Facilitating Bioinformatics Research through a Mobile Cloud with Trusted Data Provenance

Jinhui Yao[1,2], Jingyu Zhang[2], Shiping Chen[1], Chen Wang[1], David Levy[2], Qing Liu[3]

[1]*Information Engineering Laboratory, CSIRO ICT Centre, Australia*
{Firstname.Lastname}@csiro.au
[2]*School of Electrical and Information Engineeering, University of Sydney*
{jinhui, jyzhang, dlevy}@ee.usyd.edu.au
[3]*CSIRO Plant Industry, Canberra, Australia*
q.liu@csiro.au

***Abstract*—Cloud provides a cheap yet reliable outsourcing model for anyone who needs scalable computing resources. Together with the Cloud, Service Oriented Architecture (SOA) allows the construction of scientific workflows to bring together various scientific computing tools offered as services in the Cloud, to answer complex research questions. In those scientific workflows, certain critical steps need the participation of research personnel or experts. It is highly desirable that scientists have easy access, such as mobile devices, to the workflows running in the Cloud. Furthermore, since the participants in this cross-domain collaboration barely trust each other, achieving reliable data provenance becomes a challenging task. In this paper, we propose a concept of mobile-cloud by combining mobile and cloud together in a bioinformatics research application scenario. A mobile-cloud framework is developed, which facilitates the use of mobile devices to manipulate and interact with the scientific workflows running in the Cloud. The Mobile Cloud system acts as a trusted third party to record provenance data submitted by the participating services during the workflow execution. We have implemented a prototype which allows the bioinformatics workflow design and participation using mobile devices. We prove the concept of mobile-cloud with the prototype and conducted performance evaluation for the significant points of the bioinformatics workflow.**

Keywords -Cloud Computing,Accountability,Service Oriented Architecture,Mobile Cloud,Data Provenance

I. INTRODUCTION

The emergence of computing resource provisioning known as the Cloud has revolutionized classical computing. It provides a cheap yet reliable outsourcing model for anyone who needs scalable computing resources. Given the fact that many scientific breakthroughs need to be powered by advanced computing capabilities that help researchers manipulate and explore massive datasets [2], Cloud computing offers the promise of "democratizing" research, as a single researcher or small team can have access to the same large-scale computing resources as well-funded research organizations without the need to invest in purchasing or hosting their own physical IT infrastructures.

On the other hand, the concept of Service Oriented Architecture (SOA) allows flexible and dynamic collaborations among different service providers. A service can either directly be used for its mere functions or be composed with other services to form new value-added workflows [3]. Through SOA, scientific workflows can be used to bring together various scientific computing tools and resources offered as services in the Cloud to answer complex research questions. Workflows describe the relationship of individual computational components and their input and output data in a declarative way. In astronomy, scientists are using workflows to generate science-grade mosaics of the sky [4], to examine the structure of galaxies [5]. In bioinformatics, researchers are using workflows to understand the underpinnings of complex diseases [6].

In scientific workflows, certain critical steps need the participation of respective research personnel or experts. For example, how the workflow should be designed and which scientific tools need to be involved must be decided by the experts in the area. And some complex patterns generated from the experiments need to be visually inspected by the scientists, who will determine the next a few steps for further analysis. In this regard, it is highly desirable that scientists can have an easy access to the services in the Cloud so that they can design and participate in the workflows efficiently.

Furthermore, data provenance has been widely acknowledged as an important issue for scientific experiments [28-30], for the provenance data collected during the experiment can be used to understand, reproduce the experiments conducted; identify the way data are derived. However, within this service-oriented collaboration, each service provider or individual researcher is from different organizations. The cross-domain collaboration intuitively suggests that the participants should be unnecessary to fully trust each other even though they need to collaborate. This implies they will question each other, e.g., 1) if a particular participant has employed proper data provenance mechanisms during the experiment; 2) if the recorded provenance data have been or will be tampered with; and 3) if the issuer and the integrity of the provenance data are somehow verifiable. These doubts caused by the lack of trustworthiness makes achieving reliable data provenance a challenging task, hence reduce the incentives of individuals to participant in such cross-domain collaborative scientific workflow and are harmful for the wide adaptation of this computing paradigm. Therefore, a means to

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

record the provenance data during the experiments in a trustworthy way is needed.

To address the above needs, with the impressive advances in the technology, we believe using mobile devices can be an ideal solution. The processes in a workflow can be thoroughly integrated with portable devices. All activities are decided and monitored on time from the way that fit the human environment instead of forcing users to passively accept the computing results from cloud service.

In this paper, by extending our previous work [1] we propose a novel design which facilitates the use of mobile devices to manipulate and interact with the scientific workflows running in the Cloud. In our system, the users can choose the services in the Cloud to form the workflows via their mobile devices, and each mobile device can serve as one service node to be involved in the workflows designed. A significant aspect of the Mobile Cloud is its ability to provide trusted data provenance. Mobile Cloud serves as a trusted third party to record provenance data submitted by the participating services during the workflow execution. By enforcing strong accountability via the use of cryptographic techniques, the provenance data submitted by the participants in the workflow are undeniably linked to the submitter, which means its issuer and integrity can be cryptographically verified. With these verifiable provenance data recorded by a trusted platform, the collaborating entities can have a much better sense of trust in the validity of the provenance information they need to use.

The main contributions of this article are: 1) we design a Mobile Cloud system as a middleware layer to facilitate the use of mobile devices to design and interact with the scientific workflows running in the Cloud; 2) we define and illustrate the concept of strong accountability and the way it can be applied to record activity traces with provability; 3) we propose a novel approach to obtain activity traces from the execution of workflows and use them to construct data provenance graph to illustrate provenance information; and 4) we evaluate the performance of the Mobile Cloud system in the Cloud with real services.

II. THE APPLICATION SCENARIO

In the area of gene research, the recent development of the microarray technology [7] have led to rapid increase in the variety of available data and analytical tools. Some recent surveys published in Nucleic Acids Research describes 1037 databases [8] and over 1200 tools [9]. The analysis of microarray data commonly requires the biologist to query various online databases and perform a set of analysis using both local and online tools.

To illustrate with an example, here we explain the research study of the genetic cause of colorectal cancer, i.e., identify the genetic variation in human DNA that makes people susceptible to colorectal cancer. The rat azoxymethane (AOM) model of CRC is often used in dietary intervention studies as it induces mutations in genes which are also found to be mutated in human adenomas and adenocarcinomas. To define the baseline variation in global gene expression, the biologists extract RNA from mucosa scraped from colon and analyze the global gene expression using the Affymetrix Gene Chip. Data is normalized and then analyzed for differential expression.

By contrasting the results from normal and cancer mice, biologists can identify candidate genes through statistical analysis. Further analysis—such as searching for the functions known to these genes — are commonly performed to examine whether and how the candidate genes relate to the colorectal cancer. The followings are the data acquisition and analysis steps to perform the study of microarray experiment:

Quality Control. The raw microarray result data are processed, visualized and inspected by an expert, who can identify errors and discard the experiment.

Normalization. Microarray results from different samples need to be normalized before any meaningful comparison can be conducted.

Gene Differentiation. By contrasting the results from cancerous and healthy tissues, differentially expressed genes—candidate genes that are active in cancer are identified by applying some statistical methods (e.g. LIMMA).

Gene Study. Most differentially expressed genes are further studied to understand the biological functions of the disease. There are various resources available for study. For example, gene symbols and descriptions could be retrieved from the Rat Genome Database and/or BioMart. Gene Ontology (GO) and KEGG databases could provide gene functions and molecular pathways information respectively. Experts need to be involved to make good decision as which study to conduct and which database to use.

We can see that the four standard analysis procedures we listed above not only can be extremely computing intensive but also require some decision making from the research scientists or experts at certain critical steps (e.g. quality control). It easily follows that, a viable approach to conduct such researches must utilize certain computing platform that has enormous computing capacity, yet research scientists can easily interact with the platform and the computing process conducted. This is essentially the reason for which we promote the "Mobile Cloud" - a composition of the Cloud, and the mobile devices – to be a suitable paradigm for complicated bioinfomatics researches.

III. A MOBILE-CLOUD SYSTEM FOR BIOINFORMATICS REARCH

As we have established in previous sections, we propose to compose the Cloud and the mobile devices to conduct complex bioinformatics researches. The bioinformatics research scenario we chose is the study for the cause of colorectal cancer. Figure 1 shows our proposed system with this research scenario.

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

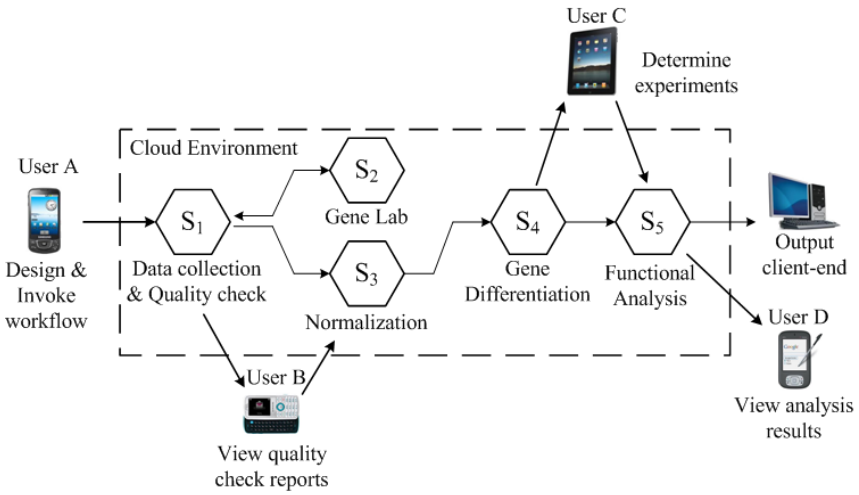


Figure 1. Overview of the proposed Moble-Cloud system

In the Cloud, different computing intensive gene research tools are deployed by different research bodies and provided as services. Outside the Cloud, research scientists or gene analysts locate the desired services in the Cloud, and use them to compose a workflow for studying the cancer. In a gene research lab, we assume the gene data in the subject microarray chips are scanned and archived in some digital database, which can be reached from the Cloud or itself could be a Cloud storage service [31] such as Amazon S3. The Mobile Cloud operates as this: a researcher (user A) designs the scientific workflow and composes the needed services in the Cloud, then he invokes the first service – "Data collection and Quality Check", which retrieves the gene data from the nominated "Gene Lab" where the gene subjects are stored, then conducts quality checks on the gene data. Once finished, the data is sent to the next service – "Normalization" and a quality report is sent to user B for confirmation. If user B confirms the data quality, the normalization service will normalize the data and send the results to "Gene Differentiation". Another report is sent to user C After the differentiation, to choose the suitable experiment for the functional analysis. When the workflow is complete, the results are sent to a client end and a final report is sent to user D. We can see in the workflow multiple research scientists are involved. They participate in the workflow by using portable or desktop devices to invoke or receive output from the services.

Our argument for using mobile devices to design and participate in the workflows is intuitive. As mentioned, in the workflow there are "critical steps" that require decision making by experts in the respective area, in order to continue the process. For example, after the quality check, an important decision needs to be made about whether the quality of the raw data suffices the requirements of the experiment. The experiment should be paused before the expert in charge has reviewed the quality check reports and confirmed the usability of the raw data. Therefore, mobile devices are indeed ideal for this task for its outstanding mobility compared to desktop computers or even laptop computers, i.e. one can freely use his mobile devices while waiting in a queue, on a bus, or even walking. Further, given the recent impressive advances in the mobile technology, the computing capability of mobile devices - however limited compared to desktops or laptops - is more than enough to run basic UI or display data sets and processing reports. Therefore, we believe mobile devices such as smart phones or tablet computers are indeed ideal to be used as light client-end to drive the heavy bioinformatics research workflows in the Cloud.

A. Overall architecture of Mobile Cloud

To enable mobile devices to construct and participate in the workflows running the Cloud, we have developed the Mobile Cloud middleware layer (MC-layer) to facilitate these. This middleware shall be deployed or even provided by the cloud environment provider in that environment to facilitate efficient interactions with the services and the clients. Figure 2 provides an overview of the architecture, which consists of a user interface (residing on mobile devices), a Cloud environment containing various services and a middleware layer consists of three function units. Their respective functionalities are summarized as follows:

- **Cloud Environment** provides various services deployed by respective providers. The services have registered their access end point with the MC-layer.
- **Service Repository/Composition** stores the information about the services in the Cloud that have registered with it. It helps the user to search for the services that best satisfy the requirements specified, and compose them into workflows.
- **Workflow Execution** conducts two jobs: (a) orchestrating workflows during the operation; (b) invoking Web services according to the workflow defined.
- **Trusted Data Provenance Unit** (TPU) records cryptographically signed provenance data submitted by the participating services during the execution of the workflows. Using the recorded data, it monitors the status of the execution and allows the clients to query data

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

provenance traces (this part will be elaborated in detail in section 4).

- **User Interface** allows users to register, design workflows and participate in a running workflow.

For mobile devices to construct workflows, they first need to send a search request to the Service Repository in order to get a list of the services/workflows they are looking for. A convenient UI has been implemented on the mobile devices to allow the users to easily design the workflows using the services listed by the Service Repository (the UI will be elaborated in the evaluations). Once the workflow have been designed, a representative XML based description script is generated to be submitted to the Service Composition unit. The Service Composition unit thus according to the script, composes the services to form the desired workflows. The services can be composed in two ways: i) centrally composed, where the MC-layer invokes the services in the sequence designed by the user; and ii) remotely orchestrated, where certain orchestration scripts such as BPEL will be generated and distributed to all the services involved for deployment.

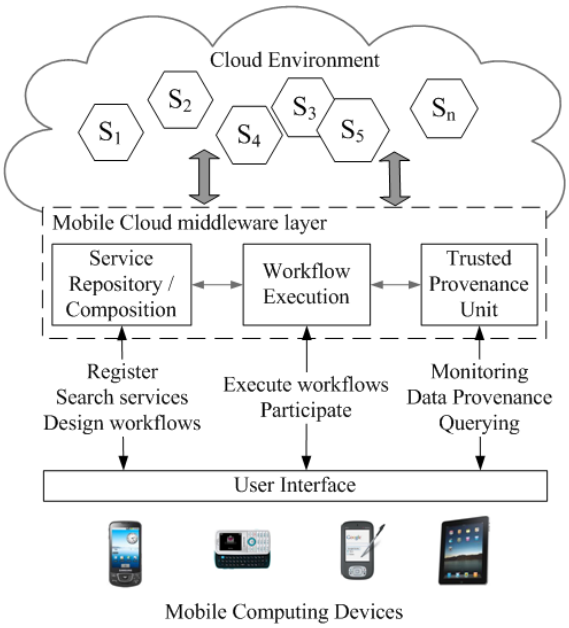


Figure 2. Overview of Mobile-Cloud architecture

B. *Workflow design through abstract description script*

In our system, the workflow designed by the users is an abstract workflow, that is, the users only need to specify the type of service needed, and the MC-layer will search its service reporsitory and select the best suited ones according to the user's specifications. Table 1 gives a sample of the workflow description script. As it is developed based on the BPEL, "sequences" and "flows" are used to specify serial and parallel composition, and "Actions" are used define the invocation operations. The sample describes the first half of the gene analysis workflow in Figure 1. In some actions, the endpoint is set to be "OPTIMAL". This is to tell the Service Composition unit to choose the best suited services.

TABLE I. SAMPLE WORKFLOW DESCRIPTION SCRIPT

```
<sequence name="main">
   <Action operation="start" invoker="client"
   endpoint="QualityCheck" type="send&forget".../>
   <Action operation="fetchGene" invoker="QualityCheck"
   endpoint="GeneLab" type ="send&receive".../>
   <flow>
      <Action operation="sendForApproval"
      invoker="QualityCheck" endpoint="user B" type
      ="send&forget".../>
      <Action operation="normalization"
      invoker="QualityCheck" endpoint="OPTIMAL" type
      ="send&forget".../>
   </flow>
   …
</sequence>
```

As we have established in our system design, mobile devices will be involved in the workflows as web services. To facilitate this, we created a customized web service engine to run on the mobile devices. Using this engine, mobile devices can both send and receive service requests, as well as interpreting the workflow description scripts delivered by the MC-layer. Once a user has designed and submitted a workflow, the workflow description script will be forwarded to the research personnel that are involved. The mobile devices they are using will interpret the workflow script and save the workflow logic. When a service request is received during the execution of the workflow, the UI will allow the user to view the content (e.g. quality check reports) and provide the list of the services that the user should send output request to according to the workflow logic (e.g. normalization services). For the technical details of the MC-layer, please refer to our previous publications about the Web Service Management System (WSMS) [13].

IV. ACCOUNTABILITY FOR COMPLIANCE AND PROVENANCE

The workflows in the Cloud are constructed using services provided by different parties who barely know each other. The correctness of the resultant workflow relies on the individual correctness of all participators. That is, if the service is compliant to the pre-defined workflow logic, or Service Level Agreement (SLA). The scientific integrity of the gene analysis results will be highly questionable if the services involved can act willy-nilly and get away with processing errors.

On the other hand, for scientific experiments not only the resultant data are considered, the steps of how these data are derived along the process can also be very valuable. It has been widely realized that data provenance plays an important role in the scientific researches [14]. It follows that, trusted data provenance mechanisms are necessary in such systems with participants from different administrative domains. Provenance data should be preserved in a trustworthy way that, the contributors of the data are committed to their truthfulness. This naturally leads us to the issue of accountability. In this

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

section, we illustrate our design to incorporate strong accountability into the "Mobile Cloud" to address these issues.

A. Accountability for trustworthiness

Accountability can be interpreted as the ability to have an entity account for its behaviors to some authorities [10]. This is achieved by binding each activity conducted to the identity of its actor with proper evidence [11]. Such binding should be achieved under the circumstance that all actors within the system are semi-trusted. That is, each identified actor may lie according to their own interest. Therefore, accountability should entail a certain level of stringency in order to maintain a system's trustworthiness. Below, we identify several desirable properties of a fully accountable system:

- **Verifiable:** The correctness of the conducted process can be verified according to the actions and their bindings recorded.
- **Non-repudiable:** Actions are bound to the actors through evidence, and this binding is provable and undeniable.
- **Tamper-evident:** Any attempt to corrupt to recorded evidence inevitably involves the high risk of being detected.

We illustrate our proposed approach in Figure 3. In our approach, accountability can be incorporated into activity-based workflow by requiring the entity conducting the process to log non-disputable evidence about the activities in a separate entity. In the figure, after incorporating accountability into an ordinary process, entity A is now required to perform logging operations before and after conducting the activity in its process. The evidence is logged in a separate entity - entity B - so that entity A cannot access the logged evidence. The evidence needed to be logged should contain enough information to describe the conducting activity. In our simple example, which is intuitive enough, the evidence should include the states of the factors concerning the start of the activity (e.g. the input variables) and the factors concerning its completion (e.g. the output value).

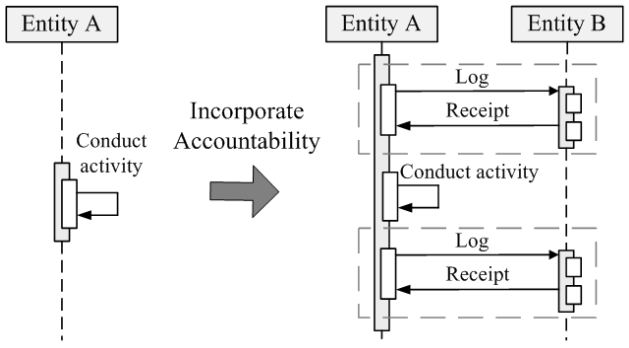


Figure 3. Example of incorporating accountability into processes

The logging operations require the employment of PKI in all involved service entities. Each of them has its own associated public-private key pair issued by certificated authorities. The logging operations are as follows:

1. The logger (entity A) signs the evidence (E) by its private key (K_{A-}) to create a digital signature of the evidence (S_A).
2. The evidence and its signature are then logged in a separate entity (entity B).
3. When received, entity B creates a receipt by signing entity A's signature with entity B's private key (K_{B-}).
4. Lastly, the receipt (S_B) is sent back to the logger (entity A) in the reply.

Assuming the digital signature is un-forgeable, the signed evidence in entity B can be used to verify entity A's compliance; and yet any corruption or deletion applied to the evidence will be discovered using the receipt received by entity A. Under the circumstance that neither of the service entities is trusted; and assume they will not conspire to cheat, this structure manages to ensure the proper preservation of evidence associated with the process conducted.

B. Logging provenance data a Trusted Provenance Unit

In Mobile Cloud, the *Trusted Provenance Unit* (TPU) acts as the separate entity B, dedicated to provide accountability to all underlying services involved in the workflow. Figure 4 shows the structure. All the mobile devices, service nodes in the Cloud as well as local computing nodes that are involved in the workflow, register with TPU and submit provenance data during the execution of the workflow.

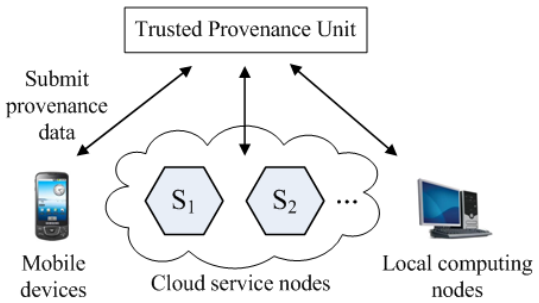


Figure 4. TPU records provenance data from various sources

The provenance data can be recorded in various ways, for instance, if the service invocations are all relayed by the MC-layer, they can be simply archived when received. Here we illustrate a generic approach to incorporate the data logging into the workflows by transforming the workflow descriptive scripts. Business process or workflows are often defined through process descriptive languages, which will be interpreted by orchestration engines (e.g. Apache ODE) to conduct the process accordingly. A good example of the process descriptive language is Business Process Execution Language (BPEL) [34]. BPEL models the business activities into several basic activity types, and then composes those types to describe the whole process. The core activity types include:

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

1. *Receive*, receiving the request from a requestor. This activity type will specify the variable to which the input data is to be assigned.

2. *Invoke*, invocation to an endpoint (service). Invoke activity type will specify the variable used as the input and the variable used to store the output data for this invocation.

3. *Reply*, replying the invocation. A variable will be specified to be returned to the requestor as the result.

To add logging activities into the workflow, we can insert *invoke* activity types into the BPEL script to invoke a certain endpoint (e.g. logging service) with the provenance data to be logged. And due to the distinct natures of *receive*, *invoke* and *reply* activity types, the rules used to decide the insertion locations are in fact quite straightforward. For the *receive* activity, an *invoke* should be inserted right after it, to log the input data received. For the *invoke* activity, one *invoke* should be inserted before this activity and another to be inserted after, to log the input data and the reply data of the invocation respectively. And finally for the *reply* activity, an *invoke* needs to be inserted just before it to log the result data that is about to be returned to the requester. The invocation endpoint for the *invoke* activities inserted (i.e. logging service) should either be a service in the same domain of the logger, or a trusted party nominated by the logger, which in turn signs the evidence on the logger's behalf and forward the signed evidence to the TPU.

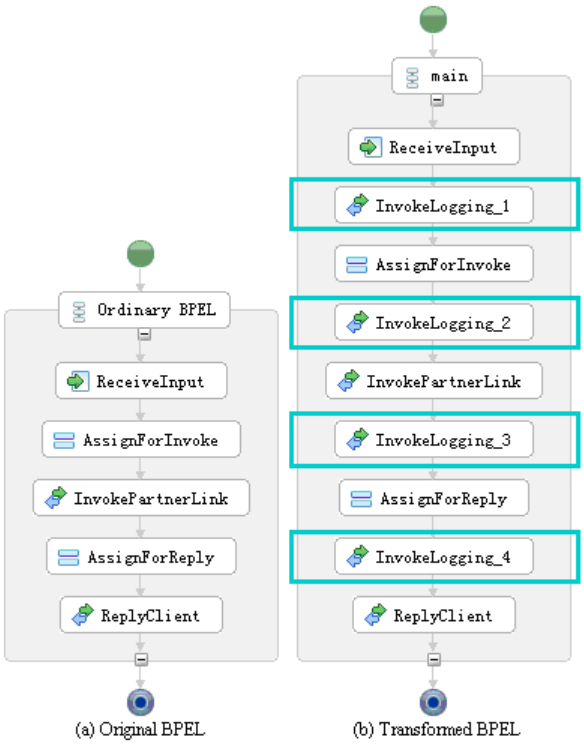


Figure 5. Transformation of BPEL

To further illustrate this transformation process, we have presented an example in Figure 5. Figure 5(a) shows the graphical view of an ordinary sample BPEL. This simple process is started by receiving an input (ReceiveInput); then a partner link (collaborating service) is invoked in turn (InvokePartnerLink), and finally, replies the result to the client (ReplyClient). Figure 5(b) is the BPEL after the transformation. We can see in Figure 5b that four logging invoke activities (the InvokeLogging) have been inserted, one after the "ReceiveInput"; one before and one after "InvokePartnerLink"; and one before "ReplyClient". Because BPEL is entirely based on xml schema, any xml schema parser will be capable of analyzing and inserting activities into it. The implementation details of the incorporation of accountability have been elaborated in our previous work [12].

Here the evidence can be any intermediate gene analysis data generated by the tools in the Cloud, or the decisions made by research personnel participated. With the evidence data logged, the core functionalities provided by the TPU are:

- Compliance verification. Through the analysis of the evidence data, the correctness of the behaviors of the underlying services is continuously validated.
- Data provenance. The evidence recorded capture the evolution path of the data as well as the entities responsible for each step.
- Workflow status monitoring. A global view over the workflow is maintained by the TPU. Such information can be used to assist the functioning of the MC-layer and the underlying services.

C. *Architectural design of Trusted Provenance Unit*

TPU is responsible of recording the provenance data from all the participants of the workflow. As accountability requires the submitter of the data to sign the data before submission to commit its truthfulness, services and the entities involved in the workflow are needed to register their identity documents (e.g. X.509) at MC-layer. When a new abstract workflow is proposed by a researcher, the Service Repository/Composition unit first find the services that best suit the specified requirements, then the filled workflow script is transformed by TPU to have logging activities (refer to [12] for details). Meanwhile, TPU uses the knowledge obtained from the documents registered to generate analysis logics to process the incoming data during the execution of the workflow. The resultant data provenance information will be delivered to the user through querying and visual displays.

The internal architectural design of TPU is shown in Figure 6. In the initialization phase, registered information about the services, like WSDL, X.509 certificate etc.; and information about the workflows, like BPEL scripts are transmitted to TPU from Service Repository/Composition unit. TPU first transform the workflow script to incorporate logging activities and send the transformed script for redeployment; then it uses the registration information received to generate two components: "Monitoring Logic" and "Provenance Logic". In the monitoring phase, the provenance data will be submitted

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

from the participants in the workflow. These data will first be analysed by the monitoring logic to find obvious compliance violations (e.g. QoS service level agreement); then be processed by the provenance logic to generate data provenance information to be stored in the data warehouse.

When the provenance data are received, the provenance logic first labels the provenance data with information regarding the "four Ws": who, when, where and what. In general, the provenance logic will add labels explaining what this provenance data is about, in which workflow (where) it is generated, at what time and by which participant (who). Then, based on the knowledge obtained from the documentation registered, the provenance logic links the different provenance data with Open Provenance Model [32] edges to form a provenance graph. An example of such graph is displayed in Figure 7. We can see from the example, the provenance information of the data pieces (circles marked with numbers) are expressed in terms of their links to the activities (round rectangles) that used or/and generate them. The figure is a visual display of the provenance graph, it is not necessarily an actual graph when stored in the data warehouse. The provenance logic simply needs to label the data so they are linked with each other.

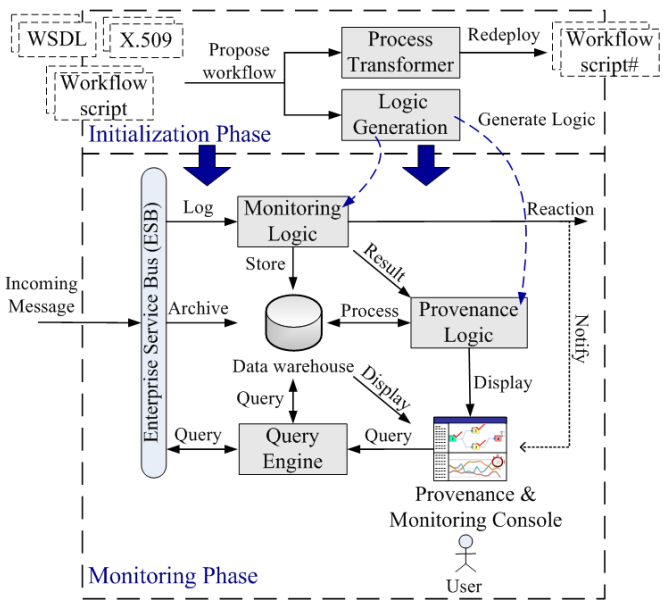


Figure 6. Internal architecture of Trusted Provenance Unit

The query engine provides an interface for the users to fetch the provenance information about specific data. In order to enable simple and efficient querying, a query language in XML is developed, called SWQL (Simple Workflow Query Language). SWQL allows the user to specify the information regarding the "four Ws" to fetch the desired provenance data. An example is shown in Listing 2. The example is a query to fetch all the differentiated gene (what) recorded from the colorectal cancer workflow (where), submitted by service A and B (who) from 9am to 5pm on 20 July 2011 (when).

The provenance and monitoring console is a graphical user interface to display provenance and monitoring information as well as let users query the data warehouse. During the execution of the workflow, the evolution of the data will be displayed in terms of the provenance graph generated by the provenance logic, and the status of the workflow will been shown. More details about the console will be discussed in the evaluation section.

TABLE II. AN EXAMPLE OF SWQL QUERY

```
<SWQL>
    <Action>Find</Action>
    <DataIdentifier>
     <Type>Differentiated gene</Type>
    </DataIdentifier>
    <EntityIdentifier>
     <Entity>Differentiation service A</Entity>
     <Entity>Differentiation service B</Entity>
    </EntityIdentifier>
    <TimeInterval>
     <From>9AM-20JUL2011</From>
          <To>5PM-20JUL2011</To>
    </TimeInterval>
    <WorkflowIdentifier>Colorectal cancer
    </WorkflowIdentifier>
     ...
</SWQL>
```

V. EVALUATION

We developed a prototype system to showcase our mobile-cloud concept. Our system consists of three parts: i) a client UI deployed in the mobile device; ii) an MC-layer for composing workflows and provenance; and iii) a number of demonstrating service nodes in Amazon EC2. We implemented five services nodes in EC2 to represent the gene research tools provided by different organizations. The services are linearly composed (one node finishes its job then invokes the next) to form a workflow using BPEL. The information about the services as well as the workflow are registered at the MC-layer, which is deployed in another computing instance in EC2. A remote user designs and invokes the workflow using the client UI locally deployed in the mobile device. With this setting, in this section, we will elaborate the implementation of the client UI; examine the communication overhead introduced when provenance data are logged at TPU during the execution; and we show some processing latency when a real gene database (KEGG) is involved in a workflow.

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

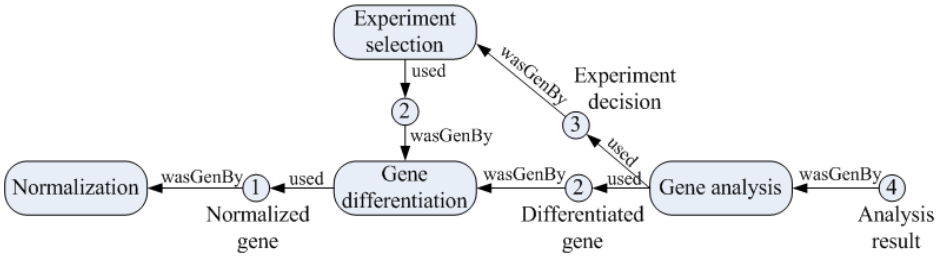
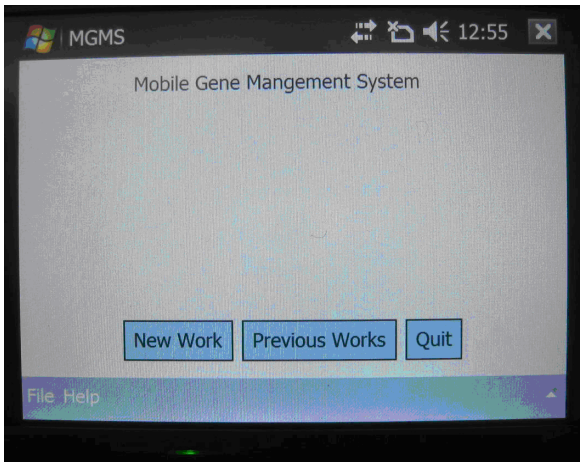


Figure 7. An example of provenance graph

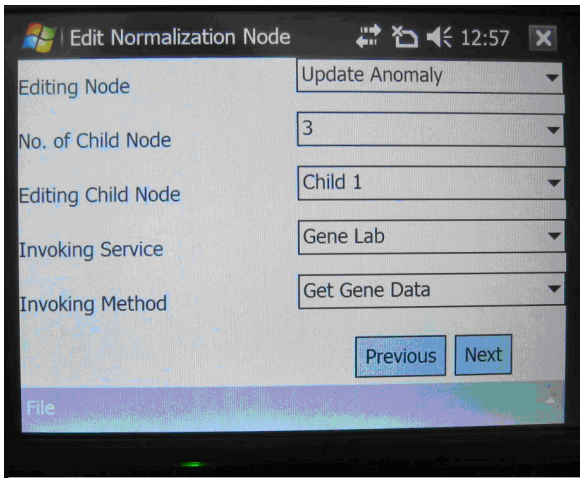
The UI on mobile device is developed using Java platform, micro edition (J2ME). The mobile web service feature is deployed and runs on a HTC 9500 mobile phone, which is running on IBM WebSphere Everyplace Micro Environment that supports a connected device configuration (CDC1.1). Figure 8 (a) and (b) show two screen shots of the Mobile Gene Management System (MGMS) - a scientific workflows design and surveillance tools. A user can define or edit a scientific process from the "New Work" button or "Previous Work" button as shown in Figure 8 (a). Then, the user can select into process items and specify their detail information as shown in Figure 8 (b). System users define the steps from four aspects, what services carry out these tasks; the number of child nodes; which methods/services are invoked; and what are the inputs and outputs of each step. Finally, an abstract workflow in BPEL will be generated and uploaded to the WSMS in Cloud, which will instantiate the abstract workflow by filling up the endpoints in the BPEL with the best concrete services URLs.

We have conducted testing to evaluate the latency introduced by incorporating the logging actions into the workflow. Figure 9(a) shows the overall latency to finish the process with untransformed BPEL scripts and with transformed ones. We have tested the workflow with request message size from 0.1KB (equivalent to a sentence) to 50KB (equivalent to a medium size document). For the process with transformed BPEL scripts to log the entire input/output messages (the series marked with "circles"), the latency introduced compared to the untransformed one (the series marked with "squares") grows as the request message becomes larger. In percentage terms, on average we observed a 30% increase in the overall process latency. Intuitively, this latency is significant to the business process; however it can be improved through the use of hash functions.

The hash of the message computed using collision-resistant hash functions (e.g., SHA-1), which is a very small digest (160 bits for SHA-1), can be logged as a substitute. Because the hashes computed are collision-resistant, which means it is theoretically impossible to have two different items with the same hash, so the hash can be logged to represent the data. We can see in the graph, the extra latency is significantly reduced if the BPEL scripts are transformed only to log the hash of the evidence (the series marked with "triangles"). In fact, since the size of the hash is fixed regardless of the data size, the extra latency almost remains constant regardless of the size of the request message. The overhead introduced will



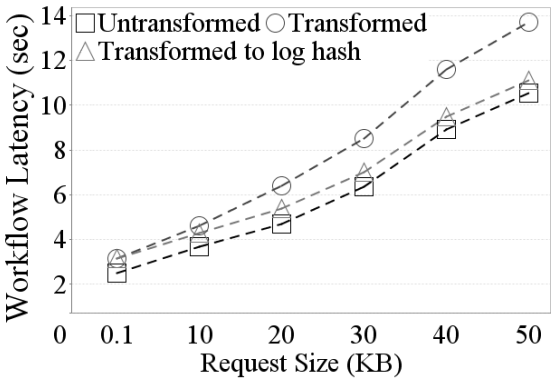
(a.) Main menu



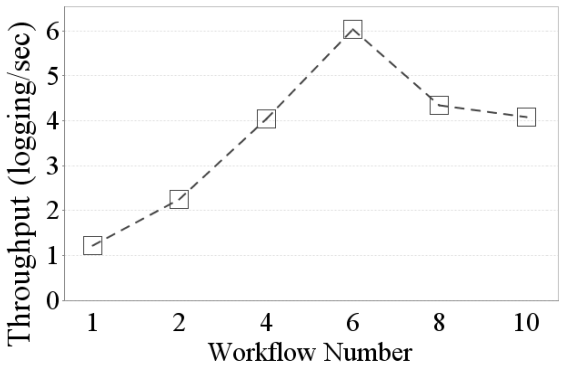
(b.) Designing a workflow

Figure 8. Screen shots of Mobile Cloud client end UI

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



(a.) Overall execution latency of the workflow



(b). Throughput of TPU under different loads

Figure 9. Performance evaluation

become more and more negligible when the size of the messages transferred increases. In practice, it is not often that the provenance data is urgently needed to be logged at runtime. When the system is idle, the provenance data can be eventually logged and verified according to the hash values. This eventual-logged strategy can further improve the performance and reduce the overhead.

As the MC-layer will be managing a number of workflows, naturally, it is interesting to find out the processing capability of the TPU. To evaluate this, we replicated the workflow we have implemented (the colorectal cancer workflow), and execute multiple workflows replicated concurrently. As such, multiple service nodes will be submitting provenance data to the TPU deployed in a computing instance simultaneously. With this setting, we evaluate the processing throughput of the TPU when it is under different loads (in terms of logging received per unit time). Figure 9(b) shows the testing results. In the figure we can see that, the processing throughput of TPU improves as the number of workflows increments, it reaches its peak when TPU is monitoring 6 workflows, and then it decays gradually if more workflows are involved in the monitoring. We tested this with messages of size 50KB, the processing operations conducted by AS involves both SLA monitoring and provenance data processing, which may need to fetch history data from the data warehouse to make conclusions. Since the computing power of a computing instance is fixed, an decrease in message size or processing complexity will shift the peak towards right to occur when more workflows are involved, and vice versa.

To evaluate the performance of gene retrieving from gene bank services, we selected 6 example genes which are the genetic causes of colorectal cancer and retrieve their genetic neighbors from KEGG disease Database [22]. We test the response time from 0 neighbors to 50 neighbors. As shown in Figure 10, it is clear that the latency is slowly increasing while we are increasing the number of neighbors. The has-581 continually yields the best performance at all stages from the 1427msec for retrieving 0 gene neighbor to 2746.8msec for getting 50 neighbors. However, has-10297 spent 2078msec to search 0 neighbors and it cost 2912.6msec for finding 50 neighbors.

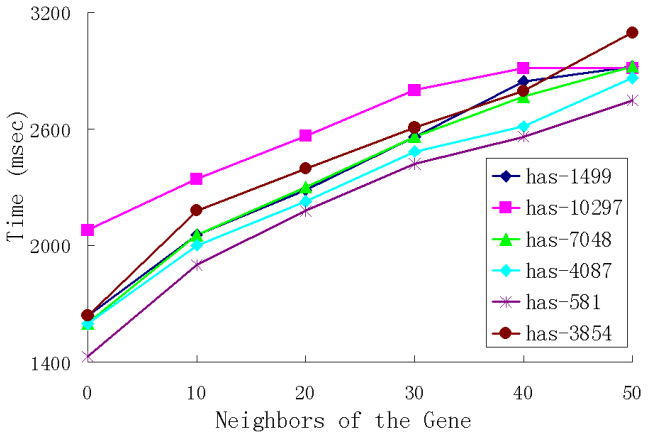


Figure 10. Gene retrieval experiment with KEGG

VI. Related Work

Mobile computing provides a luggable computation model for users. Its portability makes it very ideal for many application scenarios. To extend its limited computing power, research communities have proposed novel designs to leverage the Cloud. [23] proposed a virtual cloud system, and [24] detailed a distributed computing platform using mobile phones. They improve the capacities of mobile phones in the purpose of storage and computation. In the literature, [25-27] presented some computation offloading studies that move some parts of the applications to run on the Cloud. Executing parts of application remotely can save battery lifetimes and significantly extend computing resources. However, these

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

solutions do not support platform-independent cooperative interaction over an open network. In addition, after moving some parts of applications from stand-alone handheld devices to the cloud, several issues need to be considered in advance such as privacy, trustworthy or provenance.

The importance of provenance for scientific workflows has been widely acknowledged by various research communities. Many approaches have been proposed to record the derivations of the data during the scientific process. Approaches like [15][16] allow the designer to capture the intermediate data forms generated by the experiments at different granularities. In our work, we introduced the concept of accountability which not only provides data provenance but can enforce compliance among the service providers. Compliance assurance has been studied decently in recent years, some remarkable works include [18-21]. Our work differs from them at the point that we consider a more hostile environment where all service entities are expected to behave in any possible manner and deceive for their own benefit. Cryptographic techniques are deployed in our system to ensure the evidence (provenance data) are undeniable.

VII. CONCLUSION

Cloud computing has emerged as a way to provide a cost effective computing infrastructure for anyone with large needs for computing resources. Together with the Service Oriented Architecture, research scientists can construct scientific workflows composed of various scientific computing tools offered as services in the Cloud to answer complex research questions.

In this paper, we have described a Mobile Cloud system which enables mobile devices to design and participate in the scientific workflows running in the Cloud. The scientific researchers can use mobile devices to sketch an abstract workflow design to be submitted to the mobile cloud middleware layer, which will recommend and compose the optimal services according to the designer's requirements. On top of that, we further incorporated accountability mechanisms to provide trusted data provenance during the execution of the scientific workflows. Trusted data provenance implies that the recorded provenance data about a certain workflow is cryptographically verifiable to be attributed to the responsible services who, issued them. The provenance data thus can be used with confidence that its source is verifiable and its integrity has been preserved.

In the future development, it will be interesting to explore the utilization of the trusted provenance data collected, to improve the service recommendation for workflow design. The applicability of a particular service in a certain workflow and its performances in the past executions can provide much information to the research scientists and the recommendation system about characteristics of this service and its eligibility for the workflow under design. Another direction of development is to utilize existing workflow platforms or service repositories (e.g. BioCatalogue [33]) to construct workflows and provide trusted data provenance. In this way we can testify the concept of Mobile Cloud and trusted data provenance in the practice, improve our methodology so as to offer more value and insights to the community.

REFERENCES

[1] J. Yao, J. Zhang, S. Chen, C. Wang, D. Levy. Facilitating Bioinformatics Research with Mobile Cloud. In proc. International Conference on Cloud Computing, GRIDs, and Virtualization, pp.161-166, 2011

[2] W. Lu, J. Jackson and R. Barga. AzureBlast: A Case Study of Developing Science. In proc. Workshop on Scientific Cloud Computing, pp. 413-420, 2010.

[3] O. Moser, F. Rosenberg, S. Dustdar. Non-Intrusive monitoring and service adaptation for WS-BPEL. In proc. international conference on World Wide Web, pp. 815-824, 2008.

[4] Montage. http://montage.ipac.caltech.edu.

[5] I, Taylor, M, Shields, I. Wang, and R. Philp. Distributed P2P computing within Triana: A galaxy visualization test case. In proc. IEEE International Parallel and Distributed Processings Symposium, 2003.

[6] T. Oinn, P. Li, D.B. Kell, C. Goble, A. Goderis, M. Greenwood, D. Hull, R. Stevens, D. Turi, and J. Zhao. . Taverna/myGrid: Aligning a workflow system with the life sciences community. In *Workflows in e-Science*, Springer, 2006.

[7] M. Schena, D. Shalon, R.W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, October 1995.

[8] M. Y. Galperin. The molecular biology database collection: 2008 update. *Nucleic Acids Research*, November 2007.

[9] M. D. Brazas, J. A. Fox, T. Brown, S. McMillan, and B. F. F. Ouellette. Keeping pace with the data: 2008 update on the bioinformatics links directory. *Nucleic acids research*, 36, July 2008.

[10] R. Mulgan. Accountability: An ever-expanding concept? In: Public Administration, pp 555–573, 2000.

[11] A. R. Yumerefendi, J. S. Chase. Trust but verify: accountability for network services. In proc. ACM SIGOPS European workshop, article No. 37, 2004.

[12] J. Yao, S. Chen, C. Wang, D. Levy, and J. Zic. Accountability as a service for the cloud, in proc. IEEE International Conference on Services Computing, pp. 81–90, 2010.

[13] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed. Deploying and managing web services: issues, solutions, and directions. *VLDB J.*, 17(3):537–572, 2008.

[14] Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science, in trans. ACM SIGMOD Record, volume 34, issue 3, pp. 31-36, September 2005.

[15] I. T. Foster, J.-S. Vöckler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation, in *SSDBM*, 2002.

[16] J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer. Semantically Linking and Browsing Provenance Logs for Escience, in *ICSNW*, 2004.

[17] J. Myers, C. Pancerella, C. Lansing, K. Schuchardt, and B. Didier. Multi-Scale Science, Supporting Emerging Practice with Semantically Derived Provenance, in ISWC workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, 2003.

[18] E. Mulo, S. Dustdar, U. Zdun. Monitoring Web Service Event Trails for Business Compliance. In Proc. International Conference on Service-Oriented Computing and Applications , pp. 1-8, 2009.

[19] M. Huang, L. Peterson, A. Bavier. PlanetFlow:maintaining accountability for network services. In Proc. ACM SIGOPS Operating Systems Review, pp. 89-94, 2006.

[20] Y. Zhang, K. Lin, J.Y.J. Hsu. Accountability monitoring and reasoning in service-oriented architectures. In Trans. Service Oriented Computing and Applications, Volume 1, Number 1, pp. 35-50, 2007.

[21] A. C. Squicciarini, W. Lee, B. Thuraisingham, E. Bertino. End-to-end accountability in grid computing systems for coalition information

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

sharing. In Proc. Workshop on Cyber Security and Information Intelligence Research , 2008.

[22] M. Kanehisa, S. Goto, M. Furumichi, M. Tanable and M. Hirakawa. KEGG for representation and analysis of molecular networks involving diseases and drugs. In trans. Nucleic Acids Research, volume 38, Database issue, pp. 355-360, 2010.

[23] G. Huerta-Canepa and D. Lee. A virtual cloud computing provider for mobile devices. presented at the Proceedings of the 1st ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond, San Francisco, California, pp. 61-65, 2010.

[24] J. Zhang, et al.. mBOSSS+: A Mobile Web Services Framework. in Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific, pp. 91-96, 2010.

[25] I. Giurgiu, et al.. Calling the cloud: enabling mobile phones as interfaces to cloud applications. the 10th ACM/IFIP/USENIX International Conference on Middleware, pp. 83-102, May,2009

[26] K. Kumar and Y. Lu. Cloud Computing for Mobile Users. Computer, vol. 18, issue 99, pp. 51-56, 2010.

[27] R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Cuckoo: a Computation Offloading Framework for Smartphones. In MobiCASE '10: Proceedings of The Second International Conference on Mobile Computing, Applications, and Services, pp. 62-81, 2010.

[28] C. Scheidegger, D. Koop, E. Santos, H. Callahan, J. Freire, C. Silva. Tackling the provenance challenge one layer at a time. Concurrency and Computation: Practice and Experience, 20(5), pp. 473–483, 2008.

[29] Y. Simmhan, B. Plale, D. Gannon. Karma2: Provenance management for data driven workflow. International Journal of Web Services Research 5(2), pp. 1–22, 2008.

[30] J. Zhao, C. Goble, R. Stevens, D. Turi. Mining taverna's semantic web of provenance. Concurrency and Computation: Practice and Experience 20(5), pp. 463–472, 2008.

[31] M. Palankar, A. Iamnitchi, M. Ripeanu, S. Garfinkel. Amazon s3 for science grids: a viable solution. In: Workshop on Data-aware distributed Computing, pp. 55–64, 2008.

[32] L. Moreau, J. Freire, J. Futrelle, R. McGrath, J. Myers, P. Paulson. The open provenance model. (2007). URL: http://openprovenance.org/

[33] BioCatalogue, URL: http://www.biocatalogue.org/

[34] T. Andrews, F. Curbera, H. Dholakia, et al.: Business process execution language for web services (BPEL4WS) specifications (2003). URL http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf

2012, © Copyright by authors, Published under agreement with IARIA - www.iaria.org