Coordinating Multi-Site Construction Projects Using Federated Clouds

Ioan Petri[a,*], Tom Beach[a], Omer F. Rana[b,**], Yacine Rezgui[a]

[a] *School of Engineering, BRE Institute of Sustainable Engineering, Cardiff University, Wales, United Kingdom*
[b] *School of Computer Science & Informatics, Cardiff University, Wales, United Kingdom*

Abstract

The requirements imposed by AEC (Architecture/ Engineering/ Construction) projects with regards to data storage and execution, on-demand data sharing and complexity on building simulations have led to utilising novel computing techniques. In detail, these requirements refer to storing the large amounts of data that the AEC industry generates – from building schematics to associated data derived from different contractors that are involved at various stages of the building lifecycle; or running simulations on building models (such as energy efficiency, environmental impact & occupancy simulations). Creating such a computing infrastructure to support operations deriving from various AEC projects can be challenging due to the complexity of workflows, distributed nature of the data and diversity of roles, profiles and location of the users.

Federated clouds have provided the means to create a distributed environment that can support multiple individuals and organisations to work collaboratively. In this study we present how multi-site construction projects can be coordinated by the use of federated clouds where the interacting parties are represented by AEC industry organisations. We show how coordination

*Ioan Petri
**Omer F. Rana
Email addresses: petrii@cardiff.ac.uk (Ioan Petri), beachth@cardiff.ac.uk (Tom Beach), o.f.rana@cs.cardiff.ac.uk (Omer F. Rana), rezguiy@Cardiff.ac.uk (Yacine Rezgui)

can support (a) data sharing and interoperability using a multi-vendor Cloud environment and (b) process interoperability based on various stakeholders involved in the AEC project lifecycle. We develop a framework that facilitates project coordination with associated "issue status" implications and validate our outcome in a real construction project.

Keywords: , Coordination, AEC Projects, Collaboration, CometCloud, Clouds

---

1. Introduction

In the Architecture/ Engineering/ Construction (AEC) industry, projects are increasingly being undertaken by consortia of companies and individuals, who work collaboratively for the duration of the project. Such projects are complex and the consortia members provide a range of skills to the project from its inception to completion. During this process, various data artifacts are also generated that need to be stored and shared between project members (generally using access control strategies – which limit what can be accessed at a particular stage of the AEC project lifecycle). The planning, implementation and running of these AEC industry projects requires the formation of secure Virtual Enterprises (VEs) to enable collaboration between its members by sharing project information and resources. An important feature of the consortia is that they are dynamic in nature and are formed for the lifetime of the project [2]. Members can participate in several consortia at the same time and can join or leave a consortium as the project evolves. Cloud computing offers an important computing infrastructure to facilitate the establishment and coordination of such VEs. Cloud computing is expected to enhance capabilities that were generally offered through services made available over the Internet. As well as remote access, Cloud computing also provides enhanced security infrastructure including single sign-on capability, security between consortia members, simple setting up of networks to support VEs, distribution of computationally intensive jobs across multiple distributed processors (based on shared information about available resources) [4]. Each organisation involved in a VE may have access to its own Cloud computing system (privately managed internally within the organisation, or acquired through a public provider such as Amazon.com or Microsoft (via their Azure platform)). As it is unlikely that all members of a consortium will share the same platform, integratiion across multiple platforms is therefore an essential requirement for such VEs to function in an efficient and reliable manner [1].

In the computer science research, various efforts have been proposed to implement such multi-Clouds with research efforts focusing on Cloud interoperability e.g. the Open Cloud Computing Interface (OCCI) efforts at the Open Grid Forum [6]. OCCI provides an API and a set of protocols to enable management capability to be carried out across multiple Cloud providers. A variety of implementations are currently available, in systems such as OpenStack and OpenNebula (two open source Cloud platforms). An alternative approach to interoperability is through the development of specialist gateway nodes which enable mapping between different Cloud systems and the implementation of specialist gateways to connect different Cloud systems, the development of a Cloud Operating System (CloudOS) to connect distributed Clouds (European FP7 "UNIFY" project) to the use of specialist in-network capability to process data in network elements between different end points (GENICloud [7]). Similarly, on-line sites such as CloudHarmony [8] report over 100+ Cloud providers that offer capability ranging from storage and computation to complete application containers that can be acquired at a price, primarily using service-based access models.

On the other hand, in the AEC industry there is an increased interest in Building Information Modelling adoption. Such modelling process for various construction projects represents a complex task. This complexity comes from the construction projects which often require collaboration between employers, designers, suppliers and facilities managers through a range of design and construction tasks. Therefore, using cloud federation in a BIM context can provide a number of benefits such as: (a) reduced project failure caused by lack of effective project team integration across supply chains (b) emergence of new challenging new forms of procurement i.e. Private Finance Initiative, Public-Private Partnership and the design-build-operate and (c) decreasing the whole life cost of a building through the adoption of BIM in facilities management [3, 5].

In this paper, we present the implementation and use of a distributed Cloud system, based on requirements of the AEC sector. The resulting clouds for coordination(C4C) framework can support merging and federation of various models of an infrastructure project from multiple applications, clouds and/or actors using a secure and robust common interface. The process is based on BIM (Building Information Modelling) and data stored by each participant conforms to the IFC(Industry Foundation Classes) data model. We elaborate on the concept of project information "Issue Status" associated with a project in order to determine issuing party's status with responsibility/liability associated and considering the reliance on the data. Our approach involves the implementation of a logical "shared" space that is physically distributed across multiple sites involved in the federation. Such a shared coordination space enables various project members to interact with each other during the stages of a project. We compare our approach to general cloud federation efforts, specifically adapted for the needs of the AEC industry in Section 2. In Section 3 we present the CometCloud system and how this system has been used to create the federated cloud framework, followed by a description of the "Cloud4Coordination" (C4C) system and the associated Application Programming Interface (API) that makes use of CometCloud in Sections 4 and 5. In Section 6 we evaluate the C4C system by devising a project trial based on a real construction project and provide overall conclusions in Section 7.

2. Related work

In this section we explore several related studies in the fields of AEC collaboration and cloud federation.

2.1. Related AEC technologies

In the AEC industry the concept of decentralised repositories facilitating data storage across multiple servers represents an emerging topic. Such decentralised environments are currently enabled by specialised software such as Revit Server [24] and Bentley System's ProjectWise [23]. In these systems, data is spread between multiple servers (termed integration and caching servers in the case of Bentley, and hosts and accelerators for Revit Server).

However, current implementations do not remove the barriers of centralised repositories. This is due to the fact that despite both Revit and Bentley allowing the distribution of BIM data across multiple servers, there still remains one authoritative (or master) copy of the data, hosted at a central server. This centralized approach leads to both availability/access, security and liability concerns, as data is being hosted on the server operated by one organisation.

In addition to these commercial offerings, the concept of data storage and collaboration is also a topic of active research in the AEC sector. In their work on SocialBIM, Das et al.[25] have developed a BIM framework that primarily focuses on modelling the social interactions between stakeholders. The key development is SocialBIM's ability to allow users to contribute/download partial BIM models that are then merged/split from a "master" model held in the SocialBIM cloud system(s). While this ability to work with small "fragments" of BIMs which are then federated is a key development, the fact that the end result is still stored in a centralised way in a cloud system will be of concern to many organisations. Other work in this area includes Munkley et al.[27], who have developed technologies to synchronize data between Revit Server and an external storage server, enabling external users to see a read only copy of the Revit (central) model. While this is an interesting way of allowing increased collaboration using Revit Server, it does not adequately provide for the dynamic two way collaboration that is often required in an AEC project i.e. the ability to incorporate the results of other discipline's work (i.e. the architect, mechanical or electrical engineers) as background in your own work. Finally, this approach is further limited as it is only able to utilise the Revit proprietary data format. Additionally, Boeykens et al. [28] have developed a layered client/server approach that provides an event based communications pool between components embedded into BIM authoring packages. This novel communication approach enables the dynamic sharing of data between components. However, all data is still stored on a centralised server that listens to the event based communications and both saves and injects BIM data into the communications pool as needed. Other solutions for supporting construction BIM data sharing and interoperability include IFC ontology and IFC linked data with federated queries [12], semantic linking and semantic web paradigms with orthogonal solution vector [13] and views modelling [14] where companies work on the same model but with individual access and views. The key differentiating factor of our work is the distributed nature of our approach, where the authoritative copy of data is always stored within a discipline's own servers and is only federated with other disciplines when required. Another key differentiating factor is the increased level of dynamic communication that is possible between multiple disciplines using our approach, i.e. when a single discipline makes updates that are visible to other disciplines. These updates are automatically propagated to the relevant disciplines, without a need for the other disciplines to query if any updates have been made.

Many of these seemingly decentralised approaches (at least from a user's perspective), actually make use of centralised storage and coordination infrastructure. This is undertaken to ensure that the centralised system is adequately protected and managed, and can be monitored for any discrepancies or performance bottlenecks. Existing cloud-based deployments are no different – as they make use of a single, centralised data centre. Our approach differs from these, in that we recognize that each institution involved in an AEC project will need to provide their own computing infrastructure, and more importantly will need to integrate their in-house capability with data centre based cloud systems that may be operated by other institutions/companies. Our approach therefore makes use of a Peer-2-Peer based approach, whereby local data centres can be aggregated with those of other institutions in a seamless manner, but still provide a centralised view on the data shared by institutions involved in a single AEC project. This is achieved using the CometCloud system as described in Section 2.2.

2.2. Related cloud federated systems

Through the federation of cloud systems it has become possible to connect local infrastructure providers to a common framework where participants can exchange data and collaborate. The mechanisms used to support cloud federation can bring substantial benefits for service providers by offering facilities for accessing global services instead of increasing costs associated with building new infrastructure (which may not be fully utilized and may only be needed to support peaks in workload over short time frames). A federated cloud also enables users to host applications with their cloud provider of choice – thereby making local decisions about pricing, software libraries/ systems and deployment environments, while still being able to connect to other computational resources [30, 29, 32]. Various cloud bridging solutions are now available, such as IBM's Cast Iron Cloud Integration [10], part of the Web Sphere suite of tools for developing and deploying applications across different environments. Cast Iron enables integration, through plug-ins, with a number of IBM products (such as DB2) and systems from other vendors, such as SAP and Salesforces CRM – thereby enabling integration between in-house systems and public and private Cloud environments [17]. Many such systems remain proprietary to particular vendors however and are hard to customise to particular use scenarios. CometCloud [18] is an open source solution that has been validated in a number of scientific and financial scenarios. CometCloud has been demonstrated to work alongside specialist computing environments (such as large scale computing clusters that are part of the US TeraGrid and XSEDE projects) and public Cloud systems from Amazon (as described below) [16].

A federated system may have a number of associated access and management policies (based on the sites involved) to be considered in order to increase the utility of providers contributing resources. CometCloud supports a number of different federation models: (i) sites interact with each other using direct communication and (ii) sites interact with each other using a distributed coordination space [19]. In the C4C project, we use and extend the second of these models to enable greater autonomy to be supported for each site involved.

3. Federation in a BIM context

Collaboration in construction projects can bring together various participating companies over the (building construction) lifecycle using different systems and storage solutions. As part of this, the compatibility, control and access of data objects created is critical to the success of a project. Currently, coordination between participants is often a labour intensive manual process and can require a monopoly of software systems to be enforced. A construction project is a complex undertaking depending on a large number of very different professions and firms [22, 37]. These firms range from SMEs to large multinational corporations. Each one of these organisation will participate in the construction project for a varying time period and, in that time period, will contribute different quantities and types of data to the project, or even contribute no data. As we have previously described, while interest in cloud based BIM solutions is increasing, there are still many obstacles to BIM adoption that must be overcome. These include: (a) lack of clarity as to who owns and is responsible for BIM (b) fragmentation of BIM data across design and engineering teams and then the contractor and FM companies and (c) information is not sustained across the lifecycle and is in continuous danger of being lost due to company mergers or bankruptcy [11, 15]. In response to these obstacles we propose the use of an BIM federation overlay to implement a federated distributed BIM data model within a construction project.

3.1. CometCloud Federation

Through the federation of Cloud systems it has become possible to connect local infrastructure providers to a global marketplace where participants can transact (buy and sell) capacity on demand. The mechanisms used to support cloud federation can bring substantial benefits for service providers by offering facilities for accessing global services instead of increasing costs associated with building new infrastructure (which may not be fully utilized and may only be needed to support peaks in workload over short time frames). More importantly, organisations with spare capacity in the data centre are now provided with a simple way to monetize that capacity by submitting it to the marketplace for other providers to buy, creating an additional source of revenue.

The federation model is based on the Comet coordination "spaces" (an abstraction, based on the availability of a distributed shared memory that all users and providers can access and observe, enabling information sharing by publishing requests/offers to/for information to this shared memory). In particular, we have decided to use two kinds of spaces in the federation. First, we have a single federated management space used to create the actual federation and orchestrate the different resources. This space is used to exchange any operational messages for discovering resources, announcing changes at a site, routing users' request to the appropriate site(s), or initiating negotiations to create ad-hoc execution spaces. On the other hand, we can have multiple shared execution spaces that are created on-demand to satisfy computing needs of the users. Execution spaces can be created in the context of a single site to provision local resources or to support a *cloudburst* (i.e. when additional capacity is needed to respond to a sudden peak in demand) to public clouds or external high performance computing systems. Moreover, they can be used to create a private sub-federation across several sites. This case can be useful when several sites have some common interest and they decide to jointly target certain types of tasks as a specialized community.

As shown in Figure 1, each shared execution space is controlled by an agent that initiates the creation of such a space and subsequently coordinates access to resources for the execution of a particular set of tasks. Agents can act as a master node within the space to manage task execution, or

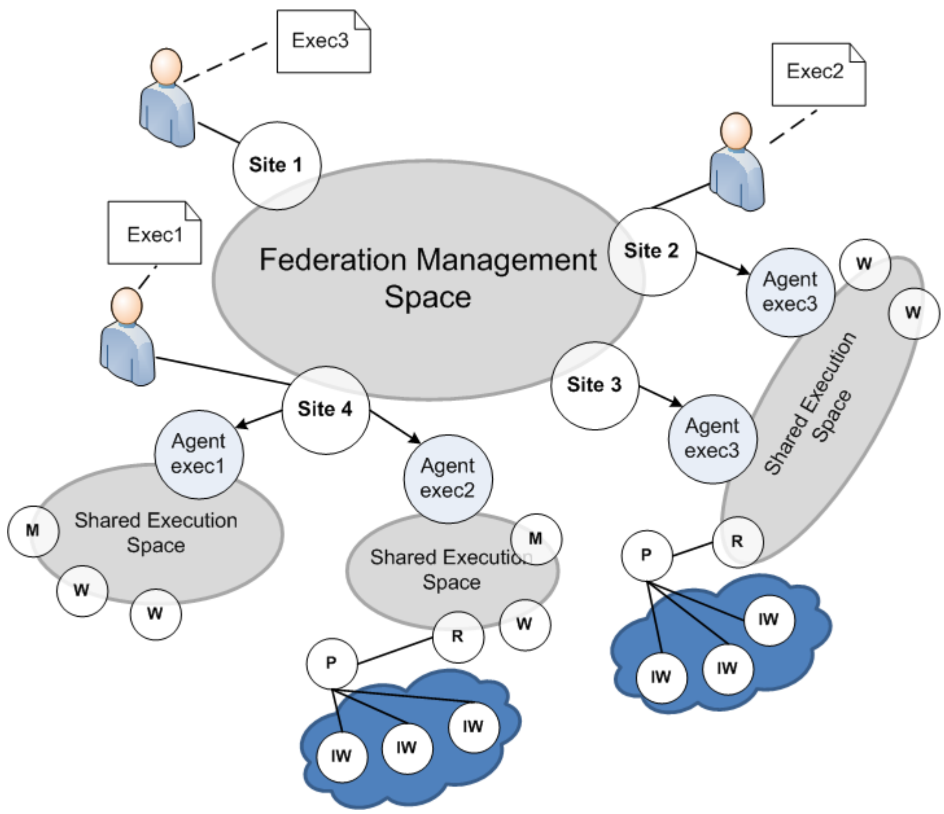


Figure 1: The overall Federation Management Space, here (M) denotes a master, (W) is a worker, (IW) an isolated worker, (P) a proxy, and (R) is a request handler.

delegate this role to a dedicated master (M) when some specific functionality is required. Moreover, an agent deploys a number of workers to carry out execution of tasks. These workers can be in a trusted network and be part of the shared execution space, or they can be hosted on external resources such as a public cloud and therefore in a non-trusted network. The first type of worker is called a "secure worker" (W) and can pull tasks directly from the space. Meanwhile, the second type of worker is called an "isolated worker" (IW) and cannot interact directly with the shared space. Instead, they have to interact through a proxy (P) and a request handler (R) to be able to retrieve task information from the space and execute these..

3.2. CometSpace

CometCloud uses a Linda-like tuple space [31] referred to as "CometSpace" which is implemented using a Peer-2-Peer overlay network. A tuple space enables the implementation of an associative memory-based search strategy, whereby the search term is described as a set of items/terms, which can be mapped against a table of stored data. This search strategy is often easier to implement in hardware and therefore provides a significant improvement in search performance. As an illustrative example, consider that there are a group of data producers and consumers, producers post their data as tuples in the space, and consumers then retrieve data that match a certain pattern. The producers/consumers only have a reference to where such data items should be posted/retrieved from, but do not need to know the physical location/ storage device for such data items. CometSpace [33] is an extension to this tuple space-based abstraction, in that the tuple space can be physically distributed across multiple sites (data centres), and a "logical" space is produced by combining these physically distributed sites. Each producer/ consumer now accesses the logical space, asynchronously, and does not need to know the physical location of the site actually hosting the data. For our needs we have updated the tuple-space mechanisms and the format of tuples to comply with requirements related to data processing, data sharing and data storage as identified in the construction sector. Therefore, a tuple becomes an array formed of {tuple-id, discipline-id, object-serialised, event-id}. In this way, a virtual shared space for storing data can be implemented by aggregating the capability of a number of distributed storage and compute resources [20]. CometCloud therefore provides a scalable backend deployment platform that can combine resources across a number of different cloud providers dynamically, often seen as a key requirement for a project in the AEC sector.

CometCloud is based on a decentralized coordination substrate, and supports highly heterogeneous and dynamic cloud infrastructures, integration of public/private clouds and cloudbursts. The coordination substrate (based on a distributed Linda-based model) is also used to support a decentralized and scalable task space that coordinates the scheduling of tasks, submitted by a dynamic set of users, onto sets of dynamically provisioned workers on available private and/or public cloud resources based on their Quality of Service (QoS) constraints such as cost or performance. These QoS constraints along with policies, performance history and the state of resources are used to determine the appropriate size and mix of the public and private clouds that should be allocated to a specific application request [18].

4. C4C project

In this section we outline the key industry-based requirements of the "Clouds-for-Coordination" (C4C) project. We subsequently describe the CometCloud-based system that has been implemented to address these requirements.

4.1. Project background

The C4C project is addressed to the AEC industry seeking to facilitate collaboration between organisations and looking at aspects related to BIM data management and sharing. As BIM presents the possibility of sharing information throughout the construction and property management sectors, the problem of trust in the data becomes important – more commonly recognised in the AEC industry through the use of 'Issue Status' for physical documents (where documents are given statuses that equate to what they can be reliably used for, and therefore what the issuing party accepts responsibility and/or liability for). There are regulations in the UK, driven by the government, to achieve fully collaborative Building Information Modelling (BIM) (with all project and asset information, documentation and data being electronic) across the AEC sector [35]. This is an especially challenging proposition as the successful delivery of a construction project is a highly complex process; requiring collaboration between designers, suppliers and facilities managers through a range of design and construction tasks. This complexity in itself is a key motivation for the use of BIM, with anticipated financial and time savings offered by its adoption [36]. Other motivating factors for BIM adoption include: (a) project failure caused by lack of effective project team integration across supply chains [37, 22], (b) emergence of new challenging new forms of procurement i.e. Private Finance Initiative, Public-Private Partnership and the design-build-operate [38, 39], and (c) decreasing the whole life cost of a building through the adoption of BIM in facilities management[40].

The C4C project addresses the issue of BIM "ownership" by adopting the approach that each party involved creates and stores (and is responsible for) their own BIM information, rather than uploading it to a central server. More specifically our architecture imposes the following key aspects: (i) the ownership of data remains with the discipline that created that data – which also delegates any updates needed on the data to the discipline ensuring that there is a consistent view also maintained by the discipline owner; (ii) the use of a coordination layer to allow other users to transparently view data and make modification to it; (iii) enable information to be replicated across multiple disciplines (but remain consistent with the data owner), allowing for fault tolerance and prevent data loss. Another important aspect of a management model for BIM data is understanding the data and the stages (workflow) of an AEC project, in the context of how a BIM model is populated with data. In order to do this an abstract process has been defined as the result of our requirements gathering execise. This process has abstracted the approaches defined in BS1192a[34].

In our coordination system we map each site to be a discipline, that can store BIM data, and can be hosted at different organisations that are part of a project. With the use of CometCloud system we deploy a working instance at each discipline by allowing a complete BIM dataset to be visualised, sourced from the information stored at multiple locations (locally managed Cloud systems), without changing how or where the original source material is kept, and ensuring that the capability of the owner to revoke and manage updates is not affected. The project goal is to create a framework for AEC project information "Issue Status", which recognises both the issuing party's status (and consequentially the responsibility/liability associated), as well as acknowledging the receiving party's need or reliance on the data.

4.2. Project implementation

In the C4C project we consider that each site is a organisation involved in a particular project can have one master (agent) and several workers. We have also considered the scenario where a new site may be added during the lifetime of the project, for instance, when a project member may gain access to additional data centres. For addressing these requirements we have developed a multi-cloud API which provides all the necessary operations for managing collaboration once an AEC project has been initiated and launched.

We implement a multi-cloud API for creating publishers, subscribers and exchanging messages within our CometCloud-based system. The key benefit of the publisher-subscriber model enables us to associate a distinct discipline reference with each data producer. A user belonging to a particular discipline (e.g. architect, electrical engineer, mechanical engineer etc) is able to have limited visibility of BIM objects across the different sites that are part of a particular project. What is visible within a specific discipline is dependent on: (i) the current stage at which particular data has been produced; (ii) the maturity of the generated object – referenced through a "suitability" level. Both of these parameters are AEC industry specific requirements, and ensure that objects can be managed and updated without conflict during the lifetime of the project.

In our implementation we consider that each object has a named owner/discipline, a last modified date and a (BS1192:2007+A1:2015) suitability code. These attributes are associated individually at the time of a model upload. The Suitability codes are defined in "BS1192:2007+A1:20015" and fulfil two roles:

(a) it is a claim or assertion made by the authoring organisation in the project, and (b) it is a licence or permission to those other roles to use the information as background to their work, up to the specified extent. We use [discipline-suitability] pairs to specify what suitability is attached to a discipline ([Discipline X - Suitability Y]). We also use suitability codes to determine when a discipline has visibility over other disciplines based on a suitability matrix. We consider that suitability can be applied to each object (per-object basis) and only objects that have a GUID i.e. inherit from IFCRoot can have a suitability. A differentiation case is at the upload stage when for convenience we specify suitability for all objects in the model to upload.

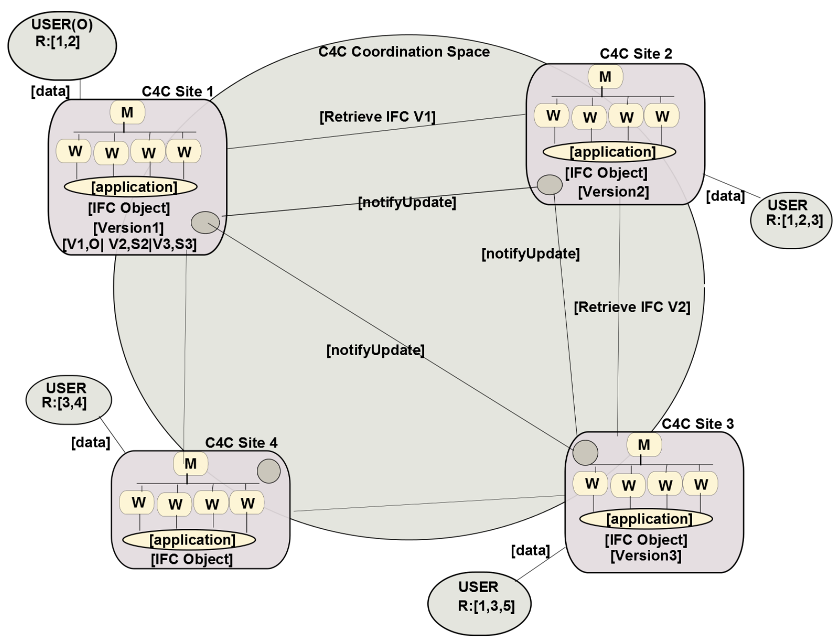


Figure 2: Clouds for coordination multi-site framework.

By using the publisher-subscriber model we enable sites to interact with each other on a common project, using publishers to generate project tasks and subscribers to execute these tasks. We consider the following properties for a site:

- Industry Foundation Class (IFC) objects: a generic language and data model for each of the sites in the coordination space. In our C4C model we operate with IFC objects.
- Roles/Disciplines: we consider that sites can have different roles/ disciplines – which are considered when propagating notification messages associated with updates to particular IFC objects, i.e. which site should be involved at project collaboration stage.

Each site must support a local C4C environment, which enables other sites to interact with it. In the workflow presented in Figure 2 and Figure 5, Site 1 creates the C4C project which is formed of IFC objects locally stored as $Version1$. All other sites participating in the project (Site 2 and Site 3) will be notified about the new project being created (based on their roles in the project). Based on the notification, Site 2 retrieves and updates the C4C project with $Version1$, Site 2 then creates a new version of the C4C project as $Version2$. When a new version is created the interested sites are again notified. Site 3 will also retrieve the latest version $Version2$ and apply updates as part of a new project version – $Version3$. Another round of notifications will be propagated to interested sites (Site 1 and Site 2). Site 4, although part of the coordination space, is expected to contribute to the project at later stages thus will not receive a notification event. It is important to note that Site 1 is the owner of the project, along with the organisation that creates the project and can always retrieve the latest version of the C4C project. In addition, Site 1 also keeps a list of the changes that have been applied to the C4C project over time in a "provenance" (metadata) file. In our example, Site 1, Site 2, Site 3 and Site 4 have associated suitabilities based on which they can access the model and have visibility over other disciplines (can access the objects updated/created by that discipline).

4.3. Computing infrastructure

Our coordination framework can be deployed on infrastructure with varying capabilities, ranging from regular servers to a cluster infrastructure. To conduct our test deployments of the C4C system, we utilised IBM Softlayer [1] virtualized cluster-based infrastructure hosted at IBM's Amsterdam Data Centre, utilising dedicated virtual servers. We utilised a total of four sets of virtualised servers to simulate a construction project with four different disciplines. These are virtual servers hosted in different physical local locations within Softlayer (simulating organisations with standard IT infrastructure and also simulating organisations utilising a cloud based data storage infrastructure), allowing us to simulate a life-like scenario where disciplines within a construction project will utilise multiple IT systems, hosted in differing locations. In the evaluation, we use a server specification of 16CPU cores with 64GB of memory. The networking infrastructure is 1Gbps Ethernet with a latency of 14 ms on average. Each server runs Ubuntu 12.4 and Java 7.

[1] https://control.softlayer.com/ Last accessed: Aug 2015

5. C4C Application Programming Interface (API)

We adapt the functionality of CometCloud for the needs of interoperability in construction projects. In this respect, we implement two APIs; one for supporting multi-cloud use based on the publisher-subscriber (master-worker) model (please refer to Table 1) and a BIM API to comply with the industry standards as presented in Figure 3. The core methods in this API are $getCurrentModel()$ and $updateModel()$: where (i) $getCurrentModel()$ fetches the latest version of the model based on suitabilities and disciplines visibility, and (ii) $updateModel()$ pushes the model with associated changes into the C4C system. For facilitating disciplines to use the background of a project we have developed methods for manipulating IFC objects and corresponding metadata. We have also developed a set of methods for enabling the distributed manipulation of these IFC objects where various disciplines associated with a project can work on the same IFC model. These APIs have roles within the coordination system: (i) to support BIM process and multi-cloud operability and (ii) to interface with the various applications that can connect to the C4C framework. In our project partners have implemented a Revit plug-in to connect Revit software (presented in Figure 6) to the C4C framework and a filtering application which selects IFC objects based on pre-defined suitability codes. The Revit plugin enables communication with the cloud system by integrating the two main API calls (i) $getModel()$ for facilitating model fetching from the cloud and (ii) $updateModel()$ for submitting model changes into the cloud.

The resulting functionality supports multi-cloud operation carried out over an IFC model, by providing mechanisms to transfer data between dif-

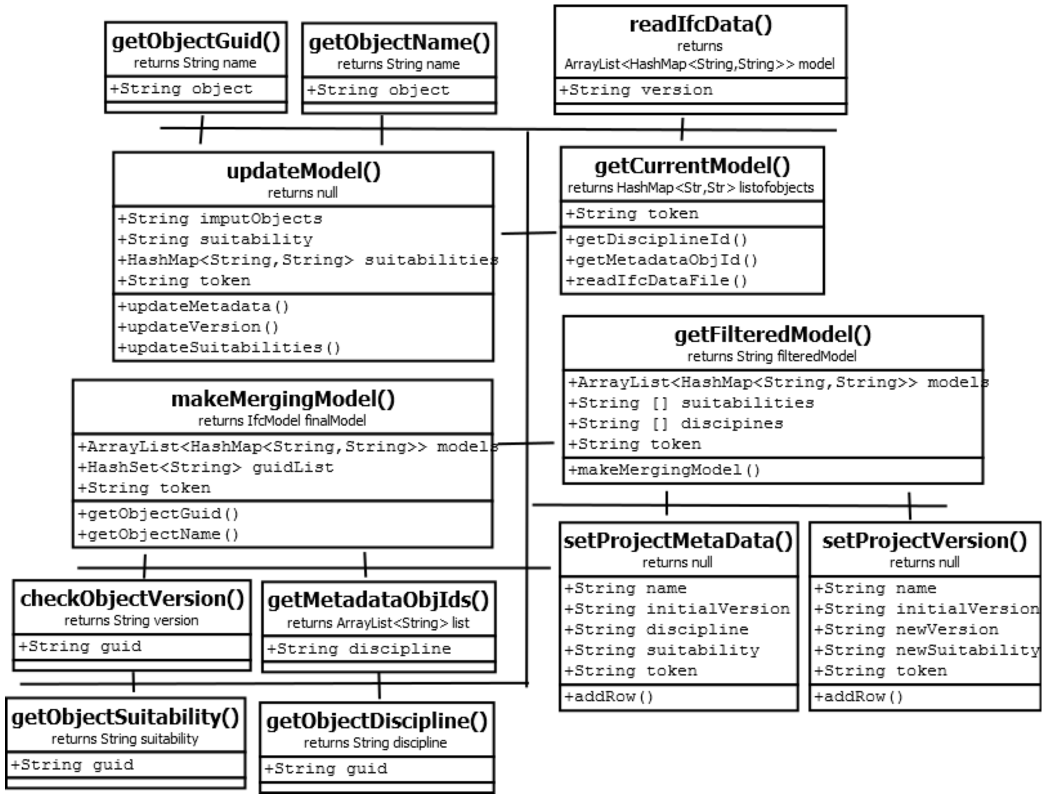


Figure 3: The C4C API

ferent disciplines. This allows disciplines to retrieve in real-time the latest version of an IFC object and to reconstruct the IFC model accordingly. Table 1 presents how the multi-cloud API can be used to enable collaboration between different partner sites.

We assume that each discipline has access to a cloud/data centre. The framework is initialized by calling "startC4CManager()" which then creates the Masters and the Workers based on specific configuration files. If a site is not set to be a Master then the C4CManager will create a proxy in order to link with the existing data centre worker by calling "createIso-

METHOD	DESCRIPTION
addC4CBootStrapNodes()	Sets the bootstrap node
addPorts()	Adds ports for later configurations
bootstrapnodeIsUp()	Checks for any working bootstrapnode
createC4CMaster()	Creates a new master
createC4CWorker()	Creates a new worker
createC4CMasterGeneric()	Implements a generic master
findFreePort()	Looks for available free ports
isBootstrapNode()	Compares the current node with the bootstrapNode
sendMsg()	Sends a message to a destination IP on a specific port
sendMsgToAll()	Sends local subscription list to all nodes(not to bootstrapnodes)
startC4CManager()	Starts federation by creating a master and worker
startC4CWorker()	Starts a C4C local worker
startC4CMasterServer()	Starts a local C4C master
startC4CIsolatedWorker()	Starts a C4C isolated local worker
checkAvailableC4CWorker()	Checks for one available worker
checkAvailableC4CWorkers()	Checking for all available workers based on the number of tasks
getAvailableC4CWorker()	Checks for an idle worker
createTaskData()	Creates data associated with a task
getTaskInfo()	Retrieves task info. based on *taskID*
selectC4CWorkerCreateTask()	Selects a worker, then creates a task to insert to tuple space

Table 1: Multi-cloud API

latedWorker()" method. After the multi-cloud entities have been created, the C4CManager starts all the associated Masters and Workers by calling "startC4CMasterServer()" and "startC4CWorker()" respectively.

For our needs we have updated the tuple-space mechanisms and the format of tuples to comply with requirements related to data processing, data sharing and data storage as identified in the construction sector. Therefore, a tuple becomes an array formed of:

tuple-id: a unique identification of the tuple

discipline-id: unique identification of the discipline

object-serialised: a serialised version of the IFC model retrieved from discipline-id

event-id: the type of operation; fetch or update

6. Evaluation

For testing our system we have conducted a trial using the data and processes from a real construction project provided by the project partner Costain identifying the Highways England construction of a new bridge on the A556, as shown in Figure 4. To undertake the project trial we have deployed our cloud coordination framework on a computing infrastructure described in Section 4.3. The objective of this trial, as agreed with project partners, is to demonstrate the benefits of collaboration in the construction of A556 junction and to demonstrate that difficult linear infrastructure models can be effectively managed by a Cloud/Hosted system to the benefit of all parties.

6.1. Project trial

In the trial we have included different project disciplines and we have provided access to the coordination system via a Revit plug in or a simplified client that utilises the API described in Section 3.1 facilitating direct access to IFC files. The disciplines involved in the project are listed below:

- Contractor - Costain.
- A cost consultant - Lee Wakemans Ltd.

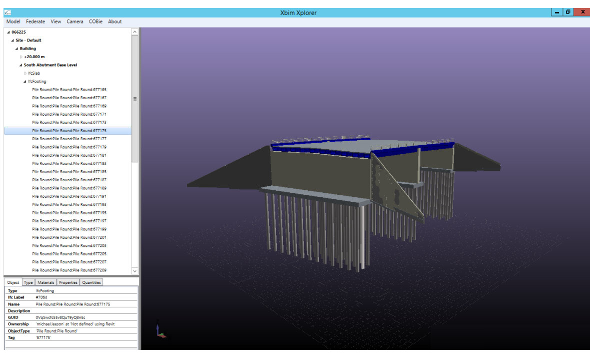
- Designer - Capita.
- Client - User.

The AEC project being considered is a bridge structure with auxiliaries, which involves different disciplines contributing to various parts of the structure. We use four disciplines:(i) C-Contractor, (ii) Q-Cost Consultant, (iii) E-Designer, (iv) O-Client. The IFC models sizes that we utilise in the demonstration are: 250MB, 145MB, 3.44MB, 48KB, all being parts of the bridge on the A556 highway. These input models used for demonstrating the coordination and the output model obtained after merging sub-models from disciplines are presented in Figure 4.

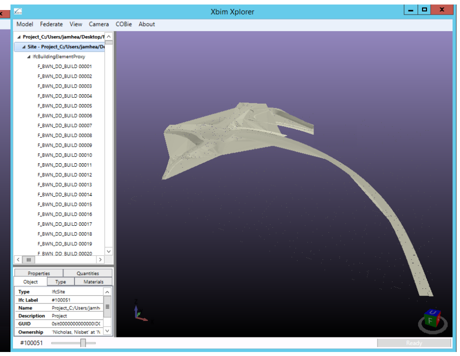
In relation to the process explained in Section 6.2, the overall framework is configured and disciplines are selected with individual roles; from a technical perspective we consider that each server acts as a hosting environment for a discipline and runs CometCloud (in a more general context, a discipline can have multiple servers). The C4C framework is dynamically created at runtime, enabling disciplines to join or leave at any given time. Based on the use of CometCloud [9], each discipline has a master process that receives task requests (IFC objects to update or retrieve) from other disciplines, and is able to forward requests to other disciplines. Each discipline can also have multiple worker processes that carry out actual task executions on locally available resources.

6.2. Framework configuration and workflow

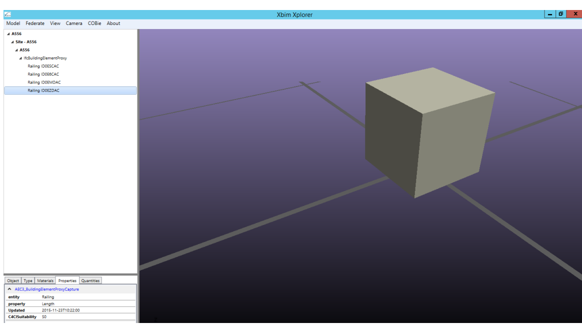
The access to the C4C framework is ensured via a user interface developed based on technical and construction industry requirements. We have



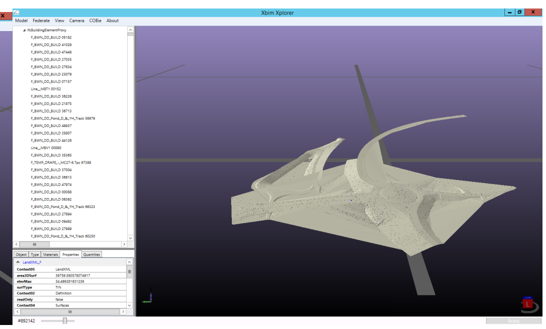
(a) Input IFC Model: Size 3.44MB



(b) Input IFC Model: Size 256MB



(c) Input IFC Model: Size 48KB



(d) Output IFC Model: Size 366MB

Figure 4: Input and output models

developed the user interface for satisfying two functions: (i) initial set up of the C4C network and (ii) ongoing management of the system. The general sequence for the creation of a C4C network is presented bellow:

Step 1: Construction Industry Client [Client] decides to run the project in C4C framework

Step 2: Client downloads C4C software from the web address.

Step 3: Client installs C4C software, determining server IP address and opening the required ports.

Step 4: Client accesses C4C software via IP address and configures pri-

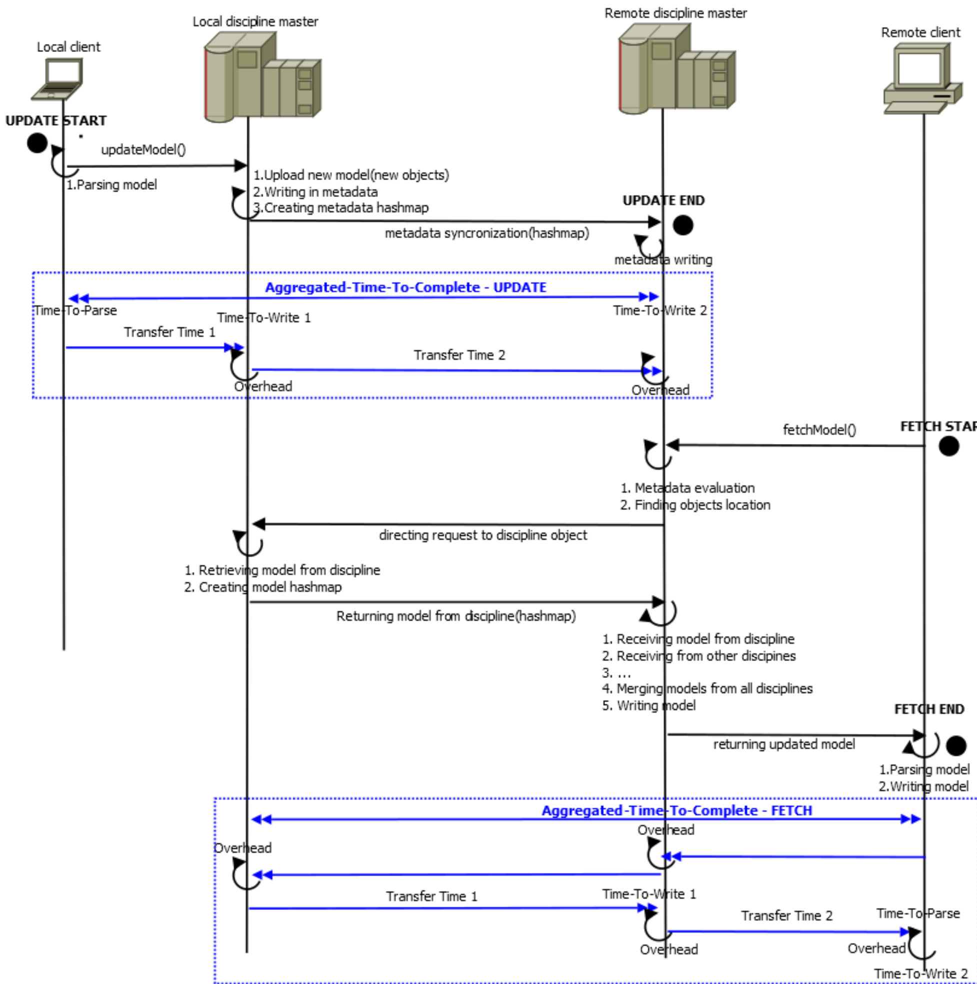


Figure 5: C4C workflow and process sequence

mary project information. Such information include: Project Name, Project Address, Client's Project Number/Reference, Client Company Name, Client Company C4C Primary Contact, Client C4C Primary Contact Email, Client's Nominated C4C Project Manager (not mandatory), Client's C4C Project Manager Email (not mandatory).

Step 5: Following the definition of the project information, the client (or nominated C4C project manager) moves on to the first configuration table. This defines the project disciplines (team members) and what information each discipline can review. The client sends invitations to project disciplines via email with a link to download the C4C software and the coordinator server IP address embedded in the email.

Step 6: Disciplines receive email and install C4C software, noting the IP address for accessing the coordination framework

Step 7: Disciplines access C4C software via IP address and configure their discipline project information

Step 9: After establishing the C4C network, other ongoing management such as adding, removing and editing disciplines and users can be achieved through accessing the same 'core' configuration page. The workflow identifying sequences within the C4C system is presented in Figure 5

6.3. Trial and validation

In this subsection we explain the entire scenario with participating disciplines and iterations that have been followed within the project trial.

Prerequisites: Four disciplines with associated users – each with an IFC viewer, the C4C Client and a terminal displaying the appropriate C4C Master Node to simulate different domains and network addresses. These disciplines are project partners and are as follows:

- Discipline: C - Contractor: Costain- Connecting to master node 5.153.52.162
- Discipline: E - Designer: Capita - Connecting to master node 5.153.52.163

- Discipline: Q - Cost consultant - Lee Wakemans Ltd- Connecting to master node 5.153.52.166
- Discipline: O - Client - Connecting to master node 5.153.52.164

Step 1 - Discipline E: Starting the process "Discipline E" creates an initial bridge model and exports into .ifc using Data Design Systems (DDS) viewer to show design, properties and ownership. Discipline E after creating the model, uploads the model "A556-CAP-7000-S06-3D-S-1001.ifc" into the C4C system with suitability S1.

Step 2 - Discipline C: Another input from a different discipline. "Discipline C" is part of the project and receives the initial bridge design proposal. Discipline C uses Design Builder viewer to colour and filter by slope. After updates, discipline C uploads its model with suitability S0.

Step 3 - Discipline E: Making changes and corrections, introducing different suitabilities. Disciplines E makes some model updates in Revit (as illustrated in Figure 6), fixing railing and adding new IFC objects then uploads the model with suitability S2.

Step 4 - Discipline Q: Using the model to get non-graphic input from a different discipline. "Discipline Q", using filtering (using the API from Figure 3), downloads a costable bridge model, excludes suitability S0, and S1, thereby excluding the ground works and the reinforcement, and generates a cost report. Discipline Q uploads the model with suitability S4.

Step 5 - Discipline O: Taking an overall view. "Discipline O" fetches a full, final integrated model with everything in it (as illustrated in Figure 7). The model A556-CAP-7000-S06-3D-S-1001.ifc is viewed in Tekla BimSight viewer to colour and filter by author and by suitability.

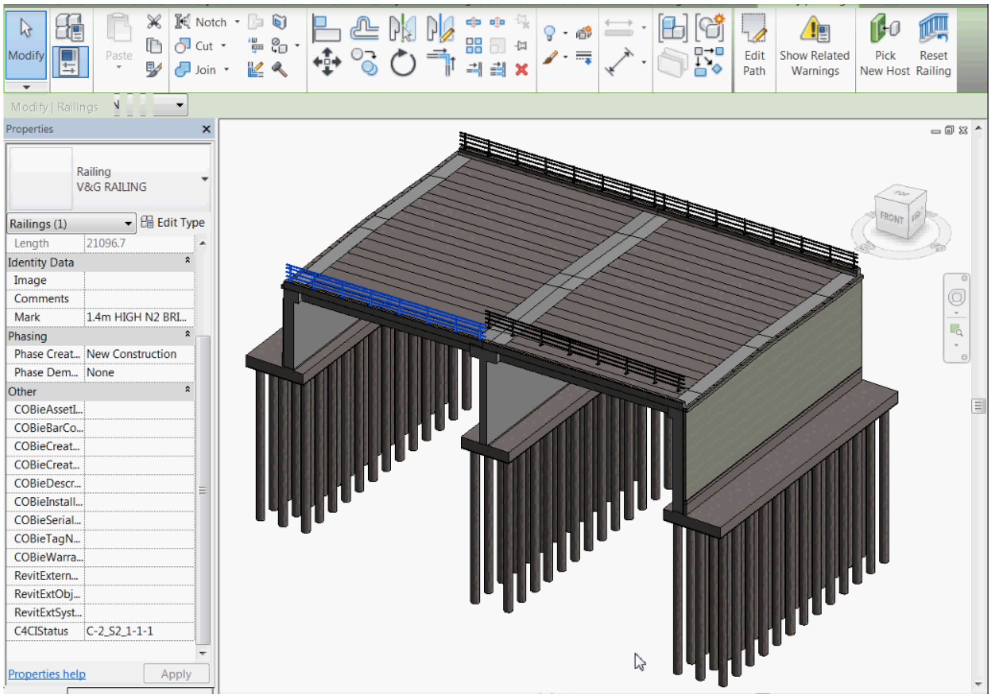


Figure 6: Revit plugin for C4C

6.4. Lessons learnt

This study is based on a collaborative cross-industry research project aiming to enable a collaboration environment for construction industry. The C4C project allows individual “nodes” containing the stored data to be “mapped” between the parties with a technology that can be deployed passively on each party’s computer systems. In essence, C4C allows a complete BIM dataset to be visualised, sourced from the information stored in the multiple locations, without changing how or where the original source material is kept, or who is responsible for that data. Bellow, we list several benefits that our framework provides in relation to multi-site construction project coordination.

Interoperability: The C4C system can support merging (not just federa-

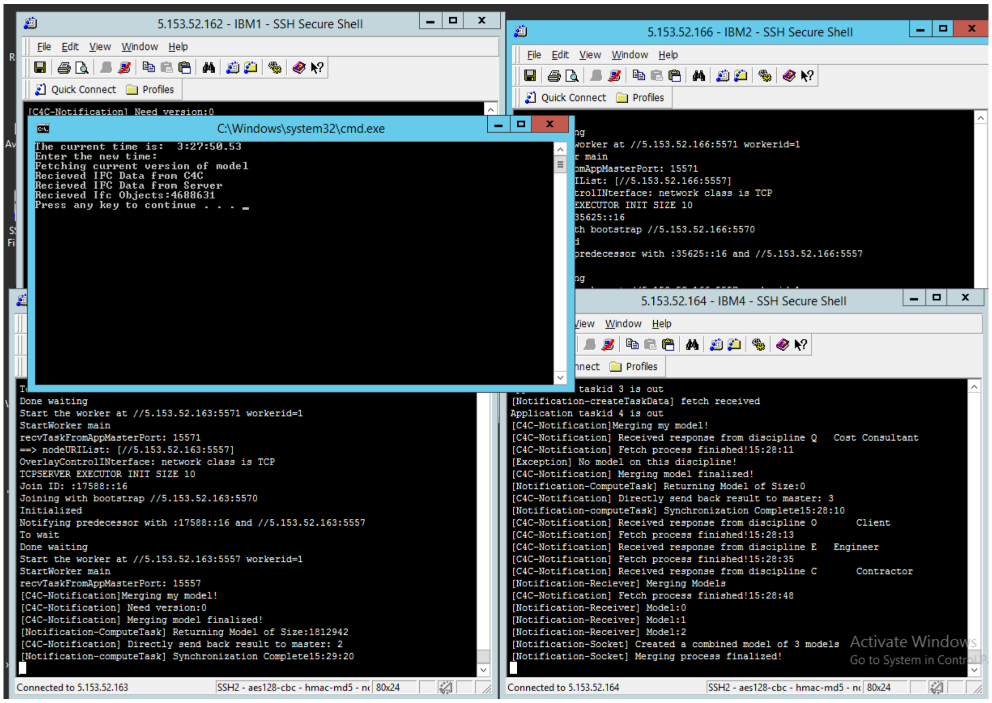


Figure 7: C4C output via terminal

tion) of various IFC models of an infrastructure project from multiple applications, clouds and/or actors (as demonstrated in Section 6), so as to be able to report from the resultant integrated model, using a secure and robust common interface. For example, the system can enable a "Constructor" to create an "integration project" in the cloud, and invite the client, the design team and his sub-contractors to join. Some sub-contractors may invite their own suppliers. All will grant the "Constructor" access to their various current cloud data services relating to the project.

Consistency: Our system can manage federated sub-models and integrate such sub-models into a single view. Based in this, a number of benefits can be observed related to: (i) detection of issues between models, such as differences in volumes (clashes) and specification (properties) and groupings (relationships) and (ii) the creation of a single model by eliminating discrepancies and duplications found in the sub-models.

Trust, Ownership, Flexibility: In our framework each party stores their data on, either their own business computer servers, or their choice of extranet and/or "Cloud" storage in accordance with their own business requirements and protocols. This flexible approach facilitates the federation of a data model in diverse locations and provides several advantages with regards to the requirements that exist in a construction project:

1. Federation is a continuous process, not an event. It proceeds continuously responding to the receipt of updates. At any time the complete model is potentially available, but so too is the list of outstanding issues.
2. Access is given to background information as is pertinent to the current task by role, status and scope and pulled by the agent (who may further restrict the view by role, status and scope).
3. Feedback to agents, whether human or automated, is via messages requesting clarification, analysis and correction. Examples include clashes, evaluations, and discrepancies.

IFC limitations: Over the development of our project we have encountered several challenges with regards to the overall modeling process and to efficaciously manage the Industry Foundation Classes (IFCs).

The most notable challenges of using this format is the issue of Globally Unique Identifiers (GUIDs). GUIDs are used by the software to identify and track objects being processed. In regards to IFC, GUIDs are used to track objects from the BIM dataset and, through this, enable BIM software to know the origin and revision history of each object within the model. Within the IFCs, objects that possess a GUID are always a subclass of IfcRoot.

GUIDs become especially important in a federated model, where the data may be spread across diverse locations and the presence of a GUID is key to tracking the replication of each object. In its current iteration, the IFC file format does not possess GUIDs for some data items (those that are not subclasses of IfcRoot), an example of this is "IfcMaterial". These objects are generally seen as being a property of an object within a BIM model rather than a stand alone object in their own right (even though in the IFC format they are represented as objects). Thus, these types of objects are always associated to an IFC object that does inherit from IfcRoot (thus possess a GUID) and can be tracked within a model. Another problem that we faced during development was the inconsistency of GUIds from CAD packages, as certain CAD packages change an object GUID during the import/export process for IFC data.

In order to rectify these IFC limitations we have implemented a filtering process which compares and thus removes all duplicated objects. This process eliminates the problems related to (a) increased size of the model and (b) duplication of data. The filtering process is performed both for objects that possess a GUID (i.e. those that inherit from IfcRoot) and for those that have no GUIDs. For objects inheriting from IfcRoot, this is performed by doing a per object comparison between the updated IFC file and the model stored on the server; any objects that have changed are updated along with any inter-dependencies. For objects that do not inherit from IfcRoot, these are managed by ensuring that any of these objects are always updated and replaced when the IFC object (possessing a GUID) that they are associated to, is updated.

7. Conclusion

This paper presents a cloud federated framework for supporting project coordination and data sharing across multiple disciplines over the lifetime of an AEC project. When companies collaborate on a particular project need to share data efficiently – moving all data to a single server or location, with subsequent access being controlled to various data sources at such a single location.

We present a coordination model that facilitates companies to maintain their own data (on a local server, within a private Cloud environment, or on storage acquired from a public Cloud provider, such as Amazon), without a need to migrate this data to a central site. We show how overlay-based Cloud environment can be created, where all participants(institutions) in a project can get access to a ”logically” shared data/compute space. This is achieved in this project by using the CometCloud system, which enables a number of different sites to be federated using the concept of a “CometSpace” which maintains physical instances of data at their original point of creation.

Access to data is facilitated through access rights mechanisms, a key advantage provided by CometCloud that supports a secure and flexible environment for multi-site construction projects(unlike other Cloud systems such as OpenStack). The key advantage of our cloud coordination framework represents the near-instant sharing of data between authorised parties in a development project, complete with quality assurance mechanisms and the ability to track and see a history for the development of any object within the dataset.

At a wider scale, we consider that our system can provide useful insides into the process of large project coordination, proposing methods for federating IFC models in distributed locations in a transparent and coherent way. We also state that our cloud-for-coordination framework can map into complex engineering workflows and can present applicability to other domains such as building energy optimisation, water regulations or smart energy grids.

Acknowledgements

We are grateful for continued support from the Rutgers Discovery Informatics Institute (RDI2) on the use of the CometCloud system – in particular Javier Diaz-Montes and Manish Parashar. The C4C project is funded by EPSRC/InnovateUK and involves: Building Research Establishment, AEC3, Costain, Lee Wakemans Ltd, RIBA Enterprises, IBM and Cardiff University.

[1] Mezgar, Istvan, and Ursula Rauschecker. "The challenge of networked enterprises for cloud computing interoperability." Computers in Industry 65(4), 657-674, 2014, http://dx.doi.org/10.1016/j.compind.2014.01.017, ISSN 0166-3615.

[2] Camarinha-Matos, Luis M. "Virtual Enterprises and Collaborative Networks (IFIP International Federation for Information Processing), 5th Working Conference on Virtual Enterprises 22–27 August 2004 Toulouse,

France, 2004, https://dx.doi.org/10.1007/b98980, ISBN: 978-1-4020-8138-5.

[3] Beach, Thomas H., et al. "Cloud computing for the architecture, engineering & construction sector: requirements, prototype & experience." Journal of Cloud Computing: Advances, Systems and Applications 2.1 (2013): 1, http://dx.doi.org/10.1186/2192-113X-2-8, ISSN: 2192-113X.

[4] Rezgui Y, Cooper G, Brandon P: Information management in a collaborative multiactor environment: the COMMIT aApproach. J Comput Civil Eng 1998, 12: 136–145. http://dx.doi.org/10.1061/(ASCE)0887-3801(1998)12:3(136)

[5] Redmond, Alan, et al. "Exploring how information exchanges can be enhanced through Cloud BIM." Automation in Construction 24 (2012): 175-183, http://dx.doi.org/10.1016/j.autcon.2012.02.003, ISSN 0926-5805.

[6] OCCI – Open Cloud Computing Interfaces – available at: http://occi-wg.org/. Last accessed: September 2016.

[7] GENICloud - available at: http://groups.geni.net/geni/wiki/GENICloud. Last accessed: March 2014.

[8] Cloud Harmony - available at: http://cloudharmony.com/. Last accessed: March 2014.

[9] CometCloud Web site - available at: http://nsfcac.rutgers.edu/CometCloud/. Last accessed: March 2014.

[10] Solution, IBM CastIron. Last accesed May 21, 2016, https://www-01.ibm.com/software/integration/cast-iron-cloud-integration/salesforce-integration/.

[11] Shelbourn, M., Bouchlaghem, N.M., Anumba, C. and Carrillo, P., Planning and implementation of effective collaboration in construction projects. Construction Innovation, 7(4), pp.357-377, 2007, http://dx.doi.org/10.1108/14714170710780101, ISSN: 1471-4175.

[12] Pieter Pauwels, Walter Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, Automation in Construction, Volume 63, 2016, pp. 100-133, http://dx.doi.org/10.1016/j.autcon.2015.12.003, ISSN 0926-5805.

[13] Seppo Torma, Semantic Linking of Building Information Models. IEEE Seventh International Conference on Semantic Computing (ICSC), 16-18 Sept. 2013, Irvine, CA, pp. 412–419, http://doi.ieeecomputersociety.org/10.1109/ICSC.2013.80

[14] G.A. van Nederveen, F.P. Tolman, Modelling multiple views on buildings, Automation in Construction, Volume 1, Issue 3, 1992, Pages 215-224, ISSN 0926-5805, http://dx.doi.org/10.1016/0926-5805(92)90014-B.

[15] Hughes, D., Williams, T. and Ren, Z., 2012. Differing perspectives on collaboration in construction. Construction Innovation, 12(3), pp.355-368, http://dx.doi.org/ 10.1108/14714171211244613, ISSN: 1471-4175.

[16] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. A

view of cloud computing. Communications of the ACM, 53(4), pp.50-58, http://dx.doi.org/ 10.1145/1721654.1721672

[17] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).

[18] H. Kim and Parashar, M., CometCloud: An Autonomic Cloud Engine, in Cloud Computing: Principles and Paradigms, John Wiley & Sons, 2011, pp. 275-297, http://dx.doi.org/10.1002/9780470940105.ch10.

[19] Hyunjoo Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish Parashar, Exploring Adaptation to Support Dynamic Applications on Hybrid Grids-Clouds Infrastructure, 1st Workshop on Scientific Cloud Computing, in conjuction with the ACM International Symposium on High Performance Distributed Computing (HPDC), Chicago, Illinois, June 20-25, 2010.

[20] Hyunjoo Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish Parashar. 2009. An Autonomic Approach to Integrated HPC Grid and Cloud Usage. In Proceedings of the 2009 Fifth IEEE International Conference on e-Science (E-SCIENCE '09). IEEE Computer Society, Washington, DC, USA, pp.366–373. http://dx.doi.org/10.1109/e-Science.2009.58

[21] M. Latham, Constructing the Team: Final Report of the Government/Industry Review of Procurement and Contractual Arrangements in the UK Construction Industry., London: HMSO.,UK - Department of Environment, Stationery Office Books, 1994.

[22] J. Egan, Rethinking construction, Department of the Environment, Transport and Regions, The report of the construction task force, Department of Trade and Industry, 2014.

[23] Bentley ProjectWise - available at: https://www.bentley.com/en/products/brands/projectwise. Last accessed: March 2016.

[24] Autodesk Revit Server - available at: https://knowledge.autodesk.com/support/revit-products. Last accessed: March 2016.

[25] Das, Moumita, Jack CP Cheng, and Srinath S. Kumar. "Social BIMCloud: a distributed cloud-based BIM platform for object-based lifecycle information exchange." Visualization in Engineering v.3, (8), p. 1-20, 2015, http://dx.doi.org/10.1186/s40327-015-0022-6.

[26] van Berlo, Leon, and Thomas Krijnen. "Using the BIM Collaboration Format in a Server Based Workflow." Procedia Environmental Sciences 22 (2014): 325-332, http://dx.doi.org/10.1016/j.proenv.2014.11.031.

[27] Munkley, J., Mohamad Kassem, and Nashwan Dawood. "Synchronous Building Information Model-Based Collaboration in the Cloud: A Proposed Low Cost IT Platform and a Case Study." Computing in Civil and Building Engineering, 89-96, 2014, http://dx.doi.org/10.1061/9780784413616.012.

[28] Boeykens, Stefan, and Koenraad Nys. "Interactive bi-directional BIM model and application linking." In Proceedings of the International As-

sociation for Shell and Spatial Structures (IASS) Symposium 2015, Amsterdam Future Visions., pp. 1-11, KIVI, 2015.

[29] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, et al. Cloud federation in a layered service model. J. Comput. Syst. Sci., 78(5):1330-1344, 2012, http://dx.doi.org/10.1016/j.jcss.2011.12.017, ISSN 0022-0000.

[30] Petri, Ioan, Tom Beach, Mengsong Zou, Javier Diaz Montes, Omer Rana, and Manish Parashar. "Exploring models and mechanisms for exchanging resources in a federated cloud." 2014 IEEE International Conference on Cloud Engineering (IC2E), pp. 215-224. IEEE, http://dx.doi.org/10.1109/IC2E.2014.9, 2014.

[31] Carriero, Nicholas, and David Gelernter. "Linda in context." Communications of the ACM 32(4) (1989): 444-458.

[32] Kim, Hyunjoo, Yaakoub el-Khamra, Shantenu Jha, and Manish Parashar. "An autonomic approach to integrated hpc grid and cloud usage." In e-Science, 2009. e-Science'09. Fifth IEEE International Conference on, pp. 366-373. IEEE, http://dx.doi.org/10.1109/e-Science.2009.58, 2009.

[33] Gelernter, David. "Multiple tuple spaces in Linda." In International Conference on Parallel Architectures and Languages Europe, pp. 20-27. Springer Berlin Heidelberg, 1989.

[34] British Standard 1192:2007 – "Collaborative production of architectural, engineering and construction information, Code of

practice". ISBN: 978-0-580-58556-2, British Standards Institute (BSI), January 31, pp. 1–38, 2008, Available online at: http://static1.1.sqspcdn.com/static/f/831240/25111743/1403778513567/BS, Last accessed: February 2016.

[35] UK Cabinet Office, "Government Construction Strategy" (2011), pp.1-43, Last accessed: February 2016.

[36] David Bryde, Marta Broquetas, Jargen Marc Volm, The project benefits of Building Information Modelling (BIM), International Journal of Project Management, Volume 31, Issue 7, October 2013, Pages 971-980, ISSN 0263-7863, http://dx.doi.org/10.1016/j.ijproman.2012.12.001, ISSN 0263-7863.

[37] M. Latham, Constructing the Team: Final Report of the Government/Industry Review of Procurement and Contractual Arrangements in the UK Construction Industry., London: HMSO., pp.140 pages, UK-Department of Environment, 1994.

[38] Barrett, Peter, and Martin Sexton. "Innovation in small, project-based construction firms." British Journal of Management 17.4 (2006): 331-346, http://dx.doi.org/10.1111/j.1467-8551.2005.00461.x, ISSN: 1467-8551.

[39] A. Dainty, D. Moore, M. Murray, "Communication in Construction: Theory and Practice", Abingdon, Oxon: Taylor & Francis, DOI: 10.4324/9780203358641, 2006.

[40] Becerik-Gerber, B., Jazizadeh, F., Li, N., and Calis, G., Application Areas and Data Requirements for BIM-Enabled Facili-

ties Management. J. Constr. Eng. Manage., 2012, 138(3), 431–442, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000433, ISSN:0733-9364.