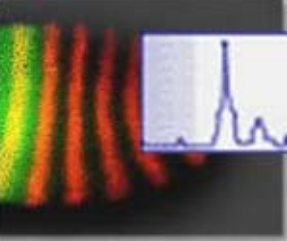


# An Information Management System for Collaboration within Distributed Working Environment

*<http://urchin.spbcas.ru/downloads/esimbios/>*

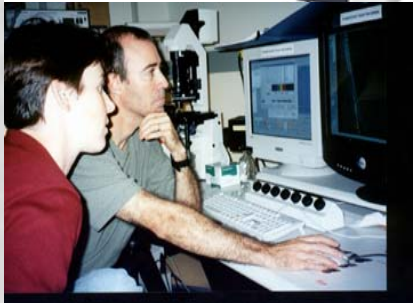
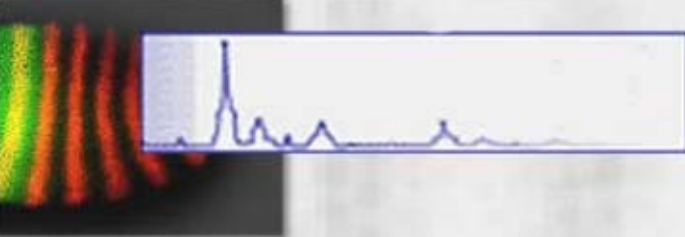
***Maria Samsonova, Andrei Pisarev,  
Konstantin Kozlov, Ekaterina  
Poustelnikova, Arthur Tkachenko***



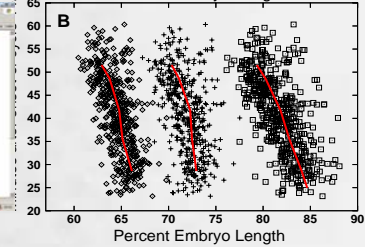
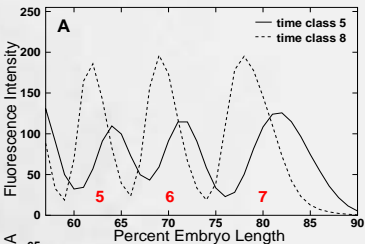
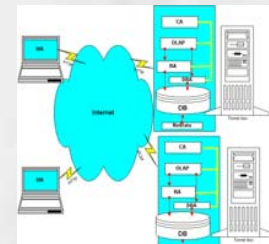
# Outline

- Motivation
- SOAP vs. REST
- iSIMBioS architecture and functionality
- Oracle BPEL and iSIMBioS comparison
- Conclusions

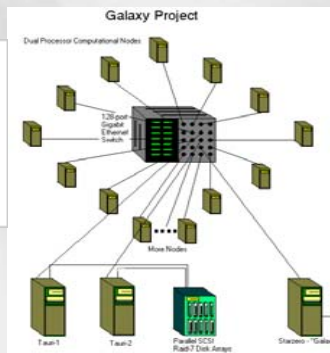
# Focus of research: the mechanisms of cell fate determination



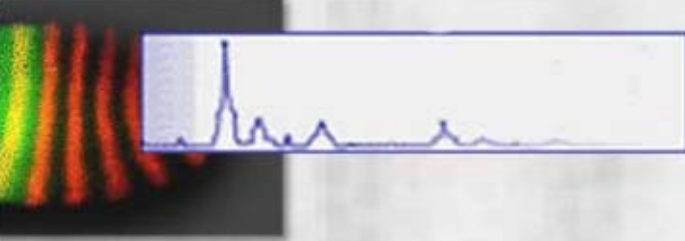
**SUNY & UCSD,  
USA**



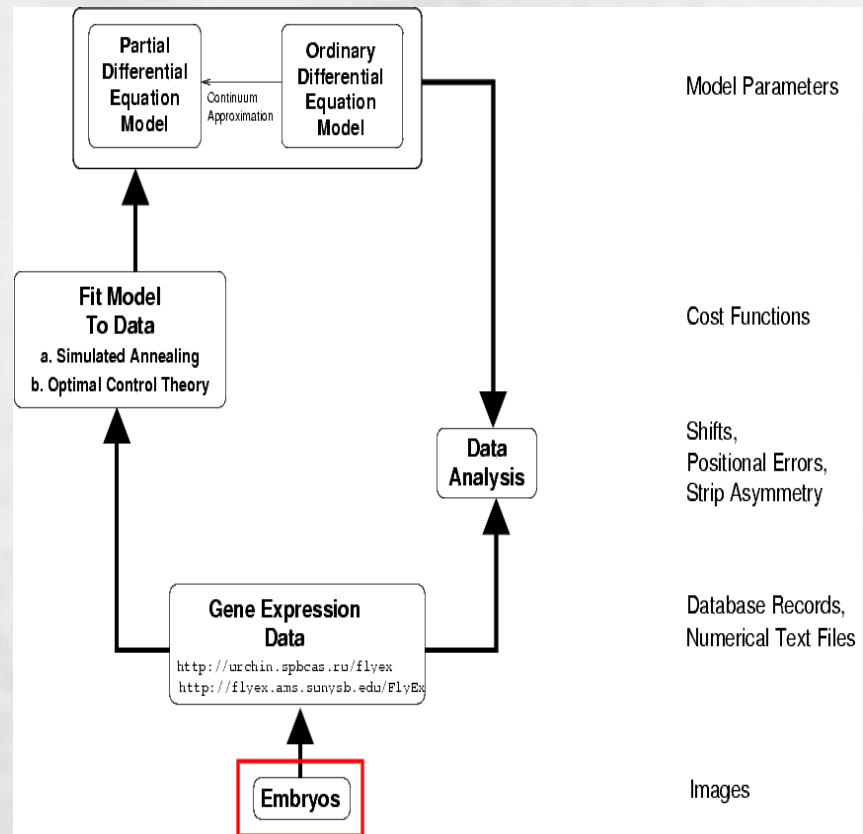
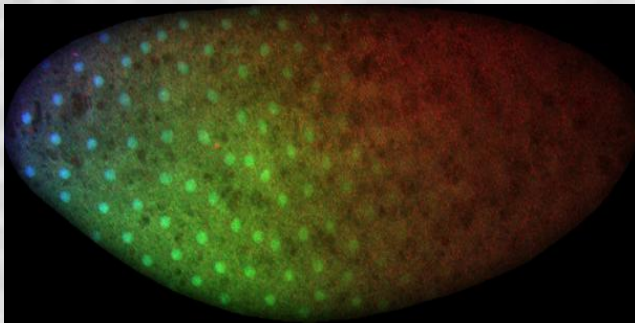
**SPbPU & FTI,  
Russia**



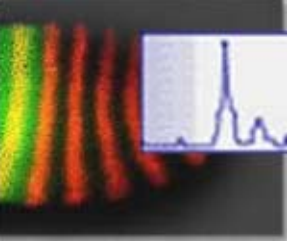
**LANL, USA & UvA, Holland**



# Fundamental problem of developmental biology: Determination of cell fate

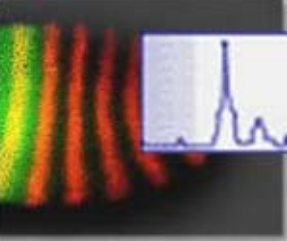






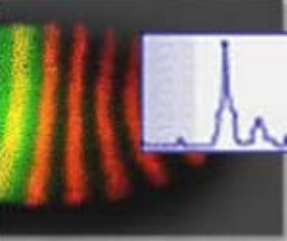
# System requirements

- flexibility in specification and modification of analysis methods;
- provision of fast and simultaneous access of multiple users to shared data and methods;
- support of distributed processing and analysis of data;
- support of autonomous task performance upon connection hang up, as well as notification about processing results;
- use of heterogeneous software/hardware platforms; provision of access through Firewall and n Proxy servers;
- scalability
- provision of continuous work, when new components are added or old one are removed;
- failure-resistance, if malfunction of hardware or software components happens;
- portability across software platforms.
- preferably based on open source software.



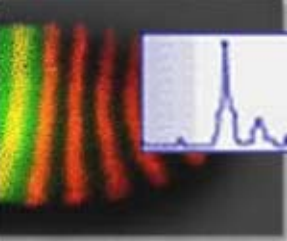
## SOAs currently used to integrate heterogeneous resources

- SOAP (Simple Object Access Protocol), developed by Microsoft, XML Protocol Working Group, W3C (World Wide Web Consortium).
- REST (Representational State Transfer), developed by Roy Fielding.



# SOAP

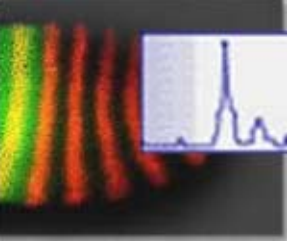
- SOAP is a protocol for exchanging XML-based messages over a computer network, normally using HTTP.
- Both SMTP and HTTP are valid application layer protocols for SOAP.
- HTTP has gained wider acceptance as it works well with today's Internet infrastructure; specifically, SOAP works well with network firewalls.



# REST

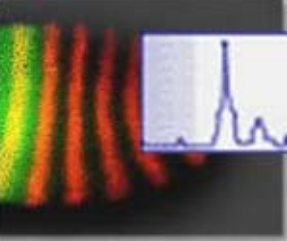
- REST is a software architectural style for distributed hypermedia systems like the world wide web.
- Design principles:
  - A stateless client/server protocol
  - A set of well-defined operations (HTTP - POST, GET, PUT and DELETE, CRUD (Create, Read, Update, and Delete))
  - A universal syntax for resource-identification
  - The use of hypermedia both for application information and application state-transitions



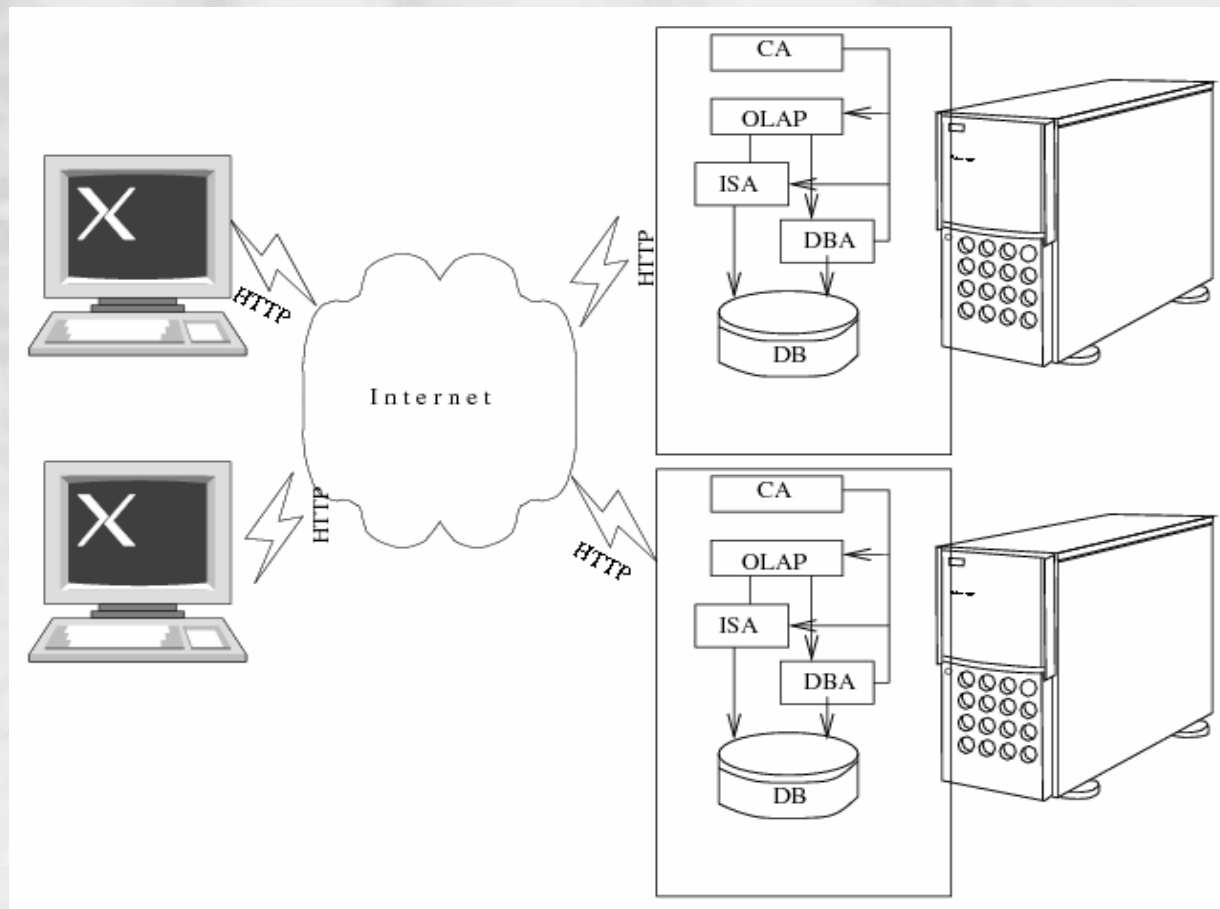


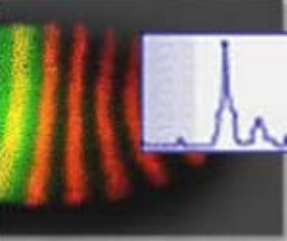
# REST vs. SOAP

- XML-representation of some data types ( images, BLOBS, matrices ) can decrease the performance of application.
- REST proponents argue that the HTTP's minimal method set and semantics, as well as its ability to extend this method set as required is sufficiently general to model any application domain.
- REST scales well with large numbers of clients, enables data transfer in streams of unlimited size and type and supports intermediaries (proxies and gateways) as data transformation and caching components.



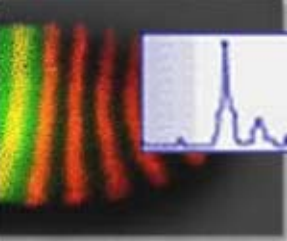
# iSIMBioS architecture





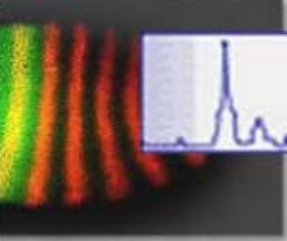
## Agent behaviour in iSIMBioS

- autonomy: agents operate without the direct intervention of humans or others, have some kind of control over their actions and internal state, can act upon connection hang up;
- social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language;
- reactivity: agents perceive the context in which they operate and react to it appropriately;
- pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.



## CA agent

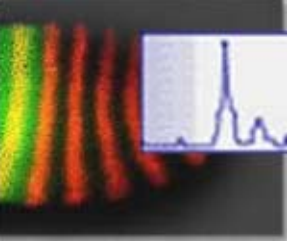
- ✓ stores in its database the list of counteragents and their URLs, list of functions, reference to monitoring program, load and authorization characteristics;
- ✓ supports agent registration;
- ✓ is notified about agent's current load, scheduled sign-off;
- ✓ notifies agents about the current system status in response to their request;
- ✓ monitors the functionality of the system;
- ✓ notifies registered agents about an agent failure and other changes in the configuration of the system;
- ✓ notifies the system administrator about changes by e-mail.



## OLAP agent

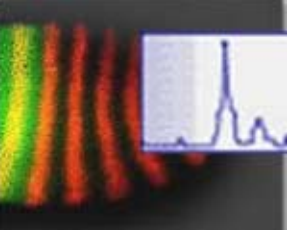
- ✓ cooperates with other agents to execute complex scenarios by implementing logical rules;
- ✓ communicates with local database via JDBC and remote database via DBA;
- ✓ interacts with registered workflow modules providing for their initialization, function calls and result output.





# Workflow and modules

- In general each scenario for image processing or data analysis consists of many steps executed by heterogeneous programs and services, which we call as modules.
- Modules communicate with each other via agents.
- An agent can insert data into a database, send it directly to the next module and modify configuration files and other auxiliary data, if necessary.
- Modules are program and services implemented as XML-PRC, SOAP, RMI, JDBC, CGI and command-like applications.



# Control of functionality

```

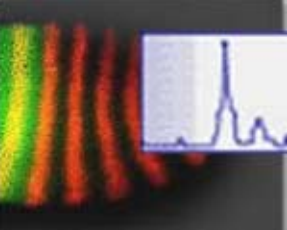
<xml><email>pisarev@spbcas.ru</
email><agentname>Venus</agentn
ame><theme>Flyex.ams functional
monitoring</theme><diagnosis>
  <rules>
    $flyex.ams=DBA.Counter|DBA|DB
    A.db2|ams.is|ams.Netdata|ams.ww
    w;
    ...
    $DB2=DBA.db2|ams.is|ams.Netdat
    a;
    $JDBC=DBA.db2|ams.is;
  </rules>
  <text>
    $flyex.ams=Flyex.ams.sunysb.edu;
    $DBA=Data Base Agent;
    $IS=Image Server;
    $DB2=IMB DB2;
    $JDBC=JDBC DB2;
  </text>
</diagnosis>

```

```

<test>
  <n>1</n>
  <name>dba.Counter</name>
  <url>
    http://flyex.ams.sunysb.edu:7005/counter?fl
    yex.jpg
  </url>
  <copy>
    MasterCopy/urchin/Counter/counter.jpg
  </copy>
  ...
  <explanation>
    http://flyex.ams.sunysb.edu:7005/counter?...
    is used for check of availability of port=7005
    DataBaseAgent service
  </explanation>
</test>
<test>

```



# User interface

The screenshot displays the Bamбу! software interface. The main window title is "Bambu! - /home/kozlov/calcs/bambu/parallel\_nonsense.ap". The menu bar includes "Run", "Stop", "Rew", "Save", "Save as", "Compile", and "Close".

The central workspace contains a flowchart with the following components and connections:

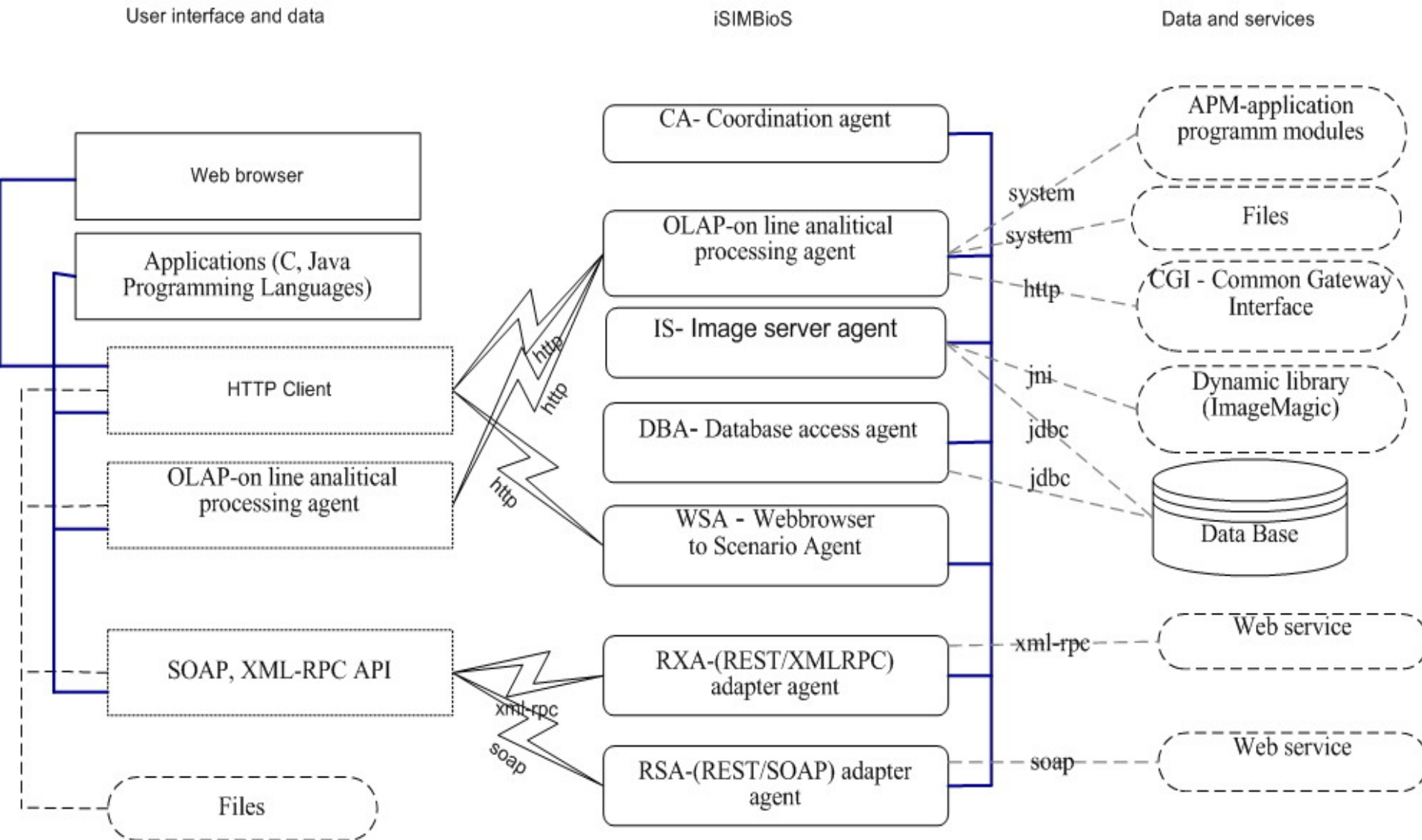
- Three "file" blocks on the left.
- A "strel" block at the bottom left.
- Three "median" blocks in the middle column.
- A "max" block in the upper middle.
- A "threshold" block in the upper right.
- Two "turn" blocks in the lower right.
- A "rotate" block in the middle right.

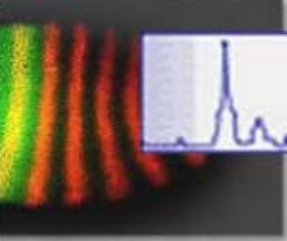
The flow starts from the "file" and "strel" blocks, goes through "median" and "max" blocks, then through "threshold" and "rotate" blocks, and finally ends at the "turn" blocks.

On the right side, there are two tool panels:

- The top panel, titled "Tools", shows a list of tools for the file "venus2.spbcas.ru": align, apee, apro, apsc, ar\_minus, ar\_plus, ar\_x, avg, blob, bolb, and chemar.
- The bottom panel, titled "Bambu! Dis", shows tools: Mozilla, display, and cplot.

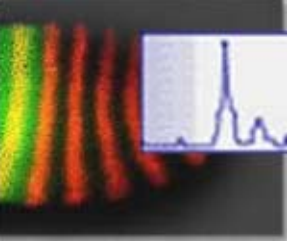
# Application program interfaces





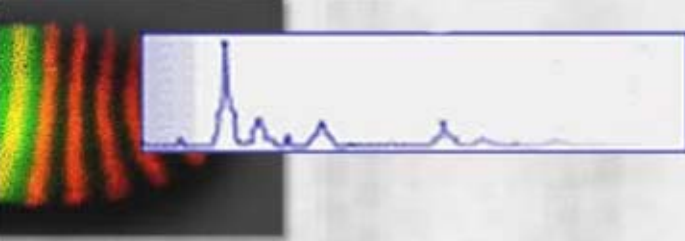
## **Oracle BPEL and iSIMBioS comparison**





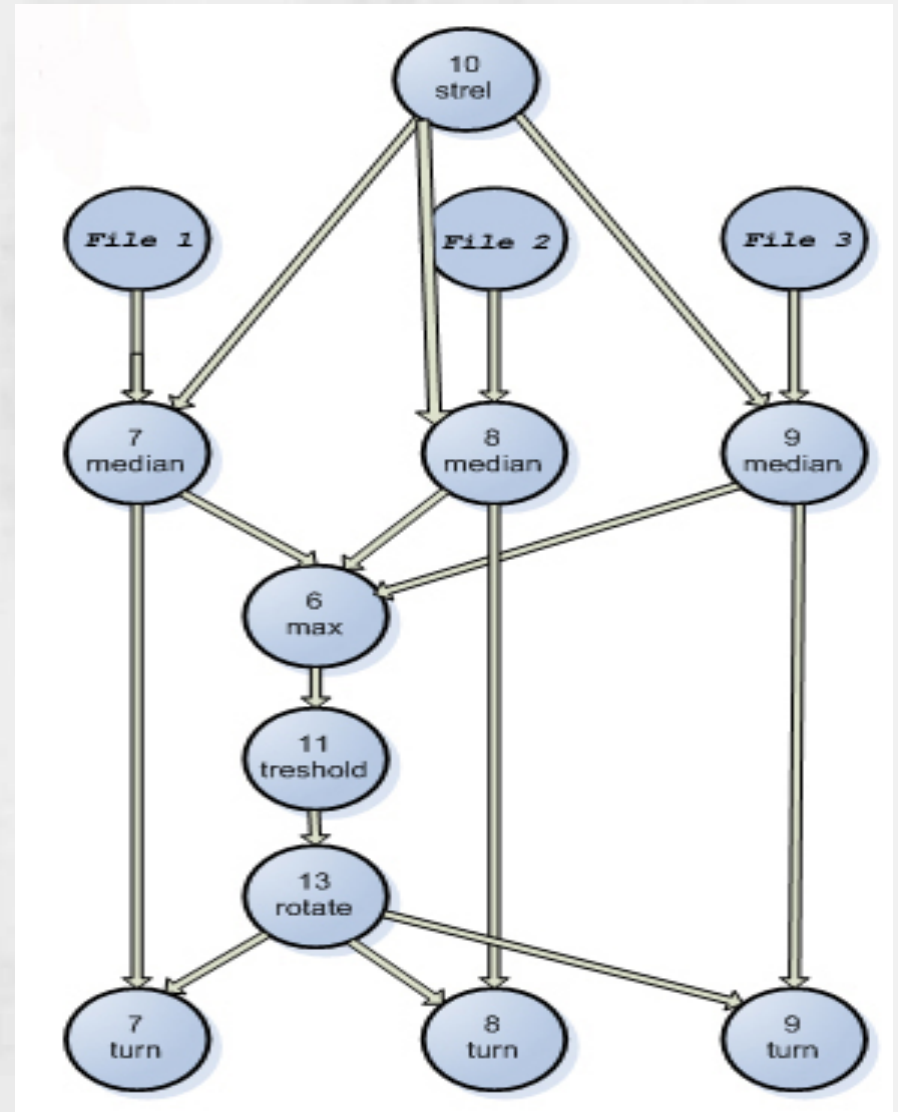
## iSIMBioS and Oracle BPEL

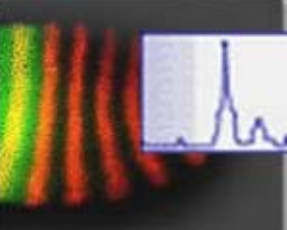
- Oracle BPEL:
  - Send data by value
  - Send data only TO and FROM Process Manager (PM) Server
  
- iSIMBioS
  - Send data by reference
  - Send data between performers (i.e. servers)



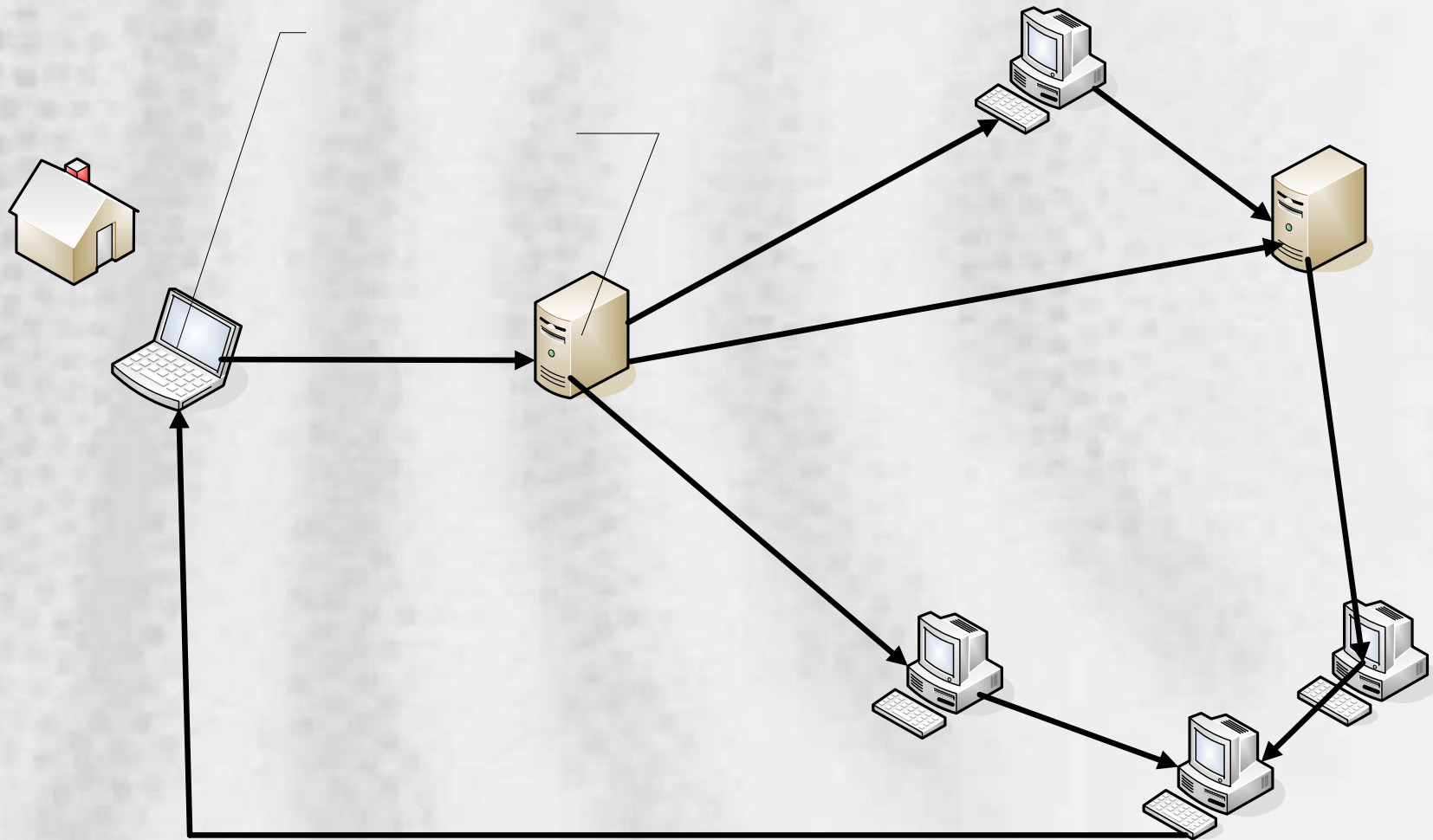
# Test workflow

- Scenario graph is wide enough
- Has good parallelizability
- Scenario height = 5, width = 3

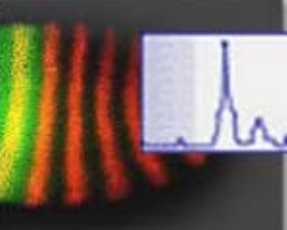




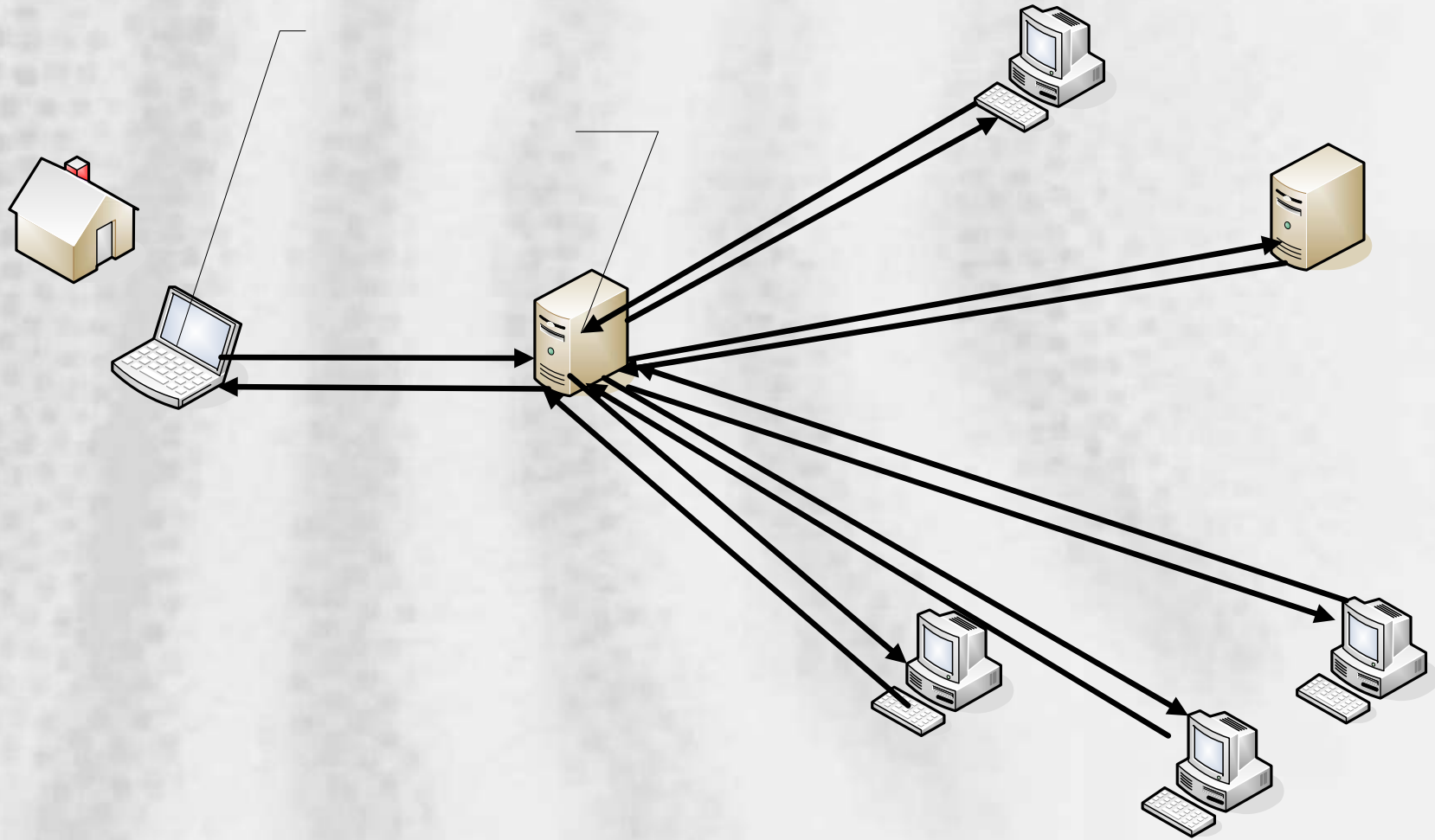
# iSIMBioS data flow



U

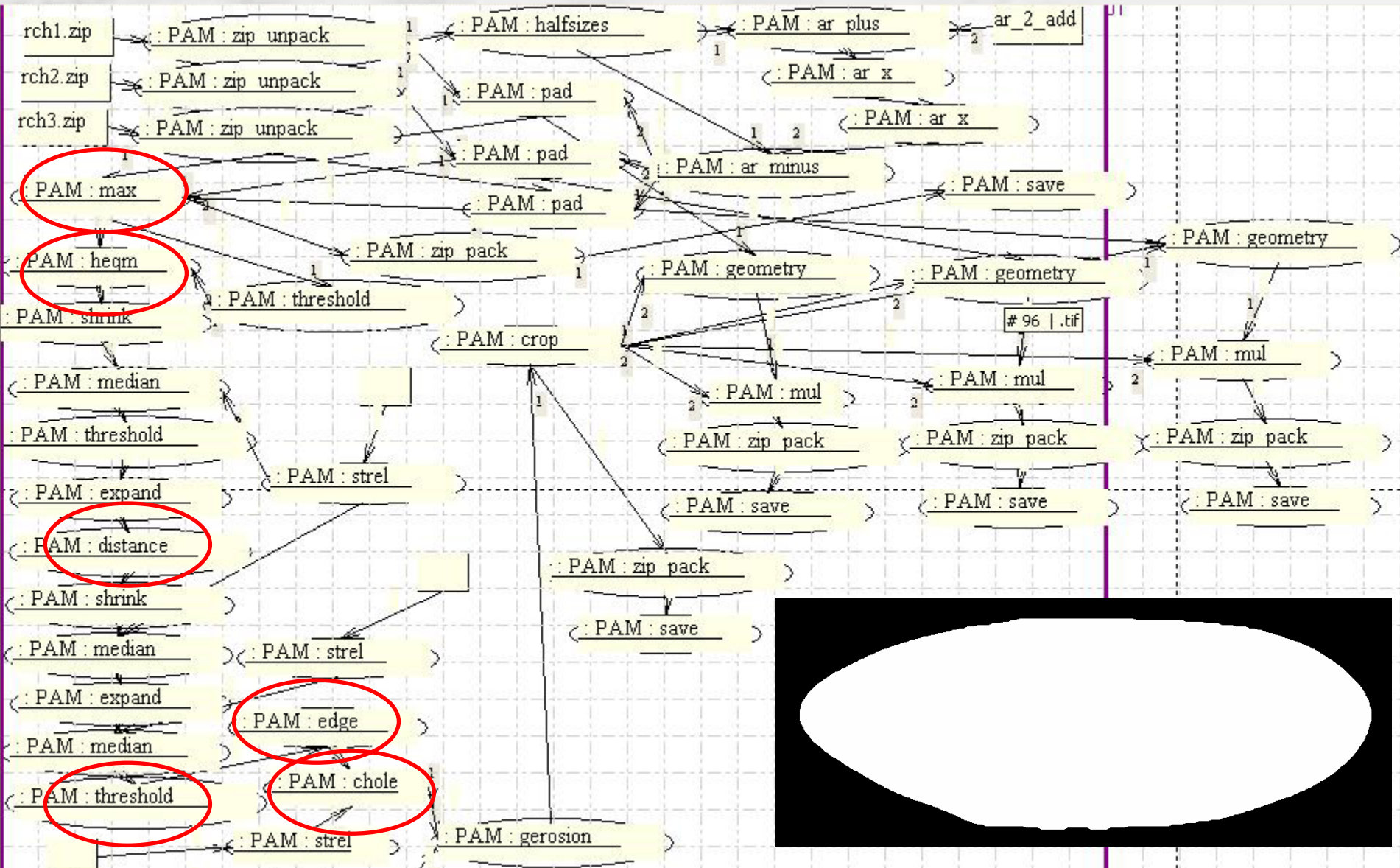


# Oracle BPEL data flow



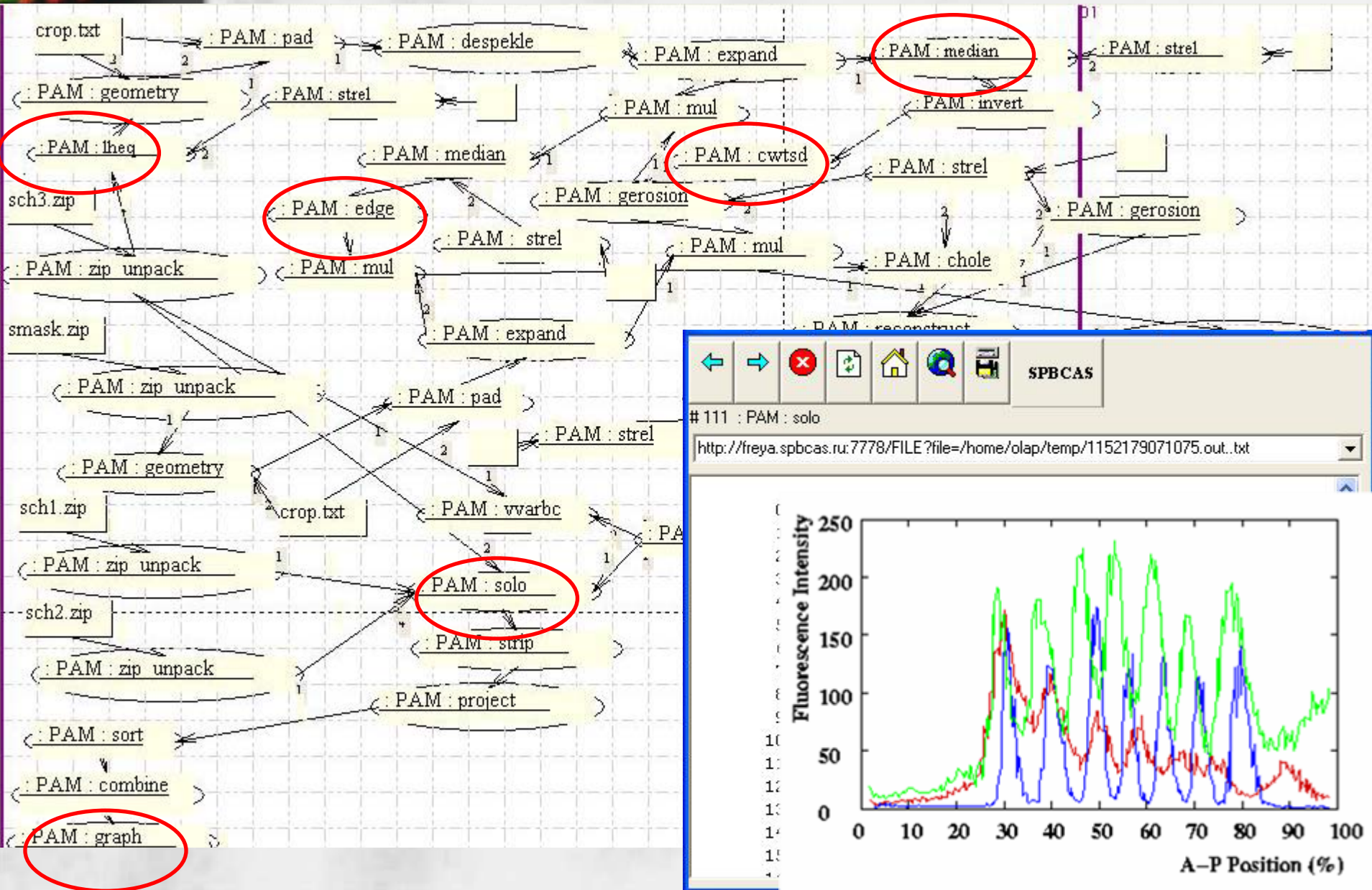


# Smooth Mask workflow





# Segmentation workflow

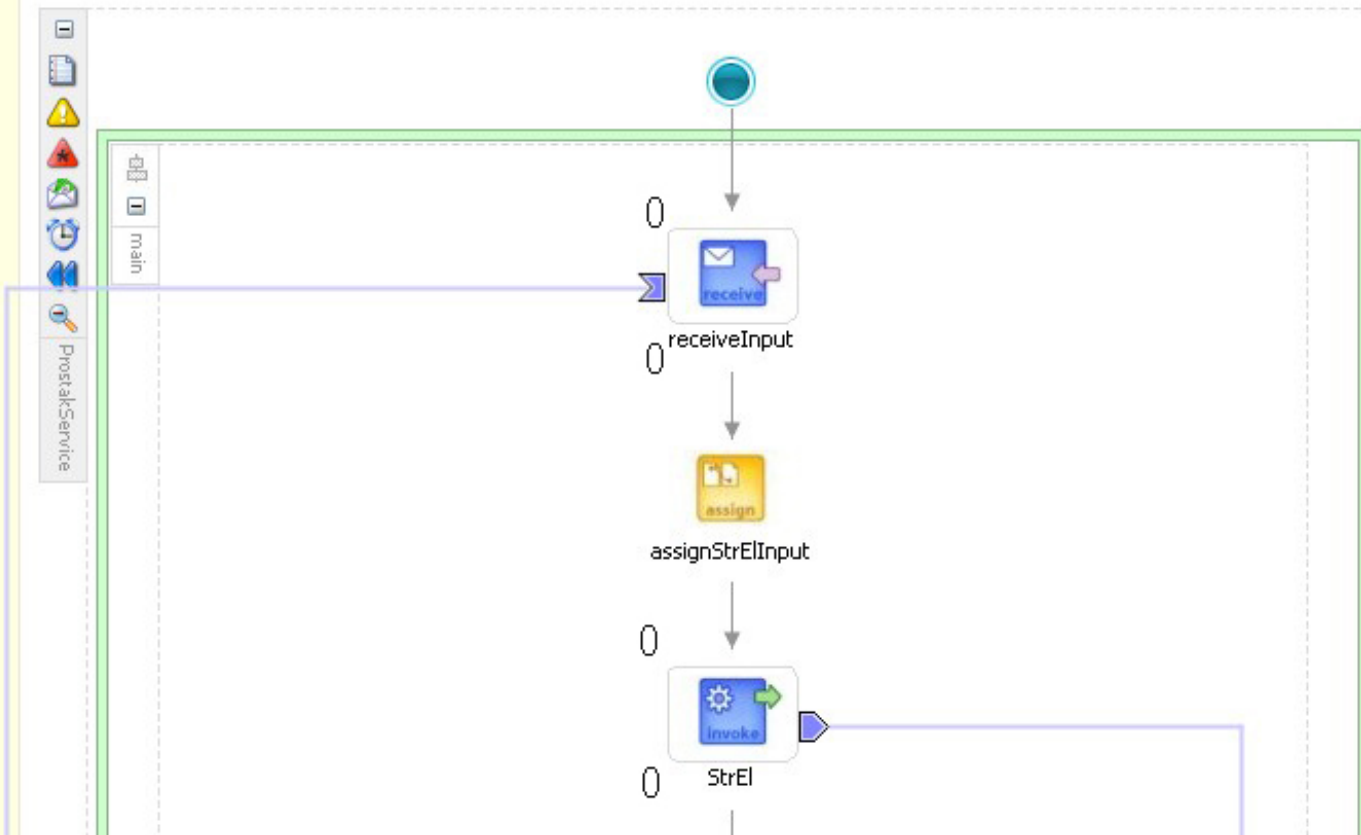


# Oracle BPEL Designer workspace(1)

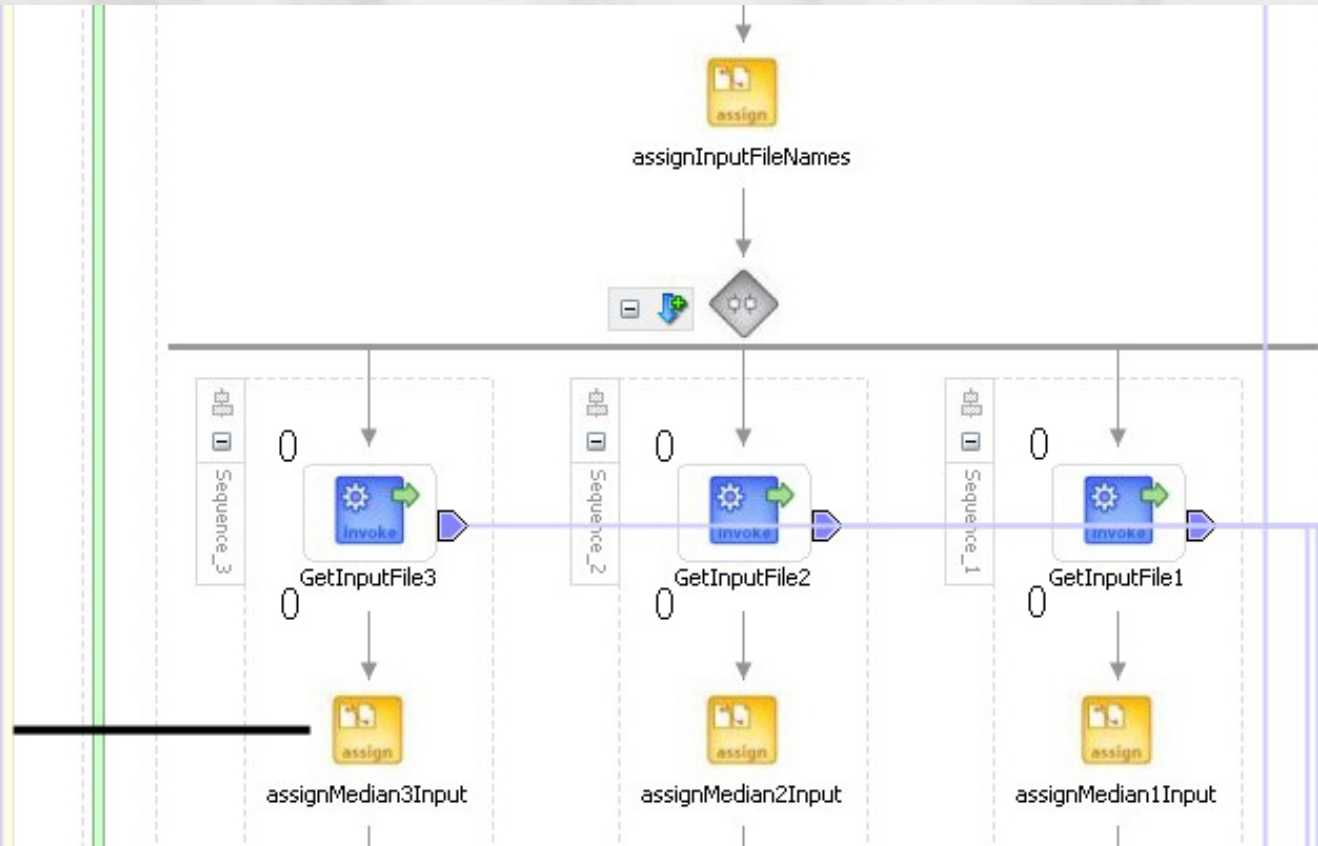
Partner Links

Partner Links

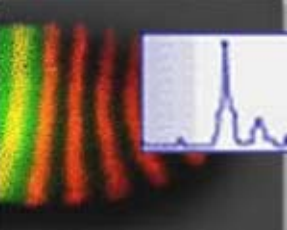
U/D



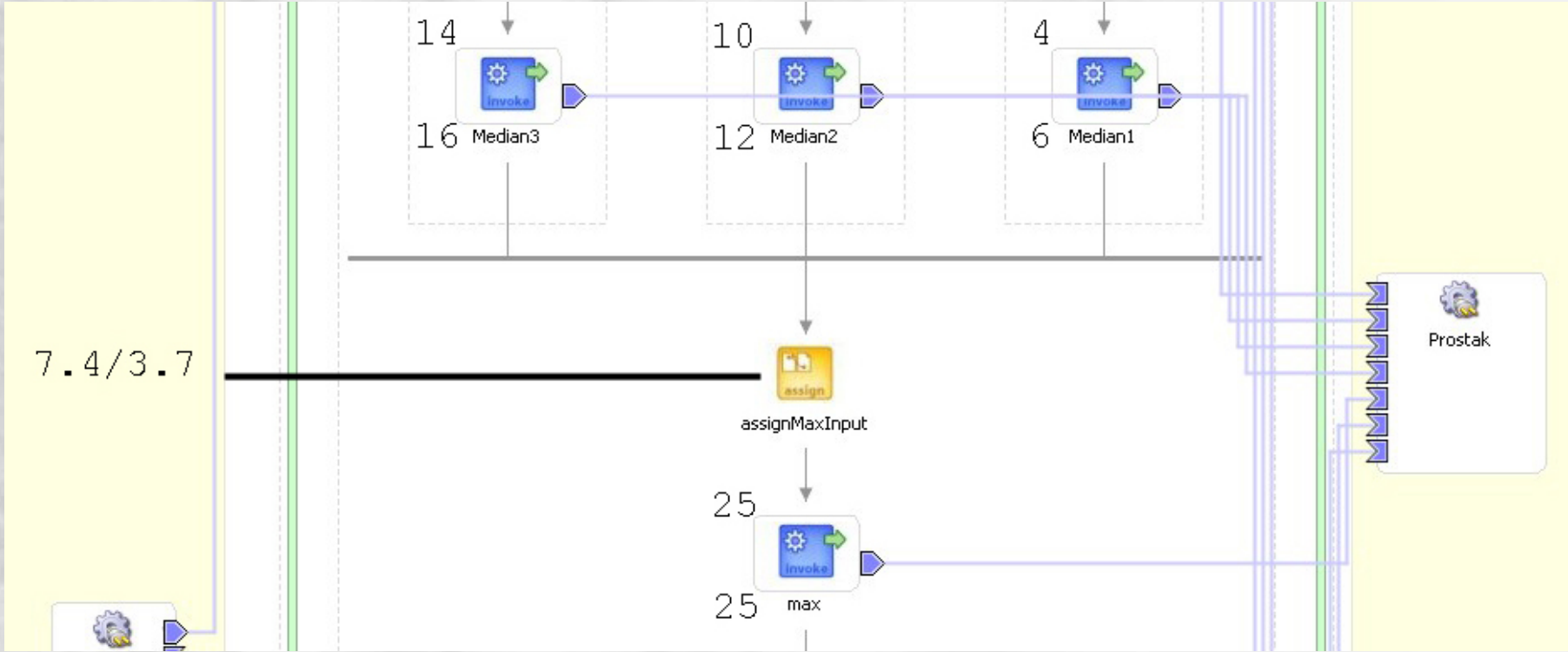
# Oracle BPEL Designer workspace(2)



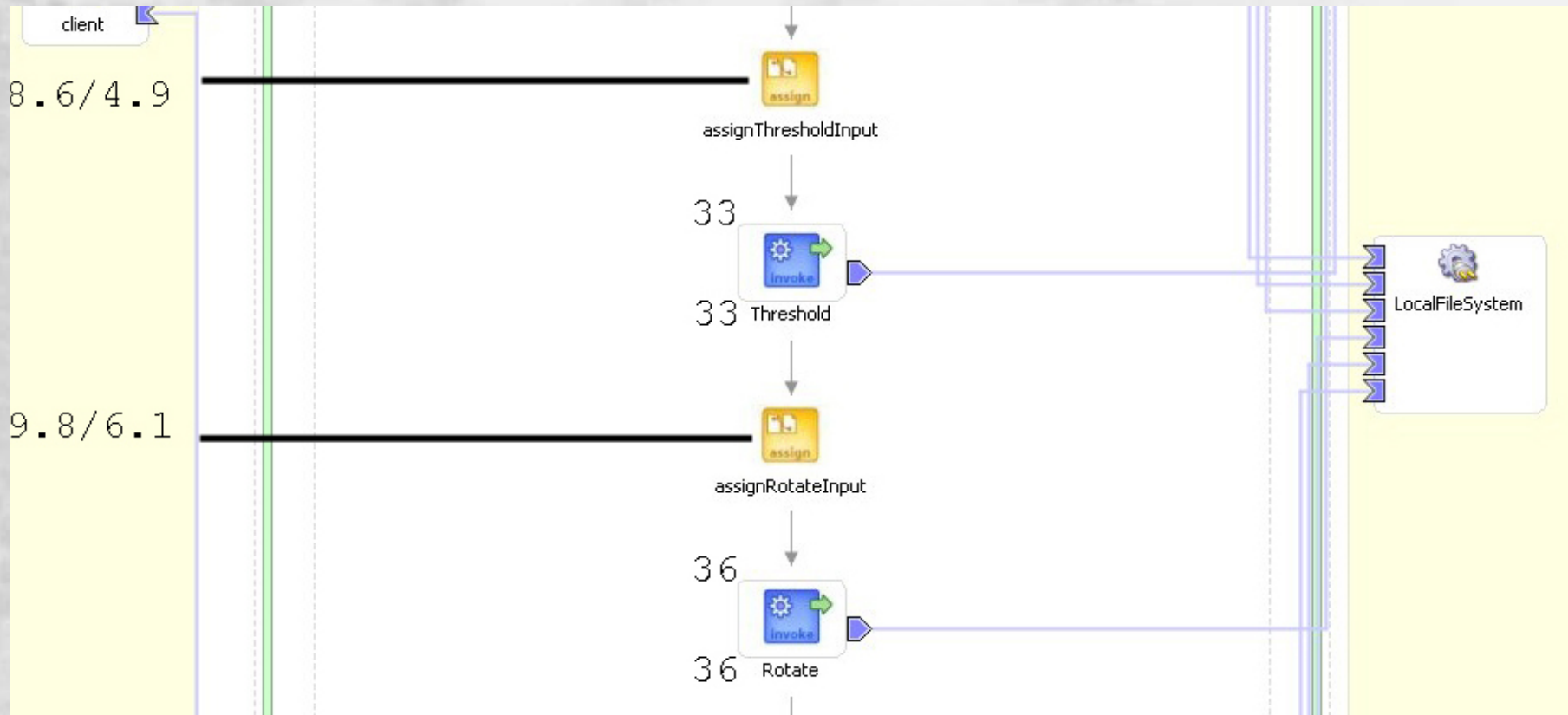
3.7/0



# Oracle BPEL Designer workspace(3)



# Oracle BPEL Designer workspace(4)

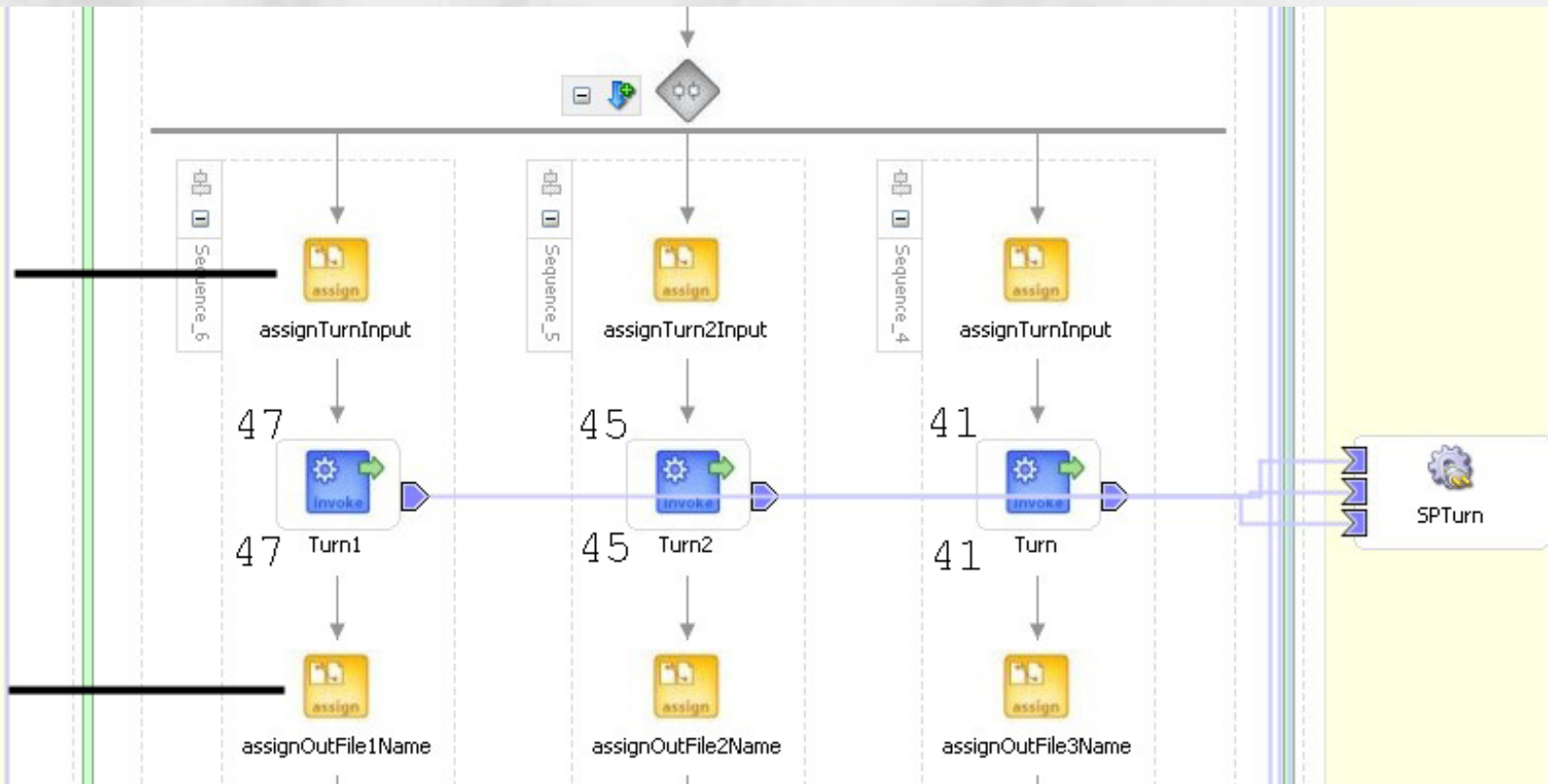


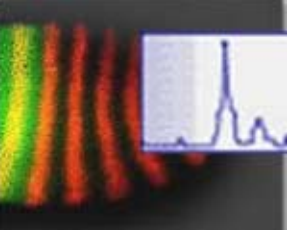


# Oracle BPEL Designer workspace(5)

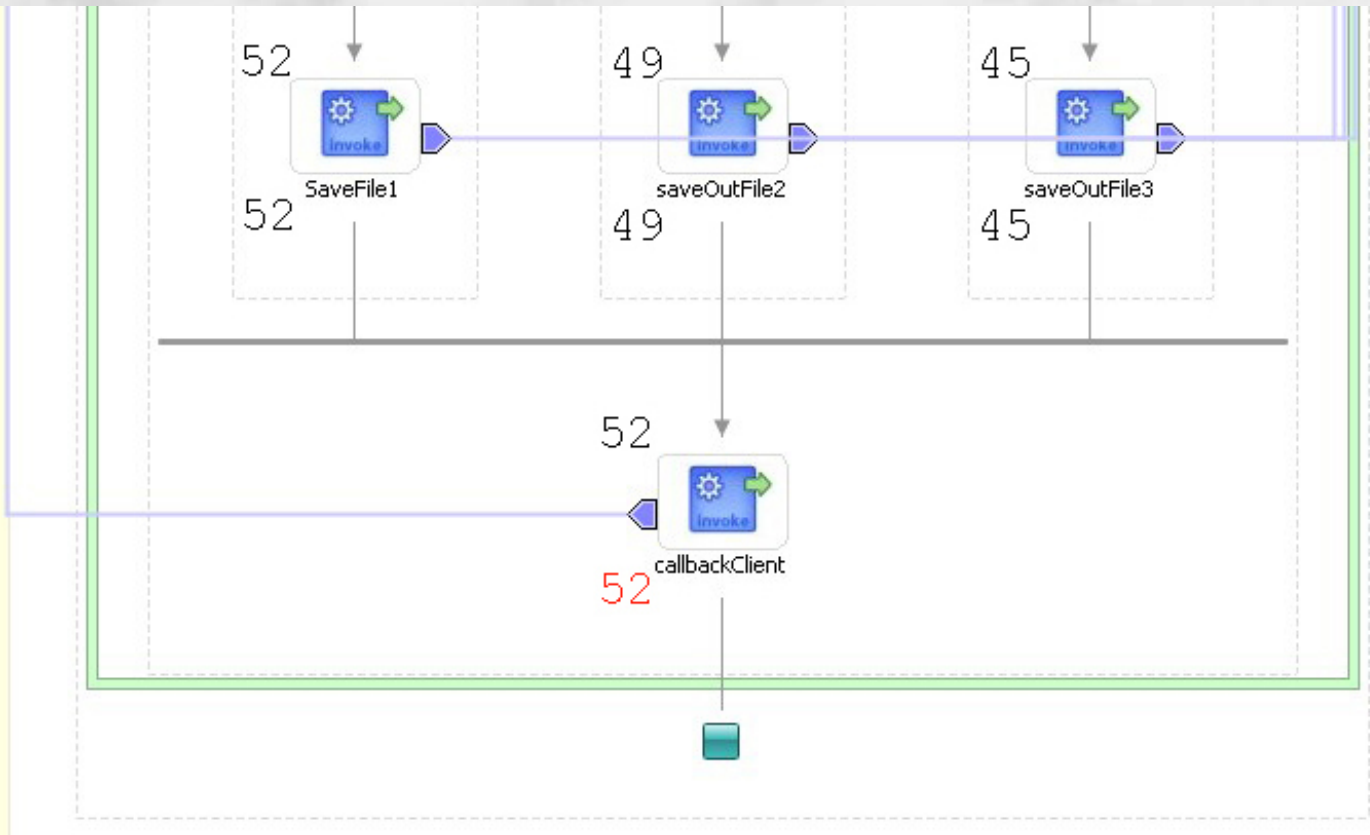
13.5/6.1

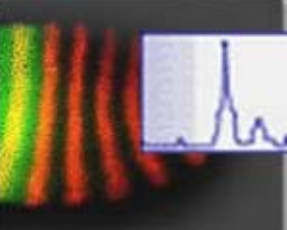
13.5/13.5





# Oracle BPEL Designer workspace(6)

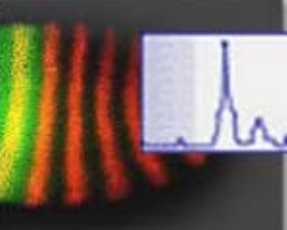




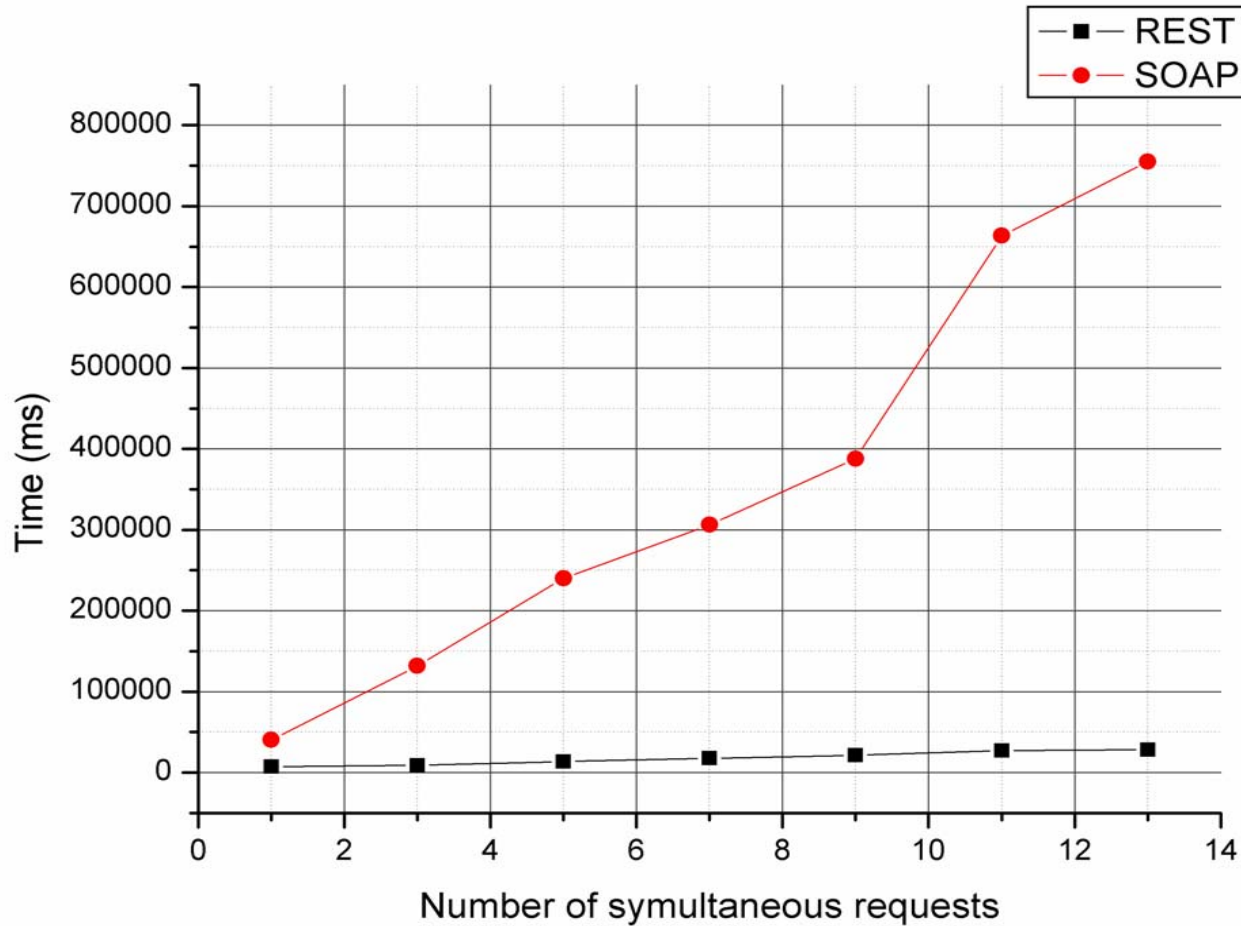
## Experimental results

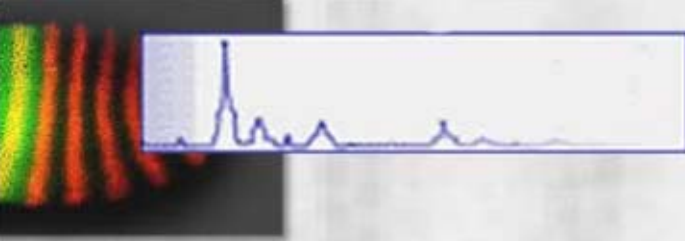
Tool used	Picture file size		
	100 Kb	350 Kb	1200 Kb
	Serving time (min)		
iSIMBioS	1	2	6
BPEL	2	10	52

Internet connection speed – 256KBit/s



# REST and SOAP client interactions with the Wavex webservice





# Conclusions

- We have designed iSIMBioS, the system prototype for collaboration within distributed working environment by applying a hybrid approach based on multiagent and REST architectural styles.
- Program agents with different custom interfaces have been used to integrate heterogeneous and geographically distributed program and services (SOAP, CORBA, RMI, CGI) into a single workflow.
- REST approach is more suitable for design of distributed image processing and analysis system, than SOAP.
- Use of pointers to data files instead of data itself decreases the amount of transferred information up to two times or even more.