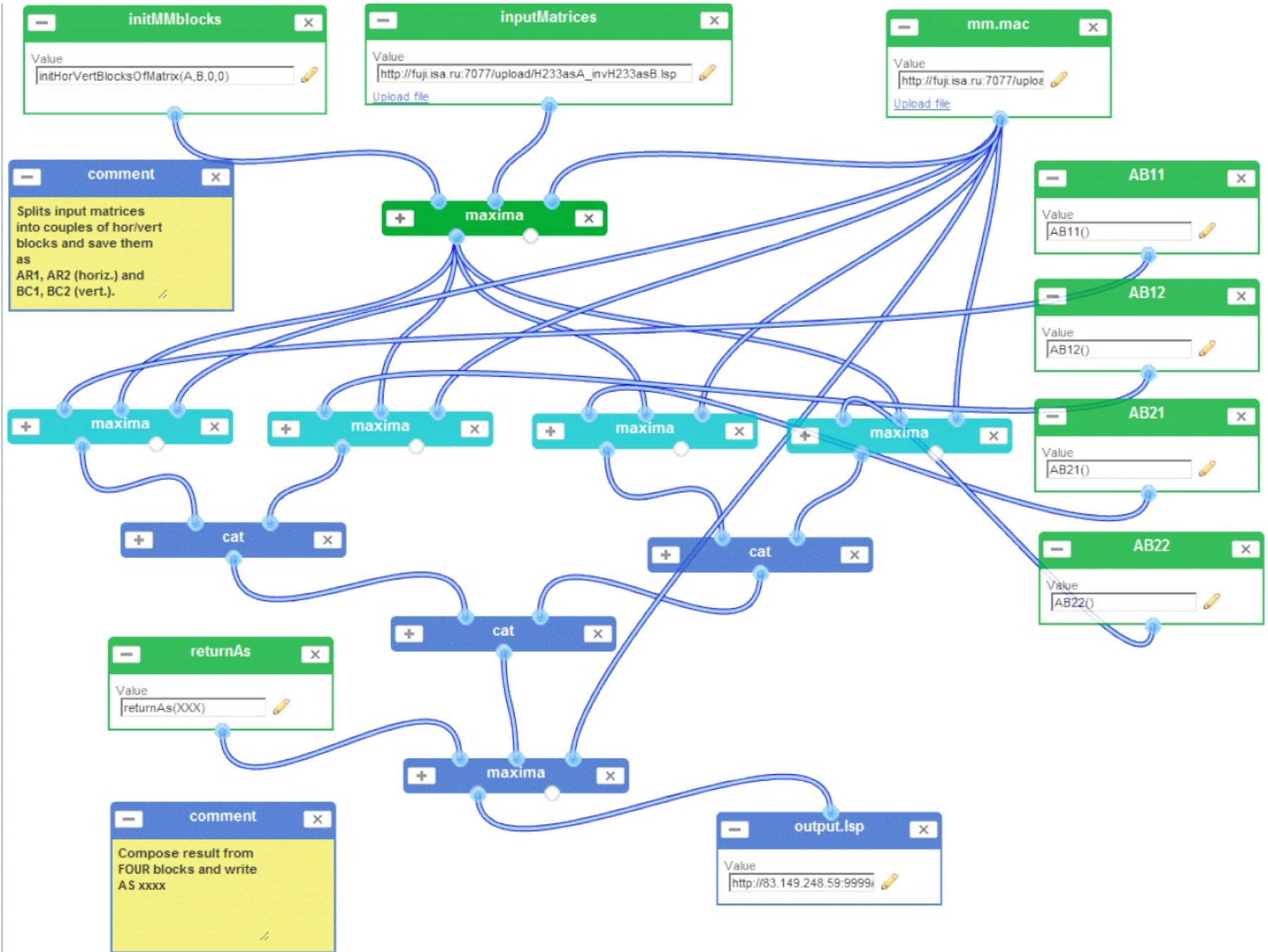
Система управления сценариями (WfMS)



Workflow-редактор



Выполнение сценария



Безопасность

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

Дальнейшее развитие

- Программный инструментарий WebSolve
 - <http://websolve.googlecode.com>
- Каталог сервисов
- Безопасность (авторизация, делегирование прав, управление группами пользователей)
- Командные и программные интерфейсы
- Масштабируемая инфраструктура для размещения сервисов

Спасибо за внимание