Governance Model for Cloud Computing in Building Information Management

Thomas Beach, *Member, IEEE*, Omer Rana, *Member, IEEE*, Yacine Rezgui, *Member, IEEE*, and Manish Parashar, *Member, IEEE*

Abstract—The AEC (Architecture Engineering and Construction) sector is a highly fragmented, data intensive, project based industry, involving a number or very different professions and organisations. The industry's strong data sharing and processing requirements means that the management of building data is complex and challenging. We present a data sharing capability utilising Cloud Computing, with two key contributions: 1) a governance model for building data, based on extensive research and industry consultation. This governance model describes how individual data artefacts within a building information model relate to each other and how access to this data is controlled; 2) a prototype implementation of this governance model, utilising the CometCloud autonomic cloud computing engine, using the Master/Work paradigm. This prototype is able to successfully store and manage building data, provide security based on a defined policy language and demonstrate scale-out in case of increasing demand or node failure. Our prototype is evaluated both qualitatively and quantitatively. To enable this evaluation we have integrated our prototype with the 3D modelling software—Google Sketchup. We also evaluate the prototype's performance when scaling to utilise additional nodes in the Cloud and to determine its performance in case of node failures.

Index Terms—Cloud computing, building information modelling, data management, distributed tuple space

1 Introduction

We present an information governance and access control model that enables data generated during the lifecycle of a project within the AEC (Architecture Engineering and Construction) sector to be securely shared via a cloud computing system and discuss our prototype implementation of this governance model using the CometCloud infrastructure described in [25]. Our focus is on a domain specific information governance model and we envision the use of both public and private cloud deployments [27], where private clouds would be used to execute licensed, proprietary software, and store and manage project-specific data within data structures specific to their domain. Public Clouds could be used to support multiple versions of a software program which can be deployed and distributed as separate cloud applications. For example, an energy simulation system can include daylight simulation, HVAC (heating, ventilation, and air conditioning) system simulation, air flow simulation within a building, etc. Similarly, document management can include the management of contracts, building permits, purchase orders, regulations, etc. [26], [34].

There are two key contributions presented in this paper: 1) an application case study demonstrating how Cloud computing can be used to support collaborative working in the AEC sector, with particular reference to the emerging Building Information Model (BIM) standard and integration with 3D open source tools for rendering such data—such as Google sketchup. We show, in particular, how a Cloud-based infrastructure can be used to facilitate coordination between different sites and roles involved in an AEC project; 2) a Cloud prototype implemented using the CometCloud system, which enables bridging between in-house computational infrastructure at participating sites involved in the AEC project and public Cloud systems, such as Amazon EC2/S3. A key novelty of this approach is the ability to use a Cloud system that uses the Linda-like tuple space model for developing a distributed, shared "space" for exchanging data between participants. Access control over data maintained in this shared space is described, based on policy descriptions that are interpreted by specialist gateway nodes. We demonstrate, in particular, how our approach to access control makes use of the BIM standard. We describe an architecture of our system and report on performance results when using an implementation of this. Currently, the entire AEC industry is moving towards adoption of collaborative approaches for information management (for instance, due to the UK government's requirement that building information be shared in a fully collaborative manner for all public projects by 2016 [14]), thereby making the approach proposed in this work of interest to a wide part of this industry sector.

1.1 Context

Information governance generally consists of policies, procedures and controls required to manage access to information at an enterprise level. In the context of this work, a governance model encompasses many aspects of access control, security and policy-based management of

- *T. Beach, O. Rana, and Y. Rezgui are with Cardiff University, U.K. E-mail: T.H.Beach@cs.cardiff.ac.uk.*
- *M. Parashar is with Rutgers University, USA. E-mail: parashar@rutgers.edu.*

Manuscript received 25 June 2012; revised 30 Mar. 2013; accepted 24 Oct. 2013. Date of publication 3 Nov. 2013; date of current version 10 Apr. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TSC.2013.50

data objects generated during an AEC project. Many incidents related to cloud security are often framed in the context of existing Web applications and data-hosting [15]. In industry related incidents, such issues remain well established challenges, e.g. phishing, downtime, data loss, password weakness and compromised hosts running botnets.

Our governance model is based on Role-based Access Control (RBAC) policies, enabling dynamic modification of access rights associated with data objects based on activities and responsibilities (''roles'') within a virtual enterprise, assigned to subjects within the system. Instead of specifying all possible actions each subject is permitted to execute, a security policy in RBAC is specified for roles, while subjects are granted to adopt roles. If a user moves to a new function within the organisation, the user can simply be assigned to a new role. RBAC also reuses role specifications by a form of inheritance whereby one role (often a manager in the organisation) can inherit the privileges of another role. Using this approach also enables us to benefit from the IETF/DMTF ''Policy-based Admission Control Framework'' [49], consisting of a Policy Repository, a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP). These architectural components can be mapped into our CometCloud implementation easily.

The problem of data management is currently of critical importance to the AEC sector. Currently, the entire construction industry is moving towards adoption of collaborative approaches for information management, due to the UK government's requirement that building information be shared in a fully collaborative manner for all public projects by 2016 [14]. There are, however, significant problems within the AEC industry that must be overcome in order to support efficient data and information sharing between sites. In a consultation we conducted with over 75 industry participants, the following key technical limitations were identified [53]:

- Fragmentation of data across design and engineering teams and then the contractor and Facilities Management companies.
- Information is not sustained across the lifecycle and is in continuous danger of being lost due to company mergers or bankruptcy.
- Access control to data is addressed by commercial and proprietary solutions that are not compatible and do not embed the project procurement path and process dimension.
- Data is not secure even when stored on servers, as these are managed and controlled by a single company or at best outsourced to a data centre.
- The network and communication overheads in using externally sourced virtualized storage.
- Privacy constraints associated with utilizing such storage for business sensitive data, as viewed by industry end users.
- Lack of support for data integrity, user authentication, data security, and access control.

The development of an information governance model will enable the AEC industry to accelerate its progress towards this ambitious government target and also provide significant efficiency savings in process, as, currently, inadequate information management is one of the key contributing factors to delays and construction waste [18], [19] for companies working on the building project. Although the work presented in this paper is focused on requirements of the AEC industry, it also has significant overlaps with collaborative working in other service-oriented environments. For instance, the document management and multi-user collaboration identified in Section 3 has significant overlaps.

This article is structured as follows: Section 2 provides the background to this work. In Section 3, we describe the overall governance model, focusing on document types, relationships between these documents, the overall process that leads to the generation of these documents and the access rights needed. A conceptual architecture is then presented in Section 4, identifying the key services involved in our system. Section 5 contains a description of the CometCloud implementation with a prototype discussed in Sections 6 and 6.1 along with validation and performance results of use.

2 Background

2.1 Building Information Modelling

Construction and Civil engineering are highly regulated, fragmented, data intensive, project-based industries dependent on a large number of very different professions and firms, with increasing data sharing and processing requirements [14], [31], [41], [42]. The process of designing, re-purposing, constructing and operating a building or facility involves a large number of disciplines including: Architecture, Structure, Mechanical & Electrical, Energy and Sustainability.

Within the industry, project failure has been common over the past decades and, amongst the notable common causes of failure is the lack of effective project team integration across the supply chain [41]. It has been found that inadequate information management is one of the key contributing factors to delays and construction waste [18], [19], [22]. A lack of compatible information systems, defacto standards and protocols, and the varying perspectives and requirements of clients and designers limit effective collaboration [11]. It is therefore necessary to identify capability that would enable:

1. alternative design proposals to be evaluated with comparative ease;
2. design artefacts to be modelled in several dimensions, thus eliminating coordination errors and subsequent expensive changes;
3. enable design data to be fed direct to manufacturing environments eliminating unnecessary intermediaries;
4. provide support for asset management subsequent to construction.

Moreover, data management across the project lifecycle and supply chain lacks an overall data management policy [6], [22]. Various architectural and engineering data sets relating to a particular project are produced and usually stored in [5]: 1) local computers often with limited network connectivity, persistence and availability; 2) independently managed, single company-owned document sever(s) where access is dictated by a company specific policy or by a charging model; and 3) ad-hoc document archives (e.g., Intranet and Web-based document management systems) in the context of a particular project based on ad-hoc access policy associated with the project.

In recent years the data collected for the entire life of a building has become termed a Building Information Model (BIM) and it is now accepted that a BIM can be defined as the process of generating and managing data and information about a building during its entire life cycle from concept design to decommissioning [2], [22]. It has also become accepted that BIM should present the complete model of the building as a ''phaseless'' workflow [37], i.e., it should be a complete 4D virtual repository of all the data about the building from its conception through to its demolition. This means that a BIM does not just include 3D models of the building structure but also: 1) management information including work plans and schedules; 2) product information (or links to product information data) about all items within the building—right down to the level of internal furnishings; and 3) building performance data collected by sensors within an operational building (i.e. heat, CO2 emissions, lighting levels). The UK government has realized the importance of built environment data and its strategic role in meeting some of its most pressing challenges, including the climate change and carbon reduction agendas [11].

2.2 Comet Cloud

We make use of the CometCloud system due to its successful deployment in other data sharing scenarios within the computational finance area [24]. CometCloud uses a Linda-like tuple space referred to as ''CometSpace''—which is implemented using a Peer-2-Peer overlay network. In this way, a virtual shared space for storing data can be implemented by aggregating capability of a number of distributed storage and compute resources. CometCloud therefore provides a scalable backend deployment platform that can combine resources across a number of different providers dynamically—a key requirement for an AEC project. The overarching goal of CometCloud is to realize a virtual computational cloud with resizable computing capability, which integrates local computational environments and public cloud services on-demand, and provides abstractions and mechanisms to support a range of programming paradigms and application requirements. Specifically, CometCloud enables policy-based autonomic *cloudbridging* and *cloudbursting*. Autonomic cloudbridging enables on-the-fly integration of local computational environments (datacenters, Grids) and public cloud services (such as Amazon EC2 and Eucalyptus), and autonomic cloudbursting enables dynamic application scale-out to address dynamic workloads and spikes in demand. Cloudbridging is useful when specialist capability available in-house needs to be integrated with high throughput computation that can be outsourced to an external cloud provider such as Amazon. Cloudbursting, on the other hand, enables scale-out of in-house computation and may not necessarily involve a change in capability between in-house and outsourced providers. CometCloud has been shown to be highly scalable (e.g., integrating over 2500 cores across several HPC systems and Amazon EC2) in different federation scenarios involving High Performance Computing (HPC) and Cloud environments [39].

2.3 BIM Adoption in the AEC Sector

A number of commercial solutions addressing the problem of BIM data storage have been developed. Commercial BIM servers are today available, including the Onuma system (www.onuma.com), Revit Server (www.autodesk.com), ProjectWise and AssetWise (www.bentley.com), Graphisoft BIM Server (www.graphisoft.com) and EDMmodelServer (www.jotne.com). However, these servers often require local infrastructure and maintenance within the organisation that is using them—tending to utilise either central (accessible to all team members over the WAN) or local (accessible to team members over the LAN) connectivity.

Recently software vendors, such as Bentley Systems and Autodesk, have begun offering hosted solutions for building data. Additionally 4Projects (4Projects.com) offer a specific project collaboration service for the AEC sector including document and drawing management, contract management, and procurement management. Many such projects make use of their own proprietary file formats (especially in relation to 3D building models). While import/export functionality to standardised formats exist, there are still issues with data interoperation, i.e. complete mapping between different formats is not possible due to the use of proprietary extensions. In addition to software to support collaboration within the AEC sector, a number of semantic resources and information management standards have been developed for the construction domain. One of the key developments is the Industry Foundation Classes (IFC) [2]. The IFCs are becoming a commonly used form for BIM. They are open data model specifications for defining building components' geometry and other physical properties in a way that enables CAD users to transfer design data between different software applications [43]. They are intended to provide an authoritative semantic definition of building elements, their properties and interrelationships. Data associated with IFC can include: textual data, images (such as building schematics); structured documents, numerical models and designer/project manager annotations [34]. The IFC specification is developed and maintained by BuildingSmart and has been included in several ISO standards. The IFC with its standard set of rules for data storage, data exchange and protocols provides an ideal framework to manage data related to a building throughout its lifecycle [44]. As-well as the IFCs, other developments within the industry include COBie (Construction Operations Building Information Exchange), agcXML, and BS1192:2007. COBie is developed as a standard data specification for structured information exchange [4].

3 BIM Governance Model

As outlined in Section 2 the Building Information Model (BIM) can be viewed as a collection of documents generated during the lifecycle of a particular AEC project. As a project matures, information is progressively added by participants who engage with the project [29]. Such information may consist of a 3D model, or may include a schedule, spreadsheet, database, or text document [30].

The key challenges in this context include the ability to:

1. aggregate documents, related to a particular facet (such as structural analysis) needed at particular stages of a AEC project lifecycle;
2. trace back when and who added a feature to a design;
3. check conformance to the BIM data model;
4. ensure that constraints identified in the design are not subsequently violated in other parts of the project lifecycle.

Collectively, we address these challenges as part of an information governance model—when data is stored over a CometCloud system. The document abstraction is a key concept in this work and represents both previously recorded static data and dynamic output generated through a software package (such as a CAD program). A document may include structured, unstructured and graphical content, simulation output, scheduling and costing information and a project brief. In our governance model a document is treated as a view or a ''lens'' onto BIM data. This idea of documents as ''lenses'' can be illustrated by comparing the information needs of different disciplines within a building project. The architect will require detailed information about all physical aspects of the building, but not access to detailed structural analysis or scheduling information used by the project manager. Conversely, the structural engineer will often require detailed architectural models in order to perform their role, however (s)he may not require this information down to the level of internal furnishings and placement of individual electrical outlets.

Previous research [16], [31] (and industry consultation) has identified the following key areas for inclusion in the governance model:

1. the lifecycle of a AEC project;
2. documents and their relationships,
3. users;
4. the multiple disciplines (as outlined in Section 1) using the documents;
5. access rights to these documents.

Concepts from these five areas are used to define access rights for various individuals involved in a building project through a gateway, which mediates access to a Cloud environment storing documents—explained in more details in Section 4. In Sections 3.1, 3.2, and 3.3 we look at each of these issues in more detail and relate them to our Cloud-based implementation.

3.1 Document Types and Relationships

When using BIM, it is rare for any two documents to be treated as completely separate entities and many documents will have relationships with others. Incremental refinements within our focus groups consultation led to four types of relationships: versioning, composition, derivation and concurrency. Each of which have different implications as to how data within the BIM as a whole is affected when a new document is added. These are discussed below, with the variable B used to represent BIM:

No relationship: A new Document D_a is added to the BIM so that: $B = B + D_a$.

Versioning: A new document D_b is created, based on an existing document D_a with changes made such that $D_b = D_a + X - Y$ where X is the addition and Y the removal of data from D_a.

Derivation: Given that a document D_b for use by discipline j is derived from document D_a which was created for discipline i then $D_b = f_j^i(D_a) + X$ where f_j^i is function to filter data and X is the new data added.

Composition: New data is added to the BIM model forming part of an existing document. For example, if the top level document is D_a and each floor within a building is represented as a document layer $D_{f0} \dots D_{f3}$ then $D_a = D_{f0} + D_{f1} + D_{f2} + D_{f3}$. Each document in the composition may possess different access rights.

Concurrency: This relationship models a situation where two documents are developed in parallel and illustrates a dependency between the two. In this relationship the contents of the documents are linked such that $D_a = X + D_b$ and $D_b = Y + D_a$ where X and Y are the actual contents of D_a and D_b respectively.

These relationships allow us to easily model four of the most common occurrences within a AEC project:

- When a user (e.g., a structural engineer) begins work on their design, they will require some of the information already present in the architects design. This is the *derived from* relationship and it allows a user to create a document that uses some or all of the information from another document.
- The *version of* relationship allows us to model the scenario when changes are made to an existing document leading to the creation of a new version. This allows the modelling of complex document structures that can occur within a construction project, where several parallel versions may be developed for experimentation by an architect before a final version is chosen for full development.
- The *composition* relationship allows the representation of a document as a collection i.e. when each floor of a structure is modelled separately and then aggregated into a single document.
- The *concurrency* relationship allows for situations when two documents are worked on concurrently and changes in one must be reflected in the other. An example includes the structural design of a building where an architect and structural engineer must work closely together to resolve design conflicts.

The use of BIM enables better use of the above relationships, primarily due to the structured (model-based) and standardised nature of the underlying data. Such structure enables a more effective use of the relationships identified above. These relationships also provide the basis for

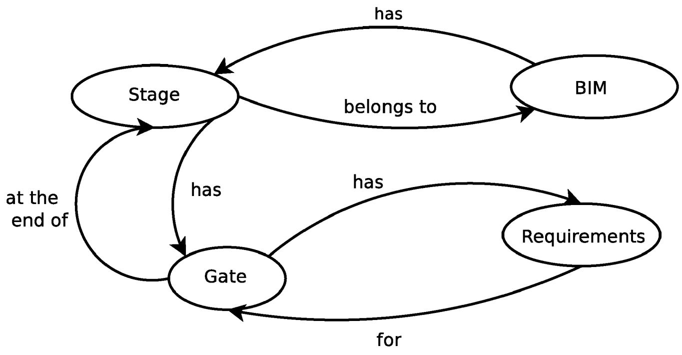

Fig. 1. Structure of a BIM.

allowing collaboration within the system, i.e., a Structural Engineer can add relevant information to a new document based on the work of the architect using the *derived from* relationship. Two users working simultaneously on an existing document (but different parts of the document) can be modeled with the *concurrency* relationship, whereas users working more closely together will be able to construct (for either later merging or for one version to be selected as the final version) multiple parallel versions of a document using the *version of* relationship.

3.2 Process and Access Rights

In our approach we consider BIM to consist of a series of *stages*, at the end of which there is a *gate*, identifying a set of either mandatory or optional requirements such as the presence of data within the BIM, the accuracy of data—as illustrated in Fig. 1. For instance, in early stages of a building lifecycle, energy efficiency figures may be rough estimates which need to be refined in later stages (when results of detailed simulations become available). This division into stages allows the tracking of the BIM model throughout the project. In essence each stage within the model can be viewed, once it has completed, as a snapshot of the BIM as it was at that time. This functionality will enable the use of the governance model as support for project managers, enabling the creation of reports that can be used to facilitate design making and ensuring BIM compliance with standards, whether imposed by legislation, the client, or internally.

3.3 Multi User Collaboration

Fig. 2 describes the various components that make up a BIM document. Currently we define five document types (based on who can access them):

1. private: document only for use of owner;
2. team: document only for use at a team level;
3. finalised: document is finalised for use by other teams;
4. Client—document is ready for release to client;
5. Archived—document has reached end of its lifecycle and no further alterations will be made.

A transaction occurs whenever a relationship between documents is created by a user [16]. The transaction entity is generated automatically, whereas the decision entity enables the user to make explicit the reasons for creating the relationship. Fig. 2 also shows all the metadata that is stored, using as a base-line, the Dublin Core metadata standard. This ensures that all elements defined in this standard are either provided explicitly within the Document object, or implicitly by its relation to data stored in other related objects within the data model.

3.4 Access Rights, Users, Disciplines, and Roles

In order to enforce a fine grained access control over documents, we use the concepts of *users*, *disciplines*, *rights*, and *roles* within the governance model—as shown in Fig. 3:

- *Users*—A user is a single actor within the system.
- *Disciplines*—An industry recognised specialisation working on a specific aspect of a project.

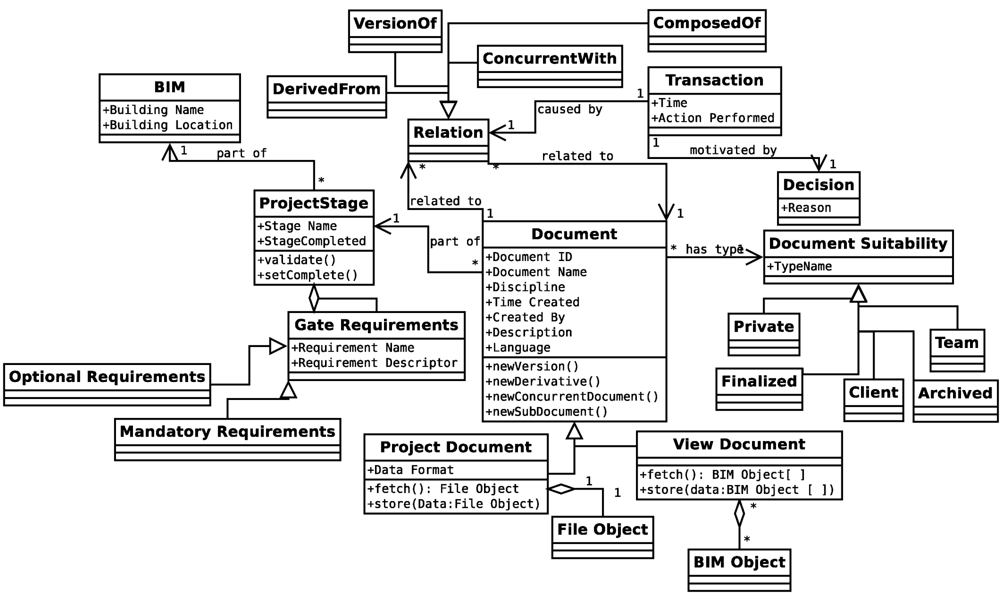

Fig. 2. BIM document.

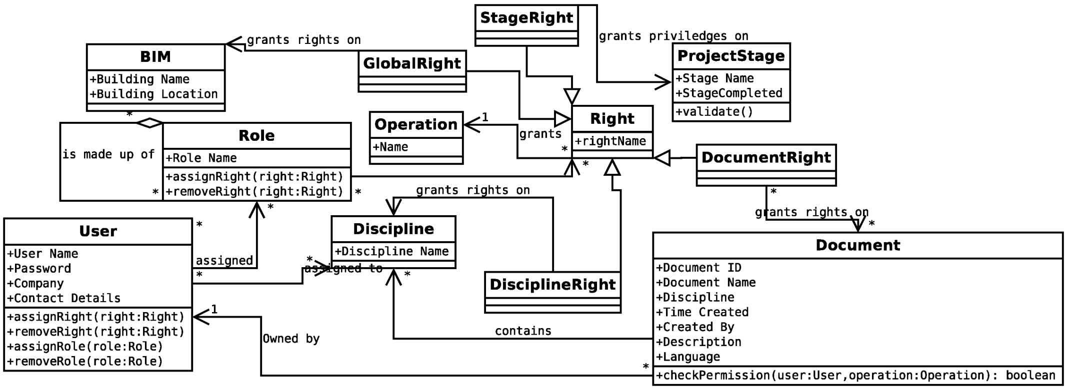

Fig. 3. Implementation of users, disciplines, and roles.

- *Rights*—The conceptualisation of a permission to perform an operation on a document.
- *Roles*—A grouping of rights that can be applied to users or entire disciplines.

A detailed diagram of how these access rights are implemented is shown in Fig. 3. This figure shows that for maximum flexibility a right can not only be applied to an individual document, but also to all documents in a BIM and all documents within a stage of the project. Additionally, the functionality to allow role aggregation is present, allowing roles to be combined.

The access rights identified in Fig. 3 are implemented using a discretionary policy language within our system. Hence, access to particular documents is based on a policy which includes a Subject, a Target, an Action, a Role, a Rule, a Condition and a Variable. A *Subject* refers to an entity which must interact with the cloud system. In a discretionary policy of this kind, each accessible Object has an associated possessor who exercises primary control over the Object. The controls are discretionary in the sense that a Subject with access permission on an object can delegate that permission to any other subject This entity can be one or more human users or another system. Hence, a Subject must have associated permissions to act upon a *Target*—i.e., the entity on which an Action is performed by a Subject. An *Action* is the operation (view, create, delete, or change document) that Subject is authorised to invoke on one or more Target(s). A *Role* is a set of policy Rules that apply to a single Subject domain. Each Role is associated with a set of norms (i.e., permissions, obligations and prohibitions). An object can play several Roles simultaneously and so it inherits the union of the norms associated with each Role it plays. A *Rule* is a statement which combines four concepts of the policy language: Subject, Condition, Action, Target. One or more Rules create a system's Role. *Conditions* are optional, enabling use of both conditional and unconditional norms. For instance, access to a document may be unconditional or based on a time constraint—such as time limited access to a document, permitted only between 08:00 and 20:00. Conditions are therefore restrictions that define if the Subject is permitted to do its desired action. A *Variable* refers to a specific component or feature of the system. For example the bandwidth or throughput can be considered as a variable. Each variable has two attributes (name and data type).

The CometCloud policy language supports all the data types which are supported by the XML Schema and these are: string, integer, floating point number (double), date, time, Boolean, URI, hexBinary, base64-Binary, dayTimeDuration, yearMonthDuration. For example if a Variable is Boolean type then the increment of the variable is not a validate operation.

4 Conceptual Architecture & Policy Descriptions

A proof of concept prototype, called CloudBIM, has been constructed using the *Master/Worker* programming model supported by CometCloud. The prototype supports a multiactor project, with several different organisations participating, each providing a set of workstations which connect to a master node. To a user it appears as if this master provides the data for the entire BIM, whereas in reality the data is distributed amongst the workers. The master nodes do not store data, but they are responsible for integrating with the workers in the cloud by way of queries.

The architecture illustrated in Fig. 4 describes how the master and worker capability is integrated into existing software, such as Autodesk Revit [28] (used widely in the AEC domain). In this three layer model the application

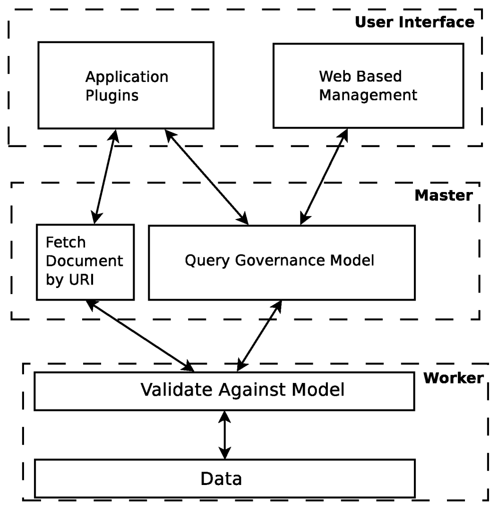

Fig. 4. CloudBIM architecture.

```
CLOUDBIMQUERY=DOCUMENTUPPLOAD | DOCUMENTDOWNLOAD |GOVERNANCEQUERY
GOVERNANCEQUERY = UPDATEQUERY | OTHER QUERY
UPDATEQUERY = 'update',' ',OBJECT,' ',FIELDLIST,' ','set',' ',FIELDLIST
OTHERQUERY = ('get | 'add' | 'delete'),' ',OBJECT,' ',FIELDLIST
DOCUMENTDOWNLOAD='fetchdoc',' ',FIELDLIST,[' ','all']
DOCUMENTUPLOAD = 'adddoc',' ',FIELDLIST,[' ','(',RELATION,{',',RELATION},')']
OBJECT='ProjectStage'|'GateRequirement'|'DocumentSuitability'|'Discipline'
          |'Operation'|'Role'|'Right'|'User'|'Notification'
          |'Flag'|'NotificationType'
RELATION=RELTYPE,' ',<ID>
RELTYPE='ver'|'der'|'comp'|'conc'
FIELDLIST=FIELD,{',',FIELD}
FIELD=OBJECT,'=',FIELDVALUE
FIELDVALUE=VALUE|('(',VALUE,{',',VALUE},')')
```

Fig. 5. EBNF definition of CloudBIM query language.

plugin for Google sketch up communicates with the master node—either by explicitly requesting a document by URI or querying the governance model. The master node then communicates with the worker nodes that actually store the data and enforce the rules set by the governance model.

The master node also acts as a security gateway enabling access to resources within the cloud based on the concepts described in Sections 3.3 and 3.4. Each gateway contains a policy, developed as rules, which indicate which user (based on their discipline, associated rights and roles) can access which document(s). In our policy descriptions, users are the Subject of our system, the Documents are the Target, and the desired actions (read, write, change suitability, etc.) a user wants to perform on the documents are the system's Actions which are governed by Rules and Conditions.

From the user's perspective there is also metadata, referred to as an Authorisation Token which keeps information about a user's identity and determines the associated access permissions. The Authorisation Tokens reside in the Master node and provide information such as the Discipline and Team a user belongs to, user's ID etc.

The policy rules of CloudBIM are presented below according to these concepts. For instance, consider the following scenarios:

1. A User is permitted to: Create a PRIVATE document, Delete/Read/Write a PRIVATE document created by him, Read/Write a document set to TEAM and the user belongs to this Team, Read a document set to FINALISED and the user belongs to the same Discipline, Read any document set to FINALISED, Change document from USER to TEAM for a PRIVATE document created by him.
2. The Team Supervisor is one level above the user and has the authority to manage the team's users. Such a user is permitted to: do anything a User can do, Delete a PRIVATE document created by the team's users, Create/Delete a TEAM document, Change document from PRIVATE or TEAM to FINALISED.
3. The Discipline Supervisor is one level above the Team Supervisor and manage users associated with a discipline. They can: do anything a Team Supervisor can do, delete a TEAM document created by the discipline's Team supervisor, Create/Delete a FINALISED document of the discipline, Change document from PRIVATE or TEAM or FINALISED to ARCHIVED.

Each worker holds a portion of the governance model along with a subset of all the actual document data within the BIM as shown in Fig. 4. A worker is responsible for storing this data and for validating that the user has access to modify the data by checking their access rights. The distributed architecture and use of CometCloud ensures that all data is replicated across the workers, meaning that there is always more than one copy of each document, allowing resilience if workers go off-line.

When the CloudBIM system is initialised a set number of workers are launched, but additional workers can be started by the Master node as required, i.e. due to congestion or failure of worker nodes. This ability to dynamically initialise additional worker nodes does, however, mean that synchronisation is required to ensure that all BIM documents are sufficiently replicated across the cloud system.

5 Cloud Implementation & Evaluation

Our implementation strategy focused on two possible uses of the governance model:

- as a capability that could be integrated into a custom user interface implemented for a specific project;
- as a capability integrated within existing software as a plug-in (such as existing CAD systems like Autodesk Revit [28] or Google sketchup [21]).

Both of these are demonstrated in this section and implemented as a set of queries. The prototype CloudBIM query language is specified in Fig. 5 in Extended Bachus Naur Form (EBNF) notation. For the sake of brevity the terms ID (a unique ID) and VALUE (a string) are not defined, also omitted are commands used to authenticate a user.

The CloudBIM query language defines six key commands: *get, add, delete, update, adddoc, and fetchdoc*. These commands allow the manipulation of objects within the governance model, however it should be noted that not all objects are able to be directly manipulated by users, some are created/updated as a side effect of other queries i.e., specifying the relationship of a new document will lead to the automatic creation of Relationship and Transaction objects as necessary by using data supplied in the *adddoc* command.

The *adddoc* and *fetchdoc* commands separate the uploading and downloading of documents from the CloudBIM

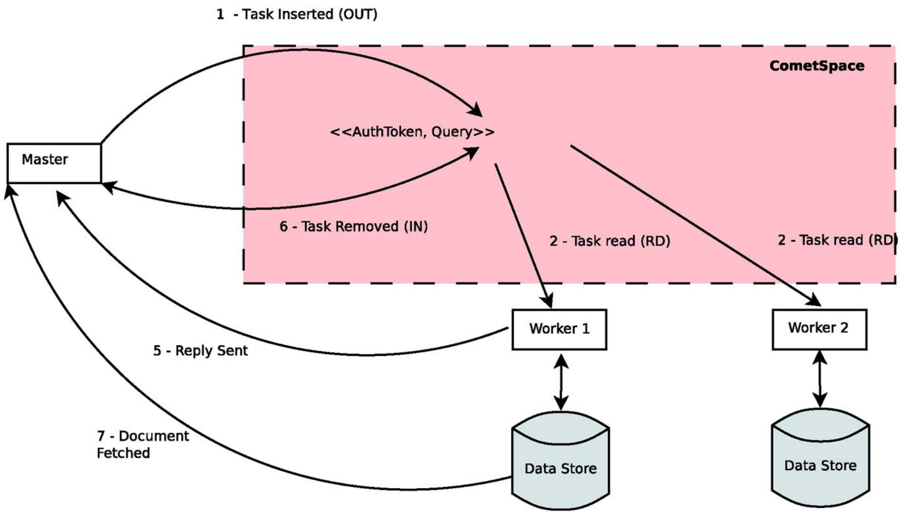

Fig. 6. Querying CloudBIM.

system from the manipulation of the objects within the governance model.

Additionally, it is worth noting that the fetchdoc command can be used to return either all matching documents (not always desirable) or just the first match.

The current prototype makes use of Google sketchup [21] to display a 3D BIM model and can export this to Autodesk Revit [28]. As outlined in the conceptual architecture in Section 4, the workers may also be used to launch external simulations (in batch mode), the outcome of which are also stored in documents, access to which is then based on our governance model. This enables integration of third-party executable software, in addition to static documents that have been pre-generated and stored in CometCloud.

Fig. 7 shows how the gateway (coordinating access to CometCloud) is used to submit tasks (using the `in`, `out`, and `rd` operations). A task is submitted as a tuple to `CometSpace`, where workers then match this task to their own capability. When a tuple matches the capability of a worker, it is removed from CometSpace and managed by that worker.

Results are returned back directly to the gateway, which forwards it to the requesting application (subject to access rights identified in the governance model). The gateway therefore acts as a Policy Decision Point and the worker as a Policy Enforcement Point—according to the IETF model [49].

In Fig. 7 we show a single gateway—however, multiple such nodes can exist to regulate access to documents for a project (with policy descriptions replicated across all project related gateways)—as described in Section 4.

Each query processed by the master is converted into one tuple that is then placed into CometSpace. In the following paragraphs we discuss two examples to illustrate query execution.

Fig. 6 shows how a query that fetches, updates or removes data from the CloudBIM system is executed. In this case the query along with an authorisation token to prove the user's identity is inserted into CometSpace. This

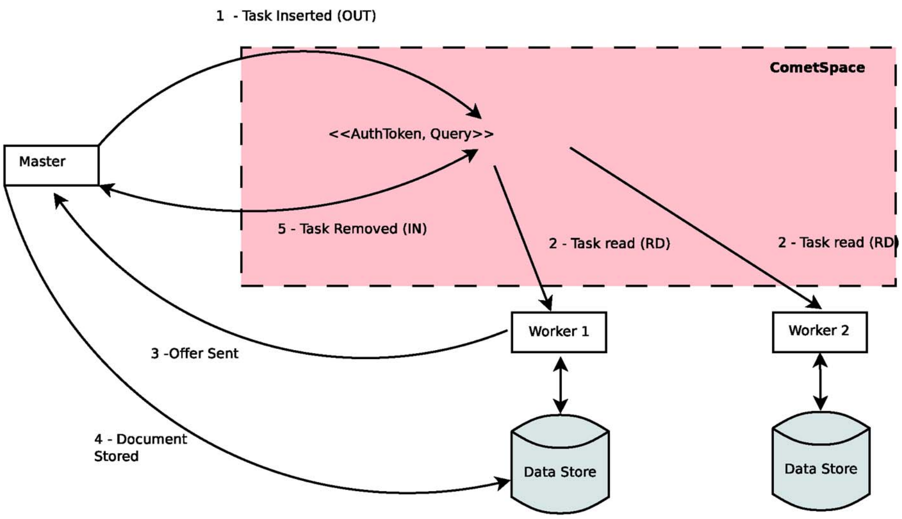

Fig. 7. Adding data to CloudBIM.

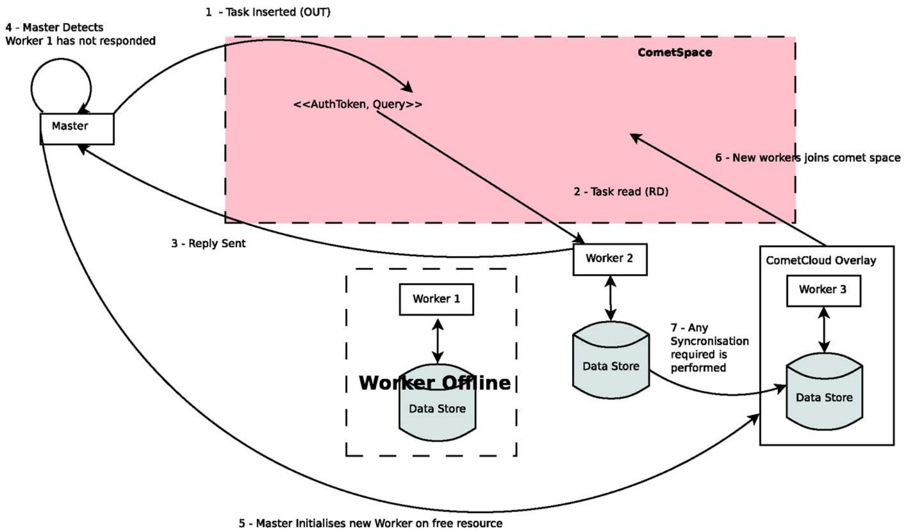

Fig. 8. Fault tolerance in CloudBIM.

tuple is then read by each worker in the system, which will process it, returning any results to the Master. Once the Master receives sufficient replies, it will collate them, return the collated result set to the user and then remove the task from CometSpace.

The second type of query, illustrated in Fig. 7, involves adding data to the CloudBIM system. These queries must ensure that the data is duplicated across the cloud. In this case, when the Master receives the data from the user it will cache the data, so no delays occur for the user while duplication takes place. The query is now inserted into the CometSpace. Once the tuple representing the query has been added into CometSpace available workers will read it and reply to the master with an offer to store the data. The master will then select an appropriate number of workers and duplicate the data to these workers. In terms of the authorisation token, in this case, it is used to determine who can request data from the Master to duplicate it locally. This is checked using the policy held by the Master for each user requesting access to CometSpace.

5.1 Matching Building Artefacts to Appropriate Workers

One of the key concerns when designing the CloudBIM system is that of performance, for users in the AEC sector it is important to allow BIM artefacts to be stored as ‘‘close’’ to the user as possible. To this end the worker nodes, when data is added to the system as shown in Fig. 7, also take into account the domain of the user concerned.

In order to do this each node is given a weighting during its initial configuration. This weighting applies to certain domains and affects the selection of the node to store the artefacts. When (as shown in Fig. 7) the worker replies with an offer to store a BIM artefact, the master will make a decision based on the weights of the workers that have replied. More specificially the process that is followed by a worker is:

1. Worker reads tuple from CometSpace.
2. Worker determines if it is currently able to store the data.
3. If it is then it uses the authorisation token to determine the domain of the user and artefact.
4. Worker returns to the master an offer to store the data containing the weighting to be used.

It should be noted however, that even if a worker has a high weighting for a specific domain, then it is still possible that the worker could not reply because, for example, it currently has no storage space available.

5.2 Fault Tolerance

The CloudBIM system also has mechanisms for fault tolerance and the ability to expand its pool of workers as required. This is an essential property to provide within a governance model to ensure availability of content.

CometSpace consists of a pool of resources/machines running the CometCloud overlay. When the CloudBIM system is launched a set of workers, defined by IP addresses in a configuration file, are initialised using nodes from this pool. If a worker fails, the procedure outlined in Fig. 8 is followed.

This procedure is that, when a query is issued, the Master node will count the number of workers that process the query and if a single worker repeatedly fails to respond within a certain time frame (the number of failures and the time-out value are configurable), then the worker is considered to have failed. While this is taking place user requests are still being processed because BIM data will

TABLE 1
Project Documents

Document	Discipline	Suitability	Notes
Architectural Model	Architect	Finalised	Several Versions used
Structural Model	Structural Engineer	Team	Derived from Architectural Model
Task Manag. Document	Project Manager	Private	
Press Release	Project Manager	Private	A Project Document

TABLE 2
Project Access Rights

Document	Architect	Structural Engineer	Project Manager
Architectural Model	✔	✔	✔
Structural Model	✖	✔	✔
Task Management Document	✖	✖	✔
Press Release	✖	✖	✔

still be available from other workers in the system (due to data duplication). Only in the case of multiple simultaneous failures would users be unable to retrieve data.

In cases where a worker (or set of workers) lose connection for a long period of time (a timeout value set by an administrator) the data on the worker is removed. This means, that when a worker that has been offline for a long period of time re-connects, it can retrieve a new set of data from other workers in the CloudBIM system, removing the risk of any invalid (outdated or deleted) data from becoming available to users.

Once a worker has failed, it is removed from the current list of workers and a new worker is added from the pool by communicating with the CometCloud overlay that will be running on the target node. Once this is done, the CometCloud overlay must then be restarted to enable correct routing of messages to the new worker. Once this has taken place, the new worker will have joined CometSpace but synchronisation may be needed to ensure that there is sufficient duplication of BIM data. This entire process takes place transparently to the user.

A similar situation is when a new worker needs to be added to the system to improve system throughput—in this case a similar process is followed: the new worker is initialised from the pool of resources and the CometCloud overlay is restarted.

6 Prototype Validation

Initial validation of the CloudBIM concept, its associated data model and security policy is carried out in two ways. The first involves describing the prototype using a number of use cases supplied by our construction project management industry partner—involving the development of a specialist user interface in Java and integration with Google sketchup. The second, described in Section 6.1, involves carrying our a performance analysis of using the operations in our query language and the fault tolerance capability described in Section 5.

In the use cases, it is assumed that the project follows the RIBA project stages [32] and these stages will make up the project lifecycle. We consider a number of actors in the system, each assigned to one of three disciplines: Architect, Project Manager and Structural Engineer. To these users and disciplines, roles will be assigned granting each entity a set of rights on the abstract documents (as described in Section 3.4) within the CloudBIM system, which are shown in Table 1. The selection of one of these documents allows the user to directly visualise its contents, either consisting of a single file or a multi-object BIM document. A high level summary of these access rights is shown in Table 2 where a tick indicates that access rights have been granted and a cross indicates that no access is permitted. Access rights in the demonstrator are granted at a document level, however these also apply on an object level, assuming that all objects within a given model inherit the same rights. Hence, the BIM project will involve objects at different levels of granularity with attached stage-based access policies.

The early phases of design work on a project will be conducted by the Architect and the Project Management disciplines. A series of documents, each detailing different design artefacts will be created by the architect. When an initial version of the design has reached a satisfactory completeness, the architect will review it and set its suitability to allow the other disciplines to utilise it. Fig. 9 shows how the prototype can model the versioning

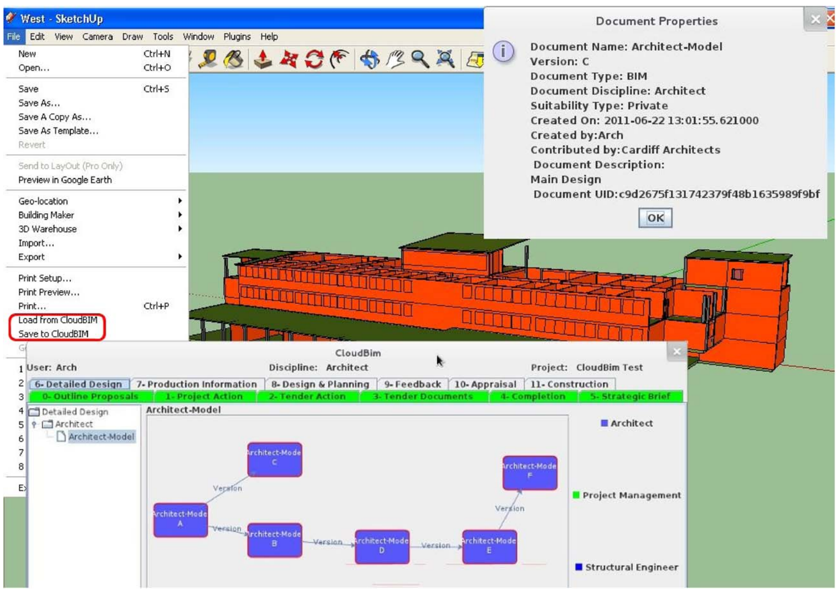

Fig. 9. CloudBIM prototype.

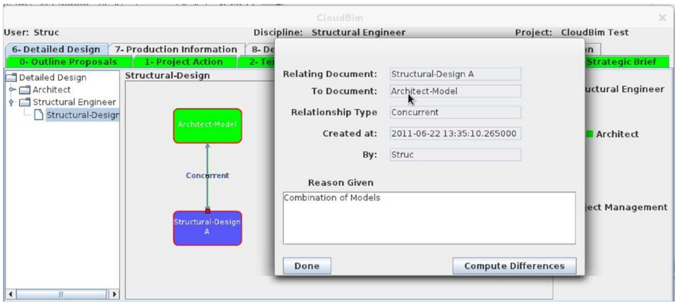

Fig. 10. CloudBIM prototype—concurrent relationship.

relationships in the architectural model and the metadata that is stored about each version of the document. It also shows how the process of the BIM project is illustrated, with the RIBA project stages presented as tabs, these tabs are then highlighted (in a pre-defined colour) to signify which stages of the project have been completed. Currently architectural documents are added to and removed from the system by the use of a plug-in developed for Google Sketchup as shown in Fig. 9.

Once the document type of the architect's model has been changed the structural engineer can then view and create a model which can then be worked on concurrently with the architectural model. A structural engineer can therefore see documents from their own discipline and the architectural model (because its suitability type is set to finalised). The concurrent relationship and the metadata stored regarding the relationship between the architectural model and the structural analysis model is shown in Fig. 10.

From Table 2 we observe that the project management discipline has access to all of the documents that currently exist within the system, and this is used to create another derived document from the architectural model. This is shown in Fig. 11.

In this section we have illustrated how the CloudBIM prototype is able to present the information stored in the governance model to the user and provide them with an accessible interface to the CloudBIM system. The CloudBIM prototype itself is only loosely coupled with the CloudBIM system, interacting only via the query language described in Section 5. This has allowed us to implement the client as a Java plug-in for Google Sketchup. This ability to develop clients solely based on the query language is important as it becomes far simpler to develop clients for other industry standard tools, such as Autodesk Revit [28] or even develop browser or mobile device-based applications.

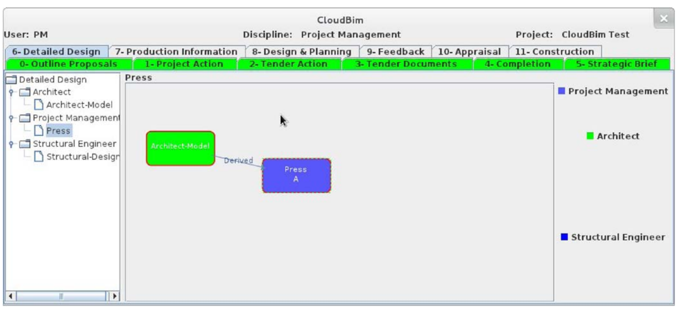

Fig. 11. CloudBIM prototype—increased access priviledges.

TABLE 3
Performance of Failure Tolerance and Addition of New Workers

1 Worker Failure	1 new Worker	2 new Workers	3 new Workers
3.22s	3.49s	3.59s	3.94s

6.1 Performance Results

In addition to evaluating the system qualitatively against the requirements defined by the AEC industry, we also evaluated the performance of the system. As part of this evaluation we collected performance data for recovering from worker failure, adding of new workers and BIM document upload/download.

The test system that these results were gathered on consists of four virtual machines. One acting as a client (running Windows XP), a second as the Master (running Fedora Core 15) and the final two as Workers (running CentOS). The client, master and workers were all connected to the campus network at a speed of 100 MBit/s, but were running in separate locations across the campus on different subnets. However, as will be discussed later, the client and the master were running on machines in the same building, but the workers were located in a different facility, this has had an impact on the performance results.

Table 3 shows the time taken for the system to recover after a single worker has failed, measured from when the Master node has flagged that the worker has failed until the system is restored with a new worker. This measurement does not include the time for any synchronisation of document data. Table 3 also shows the time taken for new workers to be added to the system, measured from when the *add worker* command is issued until the system is fully restored with the new functional worker. Our results show that the runtime of each of these tasks is very similar and, more importantly, the process is relatively fast. Recovering from worker failure and the runtime for adding new workers represent similar processes. Adding multiple workers does not significantly increase the overall deployment time.

Table 4 shows the performance of uploading data to the CloudBIM system. In this table, the governance overhead is the time taken to add the document to the governance model and validate the user's access rights. The upload time is the time taken for the client to upload the document to the master node and the duplication time is the time

TABLE 4
Document Upload Performance

File Size/MByte	Governance Overhead/s	Upload Time/s	Duplication Time/s
50	1.24	4.65	9.34
100	1.48	9.30	18.34
200	1.45	18.24	38.26
500	1.16	49.71	143.70
1000	1.53	100.60	327.51

TABLE 5
Document Download Performance

File Size/MByte	Governance Overhead/s	Download Time/s
50	0.63	6.13
100	0.58	11.49
200	0.67	25.25
500	0.69	61.74
1000	0.84	172.39

taken for the master to duplicate the data to the appropriate worker nodes. This means that from a user's perspective the actual upload time is *Governance Overhead + Upload Time*. From the results in Table 4 we can see that on average the system is achieving approximately 10 MByte/s upload speed, which is very promising performance. The duplication time speeds are slower because the duplication factor for these experiments was two, meaning that each document has to be transmitted to two workers.

Table 5 shows the speed achieved for downloading BIM data from the CloudBIM system. Here, as in the previous table, the governance model overhead represents the time taken to locate the document and validate the user's access rights to that document. The results show that the download speed from the CloudBIM system is slower than its upload speed at approximately 7.5 MByte/s. The probable cause of this is a slower network connection between the master and workers compared to the connection between the client and the master. This is an issue only when downloading a document, as it must be accessed directly from a worker, whereas when uploading a document is directly uploaded to the master and distributed to the workers from the master.

A final set of tests have been conducted to further ascertain the overhead of the governance model and that due to the distributed nature of CometCloud. Multiple simultaneous connections were established to the CloudBIM system and the average governance overhead is shown in Table 6. The request issued to the CloudBIM system was a document lookup query and the governance overhead is the time taken from when the client issues the query until it received back the URI to the document—it should be stressed that these times do not include the time needed to download the data file.

The figures in Table 6 are very encouraging, as we can see that, unsurprisingly, although the governance overhead does increase under load, it increases at below a linear rate. This demonstrates that our governance model implementation and CometCloud itself is able to scale to deal with higher levels of load. We can have additional confidence in the ability of our system to scale as it is built using the CometCloud framework which, as we have mentioned in Section 2.2, has been demonstrated to be highly scalable.

In this section we have described the performance of the CloudBIM system for several different scenarios. The results shown here are promising: the CloudBIM system is able to recover from single worker failure or can add multiple new workers in under 4 seconds. When uploading BIM data it can achieve an upload speed of approximately 10 MByte/s and a download speed of 7.5 MByte/s. More importantly the results in this section have also shown that the overhead of using the governance model to manage access to the BIM data is less than two seconds per query and we have also shown that the overhead scales at below a linear level when increasingly large numbers of nodes issue simultaneous queries.

TABLE 6
Governance Overhead

Number of Simultaneous Connections	Governance Overhead/s
1	0.73
5	1.43
10	1.29
25	1.96
50	1.66
100	2.19
200	3.16

7 Conclusion

Many AEC projects involve multiple, distributed participants who need to share and use data. Such data and the associated processes involved in manipulating the data, are increasingly being represented using the Building Information Model (BIM). Such data is often fragmented across disk drives and local storage resources of various participants involved in an AEC project. Cloud computing provides a more consolidated storage capability for such data, in addition to the ability to preserve and archive this data after the completion of a building for use by others over a long time frame.

Two key novel contributions of this work include the development of a prototype Cloud system that uses a distributed shared-space abstraction to enable data sharing in a particular industry sector—we believe this provides a much more seamless integration across different roles involved in an AEC project, and an access control/governance model, that utilises the BIM standard (taking account of disciplines, roles and access rights at different stages of a building lifecycle), integrated with our Cloud implementation. We show how our implementation can be integrated with open source tools such as Google sketchup and carry out failure and performance tests on the implementation, demonstrating scalability and fault tolerance in our CloudBIM system.

The CloudBIM system has been implemented using the CometCloud system, enabling the creation of a logically centralized, but physically distributed (implemented using a Peer-2-Peer overlay network) data sharing environment. Access to this environment is implemented through Linda-like primitives. Our governance model involves the description of a policy at each access point (moderated through a Master node), providing varying types of access to documents within CometCloud (moderated through a collection of workers). We demonstrate how access to workers is achieved through a collection of queries, thereby enabling our governance model to be used via a specialist user interface or integrated alongside open source (such as Google sketchup) or proprietary (licenced) software (such as Autodesk Revit). We validate the governance model using a scenario provided by our industry partner involved in the project and also demonstrate the performance overheads of using the model. Although the focus of this paper is on the AEC industry, our governance model and the associated query interface and the CometCloud-based implementation can be used in a number of other similar multi-partner projects in other service-oriented computing domains—such as Aerospace engineering, Computational Chemistry, etc.

References

[1] BIM Working Party Strategy PaperA Report for the Government Construction Client Group, Commissioned by the Department of Business, Innovation and Skills, 2011.

[2] IFC—ISO/PAS 16739:2005.

[3] J. Egan, *Rethinking Construction*, Department of the Environment, Transport and Regions.

[4] E.W. East, *Construction Operations Building Information Exchange (COBie)*. (Sept. 2010). [Online]. Available: http://www.wbdg.org/resources/cobie.php

[5] Y. Rezgui and J. Miles, "Exploring the Potential of SME Alliances in the Construction Sector," *Eng. Manage. (J. Amer. Soc. Civil Eng.)*, vol. 136, no. 5, pp. 558-567, May 2010.

[6] W.-C. Tsai, "Improving Communication Barriers for On-Site Information Flow: An Exploratory Study," *Adv. Eng. Informat.*, vol. 23, no. 3, pp. 323-331, July 2009.

[7] S. Titus and J. Brchner, "Managing Information Flow in Construction Supply Chains," *Construct. Innov.*, vol. 5, no. 2, pp. 71-82, 2005.

[8] M. Latham, *Constructing the Team: Final Report of the Government/Industry Review of Procurement and Contractual Arrangements in the UK Construction Industry*," HMSO, London, U.K., 1994.

[9] Y. Rezgui and A. Zarli, "Paving the Way to Digital Construction: A Strategic Roadmap," *Comput. Civil Eng.*, vol. 132, no. 12, pp. 767-777, 2006.

[10] P. Barrett and M. Sexton, "Innovation in Small, Project-Based Construction Firms," *Brit. J. Manage.*, vol. 17, no. 4, pp. 331-346, Dec. 2006.

[11] Low Carbon Construction Innovation and Growth Team: Final Report, HM Government, London, U.K., 2011.

[12] The "BROADEN" Project: Business Resource Optimisation for Aftermarket and Design on Engineering Networks, The White Rose Grid. (10 Jan. 2012) [Online]. Available: http://www.whiterose.ac.uk/node/66

[13] *Collaborative Production of Architectural, Engineering and Construction Information, Code of Practice*, British Standard 1192:2007, British Standards Institute (BSI), Jan. 31, 2008.

[14] UK Cabinet Office, Government Construction Strategy, 2011.

[15] Y. Chen and R.H. Katz, "Glimpses of the Brave New World for Cloud Security," in *HPC in Cloud*. (June 2011) [Online]. Available: http://www.hpcinthecloud.com/hpccloud/2011-02-22/glimpses_of_the_brave_new_world_for_cloud_security.html

[16] G. Cooper, C. Cerulli, B.R. Lawson, C. Peng, and Y. Rezgui, "Tracking Decision-Making During Architectural Design," *Electron. J. Inf. Technol. Construct.*, vol. 10, pp. 125-139, 2005.

[17] A. Dainty, D. Moore, and M. Murray, *Communication in Construction: Theory and Practice*. Abingdon, Oxon, U.K.: Taylor & Francis, 2006.

[18] D.K. Smith and M. Tardif, *Building Information Modeling: A Strategic Implementation Guide for Architects, Engineers, Constructors, and Real Estate Asset Managers*. Hoboken, NJ, USA: Wiley, 2009.

[19] J. Kent Holland, "Risk Management for Design Professionals in a World of Change: Bringing into Focus Green Design," in *International Risks, and New Contract Documents*. Vienna, Austria: Ardent Publ., 2010.

[20] The "GOLD" Project: Grid-Based Information Models to Support the Rapid Innovation of New High Value Added Chemicals, University of Newcastle. (10 Jan. 2012) [Online]. Available: http://www.neresc.ac.uk/projects/GOLD/

[21] 3D Modelling Software From Google (Version 8). (15 Jan. 2012) [Online]. Available: http://sketchup.google.com/

[22] R. Howard and Bjork, "Building Information Modelling: Experts' Views on Standardisation and Industry Deployment," *Adv. Eng. Informat.*, vol. 22, no. 2, pp. 271-280, Apr. 2008.

[23] R. Jardim-Goncalves and A. Grilo, "SOA4BIM: Putting the Building and Construction Industry in the Single European Information Space," *J. Autom. Construct.*, vol. 19, no. 4, pp. 388-397, July 2010.

[24] H. Kim, S. Chaudhari, M. Parashar, and C. Martyy, "Online Risk Analytics on the Cloud," in *Proc. 9th IEEE Int'l Symp. Cluster Comput. Grid International Workshop on Cloud Computing, in Conjunction*, 2009, pp. 484-489.

[25] H. Kim and M. Parashar, *CometCloud: An Autonomic Cloud Engine, Cloud Computing: Principles and Paradigms*. Hoboken, NJ, USA: Wiley, Apr. 2011, pp. 275-297.

[26] B. Kumar and J.C.P. Cheng, "Cloud Computing and Its Implications for Construction IT," in *Proc. ICCCBE*, Nottingham, U.K., June/July 2010, p. 315.

[27] NIST, Cloud Computing Definitions. (June 2011). [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[28] Autodesk Revit Architecture. (15 Jan. 2012). [Online]. Available: http://usa.autodesk.com/revit-architecture/

[29] Y. Rezgui, G. Cooper, and P. Brandon, "Information Management in a Collaborative Multiactor Environment: The COMMIT Approach," *J. Comput. Civil Eng.*, vol. 12, no. 3, pp. 136-144, July 1998.

[30] Y. Rezgui, C. Hopfe, and C. Vorakulpipat, "Generations of Knowledge Management in the Architecture, Engineering and Construction Industry: An Evolutionary Perspective," *Adv. Eng. Informat.*, vol. 24, no. 2, pp. 219-228, Apr. 2010.

[31] Y. Rezgui and J.C. Miles, *Harvesting and Managing Knowledge in Construction: From Theoretical Foundations to Business Applications*. New York, NY, USA: Taylor & Francis, 2011, 228 pp.

[32] Royal Institute of British Architects (RIBA) Plan of Work. (17 Jan. 2012). [Online]. Available: http://www.pedr.co.uk/textpage.asp?menu=1a&sortorder=130&area=main

[33] The Rule Markup Initiative (and the Rule Markup Language). (19 Jan. 2012). [Online]. Available: http://ruleml.org/

[34] A. Schlueter and F. Thesseling, "Building Information Model Based Energy/Exergy Performance Assessment in Early Design Stages," *Autom. Construct.*, vol. 18, pp. 153-163, 2009.

[35] M. Serror, J. Inoue, Y. Adachi, and Y. Fujino, "Shared Computer-Aided Structural Design Model for Construction Industry (Infrastructure)," *Comput. Aided Des.*, vol. 40, no. 7, pp. 778-788, July 2008.

[36] V. Singh, "A Theoretical Framework of a BIM-Based Multi-Disciplinary Platform," *Autom. Construct.*, vol. 20, no. 2, pp. 134-144, Mar. 2011.

[37] B. Succar, "Building Information Modelling Framework: A Research and Delivery Foundation for Industry Stakeholders," *Autom. Construct.*, vol. 18, no. 3, pp. 357-375, May 2009.

[38] M. AbdelBaky, M. Parashar, K. Jordan, H. Kim, H. Jamjoom, Z.-Y. Shae, G. Pencheva, V. Sachdeva, J. Sexton, and M. Wheeler, "Enabling High Performance Computing as a Service," *IEEE Comput.*, vol. 45, no. 10, pp. 72-80, Oct. 2012.

[39] A. Devarakonda, M. AbdelBaky, B.L. Claus, I. Rodero, M. Haldeman, and M. Parashar, "Accelerating Asynchronous Replica Exchange on Large Scale Distributed Heterogeneous Hpc Resources," in *Proc. 5th IEEE Int'l SCALE*, 2012, pp. 72-80.

[40] M. AbdelBaky, H. Kim, M. Parashar, K. Jordan, H. Jamjoom, V. Sachdeva, and J. Sexton, "Scalable Ensemble-Based Oil-Reservoir Simulations Using Blue Gene/P-as-a-Service," in *Proc. 4th IEEE Int'l SCALE*, Newport Beach, CA, USA, May 23–26, 2011.

[41] C.J. Anumba, J. Pan, R.R.A. Issa, and I. Mutis, "Collaborative Project Information Management in a Semantic Web Environment," *Eng., Construct. Arch. Manage.*, vol. 15, no. 1, pp. 78-94, 2008.

[42] L.H. Forbes and S.M. Ahmed, *Modern Construction: Lean Project Delivery and Integrated Practices*. London, U.K.: CRC Press, 2011.

[43] *The BuildingSMART BIM Handbook*, BuildingSMART, 2010.

[44] R. Sebastian and L.V. Berlo, "Tool for Benchmarking BIM Performance of Design, Engineering, and Construction Firms in The Netherlands," *Archit. Eng. Des. Manage.*, vol. 6, no. 4, pp. 254-263, Jan. 2010.

[45] *The Business Value of BIM in Europe: Getting Building Information Modeling to the Bottom Line in the United Kingdom*, McGraw-Hill, Frankfurt, Germany, 2010.

[46] C. Eastman, K. Liston, R. Sacks, and P. Teicholz, *BIM Handbook: A Guide to Building Information Modelling for Owners, Managers,*

Designers, Engineers, and Contractors. Hoboken, NJ, USA: Wiley, 2008.

[47] E. Rogers, *Diffusion of Innovations,* 4th ed., New York, NY, USA: Free Press, 1995.

[48] F. Davis, ''User Acceptance of Information Technology: System Characteristics, User Perceptions and Behavioural Impacts,'' *Int'l J. Man-Mach. Stud.*, vol. 38, no. 3, pp. 475-487, 1993.

[49] R. Yavatkar, D. Pendarakis, and R. Guerin, ''A Framework for Policy-Based Admission Control,'' in *RFC 2753*, IETF Network Working Group, 2000. (Mar. 2013). [Online]. Available: http://www.ietf.org/rfc/rfc2753.txt

[50] P. Bernstein and J. Pittman, ''Barriers to the Adoption of Building Information Modeling in the Building Industry'' Autodesk Building Solutions, Tech. Rep., 2004.

[51] G. Aranda-Mena and R. Wakefield, ''Interoperability of Building Information—Myth of Reality?'' *Proc. eWork eBusiness Archit., Eng. Construct.*, 2006, pp. 127-133.

[52] H. Ashcraft, ''Building Information Modeling: A Framework for Collaboration,'' *Construct. Lawyer*, vol. 28, pp. 1-14, 2008.

[53] Y. Rezgui, T.H. Beach, and O.F. Rana, ''A Governance Approach for BIM Management Across Lifecycle and Supply Chains Using Mixed-Modes of Information Delivery,'' *J. Civil Eng. Manage.*, vol. 19, pp. 239-258, 2013.

Thomas (Tom) Beach received the PhD degree in computer science from Cardiff University, United Kingdom. His thesis focused on code porting and application matchmaking on acceleration devices (such as GPUs). His recent work has focused on applications in the Architecture Engineering and Construction domain. This work has included the use of Cloud computing for the storage, security, management and processing of building data and the development of rule checking methodologies to allow this data to be tested against current building regulations and performance requirements. He is a member of the IEEE.

Omer Rana received the PhD degree in neural computing and parallel architectures from Imperial College, London University, United Kingdom. He is a Professor of Performance Engineering at the School of Computer Science & Informatics, Cardiff University, United Kingdom. He was previously the deputy director of the Welsh eScience Centre. Prior to joining Cardiff University, he worked as a software developer with Marshall Biotechnology Limited (in London). He is a member of the IEEE.

Yacine Rezgui received the MSc (Diplôme d'Etudes Approfondies) degree in building sciences from the Université Jussieu-Paris 6 and the PhD degree in computer science applied to the construction industry from ENPC (Ecole Nationale des Ponts et Chaussées). He is a qualified architect and holds a BRE (Building Research Establishment) Chair in 'Building Systems and Informatics' at Cardiff University. He is a member of the IEEE.

Manish Parashar is Professor of Electrical and Computer Engineering at Rutgers University. He is also the founding Director of the Rutgers Discovery Informatics Institute (RDI2), the NSF Cloud and Autonomic Computing Center (CAC) at Rutgers (CAC@Rutgers), and the The Applied Software Systems Laboratory (TASSL), and is Associate Director of the Rutgers Center for Information Assurance (RUCIA). He recently served as Program Director in the Office of Cyberinfrastructure (OCI) at the National Science Foundation (NSF). He is a member of the IEEE.