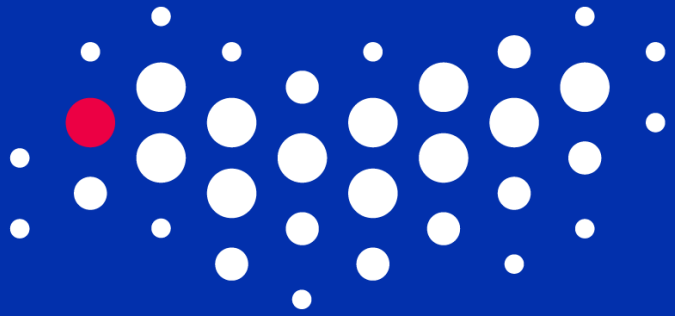


ITMO UNIVERSITY

CLAVIRE Highlights

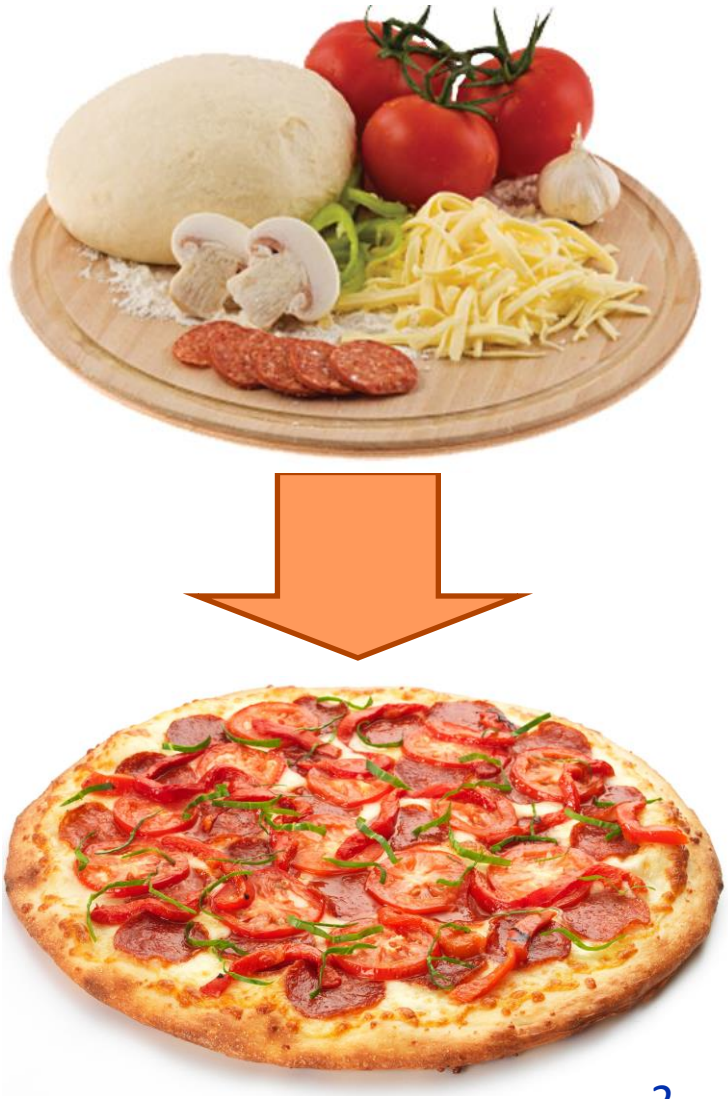
A. Boukhanovsky, D. Nasonov , S. Kovalchuk, K. Knyazkov, T. Chorov, A. Larchenko, S. Mariyn, N. Butakov

Big Data processing and analysis challenges in mega-science experiments, Dubna, Russia

Motivation

Common problems in HPC:

- Resources heterogeneity
- Packages (software) heterogeneity
- Resource and package high cohesion
- Wide user variation needs, lots of stakeholders
- Multi-level diversity of components interactions (users, packages, resources)
- Hard access and low-effective representation of complex experiments
- etc.



CLAVIRE = Cloud Applications VIRtual Environment

CLAVIRE is instrumental and technological environment that is designed to provide complete life cycle support of computation and data intensive systems (prototyping, developing, modernization) in various domain areas.

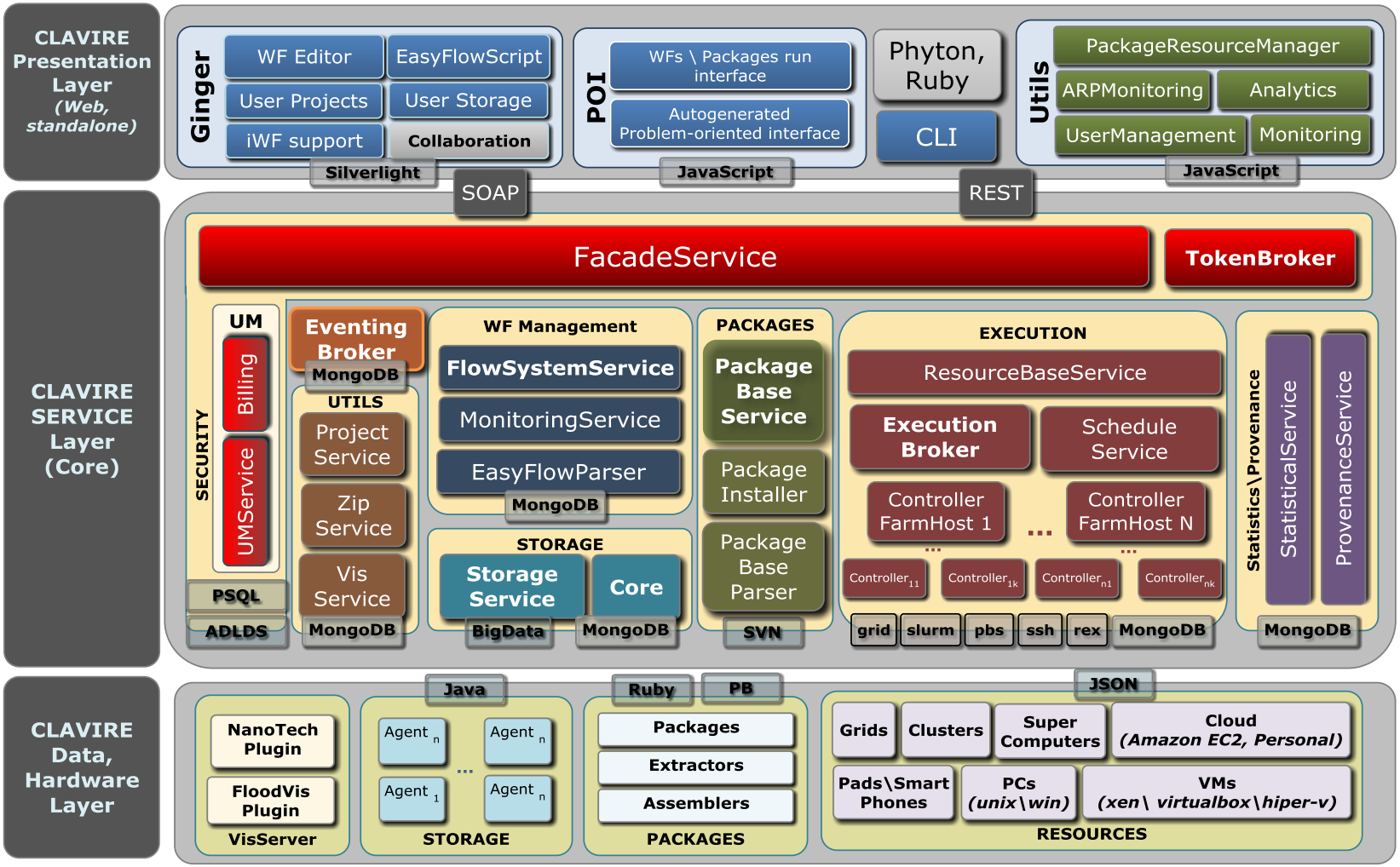
Main CLAVIRE advantages are:

1. Genericity of technology which is attained with iPSE (Intelligent Problem Solving Environment) provides the possibility to develop and maintenance virtual environments for different purposes and areas;
2. Unified approach allows easily to integrate heterogeneous data sources and computation resources (supercomputers, grids, clouds);
3. Support of interactive application management that allows development of distributed real-time systems, systems of interactive visualization and virtual reality;
4. Big Data technology native support, based on own distributed storage.

CLAVIRE application area

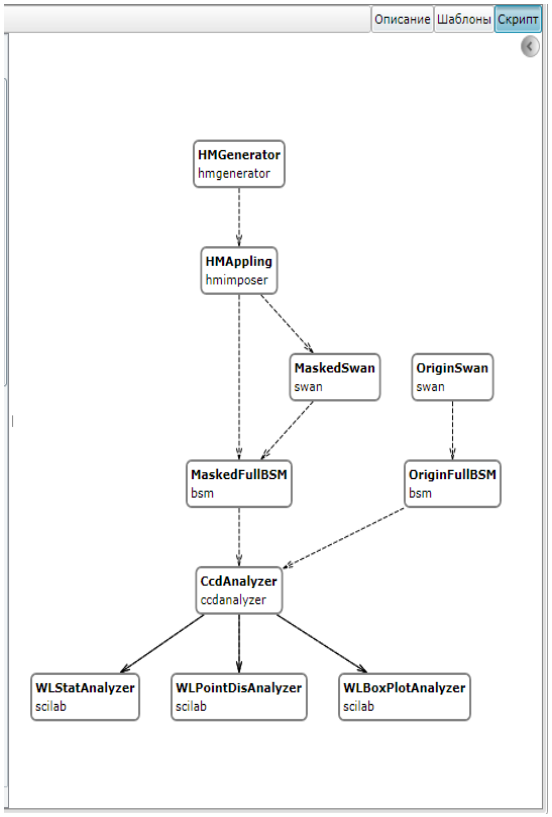
- **Collaborative domain-specific portal** – that can be used for general scientific needs as well as in educational purposes by social communities
- **Competence center –** cloud supported by a group of high-qualified experts of specific scientific area
- **Enterprise private cloud** – used for business needs with second generation cloud abilities
- **Public cloud (domain-neutral)**, market-like cloud with concept of integrated personal cloud
- **Urgent computing infrastructure**: computation support of a situation center

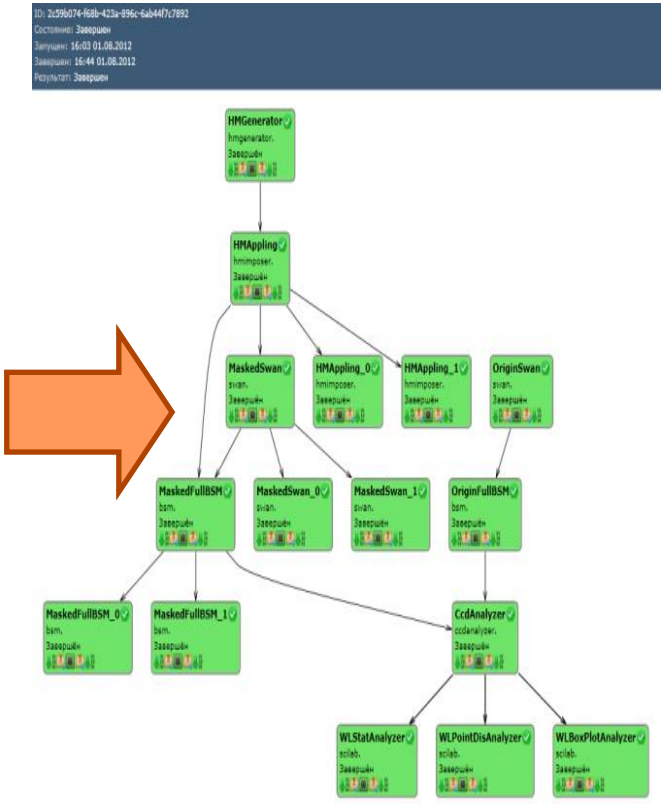
Common 3-tiered CLAVIRE architecture



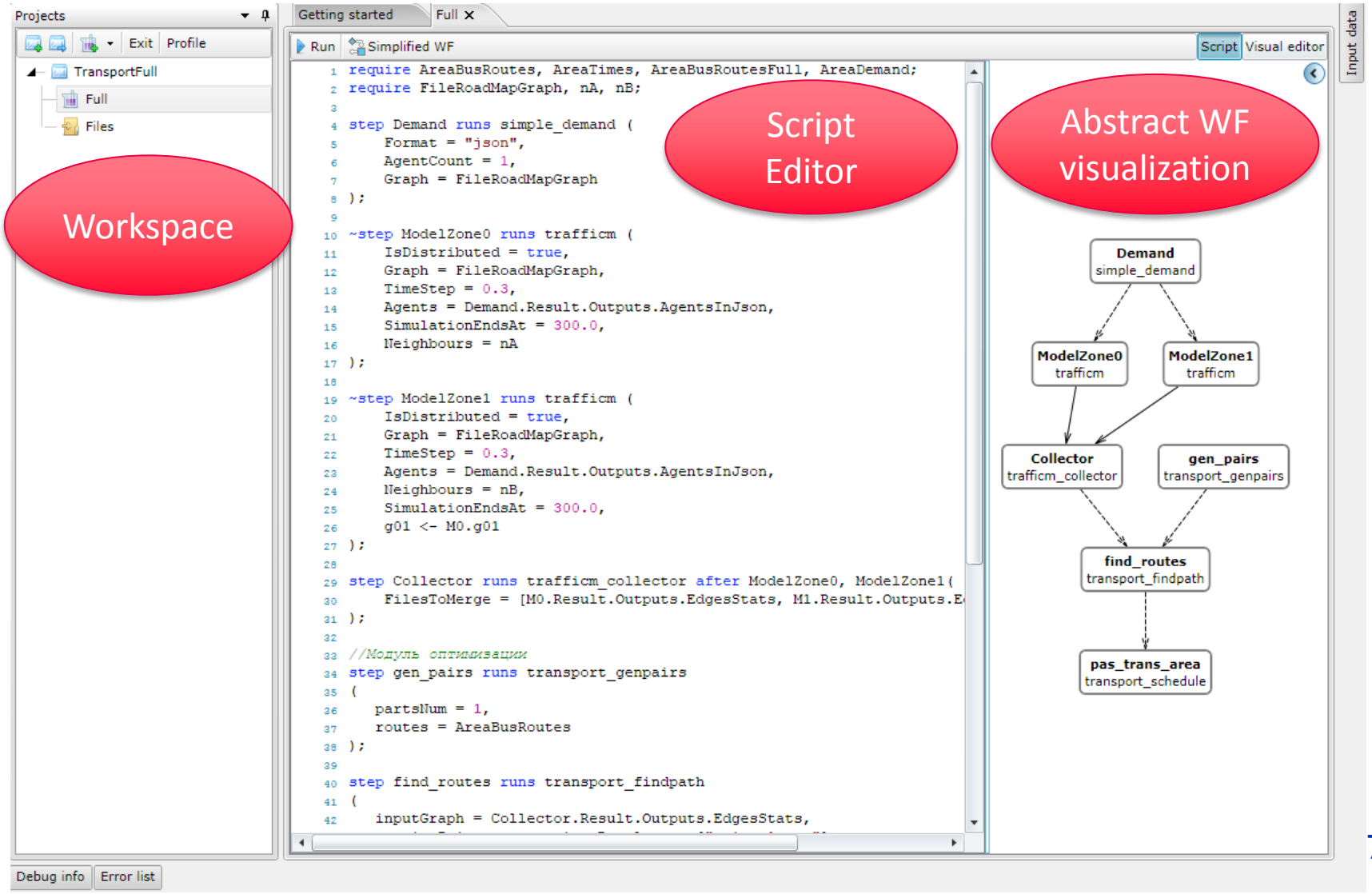
Composite Applications in CLAVIRE:

- Workflow is **DAG**
- Dependency types are:
  - **data** (parameters, files),
  - **control** (order),
  - **communication** (network communication, interaction)
- **EasyFlow** (DSL) – unified workflow description language





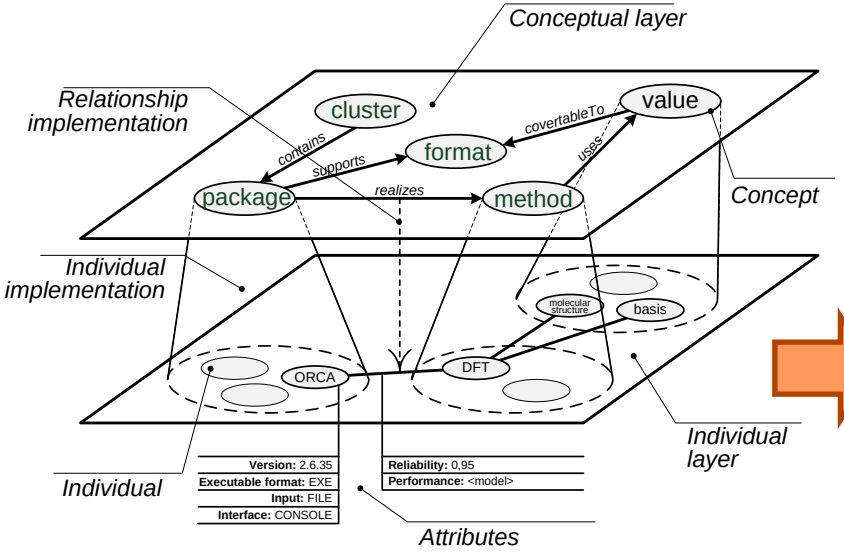
Ginger user interface. Design of WF



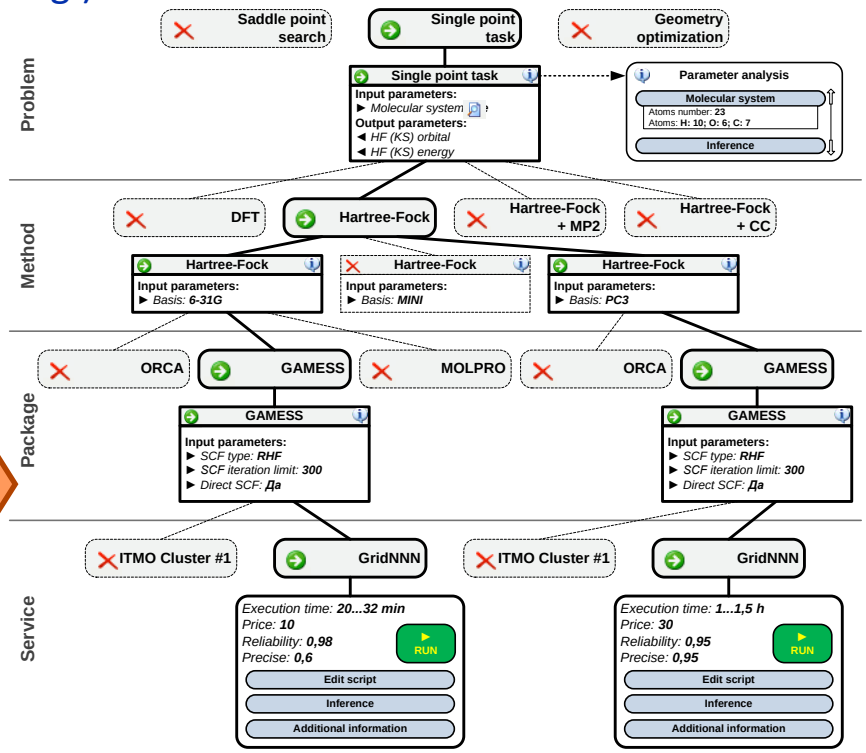
Advanced Design of WFs (1/2): procedural approach

Involving knowledge-based technologies

Using of ontology formalism for description of computational processes in domain areas



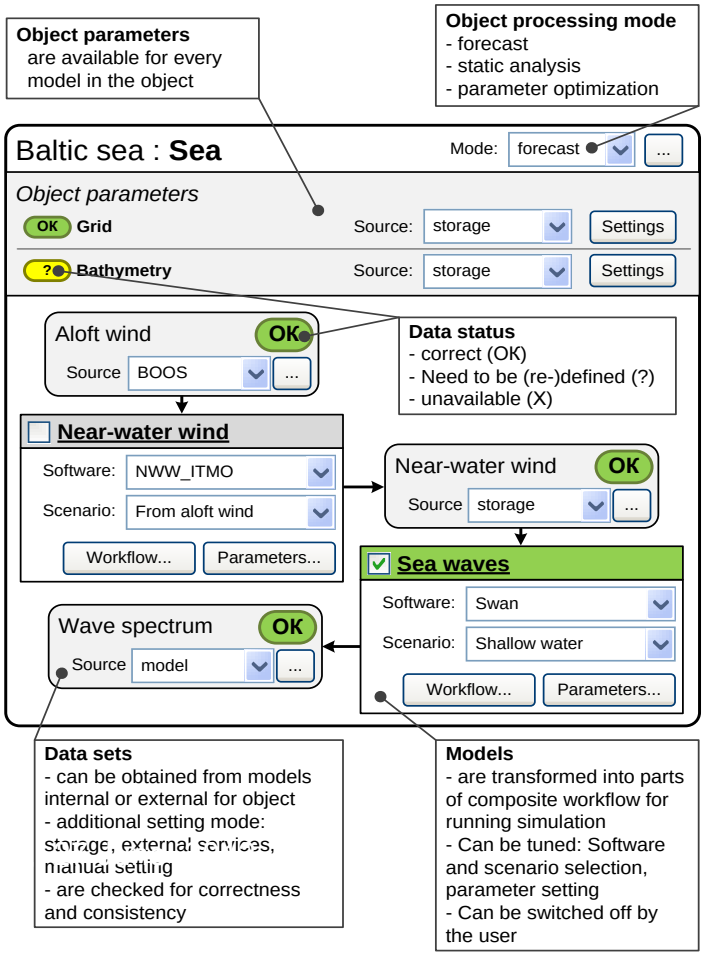
Rule-based WF's design (in quantum chemistry e.g.)



Advanced Design of WFs (2/2): system (deductive) approach. Involving “virtual objects” concept

Virtual simulation object – composition of the models for evaluation parameters of some real-life object

- Available for description of complex system existing in some environment
- Interpretable (both from domain and simulation infrastructure point of view)
- Supports interactive simulation process

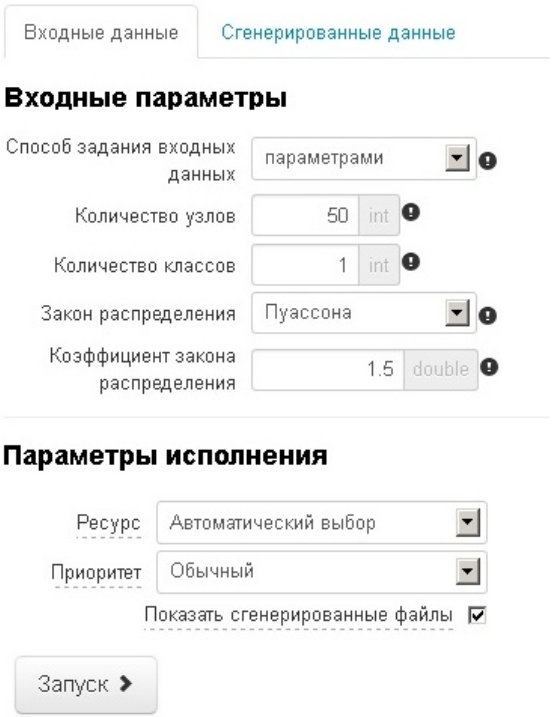


Problem oriented interface (POI)

- Unified Package description allows to generate user friendly UI
- One-click workflow execution

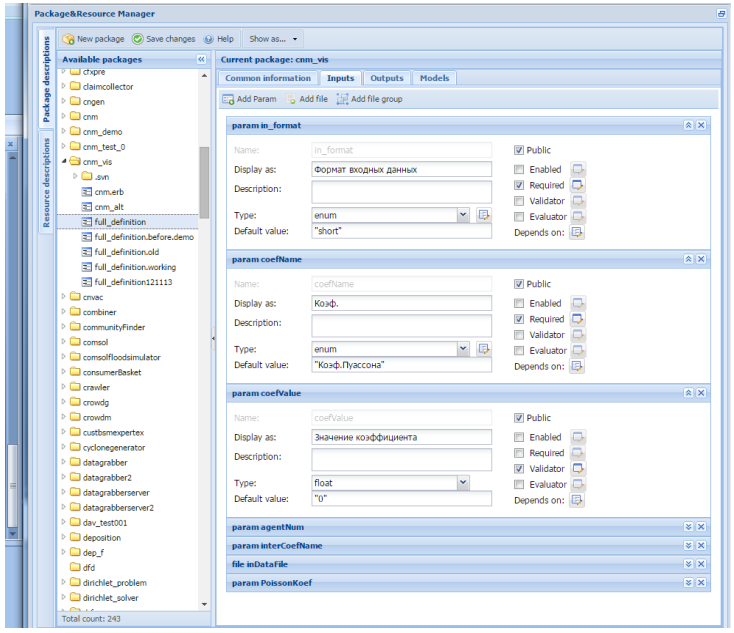
```
46  name "CNM"
47  #version nil
48  #display_as "Complex network modelling"
49  #vendor "Itmo"
50  #url "http://escience.ifmo.ru/"
51  #license "GPLv3"
52  #description "CNM help to understand and analise how
    i.e. rumors are being spread."
53  #logo ""
54
55  inputs {
56      #for model time measurement
57      public param {
58          name "PoissonKoef"
59          depends ["inDataFile?"]
60          type float
61          display "Коэффициент
    распределения Пуассона"
62          #validator {|val, ctx| val > 0.0}
63      }
64
65      #for model time measurement
66      public param {
67          name "agentNum"
68          display "Количество агентов"
69          depends ["inDataFile?"]
70          type int
```

Problem-oriented interfaces can be **automatically generated** with help of provided formal package description.



Package (software) embedding :

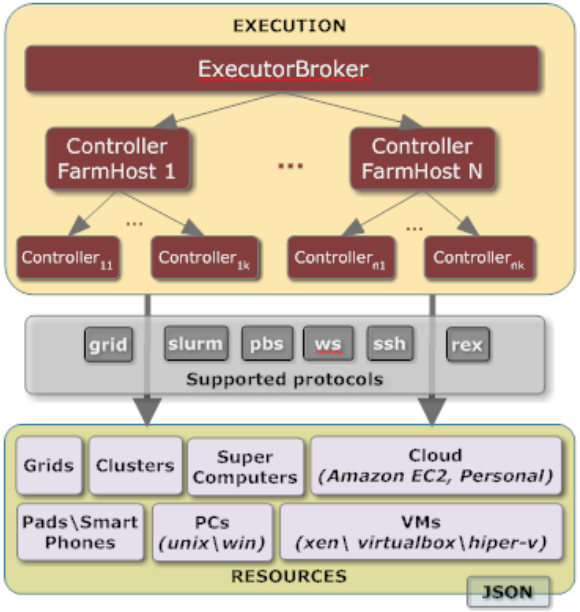
- **Retrieve necessary information** from package
- **Add package description on EasyPackage DSL in PackageBase** service, including :
  1) Common package information, input and output parameters.
  2) Configuration files templates if it's needed
- **Add package installation form to autodeploy service** if it's possible




```
name "CNM_VIS"
display_as "Complex network modelling"
vendor "Itmo"
url "http://escience.ifmo.ru/"
license "GPLv3"
description "CNM help to understand and analise how i.e. rumors are being spread."
inputs {

    public param {
        name "in_format"
        display "Формат входных данных"
        type enum ["short","full","alt"]
        default "short"
        required
    }

    public param {
        name "coefName"
        display "Коэф."
        type enum ["Коэф.Пуассона","Степ.показатель"]
        default "Коэф.Пуассона"
        required
    }

    public param {
        name "coefValue"
        display "Значение коэффициента"
        type float
        default "0"
        validator {
            |val,ctx| val <= 20
        }
    }
```

Computational resource embedding.

- **Add description** with PRManager or direct with ResourceBaseService that provides all functionality to manage resources
- **Install client** if needed (in example, REX-Windows client)
- **Physical package installation on resource,** if package is not uploaded to PackageInstallerService for automatically deployment

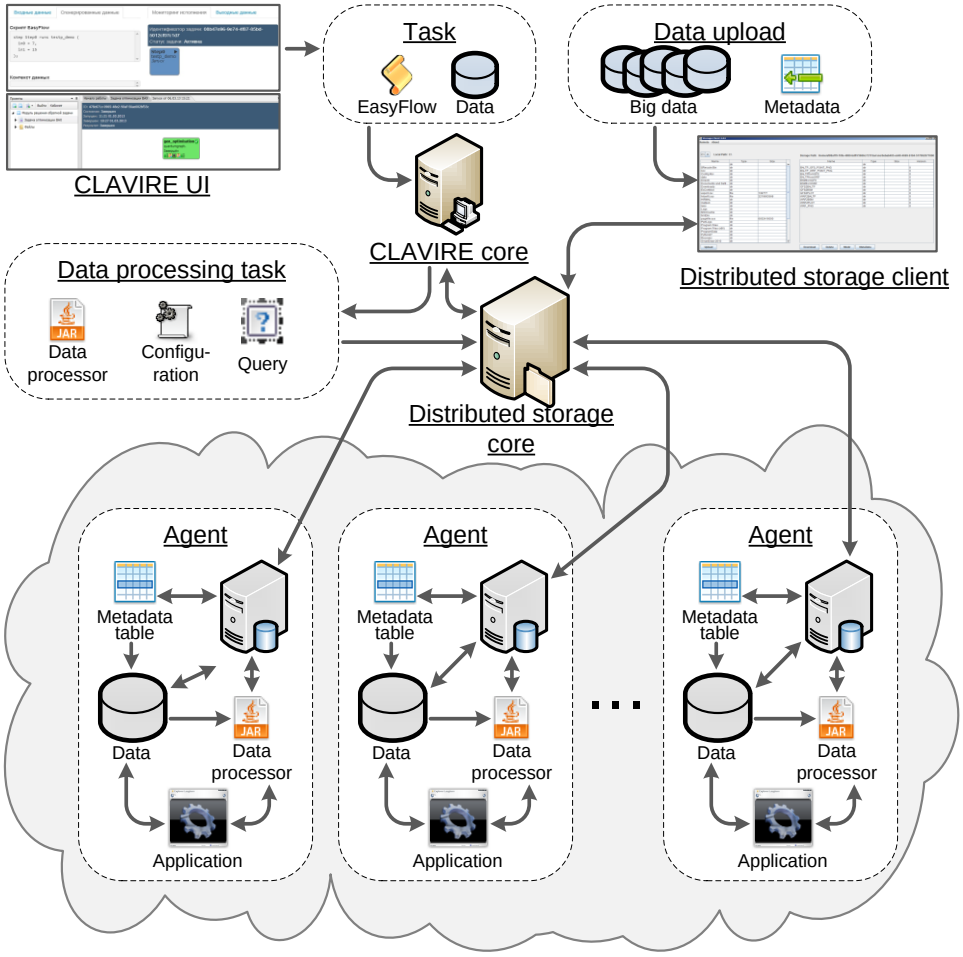


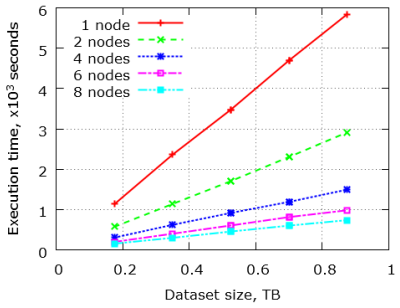

```
<resourceType> = {
        ”common resource parameters, including controller type”
    "NodeDefaults": {
        “common node default parameters, including core count, hardware parameters”
     "Packages": [{
         ”common package parameters, including functions “copyOnStart” and ”cleanDirectory”
       }]
      "Nodes": [ {
         "NodeName": "<NodeName>", node name
         "NodeAddress": "< NodeAddress >", - node address
         "Services": {"ExecutionUrl": "<ClientWCF>"} – execution path for WS
        }]}
```

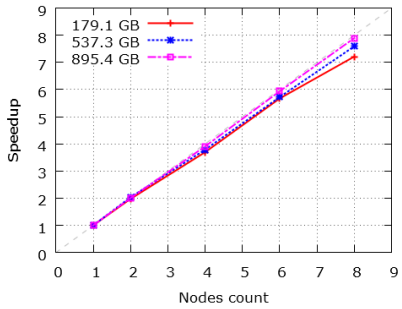
JSON format description

CLAVIRE Big Data Infrastructure

- Distributed data storage: core + agents
- Data replication and version control
- MapReduce model for distributed data (files) processing
- Java implementation of data processor
- Parallel efficiency up to 97%

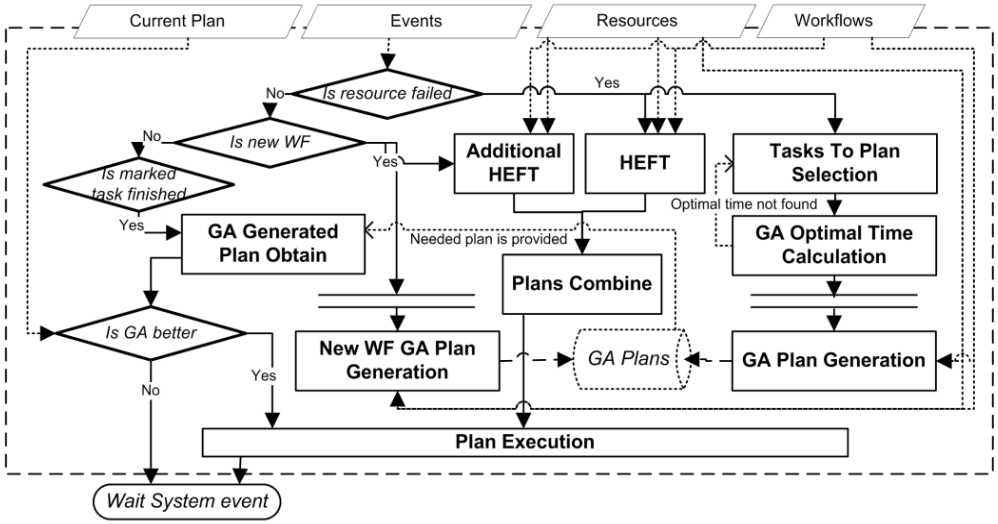


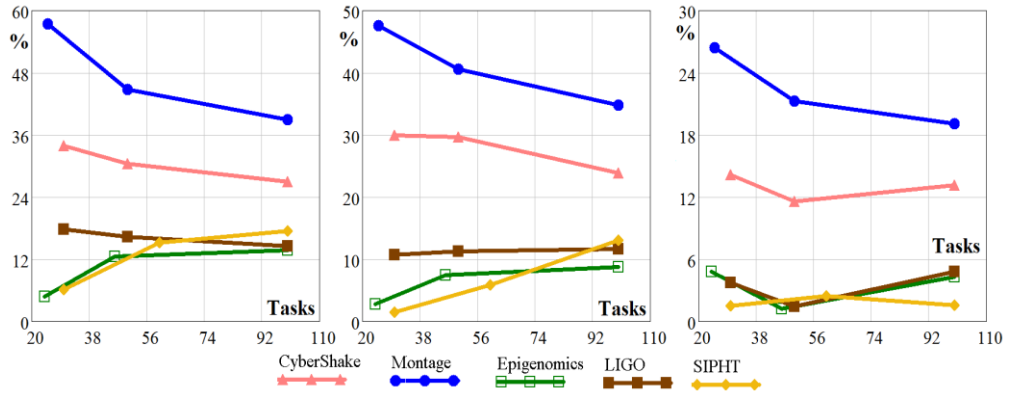




Optimization in CLAVIRE

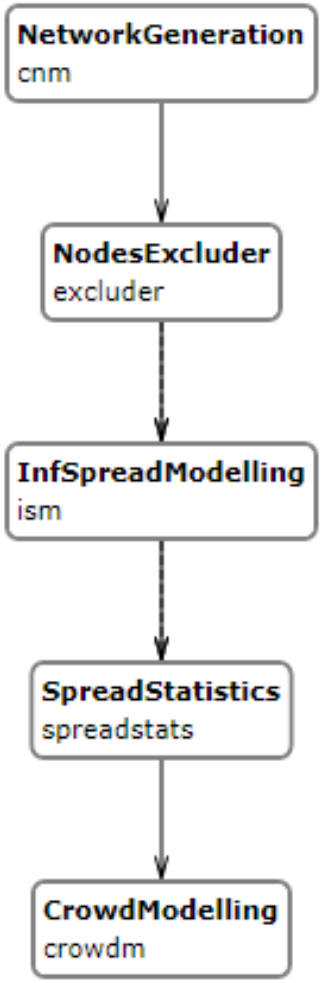
- **WF Scheduling optimization:**
  - **Hybrid algorithms** combines heuristic and meta-heuristic approaches to get best advantages from both sides;
  - **Window-based algorithms** take into account time windows to get maximized utilization with deadline constraints
- **Data placement optimization:**
  - **Static algorithm** optimizes data placement according to data usage during computation requests;
  - **Dynamic algorithm** optimizes data placement in time when environment and computation request are changing.



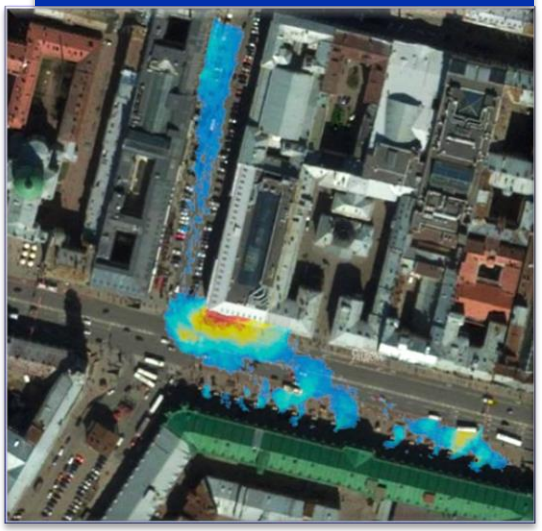


CA: flash-mob evacuation modeling

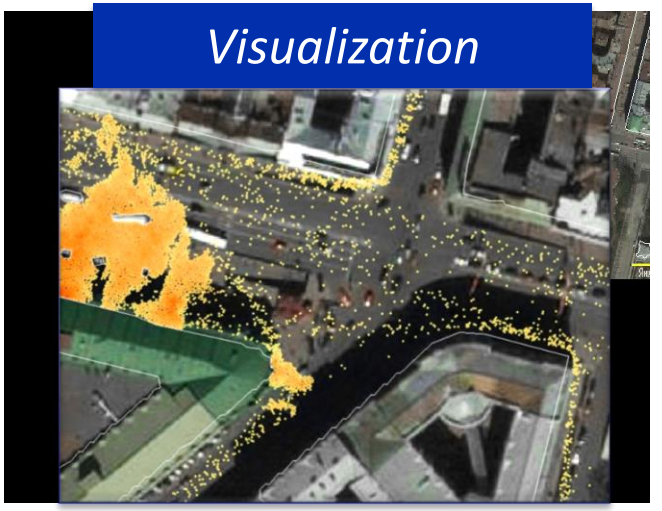
- **Generate** complex network;
- **Exclude** random nodes;
- **Modeling** of information spreading in the group of networks;
- **Aggregation** and statistics computation;
- **Evacuation** modeling and visualization;



Pressure map



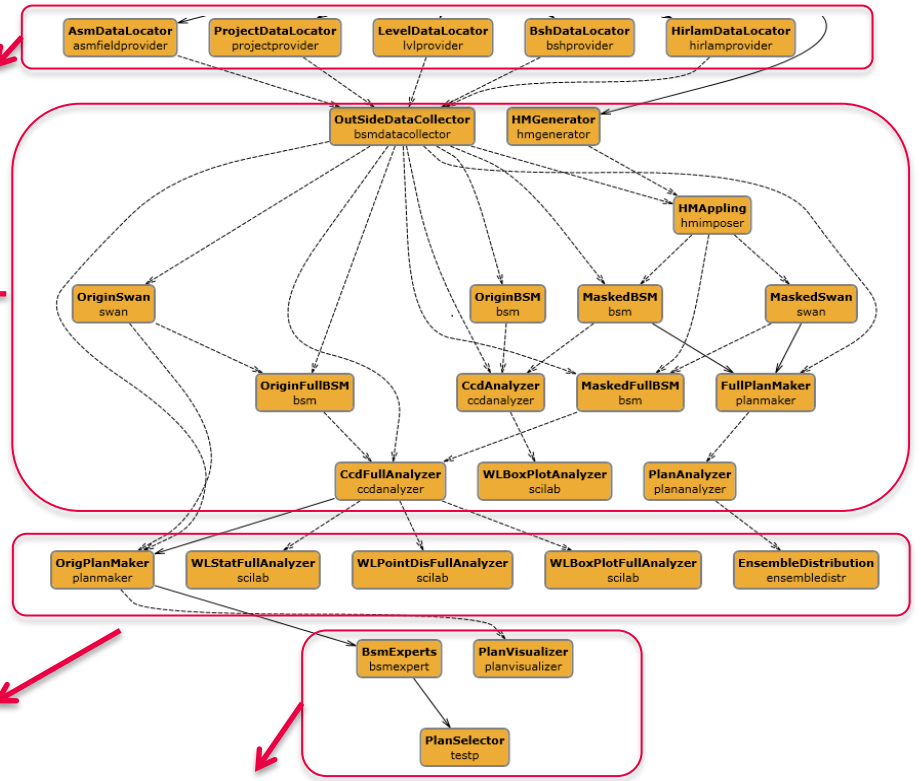
Visualization

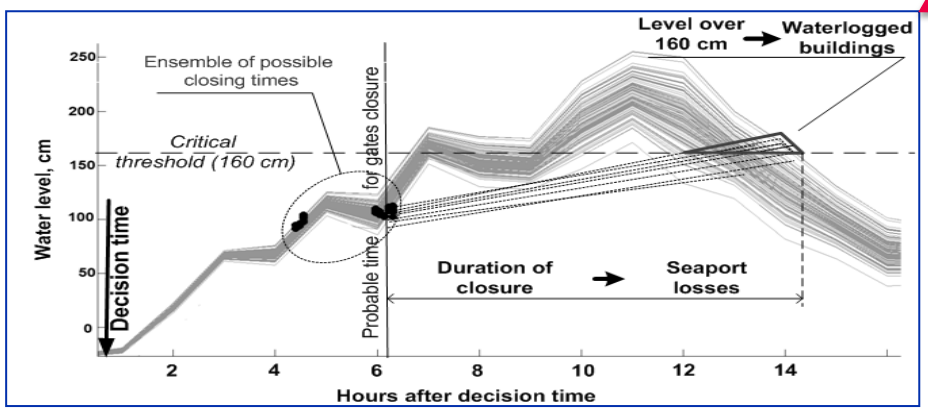


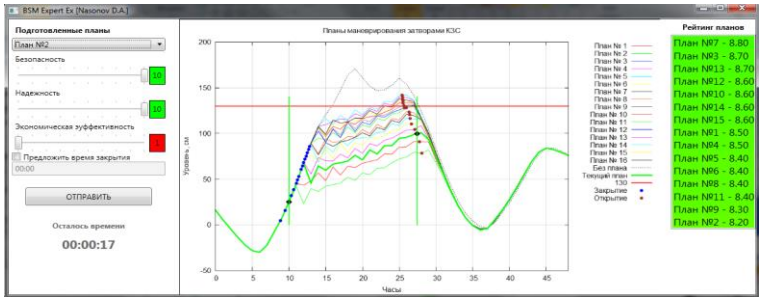
CA: Flood warning system

- **Data elements** collect data from sources
- Data **uncertainty mask** is generated and applied to wind velocity; ensemble **modeling** of wind-generated waves and water level in Baltic Sea area;
- **First Results** aggregation and visualization
- Decision support on close plan process is executed.
- **Final Result** selection

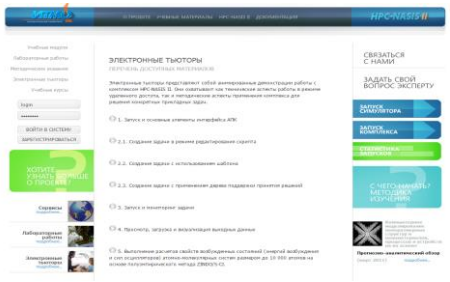
Deadline driven planning with task preemption.

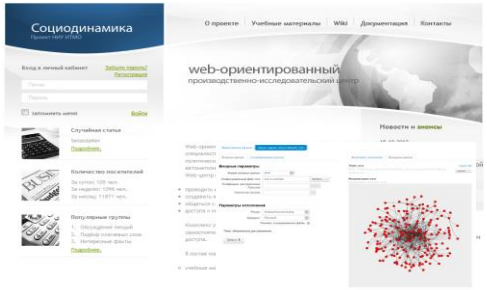






Education principles based on CLAVIRE





Computational resources and installed application packages

Commands and dataflows

Packages

5 Assessment's Packages

Educational Communities

Teachers and students

6

1, 2, 3 VSOs, SWFs, VLL's

4 Text Resourses

Text Materials

CLAVIRE

VSO, SWF, application packages descriptions

Interfaces

Component "Site"

VSOs, SWFs,...

Scientific Communities

Scientific Teams

Scientific Papers and Documents

Scientific Papers

Programming and Experimental Documentations

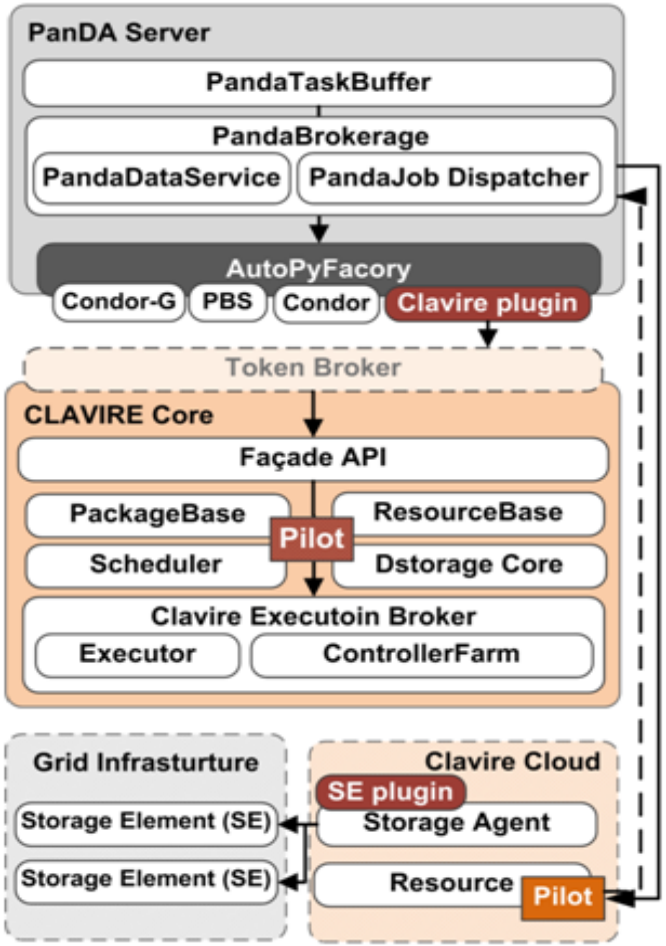
CLAVIRE with Panda integration concept

CLAVIRE can be useful for Panda, for the following reasons:

- CLAVIRE manages all CA as workflows with provided capabilities for urgent execution.
- CLAVIRE implements IWF technology .
- CLAVIRE encapsulates principles of composite applications, that allow Panda to extend its functionality on different types of services.
- CLAVIRE can operate with wide range of different resources.

Panda can be useful for CLAVIRE, for the following reasons:

- Panda can increase efficiency of used Grid resources
- Panda can optimize storage organization
- Panda provide pilot concept



Conclusion and future plans

CLAVIRE as a multipurpose platform has great potential for integration and cooperation with other computation environments, providing rich and clear interface functionality for fast composite applications development and efficiently its usage.

Besides different platform modules optimization, BigData technologies within hybrid idea of binding “code-to-data” and “data-to-code” concepts is one of most important current research directions that is investigating by our team.

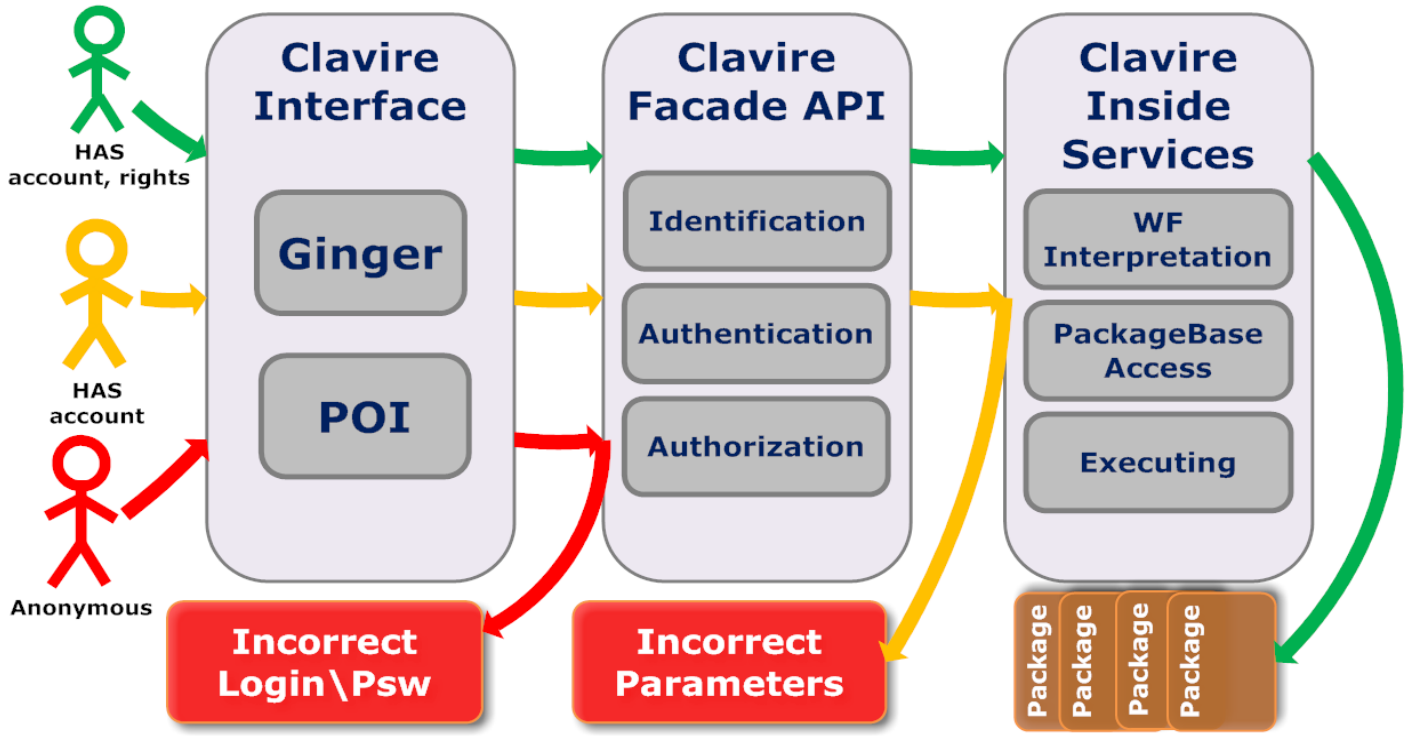
CLAVIRE VIDEO DEMOSTRATION

ITMO UNIVERSITY

Thank you for your attention!

Security model

- **Basic ACL** security model on data and service access;
- **EasyPackage limits access** to needed parameters only with prohibited direct access to resources and packages.



Workflow Management Platforms

		Taverna *University of Manchester, UK*	Kepler *Сообщество*	LONI Pipeline *University of California, USA*	WS-VLAM *University of Amsterdam, Netherlands*	Pegasus *University of Southern California, USA*	CLAVIRE *ITMO University, Russia*
WF management	*Visual*	+	+	+	+	+	+
	Script	+	–	–	–	–	+
	API	+*	–	–	–	+	+
Supported resources	*Local PC*	+	+	+	–	+	+
	Remote PC	+	+	+	–	+	+
	Web-services	+	+	+	–	–	+
	Grid	+	+	+	+	+	+
	Cloud	–	+	–	–	–	+
Windows support		+	+	–	–	–	+
Abstract WF		–	–	–	+	+	+
Interactive WF		–	–	–	+	–	+
Exception handling		+	+	+	–	+	–
Tools for history analysis		+	–	+	–	+	+