

COMPUTING SCIENCE

A Multi-Level Security Model for Partitioning Workflows over
Federated Clouds

Paul Watson

TECHNICAL REPORT SERIES

No. CS-TR-1271

September 2011

A Multi-Level Security Model for Partitioning Workflows over Federated Clouds

P. Watson

Abstract

Cloud computing has the potential to provide low cost, scalable computing, but cloud security is a major area of concern. Many organizations are therefore considering using a combination of a secure internal cloud, along with (what they perceive to be) less secure public clouds. However, this raises the issue of how to partition applications across a set of clouds, while meeting security requirements. Currently, this is usually done on an ad-hoc basis, which is potentially error-prone, or for simplicity the whole application is deployed on a single cloud, so removing the possible performance and availability benefits of exploiting multiple clouds within a single application. This paper describes an alternative to ad-hoc approaches – a method that determines all ways in which applications structured as workflows can be partitioned over the set of available clouds such that security requirements are met. The approach is based on a Multi-Level Security model that extends Bell-LaPadula to encompass cloud computing. This includes introducing workflow transformations that are needed where data is communicated between clouds. In specific cases these transformations can result in security breaches, but the paper describes how these can be detected.

Once a set of valid options has been generated, a cost model is used to rank them. The method has been implemented in a tool, which is briefly described in the paper.

Bibliographical details

WATSON, P.

A Multi-Level Security Model for Partitioning Workflows over Federated Clouds

[By] P. Watson

Newcastle upon Tyne: Newcastle University: Computing Science, 2011.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1271)

Added entries

NEWCASTLE UNIVERSITY

Computing Science. Technical Report Series. CS-TR-1271

Abstract

Cloud computing has the potential to provide low cost, scalable computing, but cloud security is a major area of concern. Many organizations are therefore considering using a combination of a secure internal cloud, along with (what they perceive to be) less secure public clouds. However, this raises the issue of how to partition applications across a set of clouds, while meeting security requirements. Currently, this is usually done on an ad-hoc basis, which is potentially error-prone, or for simplicity the whole application is deployed on a single cloud, so removing the possible performance and availability benefits of exploiting multiple clouds within a single application. This paper describes an alternative to ad-hoc approaches – a method that determines all ways in which applications structured as workflows can be partitioned over the set of available clouds such that security requirements are met. The approach is based on a Multi-Level Security model that extends Bell-LaPadula to encompass cloud computing. This includes introducing workflow transformations that are needed where data is communicated between clouds. In specific cases these transformations can result in security breaches, but the paper describes how these can be detected.

Once a set of valid options has been generated, a cost model is used to rank them. The method has been implemented in a tool, which is briefly described in the paper.

About the authors

Paul Watson is Professor of Computer Science and Director of the Digital Institute at Newcastle University where he leads SiDE - a centre of excellence in Social Inclusion through the Digital Economy (www.side.ac.uk). He graduated in 1983 with a BSc in Computer Engineering from Manchester University, followed by a PhD in 1986. In the 80s, as a Lecturer at Manchester University, he was a designer of the Alvey Flagship and Esprit EDS systems. From 1990-5 he worked for ICL High Performance Systems as a system designer of the Goldrush MegaServer parallel database server, which was released as a product in 1994. In 1995 he moved to Newcastle University where his research focusses on information management and cloud computing.

Suggested keywords

CLOUD COMPUTING

WORKFLOW

SECURITY

A Multi-Level Security Model for Partitioning Workflows over Federated Clouds

Paul Watson
School of Computing Science
Newcastle University
Newcastle-upon-Tyne, NE1 7RU, UK
Email: Paul.Watson@ncl.ac.uk

Abstract—Cloud computing has the potential to provide low-cost, scalable computing, but cloud security is a major area of concern. Many organizations are therefore considering using a combination of a secure internal cloud, along with (what they perceive to be) less secure public clouds. However, this raises the issue of how to partition applications across a set of clouds, while meeting security requirements. Currently, this is usually done on an ad-hoc basis, which is potentially error-prone, or for simplicity the whole application is deployed on a single cloud, so removing the possible performance and availability benefits of exploiting multiple clouds within a single application. This paper describes an alternative to ad-hoc approaches – a method that determines all ways in which applications structured as workflows can be partitioned over the set of available clouds such that security requirements are met. The approach is based on a Multi-Level Security model that extends Bell-LaPadula to encompass cloud computing. This includes introducing workflow transformations that are needed where data is communicated between clouds. In specific cases these transformations can result in security breaches, but the paper describes how these can be detected. Once a set of valid options has been generated, a cost model is used to rank them. The method has been implemented in a tool, which is briefly described in the paper.

I. INTRODUCTION

Cloud computing is of growing interest due to its potential for delivering cheap, scalable storage and processing. However, cloud security is a major area of concern that is restricting its use for certain applications: “Data Confidentiality and Auditability” is cited as one of the top ten obstacles to the adoption of cloud computing in the influential Berkeley report [1]. While security concerns are preventing some organizations from adopting cloud computing at all, others are considering using a combination of a secure internal “private” cloud, along with (what they perceive to be) less secure “public” clouds. Sensitive applications can then be deployed on a private cloud, while those without security concerns can be deployed externally on a public cloud. However, there are problems with this approach. Currently, the allocation of applications to clouds is usually done on an ad-hoc, per-application basis, which is not ideal as it lacks rigour and auditability. Further, decisions are often made at the level of granularity of the whole application, which is allocated entirely to either a public or private cloud based on a judgment of its overall sensitivity. This eliminates the potential benefits for partitioning an application across a set of clouds, while still meeting its overall security requirements.

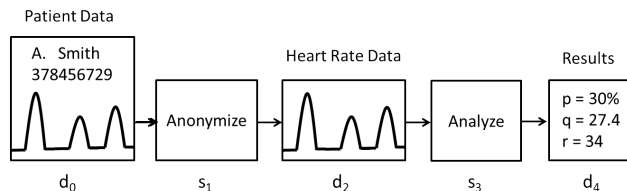


Fig. 1. An example medical data analysis workflow

For example, consider a medical research application in which data from a set of patients’ heart rate monitors is analyzed. A workflow used to analyze the data from each patient is shown in Figure 1. The input data is a file with a header identifying the patient, followed by a set of heart rate measurements recorded over a period of time. A service (*Anonymize*) strips off the header, leaving only the measurements (this application is concerned with the overall results from a cohort of patients, not with individuals). A second service (*Analyze*) then analyzes the measurements, producing a summary.

Analyzing the heart rate data is computationally expensive, and would benefit from the cheap, scalable resources that are available on public clouds. However, most organizations would be unlikely to consider storing medical records on a public cloud for confidentiality and, in some cases, legal reasons. Therefore, one solution is to deploy the whole workflow on a secure private cloud. However, this may overload the finite resources of the private cloud, resulting in poor performance, and potentially a negative impact on other applications.

An alternative solution is to partition the application between the private cloud, and an external public cloud in order to exploit the strengths of both. This could be attempted in an ad-hoc fashion by a security expert but, as we will describe, there are challenges in working out the set of partitioning options that still preserve the required security of data and services. In this paper we therefore describe an alternative to ad-hoc solutions – a method that takes an application consisting of a set of services and data connected in a workflow, and determines the valid set of deployments over a set of clouds, ensuring that security requirements are met. Although the paper is focused on workflows in which services communicate through passing data, the method can be applied

to other types of distributed system that are composed of a set of communicating components. The method is based on Multi-Level Security models [2], specifically Bell-LaPadula [3]. The result of the method is the complete set of options that meet the organization's security requirements for the application. The method introduces transformations that need to be performed on the workflows where data is communicated between clouds; the paper identifies the security issues that can be raised as a result, and the extra security checks that need to be performed to address this. When the method results in more than one valid partitioning option, there is the issue of how to choose the best. The paper shows how a cost model can be introduced to rank the valid options; a model based on price is defined, and applied to the running medical workflow example. The full method, including the cost model, has been implemented in a tool that has been built to automate and explore its application.

The paper is structured as follows. Section II gives a brief introduction to Multi-Level Security models and Bell-LaPadula. It then describes how the Bell-LaPadula rules can be applied to ensure that a workflow meets the security requirements of its constituent services and data. The method is then extended to cloud computing by assigning security levels to clouds, and building on Bell-LaPadula to define a method for determining if security requirements are met in a particular deployment of the constituent parts of a workflow onto a set of clouds. Section III then defines a method for enumerating all valid options for deploying a workflow over a set of clouds so as to meet security requirements. It highlights the issues raised when data must flow between clouds, and shows the workflow transformations and security checks that must be included in the method if security to be guaranteed. The result is a set of valid options; Section IV then introduces a cost model that can be used to select the best option. Following a review of related work (Section V) we draw conclusions and outline further work.

II. METHOD

This section describes how the Bell-LaPadula security model can be applied to workflows, and can then be extended to the deployment of workflows on clouds. Through this section, a workflow is modeled as a directed graph in which services and data are represented as nodes. Services consume zero or more data items and generate one or more data items; the edges in the graph represent the data dependencies.

A. Representing Security Requirements

The Bell-LaPadula multi-level access control model [3] is adopted, with services modeled as the subjects (S), and data as the objects (O) [4]. The security model therefore consists of the following:

- a set of actions (A) that subjects (S) can carry out on objects (O). In the case of services operating on data in a workflow, the actions are limited to read and write. Therefore, the set of actions (A) is: $A = \{r, w\}$
- a poset of security levels: L

- a permissions matrix: $M : S \times O \rightarrow A$ (the contents of the matrix are determined by the workflow design; i.e. if service s_1 reads datum d_0 then there will be an entry in the matrix: $s_1 \times d_0 \rightarrow r$; similarly, if service s_1 writes datum d_2 then there will be an entry in the matrix: $s_1 \times d_2 \rightarrow w$)
- an access matrix: $B : S \times O \rightarrow A$ (this is determined by the execution of the workflow: if there are no choice points then it will equal the permissions matrix, however, if there are choice points then it will equal a subset of the permissions matrix corresponding to the path taken through the workflow when it is executed.
- a clearance map: $C : S \rightarrow L$
- a location map: $l : S+O \rightarrow L$ (this represents the security level of each service and datum in the workflow)

In a typical Multi-Level Security scenario, the system moves through a set of states, and the model can have different values for permissions, access, clearance and location in each state. However, here the execution of a workflow is modelled as taking place within a single state. Normally a service would be expected to have a clearance that is constant across all uses of that service in workflows, however the location can be chosen specifically for each workflow, or even (though less likely) for each invocation of a workflow. However, the model itself is general, and makes no assumptions on this.

The Bell-LaPadula model states that a system is secure with respect to the above model if the following conditions are satisfied \forall subjects $u \in S$ and \forall objects $i \in O$

$$\text{authorization: } B_{ui} \subseteq M_{ui} \quad (1)$$

$$\text{clearance: } l(u) \leq c(u) \quad (2)$$

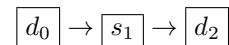
$$\text{no-read-up: } r \in B_{ui} \Rightarrow c(u) \geq l(i) \quad (3)$$

$$\text{no-write-down: } w \in B_{ui} \Rightarrow l(u) \leq l(i) \quad (4)$$

For workflows, the implications of these conditions are:

- (1) all actions carried out by services must conform to the permissions granted to those services
- (2) a service can only operate at a security level (location) that is less than or equal to its clearance
- (3) a service cannot read data that is at a higher security level than its own clearance
- (4) a service cannot write data to a lower security level than its own location.

For example, consider a service s_1 which consumes datum d_0 and produces datum d_2 :



(in these diagrams, the \rightarrow is used to show data dependency, and each block – service or datum – is uniquely identified by the subscript). The following rules must be met:

by (3)

$$c(s_1) \geq l(d_0) \quad (5)$$

and by (4)

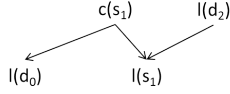


Fig. 2. The relationship between security levels for a service that consumes and produces data

$$l(d_2) \geq l(s_1) \quad (6)$$

The relationship between security levels is captured in Figure 2. Arrows represent \geq relationships.

Whilst assigning a security level to a datum in a workflow is directly analogous to assigning a level to an object (e.g. a document) in the standard Bell-LaPadula model, assigning a security level to a service may be less intuitive. The justification is that an organization may have differing levels of confidence in the set of services they wish to use. For example, they may be very confident that a service written in-house, or provided by a trusted supplier, will not reveal the data it consumes and produces to a third party either deliberately or through poor design; in contrast, there is a risk that a service downloaded from the Internet, of unknown provenance, may do just that. Therefore, the organization can assign a high security level to the former service, and a low level to the latter.

For a specific workflow, when an organization's security experts are assigning locations to services, they may in some cases choose to set the location below that of the clearance level in order to allow a service to create data that is at a lower level than its clearance level; i.e. so that the no write down rule (4) is not violated. This may, for example, take place when the expert knows that the output data will not be sensitive, given the specific data that the service will consume as input in this specific workflow. A concrete example would be a service that summarizes textual data. This has been written to a high standard, and the security expert is confident that it will not leak data to a third party. Therefore, its clearance is high. However, in one particular workflow it is known that this service will only be used to summarise public data downloaded from the World Wide Web, which is also where its output will be published. Therefore, the security expert would set the service's location to an appropriately low level so that the write down rule was not violated.

B. Cloud Security

This section describes how the Bell-LaPadula model, as applied to workflows, can be extended to encompass cloud computing.

Let us say that an organization wishes to run a particular workflow. As more than one cloud is available, a decision must be made as to where the data and services should be placed. In current practice, it is typical that a security expert or system administrator would just take a considered view on the overall security level of the workflow, and that of the clouds on which it could be deployed. For example, let us say that there are

two clouds, one a highly trusted private cloud contained within the intranet of the organization, and the other a less trusted public cloud. It may seem obvious in this case that a workflow that operates on sensitive medical data should run only on the internal cloud. Similarly, a workflow that summarises public data could be deployed on the public cloud. However, there are two problems with this approach. Firstly, it is informal, being based on an expert's judgment; a systematic approach is preferable as it will give more consistent, defensible results. Secondly, the approach deploys the whole of a workflow on a single cloud. This rules out other options that may:

- reduce cost: for example by running less sensitive, but computationally intensive, sub-parts of the workflow on a public cloud if that avoids the need to purchase expensive new servers so that the internal cloud can handle the extra load
- increase reliability: for example by having the option to run on a public cloud if the private cloud has an outage
- increase performance: for example by taking advantage of the greater processing capacity of the public cloud for the computationally intensive services in a workflow

Therefore, the rest of this section extends the security model introduced earlier in order to allow systematic decisions to be taken on where the services and data within a workflow may be deployed to ensure security requirements are met.

To do this, the location map is extended to include clouds which we denote by P (to avoid confusion with the C conventionally used to denote the clearance map):

- location map: $l : S + O + P \rightarrow L$

Also, H is added to represent the mapping from each service and datum to a cloud:

$$H : S + O \rightarrow P$$

We then add a rule that any block (service or datum) must be deployed on a cloud that is at a location that is greater than or equal to that of the block, e.g. for a block x on cloud y :

$$l(p_y) \geq l(b_x) \quad (7)$$

Returning to the example service introduced in the previous section:

$$\boxed{d_0} \rightarrow \boxed{s_1} \rightarrow \boxed{d_2}$$

if, in H , d_0 is on cloud p_a , s_1 on p_b and d_2 on cloud p_c then the following must be true:

$$l(p_a) \geq l(d_0) \quad (8)$$

$$l(p_b) \geq l(s_1) \quad (9)$$

$$l(p_c) \geq l(d_2) \quad (10)$$

This allows us to extend (6) to:

$$l(p_c) \geq l(d_2) \geq l(s_1) \quad (11)$$

The complete relationship between security levels for blocks and clouds is captured in Figure 3.

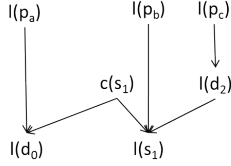


Fig. 3. The relationship between cloud and workflow block security levels

Location (l)	Clearance (c)
d_0	1
s_1	0
d_2	0
s_3	0
d_4	0
c_0	0
c_1	1

TABLE I

LOCATIONS AND CLEARANCES FOR THE MEDICAL ANALYSIS EXAMPLE

III. CALCULATING VALID DEPLOYMENT OPTIONS

Using the above model and rules, it is now possible to automatically enumerate all the valid deployment options for a workflow. These are generated in two stages. Firstly, given the following:

- the set of clouds P
- the set of services S
- the set of data O
- the map of security locations l

we can define the valid mappings of services and data onto clouds, using rule (7):

$$V : S + O \rightarrow P$$

$$V = \{b \rightarrow p | b \in S + O, p \in P, l(b) \leq l(p)\}$$

To illustrate this, we use the medical workflow of Figure 1, with two clouds. This has two services connected in a pipeline, each with one datum as input and one as output:

$$\boxed{d_0} \rightarrow \boxed{s_1} \rightarrow \boxed{d_2} \rightarrow \boxed{s_3} \rightarrow \boxed{d_4}$$

Table I shows an example location and clearance table (while the scheme is general, this example uses only two security levels: 0 and 1). Here, c_1 is a private cloud, which is at a higher security level than the public cloud c_0 . The patient data (d_0) is at the highest security level, while the other data is at the lower level as it is not confidential. Service s_1 is cleared to access confidential data at level 1, but its location has been set to 0 in this workflow so that it can produce non-confidential output at level 0 without violating the Bell-LaPadula “no-write-down” rule (4).

Based on this mapping of blocks and clouds to locations, Table II then shows the possible valid placement of each block onto the two clouds.

Block	Cloud c_0	Cloud c_1
d_0		•
s_1	•	•
d_2	•	•
s_3	•	•
d_4	•	•

TABLE II

VALID MAPPINGS OF BLOCKS TO CLOUDS

Having determined all valid mappings of services and data to clouds, the set of all valid workflow deployments is given by:

$$W : (S + O \rightarrow P) \rightarrow \{(S + O \rightarrow P)\}$$

$$= \{w \in ||V||, \forall b \in S + O. \exists p \in P. b \rightarrow p \in w, |w| = |S + O|\}$$

Where $||V||$ is the power set of V and $|w|$ is the cardinality of w . Algorithmically, in the implementation of the method, W is computed by forming the cross-product of the block-to-cloud mappings contained in V .

All possible valid workflow deployments – as defined by W – for the running example are shown in Figure 4. The cloud on which a datum or service is deployed is indicated as a superscript; e.g. d_j^a is datum j deployed on cloud a .

A. Transferring Data between Clouds

There is still an important issue to be addressed: the approach makes assumptions that are unrealistic for a practical distributed workflow system. It assumes that:

- 1) a service can generate as its output a datum directly on another cloud, without that item being first stored on the same cloud as the service
- 2) a service can consume as its input a datum directly from another cloud, without that item ever being stored on the same cloud as the service

This problem is solved in two stages. Firstly, a new type of service is introduced – *sxfer* – which will transfer data from one cloud to another (this is analogous to the exchange operator used in distributed query processing [5]). It would be implemented with sub-components running on the source and destination clouds. The *sxfer* service takes a datum on one cloud and creates a copy on another. All the workflows generated by W are then transformed to insert the transfer nodes whenever there is a inter-cloud edge in the workflow graph. There are four rules for transforming the graph:

$$\boxed{d_j^a} \rightarrow \boxed{s_i^a} \Rightarrow \boxed{d_j^a} \rightarrow \boxed{s_i^a} \quad (12)$$

$$\boxed{d_j^a} \rightarrow \boxed{s_i^b} \Rightarrow \boxed{d_j^a} \rightarrow \boxed{sxfer} \rightarrow \boxed{d_j^b} \rightarrow \boxed{s_i^b} \quad (13)$$

$$\boxed{s_i^a} \rightarrow \boxed{d_j^a} \Rightarrow \boxed{s_i^a} \rightarrow \boxed{d_j^a} \quad (14)$$

$$\boxed{s_i^a} \rightarrow \boxed{d_j^b} \Rightarrow \boxed{s_i^a} \rightarrow \boxed{d_j^a} \rightarrow \boxed{sxfer} \rightarrow \boxed{d_j^b} \quad (15)$$

Transforms (12) and (14) reflect the fact that if both nodes are deployed on the same cloud then no change is needed. In

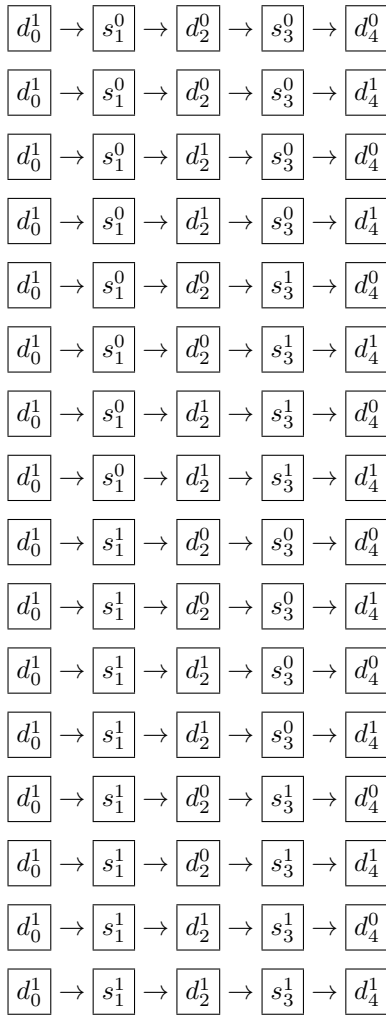


Fig. 4. All valid workflows produced by mapping blocks to clouds

contrast, (13) and (15) introduce new $\boxed{\text{sxfer}}$ nodes to transfer data between clouds.

Unfortunately, the creation of new copies of data through rules (13) and (15) introduces potential security problems. When rule (13) is applied, there is the need to check that cloud b has a sufficiently high security level to store the copy of d_j that would be created on it (the copy inherits the security level of the original). The following rule must therefore be checked to ensure this is true:

$$l(p_b) \geq l(d_j) \quad (16)$$

Similarly, for rule 15:

$$l(p_a) \geq l(d_j) \quad (17)$$

If either is violated then the workflow does not meet the security requirements, and so should be removed from the set W of valid mappings of services and data to clouds. Proof that this violation can only occur in two specific cases now follows.

Firstly, consider (16). By rule (2) we have:

$$c(s_i^b) \geq l(s_i^b) \quad (18)$$

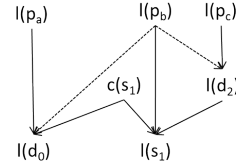


Fig. 5. The relationship between security levels after transformation for inter-cloud data transfer

First consider the case where:

$$c(s_i^b) = l(s_i^b) \quad (19)$$

i.e. the clearance of the object is equal to its location.

Rules (3) and (4) give

$$c(s_i^b) \geq l(d_j) \quad (20)$$

and

$$l(p_b) \geq l(s_i^b) \quad (21)$$

then, by (19)

$$l(p_b) \geq l(s_i^b) \geq l(d_j) \Rightarrow l(p_b) \geq l(d_j) \quad (22)$$

and rule (16) is satisfied. Therefore, in this case there are no violations.

However, if:

$$c(s_i^b) > l(s_i^b) \quad (23)$$

i.e. the clearance of the service is strictly greater than its location then combining (23) with (3) and (4) in a similar way to the above, we get:

$$l(s_i^b) < c(s_i^b) \geq l(d_j) \quad (24)$$

and

$$l(p^b) \leq l(s_i^b) < c(s_i^b) \quad (25)$$

so it is possible that

$$l(p^b) < l(d_j) \quad (26)$$

in which case rule (16) is violated and so that particular workflow deployment does not meet the security requirements.

Turning now to the data produced by services, rule (17) can be violated by transformation (15) in the case where the service s_0 writes up data (4) to a level such that:

$$l(p_a) < l(d_j)$$

The effect of the transformation rules is to modify the security lattice of Figure 2 to that of Figure 5. The arc from $l(p_b)$ to $l(d_0)$ is introduced by rule (13) which adds a copy of d_0 into the workflow, while the arc from $l(p_b)$ to $l(d_2)$ is introduced by rule (15) which adds a copy of d_2 .

Applying the transformations to each workflow in Figure 4, followed by rules (16) and (17) removes half of the possible

deployment options. Removing two duplicates created by the transformations leaves the six valid options shown in Figure 6. Another view of the remaining options is shown in Figure 7; to illustrate the security levels, clouds, data and services at level 0 are shown in red, while those at level 1 are shown in yellow. These diagrams were generated automatically by the tool we have built to implement the methods described in this paper. The aim is to provide a security expert with an easy to understand view of the possible options. Whilst a simple, linear workflow has been used here to illustrate the method, it is applicable to all workflows that can be represented by a directed graph, whatever their structure.

This does however still leave open the issue of how to choose between these valid options? The next section therefore describes how a cost model (also implemented in the tool) can be used to select the best option based on the charges made by the cloud providers.

IV. SELECTING A DEPLOYMENT OPTION WITH A COST MODEL

Once all valid options for allocating services and data to clouds have been determined, one must be selected, and used to enact the workflow. This decision could be made by a deployment expert, but this section describes how it can be achieved automatically through the use of a cost model. Different criteria may be important for different applications (e.g. dependability, performance), but this section illustrates the approach by describing a model that minimizes price.

Cloud pricing is measured using the metrics by which cloud providers allocate charges. For a cloud (p) this is represented as:

- volume of data transferred into a cloud: e_{dxi}^p
- volume of data transferred out of a cloud: e_{dxo}^p
- volume of data stored, per unit of time for which it is stored: e_{ds}^p
- time units of cpu consumed in the execution of a service: e_{cpu}^p

Cost metrics are characterised for a datum (d) as:

- data size: $size(d)$
- data longevity – the length of time the datum is stored: $longevity(d)$

Finally, the cost metric for a service (s) is characterised as:

- time units of cpu consumed in the execution of a service: $cpu(s)$

The cost model for a workflow execution can then be defined as:

$$\text{cost} = \sum_{d=0}^{d=k-1} e_{ds}^p \cdot \text{size}(d) \cdot \text{longevity}(d) + \sum_{s=0}^{s=m-1} e_{cpu}^p \cdot \text{cpu}(s) + \sum_{x=0}^{x=q-1} (e_{dxo}^{ps} + e_{dxi}^{pd}) \cdot \text{size}(d)$$

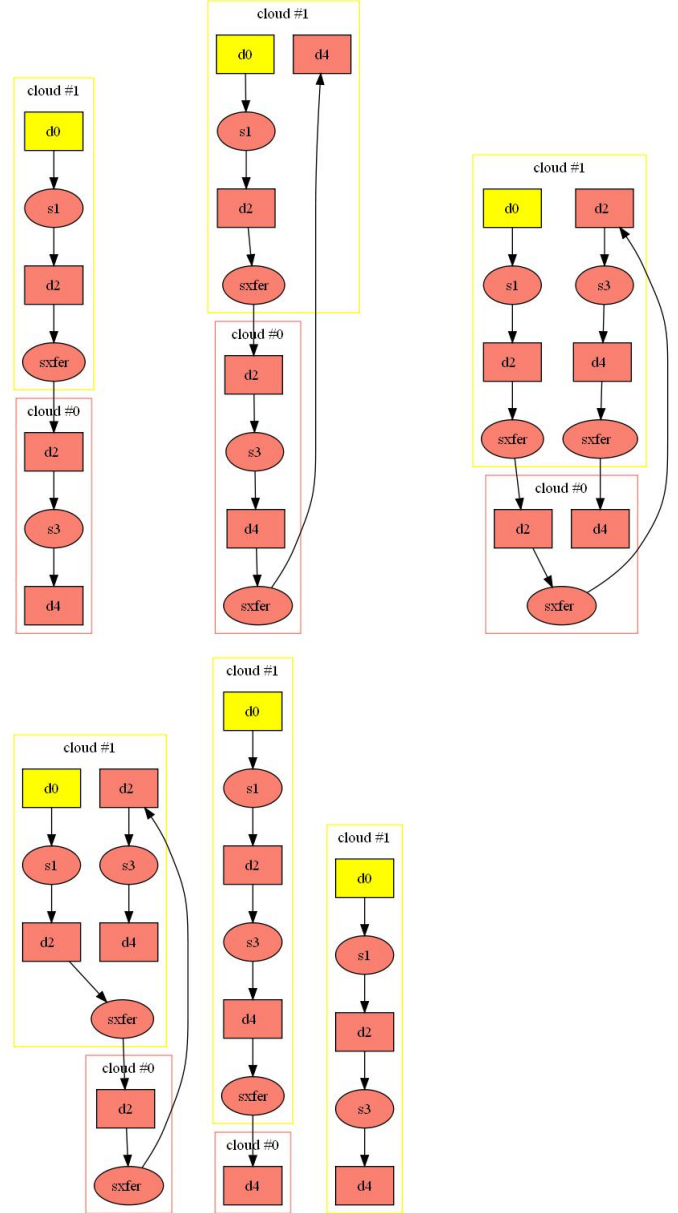


Fig. 7. The six valid cloud mappings

where k is the number of data items in the workflow, and m is the number of services, while q is the number of inter-cloud data transfers. In the third term that calculates data transfer costs, ps represents the source cloud and pd the destination cloud for the transfer.

Using the cost model requires estimates of data sizes and cpu costs. This is realistic for many workflows, and producing these estimates is made easier if performance and capacity are logged for each run, so allowing statistical analysis to generate predictions. This is, for example, done by the e-Science Central cloud platform [6] which logs data on all data sizes, and service execution times.

Two examples now highlight the use of the model. Consider

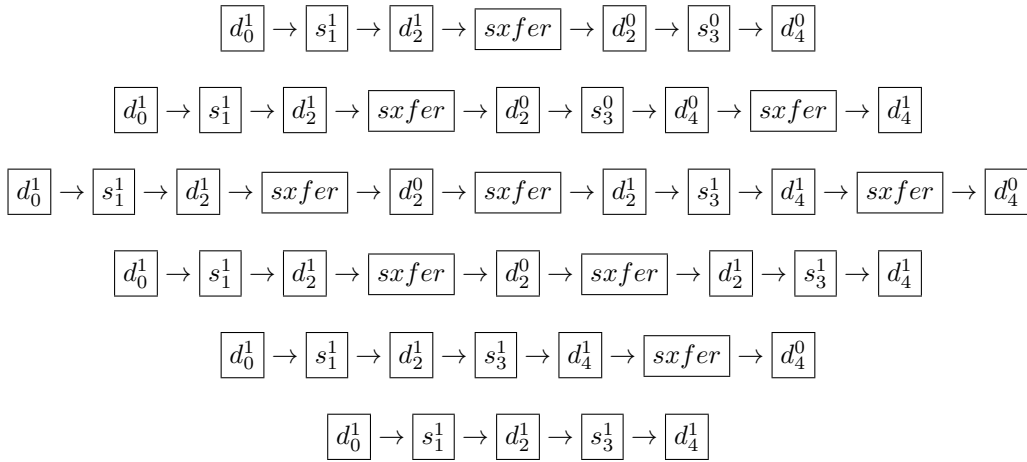


Fig. 6. The Workflows that remain valid after Transfer Blocks are Added

Cloud	Storage (GB / Month)	Transfer In (/GB)	Transfer Out (/GB)	CPU (s)
c_0	10	10	10	10
c_1	10	10	10	10

TABLE III
CLOUD COSTS: EXAMPLE 1

Block	Size (GB)	Longevity (months)	CPU (s)
d_0	10	12	
s_1			100
d_2	5	0	
s_3			50
d_4	1	12	

TABLE IV
BLOCK INFO

Option	Storage	Transfer	CPU	Total	Order
1	1320	100	1500	2920	3
2	1320	120	1500	2940	4
3	1320	220	1500	3040	6
4	1320	200	1500	3020	5
5	1320	20	1500	2840	2
6	1320	0	1500	2820	1

TABLE V
WORKFLOW DEPLOYMENT OPTIONS COSTS: EXAMPLE 1

Cloud	Storage (GB / Month)	Transfer In (/GB)	Transfer Out (/GB)	CPU (s)
c_0	5	5	5	5
c_1	10	5	5	10

TABLE VI
CLOUD COSTS: EXAMPLE 2

the valid mapping options shown in Figure 7 for the running example. In the simplest case, if the performance and cost of both clouds are equal (as in Table III), then the cost difference between options is dependent only on the number of inter-cloud communications. Table IV gives example values for the blocks in the workflow. The size of d_0 will be known as it is the input to the workflow, while that for d_2 and d_4 are estimates, perhaps based on the results of previous runs. To set the longevity values, it is assumed that the input (d_0) and output data (d_4) is stored for a year, while intermediate data (d_2) is immediately discarded once it has been consumed by a service (s_2).

Table V shows the results when the cost model is applied. Each row represents the cost of an option in Figure 6. The final column of the table gives the order of the options (from lowest to highest cost). This confirms that the cheapest is option 6, in which all the blocks are deployed on the same cloud, and so there are no inter-cloud transfer costs.

While it may seem that an option in which all services and data are deployed on a single cloud will always be the cheapest, if CPU costs vary between clouds, then inter-cloud

transfers may be worthwhile. Table VI shows clouds with a different set of cost parameters. Here, a private cloud (c_1) has higher security, but higher CPU and data costs, compared to a public cloud (c_0). The effect of plugging these values into the cost model is shown in Table VII. The result is that the best option is now the one that allocates as much work as possible to the public cloud, which has lower CPU costs.

V. RELATED WORK

The motivation for this paper came from the author's experience of cloud applications with security constraints (e.g.

Option	Storage	Transfer	CPU	Total	Order
1	1260	75	1250	2585	1
2	1320	90	1250	2660	2
3	1260	165	1500	2925	5
4	1320	150	1500	2970	6
5	1260	15	1500	2775	3
6	1320	0	1500	2820	4

TABLE VII
WORKFLOW DEPLOYMENT OPTIONS COSTS: EXAMPLE 2

healthcare applications in the SiDE project [7]). However, the general concern that security was a barrier to use of the cloud for many organizations and applications has been widely discussed [1]. The general issues associated with security and clouds are discussed in [8]. A high-level approach to deciding where an application could be deployed is discussed in [9]. Another approach to eliciting and exploiting information on the security and other properties of clouds is described in [10]. These methodologies could be valuable in assigning security levels to clouds, services and data: something which is orthogonal to the scheme described in this paper.

In [4], Bell-LaPadula is also applied to workflow security. Petri Nets are used to model the workflow, rather than the rule-based approach taken in this paper. However, its scope does not extend to considering the deployment of blocks within a workflow across a set of computational resources, as this paper does.

There has been a large body of work on using cost models to predict execution times in order to select between options for deploying workflows over grids and clouds [11] [12]. However, perhaps due to the relatively recent introduction of pay-as-you-go cloud computing, there is much less work on using price-based cost models. In [13], both execution time and price-based models are used to compare a set of options for allocating a workflow over local resources and a public cloud. The work in [14] uses non-linear programming to generate options for using clouds to execute a workflow. Security is not a consideration in any of these papers. Once the partitioning of a workflow over a set of clouds has been decided, a distributed workflow enactment engine is needed to actually run the workflow. The issues around this are discussed in [15] and a solution is proposed.

VI. CONCLUSIONS

This paper has described a new method for automatically determining valid options for partitioning a workflow of services and data across a set of clouds based on security requirements. A cost model is then used to choose between the available options. The main contribution is to show how multi-level security, which has previously been applied to workflows, can be extended to encompass the allocation of workflow services and data to clouds. This has demonstrated that the need for inter-cloud data transfers raises interesting potential security violations that need to be addressed; in the running example, this ruled out over half of the possible partitioning options. Although the paper focuses on workflows, the method can be applied to other distributed systems whose components are to be partitioned over a set of clouds.

A tool has been developed that implements the methods shown in this paper; all the results for the running example were produced by the tool. It is implemented in Haskell [16], with the workflows represented as directed graphs, which are processed in two ways. Rules (e.g. basic Bell-LaPadula and the extension to clouds) are implemented as filter functions which remove invalid workflows from the set of options. Transformations (e.g. for inter-cloud transfers) take workflows

as input and generate the modified workflows. The tool also automatically generates the diagrams that are shown in Figure 7 using the GraphViz software library [17]; these visualisations have proved to be a useful way to review the available options.

The approach is currently being extended to encompass dependability requirements. Even in its current form, the method can illuminate dependability issues; for example, analysing the set of valid options can highlight the level of dependency of the workflow on particular clouds. If (as in the running example) all valid partitioning options depend on the availability of a specific cloud, then an organization which is dependent on the workflow should ensure that this cloud has sufficiently high levels of availability, or identify a second cloud with a sufficiently high security clearance that could also be used by the workflow.

Overall, our hope is that the approach described in this paper can move the process of partitioning workflows over federated clouds from one in which a human administrator makes an informed but ad-hoc choice, to one in which a tool, such as the one built to implement this method, can determine the valid options based on a rigorous underlying set of rules, and then suggest which is the best, based on a cost model. The approach therefore has the advantage that it can reduce both security violations and execution costs.

ACKNOWLEDGMENT

The author would like to thank Leo Freitas, John Mace, Paolo Missier, Chunyan Mu, Sophie Watson, Feng Hao, Simon Woodman and Hugo Hiden for their comments and suggestions. This work was funded by the Research Councils UK Social Inclusion through the Digital Economy project EP/G066019/1.

REFERENCES

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, and Others, "Above the Clouds: A Berkeley View of Cloud Computing," University of California at Berkeley, Technical Report UCB/EECS-2009-08, 2009.
- [2] C. E. Landwehr, "Formal Models for Computer Security," *ACM Computing Surveys*, vol. 13, no. 3, pp. 247–278, Sep. 1981.
- [3] D. Bell and L. LaPadula, "Secure computer systems: Mathematical foundations," MITRE CORP BEDFORD MA, Tech. Rep., 1973.
- [4] K. Knorr, "Multilevel security and information flow in Petri net workflows," in *Proceedings of the 11th Conference on Advanced Information Systems Engineering*. Citeseer, 2001.
- [5] G. Graefe, "Encapsulation of parallelism in the Volcano query processing system," *ACM SIGMOD Record*, vol. 19, no. 2, pp. 102–111, 1990.
- [6] P. Watson, H. Hiden, and S. Woodman, "e-Science Central for CARMEN: science as a service," *Concurrency and Computation: Practice and Experience*, vol. 22, no. 17, pp. 2369–2380, 2008.
- [7] SIDE Project, "Social Inclusion through the Digital Economy," 2011. [Online]. Available: www.side.ac.uk
- [8] J. Mace, A. van Moorsel, and P. Watson, "The Case for Dynamic Security Solutions in Public Cloud Workflow Deployments," in *The First International Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments (DCDV 2011)*, Hong Kong, China, 2011.
- [9] L. N. (Capgemini), "Putting Cloud security in perspective," Capgemini Tech. Rep., 2010.
- [10] S. Pearson and T. Sander, "A mechanism for policy-driven selection of service providers in SOA and cloud environments," in *New Technologies of Distributed Systems (NOTERE), 2010 10th Annual International Conference on*, vol. 79. IEEE, 2010, pp. 333–338.

- [11] R. Buyya, "Cost-Based Scheduling of Scientific Workflow Application on Utility Grids," in *First International Conference on e-Science and Grid Computing (e-Science'05)*. Ieee, 2005, pp. 140–147.
- [12] G. Singh, C. Kesselman, and E. Deelman, "A provisioning model and its comparison with best-effort for performance-cost optimization in grids," in *Proceedings of the 16th international symposium on High performance distributed computing - HPDC '07*. New York, New York, USA: ACM Press, 2007, pp. 117 – 126.
- [13] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: The Montage example," *2008 SC - International Conference for High Performance Computing, Networking, Storage and Analysis*, no. November, pp. 1–12, Nov. 2008.
- [14] S. Pandey, K. Gupta, A. Barker, and R. Buyya, "Minimizing Cost when Using Globally Distributed Cloud Services: A Case Study in Analysis of Intrusion Detection Workflow Application," Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, Melbourne, Australia, Tech. Rep., 2009.
- [15] S. Woodman, "A Programming System for Process Coordination in Virtual Organisations," Ph.D. dissertation, Newcastle University, 2008.
- [16] S. P. Jones, L. Augustsson, D. Barton, B. Boutel, W. Burton, J. Fasel, K. Hammond, R. Hinze, P. Hudak, J. Hughes, T. Johnsson, M. Jones, J. Launchbury, E. Meijer, J. Peterson, A. Reid, C. Runciman, and P. Wadler, "The Haskell 98 Report," 2002.
- [17] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, "Graphviz - Open Source Graph Drawing Tools," *Graph Drawing*, vol. 2265, pp. 483–484, 2001.