

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/103921/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Petri, Ioan , Beach, Tom , Rana, Omer F. and Rezgui, Yacine 2017. Coordinating multi-site construction projects using federated clouds. *Automation in Construction* 83 , pp. 273-284. 10.1016/j.autcon.2017.08.011

Publishers page: <http://dx.doi.org/10.1016/j.autcon.2017.08.011>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



# Coordinating Multi-Site Construction Projects Using Federated Clouds

Ioan Petri<sup>a,\*</sup>, Tom Beach<sup>a</sup>, Omer F. Rana<sup>b,\*\*</sup>, Yacine Rezgui<sup>a</sup>

<sup>a</sup>*School of Engineering, BRE Institute of Sustainable Engineering, Cardiff University,  
Wales, United Kingdom*

<sup>b</sup>*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

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.

20 *Keywords:* , Coordination, AEC Projects, Collaboration, CometCloud,  
21 Clouds

---

## 22 **1. Introduction**

23 In the Architecture/ Engineering/ Construction (AEC) industry, projects  
24 are increasingly being undertaken by consortia of companies and individuals,  
25 who work collaboratively for the duration of the project. Such projects are  
26 complex and the consortia members provide a range of skills to the project  
27 from its inception to completion. During this process, various data arti-  
28 facts are also generated that need to be stored and shared between project  
29 members (generally using access control strategies – which limit what can be  
30 accessed at a particular stage of the AEC project lifecycle). The planning,  
31 implementation and running of these AEC industry projects requires the for-  
32 mation of secure Virtual Enterprises (VEs) to enable collaboration between  
33 its members by sharing project information and resources. An important  
34 feature of the consortia is that they are dynamic in nature and are formed  
35 for the lifetime of the project [2]. Members can participate in several con-  
36 sortia at the same time and can join or leave a consortium as the project  
37 evolves. Cloud computing offers an important computing infrastructure to  
38 facilitate the establishment and coordination of such VEs. Cloud comput-

39 ing is expected to enhance capabilities that were generally offered through  
40 services made available over the Internet. As well as remote access, Cloud  
41 computing also provides enhanced security infrastructure including single  
42 sign-on capability, security between consortia members, simple setting up  
43 of networks to support VEs, distribution of computationally intensive jobs  
44 across multiple distributed processors (based on shared information about  
45 available resources) [4]. Each organisation involved in a VE may have access  
46 to its own Cloud computing system (privately managed internally within the  
47 organisation, or acquired through a public provider such as Amazon.com or  
48 Microsoft (via their Azure platform)). As it is unlikely that all members  
49 of a consortium will share the same platform, integration across multiple  
50 platforms is therefore an essential requirement for such VEs to function in  
51 an efficient and reliable manner [1].

52 In the computer science research, various efforts have been proposed to  
53 implement such multi-Clouds with research efforts focusing on Cloud inter-  
54 operability e.g. the Open Cloud Computing Interface (OCCI) efforts at the  
55 Open Grid Forum [6]. OCCI provides an API and a set of protocols to enable  
56 management capability to be carried out across multiple Cloud providers. A  
57 variety of implementations are currently available, in systems such as Open-  
58 Stack and OpenNebula (two open source Cloud platforms). An alternative  
59 approach to interoperability is through the development of specialist gateway  
60 nodes which enable mapping between different Cloud systems and the im-  
61 plementation of specialist gateways to connect different Cloud systems, the  
62 development of a Cloud Operating System (CloudOS) to connect distributed  
63 Clouds (European FP7 “UNIFY” project) to the use of specialist in-network

64 capability to process data in network elements between different end points  
65 (GENICloud [7]). Similarly, on-line sites such as CloudHarmony [8] report  
66 over 100+ Cloud providers that offer capability ranging from storage and  
67 computation to complete application containers that can be acquired at a  
68 price, primarily using service-based access models.

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

81 In this paper, we present the implementation and use of a distributed  
82 Cloud system, based on requirements of the AEC sector. The resulting  
83 clouds for coordination(C4C) framework can support merging and federa-  
84 tion of various models of an infrastructure project from multiple applica-  
85 tions, clouds and/or actors using a secure and robust common interface. The  
86 process is based on BIM (Building Information Modelling) and data stored  
87 by each participant conforms to the IFC(Industry Foundation Classes) data  
88 model. We elaborate on the concept of project information “Issue Status”

89 associated with a project in order to determine issuing party’s status with  
90 responsibility/liability associated and considering the reliance on the data.  
91 Our approach involves the implementation of a logical “shared” space that is  
92 physically distributed across multiple sites involved in the federation. Such a  
93 shared coordination space enables various project members to interact with  
94 each other during the stages of a project. We compare our approach to gen-  
95 eral cloud federation efforts, specifically adapted for the needs of the AEC  
96 industry in Section 2. In Section 3 we present the CometCloud system and  
97 how this system has been used to create the federated cloud framework,  
98 followed by a description of the “Cloud4Coordination” (C4C) system and  
99 the associated Application Programming Interface (API) that makes use of  
100 CometCloud in Sections 4 and 5. In Section 6 we evaluate the C4C system  
101 by devising a project trial based on a real construction project and provide  
102 overall conclusions in Section 7.

## 103 **2. Related work**

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

### 106 *2.1. Related AEC technologies*

107 In the AEC industry the concept of decentralised repositories facilitating  
108 data storage across multiple servers represents an emerging topic. Such de-  
109 centralised environments are currently enabled by specialised software such  
110 as Revit Server [24] and Bentley System’s ProjectWise [23]. In these sys-  
111 tems, data is spread between multiple servers (termed integration and caching  
112 servers in the case of Bentley, and hosts and accelerators for Revit Server).

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

120 In addition to these commercial offerings, the concept of data storage and  
121 collaboration is also a topic of active research in the AEC sector. In their  
122 work on SocialBIM, Das et al.[25] have developed a BIM framework that pri-  
123 marily focuses on modelling the social interactions between stakeholders. The  
124 key development is SocialBIM's ability to allow users to contribute/download  
125 partial BIM models that are then merged/split from a "master" model held  
126 in the SocialBIM cloud system(s). While this ability to work with small  
127 "fragments" of BIMs which are then federated is a key development, the fact  
128 that the end result is still stored in a centralised way in a cloud system will  
129 be of concern to many organisations. Other work in this area includes Munk-  
130 ley et al.[27], who have developed technologies to synchronize data between  
131 Revit Server and an external storage server, enabling external users to see a  
132 read only copy of the Revit (central) model. While this is an interesting way  
133 of allowing increased collaboration using Revit Server, it does not adequately  
134 provide for the dynamic two way collaboration that is often required in an  
135 AEC project i.e. the ability to incorporate the results of other discipline's  
136 work (i.e. the architect, mechanical or electrical engineers) as background  
137 in your own work. Finally, this approach is further limited as it is only able

138 to utilise the Revit proprietary data format. Additionally, Boeykens et al.  
139 [28] have developed a layered client/server approach that provides an event  
140 based communications pool between components embedded into BIM au-  
141 thoring packages. This novel communication approach enables the dynamic  
142 sharing of data between components. However, all data is still stored on a  
143 centralised server that listens to the event based communications and both  
144 saves and injects BIM data into the communications pool as needed. Other  
145 solutions for supporting construction BIM data sharing and interoperability  
146 include IFC ontology and IFC linked data with federated queries [12], seman-  
147 tic linking and semantic web paradigms with orthogonal solution vector [13]  
148 and views modelling [14] where companies work on the same model but with  
149 individual access and views. The key differentiating factor of our work is  
150 the distributed nature of our approach, where the authoritative copy of data  
151 is always stored within a discipline’s own servers and is only federated with  
152 other disciplines when required. Another key differentiating factor is the in-  
153 creased level of dynamic communication that is possible between multiple  
154 disciplines using our approach, i.e. when a single discipline makes updates  
155 that are visible to other disciplines. These updates are automatically propa-  
156 gated to the relevant disciplines, without a need for the other disciplines to  
157 query if any updates have been made.

158 Many of these seemingly decentralised approaches (at least from a user’s  
159 perspective), actually make use of centralised storage and coordination in-  
160 frastructure. This is undertaken to ensure that the centralised system is  
161 adequately protected and managed, and can be monitored for any discrep-  
162 ancies or performance bottlenecks. Existing cloud-based deployments are no

163 different – as they make use of a single, centralised data centre. Our ap-  
164 proach differs from these, in that we recognize that each institution involved  
165 in an AEC project will need to provide their own computing infrastructure,  
166 and more importantly will need to integrate their in-house capability with  
167 data centre based cloud systems that may be operated by other institu-  
168 tions/companies. Our approach therefore makes use of a Peer-2-Peer based  
169 approach, whereby local data centres can be aggregated with those of other  
170 institutions in a seamless manner, but still provide a centralised view on the  
171 data shared by institutions involved in a single AEC project. This is achieved  
172 using the CometCloud system as described in Section 2.2.

### 173 *2.2. Related cloud federated systems*

174 Through the federation of cloud systems it has become possible to connect  
175 local infrastructure providers to a common framework where participants can  
176 exchange data and collaborate. The mechanisms used to support cloud fed-  
177 eration can bring substantial benefits for service providers by offering facil-  
178 ities for accessing global services instead of increasing costs associated with  
179 building new infrastructure (which may not be fully utilized and may only  
180 be needed to support peaks in workload over short time frames). A feder-  
181 ated cloud also enables users to host applications with their cloud provider  
182 of choice – thereby making local decisions about pricing, software libraries/  
183 systems and deployment environments, while still being able to connect to  
184 other computational resources [30, 29, 32]. Various cloud bridging solutions  
185 are now available, such as IBM’s Cast Iron Cloud Integration [10], part of the  
186 Web Sphere suite of tools for developing and deploying applications across  
187 different environments. Cast Iron enables integration, through plug-ins, with

188 a number of IBM products (such as DB2) and systems from other vendors,  
189 such as SAP and Salesforces CRM – thereby enabling integration between  
190 in-house systems and public and private Cloud environments [17]. Many such  
191 systems remain proprietary to particular vendors however and are hard to  
192 customise to particular use scenarios. CometCloud [18] is an open source so-  
193 lution that has been validated in a number of scientific and financial scenarios.  
194 CometCloud has been demonstrated to work alongside specialist computing  
195 environments (such as large scale computing clusters that are part of the US  
196 TeraGrid and XSEDE projects) and public Cloud systems from Amazon (as  
197 described below) [16].

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

### 206 **3. Federation in a BIM context**

207 Collaboration in construction projects can bring together various partic-  
208 ipating companies over the (building construction) lifecycle using different  
209 systems and storage solutions. As part of this, the compatibility, control and  
210 access of data objects created is critical to the success of a project. Currently,  
211 coordination between participants is often a labour intensive manual process

212 and can require a monopoly of software systems to be enforced. A construc-  
213 tion project is a complex undertaking depending on a large number of very  
214 different professions and firms [22, 37]. These firms range from SMEs to  
215 large multinational corporations. Each one of these organisation will partic-  
216 ipate in the construction project for a varying time period and, in that time  
217 period, will contribute different quantities and types of data to the project,  
218 or even contribute no data. As we have previously described, while interest  
219 in cloud based BIM solutions is increasing, there are still many obstacles to  
220 BIM adoption that must be overcome. These include: (a) lack of clarity as to  
221 who owns and is responsible for BIM (b) fragmentation of BIM data across  
222 design and engineering teams and then the contractor and FM companies  
223 and (c) information is not sustained across the lifecycle and is in continuous  
224 danger of being lost due to company mergers or bankruptcy [11, 15]. In re-  
225 sponse to these obstacles we propose the use of an BIM federation overlay  
226 to implement a federated distributed BIM data model within a construction  
227 project.

### 228 *3.1. CometCloud Federation*

229 Through the federation of Cloud systems it has become possible to con-  
230 nect local infrastructure providers to a global marketplace where participants  
231 can transact (buy and sell) capacity on demand. The mechanisms used to  
232 support cloud federation can bring substantial benefits for service providers  
233 by offering facilities for accessing global services instead of increasing costs as-  
234 sociated with building new infrastructure (which may not be fully utilized and  
235 may only be needed to support peaks in workload over short time frames).  
236 More importantly, organisations with spare capacity in the data centre are

237 now provided with a simple way to monetize that capacity by submitting it  
238 to the marketplace for other providers to buy, creating an additional source  
239 of revenue.

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

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

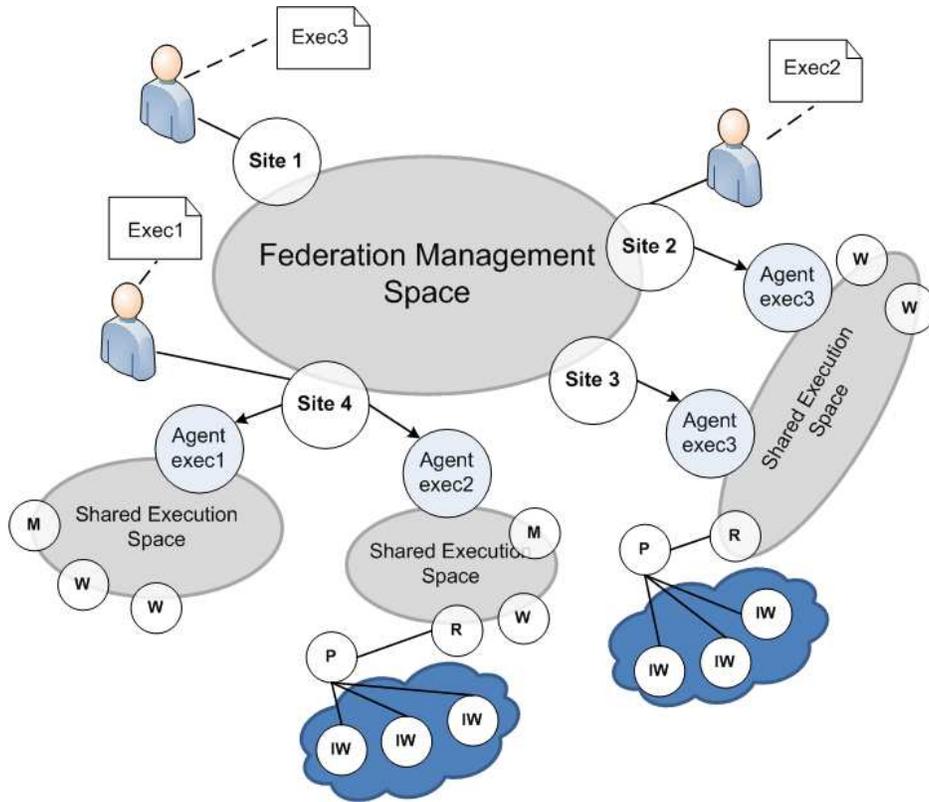


Figure 1: The overall Federation Management Space, here (M) denotes a master, (W) is a worker, (IW) an isolated worker, (P) a proxy, and (R) is a request handler.

262 delegate this role to a dedicated master (M) when some specific functionality  
 263 is required. Moreover, an agent deploys a number of workers to carry out  
 264 execution of tasks. These workers can be in a trusted network and be part  
 265 of the shared execution space, or they can be hosted on external resources  
 266 such as a public cloud and therefore in a non-trusted network. The first  
 267 type of worker is called a “secure worker” (W) and can pull tasks directly  
 268 from the space. Meanwhile, the second type of worker is called an “isolated  
 269 worker” (IW) and cannot interact directly with the shared space. Instead,

270 they have to interact through a proxy (P) and a request handler (R) to be  
271 able to retrieve task information from the space and execute these..

### 272 3.2. *CometSpace*

273 CometCloud uses a Linda-like tuple space [31] referred to as “CometSpace”  
274 which is implemented using a Peer-2-Peer overlay network. A tuple space  
275 enables the implementation of an associative memory-based search strategy,  
276 whereby the search term is described as a set of items/terms, which can be  
277 mapped against a table of stored data. This search strategy is often easier  
278 to implement in hardware and therefore provides a significant improvement  
279 in search performance. As an illustrative example, consider that there are a  
280 group of data producers and consumers, producers post their data as tuples  
281 in the space, and consumers then retrieve data that match a certain pattern.  
282 The producers/consumers only have a reference to where such data items  
283 should be posted/retrieved from, but do not need to know the physical lo-  
284 cation/ storage device for such data items. CometSpace [33] is an extension  
285 to this tuple space-based abstraction, in that the tuple space can be phys-  
286 ically distributed across multiple sites (data centres), and a “logical” space  
287 is produced by combining these physically distributed sites. Each producer/  
288 consumer now accesses the logical space, asynchronously, and does not need  
289 to know the physical location of the site actually hosting the data. For our  
290 needs we have updated the tuple-space mechanisms and the format of tuples  
291 to comply with requirements related to data processing, data sharing and  
292 data storage as identified in the construction sector. Therefore, a tuple be-  
293 comes an array formed of {tuple-id, discipline-id, object-serialised, event-id}.  
294 In this way, a virtual shared space for storing data can be implemented by

295 aggregating the capability of a number of distributed storage and compute  
296 resources [20]. CometCloud therefore provides a scalable backend deploy-  
297 ment platform that can combine resources across a number of different cloud  
298 providers dynamically, often seen as a key requirement for a project in the  
299 AEC sector.

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

#### 311 **4. C4C project**

312 In this section we outline the key industry-based requirements of the  
313 “Clouds-for-Coordination” (C4C) project. We subsequently describe the  
314 CometCloud-based system that has been implemented to address these re-  
315 quirements.

##### 316 *4.1. Project background*

317 The C4C project is addressed to the AEC industry seeking to facilitate  
318 collaboration between organisations and looking at aspects related to BIM

319 data management and sharing. As BIM presents the possibility of sharing  
320 information throughout the construction and property management sectors,  
321 the problem of trust in the data becomes important – more commonly recog-  
322 nised in the AEC industry through the use of ‘Issue Status’ for physical  
323 documents (where documents are given statuses that equate to what they  
324 can be reliably used for, and therefore what the issuing party accepts respon-  
325 sibility and/or liability for). There are regulations in the UK, driven by the  
326 government, to achieve fully collaborative Building Information Modelling  
327 (BIM) (with all project and asset information, documentation and data be-  
328 ing electronic) across the AEC sector [35]. This is an especially challenging  
329 proposition as the successful delivery of a construction project is a highly  
330 complex process; requiring collaboration between designers, suppliers and  
331 facilities managers through a range of design and construction tasks. This  
332 complexity in itself is a key motivation for the use of BIM, with anticipated  
333 financial and time savings offered by its adoption [36]. Other motivating fac-  
334 tors for BIM adoption include: (a) project failure caused by lack of effective  
335 project team integration across supply chains [37, 22], (b) emergence of new  
336 challenging new forms of procurement i.e. Private Finance Initiative, Public-  
337 Private Partnership and the design-build-operate [38, 39], and (c) decreasing  
338 the whole life cost of a building through the adoption of BIM in facilities  
339 management[40].

340 The C4C project addresses the issue of BIM “ownership” by adopting  
341 the approach that each party involved creates and stores (and is responsible  
342 for) their own BIM information, rather than uploading it to a central server.  
343 More specifically our architecture imposes the following key aspects: (i) the

344 ownership of data remains with the discipline that created that data – which  
345 also delegates any updates needed on the data to the discipline ensuring that  
346 there is a consistent view also maintained by the discipline owner; (ii) the  
347 use of a coordination layer to allow other users to transparently view data  
348 and make modification to it; (iii) enable information to be replicated across  
349 multiple disciplines (but remain consistent with the data owner), allowing  
350 for fault tolerance and prevent data loss. Another important aspect of a  
351 management model for BIM data is understanding the data and the stages  
352 (workflow) of an AEC project, in the context of how a BIM model is popu-  
353 lated with data. In order to do this an abstract process has been defined as  
354 the result of our requirements gathering exercise. This process has abstracted  
355 the approaches defined in BS1192a[34].

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

367 *4.2. Project implementation*

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

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

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

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

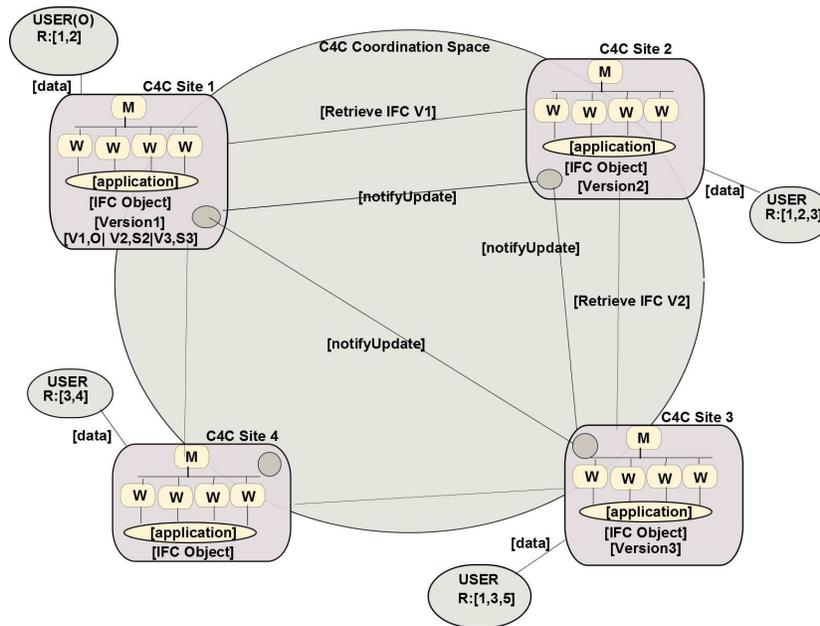


Figure 2: Clouds for coordination multi-site framework.

403 By using the publisher-subscriber model we enable sites to interact with

404 each other on a common project, using publishers to generate project tasks  
405 and subscribers to execute these tasks. We consider the following properties  
406 for a site:

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

414 Each site must support a local C4C environment, which enables other  
415 sites to interact with it. In the workflow presented in Figure 2 and Figure 5,  
416 Site 1 creates the C4C project which is formed of IFC objects locally stored  
417 as *Version1*. All other sites participating in the project (Site 2 and Site 3)  
418 will be notified about the new project being created (based on their roles in  
419 the project). Based on the notification, Site 2 retrieves and updates the C4C  
420 project with *Version1*, Site 2 then creates a new version of the C4C project  
421 as *Version2*. When a new version is created the interested sites are again no-  
422 tified. Site 3 will also retrieve the latest version *Version2* and apply updates  
423 as part of a new project version – *Version3*. Another round of notifications  
424 will be propagated to interested sites (Site 1 and Site 2). Site 4, although  
425 part of the coordination space, is expected to contribute to the project at  
426 later stages thus will not receive a notification event. It is important to note  
427 that Site 1 is the owner of the project, along with the organisation that cre-

428 ates the project and can always retrieve the latest version of the C4C project.  
429 In addition, Site 1 also keeps a list of the changes that have been applied  
430 to the C4C project over time in a “provenance” (metadata) file. In our ex-  
431 ample, Site 1, Site 2, Site 3 and Site 4 have associated suitabilities based  
432 on which they can access the model and have visibility over other disciplines  
433 (can access the objects updated/created by that discipline).

#### 434 *4.3. Computing infrastructure*

435 Our coordination framework can be deployed on infrastructure with vary-  
436 ing capabilities, ranging from regular servers to a cluster infrastructure. To  
437 conduct our test deployments of the C4C system, we utilised IBM Softlayer <sup>1</sup>  
438 virtualized cluster-based infrastructure hosted at IBM’s Amsterdam Data  
439 Centre, utilising dedicated virtual servers. We utilised a total of four sets of  
440 virtualised servers to simulate a construction project with four different dis-  
441 ciplines. These are virtual servers hosted in different physical local locations  
442 within Softlayer (simulating organisations with standard IT infrastructure  
443 and also simulating organisations utilising a cloud based data storage infras-  
444 tructure), allowing us to simulate a life-like scenario where disciplines within  
445 a construction project will utilise multiple IT systems, hosted in differing lo-  
446 cations. In the evaluation, we use a server specification of 16CPU cores with  
447 64GB of memory. The networking infrastructure is 1Gbps Ethernet with a  
448 latency of 14 ms on average. Each server runs Ubuntu 12.4 and Java 7.

---

<sup>1</sup><https://control.softlayer.com/> Last accessed: Aug 2015

## 449 5. C4C Application Programming Interface (API)

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

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

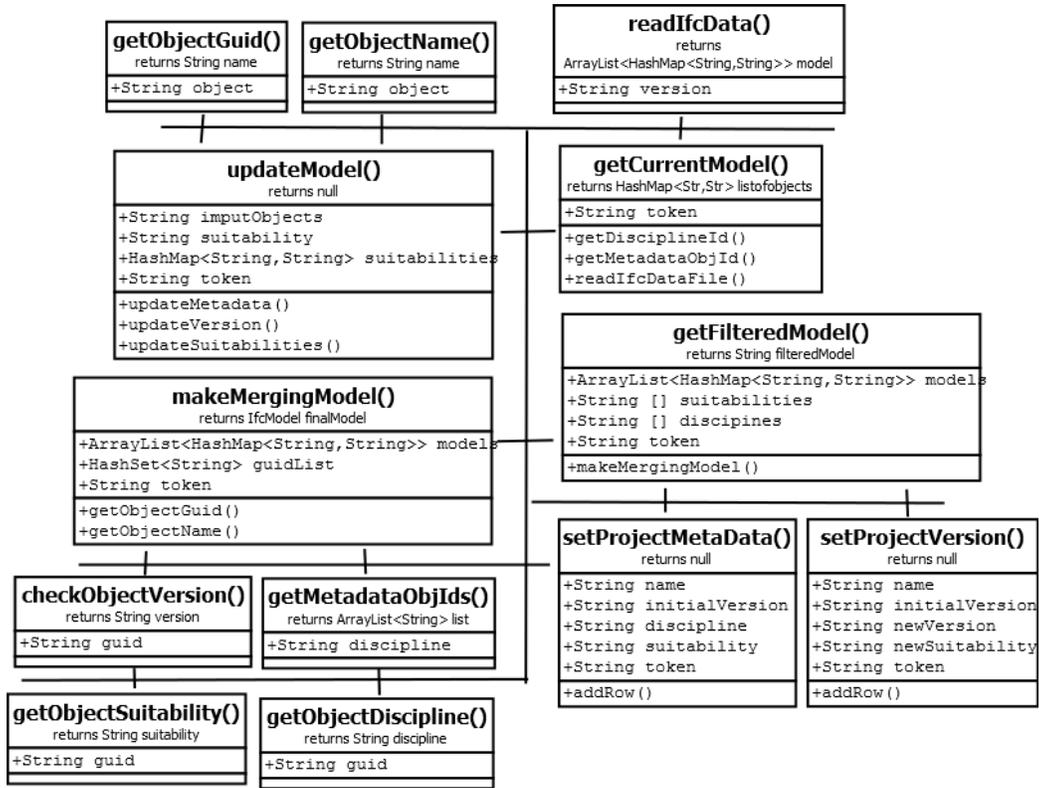


Figure 3: The C4C API

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

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

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

Table 1: Multi-cloud API

483 `latedWorker()`” method. After the multi-cloud entities have been created,  
484 the `C4CManager` starts all the associated Masters and Workers by calling  
485 “`startC4CMasterServer()`” and “`startC4CWorker()`” respectively.

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

490 **tuple-id:** a unique identification of the tuple

491 **discipline-id:** unique identification of the discipline

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

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

## 495 **6. Evaluation**

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

### 506 *6.1. Project trial*

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

- 511 • Contractor - Costain.
- 512 • A cost consultant - Lee Wakemans Ltd.

513     • Designer - Capita.

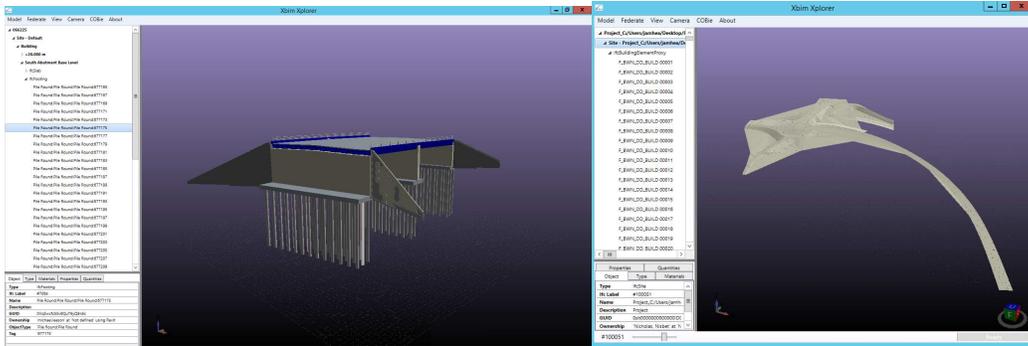
514     • Client - User.

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

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

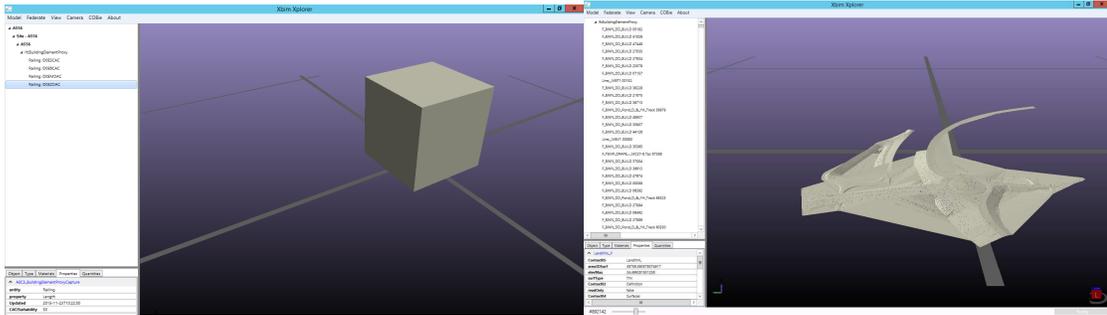
### 534 *6.2. Framework configuration and workflow*

535     The access to the C4C framework is ensured via a user interface devel-  
536 oped 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

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

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

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

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

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

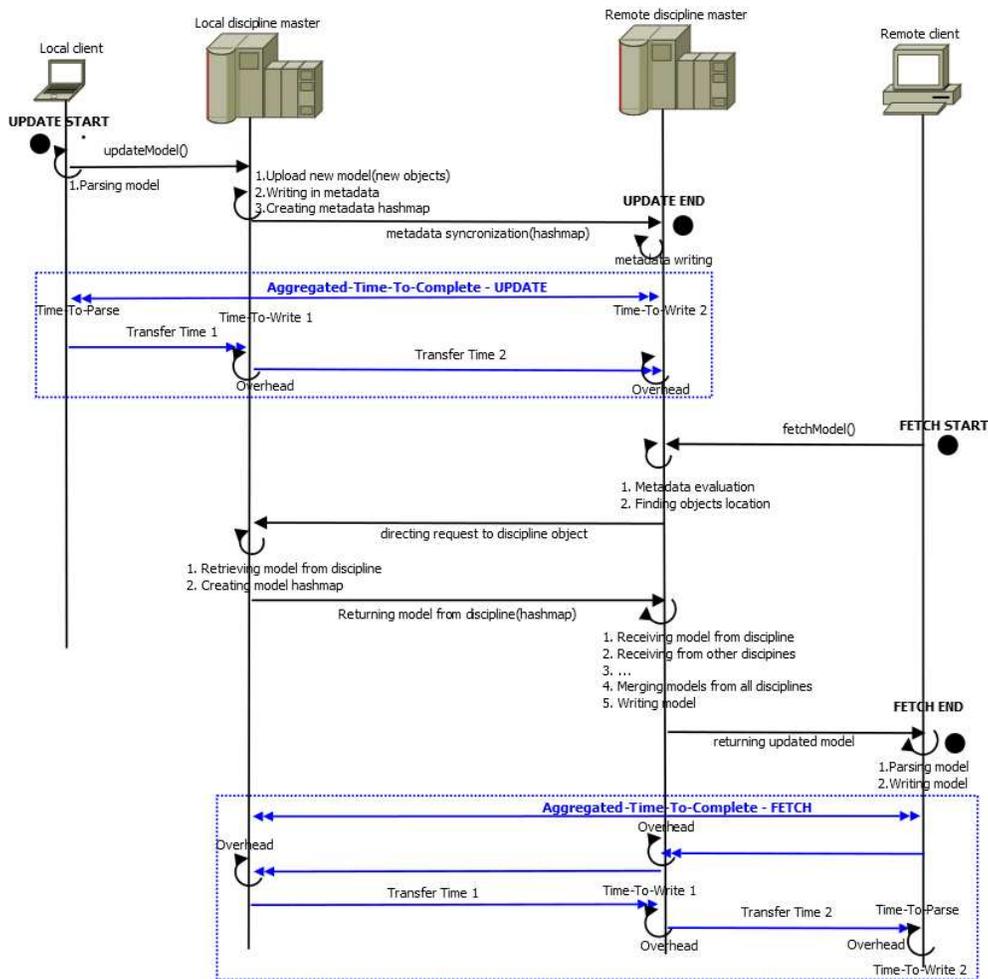


Figure 5: C4C workflow and process sequence

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

551 **Step 5:** Following the definition of the project information, the client (or

552 nominated C4C project manager) moves on to the first configuration table.  
553 This defines the project disciplines (team members) and what information  
554 each discipline can review. The client sends invitations to project disciplines  
555 via email with a link to download the C4C software and the coordinator  
556 server IP address embedded in the email.

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

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

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

565

### 566 *6.3. Trial and validation*

567 In this subsection we explain the entire scenario with participating disci-  
568 plines and iterations that have been followed within the project trial.

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

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

575     • Discipline: Q - Cost consultant - Lee Wakemans Ltd- Connecting to  
576        master node 5.153.52.166

577     • Discipline: O - Client - Connecting to master node 5.153.52.164

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

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

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

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

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

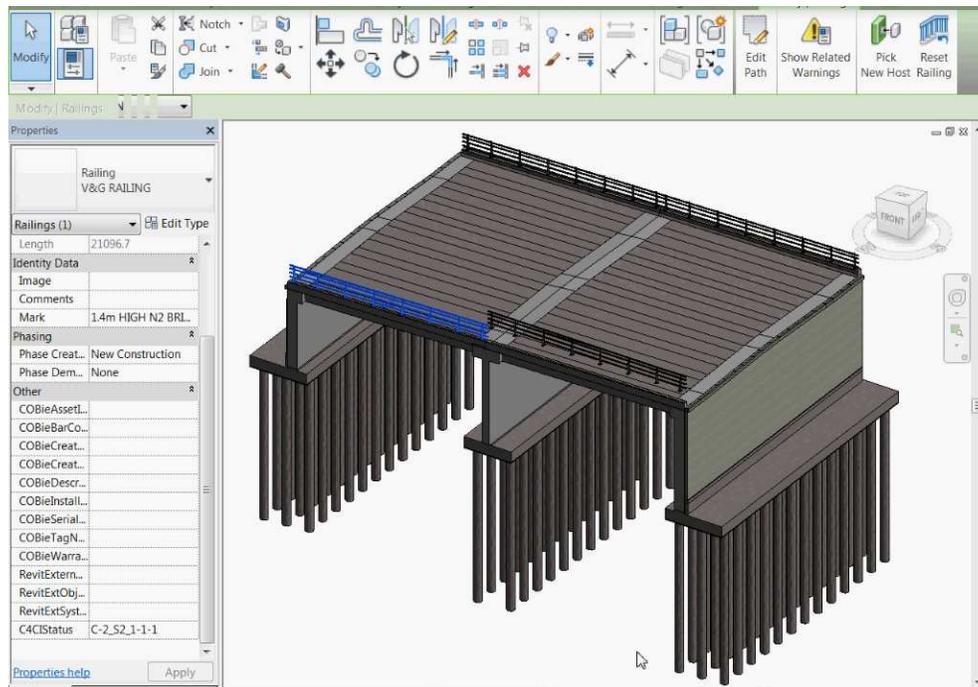


Figure 6: Revit plugin for C4C

600 *6.4. Lessons learnt*

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

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

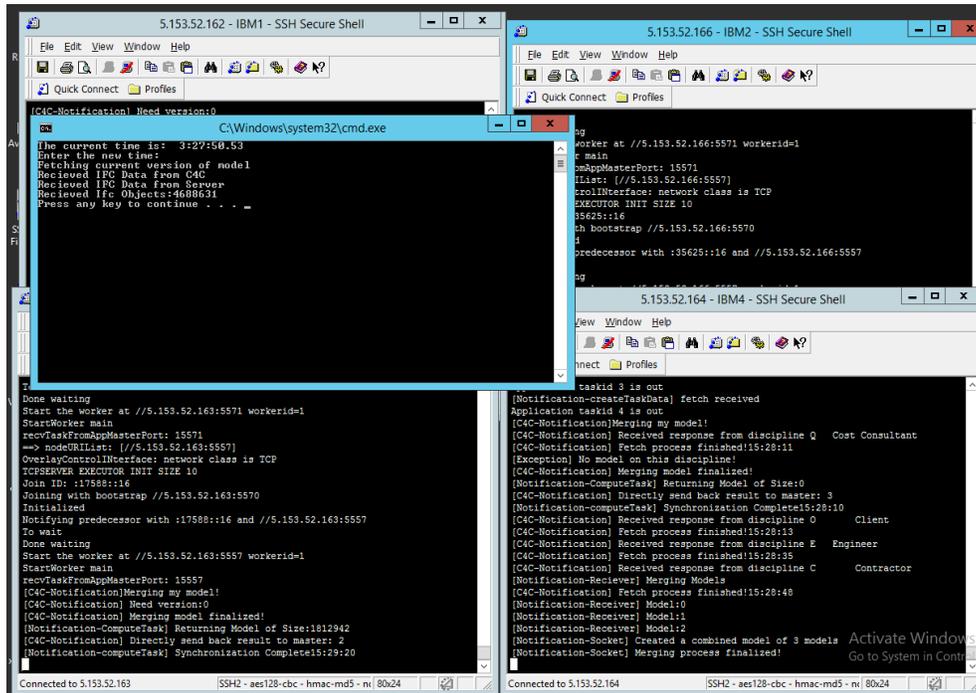


Figure 7: C4C output via terminal

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

619 **Consistency:** Our system can manage federated sub-models and integrate  
 620 such sub-models into a single view. Based in this, a number of benefits can  
 621 be observed related to: (i) detection of issues between models, such as dif-

622 ferences in volumes (clashes) and specification (properties) and groupings  
623 (relationships) and (ii) the creation of a single model by eliminating discrep-  
624 ancies and duplications found in the sub-models.

625 **Trust, Ownership, Flexibility:** In our framework each party stores their  
626 data on, either their own business computer servers, or their choice of ex-  
627 tranet and/or “Cloud” storage in accordance with their own business re-  
628 quirements and protocols. This flexible approach facilitates the federation  
629 of a data model in diverse locations and provides several advantages with  
630 regards to the requirements that exist in a construction project:

- 631 1. Federation is a continuous process, not an event. It proceeds continu-  
632 ously responding to the receipt of updates. At any time the complete  
633 model is potentially available, but so too is the list of outstanding is-  
634 sues.
- 635 2. Access is given to background information as is pertinent to the current  
636 task by role, status and scope and pulled by the agent (who may further  
637 restrict the view by role, status and scope).
- 638 3. Feedback to agents, whether human or automated, is via messages re-  
639 questing clarification, analysis and correction. Examples include clashes,  
640 evaluations, and discrepancies.

641 **IFC limitations:** Over the development of our project we have encoun-  
642 tered several challenges with regards to the overall modeling process and to  
643 efficaciously manage the Industry Foundation Classes (IFCs).

644 The most notable challenges of using this format is the issue of Globally  
645 Unique Identifiers (GUIDs). GUIDs are used by the software to identify and  
646 track objects being processed. In regards to IFC, GUIDs are used to track

647 objects from the BIM dataset and, through this, enable BIM software to know  
648 the origin and revision history of each object within the model. Within the  
649 IFCs, objects that possess a GUID are always a subclass of IfcRoot.

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

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

672 are managed by ensuring that any of these objects are always updated and  
673 replaced when the IFC object (possessing a GUID) that they are associated  
674 to, is updated.

## 675 **7. Conclusion**

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

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

691 Access to data is facilitated through access rights mechanisms, a key  
692 advantage provided by CometCloud that supports a secure and flexible en-  
693 vironment for multi-site construction projects(unlike other Cloud systems  
694 such as OpenStack). The key advantage of our cloud coordination frame-  
695 work represents the near-instant sharing of data between authorised parties

696 in a development project, complete with quality assurance mechanisms and  
697 the ability to track and see a history for the development of any object within  
698 the dataset.

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

## 706 **Acknowledgements**

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

712 [1] Mezgar, Istvan, and Ursula Rauschecker. "The challenge of networked  
713 enterprises for cloud computing interoperability." *Computers in Industry*  
714 65(4), 657-674, 2014, <http://dx.doi.org/10.1016/j.compind.2014.01.017>,  
715 ISSN 0166-3615.

716 [2] Camarinha-Matos, Luis M. "Virtual Enterprises and Collaborative Net-  
717 works (IFIP International Federation for Information Processing), 5th  
718 Working Conference on Virtual Enterprises 22–27 August 2004 Toulouse,

- 719 France, 2004, <https://dx.doi.org/10.1007/b98980>, ISBN: 978-1-4020-  
720 8138-5.
- 721 [3] Beach, Thomas H., et al. "Cloud computing for the architecture, engi-  
722 neering & construction sector: requirements, prototype & experience."  
723 Journal of Cloud Computing: Advances, Systems and Applications 2.1  
724 (2013): 1, <http://dx.doi.org/10.1186/2192-113X-2-8>, ISSN: 2192-113X.
- 725 [4] Rezgui Y, Cooper G, Brandon P: Information management in a collab-  
726 orative multiactor environment: the COMMIT aApproach. J Comput  
727 Civil Eng 1998, 12: 136–145. [http://dx.doi.org/10.1061/\(ASCE\)0887-  
728 3801\(1998\)12:3\(136\)](http://dx.doi.org/10.1061/(ASCE)0887-3801(1998)12:3(136))
- 729 [5] Redmond, Alan, et al. "Exploring how information exchanges can  
730 be enhanced through Cloud BIM." Automation in Construction 24  
731 (2012): 175-183, <http://dx.doi.org/10.1016/j.autcon.2012.02.003>, ISSN  
732 0926-5805.
- 733 [6] OCCI – Open Cloud Computing Interfaces – available at: [http://occi-  
734 wg.org/](http://occi-wg.org/). Last accessed: September 2016.
- 735 [7] GENICloud - available at: <http://groups.geni.net/geni/wiki/GENICloud>.  
736 Last accessed: March 2014.
- 737 [8] Cloud Harmony - available at: <http://cloudharmony.com/>. Last accessed:  
738 March 2014.
- 739 [9] CometCloud Web site - available at:  
740 <http://nsfcac.rutgers.edu/CometCloud/>. Last accessed: March 2014.

- 741 [10] Solution, IBM CastIron. Last accessed May 21, 2016, <https://www-01.ibm.com/software/integration/cast-iron-cloud-integration/salesforce-integration/>.
- 744 [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.
- 748 [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.
- 752 [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>
- 756 [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](http://dx.doi.org/10.1016/0926-5805(92)90014-B).
- 759 [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.
- 762 [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

- 764 view of cloud computing. *Communications of the ACM*, 53(4), pp.50-58,  
765 <http://dx.doi.org/10.1145/1721654.1721672>
- 766 [17] Mell, Peter, and Tim Grance. "The NIST definition of cloud comput-  
767 ing." (2011).
- 768 [18] H. Kim and Parashar, M., *CometCloud: An Autonomic Cloud Engine*,  
769 in *Cloud Computing: Principles and Paradigms*, John Wiley & Sons,  
770 2011, pp. 275-297, <http://dx.doi.org/10.1002/9780470940105.ch10>.
- 771 [19] Hyunjoo Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish  
772 Parashar, Exploring Adaptation to Support Dynamic Applications on  
773 Hybrid Grids-Clouds Infrastructure, 1st Workshop on Scientific Cloud  
774 Computing, in conjunction with the ACM International Symposium on  
775 High Performance Distributed Computing (HPDC), Chicago, Illinois,  
776 June 20-25, 2010.
- 777 [20] Hyunjoo Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish  
778 Parashar. 2009. An Autonomic Approach to Integrated HPC Grid and  
779 Cloud Usage. In *Proceedings of the 2009 Fifth IEEE International Confer-*  
780 *ence on e-Science (E-SCIENCE '09)*. IEEE Computer Society, Washing-  
781 ton, DC, USA, pp.366–373. <http://dx.doi.org/10.1109/e-Science.2009.58>
- 782 [21] M. Latham, *Constructing the Team: Final Report of the Govern-*  
783 *ment/Industry Review of Procurement and Contractual Arrangements*  
784 *in the UK Construction Industry.*, London: HMSO.,UK - Department of  
785 Environment, Stationery Office Books, 1994.

- 786 [22] J. Egan, Rethinking construction, Department of the Environment,  
787 Transport and Regions, The report of the construction task force, De-  
788 partment of Trade and Industry, 2014.
- 789 [23] Bentley ProjectWise - available at:  
790 <https://www.bentley.com/en/products/brands/projectwise>. Last ac-  
791 cessed: March 2016.
- 792 [24] Autodesk Revit Server - available at:  
793 <https://knowledge.autodesk.com/support/revit-products>. Last accessed:  
794 March 2016.
- 795 [25] Das, Moumita, Jack CP Cheng, and Srinath S. Kumar. "Social BIM-  
796 Cloud: a distributed cloud-based BIM platform for object-based lifecycle  
797 information exchange." *Visualization in Engineering v.3*, (8), p. 1-20,  
798 2015, <http://dx.doi.org/10.1186/s40327-015-0022-6>.
- 799 [26] van Berlo, Leon, and Thomas Krijnen. "Using the BIM Collaboration  
800 Format in a Server Based Workflow." *Procedia Environmental Sciences*  
801 *22* (2014): 325-332, <http://dx.doi.org/10.1016/j.proenv.2014.11.031>.
- 802 [27] Munkley, J., Mohamad Kassem, and Nashwan Dawood. "Syn-  
803 chronous Building Information Model-Based Collaboration in  
804 the Cloud: A Proposed Low Cost IT Platform and a Case  
805 Study." *Computing in Civil and Building Engineering*, 89-96, 2014,  
806 <http://dx.doi.org/10.1061/9780784413616.012>.
- 807 [28] Boeykens, Stefan, and Koenraad Nys. "Interactive bi-directional BIM  
808 model and application linking." In *Proceedings of the International As-*

- 809       society for Shell and Spatial Structures (IASS) Symposium 2015, Am-  
810       sterdam Future Visions., pp. 1-11, KIVI, 2015.
- 811 [29] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, et al. Cloud federation  
812       in a layered service model. *J. Comput. Syst. Sci.*, 78(5):1330-1344, 2012,  
813       <http://dx.doi.org/10.1016/j.jcss.2011.12.017>, ISSN 0022-0000.
- 814 [30] Petri, Ioan, Tom Beach, Mengsong Zou, Javier Diaz Montes, Omer  
815       Rana, and Manish Parashar. "Exploring models and mechanisms  
816       for exchanging resources in a federated cloud." 2014 IEEE Interna-  
817       tional Conference on Cloud Engineering (IC2E), pp. 215-224. IEEE,  
818       <http://dx.doi.org/10.1109/IC2E.2014.9>, 2014.
- 819 [31] Carriero, Nicholas, and David Gelernter. "Linda in context." *Communi-*  
820       *cations of the ACM* 32(4) (1989): 444-458.
- 821 [32] Kim, Hyunjoo, Yaakoub el-Khamra, Shantenu Jha, and Manish  
822       Parashar. "An autonomic approach to integrated hpc grid and cloud us-  
823       age." In *e-Science, 2009. e-Science'09. Fifth IEEE International Confer-*  
824       *ence on*, pp. 366-373. IEEE, <http://dx.doi.org/10.1109/e-Science.2009.58>,  
825       2009.
- 826 [33] Gelernter, David. "Multiple tuple spaces in Linda." In *International*  
827       *Conference on Parallel Architectures and Languages Europe*, pp. 20-27.  
828       Springer Berlin Heidelberg, 1989.
- 829 [34] British Standard 1192:2007 – "Collaborative production of ar-  
830       chitectural, engineering and construction information, Code of

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

835 [35] UK Cabinet Office, "Government Construction Strategy" (2011), pp.1-  
836 43, Last accessed: February 2016.

837 [36] David Bryde, Marta Broquetas, Jargen Marc Volm, The project bene-  
838 fits of Building Information Modelling (BIM), International Journal of  
839 Project Management, Volume 31, Issue 7, October 2013, Pages 971-  
840 980, ISSN 0263-7863, <http://dx.doi.org/10.1016/j.ijproman.2012.12.001>,  
841 ISSN 0263-7863.

842 [37] M. Latham, Constructing the Team: Final Report of the Govern-  
843 ment/Industry Review of Procurement and Contractual Arrangements  
844 in the UK Construction Industry., London: HMSO., pp.140 pages, UK-  
845 Department of Environment, 1994.

846 [38] Barrett, Peter, and Martin Sexton. "Innovation in small, project-based  
847 construction firms." British Journal of Management 17.4 (2006): 331-346,  
848 <http://dx.doi.org/10.1111/j.1467-8551.2005.00461.x>, ISSN: 1467-8551.

849 [39] A. Dainty, D. Moore, M. Murray, "Communication in Construc-  
850 tion: Theory and Practice", Abingdon, Oxon: Taylor & Francis, DOI:  
851 10.4324/9780203358641, 2006.

852 [40] Becerik-Gerber, B., Jazizadeh, F., Li, N., and Calis, G., Ap-  
853 plication Areas and Data Requirements for BIM-Enabled Facili-

854 ties Management. J. Constr. Eng. Manage., 2012, 138(3), 431–442,  
855 [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000433](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000433), ISSN:0733-  
856 9364.